

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica



Integrazione di dispositivi personali e grandi schermi negli spazi pubblici

Relatore: Prof.ssa Franca GARZOTTO

Autori:

Fabio MONTI

matr. 782577

Elisabetta A. MORELLI

matr. 782557

Anno accademico 2012/13

Ai nostri genitori

Ai nostri nonni Mario[†] e Luigi[†]

INDICE

Indice delle figure	6
Indice delle tabelle	12
Sommario	13
Abstract	15
1. Introduzione	16
1. Obiettivi della tesi.....	17
2. Struttura della tesi.....	19
2. Stato dell'arte	21
1. I sistemi di raccomandazione.....	21
1.1 Collaborative-based RS.....	23
1.2 Content-based RS	24
1.3 Knowledge-based e Utility-based RS	25
1.4 Demographic-based RS	26
1.5 Community-based RS	26
2. Group Recommendations.....	27
3. Mobile Recommendations	35
3.1 Classificazione.....	36
3.2 Mobile RS nel turismo.....	40
3.3 Esempi di mobile RS	42
4. Elicitation	51
4.1 Explicit elicitation	51
4.2 Implicit elicitation.....	52
4.3 Pro – Contro delle diverse tecniche di elicitation	54
5. Explanation.....	55
5.1 Esempi di explanation	61

6. Interazione tra utente e display pubblico.....	65
6.1 Principi di progettazione per display pubblici.....	66
6.2 Esempio di interazione tra utente e display pubblico.....	67
7. Integrazione di piccoli e grandi schermi	69
7.1 Esempi di integrazione	71
3. Analisi	78
1. Il contesto	78
2. Il dominio.....	79
3. I requisiti dell'applicazione	80
3.1 Versione Android	80
3.2 Database SQLite.....	81
3.3 Connessione Internet	81
3.4 Geolocalizzazione.....	81
3.5 Risoluzione display	81
4. I requisiti del grande schermo	82
4.1 Nomenclatura	82
4.2 Il televisore	82
4.3 Il calcolatore	82
4. Progettazione.....	84
1. Il ruolo di Facebook.....	84
2. L'integrazione di piccolo e grande schermo	88
2.1 WebSockets.....	88
2.2 NFC.....	94
3. Il database centrale	97
3.1 Perché TripAdvisor?	98
3.2 La tecnica di Web Scraping.....	99
3.3 Web Scraping di TripAdvisor.....	100
3.4 Lo schema del database.....	112
3.4.1 Entità "ristorante"	114
3.4.2 Entità "attrazione" – "attività" – "vita notturna"	119
3.4.3 Entità "shopping" – "hotel"	123
4. Il criterio di presentazione dei contenuti.....	124
5. L'infrastruttura completa.....	128
6. Modellizzazione	130
6.1 Scenari	130
6.2 Use case Diagram	134
6.3 Sequence Diagram.....	140
6.4 Activity Diagram	145

5. Prototipo realizzato.....	149
1. L'applicazione	149
1.1 La registrazione	150
1.2 La gestione delle preferenze	153
1.3 La gestione del profilo	156
1.4 La ricerca "Around Me"	157
1.5 La lista dei preferiti	159
1.6 La pagina di dettaglio	161
1.7 Controller.....	163
2. Il grande schermo	165
2.1 Il messaggio di benvenuto	165
2.2 La scelta della categoria.....	166
2.3 La lista di risultati.....	167
2.4 I dettagli dell'elemento	168
3. L'interazione tra utente e grande schermo attraverso il device personale	171
6. Conclusioni	182
1. Problemi riscontrati	184
2. Challenges affrontate.....	185
3. Sviluppi futuri	186
3.1. Scenario 1: gruppi volontari.....	187
3.2. Scenario 2: gruppi casuali.....	188
Appendice A. Manuale d'installazione	191
Appendice B. Software utilizzati	198
A.1. StarUML.....	198
A.2. Microsoft Visio 2010	198
A.3. Eclipse + ADT	199
A.4. Adobe Photoshop	199
A.5. Sublime Text	199
A.6. FileZilla.....	199
A.7. Fiddler	200
A.8. Node.js.....	200
A.9. EasyPhp.....	200
A.10. MySQL Workbench.....	201
A.11. Graph API Explorer.....	201
Ringraziamenti	202
References.....	210

INDICE DELLE FIGURE

Figura 1. Nozione di gruppo in GroupFun.	30
Figura 2. Interfaccia CoFeel per esprimere sentimenti.	31
Figura 3. Fornire feedback emotivo su una canzone.....	32
Figura 4. Interazione con CoFeel in GroupFun.....	32
Figura 5. Visualizzare feedback emotivi su GroupFun.	33
Figura 6. Lasciare commenti emotivi sulla timeline.....	34
Figura 7. Generica architettura di un mobile RS per il turismo.....	41
Figura 8. Esempio di critica in MapMobyRek.....	43
Figura 9. Mappa in MapMobyRek.....	44
Figura 10. Architettura di mTRS.	45
Figura 11. Screenshot significativi di mTRS.	46
Figura 12. Caricamento del contenuto.	47
Figura 13. Specifica dei fattori contestuali.....	48
Figura 14. Lista di item suggeriti.	49
Figura 15. Dettaglio item raccomandato.....	50
Figura 16. Notifica cambiamento condizione contestuale.....	50
Figura 17. Tecniche comuni di explicit elicitation.	52
Figura 18. Feedback esplicito.....	54
Figura 19. Feedback implicito.....	55
Figura 20. Explanation mediante “why”.	56
Figura 21. Explanation nel titolo di ogni categoria.....	57

Figura 22. Explanation mediante testo e grafico.....	57
Figura 23. Explanation mediante “why”.	58
Figura 24. Explanation nel titolo di ogni categoria.....	59
Figura 25. Esempio di explanation con “Why”	62
Figura 26. Finestra pop-up che mostra la spiegazione testuale.....	62
Figura 27. Raccomandazione per utente sigle con budget elevato.	63
Figura 28. Raccomandazione per utente non sigle con budget basso.	63
Figura 29. Dettaglio con spiegazione dell’elemento aggiunto.....	64
Figura 30. Nuova raccomandazione per modifica profilo utente.	64
Figura 31. Fasi dell’interazione con display pubblico.....	66
Figura 32. Bolle in FizzyVis.	68
Figura 33. Interazione con le bolle.	69
Figura 34. Meccanismo di Snap & Grap.	71
Figura 35. Architettura base di PACMEN.....	72
Figura 36. Pacmen game.	73
Figura 37. Piantina del sottopassaggio.....	74
Figura 38. Interazione tra display e utenti.....	75
Figura 39. Design display pubblico.	76
Figura 40. Design applicazione mobile.	77
Figura 41. Statistiche Google.	80
Figura 42. Porzione di risposta JSON.....	86
Figura 43. Impostazione permessi Facebook.....	87
Figura 44. Costruttore WebSocket.	90
Figura 45. Apertura connessione lato client.	91
Figura 46. Eventi WebSocket.	92
Figura 47. Connessione lato server.	92
Figura 48. Connessione lato client.	92
Figura 49. Header dell’handshake.....	93
Figura 50. Previsioni per spedizioni mondiali di telefoni NFC, in milioni.....	97

Figura 51. Fasi del Web Scraping.....	101
Figura 52. Codice per connessione a TripAdvisor – sezione ristoranti.	102
Figura 53. Lista di ristoranti mostrata da TripAdvisor.....	102
Figura 54. Zoom su un ristorante della lista.....	103
Figura 55. Codice per recupero info principali.	103
Figura 56. Codice per connessione pagina di dettaglio ristorante.	104
Figura 57. Pagina di dettaglio ristorante.....	104
Figura 58. Ubicazione ristorante.	104
Figura 59. Codice per recupero info geografiche.....	105
Figura 60. Dettagli aggiuntivi primo tipo.....	105
Figura 61. Dettagli aggiuntivi secondo tipo.	106
Figura 62. Codice per recupero dettagli aggiuntivi.....	106
Figura 63. Codice per passare di pagina in pagina.	107
Figura 64. Codice per invocazione API di Google.....	107
Figura 65. Codice per recupero coordinate geografiche.....	108
Figura 66. Profilo utente in TripAdvisor.	109
Figura 67. Zoom su profilo utente.	109
Figura 68. Codice per recupero utenti.....	110
Figura 69. Esempio di recensione in TripAdvisor.	111
Figura 70. Codice per recupero recensioni.	111
Figura 71. TipiCucina, FasciaPrezzo, Quartieri, Opzioni in TripAdvisor.....	115
Figura 72. Modello E/R – Ristoranti.....	118
Figura 73. Tipo attrazioni in TripAdvisor.	119
Figura 74. Tipo attività in TripAdvisor.....	119
Figura 75. Tipo vita notturna in TripAdvisor.	119
Figura 76. Modello E/R – Attrazioni.....	120
Figura 77. Modello E/R – Attività.	121
Figura 78. Modello E/R – Vita notturna.	122
Figura 79. Modello E/R – Shopping.....	123

Figura 80. Modello E/R – Hotel.....	124
Figura 81. Schema generale dell'infrastruttura.....	130
Figura 82. Use Case Diagram: accesso iniziale.....	134
Figura 83. Use Case Diagram: funzionalità principali.....	135
Figura 84. Use Case Diagram: interazione grande schermo.....	136
Figura 85. Sequence Diagram: registrazione + inserimento preferenze.....	141
Figura 86. Sequence Diagram: accesso con Facebook.....	142
Figura 87. Sequence Diagram: ricerca ristorante + aggiunta preferiti.....	142
Figura 88. Sequence Diagram: navigazione sul grande schermo.....	143
Figura 89. Sequence Diagram: cattura item dal grande schermo.....	144
Figura 90. Activity Diagram: registrazione.....	145
Figura 91. Activity Diagram: ricerca "Around Me".....	146
Figura 92. Activity Diagram: gestione preferiti.....	147
Figura 93. Activity Diagram: interazione grande schermo.....	148
Figura 94. Icona dell'applicazione.....	150
Figura 95. Pagina iniziale.....	151
Figura 96. Registrazione.....	152
Figura 97. Login con Facebook.....	152
Figura 98. Intro home.....	152
Figura 99. Home.....	152
Figura 100. Pattern di navigazione.....	153
Figura 101. Scelta della categoria.....	154
Figura 102. Preferenze sul tipo cucina.....	154
Figura 103. Preferenze sui servizi.....	154
Figura 104. Preferenze sulle attrazioni.....	155
Figura 105. Preferenze sulle attività.....	155
Figura 106. Preferenze sulla vita notturna.....	155
Figura 107. Ordinamento preferenze.....	155
Figura 108. Profilo senza preferenze.....	156

Figura 109. Profilo con preferenze.....	156
Figura 110. Legenda preferenze.....	157
Figura 111. Around Me – Restaurant.....	158
Figura 112. Around Me – POI.....	158
Figura 113. Legenda Around Me.....	158
Figura 114. Mappa Around Me.....	159
Figura 115. Lista preferiti.....	160
Figura 116. Rimuovere preferito.....	160
Figura 117. Aggiungere ai preferiti.....	160
Figura 118. Elemento non preferito.....	161
Figura 119. Elemento preferito.....	161
Figura 120. Scheda informativa.....	162
Figura 121. Legenda scheda informativa.....	162
Figura 122. Recensioni.....	163
Figura 123. Galleria immagini.....	163
Figura 124. Controller.....	164
Figura 125. Messaggio di benvenuto.....	166
Figura 126. Scelta categoria.....	167
Figura 127. Lista di risultati.....	168
Figura 128. Details elemento.....	168
Figura 129. Reviews elemento – prima versione.....	169
Figura 130. Reviews elemento – seconda versione.....	169
Figura 131. Reviews elemento – terza versione.....	170
Figura 132. Images elemento.....	170
Figura 133. Maps elemento.....	171
Figura 134. Fase preliminare all’interazione.....	172
Figura 135. Messaggio di benvenuto – grande schermo.....	173
Figura 136. Caricamento dati da Facebook – applicazione.....	173
Figura 137. Caricamento dati dal profilo – applicazione.....	173

Figura 138. Scelta categoria - grande schermo.....	174
Figura 139. Scelta categoria – applicazione.	174
Figura 140. Attesa lista di risultati – applicazione.....	175
Figura 141. Primi dodici risultati – grande schermo.....	175
Figura 142. Remote controller – applicazione.	176
Figura 143. Details ristorante – grande schermo.	176
Figura 144. Reviews ristorante – grande schermo.....	177
Figura 145. Next reviews ristorante – grande schermo.	177
Figura 146. Img ristorante – grande schermo.....	178
Figura 147. Next img ristorante – grande schermo.	178
Figura 148. Maps iniziale – grande schermo.....	179
Figura 149. Maps spostata – grande schermo.	179
Figura 150. Details ristorante desiderato – grande schermo.....	180
Figura 151. Item aggiunto ai preferiti – applicazione.	180
Figura 152. Canale di comunicazione creato con successo.....	196
Figura 153. Connessione avvenuta con successo.....	197

INDICE DELLE TABELLE

Tabella 1. Caratteristiche algoritmi di raccomandazione.....	22
Tabella 2. Trade-off algoritmi di raccomandazione.....	22
Tabella 3. Risultato dell'esperimento di Modsching.....	42
Tabella 4. Esempi di sistemi suddivisi per modalità di explanation.....	61
Tabella 5. Scenario: registrazione + inserimento preferenze.....	131
Tabella 6. Scenario: accesso con Facebook.....	132
Tabella 7. Scenario: ricerca ristorante + aggiunta ai preferiti.....	132
Tabella 8. Scenario: navigazione sul grande schermo.....	133
Tabella 9. Scenario: cattura item dal grande schermo.....	134
Tabella 10. Use Case: registrazione + inserimento preferenze.....	137
Tabella 11. Use Case: accesso con Facebook.....	137
Tabella 12. Use Case: ricerca ristorante + aggiunta preferiti.....	138
Tabella 13. Use Case: navigazione sul grande schermo.....	139
Tabella 14. Use Case: cattura item dal grande schermo.....	140
Tabella 15. Significato comandi.....	165

SOMMARIO

Negli ultimi anni abbiamo sempre più spesso assistito alla migrazione dei contenuti verso i dispositivi mobili, i quali hanno avuto e stanno tuttora avendo un grande successo e rilevanza nella nostra vita quotidiana.

Parallelamente, la grande quantità di informazioni reperibile sul Web e l'uso massiccio dei social network hanno condotto all'origine del problema noto come "overloading informativo". I sistemi di raccomandazione si inseriscono per contrastare il problema appena descritto e la loro introduzione, specialmente in ambito mobile, comporta nuove sfide da affrontare legate alle caratteristiche intrinseche dei dispositivi, tra cui le limitate dimensioni dello schermo e una modesta capacità di elaborazione dei dati.

Negli ultimi anni sta emergendo quindi un nuovo modo di condurre informazioni, attraverso display di grandi dimensioni collocati in aree metropolitane, stazioni ferroviarie, aeroporti e, più in generale, in spazi pubblici. Se da un lato essi offrono una significativa potenza di calcolo e uno spazio di presentazione maggiore, dall'altro i dispositivi mobile godono di una considerevole capacità interattiva e sono molto famigliari all'utente. L'integrazione tra piccoli e grandi schermi, quindi, si rivela la scelta vincente per sfruttare il meglio di entrambi i mondi.

Obiettivo della tesi è sviluppare un sistema innovativo che permetta di interagire, attraverso un dispositivo mobile, con i contenuti multimediali presentati su un grande schermo. Il sistema sviluppato deve essere in grado di personalizzare tali contenuti rispetto al profilo dell'utente, sfruttando i dati disponibili nei social network.

Il contesto in cui si sviluppa il progetto riguarda l'Expo, per il quale si prevede la partecipazione e il coinvolgimento di milioni di persone che, oltre a visitare il sito espositivo, coglieranno l'occasione per conoscere la città. In quest'ottica, si vuole aiutare i turisti a trovare una risposta a domande del tipo "Stasera dove posso cenare?", "Che cosa posso visitare?", "Quali attività adatte a me propone Milano?". In conclusione, il risultato ottenuto sfrutta l'integrazione tra dispositivo mobile e display pubblico per permettere all'utente di scoprire cosa offre Milano in merito ai suoi gusti e interessi personali.

ABSTRACT

In the last few years, the amount of migration of contents toward mobile devices is deeply increased. Indeed, the big amount of information that are available on web and as well as the widespread use of social networks create the problem, which is called “the information overloading”. The recommender systems are born exactly to thwart this problem even if their introduction involves new challenges. These are mainly related to intrinsic characteristics of smartphones such as small screens sizes and a limited capacity processing.

Recently, a new way to present content emerged, using large screens located in metropolitan areas, railway stations, airports or, generally speaking, in public places. From one side, public displays offer a significantly higher computational power and a larger surface for presentations, on the other side, mobile devices have powerful interaction capabilities and offer a friendly interface to the users. Therefore, the integration between small and large screens is a winning choice to get benefit from both technologies.

The main of this thesis is to develop an innovative system that allows the interaction through a mobile device with media displayed on a large screen. This system must be able to customize these contents towards user’s profiles that are created from available data on social networks. The context in which this innovative system is going to be used is the EXPO that takes place in Milan in 2015. For instance, the main idea is to help tourists either to find where they could have dinner. At the end of the day, the result of this thesis is to create an integration between personal device and public display in order to help user to find what Milan offers them.

INTRODUZIONE

Approdati ormai al Terzo Millennio ci troviamo immersi in una società complessa caratterizzata da rapide trasformazioni, da una molteplicità di orizzonti di valore e da una pluralità di modelli di comportamento. Una società in cui si sono diffusi in modo pervasivo nuovi mezzi di informazione, telematici ed informatici, che hanno contribuito a facilitare la comunicazione e l'integrazione tra culture diverse.

La diffusione dei Social Media ha cambiato profondamente la concezione del Web, portandolo ad essere il punto in cui opinioni e informazioni vengono scambiate, elaborate e condivise in tempo reale. Internet non rappresenta più solo il luogo dove si possono trovare indicazioni tecniche di un prodotto/servizio, bensì è diventato lo strumento di fiducia da consultare alla ricerca di commenti, giudizi e consigli di consumatori più abili ed esperti.

Si è radicalmente modificato il modo di prendere le decisioni: da una scelta individuale basata su proprie convinzioni, si è passati ad una scelta basata sulla condivisione di parole e giudizi appartenenti a utenti della rete informatica. Difatti la quasi totalità delle persone in possesso di un computer, Smartphone o Tablet, consulta il Web prima di acquistare un prodotto/servizio. È sufficiente pensare

all'importanza che si attribuisce a Internet nella scelta del posto in cui cenare o del luogo ideale dove trascorrere il weekend.

L'utente del Terzo Millennio non vuole spendere il proprio tempo in scelte che possono rivelarsi errate, piuttosto vuole essere certo che il servizio sia all'altezza delle sue aspettative e dei suoi interessi. Per questo motivo sono nati siti Web, vedi TripAdvisor, che implementano una logica di raccomandazione finalizzata a ricercare gli elementi più adatti alla persona.

Negli ultimi anni si è sempre più assistito alla migrazione di contenuti verso i dispositivi mobili, che hanno avuto e stanno tuttora avendo un grande successo nella vita quotidiana. La proliferazione di questi dispositivi ha fatto emergere l'esigenza di offrire servizi personalizzati agli utenti, che siano disponibili sempre e dovunque.

1. Obiettivi della tesi

Obiettivo della tesi è sviluppare un sistema innovativo che permetta di interagire, attraverso un dispositivo mobile, con larghe quantità di contenuti multimediali presentati su un grande schermo. Il sistema sviluppato deve essere in grado di personalizzare tali contenuti rispetto al profilo dell'utente, riducendo il sovraccarico informativo attraverso l'uso di tecniche di raccomandazione che sfruttano i dati degli utenti disponibili nei social network.

Il punto di partenza è un'analisi generale dei sistemi di raccomandazione: dal processo di elicitation, utilizzato per inferire le preferenze degli individui, alla tecnica di explanation, adottata per spiegare la motivazione che ha portato il sistema a fornire quei suggerimenti. In quest'ottica, si devono valutare i principali aspetti legati alla Mobile Recommendations, facendo riferimento soprattutto alle caratteristiche intrinseche di dispositivi come Smartphone e Tablet.

La fase successiva prevede la scelta del supporto fisico responsabile di visualizzare la lista di risultati proposta dal sistema. Mentre i display pubblici offrono una significativa potenza di calcolo e uno spazio di presentazione più grande, i

dispositivi mobile godono di una considerevole capacità interattiva. La scelta ricade quindi sull'integrazione di piccoli e grandi schermi, potendo così sfruttare il meglio di entrambi i mondi.

Parallelamente alla fase di analisi, è necessario costruire un database con le informazioni su ristoranti e punti di interesse della città di Milano. Data la sua popolarità, si è optato per il recupero dei dati dal sito TripAdvisor tramite Web Scraping; come specificato anche all'interno dell'elaborato, il database non viene utilizzato per fini commerciali ma semplicemente come collezione di dati per il test del lavoro.

Partendo dallo studio condotto, si è realizzato un prototipo software in grado di aiutare i turisti a rispondere a domande del tipo "Stasera dove posso cenare?", "Che cosa posso visitare?", "Quali attività adatte a me propone Milano?". Il prototipo si compone di un'applicazione mobile da installare sullo Smartphone del visitatore, che dovrà comunicare con uno tra i tanti display pubblici dislocati nella città in occasione dell'Expo. Per permettere questa integrazione si sono sviluppati il front-end e il back-end del grande schermo, il canale WebSocket per aprire la connessione e, infine, l'algoritmo di raccomandazione responsabile di generare la lista di risultati. Come spiegato con maggior dettaglio nel Capitolo 4, in realtà non si tratta di un vero e proprio algoritmo di raccomandazione, quanto piuttosto di un criterio di ranking dei risultati basato sulle preferenze espresse dall'utente e sulla popolarità degli item. Nel corso dell'elaborato talvolta, per evitare l'introduzione di altri termini che potrebbero generare confusione nel lettore, si parla di "raccomandazione" per indicare la logica di presentazione dei risultati.

Il prototipo realizzato vuole essere il meno invasivo possibile nei confronti nell'utente; le ricerche eseguite hanno mostrato che l'installazione di componenti aggiuntivi e l'utilizzo di codici/pin, per instaurare ad esempio la connessione tra Smartphone e grande schermo, complicano la percezione della User-Experience.

In conclusione, il lavoro di tesi permette al turista di risparmiare tempo, potendo conoscere in pochi e semplici passi cosa offre Milano relativamente ai suoi gusti e interessi personali.

2. Struttura della tesi

Il corpo centrale della tesi si articola nei seguenti capitoli:

- *Capitolo 2 – Stato dell'arte*

In questo capitolo si riporta una rassegna della letteratura relativa all'importanza che la raccomandazione sta assumendo nel corso degli anni, soprattutto grazie all'evoluzione del Web e al crescere delle informazioni disponibili. Dopo una panoramica generale in cui vengono toccati argomenti quali elicitation ed explanation, ci si concentra sul processo di raccomandazione in ambito mobile rivolto in particolar modo al settore del turismo. Successivamente si passa a studiare la Group Recommendations utile per capire come, a livello teorico, si possa estendere il lavoro di tesi andando oltre la singola User-Experience. Infine si è presa in considerazione l'integrazione di piccoli e grandi device e, potendo godere di qualche esempio utile, si è tracciata la base per far comunicare l'applicazione con il grande schermo.

- *Capitolo 3 – Analisi*

Questo capitolo si occupa di descrivere l'analisi fatta dagli autori negli stadi primordiali del lavoro. Il capitolo si apre con una panoramica introduttiva che spiega il contesto in cui il progetto di tesi si sviluppa. A seguito di questa premessa, viene illustrato il dominio applicativo andando a individuare le entità chiave e gli stakeholders. Successivamente si passa a stilare i requisiti, sia dal punto di vista dell'applicazione da installare sullo Smartphone/Tablet dell'utente, sia dal punto di vista del grande schermo.

- *Capitolo 4 – Progettazione*

La progettazione è una fase molto importante nel ciclo di vita di un software, in quanto permette la definizione delle linee guida per la struttura del prototipo. All'interno del capitolo si discute quindi delle tecnologie utilizzate per la comunicazione tra piccolo e grande schermo, e si ripone particolare

attenzione alla costituzione della base di dati. Quest'ultima infatti verrà interrogata dall'algoritmo per prelevare ristoranti e punti di interesse che si adattano agli interessi dell'utente. Si inserisce inoltre uno schema generale dell'infrastruttura del progetto, per far capire al lettore le componenti essenziali e il loro ruolo all'interno del lavoro. Il capitolo si conclude mostrando gli scenari principali descritti con i classici diagrammi forniti dall'UML.

- *Capitolo 5 – Prototipo realizzato*

Questo capitolo ha lo scopo di presentare il risultato finale del lavoro di tesi. Si parla inizialmente dell'applicazione mobile da installare sullo Smartphone/Tablet dell'utente, mettendo in luce le sue funzionalità. Si passa poi a mostrare l'interfaccia del grande schermo, spiegando le scelte adottate dagli autori. Infine si descrive l'interazione tra utente e display pubblico attraverso il device personale; in questo paragrafo si inserisce uno scenario che permette al lettore di comprendere le potenzialità del lavoro svolto. Durante la trattazione vengono presentati alcuni screenshots significativi, sia per quanto riguarda l'applicazione mobile che per la parte front-end del grande schermo.

- *Capitolo 6 – Conclusioni*

Si tratta del capitolo finale che racchiude in sintesi il lavoro di tesi, mettendo in evidenza i risultati conseguiti e gli sviluppi futuri. In particolare si trova la sfida della Group Recommendations che ancora oggi non trova una vera risposta. L'idea è quella di esporre le ipotesi formulate dagli autori, offrendo agli sviluppatori futuri una base e uno spunto per il loro lavoro.

STATO DELL'ARTE

Questa sezione mostra una panoramica dello stato dell'arte relativo ai temi coinvolti nello sviluppo del lavoro di tesi. Per maggiori dettagli si rimanda il lettore a prendere visione degli articoli citati nelle References.

1. I sistemi di raccomandazione

Secondo la psicologia sociale, l'innata tendenza dell'essere umano ad utilizzare al minimo le capacità cognitive è giustificata da una sorta di legittima difesa nei confronti della sovrabbondanza di informazioni a cui si trova esposto. In quest'ottica trovare gli oggetti di interesse è un problema di ingente importanza vista la sempre crescente quantità di informazioni, e ciò ha portato alla ribalta il problema di isolare gli elementi utili. Nella vita capita spesso di affrontare situazioni simili; ad esempio, quando si cerca di raccogliere opinioni e consigli da parte di persone ritenute affidabili e che possibilmente hanno gusti compatibili ai nostri.

Negli ultimi anni, i sistemi di raccomandazione (Recommender System o RS) hanno mostrato un grande potenziale per aiutare gli utenti nel processo decisionale in merito a questioni relative al gusto personale.

Per la classificazione di tali sistemi, si introduce la seguente notazione [1]: U è l'insieme di utenti le cui preferenze sono note e $u \in U$ è l'utente per cui deve essere generata la raccomandazione, I è l'insieme di elementi che possono essere consigliati e $i \in I$ è un elemento per il quale vorremmo prevedere la preferenza di u . Prima di passare alla descrizione dei vari algoritmi utilizzati, si fornisce un overview che evidenzia i principali aspetti di ognuno, compresi i pro e i contro delle varie soluzioni (Tabella 1 e Tabella 2).

Technique	Background	Input	Process
Collaborative	Ratings from U of items in I .	Ratings from u of items in I .	Identify users in U similar to u , and extrapolate from their ratings of i .
Content-based	Features of items in I	u 's ratings of items in I	Generate a classifier that fits u 's rating behavior and use it on i .
Demographic	Demographic information about U and their ratings of items in I .	Demographic information about u .	Identify users that are demographically similar to u , and extrapolate from their ratings of i .
Utility-based	Features of items in I .	A utility function over items in I that describes u 's preferences.	Apply the function to the items and determine i 's rank.
Knowledge-based	Features of items in I . Knowledge of how these items meet a user's needs.	A description of u 's needs or interests.	Infer a match between i and u 's need.

Tabella 1. Caratteristiche algoritmi di raccomandazione.

Technique	Pluses	Minuses
Collaborative filtering (CF)	A. Can identify cross-genre niches. B. Domain knowledge not needed. C. Adaptive: quality improves over time. D. Implicit feedback sufficient	I. New user ramp-up problem J. New item ramp-up problem K. "Gray sheep" problem L. Quality dependent on large historical data set. M. Stability vs. plasticity problem
Content-based (CN)	B, C, D	I, L, M
Demographic (DM)	A, B, C	I, K, L, M N. Must gather demographic information
Utility-based (UT)	E. No ramp-up required F. Sensitive to changes of preference G. Can include non-product features	O. User must input utility function P. Suggestion ability static (does not learn)
Knowledge-based (KB)	E, F, G H. Can map from user needs to products	P Q. Knowledge engineering required.

Tabella 2. Trade-off algoritmi di raccomandazione.

1.1 Collaborative-based RS

I sistemi di raccomandazione di tipo collaborativo vanno a identificare punti in comune tra gli utenti sulla base delle valutazioni da loro espresse.

L'input del motore di raccomandazione è l'insieme delle valutazioni eseguite dall'utente u sugli oggetti presenti in I . Questi giudizi vengono utilizzati per identificare gli individui in U che sono simili ad u ; in altre parole, l'algoritmo crea per ogni personaggio una "neighborhood" di persone aventi gusti simili, e suggerisce elementi che piacciono ai "neighbors" ma che l'utente non ha ancora visto.

Il punto di forza del filtraggio collaborativo si ritrova nella serendipity, intesa come la sensazione che si prova quando si scopre una cosa imprevista mentre se ne sta cercando un'altra. Ad esempio, si potrebbe proporre ad un amante della fantascienza un film d'arte poiché altri appassionati di fantascienza lo hanno trovato interessante.

La principale difficoltà di questo approccio si incontra quando lo spazio di voti è scarso, ossia vi sono pochi utenti che hanno giudicato lo stesso articolo. Ciò suggerisce che la tecnica collaborativa risulta più adatta alle situazioni in cui la densità di interesse dell'individuo è relativamente elevata rispetto all'universo di elementi disponibili. In altre parole, il successo di questo algoritmo dipende strettamente dalla disponibilità di una massa critica di utenti! A titolo di esempio si potrebbe considerare il caso di un'applicazione che raccomanda film. In essa possono esserci pellicole che sono state valutate da poche persone, e che pertanto verranno raccomandate molto raramente pur avendo ricevuto valutazioni molto positive.

Un'altra problematica da tener presente riguarda il fatto che se una persona ha gusti piuttosto particolari rispetto al resto della popolazione, la sua "neighborhood" sarà presumibilmente piccola con la conseguenza di una raccomandazione povera.

1.2 Content-based RS

I sistemi di raccomandazione basati sul contenuto vanno a identificare punti in comune tra i vari elementi, e propongono una lista di risultati ritenuti simili a quelli che l'utente ha valutato positivamente in passato.

L'input del motore di raccomandazione è l'insieme delle valutazioni eseguite dall'utente u sugli oggetti presenti in I . Questi giudizi sono utilizzati per generare un classificatore che si adatta al comportamento di u e che verrà testato su i : metodo di somiglianza. Affinché tale classificatore possa essere ritenuto affidabile, si deve collezionare un numero sufficiente di valutazioni!

Il problema principale di questo approccio è legato al fatto che le raccomandazioni verso l'utente sono limitate agli elementi ritenuti simili. Ma non sempre è una buona idea proporre tutti i film di Woody Allen ad un individuo che ha mostrato interesse verso uno solo di questi: problema di overspecialization.

Inoltre bisogna considerare che l'algoritmo può lavorare solo con caratteristiche esplicitamente associate agli elementi. Riprendendo l'esempio dell'applicazione che raccomanda film, si tenta di capire la relazione che intercorre tra le pellicole osservando solo con le informazioni disponibili: attori, registi, generi, contenuti speciali, e così via.

Contrariamente a quanto riportato sopra, i sistemi di tipo collaborativo si affidano ai giudizi espressi dagli utenti, e possono essere utilizzati per consigliare oggetti senza guardare alcun dato descrittivo. E permettono altresì di effettuare uno studio incrociato tra le preferenze dell'utente! Ad esempio, può emergere che ascoltatori amanti del free jazz trovino piacevole la musica classica; ovviamente un sistema di tipo content-based non riesce a cogliere questa sottile sfumatura!

Un problema comune ad entrambe le tecniche è il cosiddetto "portfolio effect", che descrive la situazione indesiderata in cui si raccomanda uno stock di elementi che l'individuo già possiede o, riprendendo l'esempio dei film, che ha già visto. Ma d'altro canto, in settori quali news filtering questo ragionamento decade in quanto ci possono essere storie apparentemente molto simili a quelle già lette ma che, guardando con più attenzione, presentano nuove prospettive utili.

1.3 Knowledge-based e Utility-based RS

I sistemi di raccomandazione di questo tipo non tentano di costruire generalizzazioni a lungo termine sugli utenti, piuttosto cercano di trovare una corrispondenza tra l'insieme delle opzioni disponibili e le necessità dell'individuo.

L'input del motore di raccomandazione di tipo knowledge è una descrizione dei bisogni e delle esigenze dell'utente u , che verrà utilizzata per trovare una corrispondenza tra l'elemento i e i requisiti di u .

Per recuperare le esigenze dell'individuo, si può procedere attraverso tecniche di inferenza, oppure semplicemente chiedendo alla persona di fornire una mappatura tra i suoi bisogni e le caratteristiche dei prodotti che desidera.

Nonostante l'inconveniente legato alla difficoltà di acquisire questa conoscenza iniziale, i sistemi knowledge-based sono ottimi per le esplorazioni casuali. Esse infatti sono prive di quel periodo iniziale di avvio, durante il quale i suggerimenti forniti potrebbero essere di bassa qualità a causa della scarsa conoscenza dell'individuo.

L'input del motore di raccomandazione di tipo utility è una funzione di utilità sugli elementi presenti in I , che ha il compito di descrivere le preferenze dell'utente u . Questa funzione viene applicata agli elementi presenti in I per determinare quelli che maggiormente rispondono alle necessità di u .

L'idea alla base degli algoritmi utility-based è fornire una raccomandazione sulla base dell'utilità che ogni oggetto ha per il consumatore. Naturalmente il problema centrale di questo approccio è legato alla definizione di una funzione di utilità idonea per ogni individuo!

Il vantaggio derivante dall'utilizzo di questi sistemi risiede nella possibilità di incorporare diversi fattori nel calcolo del valore del prodotto; ad esempio per un bene acquistato si può tener conto del tempo di consegna, delle condizioni di garanzia, e così via.

1.4 Demographic-based RS

I sistemi di raccomandazione di tipo demografico hanno lo scopo di classificare l'utente sulla base di determinate caratteristiche personali. Ad esempio, appartengono a questa categoria gli algoritmi che permettono di inoltrare un individuo verso una certa pagina Web in base alla lingua madre o al paese di origine.

L'input del motore di raccomandazione sono le informazioni demografiche relative all'utente u . Queste informazioni vengono utilizzate per identificare gli utenti in U che sono demograficamente simili a u .

Il vantaggio di questo approccio sta nel riuscire a generare la raccomandazione senza richiedere la lista delle valutazioni effettuate dall'individuo, indispensabile invece nelle tecniche di tipo collaborativo e content-based. La raccolta delle informazioni demografiche rappresenta però il problema principale di questo algoritmo, soprattutto per la sensibilità al tema della privacy che si fa ogni giorno sempre più delicato.

1.5 Community-based RS

I sistemi di raccomandazione community-based sono conosciuti con il nome di Social Recommender Systems, e si basano sul network di relazioni sociali di cui l'utente fa parte.

Il principio logico alla base di questi algoritmi recita così: l'individuo tende a seguire i consigli di un amico, seppure dissimile, piuttosto che ascoltare i suggerimenti provenienti da uno sconosciuto avente un profilo del tutto analogo al suo. In alcuni casi questi sistemi hanno portato ad ottimi risultati, tuttavia risentono ancora della loro giovane età!

2. Group Recommendations

Come già introdotto nel paragrafo precedente, il processo decisionale è un momento delicato a causa del sovraccarico di informazioni a cui l'individuo si trova esposto. La maggior parte dei sistemi di raccomandazione sono progettati per fornire suggerimenti ad un singolo utente ma, in alcune circostanze, gli elementi suggeriti potrebbero essere destinati non più all'uso personale ma ad un gruppo di utenti.

In generale con la parola "gruppo" si intende un insieme di persone che interagiscono le une con le altre. Nella vita quotidiana si può pensare agli amici che guardano un film insieme, ai colleghi che lavorano su un progetto comune, ai soci che pianificano gli eventi della loro azienda, e così via.

Le strategie utilizzate per fare raccomandazioni di gruppo (dall'inglese Group Recommendations) sono sostanzialmente due. La prima, chiamate **Merging Profiles**, consente la creazione di un profilo di gruppo combinando le preferenze individuali dei diversi membri; sulla base del profilo generato viene effettuata la raccomandazione. La seconda, detta **Merging Recommendations**, consiste nel generare una lista di raccomandazione per ogni partecipante e aggregare tali liste in una unica da indirizzare al gruppo.

Nell'analizzare l'efficacia delle raccomandazioni di gruppo, Baltrunas ha condotto un esperimento basato sull'aggregazione delle liste prodotte da un sistema di filtraggio collaborativo [2]. I risultati ottenuti hanno confermato sperimentalmente una credenza comune: più gli utenti sono simili tra loro, più saranno soddisfacenti le raccomandazioni presentate al gruppo al quale appartengono.

Grazie al suo esperimento, si sono raggiunte alcune importanti conclusioni. In primo luogo, le raccomandazioni individuali risultano leggermente meno efficaci se vengono paragonate a quelle rivolte a gruppi di modeste dimensioni (tipicamente 2, 3 o 4 persone). Secondariamente, l'efficacia delle raccomandazioni di gruppo non necessariamente diminuisce quando la dimensione del gruppo cresce; e in particolare si è dimostrato che questo non avviene per gruppi di utenti simili tra loro. Infine, nei casi in cui gli elementi consigliati risultino inadeguati per l'utente, aggregando i risultati in una lista ordinata e fornendo una raccomandazione per il

gruppo a cui egli appartiene, si riesce ad aumentare discretamente il potere della raccomandazione.

Nonostante i risultati ottenuti, valutare l'efficacia delle raccomandazioni di gruppo rimane un aspetto critico poiché prevede un confronto tra i suggerimenti prodotti e le reali preferenze dei membri. Una soluzione potrebbe essere l'utilizzo delle interviste, durante le quali si interrogano i partecipanti per sapere se si ritengono soddisfatti degli articoli consigliati. Purtroppo tale approccio deve tener conto del fenomeno noto come "contagio emotivo", per il quale la soddisfazione di un individuo è probabile dipendere da quella di altri individui facenti parte del gruppo.

Volendo approfondire la nozione di "gruppo", si può adoperare la seguente classificazione [3]:

- **Gruppi istituiti**

Sono composti da un numero di persone che esplicitamente scelgono di essere parte di un gruppo, in quanto condividono interessi a lungo termine. Essi sono designati per domini di raccomandazione con items di tipo intrattenimento/cultura (libri, musica, film) o riguardanti documenti (pagine Web, documenti di conferenze).

A titolo di esempio si citano *Jukola* e *PartyVote*, due sistemi in grado di fornire musica ad un gruppo di persone che frequentano un party/evento sociale. Questi sistemi lavorano senza alcun user profile, e ad ogni utente è permesso esprimere preferenze in una collezione di musica digitale selezionando una canzone, un album, un'artista o un genere. Il resto del gruppo vota il brano selezionato, facendo sì che ad ogni canzone corrisponda un peso/percentuale. Il brano con il peso maggiore sarà quella che verrà suonato al party.

- **Gruppi occasionali**

Sono composti da un numero di persone che fa qualcosa insieme occasionalmente, ossia condivide uno scopo in un particolare momento. I

sistemi di Group Recommendations che lavorano in questo ambito sono stati sviluppati per i seguenti domini: film, destinazioni turistiche e programmi TV.

A titolo di esempio si cita *Pocket RestaurantFinder*, un sistema progettato per contesti come le conferenze dove un gruppo occasionale di persone vuole cercare un ristorante dove cenare. Ogni utente compila un profilo con le preferenze sui ristoranti, come la fascia di prezzo o il tipo di cucina che privilegia. Una volta nota la composizione del gruppo, il sistema stima la preferenza dell'utente per ogni ristorante; il voto medio assegnato costituisce la base per produrre la lista di item raccomandati.

- **Gruppi casuali**

Sono composti da un numero di persone che condivide un ambiente in un particolare momento, senza espliciti interessi in comune. I domini prevalentemente coinvolti in questa classificazione sono: elementi multimediali (musica, video) ed elementi di informazione (notizie Web). Nel seguito verranno citati e brevemente presentati alcuni esempi di applicazioni che lavorano per gruppi casuali.

In-Vehicle Multimedia Recommender è un sistema che mira a selezionare elementi multimediali per un gruppo di persone che viaggiano insieme. Il sistema raccoglie e unisce i profili dei passeggeri, per effettuare un confronto tra gli item e le preferenze del gruppo.

Flytrap è un sistema che seleziona musica da riprodurre in una sala pubblica. Dal momento in cui le persone all'interno della stanza cambiano frequentemente, il sistema è configurato per prevedere il brano da riprodurre considerando le preferenze degli utenti presenti in sala al momento della selezione del brano. Un agente di "Virtual DJ" viene utilizzato per decidere automaticamente il brano da suonare; egli analizza i file MP3 suonati da un utente nel suo computer e considera le informazioni disponibili sulla musica (genere, artista, e così via).

Per concludere la trattazione, si propone un esperimento condotto da Chen e alcuni suoi collaboratori riguardante il ruolo delle emozioni nel processo di raccomandazione [4].

Lo studio condotto si focalizza principalmente sul determinare il ruolo delle informazioni emotive, progettando un'interfaccia che sia piacevole e facile da usare. A questo scopo è stato introdotto GroupFun, un sistema di raccomandazione mobile in grado di proporre brani musicali ad un gruppo di persone. Gli utenti possono creare gruppi e condividere i propri gusti con gli altri membri (Figura 1).

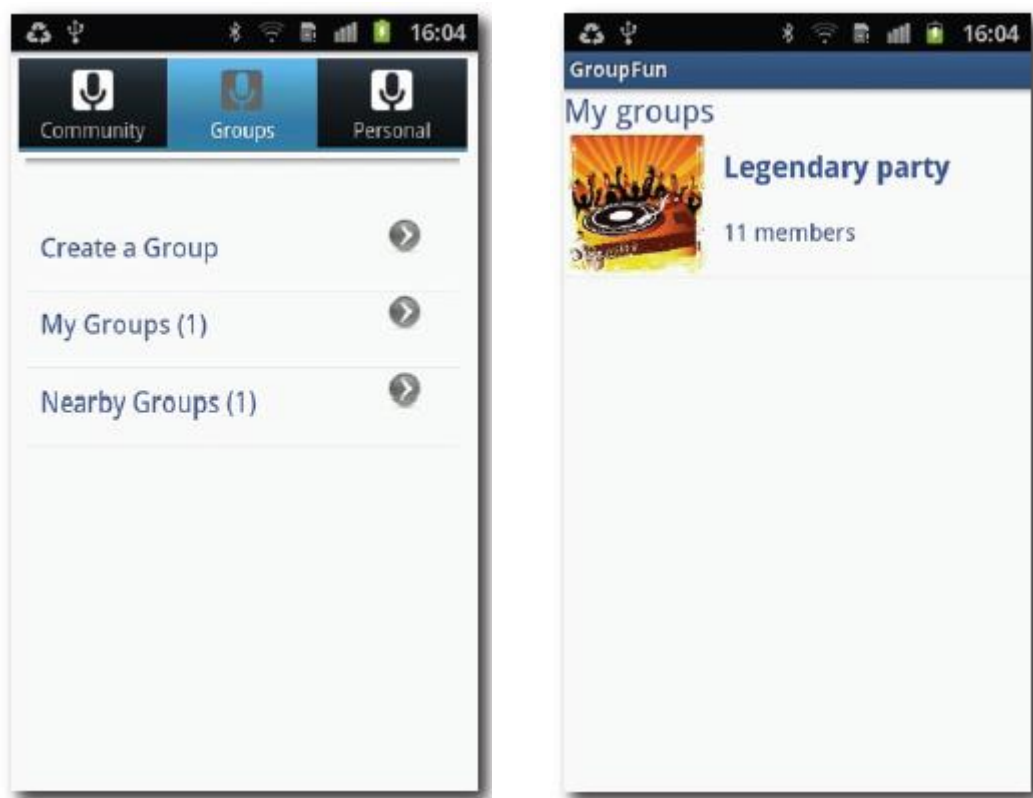


Figura 1. Nozione di gruppo in GroupFun.

Ai fini dell'esperimento è stato introdotto CoFeel, un sistema che permette all'individuo di esprimere il sentimento provato durante l'ascolto di un brano musicale. È composto da otto stati emozionali differenti; per facilitarne la visualizzazione, ad ognuno di essi viene associato un diverso colore. I cerchi

costituiscono una sorta di scala, in rappresentanza dell'intensità del sentimento provato dalla persona (Figura 2).

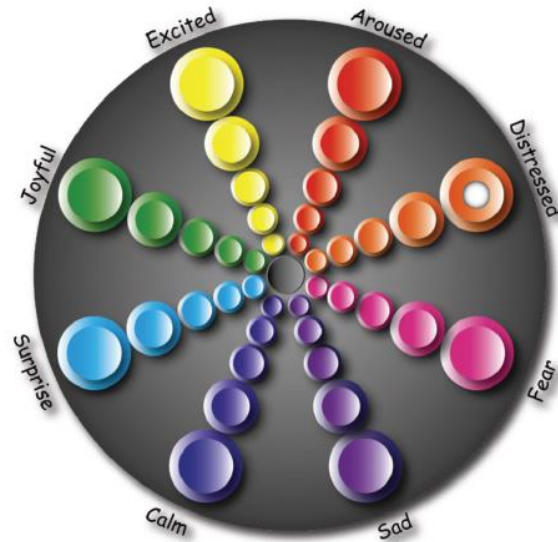


Figura 2. Interfaccia CoFeel per esprimere sentimenti.

A questo punto, CoFeel viene fatto interagire con il sistema di raccomandazione di GroupFun per cercare un miglioramento nell'esperienza di gruppo. In particolare, si è cercato di stimolare le persone a fornire feedback emotivi su una canzone, attraverso lo scambio di sentimenti con gli altri partecipanti.

Per esprimere il sentimento provato, l'utente seleziona un'emozione tra quelle presenti nell'interfaccia di CoFeel e, inclinando lo Smartphone, ne attribuisce l'intensità (Figura 3 e Figura 4).



Figura 3. Fornire feedback emotivo su una canzone.

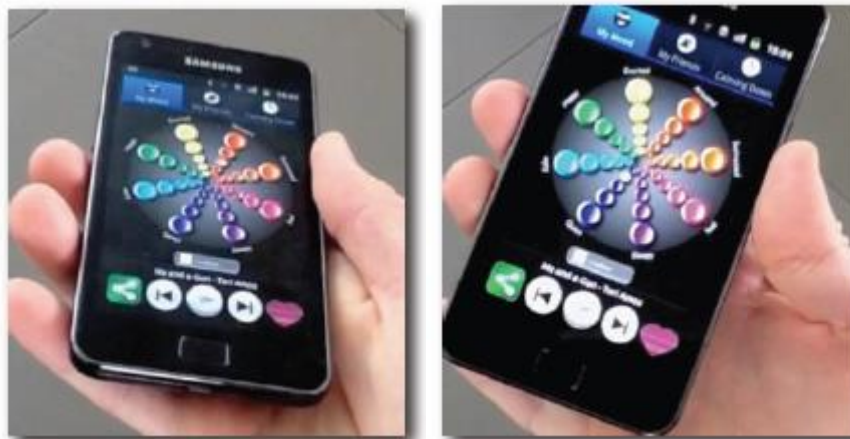


Figura 4. Interazione con CoFeel in GroupFun.

A questo punto, il sistema salva la canzone con il relativo feedback (Figura 5). I pallini a destra del titolo indicano il tipo e l'intensità dell'emozione che il brano ha suscitato nella persona; il colore corrisponde a quello presente nell'interfaccia CoFeel, l'intensità è invece rappresentata dalla trasparenza dei cerchi. Nell'esempio mostrato in figura, la canzone "We will rock you" è classificata come emozionante a livello 3 su 5.

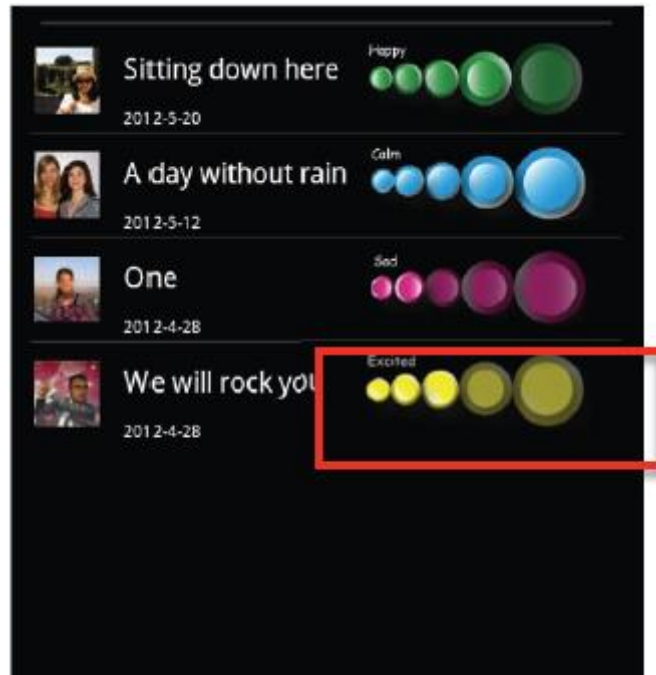


Figura 5. Visualizzare feedback emotivi su GroupFun.

Gli utenti possono inoltre dare i loro feedback emotivi sulla timeline di un canzone (Figura 6). I brani vengono visualizzati sotto forma di spartito musicale, e le emozioni rappresentate con colori diversi sempre grazie a CoFeel. L'intensità è data dalla linea sulla quale il pallino viene posto, e la posizione del punto colorato rappresenta il momento in cui l'utente lascia il feedback. Nell'esempio mostrato in figura, sullo spartito della canzone "Paradise" dei Coldplay i pallini rossi alla fine del brano rappresentano uno stesso sentimento, suscitato però con diversi livelli di intensità.

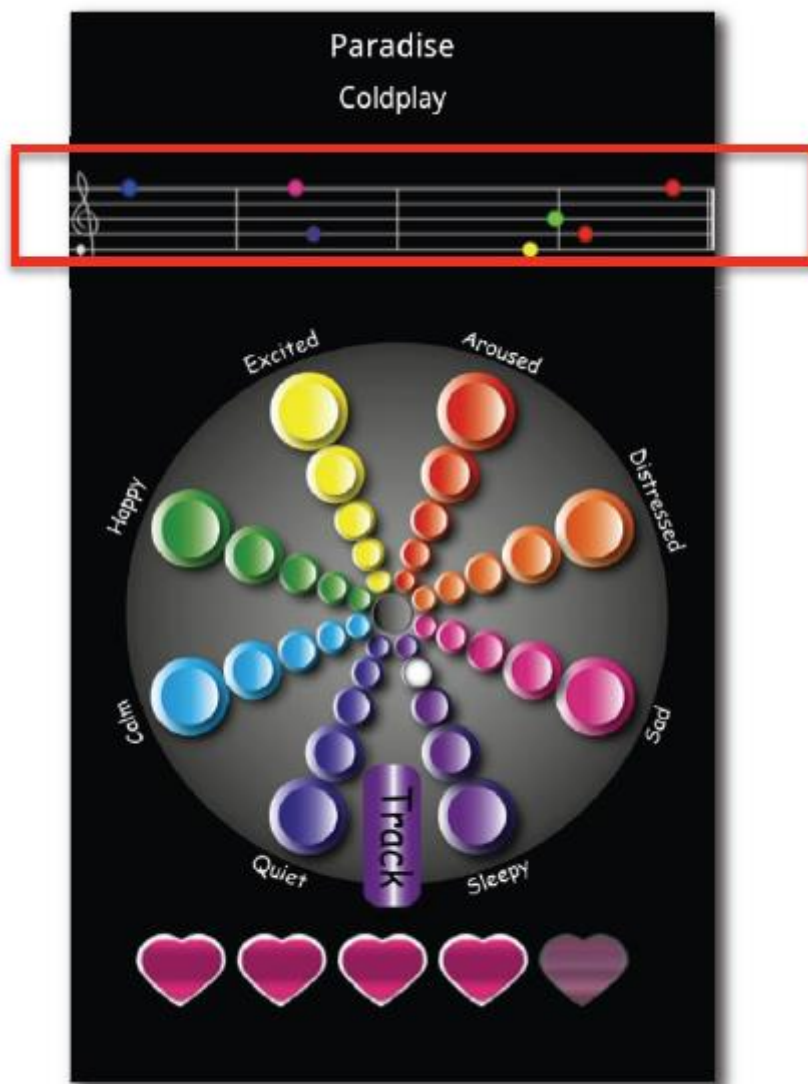


Figura 6. Lasciare commenti emotivi sulla timeline.

I risultati dell'esperimento si possono riassumere in tre concetti chiave. In primo luogo, l'individuo è più invogliato ad ascoltare la musica se è il sistema a suggerirgli le canzoni. Secondariamente, il poter giudicare un brano attraverso le emozioni è un modo alternativo di influenzare il sistema nella scelta del prossimo elemento da consigliare. Infine, la curiosità nel vedere quali sentimenti ha suscitato la stessa canzone in altre persone è stata la chiave del successo di GroupFun.

3. Mobile Recommendations

La proliferazione dei dispositivi mobile ha permesso l'offerta di servizi personalizzati disponibili sempre e ovunque; grazie alla localizzazione, i contenuti non dipendono solo dalla storia dell'interazione tra utente e sistema, ma anche dalla posizione in cui egli si trova.

Sviluppare sistemi di raccomandazione di tipo mobile introduce nuove sfide dovute alle proprietà del dispositivo stesso e al contesto del suo utilizzo [5]. Innanzitutto, rispetto ai tradizionali computer desktop, ha uno schermo più piccolo e questo può avere un impatto pesante sulle prestazioni. Il modo tradizionale di visualizzare le raccomandazioni è attraverso un elenco ordinato di elementi, dove ogni voce è tipicamente affiancata da una breve descrizione che spiega il perché è stata consigliata. Date le ridotte dimensioni del display, l'utente può essere forzato a effettuare ampio scorrimento nella lista di risultati, e più un elemento si trova verso il fondo della pagina maggiore è la probabilità che non venga considerato. Secondariamente, la capacità limitata di memorizzazione ed elaborazione porta la persona a sentirsi frustrata nell'attesa di un risultato da parte del sistema di raccomandazione. E infine, ma non per questo meno importante, bisogna considerare l'evolvere continuo dell'ambiente, dovuto allo spostamento fisico dell'individuo.

I sistemi di raccomandazione mobile possono essere rappresentati lungo tre dimensioni [6]. La prima, detta **user mobility**, si riferisce al fatto che l'utente può accedere alle informazioni da differenti località. Per esempio, un viaggiatore atterrato all'aeroporto di Vienna può utilizzare la rete wireless di un chiosco per accedere a TripAdvisor e prenotare una stanza d'albergo. Il sistema deve essere progettato in modo da sfruttare e supportare la mobilità dell'utente, mantenendo la storia dell'interazione con l'individuo.

La seconda, chiamata **device portability**, sottolinea il fatto che il device usato per accedere al sistema è mobile e come tale si sposta insieme all'utente. È la dimensione che è stata maggiormente studiata poiché impatta fortemente sulle caratteristiche

fisiche dei dispositivi (moderate dimensioni dello schermo, minor potenza computazionale e data storage limitato).

Infine con il termine **wireless connectivity** si prende in considerazione il fatto che il device si connette al sistema di raccomandazione attraverso una tecnologia wireless come WiFi, UMTS, 3G o Bluetooth.

3.1 Classificazione

Una classificazione concreta di questi sistemi è essenziale per capire le loro caratteristiche e constatare i rispettivi vantaggi e limitazioni. I parametri presi in considerazione per operare tale classificazione sono: la scelta architetturale, il grado di coinvolgimento dell'utente e i criteri osservati nella raccomandazione [7].

Per quanto riguarda l'architettura, troviamo:

- **Sistemi Web-based**

Si tratta di tipici sistemi client-server, in cui un'applicazione mobile (il client) si occupa del livello di presentazione mentre la logica di raccomandazione viene mantenuta sul server. Il client può essere fornito su browser o implementato da applicazioni Android/iOS.

- **Sistemi Stand alone**

Si riferiscono ad applicazioni mobile che incorporano la logica di raccomandazione e i contenuti da visualizzare. Vengono installate su Smartphone/Tablet e funzionano in modalità offline, con un evidente risparmio dei costi (il 3G a pagamento).

- **Sistemi Web-to-mobile**

Forniscono un'interfaccia Web in cui gli utenti selezionano il contenuto di loro interesse e costruiscono un'applicazione turistica personalizzata. Analogamente ai sistemi stand alone, l'applicazione viene installata su Smartphone/Tablet funzionando in modalità offline. Per aggiornare le

informazioni sui punti di interesse si possono utilizzare, ad esempio, connessioni ad un server remoto.

In merito al grado di coinvolgimento dell'utente al momento della presentazione degli item raccomandati, abbiamo:

- **Sistemi Pull-based**

In questi sistemi la raccomandazione è guidata da query, formulate sulla base delle richieste degli utenti. Dal momento in cui sono gli individui ad avere il controllo su ciò che viene loro presentato, i sistemi pull-based sono considerati meno invasivi. Questo perché è pensiero comune considerare intrusiva qualsiasi informazione non esplicitamente richiesta!

- **Sistemi Reactive**

Si tratta di sistemi particolari che reagiscono al cambiamento del contesto, generando una raccomandazione senza l'intervento esplicito dell'utente.

- **Sistemi Proactive**

Si riferiscono a sistemi in grado di raccomandare contenuti attraverso l'estrapolazione del contesto futuro utilizzando, ad esempio, modelli di previsione specializzati. Non fanno uso di dati correnti o storici, a differenza degli altri due tipi di sistemi presentati.

In riferimento ai criteri osservati nella raccomandazione, ricordiamo:

- **UCRS (User Constraints-based Recommender Systems)**

I sistemi UCRS lavorano su vincoli espressi dagli utenti e su informazioni dedotte implicitamente. I dati espliciti vengono ricavati attraverso un breve sondaggio presentato al primo avvio dell'applicazione, che richiede l'inserimento di informazioni demografiche e interessi della persona. La parte implicita viene derivata sulla base dell'interazione con il sistema.

Volendo mostrare un esempio di sistema UCRS, si illustra *GUIDE*: un progetto di guida turistica distribuito nella città di Lancaster. L'infrastruttura sottostante è costituita da una serie di punti di accesso wireless, la cui utilità è individuare il turista e le informazioni presenti nei punti di interesse. La raccomandazione viene fatta tenendo conto delle caratteristiche degli item (tempi di apertura e chiusura, periodo migliore per visitarli) e delle informazioni presenti nel profilo utente (dove si trova, qual è la sua velocità di camminata, quali luoghi ha già visitato). Il sistema offre le indicazioni stradali per andare da un punto di interesse a quello successivo e, se necessario, riordina l'itinerario, ad esempio nel caso in cui il turista decidesse di rimanere più tempo del previsto in un luogo.

- **LARS (Location-Aware Recommender Systems)**

I sistemi LARS rappresentano un caso speciale (o meglio una versione precoce) dei sistemi CARS, in quanto la logica di raccomandazione si basa esclusivamente su un parametro: la posizione del turista.

Nel seguito verranno brevemente presentati alcuni esempi di sistemi LARS. *GeoWhiz* utilizza la posizione per consigliare ristoranti che siano convenientemente situati nelle vicinanze dell'utente, a meno che non vi sia un criterio prioritario che giustifichi una diversa raccomandazione (presenza di intolleranze alimentari, offerte speciali, disabilità della persona, ecc.).

PECITAS offre raccomandazioni per percorsi point-to-point personalizzati nella città di Bolzano. I percorsi sono illustrati elencando le varie connessioni che l'utente deve compiere per raggiungere la destinazione con i mezzi pubblici e a piedi. Oltre a presentare il percorso più breve, il sistema può suggerire un itinerario più lungo perché, ad esempio, passa attraverso importanti attrazioni della città, a patto però che rispetti i vincoli imposti dall'utente (tempo di percorrenza, numero massimo di trasferimenti, ecc.).

- **CARS (Context-Aware Recommender Systems)**

I sistemi CARS si fondano sul concetto di context-aware, perfetto per la natura stessa dei dispositivi mobile. In particolar modo nel settore del turismo, la mobilità offre diverse dimensioni contestuali che possono essere alimentate implicitamente (cambiamenti di posizione) o dedotte (impiegare più tempo rispetto alla media nel visitare un luogo è sinonimo di interesse). Le raccomandazioni offerte da sistemi CARS possono quindi migliorare in modo significativo la soddisfazione dei turisti [9].

Entrando nello specifico, i valori di contesto possono essere catturati in diversi modi: attraverso i sensori incorporati nei device (unità GPS, bussola, videocamera), utilizzando i servizi forniti dal Web (previsioni meteo, informazioni sul trasporto pubblico), appoggiandosi ad infrastrutture di supporto (sensori distribuiti in un'area geografica per informazioni sulle temperature) o scambiando informazioni con altri utenti attraverso connessioni WPAN¹ ad hoc.

Esempi di parametri di contesto potenzialmente utili sono: la posizione, la distanza dal punto di interesse, il budget, il tempo a disposizione, la stagione, i mezzi di trasporto, le condizioni meteorologiche, la storia dell'utente, l'ambiente sociale, e così via.

Nel seguito verranno brevemente presentati alcuni esempi di sistemi CARS.

I'm feeling Loco è un sistema che considera le preferenze dell'utente, estratte dal social network Foursquare², e i vincoli spazio-temporali per generare la raccomandazione. I limiti fisici sono delimitati dalla posizione dell'individuo e dalla modalità di trasporto (a piedi, in bicicletta o in auto), automaticamente rilevati sulla base di misurazioni effettuate sfruttando l'accelerometro.

¹ Una PAN (Personal Area Network) è una rete informatica utilizzata per permettere la comunicazione tra diversi dispositivi (telefono, Tablet, ecc.) vicini ad un singolo utente. La WPAN (Wireless Personal Area Network) è una PAN che utilizza come mezzo trasmissivo le onde radio invece degli usuali cavi. La rete Bluetooth forma una WPAN.

² Foursquare è una rete sociale basata sulla geolocalizzazione disponibile tramite Web e applicazioni per dispositivi mobile. Gli utenti eseguono il check-in (la registrazione della propria posizione) tramite la versione browser del sito o attraverso applicazioni su device che utilizzano il GPS.

ReRex offre un'assistenza completa nella preparazione di un itinerario, e permette la modifica del tour in base alle circostanze e agli imprevisti che si verificano nel corso dell'escursione. Inoltre chiede all'utente di specificare in che modo un fattore contestuale influenza la valutazione di un elemento, in modo tale da generare una raccomandazione più mirata.

- **CBRS (Critique-Based Recommender Systems)**

I sistemi CBRS trovano la loro importanza nella critica: una forma di feedback che restringe lo spazio di ricerca e aiuta l'utente a trovare il prodotto desiderato in modo più efficiente. Una critica è una funzione di preferenza espressa tipicamente su una scala da 1 a 5, dove 1 indica elemento scadente e 5 indica elemento eccezionale. Ad esempio, una persona che riceve una raccomandazione per una vacanza può specificare che è alla ricerca di un soggiorno più economico, criticando la funzione di prezzo.

Fare elicitation attraverso le critiche potrebbe rivelarsi vantaggioso e particolarmente adatto allo scenario mobile. In primo luogo le preferenze sono espresse dall'utente e quindi più affidabili rispetto a quelle raccolte dall'implicit feedback. Secondariamente, lo sforzo richiesto è piuttosto basso se paragonato ad altri metodi utilizzati per fare indagini.

3.2 Mobile RS nel turismo

I sistemi di raccomandazione di tipo mobile godono di una forte applicazione nel settore del turismo [8]. L'utente impone i vincoli e il sistema restituisce i risultati in accordo con quanto richiesto; ad esempio, gli hotel possono essere selezionati per ora di check-in/check-out, prezzo, distanza dal centro città, disponibilità delle camere, possibilità di ospitare animali, e così via.

Oltre al navigatore satellitare ampiamente utilizzato per fornire indicazioni stradali, questi sistemi hanno il compito di assistere l'utente durante l'itinerario (Figura 7). Nello specifico, consentono di derivare un tour personalizzato sulla base dei vincoli imposti dal turista (tempo a disposizione, interessi di viaggio, ecc.) e, in alcuni casi,

permettono la sostituzione dei percorsi problematici con altri più appropriati (ad esempio, attraverso zone pedonali, parchi e foreste rispetto alle strade ad alto traffico).

Per mostrare i risultati si potrebbero utilizzare elenchi puntati, purché non siano troppo lunghi e carichi di informazioni; è buona norma suddividere gli elementi in più pagine, affinché ognuno di essi abbia la stessa probabilità di essere cliccato. In alternativa si possono introdurre interfacce basate sulle mappe, le quali rappresentano un valido strumento per mostrare dove sono situati i punti di interesse raccomandati dal sistema. L'unica accortezza è non eccedere nel numero di informazioni presentate, in caso contrario si creerebbe confusione invogliando l'utente a non utilizzare più l'applicazione.

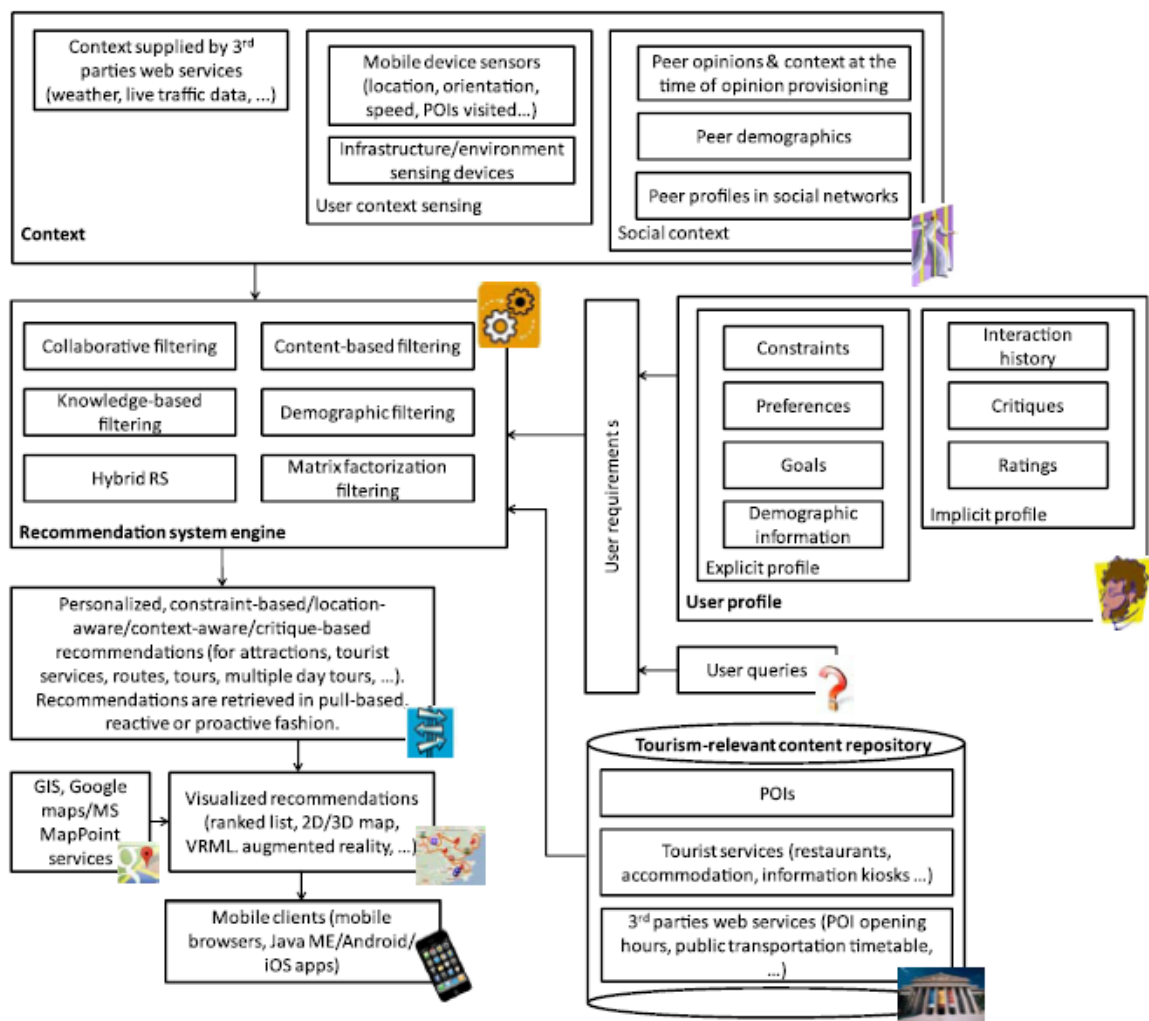


Figura 7. Generica architettura di un mobile RS per il turismo.

Per mostrare il potere della raccomandazione in ambito turistico, Modsching e i suoi collaboratori hanno condotto un esperimento su un campione di 421 turisti in visita a Gorlitz (Germania), che hanno opportunamente suddiviso in tre gruppi [10].

Al primo gruppo (Planner) è stato fornito un sistema di raccomandazione proattive per fornire percorsi personalizzati; al secondo (Explorer) hanno consegnato un sistema di raccomandazione pull-based per ricevere informazioni su richiesta; al terzo (Logger) è stato dato un tradizionale navigatore GPS.

Risultato ottenuto: i Planner e gli Explorer hanno espresso un alto gradimento per i suggerimenti ricevuti, paragonando l'esperienza a un tour guidato. Inoltre, confrontando i due gruppi dotati di RS con quello dotato di navigatore GPS, si è rilevato che i primi hanno impiegato meno della metà del tempo rispetto al secondo per vedere il quadruplo delle attrazioni (Tabella 3).

Criteria	Planner	Explorer	Logger
<i>Distance</i>	3.4 km	4.4 km	6.6 km
<i>Duration</i>	1.3 h	1.7 h	4 h
<i>Visited sights</i>	14	16	4
<i>Visit time (mode)</i>	4 min	4 min	2 min

Tabella 3. Risultato dell'esperimento di Modsching.

3.3 Esempi di mobile RS

Per concludere la trattazione, si mostrano alcuni esempi interessanti di sistemi di raccomandazione mobile.

MapMobyRek è un'applicazione utilizzata per suggerire ristoranti, pizzerie, spaghetterie nella città di Trento; ognuno di questi posti è descritto da un insieme di caratteristiche, tra cui nome, tipologia, costo, e altro ancora.

Si basa su un approccio di tipo constraint-based con rilassamento dei vincoli per eliminazione/aggiornamento delle preferenze. Questo perché si parte dal presupposto che la persona non sia sempre in grado di esprimere completamente i propri gusti mediante un'unica richiesta iniziale.

L'interazione tra utente e sistema, di tipo critique-based, fa sì che l'espressione della preferenza sia esplicita (Figura 8).

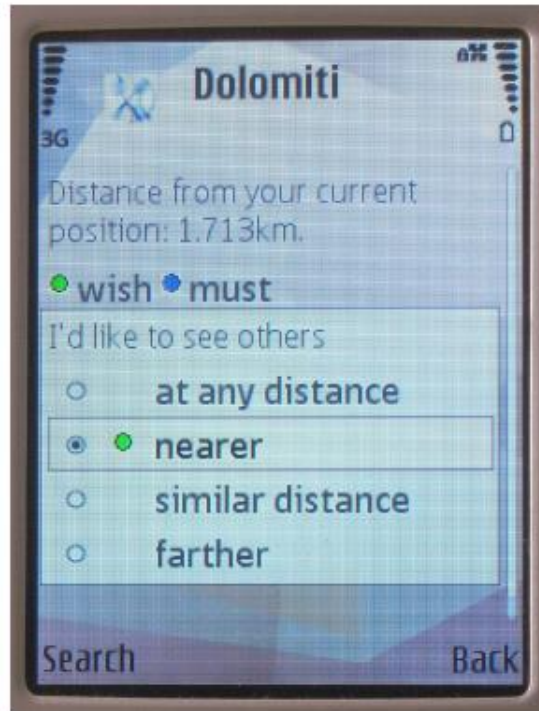


Figura 8. Esempio di critica in MapMobyRek.

Gli item suggeriti vengono mostrati su una mappa, dove è possibile distinguere raccomandazioni buone da raccomandazioni deboli attraverso l'uso di icone colorate. L'effetto di una critica si manifesta come un progressivo cambiamento del colore e/o della forma di un elemento; nella figura riportata, le icone con dimensioni più piccole stanno scomparendo dalla mappa a seguito di una critica mossa dall'utente (Figura 9).

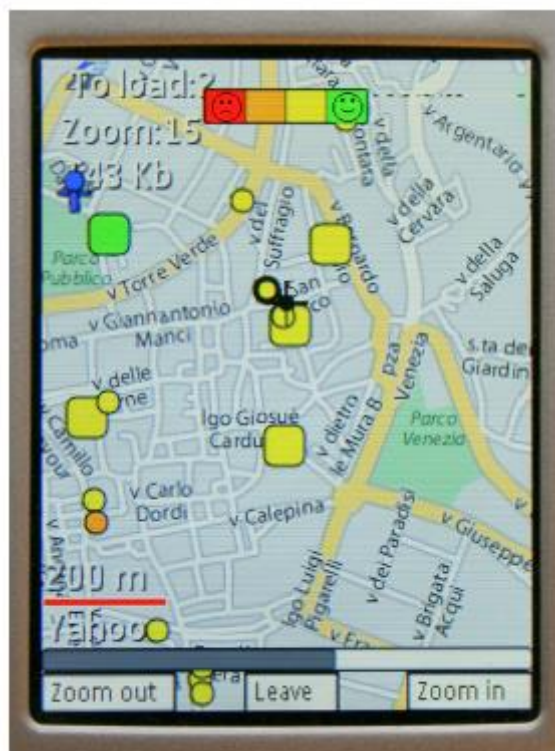


Figura 9. Mappa in MapMobyRek.

TRS è un'applicazione Web basata su MyMytiline, un framework turistico Web-to-mobile che permette la costruzione di una guida personalizzata. Nel prototipo originario, l'utente sceglie gli elementi dopo aver navigato all'interno di tutti i contenuti turistici disponibili; l'item scelto viene incluso nella guida mobile e adattato allo Smartphone in base alle sua dimensione e risoluzione.

La versione mobile di TRS (mTRS) integra numerose estensioni che affrontano i vari aspetti della tourism personalization. In primo luogo, permette agli utenti di contribuire in modo collaborativo caricando e condividendo informazioni turistiche, come giudizi, commenti e contenuti multimediali relativi al punto di interesse. Tali contenuti possono essere votati e commentati in modo da rendere la raccomandazione ancora più efficiente. Attraverso l'impiego di tecniche di filtraggio collaborativo, si è in grado di fornire suggerimenti personalizzati sulla base delle valutazioni di persone aventi gusti simili all'utente.

Secondariamente, mTRS fa sì che le raccomandazioni non dipendano solamente dalla storia dell'interazione tra utente e sistema, ma anche dal contesto (ad esempio, le condizioni meteorologiche locali o i luoghi già visitati).

Infine, supporta la comunicazione diretta e l'interazione sociale tra i turisti che condividono interessi simili e che si trovano in luoghi vicini (Figura 10).

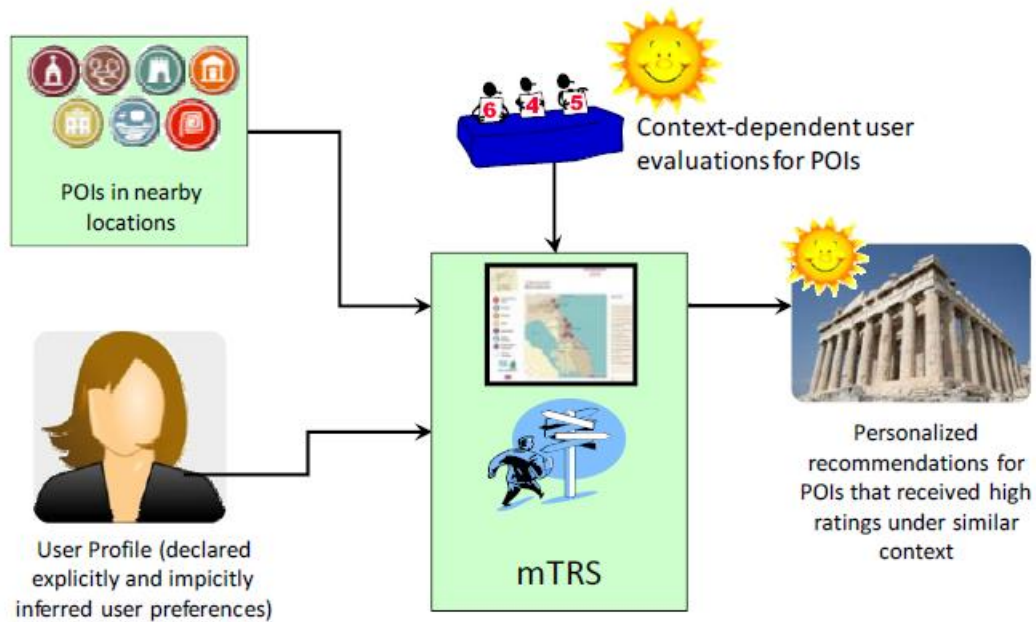


Figura 10. Architettura di mTRS.

mTRS sostiene la context-aware rating, grazie alla quale si attribuisce maggiore credibilità a utenti che caricano recensioni, valutazioni e commenti mentre si trovano sul posto (tramite i loro dispositivi mobile) rispetto ad altri che eseguono azioni simili attraverso interfacce Web remote. Per fare ciò il sistema assegna pesi diversi ai contenuti forniti dai turisti a seconda dell'infrastruttura tecnologica utilizzata.

In questa analisi bisogna tenere presente che le persone spesso faticano a interagire con un portale turistico a distanza durante il viaggio, a causa della mancanza di infrastrutture di rete, o addirittura evitano le connessioni dati dal dispositivo, a causa degli elevati costi di roaming. Per risolvere questo inconveniente mTRS

permette l'uso di reti WSN³ per fornire agli utenti un mezzo comodo e poco costoso con il quale caricare le informazioni e valutare gli item. I sensori sono distribuiti vicino ai punti di interesse e, quando il turista si trova in prossimità di uno di essi, viene chiesto di connettersi a un server di contenuti remoti (portale). A questo punto si esegue un handshake via Bluetooth con il nodo sensore al fine di stabilire la connessione e, una volta sincronizzati, l'applicazione riceve l'id del punto di interesse e si procede con la valutazione (Figura 11).

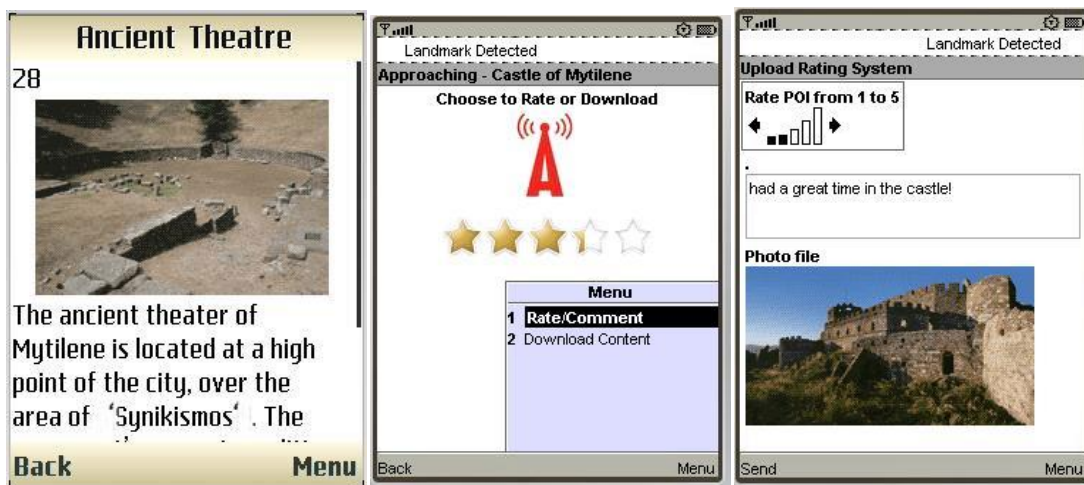


Figura 11. Screenshot significativi di mTRS.

Le principali problematiche riscontrate sono dovute a carenze di tecnologia Bluetooth, disconnessioni di rete causate dalla mobilità dell'utente ed elevato tempo di caricamento per foto di grandi dimensioni. In generale gli studi condotti dimostrano che i turisti sono rimasti soddisfatti dell'esperienza, e vorrebbero poter utilizzare questa applicazione anche in futuro, per rendere i loro viaggi più interattivi e ottenere raccomandazioni personalizzate su cosa visitare.

In Figura 12 si vede un utente accanto a un nodo sensore posizionato in questo caso vicino all'ingresso del Castello di Mytilene; nell'angolo in alto a sinistra si vede il

³ Con il termine WSN (Wireless Sensor Network) si indica una determinata tipologia di rete che, caratterizzata da un'architettura distribuita, è realizzata da un insieme di dispositivi elettronici autonomi in grado di prelevare dati dall'ambiente circostante e di comunicare tra loro.

nodo sink a cui destinare i dati raccolti dai nodi sensori. Il nodo sink trasmetterà successivamente i dati raccolti al Web Server mTRS.



Figura 12. Caricamento del contenuto.

ReRex è un'applicazione sviluppata per iOS basata su un sistema di raccomandazione mobile di tipo context-aware [9]. L'applicazione interagisce con il server mediante un protocollo basato su XML; il client effettua una richiesta di raccomandazione che precisa le condizioni contestuali e il server risponde con un elenco di punti di interesse (comprese le immagini e le descrizioni).

Viene data la possibilità al turista di modificare in qualsiasi momento i valori contestuali per ricevere un elenco aggiornato di item; in questo modo, si esplora il sistema e se ne valuta verosimilmente la qualità delle raccomandazioni.

Nella fase iniziale dell'interazione con *ReRex*, l'utente imposta il contesto della visita abilitando o disabilitando alcuni fattori (Figura 13sx). Le condizioni di contesto quali l'ora, il giorno della settimana, la distanza dal punto di interesse, il tempo, la stagione e la temperatura si ottengono automaticamente da altri componenti server; le restanti, se attivate, devono essere inserite manualmente

dall'utente. Ad esempio, se "Temperatura" e "Tempo" sono impostati a ON, il sistema terrà conto dei loro valori correnti interrogando un servizio di previsioni meteo esterno; se "Companion" è impostato a ON, viene chiesto all'utente di immettere la composizione del gruppo (Figura 13dx).

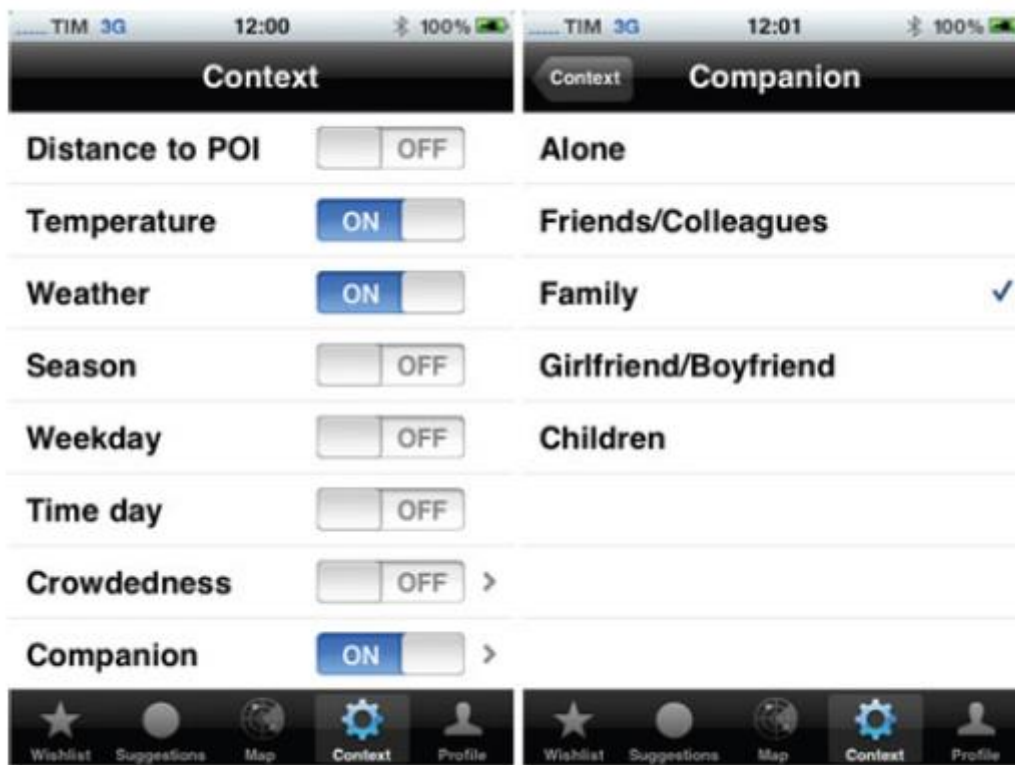


Figura 13. Specifica dei fattori contestuali.

A questo punto vengono suggeriti gli item che maggiormente si adattano alle preferenze impostate dalla persona (Figura 14). Nel caso in cui l'utente non si ritenesse soddisfatto di ciò che gli è stato presentato, può ottenere ulteriori risultati utilizzando il bottone "More".

Nelle figure mostrate, si può notare la presenza di un'icona che raffigura un piccolo orologio e una freccia verde. Come si vede dalla pagina di dettaglio dell'item, tale simbolo indica che il punto di interesse risulta particolarmente adatto da visitare con la famiglia, in accordo con le preferenze espresse all'inizio.

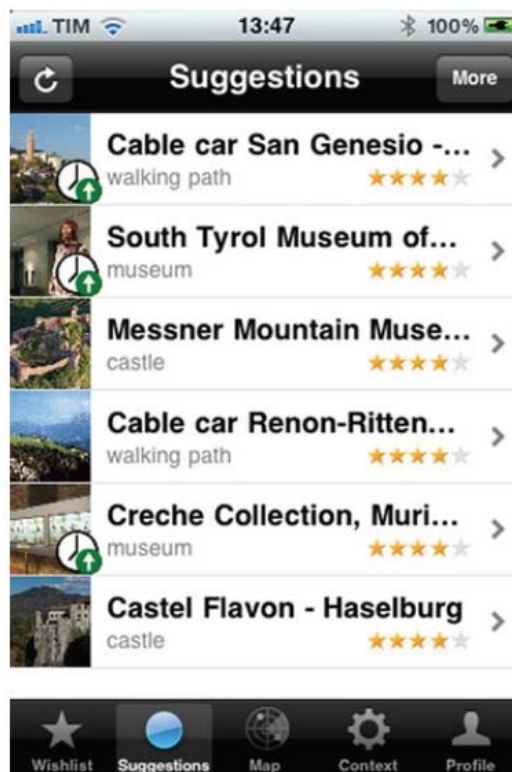


Figura 14. Lista di item suggeriti.

L'utente può selezionare un articolo dalla lista dei suggerimenti e andare alla sua pagina di dettaglio (Figura 15). Nella parte alta di questa schermata si trovano tre bottoni: "Details" per ottenere informazioni più specifiche sull'elemento scelto, "Map" che mostra la mappa Google con evidenziata la posizione del punto di interesse e quella attuale dell'utente, e "Feedback" che consente di rilasciare un commento.

Ogni item raccomandato può essere salvato nella lista dei desideri, mediante l'apposito bottone avente come icona una stella e un "+" verde. L'interfaccia della lista dei desideri è apparentemente simile a quella dei suggerimenti, e l'utente può in ogni momento eliminare un elemento da tale lista.



Figura 15. Dettaglio item raccomandato.

Un'importante funzionalità di ReRex è legata al suo comportamento attivo. Se una condizione contestuale sta cambiando, ad esempio la temperatura si abbassa, l'utente viene messo al corrente del cambiamento (Figura 16).

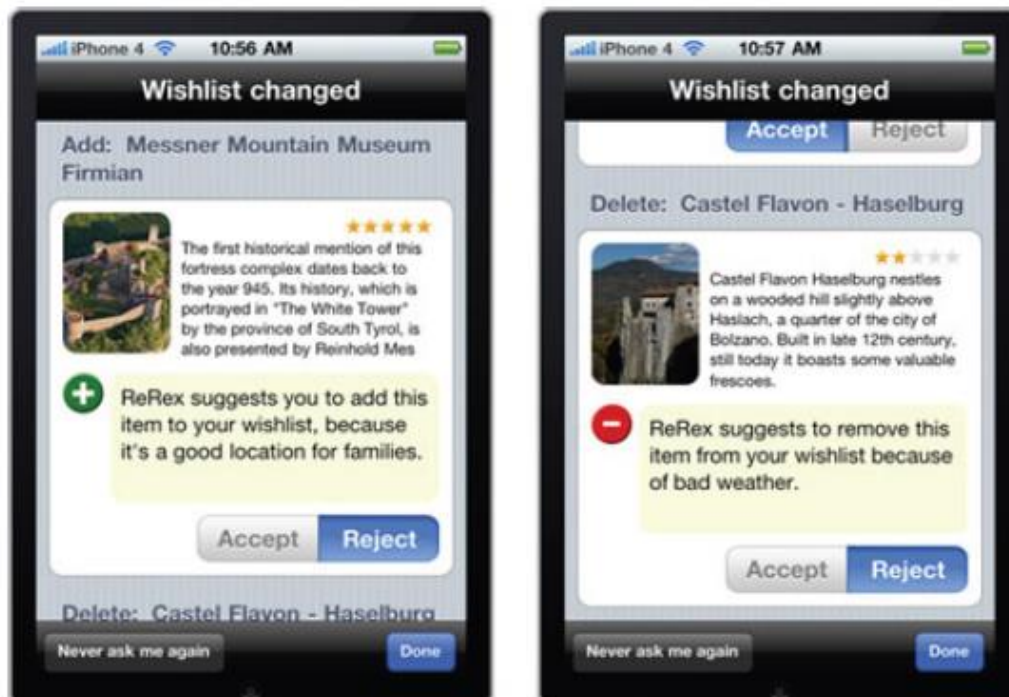


Figura 16. Notifica cambiamento condizione contestuale.

4. Elicitation

L'obiettivo di un sistema di raccomandazione è aiutare gli individui a trovare oggetti di loro interesse. La domanda sorge spontanea: se un nuovo utente si affaccia al sistema, come può l'algoritmo dare suggerimenti se non sa nulla della persona che ha di fronte?

Raccogliere e interpretare i gusti personali dell'individuo è un'operazione piuttosto delicata e prende il nome di elicitation. Affinché i sistemi di raccomandazione risultino efficaci è necessario rafforzare tale processo per collezionare quante più informazioni possibili, in grado di misurare l'interesse dell'utente verso un elemento o un gruppo di elementi.

Le tecniche utilizzate si classificano in explicit elicitation e implicit elicitation; negli ultimi anni sta prendendo piede una nuova strada frutto della combinazione tra le due [11].

4.1 Explicit elicitation

Come illustrato da Claypool in uno dei suoi studi, la soluzione più comune per raccogliere le preferenze dell'utente è affidarsi a valutazioni di tipo esplicito; in questo modo viene fornito un meccanismo per esprimere in modo inequivocabile i propri interessi.

Le informazioni raccolte sono significative ma costringono l'utente a impegnarsi in attività aggiuntive che vanno oltre il suo normale comportamento, e questo a volte può rilevarsi un ostacolo non indifferente.

Generalmente i dati vengono collezionati per mezzo di questionari, sondaggi, o indagini, ma negli ultimi anni si stanno affermando anche altre tecniche di explicit feedback. A titolo di esempio, si vuole citare il rating attraverso le stelle (usato in Amazon o Movielens), l'uso del like (presente nei social network come Facebook), e infine la nuova funzionalità aggiunta da Google (Figura 17).

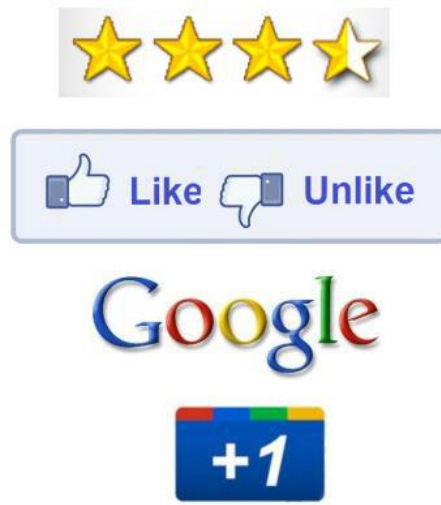


Figura 17. Tecniche comuni di explicit elicitation.

4.2 Implicit elicitation

Le tecniche di implicit feedback consentono di valutare un item senza l'intervento diretto dell'utente, ma semplicemente osservando la sua naturale interazione con il sistema. In altre parole, la valutazione viene eseguita senza che la persona ne sia a conoscenza, attraverso l'acquisizione di informazioni estrapolate dalle azioni e dai comportamenti adottati durante la navigazione. Ad esempio, quando si visualizza una notizia o un articolo sul Web, a seconda del tempo impiegato nella lettura, il sistema capisce se il lettore gradisce o meno il contenuto che sta leggendo.

Per analizzare l'impatto dell'implicit feedback, Claypool e i suoi collaboratori hanno sviluppato un browser, chiamato *The Curious Browser*, in grado di catturare le azioni degli utenti durante la navigazione sul Web [12]. Dall'analisi è emerso che gli indicatori principalmente osservati sono stati il tempo speso a visualizzare una pagina e il numero di scrolling effettuati; la combinazione tra questi due fattori ha evidenziato un esplicito interesse da parte dell'utente verso quel contenuto.

Il primo fattore si calcola come la differenza tra l'istante di apertura e l'istante di chiusura della pagina Web. Nel calcolo però vengono erroneamente considerate le pause durante le quali, ad esempio, l'utente riduce a icona la finestra e successivamente riprende la lettura.

Per compensare l'imprecisione del primo indicatore sono state introdotte altre misure, quali il movimento del mouse e lo scroll della pagina, in quanto si presuppone che se l'individuo scrolla verso il basso significa che sta leggendo ed è interessato al contenuto presentato.

Concludendo la rassegna sull'implicit elicitation, si riporta un importante studio condotto da Joachims e i suoi colleghi che hanno voluto analizzare come le persone interagiscono con l'elenco dei risultati presentati dal motore di ricerca Google [13]. L'importanza di questo studio non è legata solo a rilevare e valutare le azioni degli utenti, ma soprattutto nel processo decisionale che porta all'azione. A tal proposito si è iniziato a parlare di *Eye Tracking*: una tecnica che consente di registrare i movimenti degli occhi e fornire un resoconto del comportamento subconscio degli utenti. Nel contesto del Web, le fissazioni oculari sono la metrica più pertinente per valutare l'elaborazione delle informazioni nella ricerca online, e sono definite come uno sguardo spazialmente stabile della durata di circa 200-300 millisecondi durante la quale l'attenzione visiva è diretta ad una specifica area del display. La dilatazione della pupilla, invece, è una misura in genere utilizzata per indicare eccitazione o interesse verso il contenuto visualizzato: maggiore è il diametro della pupilla, maggiore sarà l'interesse verso l'item.

I problemi principali legati all'implicit feedback riguardano la probabilità di incorrere in raccomandazioni sbagliate a causa di una elicitation non del tutto corretta, e al fatto che gli utenti si possano sentire in qualche modo osservati nei loro movimenti e violati nella propria privacy.

Inoltre non vi è alcun segnale per esprimere un feedback negativo. Mentre con le valutazioni di tipo esplicito l'utente può stabilire se un item è di suo gradimento o meno, qui non si può assumere zero feedback come segnale che alla persona non piaccia quell'elemento. Come se non bastasse, la mancanza di metriche chiare fa sì che le tecniche di tipo implicito risultino imprecise. Ad esempio, se si stanno confrontando diverse serie tv, si può presupporre che l'utente non ami una serie se l'ha guardata una sola volta. Lo stesso ragionamento non lo si può però applicare ai

film cinematografici, poiché la maggior parte delle persone guardano lo stesso film al cinema una sola volta pur trovandolo piacevole e interessante.

4.3 Pro – Contro delle diverse tecniche di elicitation

Gli studi condotti dimostrano che gli utenti tendono a preferire un'elicitation di tipo esplicito, in quanto sentono di avere maggior controllo potendo esprimere direttamente le loro preferenze. D'altra parte però concordano nell'affermare che le tecniche di implicit feedback risultano meno fastidiose e invadenti. Per tale ragione spesso si ricorre a metodi combinati, in grado di costruire una rappresentazione più accurata degli interessi degli individui.

Moling e i suoi collaboratori hanno condotto un test per utilizzare congiuntamente le tecniche di explicit e implicit feedback [14]. Il loro esperimento si basa su RLradio, un lettore musicale che offre musica in streaming da diversi canali, ognuno dei quali presenta un genere diverso. La tecnica di explicit feedback è presente nella fase iniziale dove si chiede all'utente di specificare la percentuale di musica di quel canale che desidera ascoltare (Figura 18).

Genre	Hours
Blues	25
Jazz Fusion	40
Pop Top Hits	0
Metal	0
Relax	20
Classic	0
Rock	0
Opera	0
Baroque	0

Missing 15 hours.

Save

Figura 18. Feedback esplicito.

La tecnica di implicit feedback, invece, è dedotta sulla base dell'interazione tra utente e sistema. L'individuo può ascoltare tutti i brani di uno stesso genere, simbolo di un forte interesse per quel tipo di musica, oppure richiedere una nuova traccia con l'azione sul bottone "Next" perché presumibilmente non gradisce ciò che è in fase di riproduzione (Figura 19).

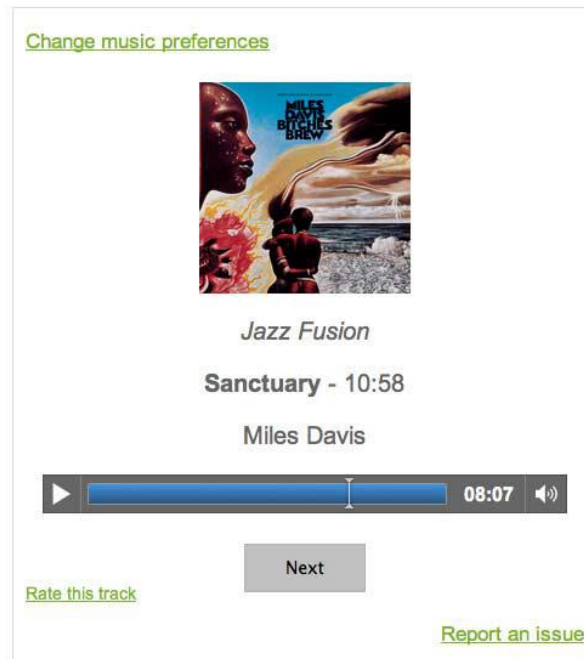


Figura 19. Feedback implicito.

5. Explanation

L'explanation gioca un ruolo fondamentale nel migliorare l'esperienza utente, esponendo il ragionamento che sta dietro una certa raccomandazione. Potrebbe aiutare le persone a trovare ciò che vogliono, convincerle a provare o acquistare un prodotto e chiarire il motivo per cui un articolo è stato consigliato. Tutto questo aumenta il grado di soddisfazione dell'utente e la sua fiducia verso il sistema.

Visto il suo largo campo di azione, è importante concordare ciò che l'explanation sta cercando di raggiungere. Un articolo interessante pubblicato sul Wall Street Journal dal titolo "Cosa fare quando TiVo pensa che tu sia gay" descrive la frustrazione di un utente davanti a scelte irrilevanti fatte da TiVo: un videoregistratore incaricato di

memorizzare programmi che potrebbero interessare al proprietario sulla base di quello che egli ha registrato in passato [15]. Il caso emblematico è dato dal signor Iwanyk, il quale sospettava che il suo TiVo lo ritenesse gay dal momento in cui inspiegabilmente ha continuato a registrare programmi con tematiche sull'omosessualità. Questo utente chiaramente meritava una spiegazione a tutto ciò! Capita l'importanza dell'explanation, si vuole sperimentare sotto quali aspetti risulti più efficace: uso della grafica vs testo, quantità di informazioni da mostrare, ecc. La modalità più semplice di mostrare una spiegazione è quella di creare un "why" che racchiuda le motivazioni alla base della raccomandazione (Figura 20).

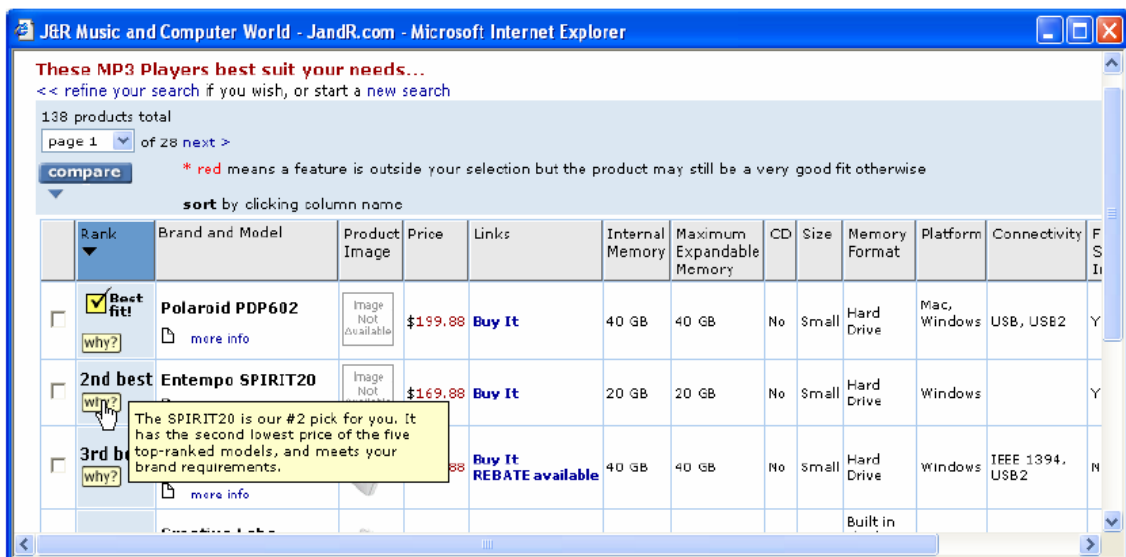


Figura 20. Explanation mediante "why".

Come tecnica alternativa e potenzialmente più efficace, si può utilizzare un'interfaccia dove vengono raggruppate in categorie le raccomandazioni che forniscono una soluzione di compromesso (Figura 21).

Search Results							
There are three apartments satisfying your preferences on price, bathroom and distance							
ID	Type	Price (Fs)	Area (m2)	Bathroom	Kitchen	Distance (mins)	
27	room in a house	500	15	private	private	5	Basket
30	room in a house	500	22	private	not available	10	Basket
71	room in a house	490	18	private	not available	10	Basket
Although these apartments are slightly expensive, they offer superior benefits on some of the other attributes							
ID	Type	Price (Fs)	Area (m2)	Bathroom	Kitchen	Distance (mins)	
77	shared apartment	550	20	private	not available	10	Basket
34	room in a house	550	25	shared	private	10	Basket
							More
These apartments satisfy your price need, but not on all other preferences							
ID	Type	Price (Fs)	Area (m2)	Bathroom	Kitchen	Distance (mins)	
69	shared apartment	470	15	shared	shared	10	Basket
72	room in a house	500	12	shared	private	15	Basket
							More

Figura 21. Explanation nel titolo di ogni categoria.

Un'ulteriore modalità sta nell'utilizzare contemporaneamente sia una parte testuale che una parte grafica. Il vantaggio derivante dall'adozione di questa strategia risiede nel fatto che l'informazione visiva può permettere alle persone di sviluppare una più profonda comprensione dei dati. Nell'esempio mostrato, si tenta di esporre la differenza tra due abitazioni con l'aiuto di un grafico (Figura 22).



Figura 22. Explanation mediante testo e grafico.

Alla ricerca della migliore tecnica per presentare l'exploration, Pearl Pu e i suoi collaboratori hanno dimostrato che la visualizzazione di un insieme diversificato di

risultati (Figura 24), risulta più efficace rispetto ad un'interfaccia che presenta il semplice link "Why?" (Figura 23) [16]. Questo perché la maggior parte dei partecipanti ha ritenuto più semplice prendere una decisione potendo confrontare tra loro i diversi prodotti consigliati. A tal proposito, hanno pensato potesse essere utile apportare una piccola modifica all'interfaccia inserendo un trade-off vector, la cui funzionalità sarebbe quella di indicare se l'attributo della raccomandazione è migliorato (↑) o peggiorato (↓) rispetto al candidato superiore. Riprendendo l'esempio della tecnologia, il trade-off vector {(prezzo, ↑); (velocità del processore, ↓); (memoria, ↓); (dimensioni del disco rigido, ↑); (dimensioni del display, ↑); (peso, ↓)} vuole esprimere "il prodotto in questione, rispetto al notebook superiore consigliato, ha un prezzo inferiore, presenta un hardisk di dimensioni maggiori e un display più grande, ma di contro è più pesante, la velocità del processore è più bassa e la sua memoria è limitata".

The most popular product								
	Manufacturer	Price	Processor speed	Battery life	Installed memory	Hard drive capacity	Display size	Weight
⊕	-	\$2'095.00	1.67 GHz	4.5 hours	512 MB	80 GB	38.6 cm	2.54 kg

We also recommend the following products								
	Manufacturer	Price	Processor speed	Battery life	Installed memory	Hard drive capacity	Display size	Weight
○	Why?	\$1'220.49	1.8 GHz	5 hours	1 GB	100 GB	38.1 cm	2.95 kg
○	Why?	\$2'148.99	2.0 GHz	4 hours	1 GB	100 GB	39.1 cm	2.90 kg
○	Why?	\$1'379.00	3.3 GHz	2 hours	512 MB	100 GB	43.2 cm	4.31 kg
○	Why?	\$1'179.00	3.2 GHz	2 hours	512 MB	80 GB	39.1 cm	3.62 kg
○	Why?	\$1'529.00	1.7 GHz	6.5 hours	512 MB	80 GB	33.8 cm	1.77 kg
○	Why?	\$1'599.00	1.7 GHz	6.5 hours	512 MB	80 GB	33.8 cm	1.91 kg
○	Why?	\$1'425.00	1.6 GHz	5.5 hours	512 MB	80 GB	39.1 cm	2.86 kg
○	Why?	\$2'235.00	1.8 GHz	2.5 hours	1 GB	100 GB	43.2 cm	3.99 kg
○	Why?	\$1'190.00	3.2 GHz	1 hours	512 MB	80 GB	39.1 cm	3.72 kg

Figura 23. Explanation mediante "why".

The most popular product								
Manufacturer	Price	Processor speed	Battery life	Installed memory	Hard drive capacity	Display size	Weight	
☉	-	\$2'095.00	1.67 GHz	4.5 hour(s)	512 MB	80 GB	38.6 cm	2.54 kg

We also recommend the following products because they are cheaper and lighter, but have lower processor speed								
Manufacturer	Price	Processor speed	Battery life	Installed memory	Hard drive capacity	Display size	Weight	
○	-	\$1'499.00	1.5 GHz	5 hour(s)	512 MB	80 GB	33.8 cm	1.91 kg
○	-	\$1'739.99	1.5 GHz	4.5 hour(s)	512 MB	80 GB	36.6 cm	2.49 kg
○	-	\$1'625.00	1.5 GHz	5 hour(s)	512 MB	80 GB	30.7 cm	2.09 kg
○	-	\$1'426.00	1.5 GHz	5 hour(s)	512 MB	60 GB	30.7 cm	2.09 kg
○	-	\$1'929.00	1.2 GHz	4 hour(s)	512 MB	60 GB	26.9 cm	1.41 kg
○	-	\$1'595.00	1 GHz	5.5 hour(s)	512 MB	40 GB	26.9 cm	1.41 kg

they have higher processor speed and bigger hard drive capacity, but are heavier								
Manufacturer	Price	Processor speed	Battery life	Installed memory	Hard drive capacity	Display size	Weight	
○	-	\$1'220.49	1.8 GHz	5 hour(s)	1 GB	100 GB	38.1 cm	2.95 kg
○	-	\$2'148.00	2 GHz	4 hour(s)	1 GB	100 GB	39.1 cm	2.9 kg
○	-	\$1'379.00	3.3 GHz	2 hour(s)	512 MB	100 GB	43.2 cm	4.31 kg
○	-	\$2'235.00	1.8 GHz	2.5 hour(s)	1 GB	100 GB	43.2 cm	3.99 kg
○	-	\$2'319.00	1.7 GHz	4.5 hour(s)	512 MB	100 GB	43.2 cm	3.13 kg
○	-	\$2'075.00	1.8 GHz	1.57 hour(s)	512 MB	100 GB	43.2 cm	4.4 kg

they are lighter and have longer battery life, but smaller display size								
Manufacturer	Price	Processor speed	Battery life	Installed memory	Hard drive capacity	Display size	Weight	
○	-	\$1'529.00	1.7 GHz	6.5 hour(s)	512 MB	80 GB	33.8 cm	1.77 kg
○	-	\$1'599.00	1.7 GHz	6.5 hour(s)	512 MB	80 GB	33.8 cm	1.91 kg

Figura 24. Explanation nel titolo di ogni categoria.

Alcuni modi di presentare la raccomandazione possono essere considerati come forme di explanation. Infatti, l'importanza di un elemento può essere messa in risalto ponendolo in cima ad una lista, evidenziandolo con un diverso colore e dimensione del carattere, oppure ancora facendo uso di rating, come i numeri o le stelle. Più in generale, il risultato di una raccomandazione può essere mostrato con una delle seguenti tecniche (Tabella 4):

- **Top item**

Il modo più semplice è offrire all'utente il migliore elemento che si adatta alle sue esigenze. Supponiamo che una persona sia interessata allo sport, e che apprezzi particolarmente il gioco del calcio. L'explanation risultante potrebbe essere: "Guardi molto sport, in particolare il calcio; questo è l'elemento più popolare e più recente della coppa del mondo". È da notare che, nell'esempio mostrato, ci si è basati sulla storia dell'utente, ma in altri casi il sistema potrebbe utilizzare le informazioni presenti nel suo profilo.

- **Top N-items**

Il sistema presenta più elementi contemporaneamente, offrendo all'utente la possibilità di scegliere tra una lista di oggetti che rispondono maggiormente alle sue esigenze. Questa modalità si inserisce molto bene in un dominio di grandi dimensioni, come quello delle news, dove è probabile che una persona coltivi più interessi. In quest'ottica è importante che il sistema sia in grado di spiegare la relazione intercorrente tra gli elementi presentati.

- **Similar to top item(s)**

Una volta che l'utente mostra la preferenza per uno o più elementi, il sistema di raccomandazione offre oggetti simili a quelli verso cui ha mostrato interesse. Supponiamo che ad una persona sia piaciuto il libro "Great Expectations" di Charles Dickens; il sistema, in accordo con quanto detto, proporrà un altro libro dello stesso autore. L'explication risultante potrebbe essere: "Ti potrebbe piacere anche "Oliver Twist" di Charles Dickens".

Considerando le preferenze di altri individui aventi gusti simili a quelli dell'utente, l'explication cambierà in: "Le persone con i tuoi stessi interessi hanno trovato piacevole leggere "Oliver Twist" di Charles Dickens".

- **Predicted ratings for all items**

Con questa modalità la raccomandazione viene presentata come una valutazione prevista su una scala numerica che va, ad esempio, da 0 a 5. Supponiamo che una persona ami lo sport, e in particolare il calcio. Il sistema, in accordo con quanto detto, offrirà più frequentemente elementi inerenti al calcio piuttosto che all'hockey. L'utente può interrogare il sistema chiedendo come mai è stato previsto un rate così basso per i risultati di hockey, e l'explication risultante potrebbe essere: "Questo è un elemento di sport, ma si tratta di hockey. Tu non hai mai mostrato un grande interesse verso l'hockey!".

- **Structured overview**

Il sistema fornisce una struttura che consente di visualizzare il trade-off tra i vari elementi. Il vantaggio di una panoramica strutturata è che l'utente può confrontare i vari articoli, e vedere se ve ne sono altri disponibili nel caso in cui la raccomandazione attuale non abbia soddisfatto pienamente le sue esigenze.

SYSTEM	ITEM TYPE	PRESENTATION EXPLANATION
Amazon.com	Books, Movies, ...	Similar to top items(s)
Findory	News	Similar to top items(s)
LibraryThing.com	Books	Similar to top items(s)
Organizational Structure	Digital cameras, notebook	Structured overview
ACORN	Movies	Structured overview, Top N-items
LIBRA	Books	Top N-items, Predicted ratings
LoveFilm.com	Movies	Top N-items, Predicted ratings
MovieLens	Movies	Top N-items, Predicted ratings
News Dude	News	Top N-items
Adaptive place advisor	Restaurants	Top item
Pandora.com	Music	Top item
Qwikshop	Digital cameras	Top item, Similar to top items(s)
SASY	Holiday	Top item, Similar to top items(s)
Top Case	Holiday	Top item, Similar to top items(s)

Tabella 4. Esempi di sistemi suddivisi per modalità di explanation.

5.1 Esempi di explanation

Per comprendere meglio l'importanza dell'explanation, vediamo alcuni esempi. Cramer e i suoi collaboratori hanno condotto un esperimento per raccomandare opere d'arte ai visitatori del Rijksmuseum di Amsterdam, basandosi su un algoritmo di tipo content-based [17]. Per ogni articolo consigliato, il sistema propone

un link “Why?”, il quale rimanda ad una finestra pop-up contenente l’elenco delle proprietà in comune tra l’opera d’arte raccomandata e le altre opere d’arte che l’utente ha valutato positivamente in passato (Figura 25 e Figura 26).



Figura 25. Esempio di explanation con “Why”.



Figura 26. Finestra pop-up che mostra la spiegazione testuale.

Czarkowski ha invece testato l’efficacia dell’explanation con un’applicazione, in grado di raccomandare la vacanza ideale sulla base delle informazioni personali contenute nel profilo utente [18].

Nel primo caso il sistema segnala un alloggio idoneo ad una persona single e senza limitazioni economiche particolari (Figura 27).

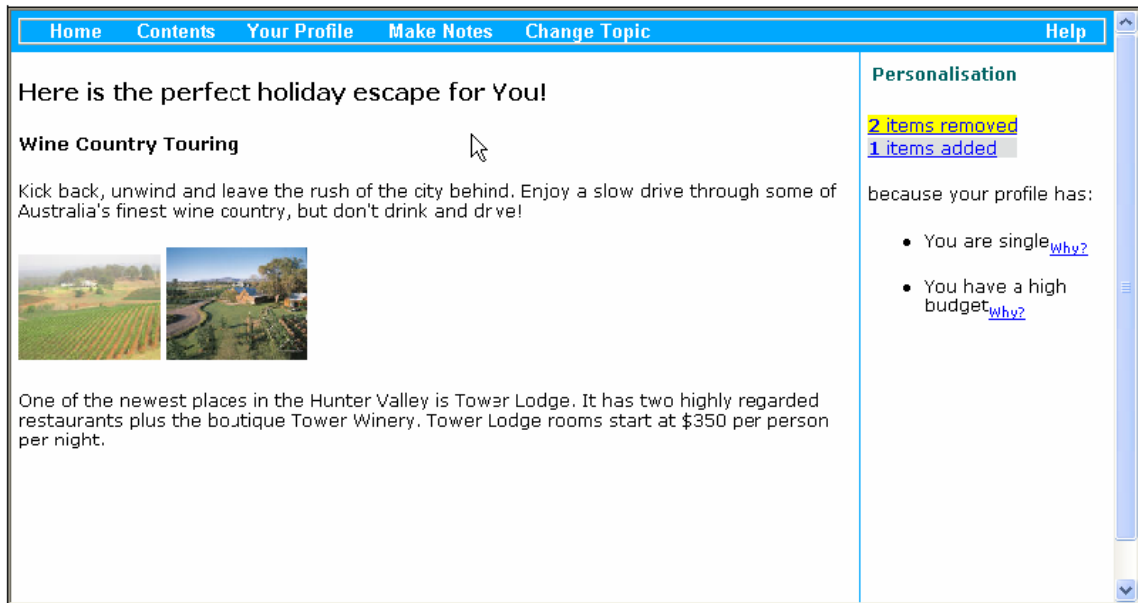


Figura 27. Raccomandazione per utente sigle con budget elevato.

Nel secondo caso il sistema propone una soluzione che si adatta particolarmente per un utente fidanzato/sposato – notare la scritta “This is a great opportunity to spend some romantic, quality time with your partner” – con a disposizione un budget ridotto (Figura 28).

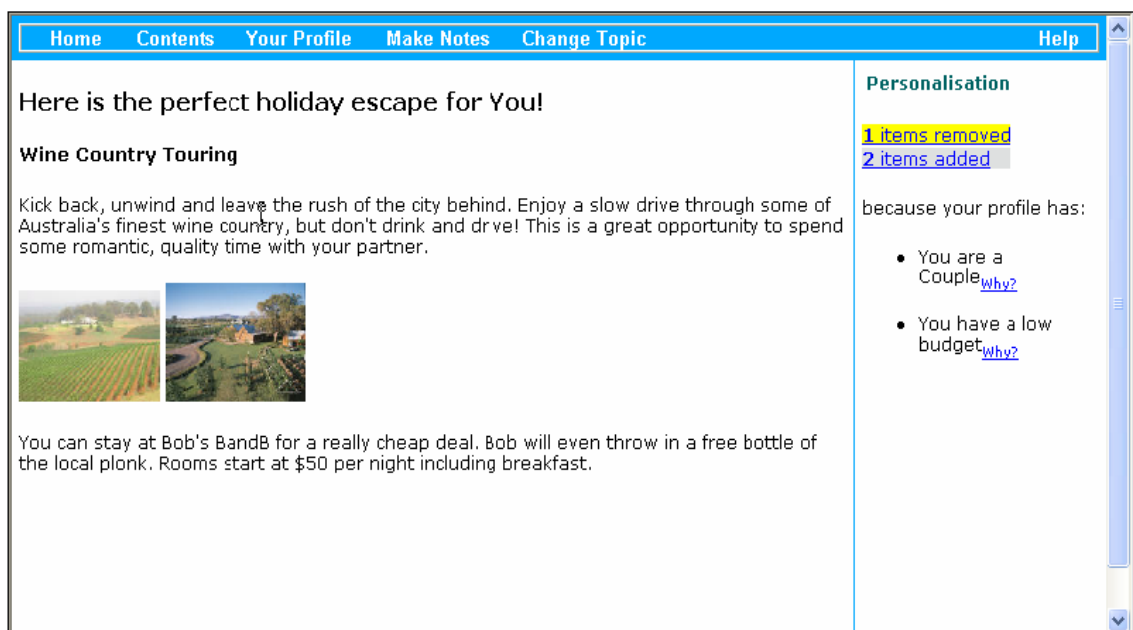


Figura 28. Raccomandazione per utente non sigle con budget basso.

In entrambe le figure il pannello laterale sulla destra contiene la “Personalization”, ovvero le modifiche apportate dal sistema a seguito di una variazione nei dati contenuti nel profilo utente. Gli elementi ritenuti non più adatti alla persona saranno rimossi dalla lista dei risultati (item con sfondo giallo), mentre altri verranno aggiunti perché appropriati al nuovo profilo utente (item con sfondo grigio). Nella transizione da *utente single e con budget elevato* a *utente fidanzato/ sposato con budget basso* (da Figura 27 a Figura 28), il sistema reagisce modificando la lista degli elementi raccomandati (Figura 29 e Figura 30).

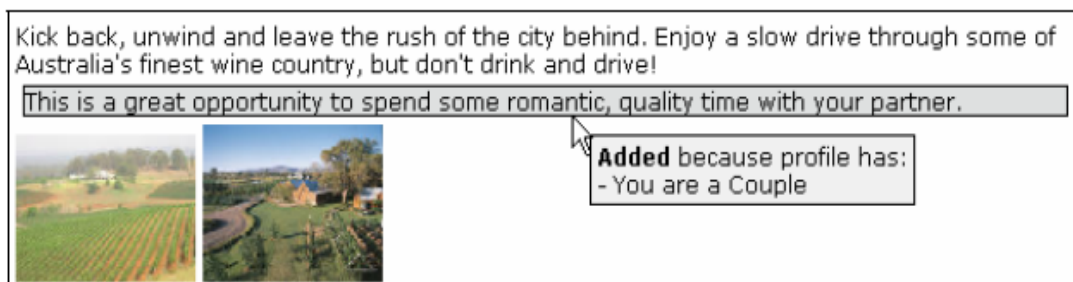


Figura 29. Dettaglio con spiegazione dell'elemento aggiunto.


Home	Contents	Your Profile	Make Notes	Change Topic	Help
<p>Instructions</p> <p>Move your mouse over a coloured section below to pop up the reason it was removed or added</p> <p>Key:</p> <ul style="list-style-type: none"> Removed by personalisation Added by personalisation No colour - content added for everyone 					<p>Personalisation Highlighted</p> <p>1 items removed</p> <p>2 items added</p> <p>Return to normal page view</p> <p>because your profile has:</p> <ul style="list-style-type: none"> You are a Couple why? You have a low budget why?
<p>Here is the perfect holiday escape for You!</p> <p>Wine Country Touring</p> <p>Kick back, unwind and leave the rush of the city behind. Enjoy a slow drive through some of Australia's finest wine country, but don't drink and drive!</p> <p>This is a great opportunity to spend some romantic, quality time with your partner.</p>  <p>One of the newest places in the Hunter Valley is Tower Lodge. It has two highly regarded restaurants plus the boutique Tower Winery. Tower Lodge rooms start at \$350 per couple per night.</p> <p>You can stay at Bob's BandB for a really cheap deal. Bob will even throw in a free bottle of the local plonk. Rooms start at \$50 per couple per night and include breakfast.</p> <p>Click here to return to normal page view</p>					

Figura 30. Nuova raccomandazione per modifica profilo utente.

6. Interazione tra utente e display pubblico

Negli ultimi anni sta emergendo un nuovo modo di condurre informazioni. Schermi interattivi e display pubblici vengono distribuiti in ambienti urbani, centri commerciali e vetrine di negozi. Aeroporti, stazioni ferroviarie e stadi stanno vivendo una trasformazione da tradizionale a digitale, permettendo la nascita di una nuova esperienza utente [19].

Allo stato attuale, la maggior parte degli schermi digitali mostrano video o grafici senza dare la possibilità agli utenti di poterli manipolare. Tuttavia, tali schermi presto cederanno il passo ai display interattivi, i cui meccanismi di interazione spaziano dal touchscreen all'uso di un dispositivo mobile come remote controller.

Le sfide da affrontare sono molteplici. Come presentare le informazioni in un ambiente affollato? Quali tecniche utilizzare per comunicare con le persone nel modo meno invasivo possibile e socialmente accettabile? Che tipo di input e interfaccia tecnologica è necessario sviluppare per consentire un'efficace interazione?

A differenza di molte altre tecnologie informatiche, i display pubblici non sono proprietà dei loro utenti primari. L'esperienza quindi non può essere paragonabile a quella che l'uomo ha con il personal computer o con il proprio Smartphone; in altre parole, l'interazione con i display pubblici non inizia con la volontà dell'utente di interagire! Per capire meglio il significato di quanto appena scritto, analizziamo le fasi di cui si compone (Figura 31).

Le persone inizialmente passano, senza mostrare alcuna volontà di interagire. Nella seconda fase, sono catturate dai contenuti presentati e lo si può notare dal fatto che, ad esempio, stanno sorridendo o girano la testa in quella direzione. Vi è poi la "Suble Interaction", dove gli utenti iniziano ad interagire con il display attraverso gesti o movimenti e si verifica, ad esempio, quando muovono una mano per vedere che effetto provoca sullo schermo. Successivamente si passa all'interazione diretta, nella quale la persona si impegna e tipicamente si posiziona al centro del display. Terminata la loro esperienza, l'utente va via lasciando il posto ad altri individui.

Per superare la prima soglia che separa “Passing by” da “Viewing and Reacting”, è necessario cogliere in qualche modo l’attenzione dei passanti. Per superare la seconda soglia che separa “Viewing and Reacting” da “Subtle Interaction”, bisogna indurre la curiosità nelle persone. Per superare tutte le altre soglie, invece, gli utenti devono essere motivati.

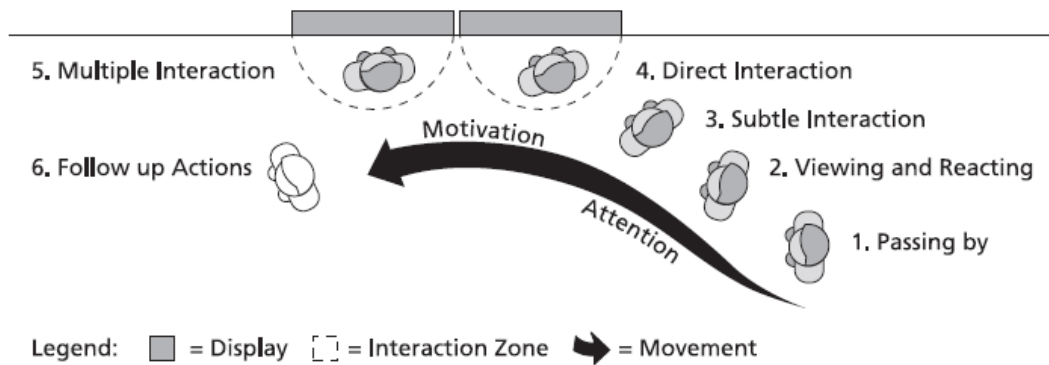


Figura 31. Fasi dell'interazione con display pubblico.

Kuikkaniemi e i suoi collaboratori hanno scritto un elaborato dove ciò che emerge è il cambiamento della vita urbana a seguito dell’introduzione di schermi interattivi [20]. Secondo quanto riportato, la progettazione di un display pubblico è paragonabile alla messa in scena di uno spettacolo teatrale, che richiede attenzione a una serie di dettagli: il layout dello schermo, l’illuminazione, il suono, e così via.

All’interno dell’elaborato si parla di un nuovo modo di concepire il display pubblico. Non più televisori rettangolari piani, ma qualsiasi superficie, indipendentemente dalla sua forma, sarà candidata a diventare fruitore di contenuti. Liberato dalle sue restrizioni, il contenuto dello schermo entrerà a far parte del mondo reale!

6.1 Principi di progettazione per display pubblici

Estetica. Molto spesso i display sono situati in una posizione tale da divenire parte integrante del loro ambiente. Pertanto si deve considerare attentamente l’estetica

della informazioni visualizzate, in modo da catturare l'attenzione dei passanti. Studi condotti dimostrano che un display troppo reattivo tende a distrarre l'utente, mentre uno eccessivamente lento lo annoia e lo allontana.

Durata. Il display dovrebbe esser progettato per sostenere attività di breve durata. Questo suggerisce che l'avvio e la conclusione di un'interazione dovrebbe essere piuttosto veloce, senza richiesta esplicita di un login/logout. Ad esempio il semplice allontanandosi dall'area dove risiede il display dovrebbe interrompere l'interazione.

Usabilità. Non deve esserci una formazione preventiva che permette l'utilizzo del display, anzi l'apprendimento deve avvenire in loco durante l'esplorazione dei contenuti. In questo modo, trattandosi di un luogo pubblico, le persone potranno imparare nuove funzionalità osservando altri utenti all'opera.

Multimedialità. Le informazioni comunicate devono essere comprensibili, e invogliare le persone a interagire per approfondire l'argomento. Il testo è sempre esaustivo, ma una foto vale più di mille parole e, in alcuni casi, il suono fa entrare la persona in una dimensione diversa che arricchisce la sua esperienza.

Multiutente. Non solo più utenti dovrebbero potersi collegare e interagire con il display, ma il display stesso dovrebbe essere in grado di comunicare con le persone in maniera socialmente accettabile in base al loro livello di attenzione e di apertura a ricevere maggiori informazioni. A tal proposito si suggerisce l'osservazione di alcuni parametri quali la velocità dell'utente, la direzione della sua camminata, lo sguardo, la vicinanza al grande schermo, e così via.

6.2 Esempio di interazione tra utente e display pubblico

Alla base dello studio di Coutrix e dei suoi collaboratori c'è la convinzione che raramente le persone si avvicinano allo schermo interattivo con un compito preciso in mente. Pertanto, il display deve essere in grado di invogliare e incuriosire i passanti per attirarli a sé e indurli a iniziare l'interazione.

Il progetto condotto si chiama FizzyVis e prevede l'installazione di uno schermo multitouch in occasione di un festival jazz [21]. I display multitouch permettono a più persone di interagire nello stesso tempo, avendo così anche un risvolto a livello

sociale. I passanti possono entrare e uscire spontaneamente, e iniziare a giocare in parallelo o contemporaneamente. Questo oltre a condividere l'esperienza di gioco, fornisce i presupposti per il learning collettivo dei contenuti.

L'elemento base di FizzyVis è la bolla, che obbedisce ad una simulazione di gravitazione sul grande schermo; l'effetto è introdotto con la finalità di stimolare e incuriosire i passanti. La gravitazione, tuttavia, viene disattivata quando la bolla viene ingrandita attraverso un semplice tocco. È possibile ridimensionare la bolla, rispettando le misure massime e minime imposte dal sistema, con un gesto a due dita ormai comune nelle interfacce multitouch. Il dettaglio delle informazioni varia a seconda della dimensione della bolla; pertanto l'azione di ingrandimento può essere interpretata come volontà da parte dell'utente di approfondire i contenuti.

Idealmente tutte le bolle sono visibili sullo schermo. È possibile assegnare priorità diverse attraverso la dimensione e il peso (caratteristica che impatta sulla dinamica della gravitazione), e questo può essere configurato all'inizio per scegliere su cosa concentrare l'attenzione del passante (Figura 32).

Le bolle inoltre interagiscono tra loro attraverso link magnetici, ovvero l'ingrandimento di una bolla comporta l'apertura automatica di un'altra bolla contenente informazioni correlate.

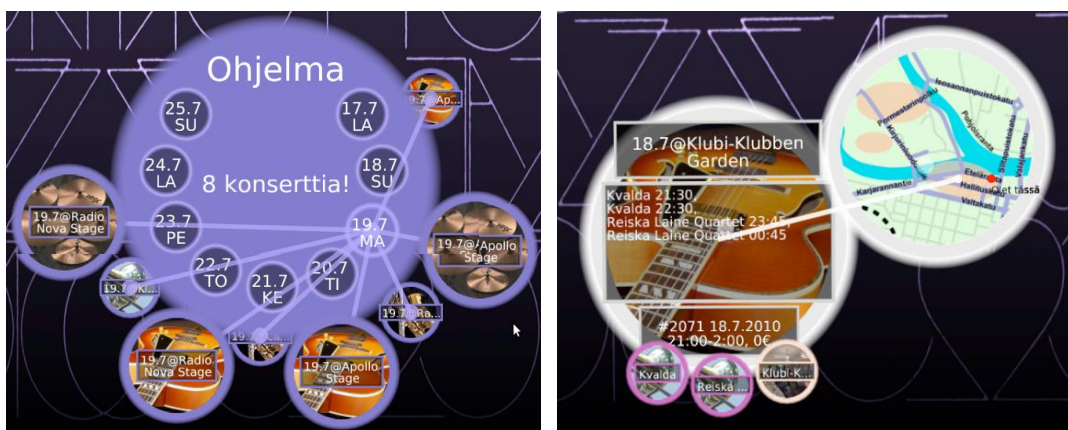


Figura 32. Bolle in FizzyVis.

I risultati del questionario evidenziano che le persone hanno trovato utile la presenza del display per reperire le informazioni, il quale è stato piacevole da utilizzare per l'aspetto gioco e divertente delle bolle (Figura 33).

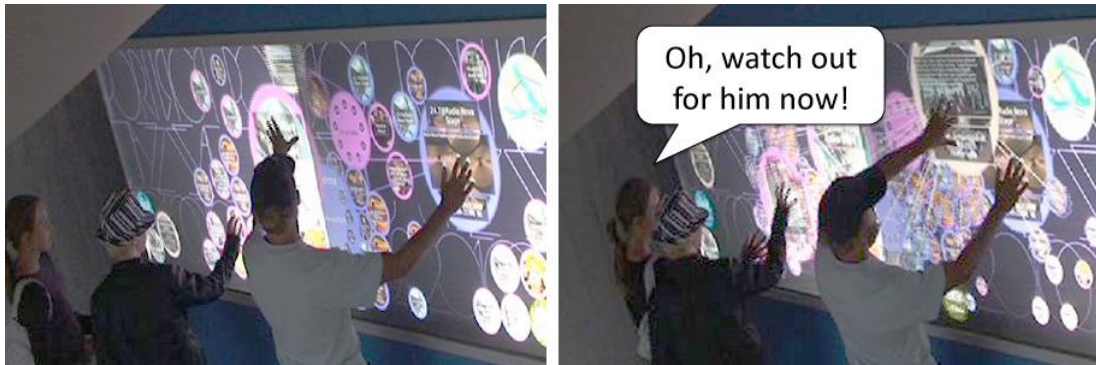


Figura 33. Interazione con le bolle.

7. Integrazione di piccoli e grandi schermi

Come accennato nel paragrafo precedente, negli ultimi anni sta emergendo un nuovo modo di condurre informazioni mediante l'impiego di display di grandi dimensioni collocati in luoghi pubblici, come aeroporti, stazioni ferroviarie, musei,... Nella maggior parte dei casi questi schermi hanno come unica funzionalità quella di trasmettere informazioni a tutti coloro che sostano nell'area predisposta. E anche se in certi casi viene previsto un supporto interattivo tramite touchscreen, i contenuti visualizzati sono indipendenti dagli interessi e dalle preferenze degli utenti. L'idea è quindi portare i display a diventare un'opportunità di scambio con le persone, al di là del semplice compito di visualizzare informazioni.

Studi precedenti dimostrano che le tecniche di interazione sfruttano sia segnali impliciti (posizione del corpo, orientamento della testa e vicinanza dell'utente al display) che azioni esplicite derivate dai movimenti delle mani e dal touchscreen. Non è difficile convincersi che tutto ciò concentra il potere della comunicazione verso un singolo individuo, e anche in caso di multi touchscreen il numero di informazioni che si possono visualizzare contemporaneamente rimane comunque modesto. Inoltre va sottolineato che l'uso del touchscreen richiede che gli utenti

siano “a portata di mano”, limitando l’interazione alle persone che si trovano nelle prime file di fronte al display e oscurando la visibilità a coloro che invece stanno dietro.

In questo contesto l’utilizzo di dispositivi mobile potrebbe potenzialmente consentire a intere folle di interagire simultaneamente con il grande schermo. E la familiarità verso il proprio device stimolerebbe l’individuo ad avvicinarsi con il grande schermo, superando quella resistenza iniziale. Ci sono però diversi aspetti da prendere in considerazione quando si affronta questo delicato argomento. Che tipo di dispositivi mobile utilizzare? Quale tecnologia di connessione usare? Quanti utenti dovrebbero essere in grado di connettersi simultaneamente al servizio?

Mentre i display condivisi in genere offrono una significativa potenza di calcolo, una maggiore larghezza di banda, e uno spazio di presentazione più grande, i dispositivi mobile godono di una considerevole diffusione e di una grande capacità interattiva. Il collegamento tra di essi è un modo ovvio per sfruttare il meglio di entrambi i mondi, ma la stretta sinergia avviene solamente quando gli utenti dei dispositivi mobile possono contare sulla capillarità dei display condivisi.

La maggior parte delle interazioni di questo tipo si basa sul Bluetooth: protocollo molto diffuso tra i telefoni cellulari ma debole a causa della forzata fase preliminare di accoppiamento. Infatti, prima che i device possano scambiarsi contenuti o condividere informazioni, deve avvenire una sorta di identificazione dove i sistemi si scambiano un codice pin. In un contesto in cui molte persone vogliono interagire con un display condiviso, questa soluzione diventa impraticabile. Si potrebbe superare la limitazione utilizzando il BlueCasting, un server in grado di inviare informazioni ad ogni dispositivo Bluetooth opportunamente configurato che rientra in un certo raggio. Tuttavia questo approccio è irritante per gli utenti in quanto ricevono contenuti (non scelti da loro!) senza aver avviato nemmeno la comunicazione.

In generale sarebbe bene valutare gli ostacoli legati ad una visualizzazione condivisa, ovvero la gestione di più input contemporanei e il grado di tolleranza per ritardi di comunicazione non prevedibili.

7.1 Esempi di integrazione

Un primo esempio di integrazione viene proposto da Maunder che, dopo aver lavorato su diversi prototipi, ha realizzato una tecnica chiamata “Snap & Grap” [22]. Questa tecnica permette all’utente di selezionare il contenuto da approfondire, facendo una foto alla porzione di schermo che contiene i media ritenuti interessanti. La fotografia viene inviata al display pubblico tramite Bluetooth, il quale la elabora e invia il media corrispondente al dispositivo dell’utente (Figura 34).

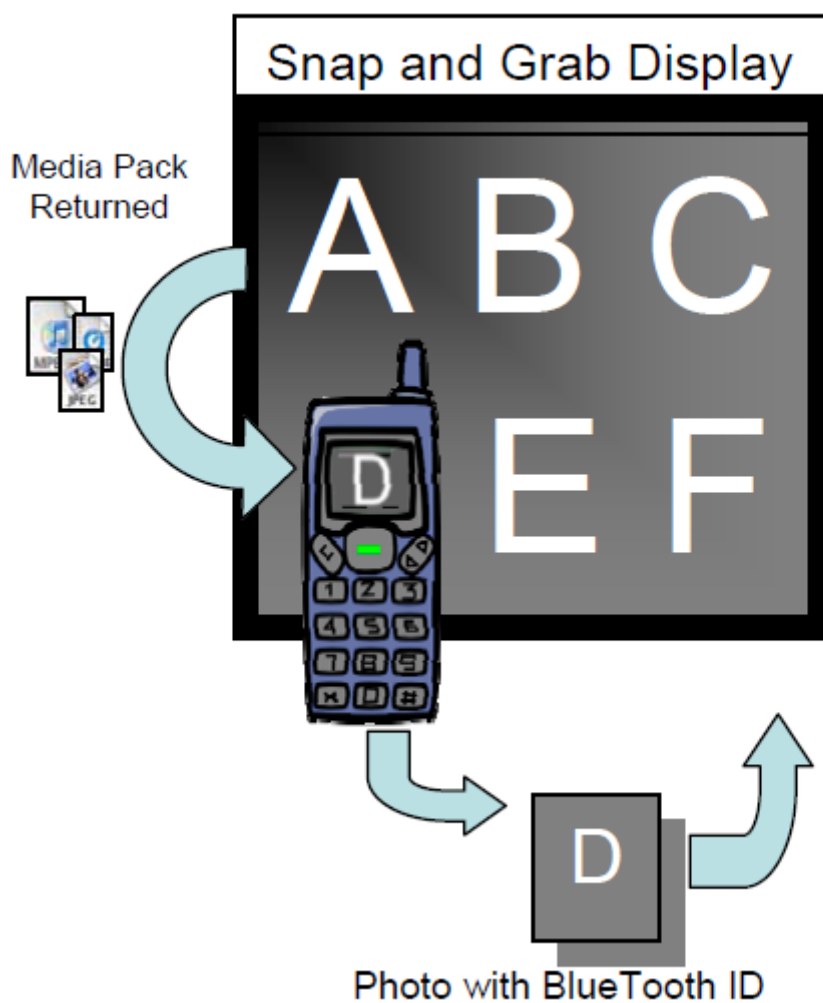


Figura 34. Meccanismo di Snap & Grap.

Sistemi basati su Bluetooth, come Snap & Grab, devono conoscere l’indirizzo del dispositivo client a cui inviare eventuali dati. Esso può essere incorporato all’interno

degli oggetti scambiati, oppure determinato chiedendo all'utente di trovare il proprio tra un set di indirizzi Bluetooth proposti. Snap & Grab utilizza il primo metodo e, pertanto, il server estrae l'indirizzo del dispositivo dall'immagine inviata dall'utente; non si richiede software aggiuntivo da installare sul client e questo rende l'interazione piuttosto semplice.

Un altro esempio interessante viene mostrato da Anzures, il quale nel suo elaborato ha sperimentato un'interazione di tipo "personal device based" realizzando la piattaforma chiamata PACMEN [23].

L'architettura presenta un display pubblico, situato in una posizione strategica affinché i giocatori possano vedere bene senza essere troppo vicini, e un server utilizzato per la comunicazione. I client non interagiscono direttamente con il grande schermo, ma comunicano con il server attraverso una rete wireless; questa soluzione permette una buona scalabilità e compatibilità a basso costo (Figura 35).

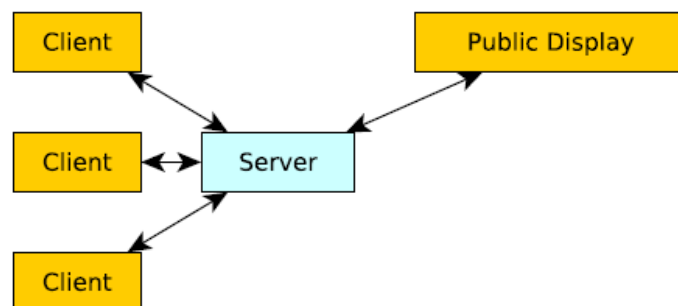


Figura 35. Architettura base di PACMEN.

Lato server, è indispensabile la presenza di un robusto sistema di comunicazione, che supporti una topologia dinamica dove i client sono liberi di entrare e uscire in ogni momento.

Lato client, l'unica applicazione installata sul dispositivo mobile si chiama SWFLoader ed è progettata per funzionare come un browser Web capace di connettersi ad un servizio, scaricare ed eseguire i contenuti offerti. Una volta lanciata, l'applicazione chiede all'utente di inserire il numero del server, che si presume essere visibile a tutti coloro che si trovano nelle vicinanze del display.

Dopo l'immissione, SWFLoader contatta il server e avvia il download della corrispondente applicazione in modo tale da poter iniziare l'interazione.

La piattaforma PACMEN è stata utilizzata per creare una versione multiplayer del famoso gioco Pacman⁴ (Figura 36). L'applicazione scaricata dagli utenti è un pad direzionale usato per controllare la creatura sul display pubblico; è possibile iniziare o abbandonare la partita in qualsiasi momento senza interrompere il gioco degli altri utenti. Pacmen prevede al massimo quattro giocatori contemporaneamente connessi al server; quando un giocatore viene catturato dal fantasma o lascia il gioco, cede il suo posto ad un altro in attesa.

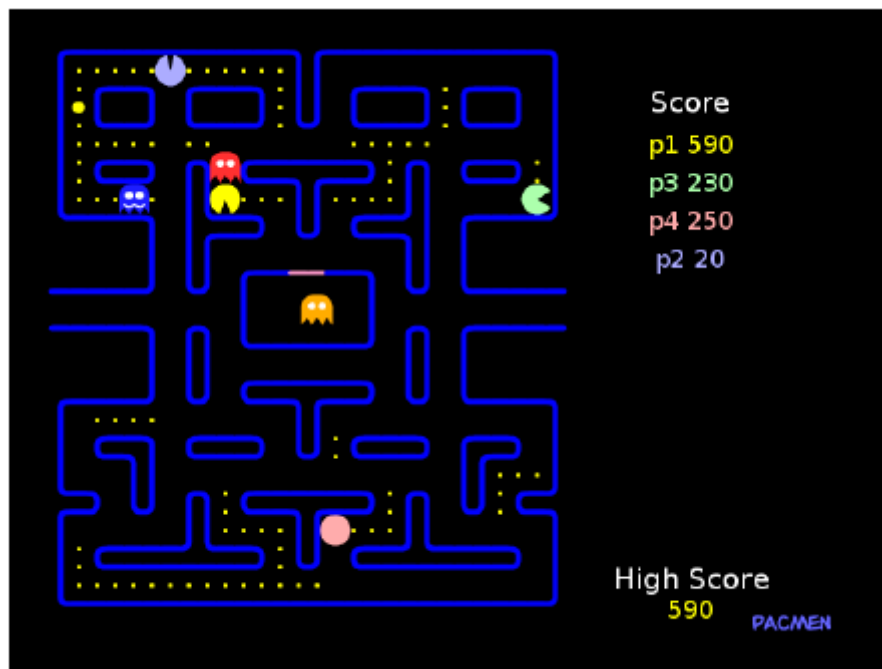


Figura 36. Pacmen game.

L'ultimo esempio interessante che si vuole riportare si chiama SWW (Sapporo World Window) ed è stato da realizzato da Choi e i suoi collaboratori [24]. SWW è un'applicazione screen-based sviluppata per il nuovo sottopassaggio di Sapporo,

⁴ Pacman è un famoso videogioco dove si deve guidare una creatura sferica di colore giallo lungo un labirinto facendole mangiare tutti i numerosi puntini disseminati ed evitando di farsi toccare dai fantasmi, pena la perdita immediata di una delle vite a disposizione.

capitale dell'isola di Hokkaido. Ogni anno circa 14 milioni di persone visitano quest'isola a nord del Giappone, nota soprattutto per il suo ambiente naturale, il cibo fresco e il lungo nevosso inverno, rendendo Sapporo un vivace centro urbano.

Per promuovere l'iniziativa di diventare una "Creative City", sono stati installati dieci grandi schermi all'interno del sottopassaggio. Ognuno di essi mostra video generati dagli utenti riguardanti vari aspetti della città, e una mappa che in tempo reale mostra la loro interazione con Sapporo (Figura 37).

L'applicazione mira a coinvolgere il pubblico funzionando come unico punto di collegamento per le interazioni socio-culturali e tecnologiche; in questo modo SWW permette alle persone di avere esperienze significative con altri utenti e con i luoghi di Sapporo attraverso il proprio telefono cellulare.

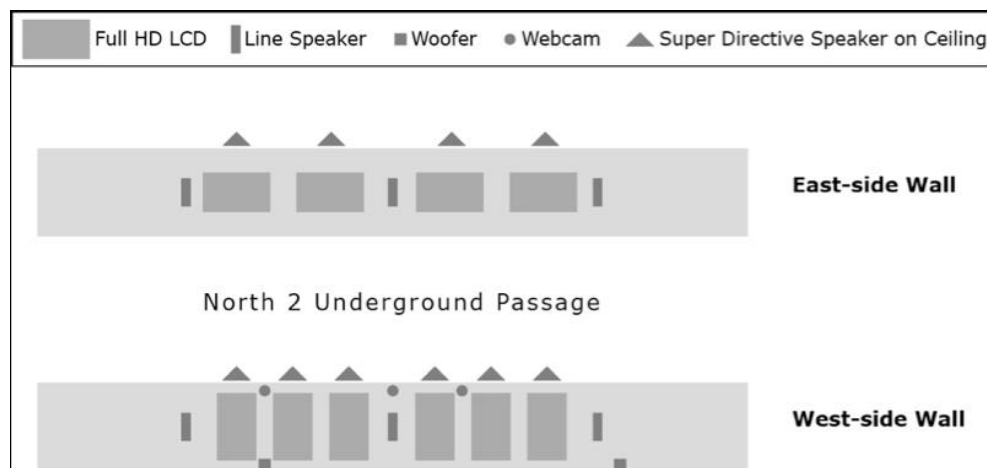


Figura 37. Piantina del sottopassaggio.

Obiettivo generale del progetto è creare un ambiente dove le persone possano scoprire l'arte, la cultura, le attrazioni turistiche e altre informazioni rilevanti di Sapporo, servendosi di dieci grandi schermi e di altoparlanti installati all'interno del sottopassaggio. I pedoni possono interagire con un display tramite il proprio dispositivo mobile per ottenere dettagli su dove si trova il punto di interesse, l'itinerario di viaggio, i consigli e i commenti di chi c'è stato, e così via (Figura 38). Inoltre può vedere cosa altri utenti stanno guardando sul cellulare e decidere di

comunicare con loro; questa funzione incoraggia la condivisione di informazioni e fa nascere un senso di “unione” nell’esplorare la città insieme.



Figura 38. Interazione tra display e utenti.

Ogni schermo presenta un QR code per permettere ai passanti di connettersi al contenuto specifico visualizzato (Figura 39). Nonostante la grande diffusione di cellulari dotati di fotocamera, viene fornito anche un sito Web per consentire la visione anche a coloro che posseggono cellulari non dotati di questa funzionalità. Inoltre la presenza del sito Web contribuisce a mantenere la privacy di individui piuttosto sensibili che non si sentono a proprio agio davanti ad un display pubblico durante la scansione del QR code.

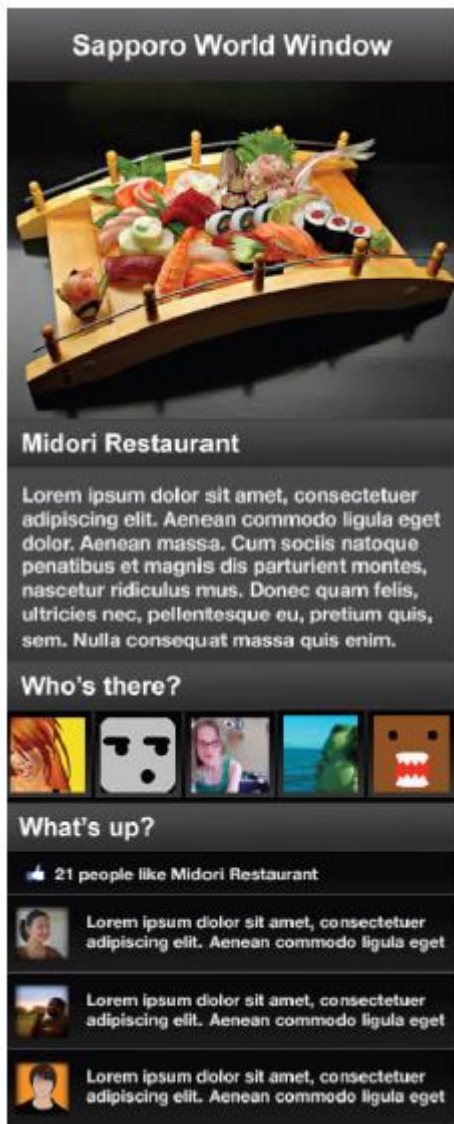
I contenuti social, prelevati interrogando diverse fonti, vengono visualizzati in fumetti animati posti in cima all’immagine o al filmato; nell’esempio mostrato 21 persone hanno messo un like su Facebook al ristorante, 4 utenti si trovano attualmente nel posto e 7 passanti hanno fatto lo screen del contenuto tramite QR code. Vengono inoltre visualizzati due commenti postati sui social network (Facebook o Twitter) e la distanza in minuti rispetto alla posizione del display pubblico.



Figura 39. Design display pubblico.

Nella schermata principale dell'applicazione mobile viene visualizzata l'immagine o il video così come è stato presentato sullo schermo, per dare idea all'utente di interagire con il display desiderato. In aggiunta, viene fornita una descrizione testuale circa l'oggetto selezionato (Figura 40a) e una tab bar consente la visualizzazione della mappa con le indicazioni per raggiungere la destinazione (Figura 40b).

Attraverso la visualizzazione di contenuti social su schermi pubblici o privati (Smartphone), si fornisce un indicatore dell'attività sociale del punto di interesse e un canale di comunicazione per scambiarsi informazioni su di esso. Non è difficile convincersi che la condivisione degli interessi può migliorare sensibilmente l'esperienza utente, e in SWW la si può ritrovare in una griglia a matrice contenente altri punti di interesse scelti dalle persone (Figura 40c).



a)



b)



c)



d)

Figura 40. Design applicazione mobile.

ANALISI

Nell'ingegneria del software, l'analisi è l'indagine preliminare che ha lo scopo di definire (il più precisamente possibile) il problema da risolvere. In questa fase ci si concentra sullo studio del dominio applicativo, utile a capire il contesto nel quale il prodotto software deve inserirsi. Sulla base di quanto emerso, vengono stilati i requisiti; nel lavoro di tesi si devono prendere in considerazione i requisiti legati all'applicazione mobile e al grande schermo.

1. Il contesto

Il contesto in cui si sviluppa il progetto riguarda l'Expo: l'esposizione universale di natura non commerciale che avrà luogo a Milano tra il 1° maggio e il 31 ottobre del 2015. L'evento vedrà la partecipazione e il coinvolgimento di milioni di persone che, oltre a visitare il sito espositivo, coglieranno l'occasione per conoscere la città. In quest'ottica è prevista la dislocazione di imponenti schermi, la cui funzionalità principale sarà mostrare informazioni e contenuti multimediali sui principali luoghi di interesse di Milano.

L'idea alla base del progetto è trovare un modo per riconoscere il tipo di visitatore che si trova di fronte al display, affinché quest'ultimo possa mostrare un elenco di punti di interesse personalizzato in base alle caratteristiche e ai gusti della persona.

Nel riconoscimento dell'utente si vuole porre particolare attenzione all'uso dei social network, i quali permettono di derivare preferenze implicite osservando ad esempio le pagine Facebook su cui ha messo "Mi piace".

Infine, il lavoro di tesi vuole sperimentare l'integrazione di piccoli e grandi schermi come espediente per invogliare il turista ad interagire con il display pubblico. Secondo quanto è emerso dallo stato dell'arte, la familiarità della persona verso il device personale (Smartphone o Tablet) stimola l'individuo ad approcciarsi con il display pubblico, superando la resistenza iniziale.

2. Il dominio

Dopo aver analizzato il contesto in cui il progetto si inserisce, si sono rilevate le seguenti entità chiave:

- L'utente, ovvero il proprietario del device;
- L'applicazione che sarà installata sul device per permettere l'integrazione;
- Il grande schermo che si occuperà di mostrare i contenuti;
- Il lettore NFC che sarà responsabile della comunicazione tra device personale e grande schermo.

Nella definizione del dominio è importante stabilire con precisione gli stakeholders, ovvero coloro a cui il progetto in questione si rivolge. Un'attenta analisi ha portato ad affermare che i principali destinatari sono tutti coloro che si trovano a visitare Milano in occasione dell'Expo e che utilizzano l'applicazione mobile per:

- conoscere ristoranti e/o punti di interesse che si trovano vicino a loro;
- interagire con il grande schermo per sapere quali ristoranti e/o punti di interesse si adattano maggiormente ai loro gusti personali;
- salvare una lista di ristoranti e/o punti di interesse.

3. I requisiti dell'applicazione

L'analisi dei requisiti è un passo importante nell'ingegneria del software in quanto permette di individuare le caratteristiche generali del prototipo che si andrà a sviluppare. Nel seguito verranno presentati i requisiti hardware e software legati all'applicazione mobile da installare sul device personale.

3.1 Versione Android

L'applicazione è stata sviluppata per il sistema operativo Android con un supporto che va dalla versione 2.3 (API di livello 10) fino all'ultima versione disponibile 4.4.2 (API di livello 19). La scelta della versione minima ricade su una funzionalità espletata dall'applicazione che richiede l'utilizzo di Google Play Services, il quale ha generato alcune problematiche con la versione 2.2 e precedenti.

Le statistiche riportate da Google nel primo trimestre del 2014 rafforzano la scelta effettuata: ad oggi la versione 2.2 ricopre una fetta di mercato molto piccola, circa 1.3%, ed è quindi ragionevole pensare che nel 2015 questa percentuale si abbasserà ancora di più (Figura 41).

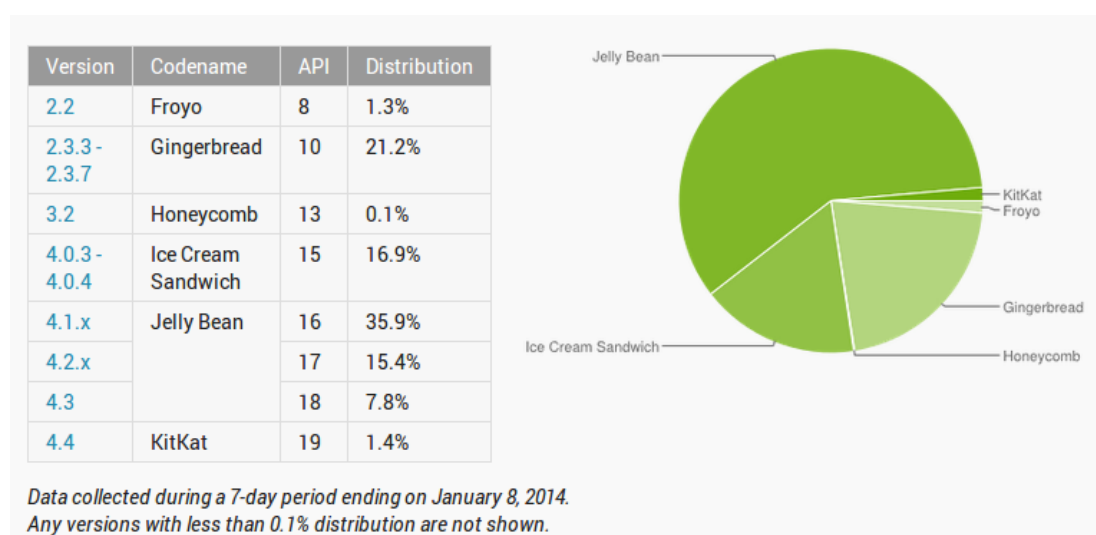


Figura 41. Statistiche Google.

3.2 Database SQLite

L'applicazione utilizza un database SQLite per salvare i dati necessari al proprio funzionamento. A differenza di altri database SQL, non ha un processo server separato ma legge e scrive direttamente su file ordinari sul disco. È molto leggero e occupa poco spazio, pertanto è la tecnologia perfetta per creare e gestire basi di dati in un ambiente come quello degli applicativi mobile dove le risorse sono limitate.

Nel progetto di tesi è stato utilizzato per salvare le informazioni anagrafiche dell'utente fornite in fase di registrazione, seguite dalle preferenze su ristoranti e/o punti di interesse. Infine, se presente, conterrà la sua lista dei preferiti.

3.3 Connessione Internet

L'applicazione, per poter funzionare correttamente, necessita di una connessione Internet costante. Infatti per poter recuperare le informazioni di ristoranti e/o punti di interesse deve comunicare con il Web Server, sul quale si trova il database centrale (vedi Paragrafo "Il database centrale" del Capitolo 4).

Le comunicazioni con il Web Server vengono effettuate via HTTP, tramite i metodi GET e POST. Il formato utilizzato per lo scambio dei dati è il JSON, e ogni comunicazione è autenticata mediante il meccanismo della HTTP Basic Authentication.

3.4 Geolocalizzazione

È richiesta l'attivazione del GPS per ottenere la lista di ristoranti e/o punti di interesse in prossimità del punto in cui si trova l'utente. Per la parte di integrazione con il grande schermo invece non è necessaria.

3.5 Risoluzione display

Affinché sia fattibile utilizzare il device come remote controller, si consiglia l'uso di Smartphone aventi una risoluzione del display maggiore o uguale a 540x960 pixel.

4. I requisiti del grande schermo

L'analisi dei requisiti è un passo importante nell'ingegneria del software in quanto permette di individuare le caratteristiche generali del prototipo che si andrà a sviluppare. Nel seguito verranno presentati i requisiti hardware e software legati al grande schermo.

4.1 Nomenclatura

Nel progetto di tesi, il grande schermo è responsabile della visualizzazione dei contenuti suggeriti dal sistema sottostante.

Per evitare inutili confusioni, con l'accezione "grande schermo" si intende l'insieme di televisore (front-end) e computer (back-end). Questo perché il televisore, di per sé, ha come unico compito quello di mostrare le informazioni; ma affinché possa visualizzare contenuti personalizzati è necessario un calcolatore che incorpori la parte relativa alla logica.

4.2 Il televisore

Televisore e calcolatore vengono collegati tramite cavo HDMI; pertanto l'unico requisito hardware imposto al televisore è la presenza della porta adatta a questo tipo di collegamento.

4.3 Il calcolatore

Il calcolatore deve aver installato Node.js (il quale a sua volta richiede la presenza di Python) per la creazione dell'applicazione server-side, e un tool per permettere la comunicazione con il database.

La base di dati che viene interrogata non è quella presente sul Web Server, ma una sua copia trasferita in locale. La replicazione sicuramente non è la scelta ottima a livello architetturale, soprattutto per l'alta probabilità di incorrere in un'inconsistenza dei dati. Per ovviare a questo problema, è stato scritto uno script

PHP che ogni giorno (ad esempio di notte) verifica la presenza di diversità tra la base di dati sul Web Server e quella in locale e, in caso affermativo, provvede a propagare le modifiche. Ciò che ha spinto gli autori ad adottare questa soluzione è la riduzione significativa del tempo nella generazione dei contenuti personalizzati, dal momento in cui le query sono eseguite in locale e non sul Web Server. E un tempo di risposta veloce è sicuramente una caratteristica desiderabile per invogliare l'utente a interagire con il grande schermo.

PROGETTAZIONE

Nell'ingegneria del software, la progettazione definisce le linee guida della struttura del prototipo, in funzione dei requisiti imposti durante l'analisi preliminare. In altre parole, ha lo scopo di definire (a un certo livello di dettaglio) la soluzione del problema. All'interno del capitolo verranno affrontate le principali componenti dell'architettura, seguite da una trattazione minuziosa sull'integrazione di device personale e grande schermo. Infine si potranno trovare gli scenari significativi, descritti attraverso i classici diagrammi forniti dall'UML.

1. Il ruolo di Facebook

Come anticipato nella fase di analisi, si vuole sperimentare l'uso dei social network nel processo di raccomandazione. Nella fattispecie si è scelto di utilizzare Facebook perché, oltre a permettere il riconoscimento dell'utente sulla base dei dati contenuti nel suo profilo, consente di derivare una serie di preferenze implicite osservando le attività del suo diario.

Affinché sia possibile integrare Facebook in questo processo, l'applicazione deve permettere una doppia modalità di registrazione: la prima classica compilando un

form con nome, cognome e altri dati ritenuti importanti, la seconda effettuando l'accesso attraverso l'account del social network.

Prima di capire come vengono recuperate le informazioni, si vuole metter in evidenza l'importanza del "Social Graph": la struttura a grafo su cui poggia Facebook. In questa struttura elementi quali utenti, pagine, gruppi, foto e video rappresentano i nodi, collegati tra loro tramite archi (o connessioni). Ad esempio, dal nodo *user* partono le connessioni *friends*, che collegano ogni utente ad altri utenti (i suoi amici), e le connessioni *likes*, che legano un utente alle pagine per cui ha espresso apprezzamento cliccando "Mi piace".

L'unica modalità di interazione con il grafo sociale è l'utilizzo di chiamate HTTP alle API di Facebook. In particolare, si trovano le GRAPH API REST indicate per la lettura e scrittura sul grafo, e le OPEN GRAPH PROTOCOL (OGP) che permettono l'inserimento di un qualsiasi oggetto nel grafo attraverso la specifica di alcuni metadati. Dal momento in cui nel progetto di tesi si punta solamente all'estrapolazione di informazioni, vengono tralasciate le OGP.

Ad ogni entità del grafo è associato un identificativo univoco da utilizzare nel momento in cui si vuole accedere alle sue proprietà. Per farlo non si deve far altro che contattare l'indirizzo <https://graph.facebook.com/<id>>, mettendo al posto di <id> l'identificativo associato all'oggetto considerato.

Il vero vantaggio sta nel modo in cui vengono restituite le risposte; il formato utilizzato è il JSON, molto utile per chi sviluppa soprattutto in ambiente Java. Per fare un esempio si può considerare la pagina ufficiale della Facebook Platform, la quale ha come identificativo 19292868552. Per accedere alle proprietà dell'elemento si contatta l'indirizzo <https://graph.facebook.com/19292868552>; per evidenti motivi di leggibilità, si riporta solo una porzione tagliata della risposta (Figura 42).

```

{
  "about": "Build, Grow, and Monetize your app with Facebook Platform solutions.",
  "category": "Product/service",
  "checkins": 1,
  "company_overview": "Facebook Platform enables anyone to build social apps on :
  "is_published": true,
  "talking_about_count": 14877,
  "username": "FacebookDevelopers",
  "website": "http://developers.facebook.com",
  "were_here_count": 0,
  "id": "19292868552",
  "name": "Facebook Developers",
  "link": "https://www.facebook.com/FacebookDevelopers",
  "likes": 2968321,
  "cover": {
    "cover_id": "10152004458663553",
    "source": "https://fbcdn-sphotos-c-a.akamaihd.net/hphotos-ak-prn2/v/t1.0-9/;
    oh=3c395eb5863c5b761df5e0045d911d22&oe=53A6FB6C&__gda__=1403815036_74993d6e028e5d;
    "offset_y": 0,
    "offset_x": 0
  }
}

```

Figura 42. Porzione di risposta JSON.

Nel progetto di tesi ciò che è importante recuperare sono le informazioni contenute nel profilo utente, i like assegnati e le locations che ha postato nel suo diario. Gli indirizzi che consentono il recupero di questi dati sono rispettivamente:

https://graph.facebook.com/me? access_token =< access_token >,

https://graph.facebook.com/me/likes? access_token =< access_token >,

https://graph.facebook.com/me/locations? access_token =< access_token >.

Gli URL di richiesta presentati mostrano due parametri importanti: *me* e *access_token*. Il primo non è altro che uno speciale identificatore usato per fare riferimento all'utente corrente. Il secondo, invece, riguarda il processo di accesso autenticato e autorizzato alle informazioni contenute nel grafo sociale di Facebook. Questo meccanismo è basato sul protocollo OAuth 2.0 che vincola l'acquisizione di un *access_token* prima di effettuare qualsiasi altra operazione. L'*access token* è una stringa casuale il cui scopo è identificare la sessione, alla quale sono stati accordati i permessi per un determinato insieme di azioni.

Nel lavoro di tesi, prima di poter accedere alle informazioni desiderate, si è dovuto specificare il set di dati da estrapolare (Figura 43).

Select Permissions

User Data Permissions Friends Data Permissions Extended Permissions

<input checked="" type="checkbox"/> user_about_me	<input type="checkbox"/> user_actions.books	<input type="checkbox"/> user_actions.music
<input type="checkbox"/> user_actions.news	<input type="checkbox"/> user_actions.video	<input checked="" type="checkbox"/> user_activities
<input checked="" type="checkbox"/> user_birthday	<input checked="" type="checkbox"/> user_checkins	<input type="checkbox"/> user_education_history
<input type="checkbox"/> user_events	<input checked="" type="checkbox"/> user_friends	<input type="checkbox"/> user_games_activity
<input type="checkbox"/> user_groups	<input checked="" type="checkbox"/> user_hometown	<input checked="" type="checkbox"/> user_interests
<input checked="" type="checkbox"/> user_likes	<input checked="" type="checkbox"/> user_location	<input checked="" type="checkbox"/> user_notes
<input type="checkbox"/> user_online_presence	<input type="checkbox"/> user_photo_video_tags	<input type="checkbox"/> user_photos
<input type="checkbox"/> user_questions	<input checked="" type="checkbox"/> user_relationship_details	<input checked="" type="checkbox"/> user_relationships
<input type="checkbox"/> user_religion_politics	<input type="checkbox"/> user_status	<input type="checkbox"/> user_subscriptions
<input type="checkbox"/> user_videos	<input type="checkbox"/> user_website	<input type="checkbox"/> user_work_history

Basic permissions included by default.

Get Access Token Clear Cancel

Select Permissions

User Data Permissions Friends Data Permissions **Extended Permissions**

<input type="checkbox"/> ads_management	<input type="checkbox"/> ads_read	<input type="checkbox"/> create_event
<input type="checkbox"/> create_note	<input type="checkbox"/> email	<input checked="" type="checkbox"/> export_stream
<input type="checkbox"/> manage_friendlists	<input type="checkbox"/> manage_notifications	<input type="checkbox"/> manage_pages
<input type="checkbox"/> photo_upload	<input type="checkbox"/> publish_actions	<input type="checkbox"/> publish_checkins
<input type="checkbox"/> publish_stream	<input type="checkbox"/> read_friendlists	<input type="checkbox"/> read_insights
<input type="checkbox"/> read_mailbox	<input type="checkbox"/> read_page_mailboxes	<input type="checkbox"/> read_requests
<input checked="" type="checkbox"/> read_stream	<input type="checkbox"/> rsvp_event	<input type="checkbox"/> share_item
<input type="checkbox"/> sms	<input type="checkbox"/> status_update	<input type="checkbox"/> video_upload
<input type="checkbox"/> xmpp_login		

Basic permissions included by default.

Get Access Token Clear Cancel

Figura 43. Impostazione permessi Facebook.

2. L'integrazione di piccolo e grande schermo

Prima di iniziare la trattazione riguardante l'integrazione di personal device e grande schermo, si vuole esporre brevemente le esigenze alla base del progetto di tesi che hanno spinto gli autori a utilizzare una tecnologia piuttosto che un'altra.

Prima di tutto è necessario creare una connessione sicura tra i due dispositivi per proteggere i dati sensibili dell'utente da possibili attacchi di sniffing e phishing. Tuttavia, si vuole evitare di ricorrere all'inserimento di codici/pin perché, come già osservato nello stato dell'arte, risulterebbe più invasivo nei confronti dell'utente.

Altro punto importante riguarda il tempo di risposta: si desidera una discreta velocità di connessione affinché la persona possa interagire in real-time con il grande schermo, limitando il più possibile i tempi morti che annoierebbero l'utente costringendolo ad allontanarsi.

Infine occorre che il canale instaurato tra i dispositivi rimanga sempre aperto fino alla chiusura della connessione; in questo modo il grande schermo può rimanere in ascolto delle richieste inviate dall'utente tramite il device personale.

2.1 WebSockets

Il Socket è un'astrazione software progettata per permettere a due processi, risiedenti su macchine diverse, di poter trasmettere e ricevere dati attraverso una rete. Viene creato dinamicamente e ha un ciclo di vita limitato: viene aperto, utilizzato per la lettura/scrittura e infine chiuso. Metaforicamente può essere pensato come un lungo "tubo" dove alle estremità si trovano le entità che intendono comunicare.

Si tratta di un componente fondamentale nelle comunicazioni di rete di tipo client-server, dove il server si deve preoccupare di creare tanti Socket quanti sono i client a lui connessi. In informatica questo genere di architettura prevede che un'entità, detta client, si connetta ad un'altra entità, chiamata server, per la fruizione di un certo servizio (il classico esempio è la condivisione delle risorse). Nell'ambito Web, il browser si comporta come un client che sfrutta il protocollo HTTP per richiedere

pagine ad un server remoto. L'architettura client-server si presta molto bene in un contesto dove la richiesta viene fatta in modo esplicito, ad esempio attraverso il click su un determinato link, mentre non è adatta a situazioni in cui è necessaria una comunicazione bidirezionale.

Nel cercare di capire quale potesse essere la soluzione migliore per permettere l'integrazione di personal device e grande schermo, si è incontrata la tecnica del long-polling in grado di gestire eventi server-side. Il client appena avviata l'applicazione effettua una chiamata verso il server; se quest'ultimo non ha eventi da comunicare mantiene aperta la connessione e la imposta in uno stato di "pending" in attesa che accada qualcosa. In questo modo esiste un canale diretto tra client e server da utilizzare per la comunicazione; nonostante non sia particolarmente difficile da implementare, questa tecnica non può essere eseguita in qualsiasi Web Server proprio per la caratteristica di avere richieste in stato "pending". I principali Web Server e Application Server non offrono questa funzionalità nativa perché agiscono, diversamente da Node.js, in maniera sincrona; hanno a disposizione un numero finito di "slot di richieste", terminato il quale sono obbligati a rilasciarne qualcuna.

Riprendendo il discorso di integrazione tra personal device e grande schermo, nel progetto di tesi si è utilizzata una nuova tecnologia introdotta con l'HTML5: il WebSocket. Basata su un set di API, è in grado di fornire canali di comunicazione full-duplex attraverso l'utilizzo di un singolo Socket. Rispetto al long-polling è più efficiente in quanto ha una latenza più bassa, il canale non viene chiuso per scadenza del timeout e permette l'invio di messaggi da server a client quasi in real-time.

Si basa sul protocollo TCP e la sua unica correlazione con l'HTTP è nel modo in cui fa l'handshake. Le comunicazioni sono in genere fatte attraverso la porta TCP 80, ma se ne possono utilizzare altre. È supportato attualmente da numerosi browser, incluso Google Chrome, Internet Explorer, Firefox, Safari e Opera.

Per poter utilizzare i WebSockets è necessario implementare i componenti client e server; l'aspetto cruciale risiede chiaramente in quest'ultimo che dovrà essere sufficientemente robusto per permettere l'instaurazione della connessione.

Relativamente alla parte client è indispensabile la conoscenza del linguaggio Javascript, mentre sul lato server si può scegliere tra Java, PHP, Node.js, Python e altri minori. In questa tesi, si è optato per l'utilizzo di Node.js in quanto si avvicina molto al linguaggio Javascript e permette di gestire in maniera semplice e veloce i WebSockets.

Ovviamente questa tecnologia non rappresenta la soluzione a tutti i problemi: l'HTTP riveste ancora un ruolo chiave nella comunicazione tra client e server per trasferimento dati di tipo one-time. Inoltre i WebSockets possono essere utilizzati solo se gli utenti utilizzano moderni browser con Javascript abilitato, e bisogna considerare possibili impatti sull'architettura di rete; infatti mantenendo una connessione persistente potrebbe richiedere molte più risorse rispetto ad un Web Server standard. Infine vi è la possibilità che i proxy possano interferire con le comunicazioni dal momento che usano il meccanismo di "upgrade" del protocollo HTTP; potrebbe accadere che il browser li supporti, ma non si possa usufruirne a causa di qualche proxy o server intermedio che si rifiuta di farli passare.

Per implementare il WebSocket lato client è necessario istanziare la connessione con il server con cui si è deciso di instaurare il dialogo. A tal proposito si utilizza un costruttore, al quale viene passato come parametro l'url del server WebSocket e, opzionalmente, una stringa o un array di stringhe (Figura 44). Queste ultime vengono utilizzate per indicare dei sotto-protocolli, per permettere, ad esempio, che un singolo server possa gestire diversi tipi di interazioni a seconda del protocollo specificato.

```
WebSocket WebSocket (
    in DOMString url,
    in optional DOMString[] protocols
);
```

Figura 44. Costruttore WebSocket.

Nell'esempio seguente si effettua la connessione ad un server pubblico che offre un servizio di risposta echo, attraverso l'url "ws://echo.WebSocket.org/" (Figura 45). Nel progetto di tesi, invece, si fa riferimento ad un server appositamente creato per supportare la comunicazione con l'applicazione; è necessario quindi modificare l'url di destinazione lasciando inalterata la sigla "ws://". Eventualmente al posto di "ws://" si può utilizzare "wss://" che definisce una protezione sicura per lo scambio di dati (l'analogo di HTTPS).

```
var connection = new WebSocket('ws://echo.WebSocket.org/');
```

Figura 45. Apertura connessione lato client.

Se l'operazione va a buon fine, si avrà a disposizione un oggetto connection le cui API permettono di inviare e ricevere messaggi.

I WebSockets mettono a disposizione quattro eventi fondamentali (Figura 46):

1. onopen, chiamato all'apertura di un nuovo WebSockets;
2. onerror, chiamato quando si verifica un errore nella connessione tra client e server;
3. onmessage, chiamato quando il client riceve un messaggio da parte del server;
4. onclose, chiamato quando viene chiusa la connessione.

```
//Quando apriamo la connessione mandiamo un messaggio di testo
//con scritto 'Ping' al server
connection.onopen = function() {
    connection.send('Ping');
};

//Stampiamo sulla console l'errore che si è presentato
connection.onerror = function(error) {
    console.log('WebSocket Error: ' + error);
};
```

```

//Recuperiamo i messaggi inviati dal server
connection.onmessage = function(e) {
    console.log('Server: ' + e.data);
};

//Chiudiamo la connessione
connection.onclose = function() {
    connection.close();
};

```

Figura 46. Eventi WebSocket.

Dopo aver realizzato il client, si passa all'implementazione del server WebSocket. Nel progetto di tesi si è scelto di utilizzare Node.js, ma si possono adoperare anche altri linguaggi di programmazione come C++, Python, PHP e così via. Nel Web si possono trovare esempi di codice funzionanti che possono essere riadattati per implementare il server su misura.

Come già anticipato, nella creazione della connessione è possibile definire uno o più sotto-protocolli che servono per gestire diverse modalità di comunicazione tra client e server. Di fatto il protocollo WebSocket è visto come quello portante mentre i sotto-protocolli permettono di specializzare la tipologia della connessione in base alla sua funzione; riprendendo l'esempio del Web, il protocollo portante è il TCP/IP mentre i sotto-protocolli sono HTTP e FTP.

```

var connection = request.accept('echo-protocol', request.origin);

```

Figura 47. Connessione lato server.

```

var connection = new WebSocket('ws://localhost:8080/',
    'echo-protocol');

```

Figura 48. Connessione lato client.

Dopo aver implementato correttamente il client e il server si può procedere con l'handshake. In questa delicata fase i dettagli della connessione vengono negoziati e

ciascuna delle parti può decidere di uscirne, anche prima del completamento, nel caso in cui le condizioni si rivelassero sfavorevoli.

Il client deve per prima cosa avviare il processo di WebSocket handshake attraverso un comando HTTP, chiamato Upgrade, introdotto con il protocollo HTTP/1.1.

Grazie agli strumenti per sviluppatori messi a disposizione da Google Chrome, è possibile visualizzare l'handshake; l'header della richiesta e della risposta mostrano preziose informazioni (Figura 49).

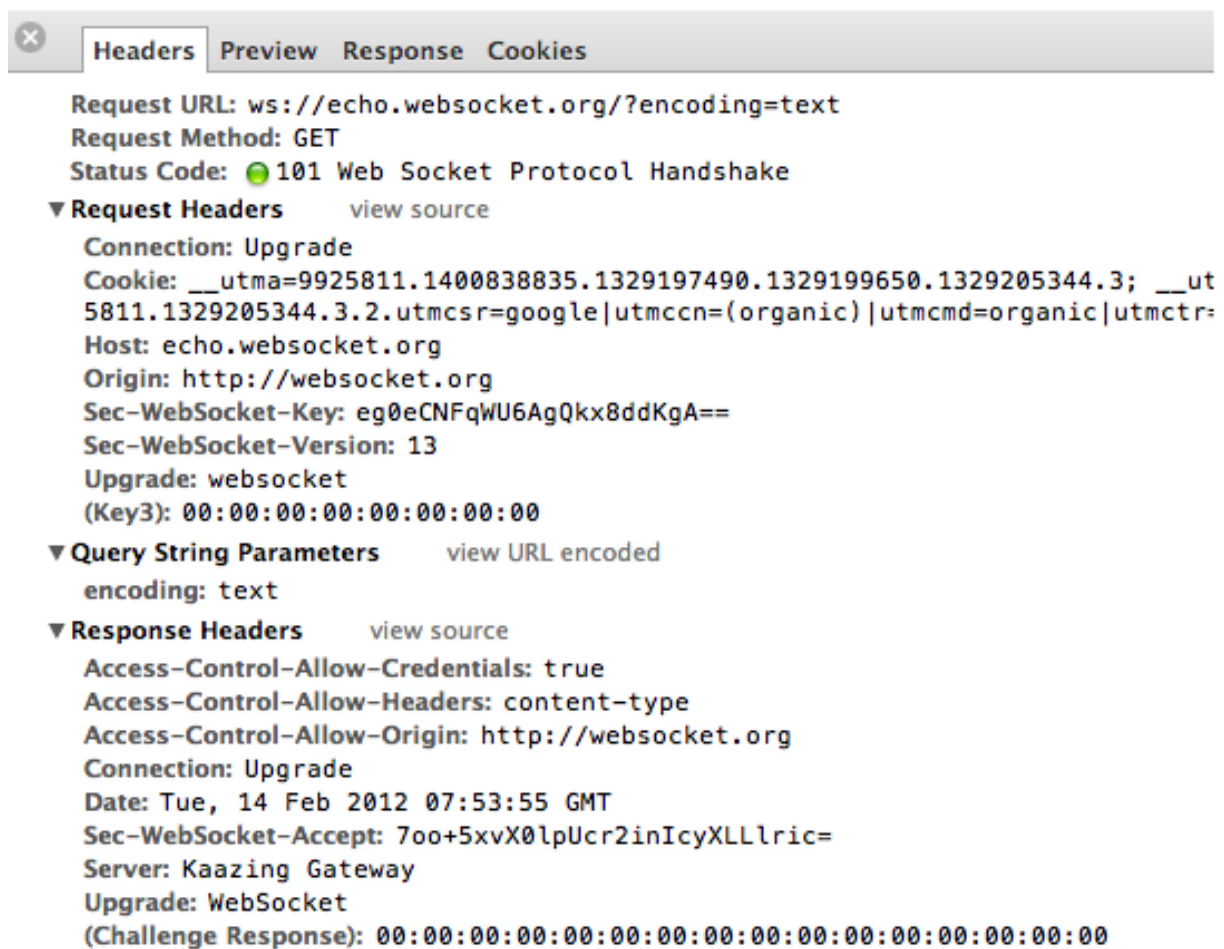


Figura 49. Header dell'handshake.

Come prima cosa si nota il valore "Upgrade" associato alla *connection*, che sta ad indicare l'apertura di una connessione di tipo WebSocket.

Secondariamente il campo *origin* presente nella richiesta permette di conoscere il mittente dei messaggi, facendo sì che il server possa sapere se essi provengono da una fonte autorizzata o meno.

Infine si trova la chiave di sicurezza inviata dal client *Sec – WebSocket – Key* che compone la prima parte dell'handshake; è generata in modo casuale e codificata come una stringa base64 di 16 byte.

Il server, una volta ricevuta la richiesta, va a prendere il valore del campo *Sec – WebSocket – Key*. Ad essa concatena la stringa `258EAF5 – E914 – 47DA – 95CA – C5AB0DC85B11` e sottopone il tutto all'algoritmo di hashing SHA-1. Il risultato viene salvato nel campo *Sec – WebSocket – Accept* presente nell'header della risposta del server.

Se l'handshake si conclude con successo l'oggetto connection di tipo WebSocket creato precedentemente avrà il campo *readyState* impostato a 1; gli altri valori che potrebbe assumere sono 0 (in connessione), 2 (in chiusura) e 3 (canale chiuso). Tale variabile permette di identificare in che stato si trova la connessione.

2.2 NFC

Per NFC (acronimo di Near Field Communication) si intende una tecnologia di comunicazione a radiofrequenza bidirezionale sviluppata congiuntamente da Philips, Sony e Nokia. Si parla di "tecnologia a raggio corto" in quanto la comunicazione viene instaurata quando due dispositivi vengono accostati entro un raggio limitato: a livello teorico si stima un massimo di 10-20 cm, ma nella pratica si garantisce 4 cm.

Questa tecnologia mette a disposizione tre modalità di comunicazione:

1. *Card emulation*

Permette ad un dispositivo di emulare un Tag NFC secondo gli standard ISO 14443-A e ISO 14443-B. I dati devono essere gestiti in maniera sicura all'interno del dispositivo (tipicamente uno Smartphone) che si comporta come un componente passivo.

2. *Reader/writer*

In questa modalità ad un dispositivo è consentito leggere e scrivere su un Tag NFC passivo.

3. *Peer-to-peer*

Consente la creazione di un canale di comunicazione tra due dispositivi lungo il quale avviene lo scambio delle informazioni. In maniera alternata ogni device si comporterà da Reader o da Tag NFC secondo lo standard ISO 18092.

Stabilita la modalità di comunicazione, l'interazione può avvenire secondo:

- *Phone to Phone*: due Smartphone NFC possono comunicare direttamente in modalità peer-to-peer.
- *Phone to Device*: uno Smartphone NFC può comunicare con un altro dispositivo NFC (tipicamente un lettore di prossimità per rilevazione della presenza, controllo di accesso, ...) sempre in modalità peer-to-peer.
- *Phone to Tag/Sticker/SmartCard*: uno Smartphone può leggere o scrivere dati su un tag/sticker/smartcard NFC.
- *Phone to Reader*: uno Smartphone NFC in modalità card emulation può interagire con un lettore NFC fornendo i dati memorizzati attraverso il Secure Element.

Diversamente dal sistema Bluetooth non è necessario alcun codice per stabilire la connessione potendo così godere di un handshake più veloce. Inoltre è stata sviluppata con una particolare attenzione alla sicurezza dei dati e delle transazioni grazie ad un sistema con crittografia DES.

Prima di spostare definitivamente l'attenzione sull'NFC si è preso in considerazione anche la tecnologia chiamata WiFi Direct, la quale permette la creazione di una rete peer-to-peer tra due dispositivi con il risultato di uno scambio di dati effettuato alla velocità di una rete WiFi. Nonostante questo grande vantaggio, si è dovuto

escludere questa soluzione in quanto supportata solo da dispositivi Android con versione superiore alla 4.0.

Nel progetto di tesi l'NFC viene impiegato per permettere al grande schermo di riconoscere la persona che ha di fronte: chi è, quanti anni ha, da dove viene, quali sono i suoi gusti e così via. Se l'utente ha effettuato il login con Facebook, vengono estrapolate le informazioni demografiche contenute nel profilo, i like che ha fatto e le locations presenti sul diario. Se l'utente ha optato per la registrazione classica, si fa riferimento ai dati che egli ha inserito durante questa fase iniziale. In entrambi i casi vengono anche recuperate le eventuali preferenze dichiarate dall'utente mediante l'apposita funzionalità presente nell'applicazione.

Per permettere questo riconoscimento l'individuo deve prendere il proprio Smartphone/Tablet (dotato ovviamente della tecnologia NFC), avvicinarsi al grande schermo e appoggiarlo sul lettore. Stabilita la comunicazione, il display invia il proprio indirizzo IP permettendo al dispositivo mobile di collegarsi e iniziare quindi l'interazione.

I sistemi operativi che supportano pienamente la tecnologia NFC sono Android e Windows Phone, e sul mercato si stanno sempre più affermando i dispositivi che la integrano. La domanda sorge spontanea: come può un utente dotato di iPhone o Smartphone Android senza questa tecnologia interfacciarsi con il grande schermo? Fortunatamente si possono utilizzare alcuni particolari adattatori creati da alcune aziende, quali la statunitense Moneto. Per Android viene fornita una MicroSD, in grado di "trasformare" lo Smartphone in un dispositivo NFC, e un adesivo da applicare alla batteria che amplifica il segnale. Per Apple, non essendoci un ingresso MicroSD ed essendo la batteria incorporata nel case, si è creata una cover in grado di dotare lo Smartphone di tale tecnologia.

In ogni caso i dati provenienti da uno studio della IHS Technology⁵ ci rassicurano: l'NFC sarà incluso nel 64% dei telefoni cellulari nel 2018, e già nel 2015 si può

⁵ IHS Inc. (IHS) è una società con sede a Douglas County, Colorado, Stati Uniti. Serve clienti internazionali in nove grandi settori, tra cui aerospaziale, chimico, energetico, finanziario, tecnologico e altro ancora. Tra le sue mansioni si ricorda in particolar modo i servizi di consulenza e la creazione di applicazioni software personalizzate.

vedere un raddoppio di dispositivi dotati di questa tecnologia rispetto al 2013 (Figura 50).

“ La maggior parte dei produttori di Smartphone stanno adottando le comunicazioni wireless NFC e tecnologia di pagamento nei loro prodotti come standard de facto.”

dichiara Don Tait – analista senior dei mercati finanziari di IHS

“ Questo forte impulso consentirà al mercato dei cellulari NFC di superare le barriere, tra cui la mancanza di servizi e applicazioni interessanti, e il lento progredire della creazione delle infrastrutture necessarie.”

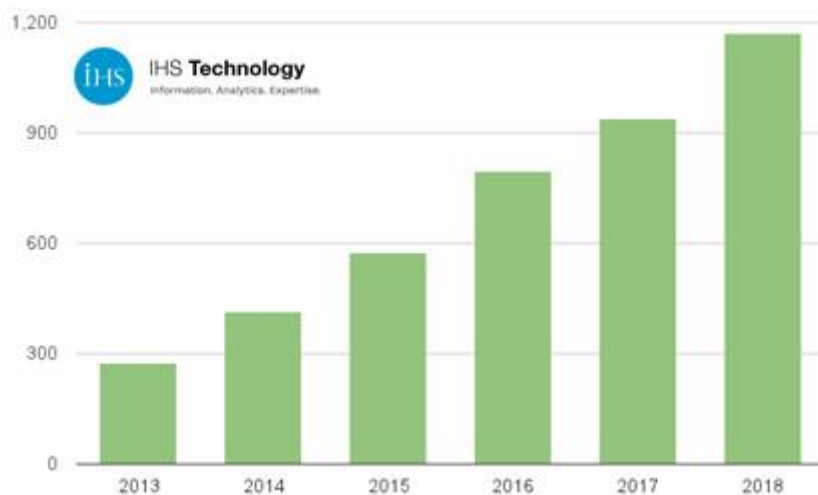


Figura 50. Previsioni per spedizioni mondiali di telefoni NFC, in milioni.

3. Il database centrale

Come già anticipato nel capitolo precedente, affinché il grande schermo possa presentare contenuti personalizzati è necessario costruire una base di dati che contenga le informazioni su ristoranti e punti di interesse della città di Milano. La

costruzione del database è avvenuta applicando la tecnica di Web Scraping al portale di TripAdvisor, reperibile all'indirizzo <http://www.tripadvisor.co.uk>.

3.1 Perché TripAdvisor?

TripAdvisor è il sito di viaggi più grande del mondo⁶, nato per aiutare i viaggiatori a pianificare la vacanza perfetta. Offre consigli di viaggio affidabili, pubblicati da veri viaggiatori, e un'ampia serie di funzionalità di ricerca con collegamenti diretti agli strumenti di prenotazione. I siti a marchio TripAdvisor rappresentano la più grande community di viaggiatori del mondo, con oltre 200 milioni di visitatori ogni mese⁷ e oltre 100 milioni di recensioni e opinioni.

Un'idea nata a cavallo del nuovo millennio ma che, a differenza di molte altre, si è rivelata vincente. La sua peculiarità è stata sempre quella di coinvolgere gli stessi viaggiatori nell'elargire informazioni e consigli grazie alla possibilità di poter scrivere la recensione del viaggio sul sito stesso.

Dalla sua fondazione sono passati diversi anni ma il principio di TripAdvisor è rimasto immutato. Nonostante la crescita esponenziale, le vulnerabilità non sono passate inosservate in particolare quelle relative all'alto numero di recensioni false. Nel settore alberghiero, le recensioni false si possono ritrovare in consigli scritti da utenti che non sono mai stati ospiti dell'hotel recensito, oppure valutazioni scritte dallo stesso albergatore allo scopo di farsi pubblicità.

Nonostante questa pecca, uno studio svolto da PhoCusWright⁸ mostra che l'87% dei rispondenti al sondaggio ritiene le recensioni uno stimolo a sentirsi più sicuri nelle decisioni, e che 8 persone su 10 sono d'accordo nell'affermare che le recensioni di TripAdvisor aiutano a migliorare l'esperienza di viaggio.

⁶ Fonte: comScore Media Metrix per i siti di TripAdvisor Worldwide, Gennaio 2013.

⁷ Fonte: Google Analytics, dati mondiali, Aprile 2013.

⁸ Fonte: sondaggio condotto da PhoCusWright nel settembre 2012 su 2.739 rispondenti. I partecipanti allo studio sono stati coinvolti casualmente attraverso un link di invito pop-up su TripAdvisor.com

“ Tutti i viaggiatori del mondo vogliono organizzare il viaggio perfetto e niente aiuta nella pianificazione più delle recensioni autentiche di persone reali che hanno precedentemente visitato la destinazione di interesse.”

dichiara Stephen Kaufer – cofondatore e amministratore delegato di TripAdvisor

“ Avere a disposizione 100 milioni di recensioni e opinioni, offre una panoramica straordinaria per ogni decisione di viaggio e ringraziamo la nostra appassionata community per le loro opinioni che continuano sempre più ad aiutare i viaggiatori a organizzare il viaggio perfetto.”

3.2 La tecnica di Web Scraping

Una grande quantità di informazioni è presente sul Web ma raramente si presenta nella forma desiderata dall'utente. Il risultato è che le persone si trovano costrette a spendere ore e ore per copiare dati in fogli di calcolo Excel o documenti Word. Oltre all'evidente uso inefficiente del proprio tempo, la presenza di grandi volumi di dati rende questa operazione tanto onerosa quanto complessa.

Con il termine Web Scraping si fa riferimento ad un processo di estrazione dati da pagine HTML, al fine di trasformare il contenuto prelevato in informazione strutturata utilizzabile in altri contesti. Si basa sul funzionamento di alcuni programmi software che automaticamente eseguono ciò che la persona fa quando seleziona, copia e incolla un testo.

Il processo di Web Scraping parte con una fase iniziale, detta crawling, dove si vanno a recuperare le pagine HTML interessate; dato un indirizzo di partenza (o un insieme di indirizzi) e alcune condizioni (ad esempio il numero massimo di link da percorrere e il tipo di file da ignorare), il crawler scarica tutte le pagine visitate seguendo i link tra di esse.

Ogni pagina scovata dal crawler viene elaborata dall'applicativo di scraping per trovare ed estrarre le informazioni ritenute interessanti. Tale operazione può essere svolta in più modalità: tramite l'utilizzo delle classi CSS, imprescindibili per lo sviluppo grafico delle pagine Web e quindi sempre presenti, oppure tramite

l'ispezione del DOM⁹. I dati estratti vengono salvati e conservati in diversi formati di database come MySQL, MS Access, e così via.

In linea teorica qualsiasi contenuto Web visibile tramite un browser può essere scaricato dalla rete e processato. Ma tutto ciò è legale? Anche se il contenuto è online non significa che lo si può prendere e utilizzare come meglio si crede. Ci sono dei vincoli legali, morali ed etici da rispettare, in particolar modo se si usano tali informazioni in un contesto commerciale a scopo di lucro.

A seconda delle finalità con cui si maneggiano i dati estratti, i processi di crawling e scraping possono violare i termini di contratto e di utilizzo imposti dal sito target. Essi infatti sono regolamentati da uno "standard de-facto" (il protocollo di esclusione dei robot) che consiste in un file robots.txt posto nella cartella home del sito Web – <https://<server>/robots.txt>. Questo file è solitamente utilizzato per indicare la politica adottata relativamente al crawling e allo scraping dei contenuti, comunicando i percorsi HTTP visitabili e indicizzabili dal sito target.

Il protocollo di esclusione dei robot, non essendo integrato in quello HTTP, non forza in alcun modo il crawler a seguire e rispettare le regole imposte; per questo motivo i siti che proibiscono esplicitamente il processo di scraping (ad esempio Facebook) adottano spesso tecniche difensive.

3.3 Web Scraping di TripAdvisor

Il processo di Web Scraping si compone di pochi e semplici step (Figura 51):

1. **Connect**, per permettere il collegamento al sito remoto tramite HTTP o FTP;
2. **Extract**, che consente di estrapolare le informazioni ritenute interessanti;
3. **Process**, che consiste nel filtrare i dati ricavati mettendoli in un formato utile e comprensibile al contesto di lavoro;
4. **Save**, per effettuare il salvataggio delle informazioni nel formato desiderato.

⁹ DOM (Document Object Model) è una forma di rappresentazione dei documenti strutturati; non pone particolari limitazioni alla struttura dati di un documento e permette la sua visualizzazione sotto forma di albero. Un classico esempio di DOM è l'albero dei tag HTML di cui è costituita una pagina web.



Figura 51. Fasi del Web Scraping.

Comprese le fasi di cui si compone il processo, è importante analizzare la struttura della pagina Web per individuare le informazioni che si vogliono estrarre; una volta che si ha ben chiaro il “dove cercare” e il “cosa prendere”, si scrive il codice che in modo automatico esegue il lavoro.

Prima di riportare la struttura del database finito, si ritiene opportuno inserire le porzioni significative di codice che hanno permesso l’extrapolazione dei dati.

Come specificato nella precedente sezione, la prima cosa da fare è individuare la pagina Web da cui iniziare il processo di scraping. Una volta nota questa informazione, è opportuna salvarla in una variabile testuale da utilizzare all’interno del codice. Nel progetto di tesi, la stringa *html* è stata formattata per contenere due parametri aggiuntivi, *strIdCitta* e *strCitta*, che rappresentano rispettivamente l’id e il nome della città di cui si sta facendo Web Scraping. L’introduzione di questi parametri rende il codice riutilizzabile nel caso in cui si decidesse di raccogliere i dati su ristoranti e punti di interesse di un’altra città oltre Milano.

Dopo aver impostato correttamente la stringa, è possibile effettuare la connessione alla pagina specificata e ottenere le informazioni desiderate. Le porzioni di codice

riportate in questo paragrafo fanno riferimento all'extrapolazione dei dati riferiti ai ristoranti; analoghi ragionamenti possono essere estesi per il recupero delle informazioni sui punti di interesse e, in forma minore, sugli hotel.

Ad ogni modo, una volta effettuata la connessione, si riesce a ottenere la lista dei ristoranti mostrata da TripAdvisor filtrando gli oggetti della classe *listing* presenti nel sorgente della pagina (Figura 52 e Figura 53).

```
www.tripadvisor.co.uk/Restaurants-g187849-Milan_Lombardy.html ☆  
  
html = "http://www.tripadvisor.co.uk/Restaurants-g".  
      concat(strIdCitta).concat("-").  
      concat(strCitta).concat(".html");  
doc = Jsoup.connect(html).timeout(0).get();  
ristoranti = doc.getElementsByClass("listing");
```

Figura 52. Codice per connessione a TripAdvisor – sezione ristoranti.

Cuisine

- All (4,881)
- African (16)
- American (116)
- Asian (97)
- Bakery (53)
- Barbecue (23)

[More cuisines](#)

Price

- All (4,881)
- £ (59)
- ££ (256)
- £££ (289)
- ££££ (335)

Neighbourhoods

- All (4,881)
- Centro Storico (550)
- Isola (1)
- Magenta (67)
- Porta Ticinese (94)
- Porta Venezia (79)

Restaurants travellers recommend

4,881 of 4,881 shown

Sorted by

Pasticceria Castelnuovo

Ranked #1 of 4,881 in Milan
●●●●● 635 reviews

"Friendly service, fantastic pastri..." 22/02/2014
"Excellent food, lovely owner!" 10/02/2014

Cuisines: Dessert, Café, Tea Room
[Map](#) | [Visitor photos \(100\)](#)

La Gelateria della Musica

Ranked #2 of 4,881 in Milan
●●●●● 1,310 reviews

"Superb Ice-Cream" 31/03/2014
"The choice of gelato was very wide..." 14/03/2014

Cuisines: Ice cream
[Map](#) | [Visitor photos \(54\)](#)

Figura 53. Lista di ristoranti mostrata da TripAdvisor.

Analizzando la lista di ristoranti ottenuta, si definisce l'insieme delle informazioni utili per la costituzione della base di dati. In questa pagina ciò che è importante

recuperare è il nome, il voto medio, il dettaglio della cucina, il dettaglio del prezzo e l'indirizzo della pagina Web di ogni ristorante mostrato (Figura 54).

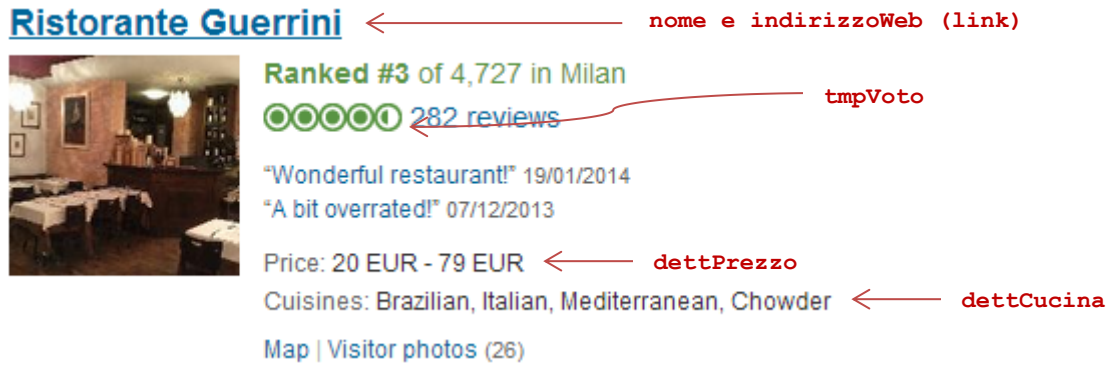


Figura 54. Zoom su un ristorante della lista.

Per recuperare le suddette informazioni, si effettua un ciclo for e, per ogni ristorante presente nella lista, si vanno a filtrare le informazioni sulla base di particolari classi e attributi presenti nel sorgente della pagina Web. Ad esempio per prendere il dettaglioCucina si utilizza la classe *cuisine*, mentre per il recupero dell'indirizzoWeb si utilizza l'attributo *href* (Figura 55).

```

for (Element ristorante : ristoranti) {
    inizializza();
    Elements titoli = ristorante.getElementsByClass("property_title");
    Elements noContent = ristorante.getElementsByClass("noContent");
    if (noContent.isEmpty()) {
        for (Element titolo : titoli) {
            nome = titolo.text();
            indirizzoWeb = titolo.attr("href");
        }
    }
    Elements voti = ristorante.getElementsByClass("sprite-ratings");
    if (!voti.isEmpty())
        for (Element voto : voti)
            tmpVoto = Float.valueOf(voto.attr("content"));
    Elements cucine = ristorante.getElementsByClass("cuisine");
    if (!cucine.isEmpty())
        for (Element cucina : cucine)
            dettCucina = cucina.text();
    Elements prezzi = ristorante.getElementsByClass("price");
    if (!prezzi.isEmpty())
        for (Element prezzo : prezzi)
            dettPrezzo = prezzo.text();
    salvaRistorante(nome, dettCucina, dettPrezzo, tmpVoto, indirizzoWeb);
}
}

```

Figura 55. Codice per recupero info principali.

Sfruttando l'indirizzo Web recuperato, si può raggiungere la pagina di dettaglio di ogni ristorante ed estrarre ulteriori informazioni significative (Figura 56 e Figura 57).

```
String htmlElemento = "http://www.tripadvisor.co.uk";  
htmlElemento = htmlElemento.concat(indirizzoWeb);  
doc = Jsoup.connect(htmlElemento).timeout(0).get();
```

Figura 56. Codice per connessione pagina di dettaglio ristorante.



Figura 57. Pagina di dettaglio ristorante.

Analizzando la pagina di dettaglio, si ritiene utile recuperare l'indirizzo e la località del ristorante in questione. Per estrarre queste informazioni si utilizzano rispettivamente le classi *street – address*, *extended – address* e *locality* presenti nel sorgente della pagina Web (Figura 58 e Figura 59).



Figura 58. Ubicazione ristorante.


```

Elements indirizziVia = doc.getElementsByClass("street-address");
if (!indirizziVia.isEmpty())
    for (Element indirizzoVia : indirizziVia)
        dettIndirizzo = indirizzoVia.text();
Elements indirizziExtended = doc.getElementsByClass("extended-address");
if (!indirizziExtended.isEmpty())
    for (Element indirizzoExtended : indirizziExtended)
        dettIndirizzoExt = indirizzoExtended.text();
Elements indirizziCAP = doc.getElementsByClass("locality");
if (!indirizziCAP.isEmpty())
    for (Element indirizzoCAP : indirizziCAP)
        dettLocalita = indirizzoCAP.text();

```

Figura 59. Codice per recupero info geografiche.

Sempre nella pagina di dettaglio, potrebbero esser presenti dettagli aggiuntivi in grado di descrivere meglio il ristorante. Per estrarre queste informazioni si utilizza la classe *listing_details* presente sempre nel sorgente della pagina Web (Figura 60, Figura 61 e Figura 62).

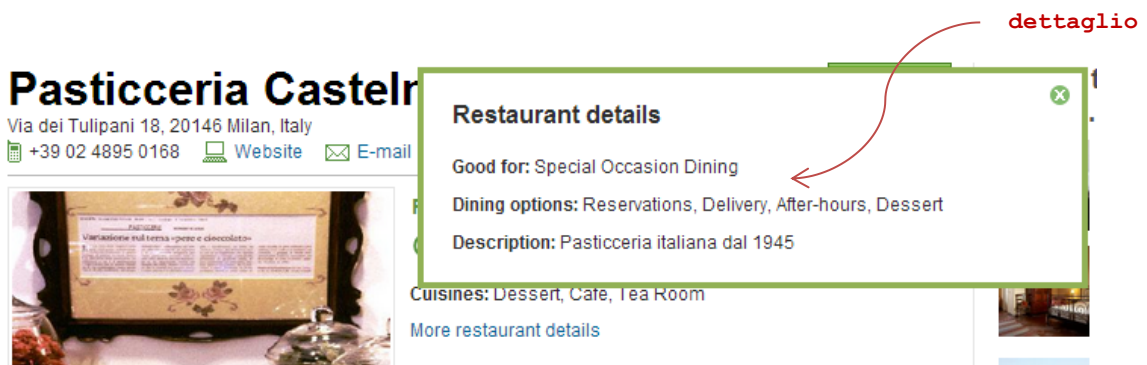


Figura 60. Dettagli aggiuntivi primo tipo.

La Piccola Bottega dei Sapori

Via Magellano, 1 | Ingresso da Via Imbonati, Milan, Italy

Add to trip

02 36517463 Website E-mail Update restaurant details



Ranked #110 of 4,881 restaurants in Milan

dettaglio

16 Reviews

Cuisines: Italian, Mediterranean, Gastropub, Pasta

Dining options: Reservations, Takeout

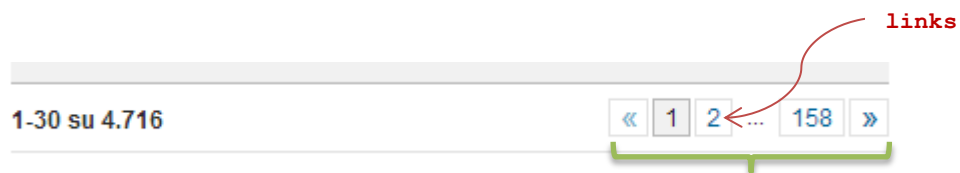
Description: La Piccola Bottega dei Sapori è un piccolo negozio che nasce nel 2011, da un insieme di passioni, quella per la buona cucina italiana, per le tradizioni regionali e le specialità locali. L'amore per il mondo del vino e della birra, e l'armonia dei loro accostamenti con il cibo. Per la bontà e la genuinità dei prodotti

Figura 61. Dettagli aggiuntivi secondo tipo.

```
html = "http://www.tripadvisor.co.uk/" + indirizzoWeb;  
doc = Jsoup.connect(html).timeout(0).get();  
Elements details = doc.getElementsByClass("listing_details");  
if (details != null)  
    for (Element detail : details) {  
        Elements typeDetails = detail.getElementsByClass("detail");  
        if (!typeDetails.isEmpty())  
            for (Element typeDetail : typeDetails)  
                if (!dettaglio.contains(typeDetail.text()))  
                    dettaglio = dettaglio.concat(typeDetail.text() + " | ");  
    }  
}
```

Figura 62. Codice per recupero dettagli aggiuntivi.

La lista di ristoranti presentati da TripAdvisor è suddivisa in pagine, ognuna delle quali contiene al massimo trenta elementi. È necessario quindi scrivere una porzione di codice che permetta il passaggio a quella successiva e così via, fino al recupero di tutti i risultati. Attraverso le variabili *nextPages* e *links* si è potuto simulare il click del mouse su ogni numero di pagina, potendo così salvare la totalità dei ristoranti della città di Milano (Figura 63).



```

nextPages = doc.getElementsByClass("pgLinks");
for (Element nextPage : nextPages)
    links = nextPage.getElementsByClass("sprite-pageNext");
if (links != null)
    for (Element link : links) {
        String temp = link.attr("href");
        temp = temp.substring(0,temp.lastIndexOf("#"));
        html = "http://www.tripadvisor.co.uk".concat(temp);
        break;
    }

```

Figura 63. Codice per passare di pagina in pagina.

Osservando il dominio applicativo del progetto, si è pensato potesse essere utile recuperare le coordinate geografiche di ogni ristorante e/o punto di interesse della città di Milano. A tal proposito si sono invocate le API¹⁰ di Google, facendo attenzione a impostare correttamente l'URL con indirizzo, località e provincia. Ad esempio, per recuperare le coordinate di un ristorante che si trova in Viale Abruzzi 32, Milano (MI) devo impostare: *indirizzo = "viale + abruzzo + 32"*, *localita = "20131"*, *provincia = "milano"* in modo tale da avere *url = "http://maps.googleapis.com/maps/api/geocode/json?address = viale + abruzzo + 32,20131,milano&sensor = true"*. Ovviamente questo lavoro non deve essere fatto manualmente per ogni ristorante, ma attraverso opportune funzioni che estrapolano i dati dal sito Web e li riadattano per essere inseriti nell'*url* (Figura 64).

```

URL url = new URL("http://maps.googleapis.com/maps/api/geocode/json?address="+
    indirizzo + "," + localita + "," + provincia + "&sensor=true");
URLConnection urlConnection = (URLConnection) url.openConnection();
InputStream is = urlConnection.getInputStream();
BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(is, "utf-8"));
StringBuilder responseString = new StringBuilder();
String line = "";
while ((line = bufferedReader.readLine()) != null)
    responseString.append(line);
JSONObject obj = new JSONObject(responseString.toString());
JSONArray results = obj.getJSONArray("results");
status = obj.getString("status");

```

Figura 64. Codice per invocazione API di Google.

¹⁰ Con il termine API (Application Programming Interface) si indica un insieme di procedure disponibili al programmatore per l'espletamento di un determinato compito all'interno di un certo programma. Spesso con tale termine si intendono le librerie software disponibili in un certo linguaggio di programmazione. Le API permettono ai programmatori di evitare la scrittura di codice da zero, entrando nel più vasto concetto di riuso del codice.

Se lo *status* è OK allora vuol dire che l'invocazione è andata a buon fine, e si può procedere con il recupero delle coordinate geografiche e con l'aggiornamento dei valori nel database (Figura 65).


```
"formatted_address" : "Viale Abruzzi, 32, 20131 Milano, Italia",
"geometry" : {
  "location" : {
    "lat" : 45.4779322,
    "lng" : 9.217353899999999
  },
}


if(status.compareTo("OK") == 0) {
  for (int i = 0; i < results.length(); i++) {
    JSONObject result = results.getJSONObject(i);
    JSONObject geo = result.getJSONObject("geometry");
    JSONObject loc = geo.getJSONObject("location");
    Latitudine = loc.getDouble("lat");
    Longitudine = loc.getDouble("lng");
    stmt = con.prepareStatement("UPDATE ristorante SET indirizzo =?," +
      "indirizzoExt =?, localita =?, latitudine =?, longitudine =?" +
      "WHERE paginaWeb =?");
    stmt.setString(1, indirizzoCompleto);
    stmt.setString(2, dettIndirizzoExt);
    stmt.setString(3, CAP);
    stmt.setString(4, Lat);
    stmt.setString(5, Lng);
    stmt.setString(6, indirizzoWeb);
    stmt.executeUpdate();
  }
}
```

Figura 65. Codice per recupero coordinate geografiche.

In accordo con gli obiettivi del progetto, affinché sia possibile suggerire un insieme di contenuti che sia personalizzato e ordinato, è indispensabile salvare nella base di dati l'insieme delle valutazioni associate ad ogni ristorante e/o punto di interesse.

Ogni utente registrato su TripAdvisor dispone di un profilo personale, in cui sono presenti le principali informazioni demografiche e il numero di contributi dati, in termini di recensioni scritte, voti attribuiti e foto inserite (Figura 66, Figura 67 e Figura 68).

 **johnashley** **139 Contributions**
 9 Reviews 126 Ratings 4 Photos Continue Browsing »



Show all cities (115) Send Message

102 Cities **26** Countries

About Me

Since Aug 2003
 Lives in lisbellaw n.ireland
 50-64 year old male

Badges

- 9 reviews (Senior Reviewer)
- 3 hotel reviews
- 5 attraction reviews
- Reviews in 5 cities
- 7 Helpful votes

57,109 total miles travelled

14% of the world travelled

Figura 66. Profilo utente in TripAdvisor.

About Me

Since Aug 2003

Lives in lisbellaw n.ireland

50-64 year old male

etaMinima

etaMassima

città

sezzo



Figura 67. Zoom su profilo utente.

Quando un utente rilascia una valutazione, la lista di recensioni presente sul portale viene aggiornata in modalità LIFO: l'ultima scritta viene posta in cima alla lista.

Ad ogni modo, ciò che è importante recuperare è lo username dell'utente che ha recensito l'item, il rating numerico espresso e infine, se presente, il commento testuale (Figura 69 e Figura 70).



Figura 69. Esempio di recensione in TripAdvisor.

```

Elements divReviewSelectors = docTemp.getElementsByClass("reviewSelector");
if (!divReviewSelectors.isEmpty()) {
    for (Element divReviewSelector : divReviewSelectors) {
        Elements username = divReviewSelector.getElementsByClass("greyMemberNames");
        nickname = username.get(0).text();
        Elements ratings = divReviewSelector.getElementsByClass("sprite-ratings");
        rating = Float.valueOf(ratings.get(0).attr("content"));
        Elements divDescrizioni = divReviewSelector.getElementsByClass("entry");
        Elements descrizione = divDescrizioni.get(0).getElementsByTag("p");
        recensione = descrizione.text();
        stmtTemp = con.prepareStatement("SELECT * FROM voto_rist_utente WHERE " +
            "idRistorante =? AND username =?");
        stmtTemp.setInt(1, idRist);
        stmtTemp.setString(2, nickname);
        rsTemp = stmtTemp.executeQuery();
        if (!rsTemp.first()) {
            stmtTemp = con.prepareStatement("INSERT INTO voto_rist_utente (idRistorante," +
                "username,voto,recensione) VALUES (?,?,,?)");
            stmtTemp.setInt(1, idRist);
            stmtTemp.setString(2, nickname);
            stmtTemp.setFloat(3, rating);
            stmtTemp.setString(4, recensione);
            stmtTemp.executeUpdate();
        }
    }
}

```

Figura 70. Codice per recupero recensioni.

3.4 Lo schema del database

Il database creato presenta una serie di tabelle per contenere le principali informazioni su ristoranti, hotel e punti di interesse della città di Milano. Nonostante il lavoro di scraping effettuato, il progetto di tesi non prenderà in considerazione la parte relativa agli hotel, che quindi non compariranno né sull'applicazione mobile né tra i contenuti mostrati dal grande schermo. Secondo gli autori, infatti, le persone interessate a partecipare all'Expo avranno già preventivamente prenotato il loro pernottamento a Milano.

All'interno di questa sezione si riporta lo schema E/R della base di dati, il quale viene frammentato in diverse immagini per motivi di leggibilità. Prima di passare però alla descrizione delle tabelle, si vuole specificare la convenzione utilizzata e il significato di alcune variabili utili a capire quanto riportato successivamente.

Per quanto riguarda le convenzioni, gli attributi in **grassetto** non possono essere nulli mentre quelli in **grassetto sottolineato** rappresentano le chiavi primarie.

Le variabili da tener in considerazione, invece, sono *entità* che può assumere un valore del seguente insieme {*ristorante, attrazione, attività, vita notturna, shopping, hotel*} e *istanza* che rappresenta uno specifico elemento dell'entità. Ad esempio, se entità vale "ristorante", una relativa istanza può essere "Pasticceria Castelnuovo" o "Ristorante Guerrini".

La tabella **ENTITÀ** presenta il seguente set di attributi:

- ***idEntita'* (PK)** → è un valore numerico che si autoincrementa e serve a identificare univocamente l'istanza.
- *nome*
- *dettaglio* → se presente, fornisce una descrizione più dettagliata dell'istanza. Ad esempio per le attrazioni e le attività si possono trovare informazioni sulla durata della visita o sul prezzo del ticket. I ristoranti, inoltre, prevedono due attributi aggiuntivi *dettaglioCucina* e *dettaglioPrezzo*.
- *telefono*

- **voto** → rappresenta il voto medio attribuito all'istanza.
- **email**
- **indirizzo**
- **indirizzoExt** → se presente, fornisce informazioni aggiuntive sull'ubicazione dell'istanza. I tipici esempi sono "ad angolo con...", "all'ottavo piano di...", "ingresso in via...". Questo campo viene inoltre utilizzato per correggere le eventuali informazioni errate presenti nell'**indirizzo**.
- **localita** → in genere contiene il CAP associato al luogo dove si trova l'istanza, in caso contrario riporta la generica scritta "Milano".
- **latitudine** → questa informazione non si trova su TripAdvisor, ma viene calcolata utilizzando le API di Google come mostrato nella sezione precedente.
- **longitudine** → questa informazione non si trova su TripAdvisor, ma viene calcolata utilizzando le API di Google come mostrato nella sezione precedente.
- **imgPrincipale** → si tratta della foto che appare accanto alle informazioni principali. Viene recuperata per essere inserita nella scheda informativa dell'istanza, sia sull'applicazione mobile sia sul grande schermo.
- **isPresent** → è un campo booleano che vale 1 quando l'istanza è presente sul sito, e 0 in caso contrario. È utile nel momento in cui si vogliono eliminare dal database gli elementi che sono state rimossi da TripAdvisor. Questo campo è presente solo nella tabella Ristorante, in quanto è l'unica soggetta a frequenti rimozioni sul sito Web.
- **indirizzoWeb** → è l'indirizzo della pagina Web in cui si trova la scheda di dettaglio dell'istanza.

La tabella **UTENTE** presenta il seguente set di attributi:

- **username (PK)**
- **citta**
- **Sesso**
- **fasciaEtaMinima**

- *fasciaEtaMassima*

La tabella **VOTO_ENTITÀ_UTENTE** fa da ponte tra la tabella **ENTITÀ** e la tabella **UTENTE**. Un utente può effettuare più di una valutazione, ma una e una sola per ogni istanza dell'entità! Ad esempio, può valutare più ristoranti, l'importante è che valuti una sola volta lo specifico ristorante "Pasticceria Castelnuovo".

Ad ogni modo, la tabella presenta il seguente set di attributi:

- ***idEntita'* (PK)**
- ***username* (PK)**
- ***voto*** → contiene il rating numerico espresso dall'utente circa l'istanza considerata; la valutazione avviene su una scala da 1 a 5.
- ***recensione*** → se presente, contiene la recensione testuale scritta in fase di valutazione.

La tabella **ENTITÀ_IMG** associa un'istanza all'url delle immagini presenti nella sua galleria fotografica. Nel rispetto delle cardinalità inserite nello schema E/R, un'istanza deve presentare almeno un'immagine (1..N), la quale è associata ad una sola istanza (1..1).

Ad ogni modo, la tabella presenta il seguente set di attributi:

- ***imgUrl* (PK)**
- ***idEntita'* (PK)**

3.4.1 Entità "ristorante"

Associate all'entità "ristorante" vi sono altre tabelle che consentono di descrivere il tipo di cucina servita, la fascia di prezzo da pagare per consumare, il quartiere in cui si trova il locale e, infine, i servizi messi a disposizione (Figura 71).

Tipi Cucina **Restaurants travellers recommend**

4,746 of 4,746 shown

Sorted by

Pasticceria Castelnovo

Ranked #1 of 4,746 in Milan
 ●●●●● 588 reviews

"Return trip and new discovery" 04/01/2014
 "UNACCEPTABLE!!!" 03/01/2014

Cuisines: Dessert, Café, Tea Room
 Map | Visitor photos (66)

La Gelateria della Musica

Ranked #2 of 4,746 in Milan
 ●●●●● 1,219 reviews

"Best gelato" 13/01/2014
 "MMMMM" 11/01/2014

Cuisines: Ice cream
 Map | Visitor photos (51)

Ristorante Guerrini

Ranked #3 of 4,746 in Milan
 ●●●●● 287 reviews

"Wonderful restaurant!" 19/01/2014
 "A bit overrated!" 07/12/2013

Price: 20 EUR - 79 EUR
 Cuisines: Italian, Mediterranean, Chowder

Cuisine

All (4,746)
 African (15)
 American (116)
 Asian (95)
 Bakery (49)
 Barbecue (22)
[More cuisines](#)

Price

All (4,746)
 € (56)
 €€ (250)
 €€€ (282)
 €€€€ (323)

Neighbourhoods

All (4,746)
 Centro Storico (550)
 Isola (1)
 Magenta (67)
 Porta Ticinese (94)
 Porta Venezia (78)

Dining options

All (4,746)
 After-hours (1,499)
 Bar Scene (40)
 Breakfast/Brunch (368)
 Buffet (25)
 Business (145)
[More dining options](#)

Fascia Prezzo

Quartieri

Opzioni

Figura 71. TipiCucina, FasciaPrezzo, Quartieri, Opzioni in TripAdvisor.

La tabella **TIPOCUCINA** presenta il seguente set di attributi:

- ***idTipocucina (PK)*** → è un valore numerico che si autoincrementa e serve a identificare univocamente il tipo di cucina.
- ***tipocucina*** → indica la tipologia di cucina servita nel ristorante, ad esempio "Japanese", "Dessert" e così via.

La tabella **RIST_TIPOCUC** fa da ponte tra la tabella **RISTORANTE** e la tabella **TIPOCUCINA**. Nel rispetto delle cardinalità inserite nello schema E/R, un ristorante

può presentare diversi tipi di cucina (0..N), mentre un tipo di cucina deve necessariamente fare riferimento ad almeno un ristorante (1..N).

Si vuole sottolineare che TripAdvisor inserisce nell'elenco dei tipi cucina anche quelli non associati ad alcun ristorante; nel contesto della tesi, queste informazioni sono irrilevanti e pertanto non sono state prese in considerazione.

Ad ogni modo, la tabella presenta il seguente set di attributi:

- **idRistorante (PK)**
- **idTipoCucina (PK)**

La tabella **FASCIAPREZZO** presenta il seguente set di attributi:

- **idFasciaPrezzo (PK)** → è un valore numerico che si autoincrementa e serve a identificare univocamente la fascia di prezzo.
- **fascia** → indica la fascia di prezzo in cui si cataloga il ristorante, ossia "€€€€" (fascia alta), "€€€" (fascia medio-alta), "€€" (fascia media) e "€" (fascia bassa).

La tabella **RISTPREZZO** fa da ponte tra la tabella **RISTORANTE** e la tabella **FASCIAPREZZO**. Nel rispetto delle cardinalità inserite nello schema E/R, un ristorante può appartenere a diverse fasce di prezzo (0..N), mentre una fascia di prezzo deve necessariamente essere associata ad almeno un ristorante (1..N).

Ad ogni modo, la tabella presenta il seguente set di attributi:

- **idRistorante (PK)**
- **idFasciaPrezzo (PK)**

La tabella **QUARTIERE** presenta il seguente set di attributi:

- **idQuartiere (PK)** → è un valore numerico che si autoincrementa e serve a identificare univocamente il quartiere.
- **quartiere** → indica il nome del quartiere nel quale si trova il ristorante, ad esempio "Centro Storico", "Magenta" e così via.

La tabella **RIST_QUAR** fa da ponte tra la tabella **RISTORANTE** e la tabella **QUARTIERE**. Nel rispetto delle cardinalità inserite nello schema E/R, un ristorante può appartenere a più quartieri (0..N), mentre un quartiere deve necessariamente presentare almeno un ristorante (1..N).

Ad ogni modo, la tabella presenta il seguente set di attributi:

- **idRistorante (PK)**
- **idQuartiere (PK)**

La tabella **OPZIONI** presenta il seguente set di attributi:

- **idOpzione (PK)** → è un valore numerico che si autoincrementa e serve a identificare univocamente l'opzione.
- **opzione** → indica il servizio che viene offerto da quel ristorante, ad esempio "Child-friendly", "Romantic" e così via.

La tabella **RIST_OPZ** fa da ponte tra la tabella **RISTORANTE** e la tabella **OPZIONI**. Nel rispetto delle cardinalità inserite nello schema E/R, un ristorante può presentare più opzioni (0..N), mentre un'opzione deve necessariamente fare riferimento ad almeno un ristorante (1..N).

Ad ogni modo, la tabella presenta il seguente set di attributi:

- **idRistorante (PK)**
- **idOpzione (PK)**

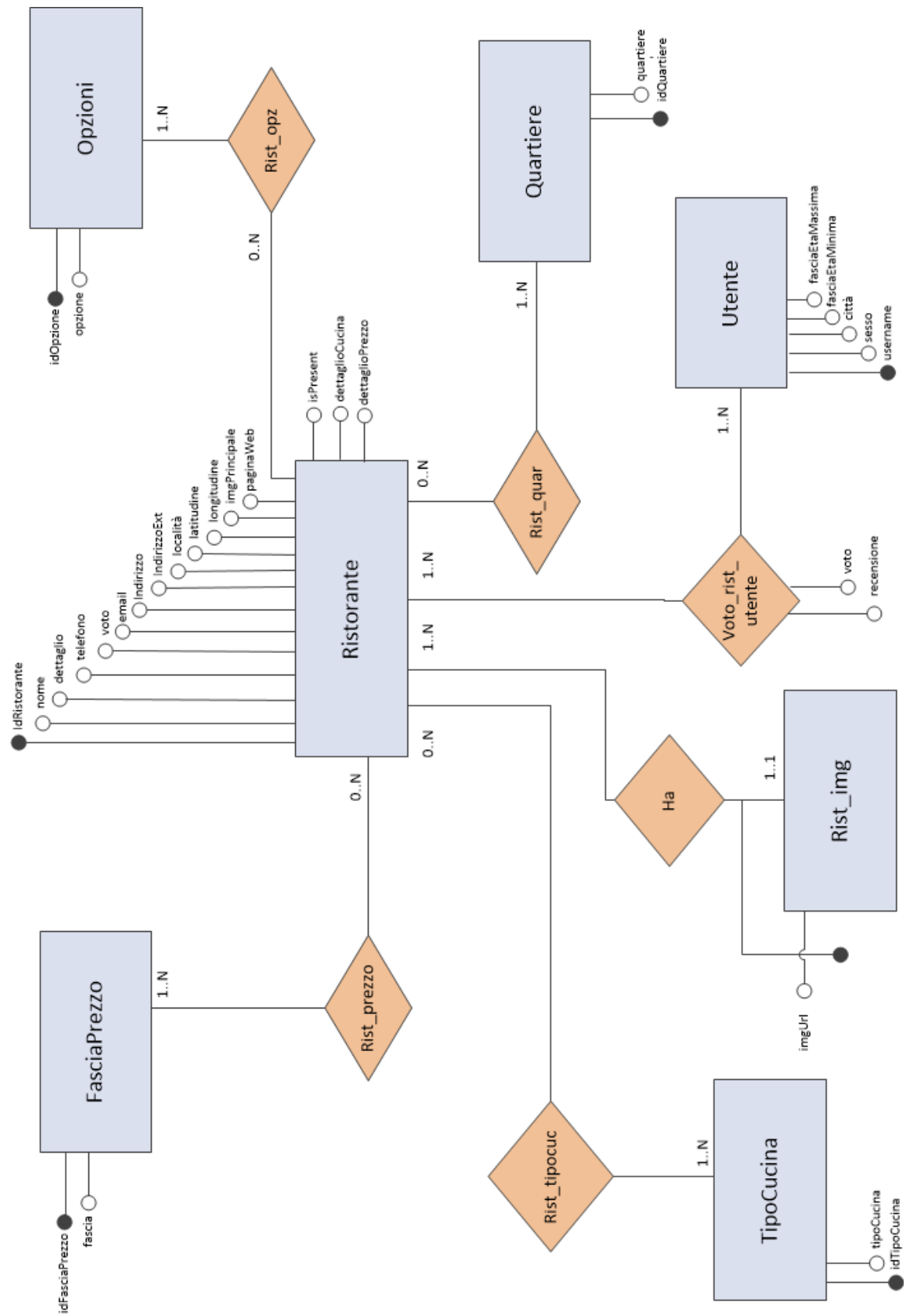


Figura 72. Modello E/R – Ristoranti.

3.4.2 Entità “attrazione” – “attività” – “vita notturna”

Associata all’entità “attrazione”/”attività”/”vitanotturna” vi è un’ulteriore tabella che ne definisce la categoria (Figura 73, Figura 74 e Figura 75).

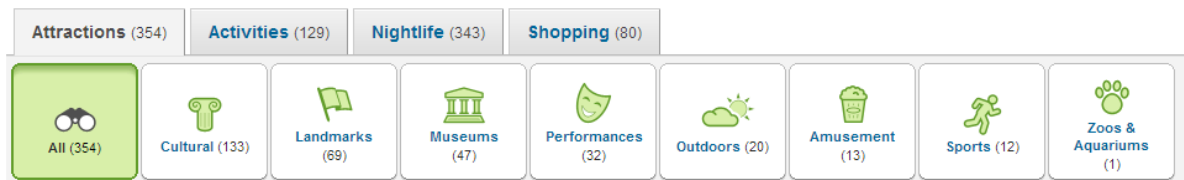


Figura 73. Tipo attrazioni in TripAdvisor.



Figura 74. Tipo attività in TripAdvisor.

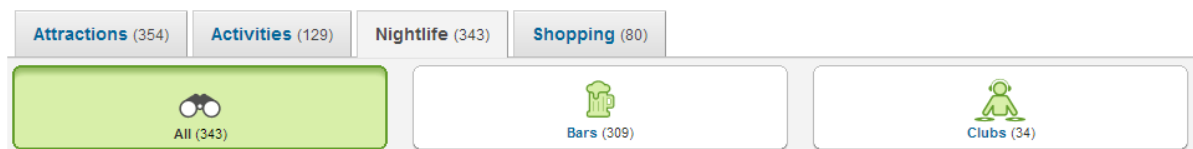


Figura 75. Tipo vita notturna in TripAdvisor.

La tabella **TIPOENTITÀ** presenta il seguente set di attributi:

- **idTipoEntita' (PK)** → è un valore numerico che si autoincrementa e serve a identificare univocamente il tipo di entità.
- **nome** → indica il nome del tipo di entità, che chiameremo categoria per semplicità. Ad esempio nel caso di attrazioni si trova “Cultural”, “Museum”, mentre per le attività si trova “Wellness & Spa”, e così via.
- **paginaWeb** → indirizzo della pagina Web in cui si trova l’elenco delle entità associate a quella categoria.

La tabella **ENTITÀ_TIPO** fa da ponte tra la tabella **ENTITÀ** e la tabella **TIPOENTITÀ**. Nel rispetto delle cardinalità inserite nello schema E/R, un'entità fa riferimento ad un solo tipo (1..1), mentre un tipo deve necessariamente presentare almeno una entità (1..N).

Ad ogni modo, la tabella presenta il seguente set di attributi:

- **idAttrazione (PK)**
- **idTipoAttrazione (PK)**

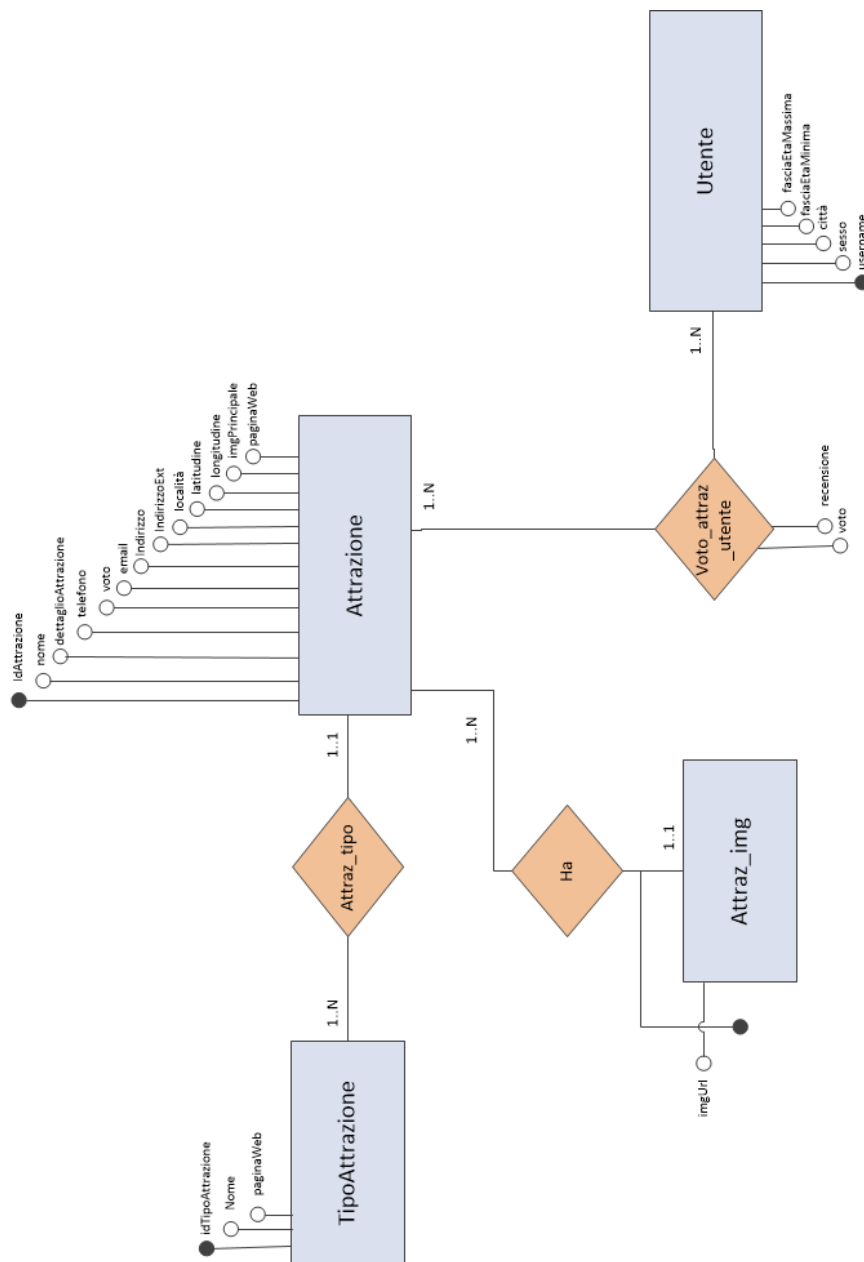


Figura 76. Modello E/R – Attrazioni.

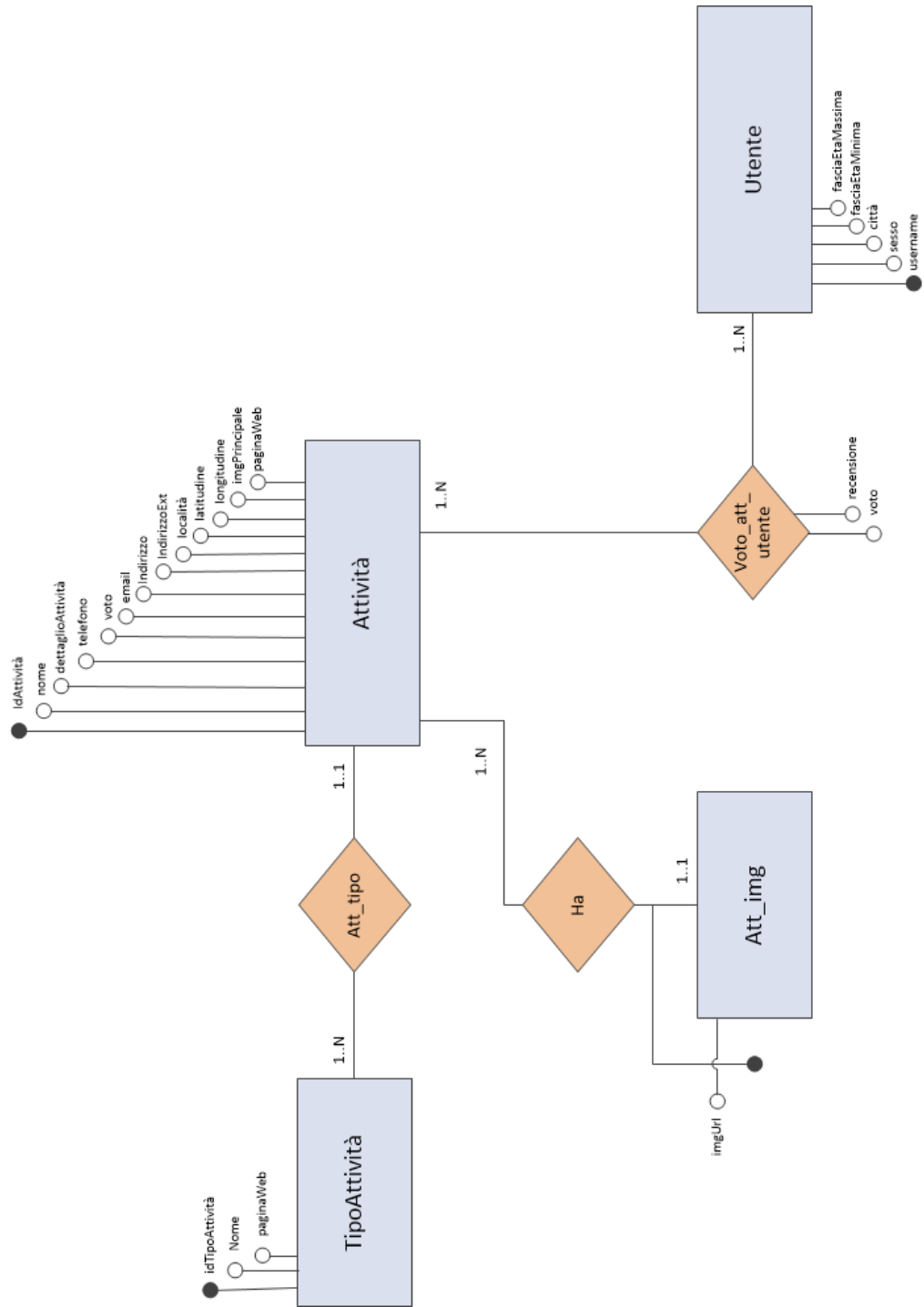


Figura 77. Modello E/R – Attività.

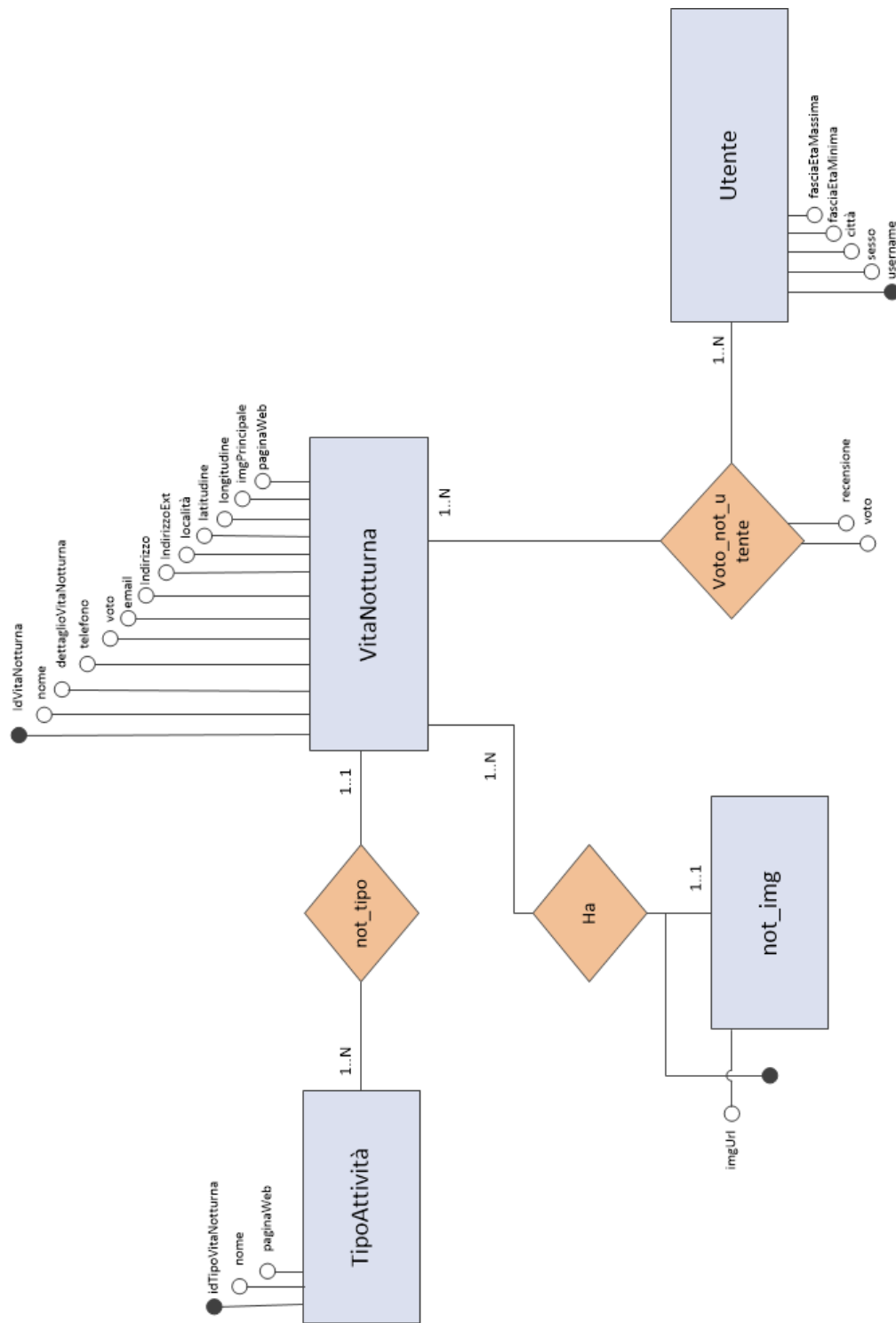


Figura 78. Modello E/R – Vita notturna.

3.4.3 Entità “shopping” – “hotel”

Sul portale di TripAdvisor, l’entità “shopping” non presenta categorie associate. Mentre per l’entità “hotel” si è deciso di non prendere in considerazione ulteriori informazioni, quali categorie e servizi, per non appesantire ulteriormente il database dal momento in cui non verrà utilizzato nel progetto di tesi.

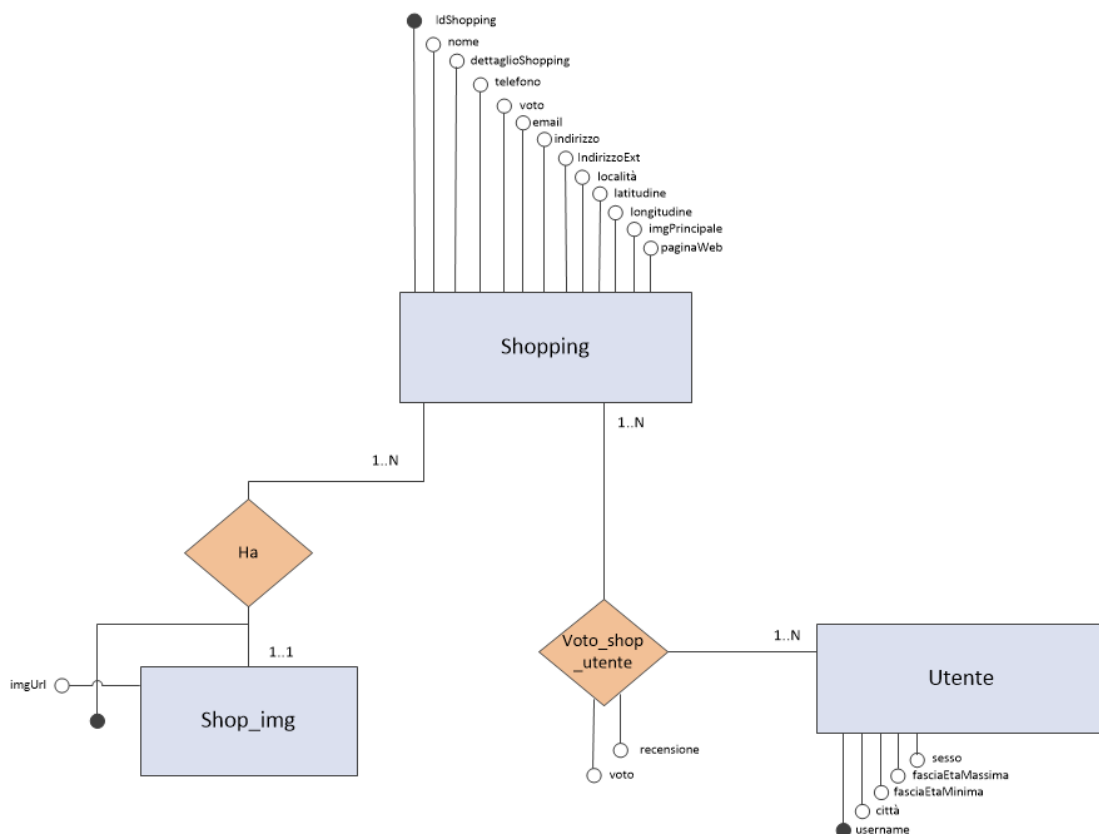


Figura 79. Modello E/R – Shopping.

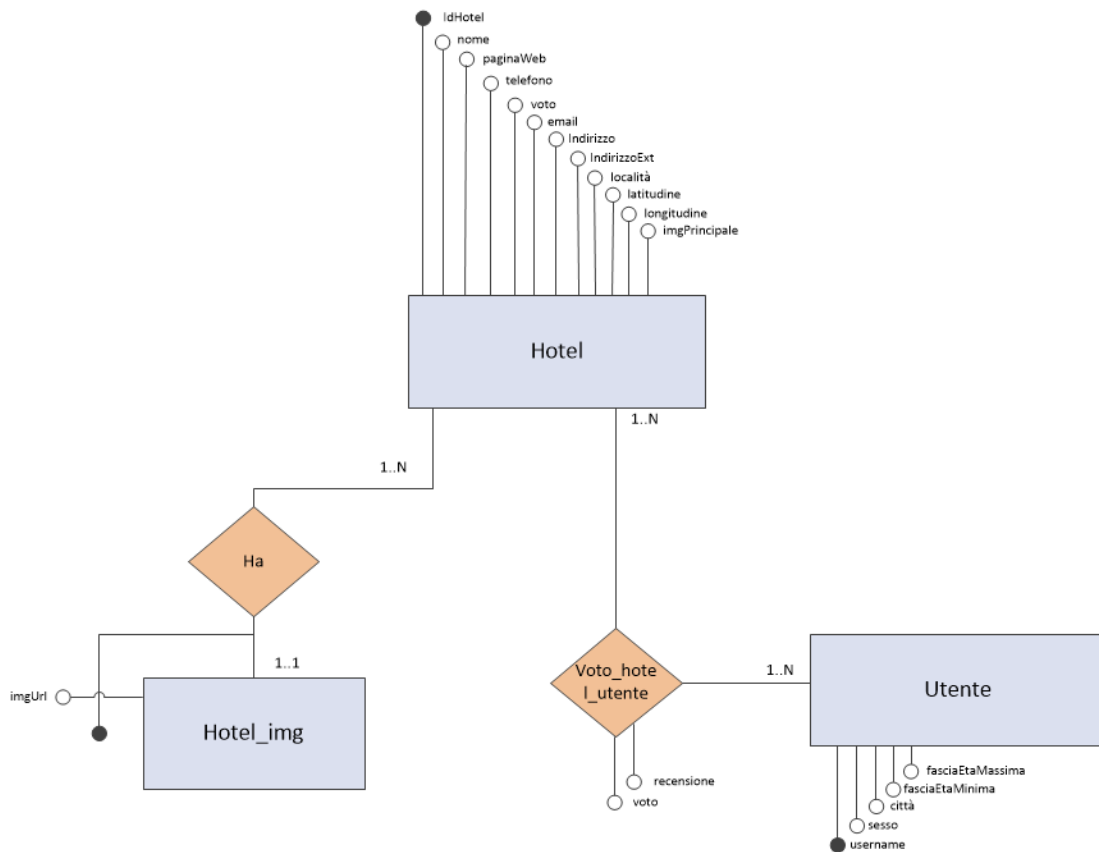


Figura 80. Modello E/R – Hotel.

4. Il criterio di presentazione dei contenuti

Il criterio utilizzato si fonda essenzialmente sul concetto di clustering, una tecnica di analisi dei dati volta alla selezione e al raggruppamento di elementi omogenei. Nel progetto di tesi il clustering ha permesso di raggruppare ristoranti e/o punti di interesse presenti nel database in base alle preferenze espresse dall'utente, alle informazioni anagrafiche (essenzialmente età e sesso) e ai dati raccolti da Facebook (in particolare like e locations). In ogni sottogruppo i risultati ottenuti sono stati ordinati sulla base del Shrinkage Factor, uno stimatore che serve a definire l'importanza di un item rispetto agli altri.

La formula dello Shrinkage Factor è la seguente:

$$\hat{\tau} = \frac{\tau \times \eta + \kappa \times \eta}{\eta + \kappa}$$

dove:

- K , chiamato shrink factor, è un numero compreso tra 3 e 20; nella tesi si sono fatti test con tre valori differenti di K (nello specifico 3, 5 e 10) e sulla base dell'accuratezza dei risultati si è optato per utilizzare il valore 5.
- η rappresenta la popolarità dell'item, equivalente al numero di valutazioni che ha ricevuto da parte degli utenti.
- τ è il rating medio dell'item, calcolato mediante il significato matematico di media.

Dopo questa panoramica introduttiva, ci si addentra nello specifico per stilare il funzionamento dell'algoritmo utilizzato. Per una migliore comprensione dei passaggi, le variabili in gioco vengono colorate in modo differente così che il lettore possa capire immediatamente a che cosa ci si sta riferendo.

Il sistema può essere visto come una black-box che riceve in ingresso una particolare stringa e fornisce in uscita la lista di elementi che maggiormente si adattano ai gusti dell'utente. La stringa in ingresso è formattata secondo $input = età | sesso | listapreferenze | categoriaselezionata$, dove:

- *età* e *sesso* sono le informazioni anagrafiche dell'utente e vengono prelevate dal profilo personale.
- *listapreferenze* viene caricata con l'insieme delle preferenze espresse dalla persona attraverso l'apposita funzionalità presente nell'applicazione. Ogni voce di questa lista è costituita da una coppia $\langle sottocategoria, categoria \rangle$, dove *categoria* può assumere uno dei seguenti valori {*restaurant's cuisine*, *restaurant's option*, *attraction*, *activity*, *nightlife*}. Anticipando quanto esposto nel capitolo seguente, nel momento in cui l'utente decide di inserire le proprie preferenze sceglie innanzitutto la *categoria* e successivamente seleziona una o più *sottocategorie* mostrate dal sistema. A titolo di esempio, se l'individuo volesse far conoscere i suoi gusti in merito al tipo di cucina di un ristorante dovrebbe selezionare la categoria "restaurant's cuisine" e, dall'elenco mostrato dal sistema, optare per "Japanese", "Chinese" e"

Italian". In questo modo la *listapreferenze* sarebbe caricata con *< japanese, cuisine > < chinese, cuisine > < italian, cuisine >*.

- *categoriaselezionata* è la macro area selezionata dal turista quando si trova di fronte al grande schermo e può assumere uno dei seguenti valori {*Restaurant, Attraction, Activity, Shopping, Nightlife, All*}.

A questo punto la stringa *input* viene splittata dal sistema in **tre variabili**, contenenti rispettivamente *età, sesso* e *categoriaselezionata*, e **un array *preferenze*** in cui viene caricata *listapreferenze*. Il motivo di questa particolare procedura risiede nell'impossibilità di passare direttamente un array o una collezione di oggetti al sistema.

Malgrado questo inconveniente bisogna considerare che se un utente ha effettuato l'accesso tramite Facebook, la stringa *input* risulta incompleta mancando le preziose informazioni presenti nel social network. Per recuperare tali dati si sfrutta il token di Facebook relativo al proprietario dello Smartphone.

A questo punto, tramite le Open Graph API messe a disposizione dal social network per gli sviluppatori, si riescono a ottenere le informazioni desiderate; nel progetto di tesi ci si limita a recuperare like e locations. I dati vengono restituiti sotto forma di JSON Array, nel quale ogni elemento è costituito da un campo denominato *category*. Ciclando su questo vettore si va a verificare se il contenuto di ogni campo *category* è presente all'interno del seguente insieme di valori {*Arts/entertainment/nightlife, Attraction/things to do, Book Shop, Outdoor Gear/Sporting Goods, Church/Religious Organisation, Club, Food/grocery, Pub/bar, Restaurant/cafe, Spa/Beauty/personal care, Sports venue, Sports/recreation/activities, Tours/sightseeing, Bar, Food/beverages, Landmark*}.

Se la condizione è verificata viene fatta un'altra richiesta alle API di Facebook per ottenere la *category_list*, ovvero la lista delle sottocategorie che definiscono in maniera più dettagliata la *category* dell'elemento. Affinché queste informazioni possano essere riutilizzate dall'algoritmo, devono essere salvate all'interno del database. Si crea quindi una nuova tabella, chiamata **categoryFB**, nella quale ogni

riga è costituita da tre campi: *id*, *sottocategoria* e *categoria*. Il campo *sottocategoria* conterrà una specifica voce della *category_list*, il campo *categoria* a sua volta presenterà la *category* di riferimento. Una riga della tabella potrebbe essere "1, Sushi, Restaurant/cafe".

A questo punto si può ampliare l'array *preferenze* aggiungendo le informazioni appena salvate nella tabella *categoryFB* del database. Queste ultime infatti costituiscono una sorta di preferenza implicita, in quanto non è stata direttamente inserita dall'utente ma la si può dedurre osservando i like e le locations che ha postato. Chiaramente nel caso in cui la persona non abbia effettuato il login con Facebook, non abbia mai messo un like significativo o non abbia mai postato una location, l'array *preferenze* rimane invariato!

Prima di aggiungere le preferenze implicite è necessario passare attraverso una fase intermedia, in grado di uniformare i nuovi dati a quelli esistenti nel database.

Lo step preliminare dell'algoritmo prevede infatti di scaricare tutti i possibili valori di *category_list* mediante le API di Facebook (link https://graph.facebook.com/search?topic_filter=all&type=placetopic&access_token=<access_token>) e di effettuare un match con il dominio di TripAdvisor. Ad esempio, all'istanza della *category_list* "Sushi Restaurant" viene associata come *sottocategoria* "Japanese" e "Sushi" e come *categoria* "cuisine"; oppure ancora all'istanza della *category_list* "Arts & Entertainment" viene associata come *sottocategoria* "Performances" e "Amusement" e come *categoria* "categorypoi".

A questo punto, potendo disporre del matching effettuato si vanno a prelevare le informazioni riferite all'utente che andranno inserite nell'array *preferenze*; la stringa *input* è completa e può essere pertanto passata al sistema.

Gli scenari che ne derivano seguono strade differenti a seconda del contenuto di *categoriaselezionata*. In particolare, se essa contiene una specifica categoria (scelta tra "Restaurant", "Attraction", "Activity", "Shopping" e "Nightlife") allora la lista presentata conterrà un sottoinsieme di elementi, se invece contiene "All" la lista presentata riguarderà tutti gli item presenti nel database indipendentemente dalla loro categoria.

La modalità di recupero delle informazioni è uguale e viene effettuata eseguendo opportune interrogazioni sulla base di dati. La prima query lanciata va a prendere gli elementi che potrebbero piacere all'utente in base all'array *preferenze*, il quale, come specificato sopra, contiene sia le preferenze esplicite che quelle implicite. Secondariamente si effettua una query per selezionare gli oggetti che sono stati valutati da persone aventi età e sesso dell'interessato (salvate rispettivamente nelle variabili *età* e *sesto*). Infine si genera una query in grado di prelevare gli item relativi alla *categoriaselezionata*; nel caso in cui l'utente abbia scelto "All" verranno presi tutti gli elementi del database senza distinzione di categoria.

Le tre query vengono raggruppate utilizzando una UNION che non genera duplicati, mantiene l'ordine imposto dallo Shrinkage Factor e restituisce i risultati impostando come query più importante la prima e così a scalare.

5. L'infrastruttura completa

Affinché il lettore possa avere una visione d'insieme di quanto esposto finora, si riporta lo schema dell'infrastruttura alla base del progetto di tesi (Figura 81).

All'interno si trovano i seguenti componenti:

- **Facebook**

Interrogato dall'applicazione nella fase di "Log in", restituisce il token che verrà utilizzato per estrapolare likes e locations presenti sul diario Facebook dell'utente.

- **Web Server**

Interrogato dallo Smartphone, permette di conoscere la lista di items che si trovano in prossimità dell'utente (funzionalità di "Around Me" presente nell'applicazione) e di ottenere informazioni, recensioni e immagini relative a un dato ristorante e/o punto di interesse. Su di esso infatti si trova il database centrale contenente i dati estratti da TripAdvisor mediante la tecnica di Web Scraping.

- **NodeJs**

Installato sul calcolatore, permette la creazione del canale WebSocket utilizzato per l'integrazione di piccolo e grande schermo.

- **Lettore NFC**

È collegato al calcolatore e su di esso viene appoggiato lo Smartphone nella fase preliminare all'integrazione di piccolo e grande schermo. A seguito di questo contatto, il calcolatore riceverà le informazioni dell'utente (preferenze, dati registrati/token di Facebook), mentre lo Smartphone riceverà l'indirizzo IP del calcolatore al quale connettersi per iniziare la comunicazione.

- **WebSocket Channel**

È il canale di comunicazione che viene instaurato per permettere la comunicazione tra Smartphone e grande schermo (passando attraverso il calcolatore che elabora le informazioni e restituisce i risultati).

- **Public Display**

È collegato al calcolatore e ha come unico compito quello di mostrare i contenuti che gli vengono passati da quest'ultimo.

- **Local Server**

Installato sull'elaboratore, contiene:

- DB Items, che costituisce la copia replicata del database presente sul Web Server;
- DB Items Display, nel quale vengono salvati gli elementi suggeriti dall'algoritmo;
- Page Generator, che si occupa di generare le pagine Web da mostrare a video prendendo le informazioni dai due database a seconda dell'esigenza.

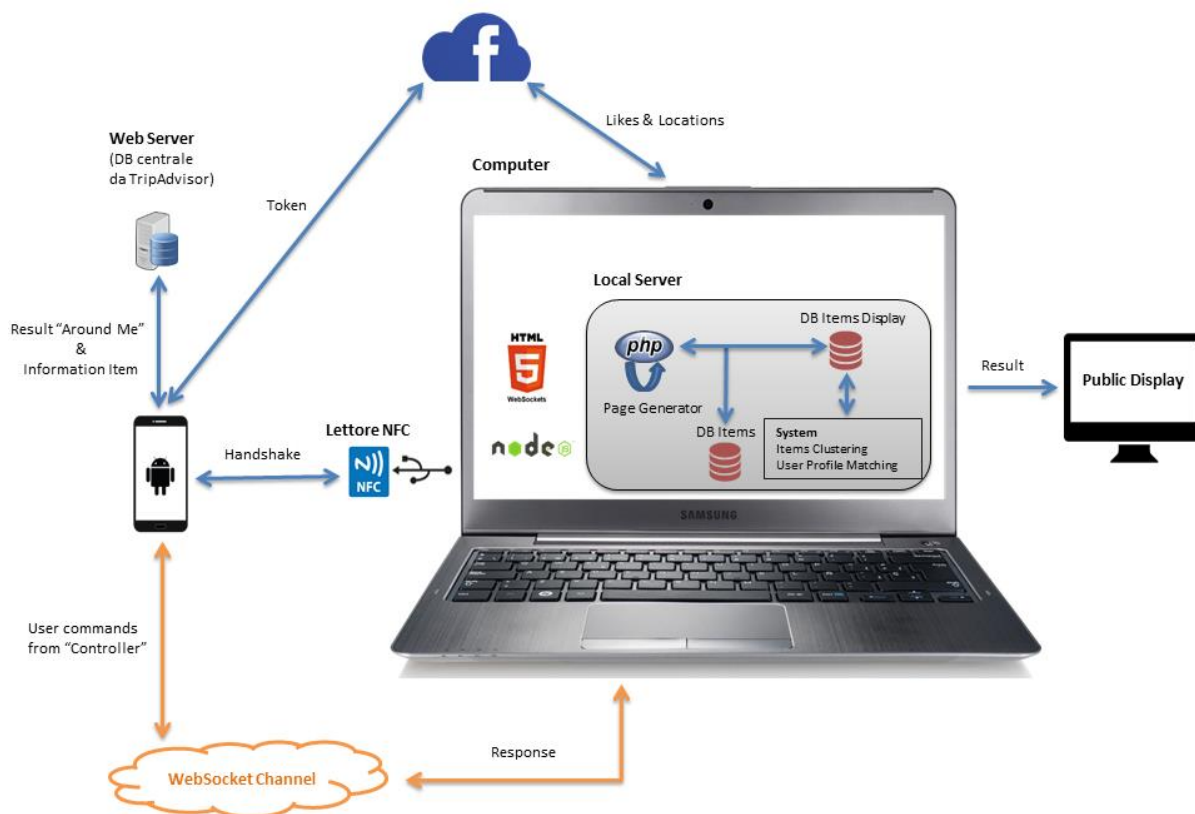


Figura 81. Schema generale dell'infrastruttura.

6. Modellizzazione

La modellizzazione dei casi d'uso è costituita da due proiezioni: statica e dinamica. La prima viene completamente catturata attraverso gli Use Case Diagram, mentre per la seconda si utilizzano i Sequence Diagram e gli Activity Diagram. Prima di mostrare i diagrammi UML generati, si definiscono gli scenari più importanti relativi al progetto di tesi.

6.1 Scenari

In linea generale per scenario si intende una descrizione narrativa dell'esperienza che gli attori fanno all'interno del sistema. Nel progetto di tesi, l'unico attore è il proprietario del device che utilizza l'applicazione mobile.

Nome dello scenario	Registrazione classica + inserimento preferenze
Attori partecipanti	Ambra: Proprietaria di uno Smartphone
Flusso degli eventi	<ol style="list-style-type: none"> 1. Ambra scarica gratuitamente l'applicazione. 2. L'applicazione presenta una schermata in cui è possibile scegliere tra login con Facebook o registrazione classica. 3. Ambra opta per la registrazione classica e compila i campi presenti nella pagina. 4. L'applicazione verifica i dati immessi e registra l'utente. 5. L'applicazione presenta una schermata in cui è possibile inserire le proprie preferenze su ristoranti e/o punti di interesse oppure andare direttamente alla home. 6. Ambra decide di inserire le preferenze. 7. L'applicazione presenta una schermata in cui scegliere la categoria generale. 8. Ambra sceglie una categoria in cui esprimere i propri gusti. 9. L'applicazione propone la lista delle sottocategorie associate a quella selezionata da Ambra. 10. Ambra sceglie al massimo tre sottocategorie tra quelle presentate, e ripete l'operazione per tutte le categorie che intende specificare. 11. L'applicazione presenta una schermata in cui è possibile riordinare le preferenze immesse. 12. Ambra riordina le preferenze. 13. L'applicazione registra le preferenze e mostra la home dell'applicazione.

Tabella 5. Scenario: registrazione + inserimento preferenze.

Nome dello scenario	Accesso con Facebook
Attori partecipanti	Daniele: Proprietario di uno Smartphone
Flusso degli eventi	<ol style="list-style-type: none"> 1. Daniele scarica gratuitamente l'applicazione. 2. L'applicazione presenta una schermata in cui è possibile scegliere tra login con Facebook o registrazione classica. 3. Daniele decide di effettuare l'accesso tramite Facebook e da il consenso per accedere ai dati personali.

	<ol style="list-style-type: none"> 4. L'applicazione procede con il recupero delle credenziali da Facebook e registra l'utente. 5. L'applicazione presenta una schermata in cui è possibile inserire le proprie preferenze su ristoranti e/o punti di interesse oppure andare direttamente alla home. 6. Daniele decide di andare direttamente alla home.
--	--

Tabella 6. Scenario: accesso con Facebook.

Nome dello scenario	Ricerca ristorante + aggiunta ai preferiti
Attori partecipanti	Daniele: Proprietario di uno Smartphone con l'applicazione installata.
Flusso degli eventi	<ol style="list-style-type: none"> 1. Daniele vuole sapere i ristoranti che si trovano vicino a lui, dalla home seleziona "Around Me". 2. L'applicazione elabora la posizione dell'utente e restituisce una lista di risultati. 3. Daniele visualizza i dettagli di un ristorante e lo aggiunge alla lista dei preferiti cliccando sull'icona apposita. 4. L'applicazione aggiunge il ristorante alla lista dei preferiti dell'utente.

Tabella 7. Scenario: ricerca ristorante + aggiunta ai preferiti.

Nome dello scenario	Navigazione sul grande schermo
Attori partecipanti	Ambra: Proprietaria di uno Smartphone con l'applicazione installata.
Flusso degli eventi	<ol style="list-style-type: none"> 1. Ambra decide di interagire con il grande schermo, dalla home seleziona "Controller" e appoggia il device sul lettore NFC per permettere il riconoscimento. 2. Il grande schermo riconosce l'utente e mostra un messaggio di benvenuto. 3. Contemporaneamente, il grande schermo e l'applicazione visualizzano una schermata per la scelta della categoria di contenuti da presentare. 4. Ambra vuole conoscere le attrazioni e seleziona "Attraction" sul device. 5. Contemporaneamente, il grande schermo mostra la lista di attrazioni adatte ad Ambra mentre l'applicazione presenta una serie di bottoni per permettere l'interazione.

	<ol style="list-style-type: none"> 6. Ambra, non trovando ciò che le interessa tra i primi dodici risultati, effettua l'azione "swipe a sinistra" sul device. 7. Il grande schermo presenta altri dodici items che potrebbe interessare all'utente. 8. Ambra vuole vedere i dettagli della seconda attrazione, si posiziona utilizzando i bottoni con le frecce ed effettua l'azione "swipe verso il basso" per entrare nella scheda informativa. 9. Il grande schermo presenta una schermata in cui mostra le informazioni dell'attrazione. 10. Ambra vuole vedere dove si trova sulla mappa e si sposta di schermata in schermata utilizzando i bottoni con le frecce. 11. Il grande schermo presenta la mappa indicando dove si trova l'attrazione. 12. L'applicazione abilita i bottoni per spostarsi sulla mappa. 13. Ambra si allontana dallo schermo. 14. Viene chiusa l'interazione: l'applicazione torna alla home e il grande schermo mostra il filmato iniziale.
--	--

Tabella 8. Scenario: navigazione sul grande schermo.

Nome dello scenario	Cattura item dal grande schermo
Attori partecipanti	Ambra: Proprietaria di uno Smartphone con l'applicazione installata.
Flusso degli eventi	<ol style="list-style-type: none"> 1. Ambra decide di interagire con il grande schermo, dalla home seleziona "Controller" e appoggia il device sul lettore NFC per permettere il riconoscimento. 2. Il grande schermo riconosce l'utente e mostra un messaggio di benvenuto. 3. Contemporaneamente, il grande schermo e l'applicazione visualizzano una schermata per la scelta della categoria di contenuti da presentare. 4. Ambra vuole conoscere i ristoranti e seleziona "Restaurant" sul device. 5. Contemporaneamente, il grande schermo mostra la lista di ristoranti adatti ad Ambra mentre l'applicazione presenta una serie di bottoni per permettere l'interazione. 6. Ambra vuole vedere i dettagli del quinto

	<p>ristorante, si posiziona utilizzando i bottoni con le frecce ed effettua l'azione "swipe verso il basso" per entrare nella scheda informativa.</p> <ol style="list-style-type: none"> 7. Il grande schermo presenta una schermata in cui mostra le informazioni del ristorante. 8. Ambra vuole aggiungere il ristorante tra i preferiti ed esegue l'azione "cattura". 9. L'applicazione notifica l'aggiunta dell'item alla lista dei preferiti. 10. Ambra esce dall'applicazione. 11. Viene chiusa l'interazione e il grande schermo mostra il filmato iniziale.
--	--

Tabella 9. Scenario: cattura item dal grande schermo.

6.2 Use Case Diagram

I casi d'uso rappresentano le principali funzionalità richieste dagli stakeholders; in particolare gli Use Case Diagram definiscono il comportamento del sistema dal punto di vista dell'utente. In seguito verranno presentati i principali casi d'uso derivanti dagli scenari esposti nel Paragrafo 6.1.

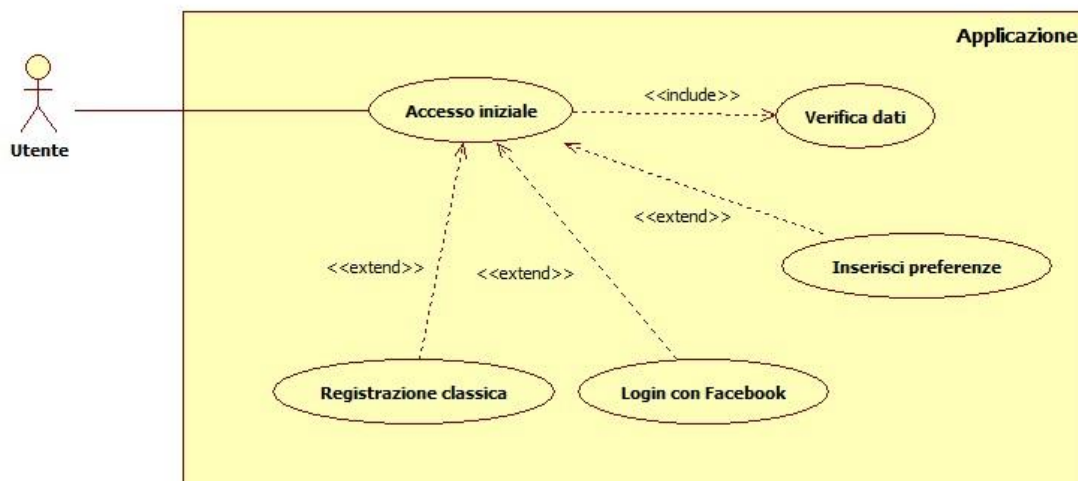


Figura 82. Use Case Diagram: accesso iniziale.

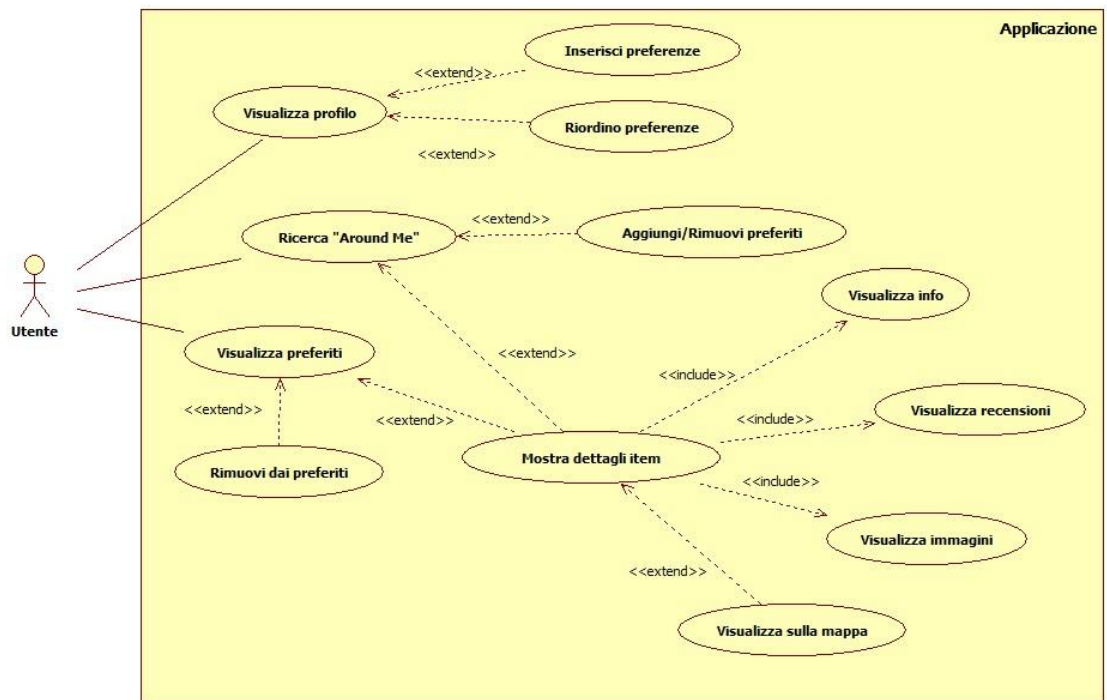


Figura 83. Use Case Diagram: funzionalità principali.

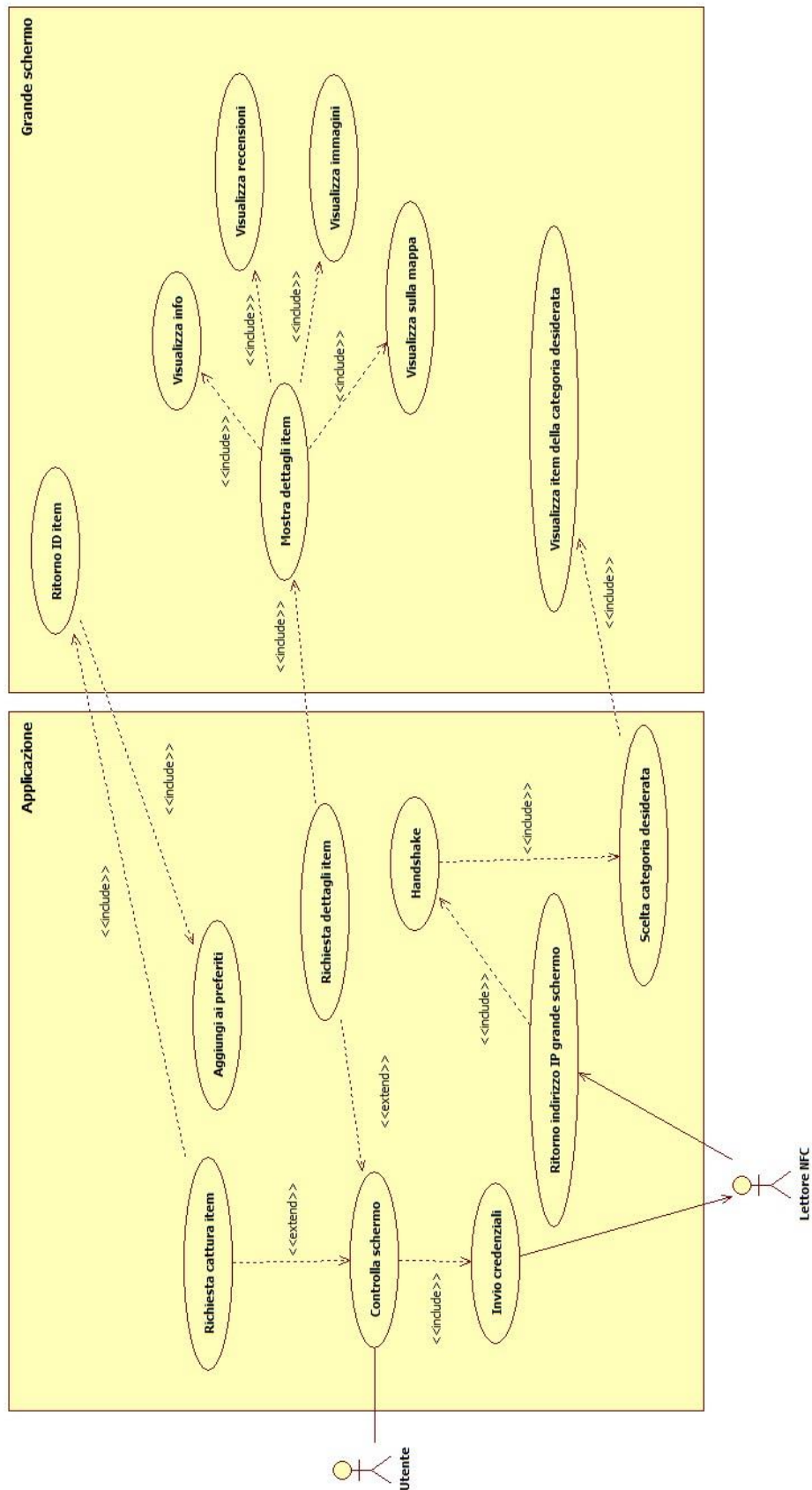


Figura 84. Use Case Diagram: interazione grande schermo.

Nome del caso d'uso	Registrazione classica + inserimento preferenze
Attori partecipanti	Utente
Condizione di ingresso	1. L'Utente avvia l'applicazione.
Flusso degli eventi	<p>2. L'applicazione presenta una schermata in cui è possibile scegliere tra login con Facebook o registrazione classica.</p> <p>3. L'Utente opta per la registrazione classica e compila i campi presenti nella pagina.</p> <p>4. L'applicazione presenta una schermata in cui è possibile scegliere se inserire le proprie preferenze su ristoranti e/o punti di interesse oppure andare direttamente alla home.</p> <p>5. L'Utente decide di inserire le preferenze.</p>
Condizione d'uscita	6. L'applicazione registra l'utente, inserendo i dati anagrafici e le preferenze espresse.
Eccezioni	<ul style="list-style-type: none"> • Nel caso in cui, al punto 3, l'applicazione vede che l'Utente ha meno di 15 anni lo comunica e il caso d'uso rimane nel punto 3. • Nel caso in cui, al punto 3, l'applicazione vede che l'Utente non ha compilato tutti i campi obbligatori lo comunica e il caso d'uso rimane al punto 3.

Tabella 10. Use Case: registrazione + inserimento preferenze.

Nome del caso d'uso	Accesso con Facebook
Attori partecipanti	Utente
Condizione di ingresso	1. L'Utente avvia l'applicazione.
Flusso degli eventi	<p>2. L'applicazione presenta una schermata in cui è possibile scegliere tra login con Facebook o registrazione classica.</p> <p>3. L'Utente decide di effettuare l'accesso tramite Facebook e dà il consenso all'applicazione di accedere ai dati personali.</p>
Condizione d'uscita	4. L'applicazione procede con il recupero delle credenziali.
Eccezioni	<ul style="list-style-type: none"> • Nel caso in cui, al punto 3, l'applicazione verifica problematiche legate al login con Facebook, lo notifica all'Utente e il caso d'uso riprende dal punto 2.

Tabella 11. Use Case: accesso con Facebook.

Nome del caso d'uso	Ricerca ristorante + aggiunta ai preferiti
Attori partecipanti	Utente
Condizione di ingresso	1. L'Utente avvia l'applicazione.
Flusso degli eventi	2. L'applicazione presenta la home. 3. L'Utente vuole sapere i ristoranti che si trovano vicino a lui e seleziona "Around Me". 4. L'applicazione elabora la posizione dell'Utente e restituisce una lista di risultati. 5. L'Utente visualizza i dettagli di un ristorante e lo aggiunge alla lista dei preferiti cliccando sull'apposita icona.
Condizione d'uscita	6. L'applicazione aggiunge il ristorante alla lista dei preferiti dell'Utente.
Eccezioni	<ul style="list-style-type: none"> • Nel caso in cui, al punto 4, l'applicazione verifica problematiche legate al recupero della posizione, lo notifica all'Utente e il caso d'uso riprende dal punto 3.

Tabella 12. Use Case: ricerca ristorante + aggiunta preferiti.

Nome del caso d'uso	Navigazione sul grande schermo
Attori partecipanti	Utente
Condizione di ingresso	1. L'Utente avvia l'applicazione.
Flusso degli eventi	2. L'applicazione presenta la home. 3. L'Utente decide di interagire con il grande schermo, dalla home seleziona "Controller" e appoggia il device sul lettore NFC per permettere il riconoscimento. 4. Il grande schermo riconosce l'utente e mostra un messaggio di benvenuto. 5. Contemporaneamente, il grande schermo e l'applicazione visualizzano una schermata per la scelta della categoria di contenuti da presentare. 6. L'Utente vuole conoscere le attrazioni e seleziona "Attraction" sul device. 7. Contemporaneamente, il grande schermo mostra la lista di attrazioni adatte all'Utente mentre l'applicazione presenta una serie di bottoni per permettere l'interazione. 8. L'Utente non trovando ciò che gli interessa tra i primi dodici risultati, effettua l'azione "swipe a sinistra" sul device.

	<p>9. Il grande schermo presenta altri dodici items che potrebbe interessare all'Utente.</p> <p>10. L'Utente vuole vedere i dettagli della seconda attrazione, si posiziona utilizzando i bottoni con le frecce ed effettua l'azione "swipe verso il basso" per entrare nella scheda informativa.</p> <p>11. Il grande schermo presenta una schermata in cui mostra le informazioni dell'attrazione.</p> <p>12. L'Utente vuole vedere dove si trova sulla mappa e si sposta di schermata in schermata utilizzando i bottoni con le frecce.</p> <p>13. Il grande schermo presenta la mappa indicando dove si trova l'attrazione.</p> <p>14. L'applicazione abilita i bottoni per spostarsi sulla mappa.</p>
Condizione d'uscita	15. L'Utente si allontana dallo schermo e viene chiusa l'interazione.
Eccezioni	<ul style="list-style-type: none"> • Nel caso in cui, al punto 3, l'applicazione e il grande schermo verificano problematiche legate al riconoscimento tramite NFC, lo notificano all'Utente e il caso d'uso riprende dal punto 2. • Nel caso in cui, al punto 4, l'applicazione e il grande schermo verificano problemi legati alla connessione, lo notificano all'Utente e il caso d'uso riprende dal punto 2.

Tabella 13. Use Case: navigazione sul grande schermo.

Nome del caso d'uso	Cattura dal grande schermo
Attori partecipanti	Utente
Condizione di ingresso	1. L'Utente avvia l'applicazione.
Flusso degli eventi	<p>2. L'applicazione presenta la home.</p> <p>3. L'Utente decide di interagire con il grande schermo, dalla home seleziona "Controller" e appoggia il device sul lettore NFC per permettere il riconoscimento.</p> <p>4. Il grande schermo riconosce l'utente e mostra un messaggio di benvenuto.</p> <p>5. Contemporaneamente, il grande schermo e l'applicazione visualizzano una schermata per la scelta della categoria di contenuti da presentare.</p> <p>6. L'Utente vuole conoscere i ristoranti e seleziona "Restaurant" sul device.</p>

	<p>7. Contemporaneamente, il grande schermo mostra la lista di ristoranti adatti all'Utente mentre l'applicazione presenta una serie di bottoni per permettere l'interazione.</p> <p>8. L'Utente vuole vedere i dettagli del quinto ristorante, si posiziona utilizzando i bottoni con le frecce ed effettua l'azione "swipe verso il basso" per entrare nella scheda informativa.</p> <p>9. Il grande schermo presenta una schermata in cui mostra le informazioni del ristorante.</p> <p>10. L'Utente vuole aggiungere il ristorante tra i preferiti ed esegue l'azione "cattura".</p> <p>11. L'applicazione notifica l'aggiunta dell'item alla lista dei preferiti.</p>
Condizione d'uscita	12. L'Utente esce dall'applicazione e viene chiusa l'interazione.
Eccezioni	<ul style="list-style-type: none"> • Nel caso in cui, al punto 3, L'applicazione e il grande schermo verificano problematiche legate al riconoscimento tramite NFC, lo notificano all'Utente e il caso d'uso riprende dal punto 2. • Nel caso in cui, al punto 4, l'applicazione e il grande schermo verificano problemi legati alla connessione, lo notificano all'Utente e il caso d'uso riprende dal punto 2.

Tabella 14. Use Case: cattura item dal grande schermo.

6.3 Sequence Diagram

Il Sequence Diagram è un diagramma previsto dall'UML atto a descrivere uno specifico scenario; il suo compito è quello di enfatizzare le azioni che si svolgono tra le diverse entità coinvolte.

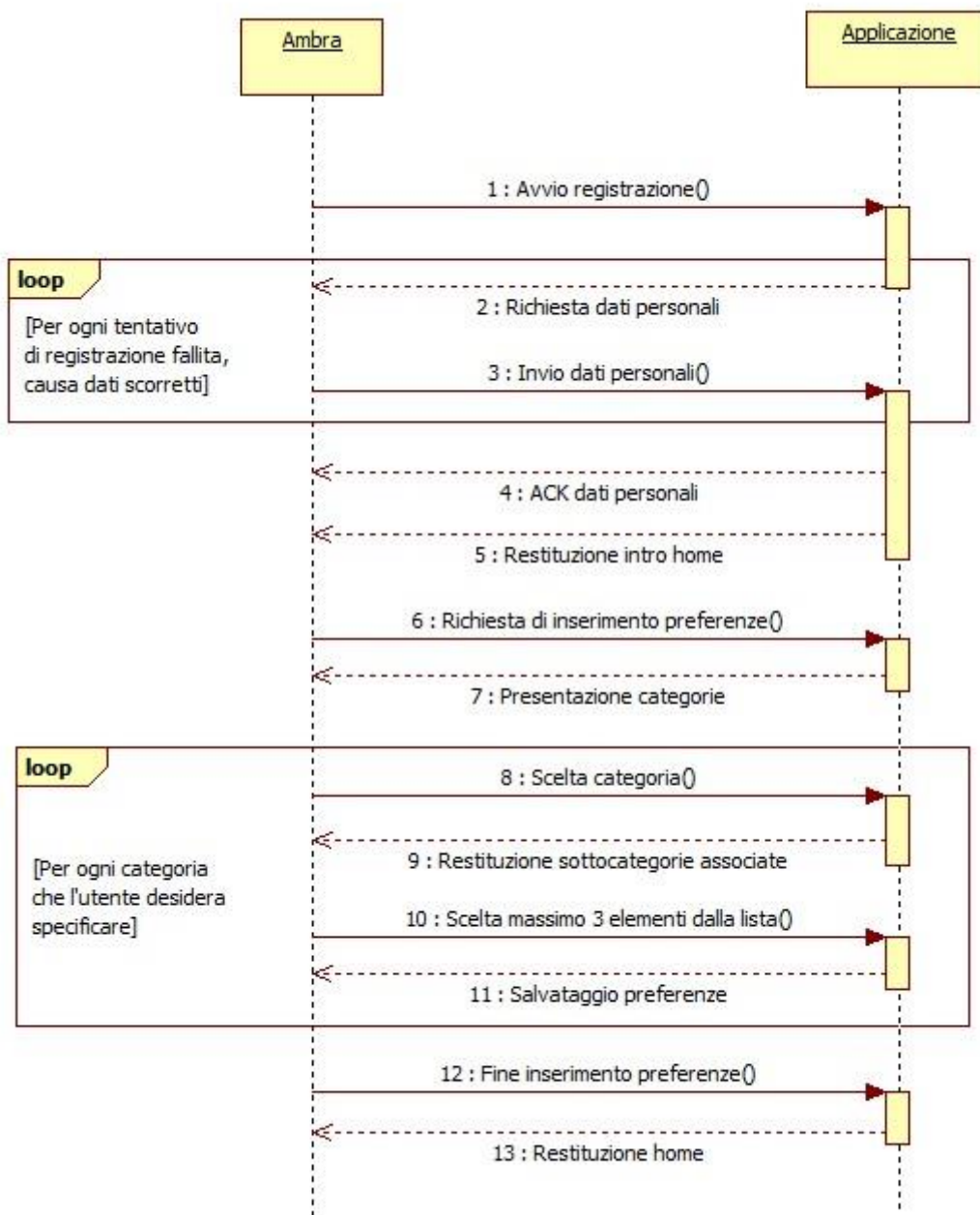


Figura 85. Sequence Diagram: registrazione + inserimento preferenze.

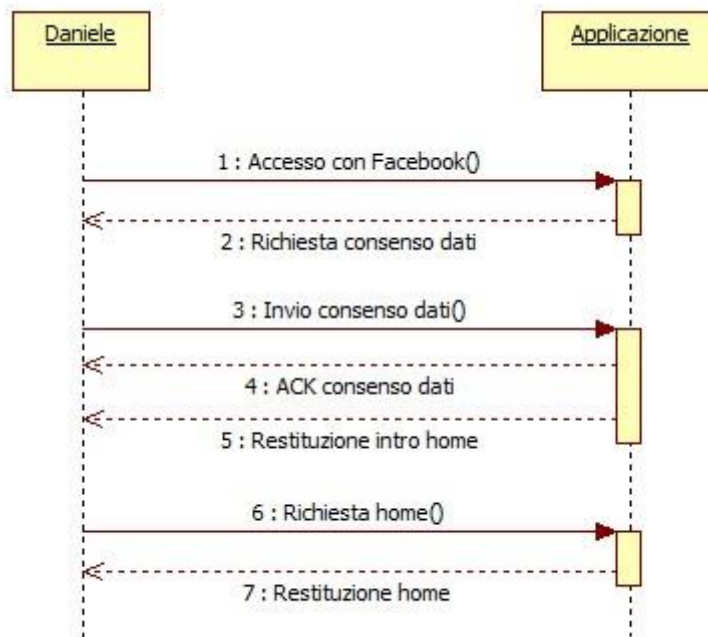


Figura 86. Sequence Diagram: accesso con Facebook

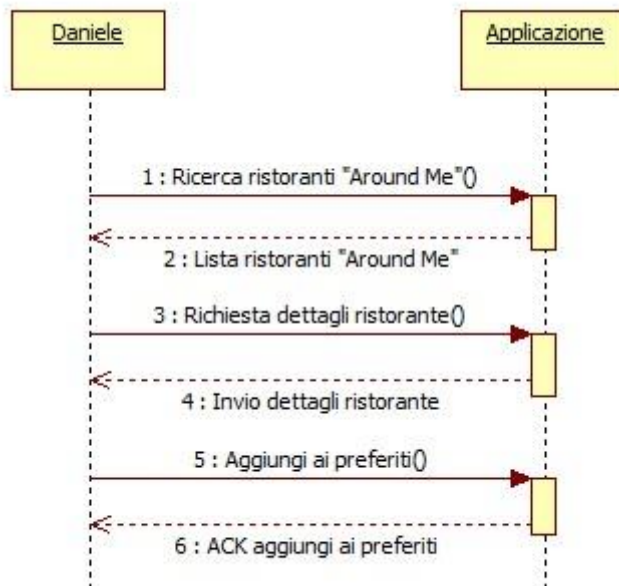


Figura 87. Sequence Diagram: ricerca ristorante + aggiunta preferiti.

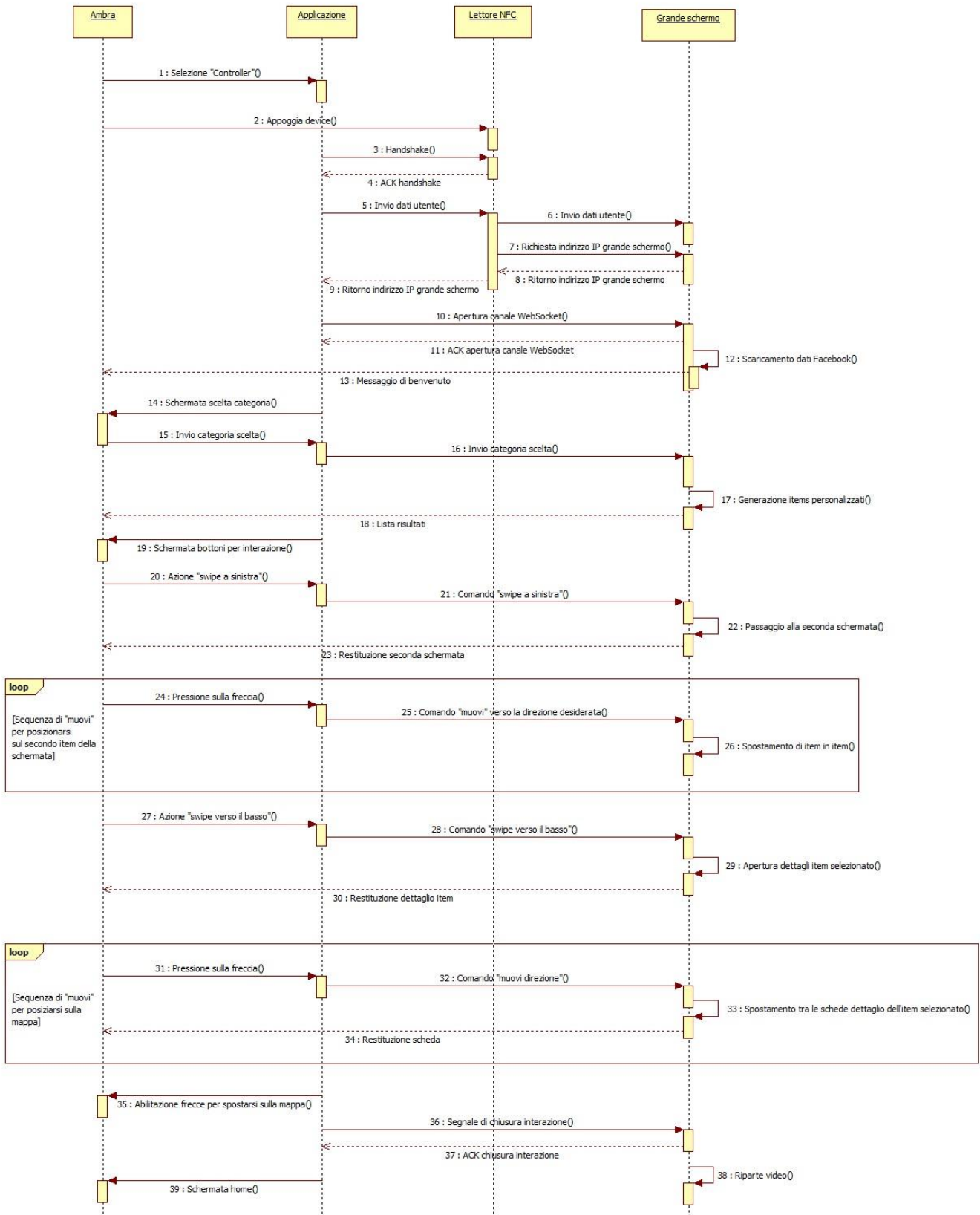


Figura 88. Sequence Diagram: navigazione sul grande schermo.

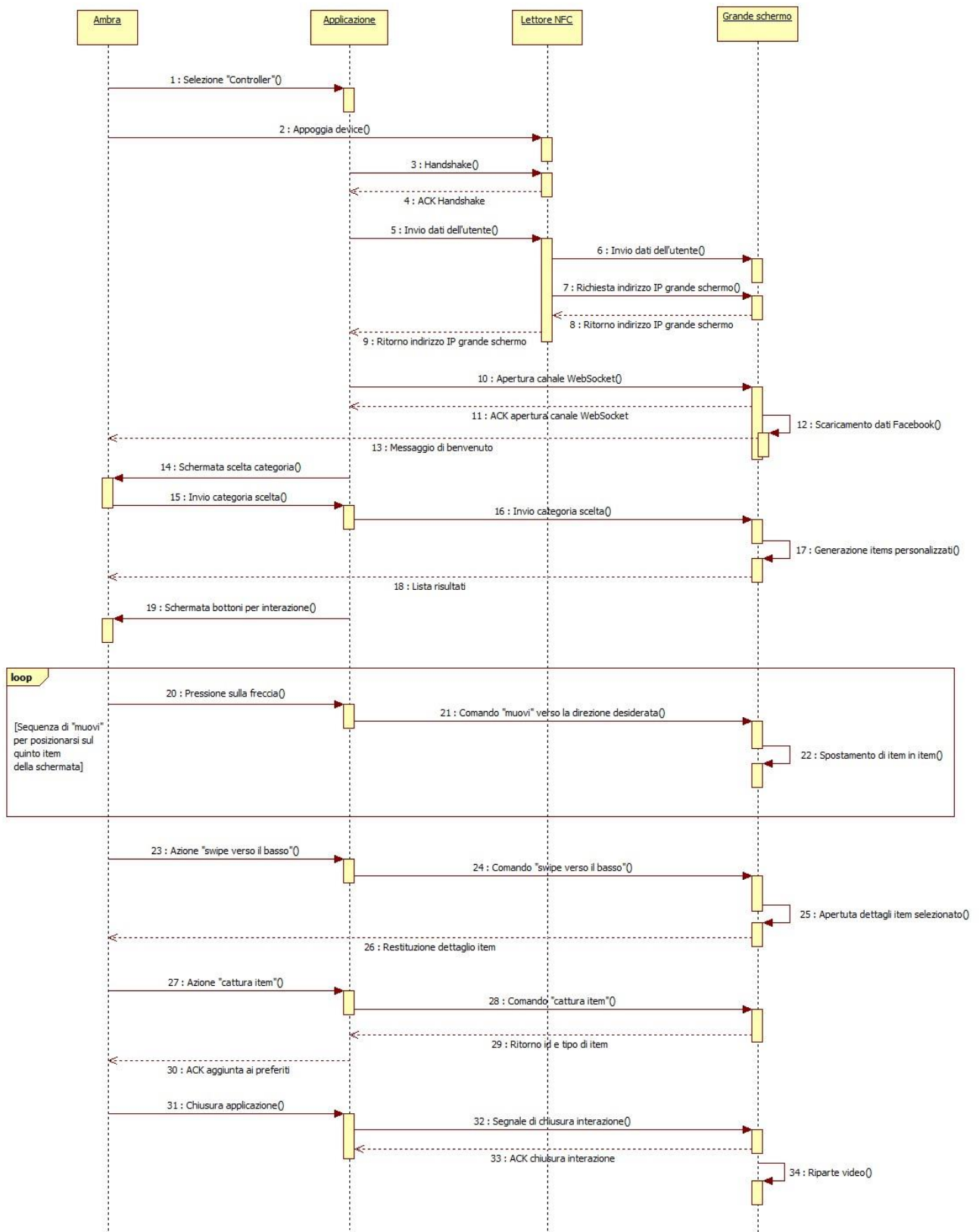


Figura 89. Sequence Diagram: cattura item dal grande schermo.

6.4 Activity Diagram

L'Activity Diagram è un diagramma utilizzato per descrivere una sequenza di operazioni il cui svolgimento permette la definizione di un'attività più complessa. È spesso usato come modello complementare del Sequence Diagram per descrivere il comportamento dinamico del sistema.

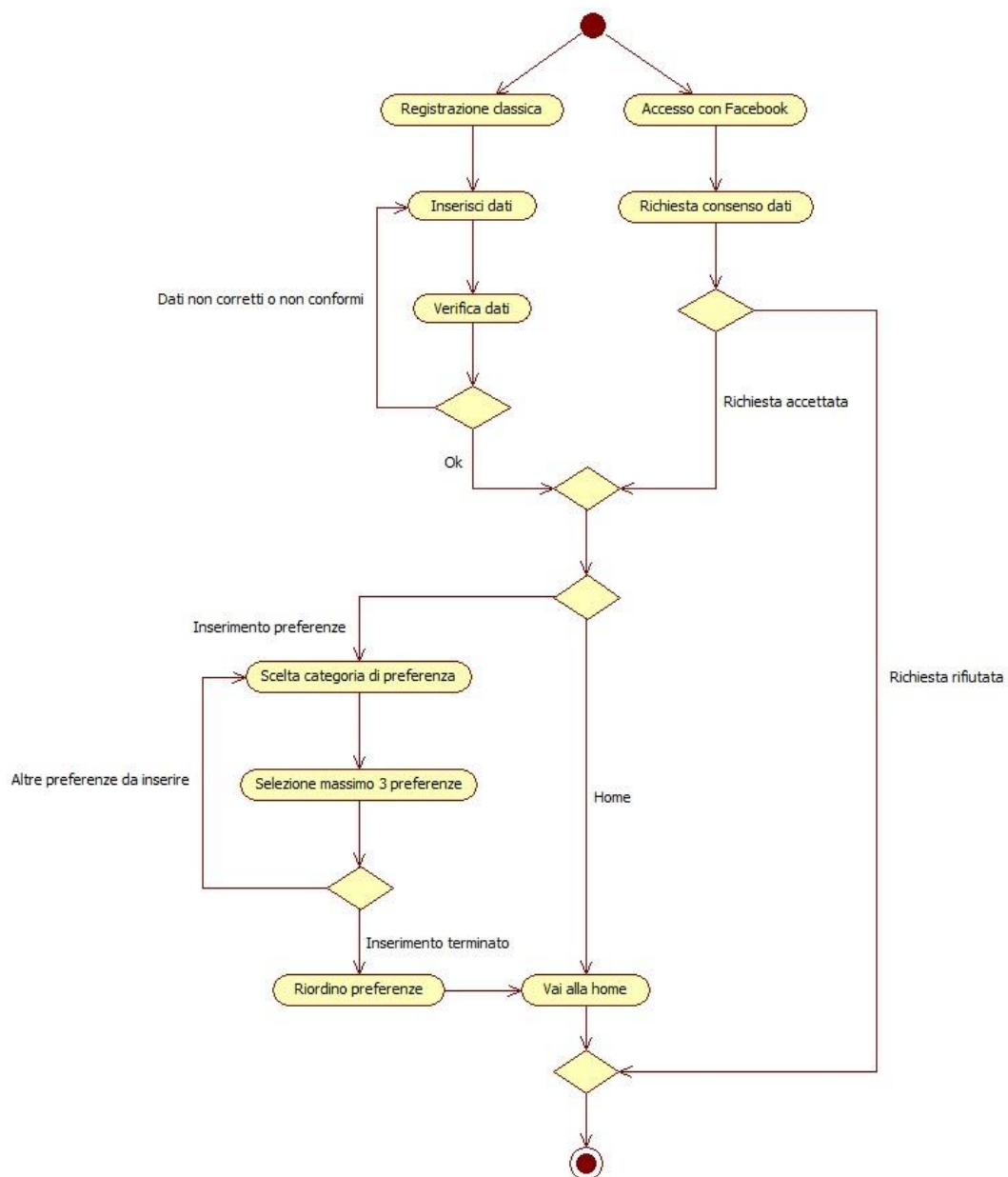


Figura 90. Activity Diagram: registrazione.

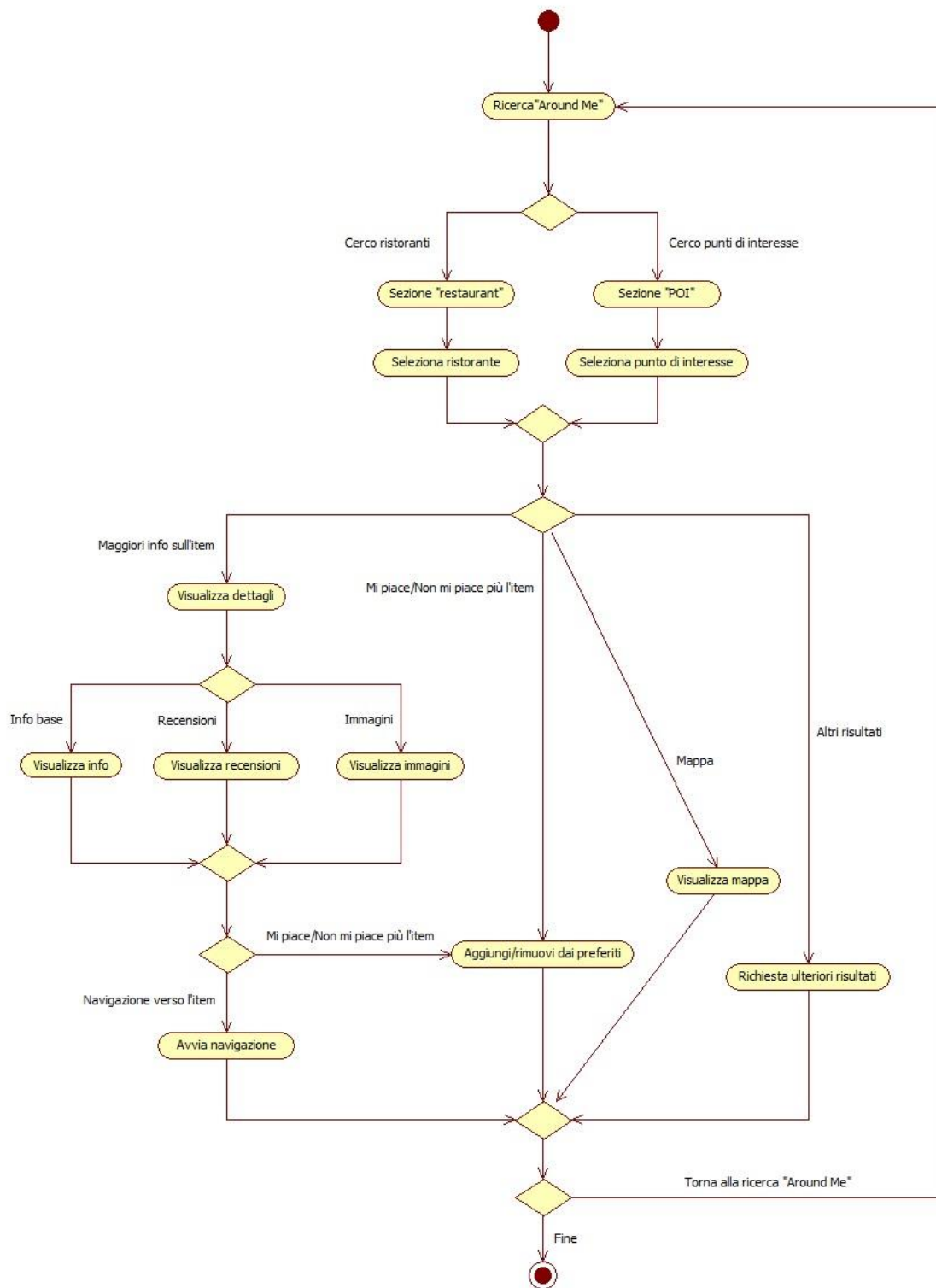


Figura 91. Activity Diagram: ricerca "Around Me".

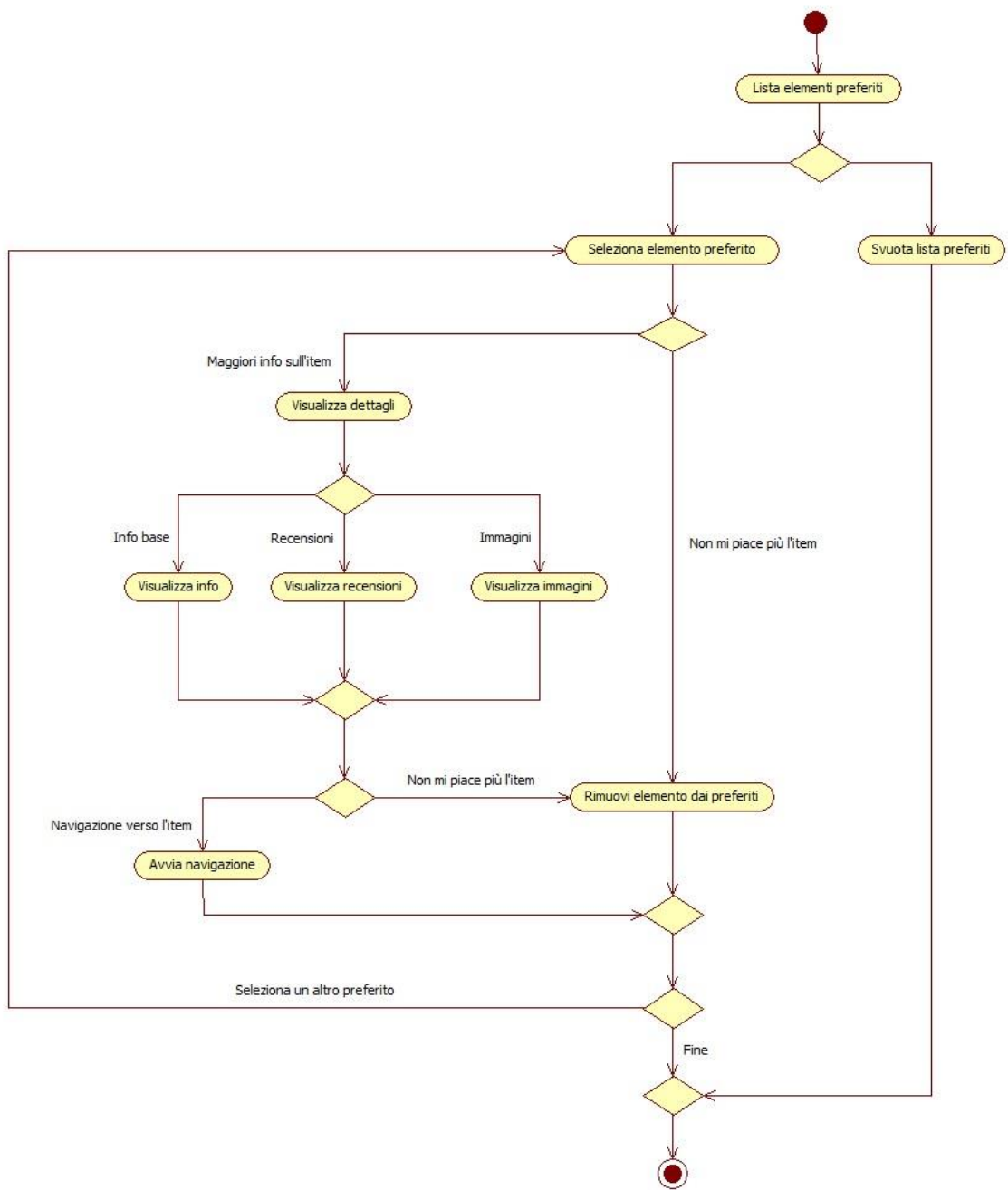


Figura 92. Activity Diagram: gestione preferiti.

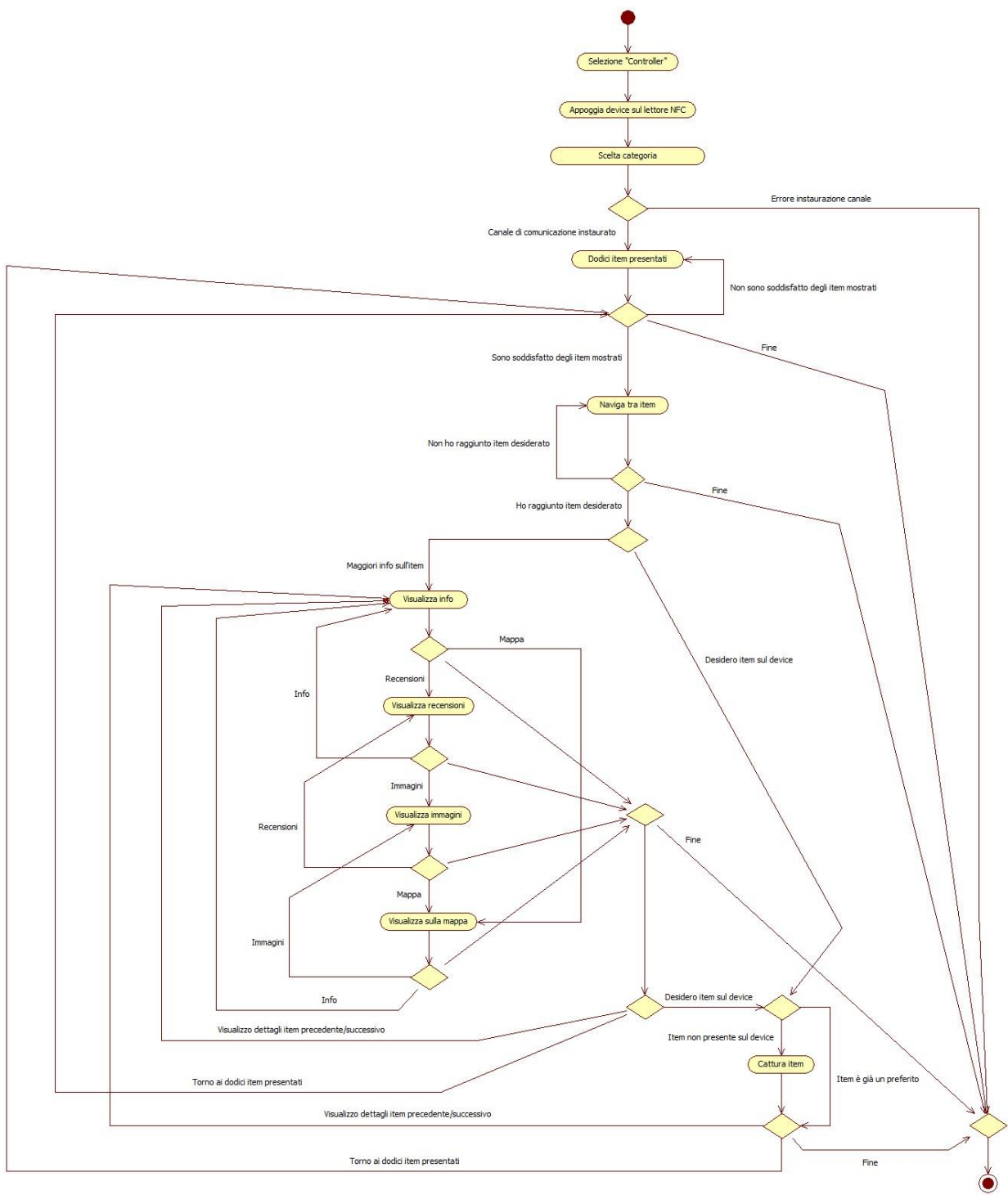


Figura 93. Activity Diagram: interazione grande schermo.

PROTOTIPO REALIZZATO

Dopo aver esposto il contesto applicativo, i requisiti tecnici e funzionali richiesti al prototipo e mostrato l'architettura alla base del progetto, si vuole presentare il "risultato finale" del lavoro di tesi. La prima parte del capitolo evidenzia le funzionalità messe a disposizione dall'applicazione mobile, mentre la seconda parte si concentra sull'integrazione di device personale e grande schermo.

1. L'applicazione

"Nutrire il pianeta, Energia per la vita" è il tema scelto per l'Esposizione Universale di Milano che si terrà nel 2015. Il focus principale della fiera è affrontare il problema della nutrizione dell'Uomo, nel rispetto della Terra sulla quale vive e dalla quale attinge le sue risorse vitali ma esauribili. Alimentazione, sostenibilità, ricerca e sviluppo sono i punti chiave su cui si concentra l'Expo. L'obiettivo è trovare il modo di garantire cibo e acqua a tutta la popolazione mondiale, sconfiggendo la fame, la sete, la mortalità infantile e la malnutrizione che colpiscono ancora oggi milioni e milioni di persone sul Pianeta.

Nel disegnare l'icona dell'applicazione gli autori di questa tesi hanno voluto tener conto del delicato tema alla base dell'esposizione. In particolare ci si è focalizzati sul

fatto che l'uomo, prima di pensare all'innovazione e alla tecnologia, dovrebbe imparare a conoscere e rispettare la natura. Solo così potrà sfruttare le risorse del Pianeta come "trampolino di lancio" per lo sviluppo futuro seguendo la scia lanciata dall'Expo. L'icona raffigura la Terra immersa nella semplicità della natura. I colori utilizzati non sono affatto casuali: l'azzurro con cui si è disegnato il globo richiama l'acqua, elemento fondamentale per la sostenibilità dell'ecosistema; il verde con cui si sono colorate le foglie è da sempre il colore della speranza, la speranza di realizzare ciò che si prefigge l'Expo, e infine il rosso in rappresentanza dell'amore che tutti gli uomini dovrebbe provare verso il proprio Pianeta (Figura 94).



Figura 94. Icona dell'applicazione.

1.1. La registrazione

Per iniziare ad usare l'applicazione è necessario registrarsi. In questa fase non si richiede all'utente di creare una coppia (*username, password*) con la quale effettuare i futuri accessi. La ragione di questa scelta ricade sul fatto che il periodo di vita dell'applicazione è limitato ai giorni di permanenza del turista presso la città di Milano. L'utente quindi, una volta registrato, rimane automaticamente loggato fino a quando non disinstalla l'applicazione dal device personale.

La registrazione può essere vista come una fase preliminare dove ha inizio la conoscenza dell'utente. E nel lavoro di tesi, sono state previste due modalità:

effettuare il login tramite Facebook oppure seguire la procedura standard di registrazione (Figura 95).

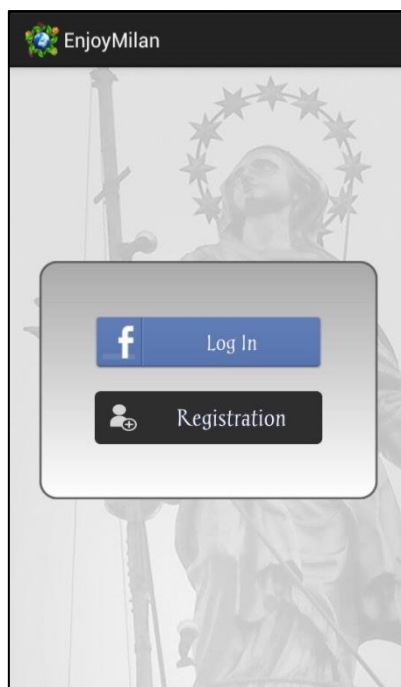


Figura 95. Pagina iniziale.

La scelta di loggarsi con il social network presuppone il consenso al trattamento dei dati personali; in caso contrario non sarà possibile utilizzare l'applicazione con le credenziali di Facebook (Figura 97).

Il recupero delle informazioni presenti sul social network ha come unico scopo quello di collezionare un profilo completo della persona, dai dati anagrafici ai gusti personali. In questo modo il grande schermo può suggerire una lista di ristoranti e/o punti di interesse che rispecchi i suoi interessi.

Optando invece per la registrazione classica, viene mostrata una schermata in cui inserire nome, cognome, data di nascita, sesso e città di residenza; tutti i campi sono obbligatori e vanno pertanto compilati per completare la procedura (Figura 96).

La registrazione richiede un'età minima di 15 anni; la ragione di questa scelta sta nello scoraggiare i ragazzini all'uso dell'applicazione, soprattutto nella parte di interazione con il grande schermo che non deve essere vista come un gioco!

Terminata la procedura di registrazione, l'utente può scegliere di andare direttamente alla home oppure compilare un questionario per esprimere le proprie preferenze in merito a ristoranti e/o punti di interesse (Figura 98 e Figura 99).

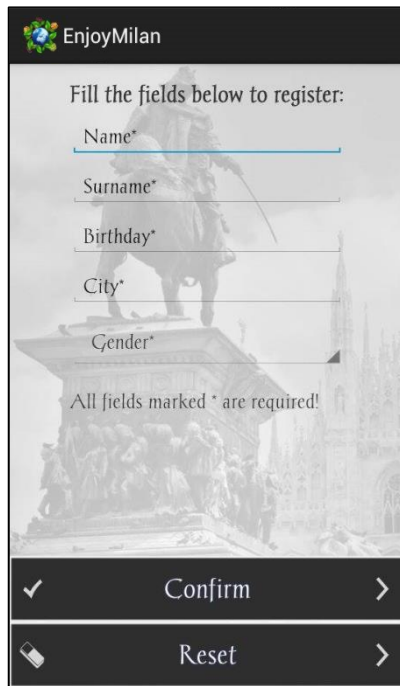


Figura 96. Registrazione.

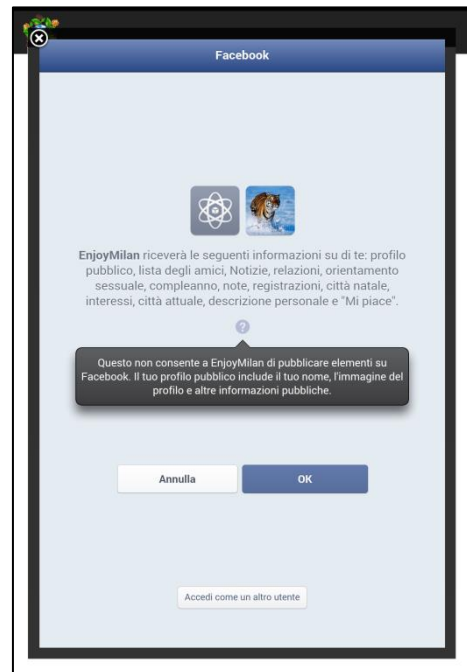


Figura 97. Login con Facebook.



Figura 98. Intro home.



Figura 99. Home.

1.2. La gestione delle preferenze

Questa funzionalità consente all'utente di esprimere le proprie preferenze in merito a ristoranti e/o punti di interesse. In particolare, viene data la possibilità di inserire un massimo di tre preferenze sul tipo di cucina e i servizi offerti da un ristorante, oppure sulle categorie di attrazioni, attività e vita notturna che lo possono interessare.

Il pattern di navigazione è di tipo *Index + Guided Tour* (Figura 100), grazie al quale il turista può decidere di esprimere la sua opinione su tutti gli argomenti o solo su una parte di essi (Figura 101 – Figura 106).

Le frecce ◀ e ▶ permettono di spostarsi di pagina in pagina in senso circolare, così come indicato dalle frecce arancioni ↔.

La freccia ▲ riporta alla radice dove avviene la scelta della categoria, così come indicato dalle frecce azzurre ↔.

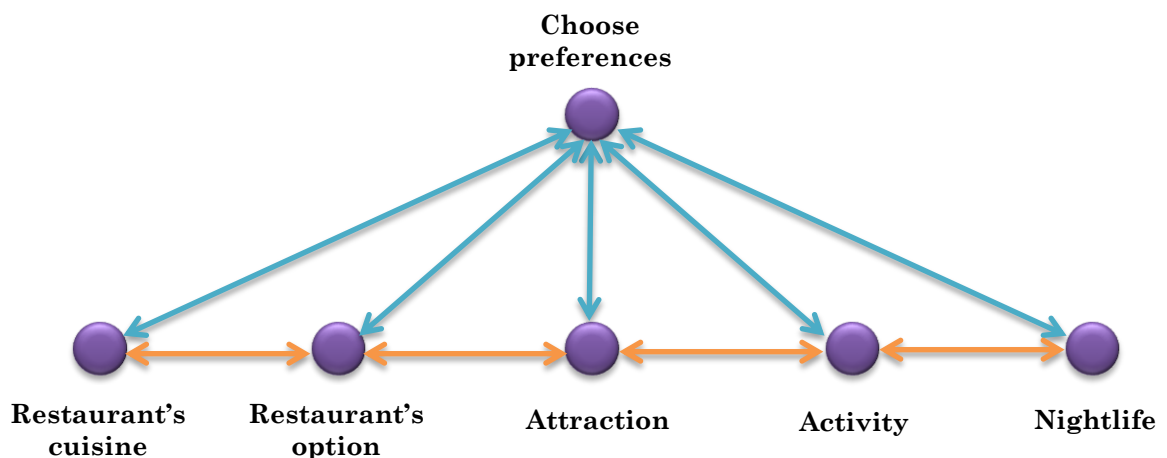


Figura 100. Pattern di navigazione.



Figura 101. Scelta della categoria.



Figura 102. Preferenze sul tipo cucina.



Figura 103. Preferenze sui servizi.



Figura 104. Preferenze sulle attrazioni.



Figura 105. Preferenze sulle attività.

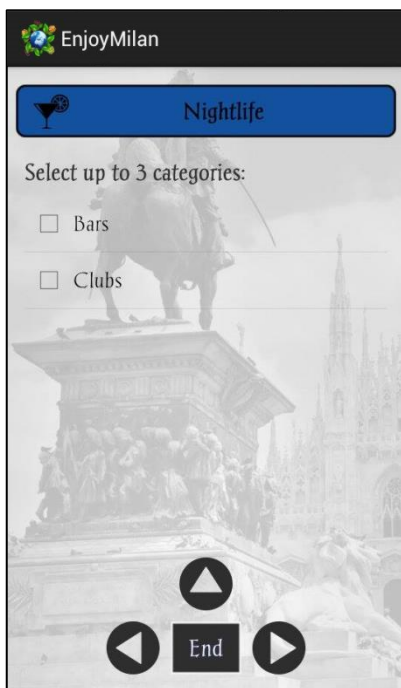



Figura 106. Preferenze sulla vita notturna.




Figura 107. Ordinamento preferenze.

1.3. La gestione del profilo

La pagina del profilo è accessibile dalla home, utilizzando l'icona  situata in alto a destra. All'interno di questa sezione, sono riassunte le informazioni anagrafiche espresse in fase di registrazione o estrapolate da Facebook dopo aver fatto il login.

Inoltre, se presente, viene riportata la lista delle preferenze dichiarate attraverso l'apposita funzionalità (Figura 108 e Figura 109).

Ogni preferenza all'interno della lista è accompagnata da un'icona, la quale simboleggia la categoria di appartenenza; affinché l'utente comprenda il significato di tali simboli, si riporta una breve legenda accessibile tramite l'icona  situata in alto a destra (Figura 110).

Per inserire, o aggiornare, le preferenze si utilizza l'apposita funzionalità *Explain your preferences*, o rispettivamente *Update preferences*, che porta l'utente alla schermata di scelta della categoria come mostrato nella Sezione 1.2; *Order preferences* invece, come dice la parola stessa, permette di riordinare la lista *Preferences* presente all'interno del profilo.

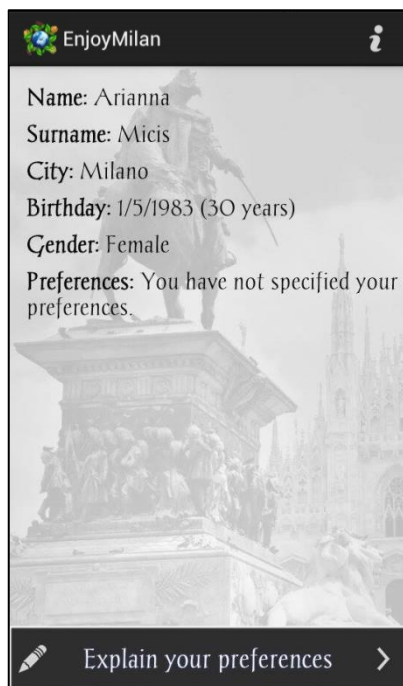


Figura 108. Profilo senza preferenze.



Figura 109. Profilo con preferenze.

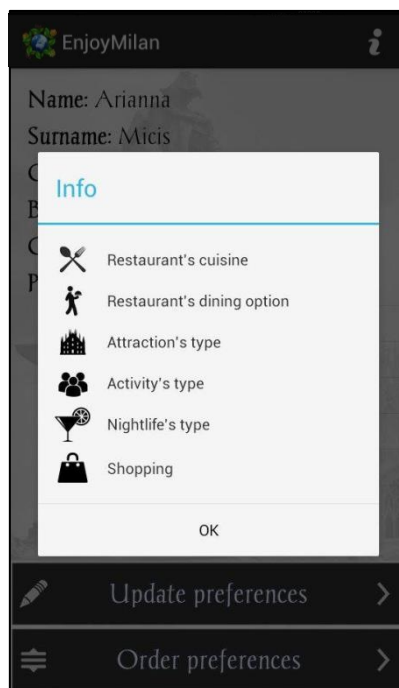


Figura 110. Legenda preferenze.

1.4 La ricerca “Around Me”

La ricerca “Around Me” è accessibile dalla home, utilizzando l’apposito bottone *Around Me*. Questa funzionalità permette all’utente di conoscere cosa si trova vicino a lui, in particolare ristoranti e punti di interesse presenti in un raggio di circa 10 km rispetto alla sua posizione geografica. Chiaramente per poter ricevere la lista di risultati è necessario essere connessi ad una rete (WiFi o dati indifferentemente), e abilitare il GPS dalle impostazioni del proprio Smartphone; in caso contrario sarà compito dell’applicazione segnalare l’errore all’utente.

La schermata risulta suddivisa in due sezioni: *RESTAURANT* e *POI*. A prescindere dalla tab selezionata, vengono restituiti i primi venti elementi ordinati in base alla distanza rispetto alla posizione geografica dell’individuo. Per ottenere ulteriori risultati si utilizza il bottone *More* presente in fondo alla pagina.

Ogni riga della suddetta lista contiene: un’icona che rappresenta la tipologia di elemento, il nome e l’indirizzo del luogo, e infine la distanza rispetto alla posizione rilevata dal GPS (Figura 111 e Figura 112).


Dal momento in cui la tipologia di elemento potrebbe non essere immediatamente percepibile osservando il simbolo, si è introdotta una piccola legenda accessibile mediante l'icona  presente in alto a destra (Figura 113).




Figura 111. Around Me – Restaurant.



Figura 112. Around Me – POI.



Figura 113. Legenda Around Me.

L'icona  presente in alto a destra permette di visualizzare sulla mappa i ristoranti o i punti di interesse mostrati dalla ricerca "Around Me" (Figura 114).

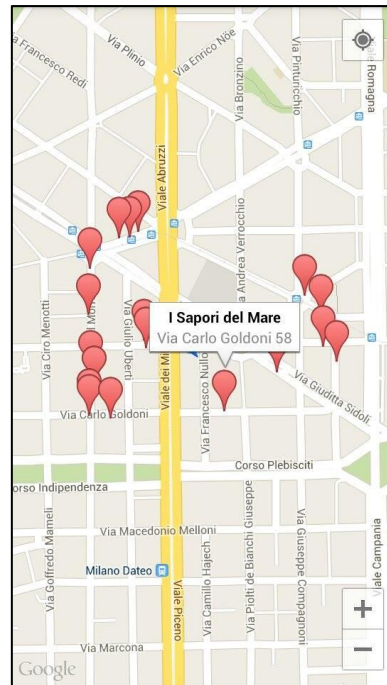



Figura 114. Mappa Around Me.

1.5 La lista dei preferiti

La lista dei preferiti è accessibile dalla home, utilizzando l'apposito bottone *My Favorites*. Si tratta di una sezione nella quale l'utente può salvare i ristoranti e i punti di interesse che gli piacciono, in modo tale da poterli consultare in ogni momento (Figura 115).

È possibile eliminare un elemento tenendo premuto sullo stesso e selezionando *Remove favorite* (Figura 116); per eliminare l'intero contenuto della lista dei preferiti si utilizza l'icona  presente in alto a destra.

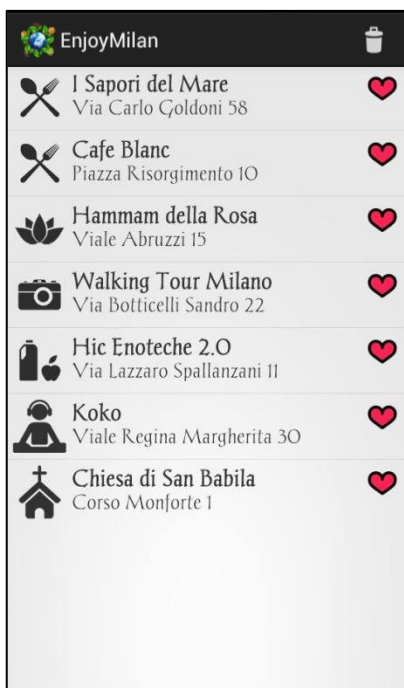


Figura 115. Lista preferiti.

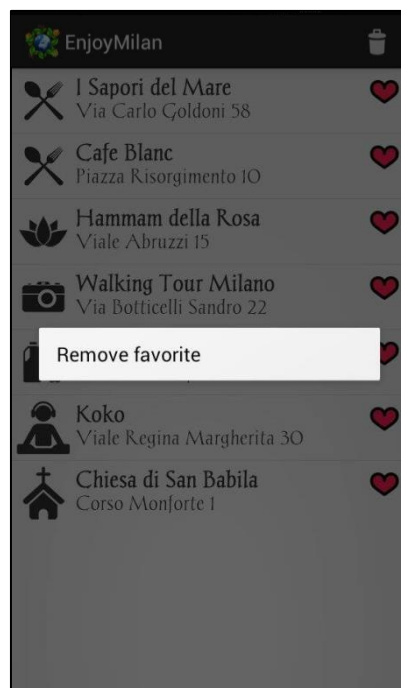


Figura 116. Rimuovere preferito.

Per salvare un elemento all'interno della lista dei preferiti, l'utente può:

- Dalla ricerca "Around Me", tenere premuto sull'oggetto desiderato e selezionare *Add to favorites* (Figura 117).

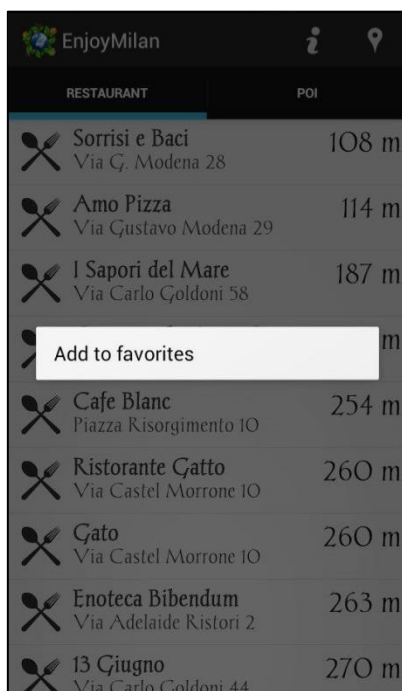


Figura 117. Aggiungere ai preferiti.



- Dalla pagina di dettaglio di un ristorante o punto di interesse, utilizzare l'icona  presente in alto a destra; nel momento in cui l'item viene aggiunto alla lista dei preferiti, l'icona cambia colore  e viene notificata l'avvenuta operazione (Figura 118 e Figura 119).



Figura 118. Elemento non preferito.

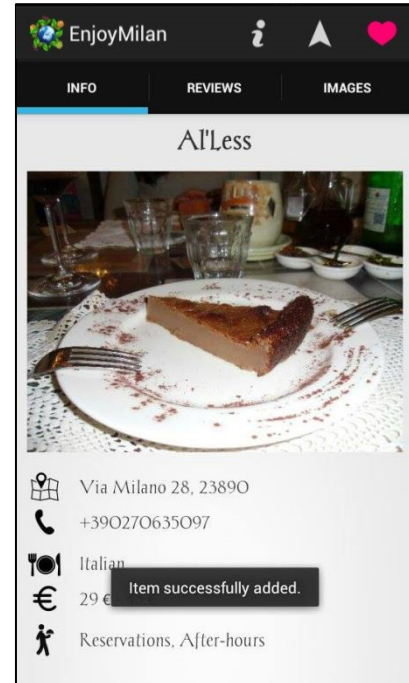



Figura 119. Elemento preferito.


- Durante l'interazione con il grande schermo, utilizzare la funzionalità di cattura  (vedi Paragrafo 3).

1.6 La pagina di dettaglio

La pagina di dettaglio è accessibile sia dalla ricerca "Around Me" sia dalla lista dei preferiti, cliccando sull'item di cui si desidera avere maggiori informazioni.

La schermata risulta suddivisa in tre sezioni: *INFO*, *REVIEWS* e *IMAGES*.

La sezione *INFO* fornisce una sorta di scheda informativa dove vengono mostrate le caratteristiche principali dell'oggetto; se si tratta di un ristorante si potranno trovare

informazioni riguardanti il tipo di cucina o la fascia di prezzo richiesta per consumare, se si tratta di un'attrazione ci sarà l'indicazione del prezzo del ticket, della durata della visita, e così via (Figura 120). Ancora una volta per aiutare l'utente a capire il significato dei simboli utilizzati, si riporta una legenda accessibile mediante l'icona  presente in alto a destra (Figura 121).

La sezione *REVIEWS* riporta le valutazioni associate all'elemento; per ognuna di esse viene mostrato lo username dell'utente che l'ha rilasciata, il voto assegnato e la parte testuale se presente (Figura 122).

La sezione *IMAGES* è una sorta di galleria fotografica; selezionando una figura è possibile visualizzarla in modalità full screen e, attraverso l'uso dello swipe a destra o a sinistra, si può scorrere tra le immagini presenti nella galleria (Figura 123).



Figura 120. Scheda informativa.

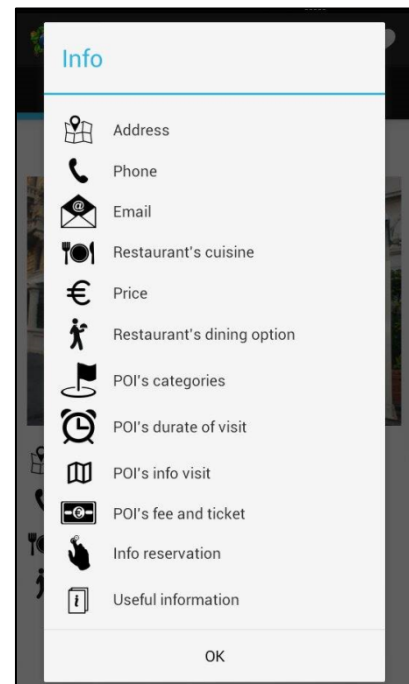



Figura 121. Legenda scheda informativa.





Figura 122. Recensioni.



Figura 123. Galleria immagini.

L'icona  presente in alto a destra avvia la navigazione verso l'elemento considerato.

L'icona  /  consente di aggiungere/rimuovere un oggetto dalla lista dei preferiti; come già anticipato nel precedente paragrafo, il colore del cuore permette di cogliere immediatamente se l'item è un preferito oppure no.

1.7 Controller

Come già specificato all'interno dei capitoli precedenti, viene data la possibilità all'utente di controllare il grande schermo attraverso il proprio Smartphone o Tablet. In altre parole, il dispositivo personale diventa il remote controller con cui navigare tra i contenuti presentati dal display pubblico.

Questa funzionalità è accessibile dalla home, utilizzando l'apposito bottone *Controller*. L'interfaccia del remote controller è costituita da una serie di bottoni che, insieme ad alcune gestive specifiche, consentono l'interazione con il grande schermo (Figura 124 e Tabella 15).

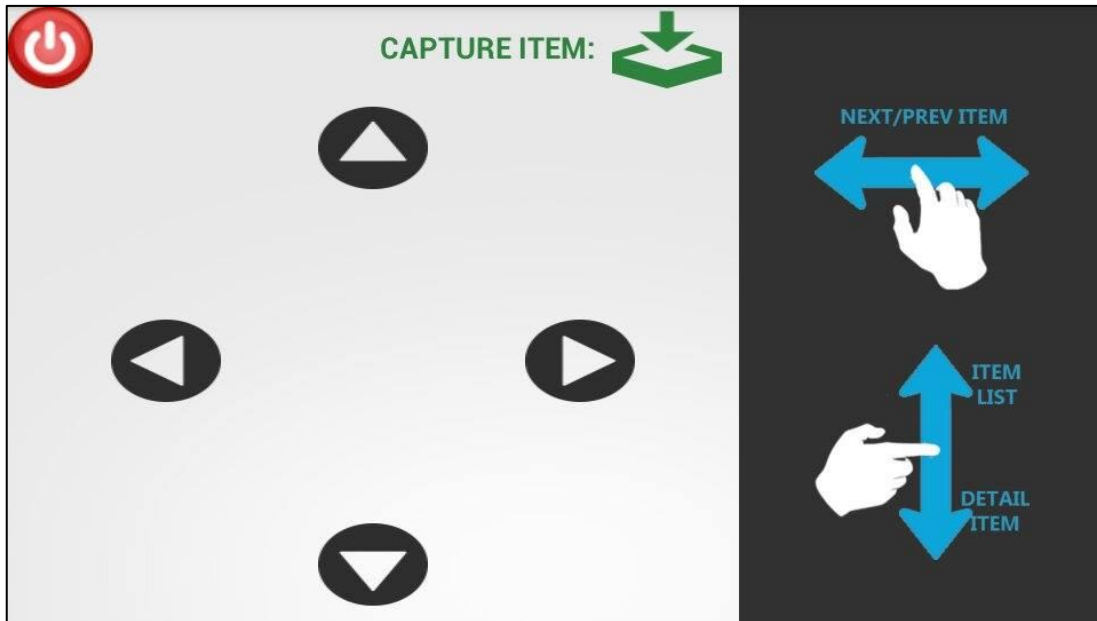


Figura 124. Controller.

Icona	Contesto: lista di item	Contesto: dettaglio item
	Permette il passaggio alla pagina precedente o successiva rispetto alla lista di risultati.	Permette il passaggio al dettaglio dell'item precedente o successivo.
	Swipe verso il basso permette di scendere nel dettaglio dell'item corrente. Swipe verso l'alto disabilitato!	Swipe verso l'alto permette di tornare alla pagina contenente la lista di risultati. Swipe verso il basso disabilitato!
	Permettono lo spostamento di item in item all'interno della pagina corrente, secondo il senso della freccia selezionata.	Freccia top e freccia bottom permettono di passare di scheda in scheda secondo l'ordine: Details → Reviews → Images → Maps → Details → ... Ad eccezione delle schede Reviews e Images, la freccia next e la freccia previous sono invisibili!

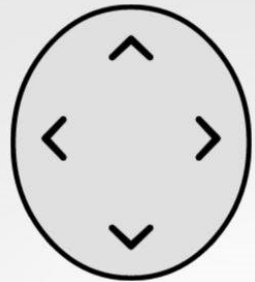


	<p>Sono invisibili!</p>	<p>Viene mostrato solo nella scheda Maps, per permettere lo spostamento del focus sulla mappa. Ad eccezione di questa scheda, sono sempre invisibili!</p>
	<p>Permette la cattura dell'item, ovvero l'aggiunta dell'elemento alla lista dei preferiti presente nell'applicazione.</p>	
	<p>Permette la chiusura della connessione con il grande schermo.</p>	

Tabella 15. Significato comandi.

Per completare la trattazione, gli autori hanno inserito alla fine di questo capitolo uno scenario completo dove per ogni comando del device viene mostrato il cambiamento di stato del grande schermo (una sorta di meccanismo azione-reazione).

2. Il grande schermo

Riprendendo ciò che era stato definito in fase di analisi, in occasione dell'Expo si prevede la dislocazione di schermi in vari punti della città, il cui compito è mostrare informazioni e contenuti multimediali sui principali luoghi di interesse di Milano.

Nel progetto di tesi, si vuole sfruttare la presenza di questi display per far conoscere al turista cosa può offrire Milano, in relazione ai suoi gusti e interessi personali. E, attraverso il device personale, permettere all'individuo di interagire con i contenuti mostrati dal grande schermo.

2.1. Il messaggio di benvenuto

Quando l'utente decide di iniziare l'interazione, il grande schermo cambia stato mostrando un messaggio di benvenuto (Figura 125).

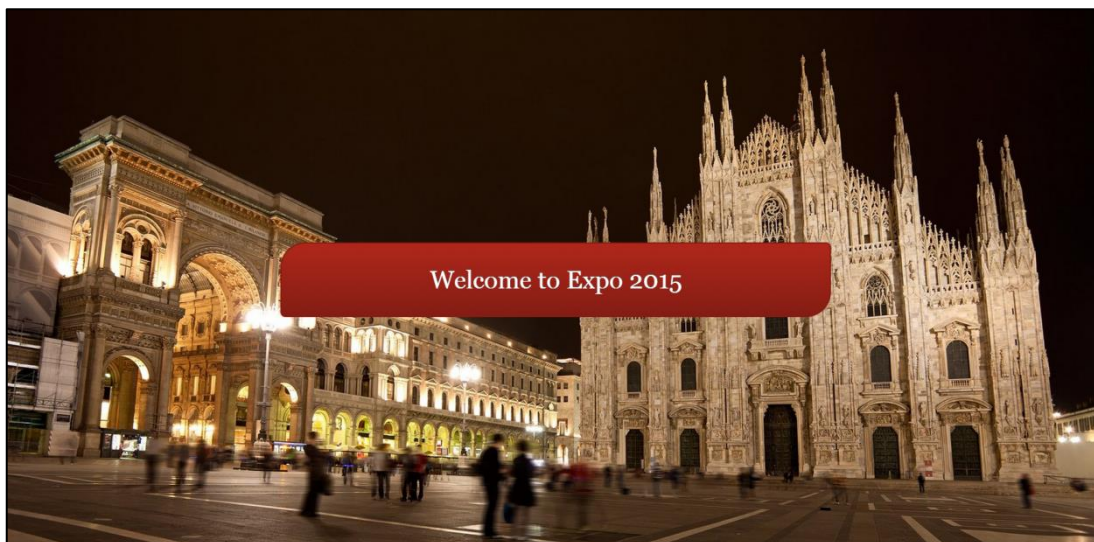


Figura 125. Messaggio di benvenuto.

2.2. La scelta della categoria

Prima di generare la lista di risultati, viene data la possibilità all'individuo di scegliere cosa vorrebbe conoscere tra ristoranti, attrazioni, attività, locali e negozi. La scelta *All* restituisce un insieme misto, in cui sono presenti in diverse misure le categorie citate sopra (Figura 126).

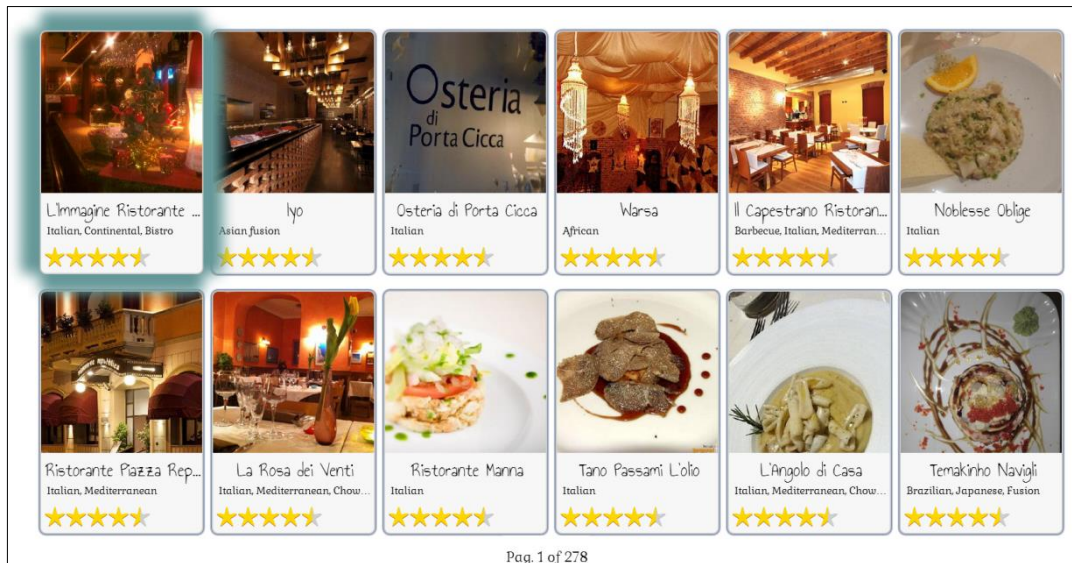


Figura 126. Scelta categoria.

2.3. La lista di risultati

Una volta scelta la categoria, il grande schermo mostra una serie di elementi appartenenti ad essa che, secondo l'algoritmo sottostante, rispecchiano maggiormente gli interessi della persona.

Per ogni oggetto viene mostrata una foto, il nome e il voto medio attribuito da TripAdvisor sulla base delle valutazioni degli utenti. In fondo alla schermata si trova l'indicazione della pagina corrente rispetto all'intera lista di risultati (Figura 127).



Pag. 1 of 278

Figura 127. Lista di risultati.

2.4. I dettagli dell'elemento

Per ogni elemento della lista presentata è possibile visualizzarne i dettagli.

In particolare:

- *Details* mostra le principali informazioni quali indirizzo, numero di telefono, contatto email, e altro ancora (Figura 128);



Figura 128. Details elemento.

- *Reviews*, per la quale gli autori hanno previsto tre varianti:

- o La prima versione presenta tutte le valutazioni associate all'elemento, suddividendole in blocchi di dodici per ogni pagina. Per ognuna di esse viene fornito il rating, lo username ed eventuali informazioni su sesso ed età del recensore. La parte testuale, anche se presente nel database, non viene mostrata per motivi di leggibilità (Figura 129).

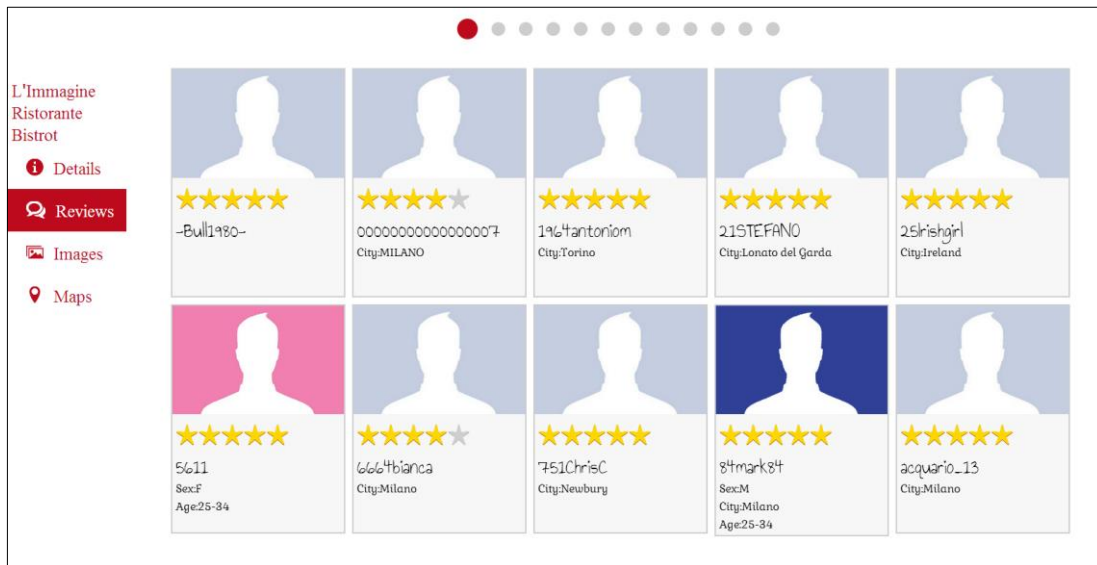


Figura 129. Reviews elemento – prima versione.

- o La seconda versione fornisce una sintesi delle valutazioni espresse su TripAdvisor, classificandole per età e sesso degli utenti (Figura 130).

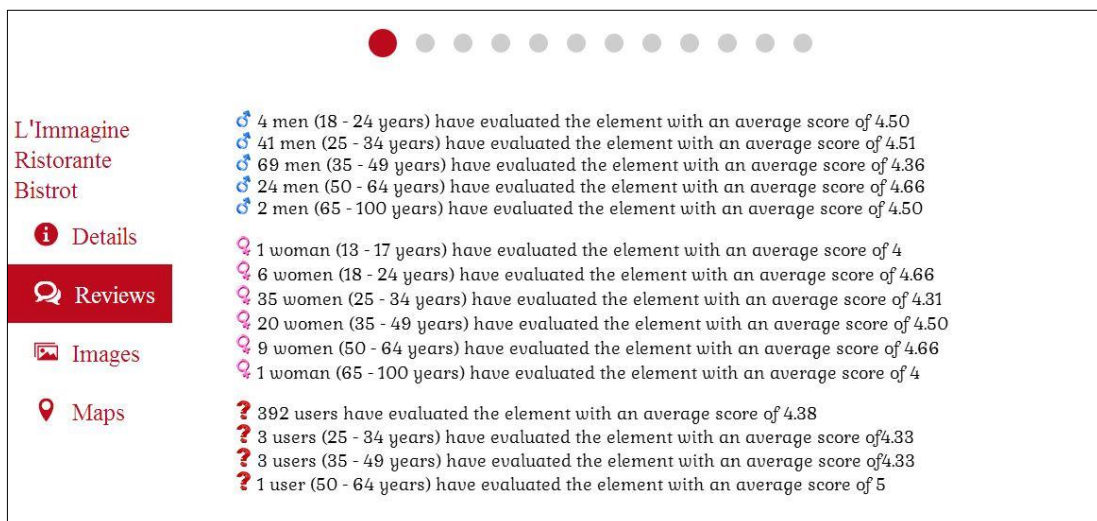


Figura 130. Reviews elemento – seconda versione.

- La terza versione, infine, è quella che attualmente viene utilizzata nel progetto di tesi. Fornisce sempre una sintesi delle valutazioni con l'unica differenza che, invece di usare una descrizione testuale a volte noiosa e fuorviante, si introducono i grafici a torta (Figura 131).

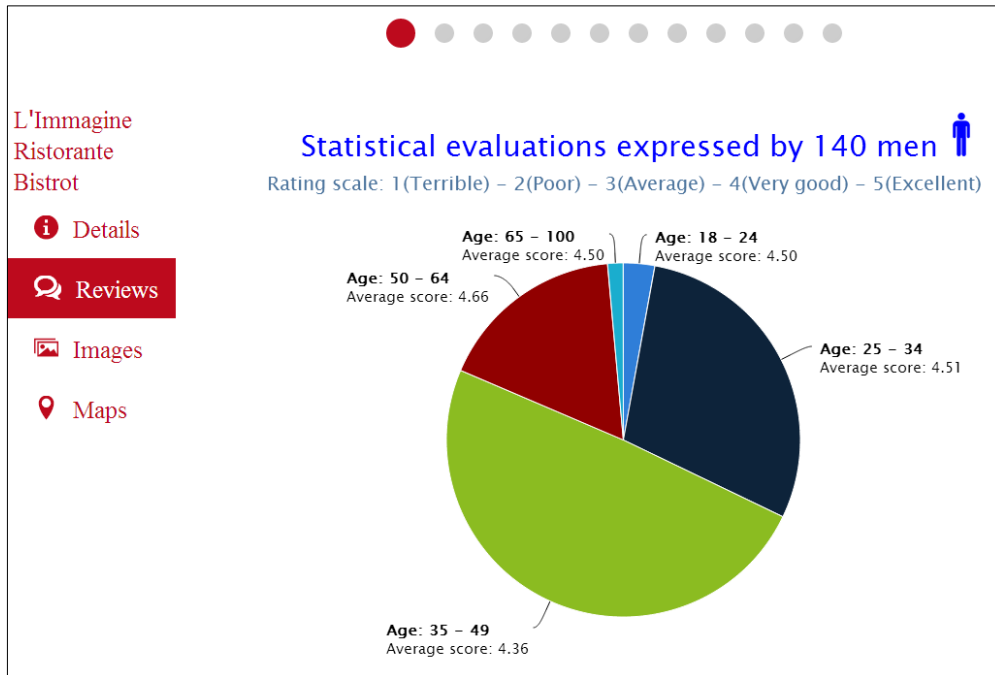


Figura 131. Reviews elemento – terza versione.

- *Images* visualizza in modalità full screen le immagini presenti nella galleria fotografica dell'item (Figura 132);

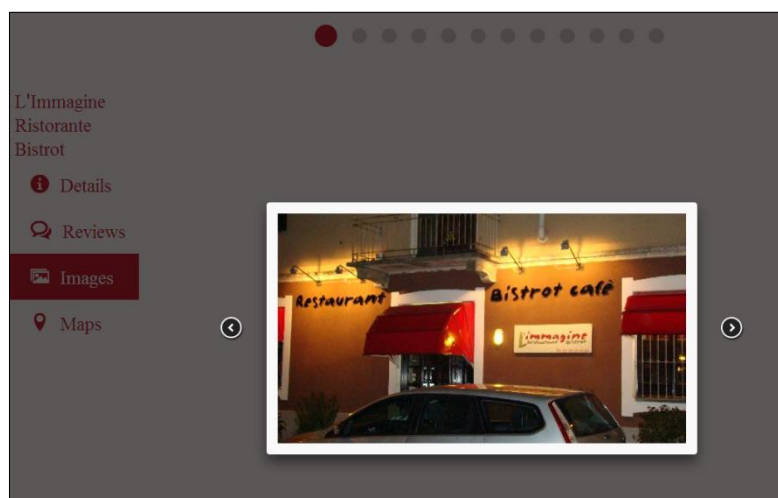


Figura 132. Images elemento.

- *Maps* mostra la posizione dell'oggetto sulla cartina (Figura 133).

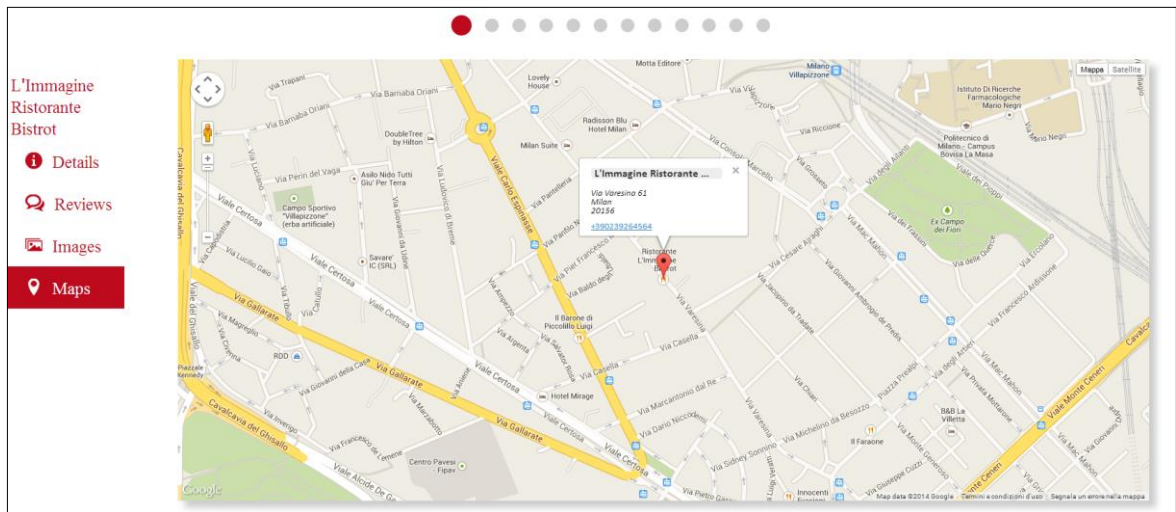


Figura 133. Maps elemento.

In tutte le immagini mostrate si nota la presenza di due barre. La barra superiore serve a contestualizzare l'oggetto evidenziato; in particolare permette di sapere che posizione occupa rispetto ai dodici elementi mostrati grazie al pallino colorato. La barra laterale, invece, risulta utile quando si scende nel dettaglio di un elemento per capire cosa si sta visualizzando tra scheda informativa, recensioni, galleria di immagini e posizione sulla mappa.

3. L'interazione tra utente e grande schermo attraverso il device personale

Questo è senza ombra di dubbio l'aspetto più interessante del lavoro di tesi. Dopo aver definito gli obiettivi e spiegato le tecnologie utilizzate, si vuole mostrare uno scenario completo in grado di metter in risalto le potenzialità di questo tipo di interazione.

Supponiamo che un utente stia camminando per Milano e che, ad un certo punto, si accorga della presenza di un display. Per semplicità consideriamo il caso in cui il display pubblico sia libero, ovvero nessun altro turista lo sta utilizzando.

Dalla home dell'applicazione seleziona "Controller" e, seguendo le indicazioni riportate nel messaggio, posiziona lo Smartphone sul lettore NFC situato di fronte al grande schermo (Figura 134).



Figura 134. Fase preliminare all'interazione.

Dal momento in cui la relazione tra grande schermo e device è di tipo unario, non è consentito a più persone contemporaneamente di usufruire di questo servizio. Se il display pubblico è già occupato nell'interazione con una persona, il nuovo arrivato deve aspettare la chiusura della connessione esistente prima di aprirne una nuova. Sarà compito dell'applicazione notificare all'utente l'impossibilità di iniziare la comunicazione invitandolo ad attendere il proprio turno.

Se l'operazione va a buon fine, il grande schermo visualizza un messaggio di benvenuto (Figura 135) e inizia a prelevare le informazioni personali dell'utente dall'account di Facebook (Figura 136) o dal profilo costruito in fase di registrazione (Figura 137). Per evitare l'insorgere di situazioni inattese dove vengono prelevati solo una parte dei dati richiesti, non sarà possibile utilizzare lo Smartphone durante questa fase!



Figura 135. Messaggio di benvenuto – grande schermo.

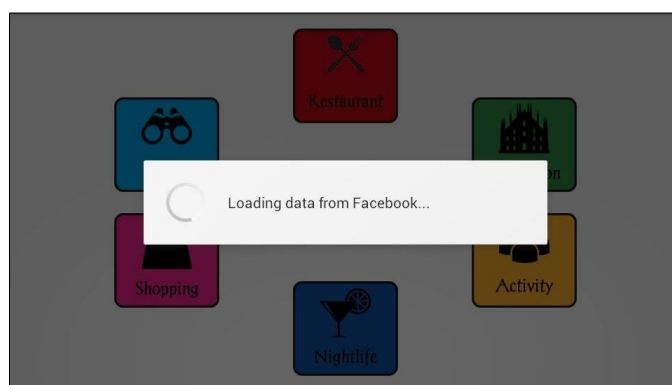


Figura 136. Caricamento dati da Facebook – applicazione.

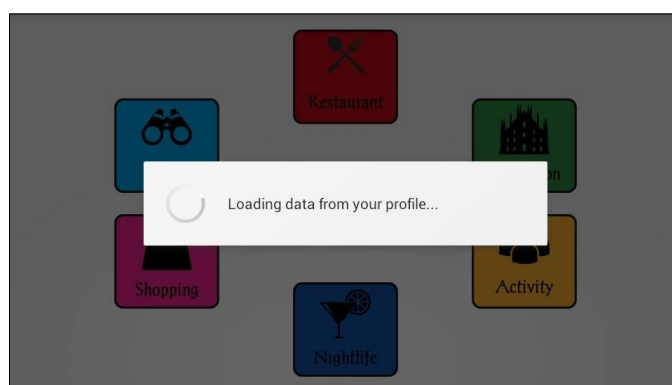


Figura 137. Caricamento dati dal profilo – applicazione.

Terminato il processo di scaricamento delle informazioni, Smartphone e grande schermo mostreranno contemporaneamente una schermata contenente sei macro aree colorate: “Restaurant”, “Attraction”, “Activity”, “Shopping”, “Nightlife” e “All” (Figura 138 e Figura 139).

In questo modo viene data la possibilità al turista di scegliere cosa farsi consigliare dal display, potendo selezionare una specifica categoria oppure optare per un lista di risultati mista.



Figura 138. Scelta categoria - grande schermo.



Figura 139. Scelta categoria – applicazione.

Supponiamo che l'utente selezioni "Restaurant". Il grande schermo, a questo punto, prende tutti gli elementi di tipo "Restaurant" e li ordina sulla base dei gusti dell'individuo per generare la lista personalizzata. Durante questa fase, lo Smartphone segnala all'utente di attendere qualche istante per completare l'elaborazione di tale lista (Figura 140).

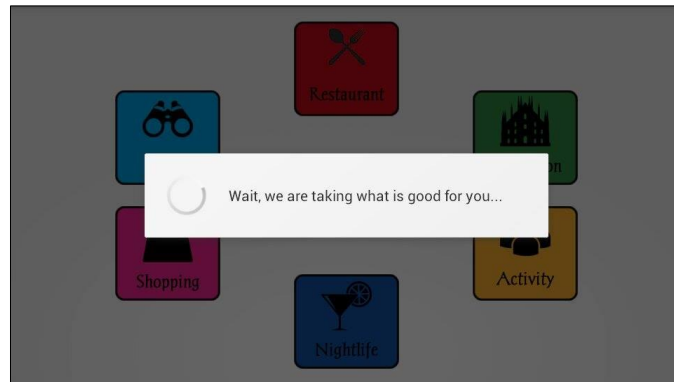


Figura 140. Attesa lista di risultati – applicazione.

Una volta ottenuta la lista, il grande schermo mostra i primi dodici risultati e lo Smartphone presenta l'interfaccia del remote controller per iniziare l'interazione (Figura 141 e Figura 142).

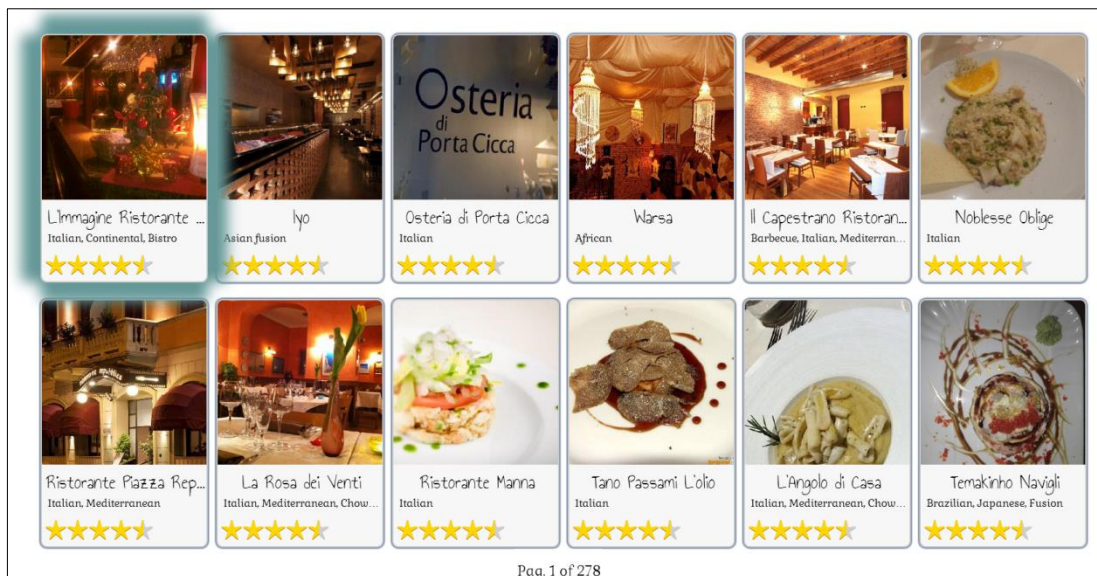


Figura 141. Primi dodici risultati – grande schermo.

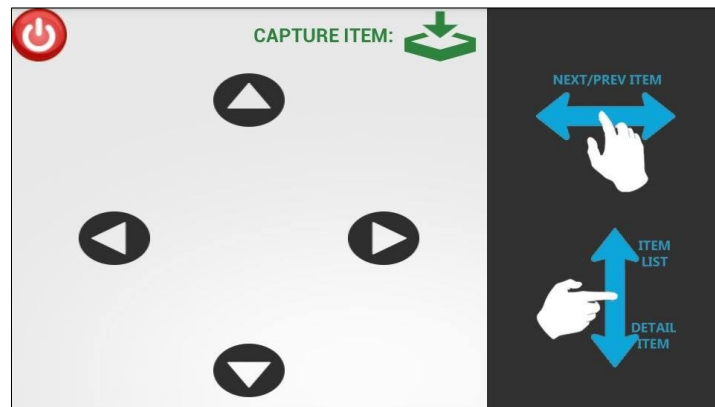


Figura 142. Remote controller – applicazione.

Supponiamo che l'utente sia interessato a conoscere il ristorante "L'immagine Ristorante Bistrot". In accordo con i comandi presenti sul controller, fa swiipe verso il basso ed entra nei *Details* del ristorante (Figura 143).



Figura 143. Details ristorante – grande schermo.

Utilizzando la freccia ▼ si sposta sulla scheda *Reviews* la quale mostra un primo grafico a torta (Figura 144). Attraverso le frecce ◀ e ▶, se presenti, può spostarsi di grafico in grafico per vedere la statistica delle valutazioni espresse dagli uomini o dalle donne su TripAdvisor (Figura 145).

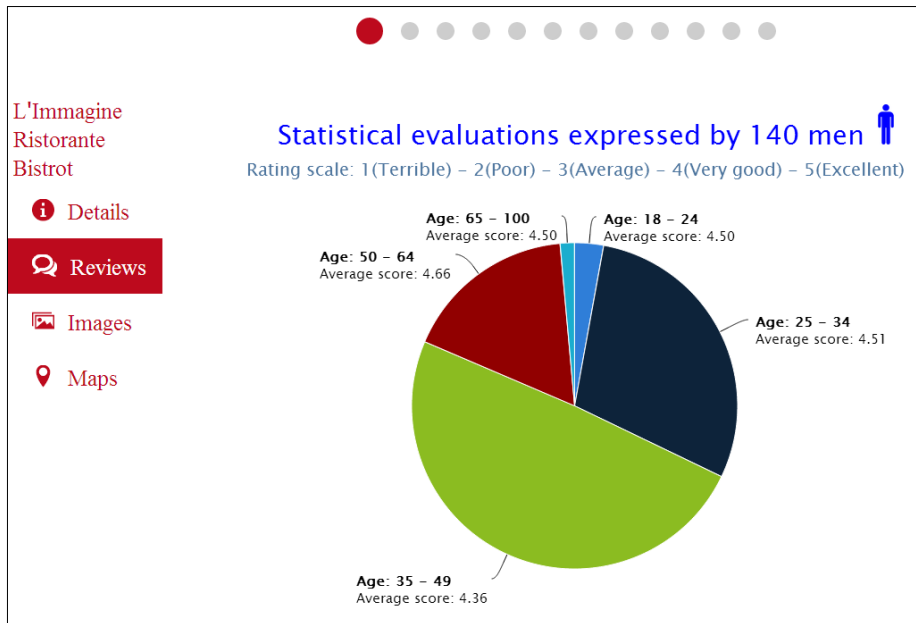


Figura 144. Reviews ristorante – grande schermo.

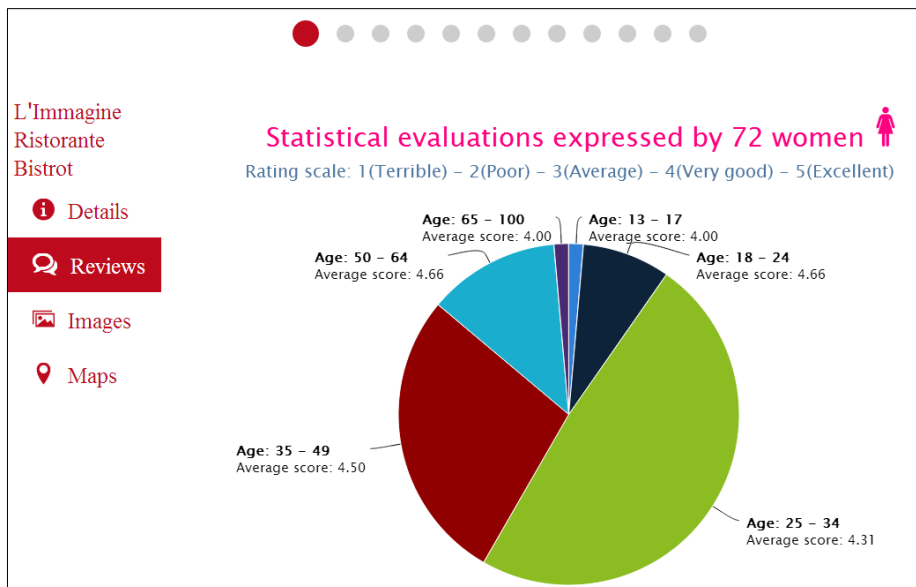


Figura 145. Next reviews ristorante – grande schermo.

Utilizzando la freccia ▼ si sposta sulla scheda *Images* (Figura 146). Attraverso le frecce ◀ e ▶, se presenti, può spostarsi di immagine in immagine scorrendo la galleria fotografica del ristorante (Figura 147).

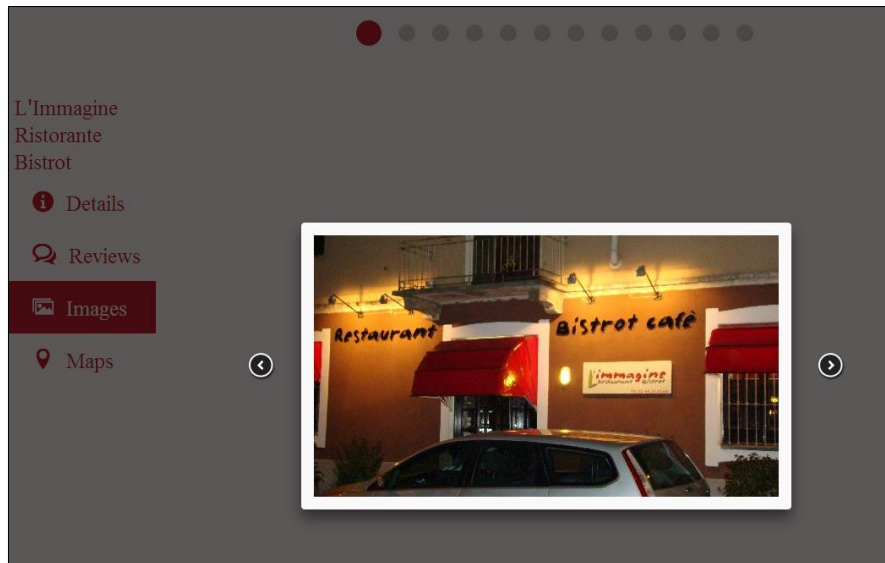


Figura 146. Img ristorante – grande schermo.

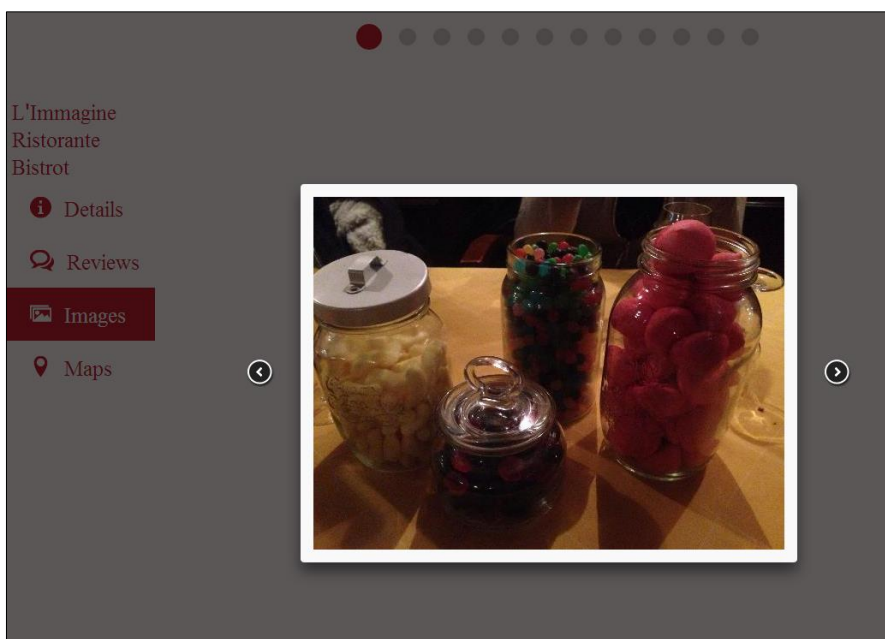


Figura 147. Next img ristorante – grande schermo.

Utilizzando la freccia ▼ si sposta sulla scheda *Maps* la quale mostra la posizione del ristorante (Figura 148). A questo punto nell'interfaccia del remote controller compare ◀▶, il quale permette di spostarsi all'interno della mappa (Figura 149).

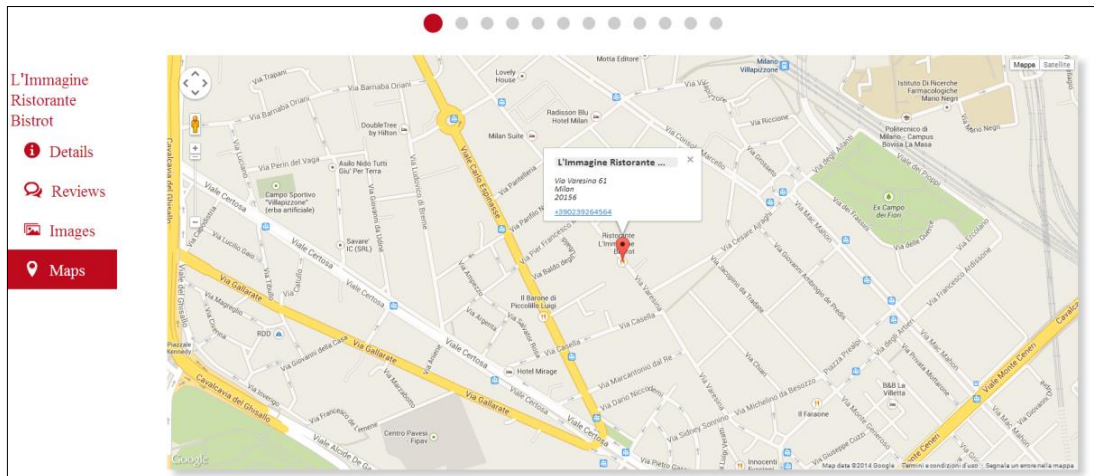


Figura 148. Maps iniziale – grande schermo.

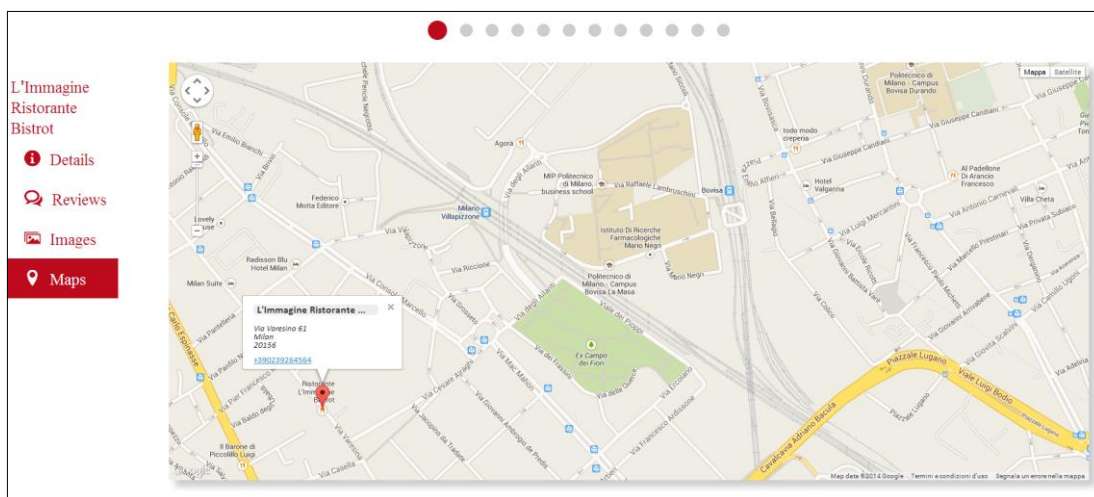



Figura 149. Maps spostata – grande schermo.

Supponiamo ora che l'utente non sia pienamente soddisfatto di ciò che ha visto, e che voglia conoscere i dettagli di altri ristoranti presentati dal sistema. In accordo con i comandi del controller, fa swipe verso sinistra fino a quando non raggiunge l'elemento desiderato (Figura 150).



Figura 150. Details ristorante desiderato – grande schermo.

Dopo aver letto la scheda informativa del ristorante, l'utente si ritiene soddisfatto e decide di aggiungerlo alla lista dei preferiti, così da poterlo consultare in ogni momento. Utilizzando il bottone , l'item viene aggiunto alla lista dei preferiti presente sull'applicazione (Figura 151).

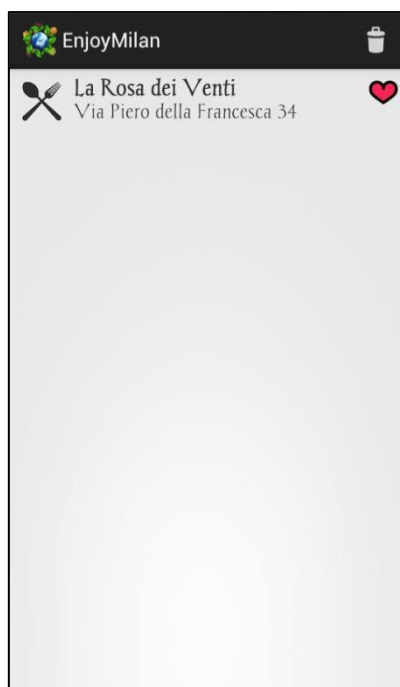


Figura 151. Item aggiunto ai preferiti – applicazione.

A questo punto l'utente può:

- Continuare a navigare tra i contenuti raccomandati dal grande schermo;
- Tornare alla schermata di scelta della categoria utilizzando il back dello Smartphone;
- Chiudere la comunicazione con il grande schermo.

In merito all'ultimo punto, gli autori hanno previsto quattro modalità:

1. Utilizzare l'apposito bottone presente nell'interfaccia del remote controller;
2. Dalla schermata di scelta della categoria utilizzare nuovamente il back dello Smartphone;
3. Mettere l'applicazione in background;
4. Allontanarsi di almeno 20 passi dal grande schermo.

Le ultime due modalità sono state introdotte per liberare il display pubblico e renderlo disponibile ad altre persone, nel caso in cui l'utente si dimenticasse di chiudere esplicitamente la comunicazione.

CONCLUSIONI

L'informazione digitale si sta diffondendo sempre di più nell'ambiente che ci circonda; è ovunque e disponibile in qualsiasi momento, grazie alla diffusione di dispositivi tascabili come Smartphone e Tablet.

Allo stesso tempo, la sovrabbondanza di informazioni fa scaturire l'esigenza di estrapolare i contenuti rilevanti dal vasto insieme di dati disponibili. Ed ecco che iniziano ad affermarsi i sistemi di raccomandazione, il cui compito è aiutare le persone a trovare ciò che vogliono. L'introduzione di tali sistemi impatta fortemente sul modo dell'individuo di prendere le decisioni: non è più l'uomo a cercare e trovare una soluzione alle proprie esigenze, ma è il sistema a proporgli ciò che maggiormente si adatta ai suoi bisogni.

La crescita è guidata anche dalla proliferazione dei dispositivi mobile che non sono più soltanto lo strumento con cui accedere a Internet in mobilità, ma sempre più un modo per integrare la fruizione di altri contenuti. L'esigenza di offrire servizi personalizzati agli utenti si scontra però con le sfide introdotte dalla Mobile Recommendations, che deve fare i conti con le caratteristiche intrinseche del dispositivo, prima fra tutte le limitate dimensioni dello schermo. Si affaccia quindi una nuova forma di presentazione dei contenuti, attraverso l'uso di grandi schermi

situati in aree metropolitane, stazioni ferroviarie, aeroporti e, più in generale, negli spazi pubblici.

Alla base del progetto di tesi, è stato condotto uno studio atto a supportare la personalizzazione di grandi quantità di contenuti multimediali presentati sui grandi schermi, e ad investigare l'uso dei dispositivi mobile per interagire con essi.

Sulla base delle analisi svolte, si è realizzato un sistema in grado di far conoscere ai turisti in visita a Milano i ristoranti e i punti di interesse che maggiormente si adattano alle sue esigenze e interessi personali.

Analizzando i risultati, ci si ritiene soddisfatti del progetto sviluppato. In particolare l'applicazione creata risulta intuitiva e semplice da utilizzare; l'inserimento di icone a dispetto di lunghe informazioni testuali si è rivelata una scelta vincente nel rendere l'interfaccia user-friendly. La comunicazione con il grande schermo è stata implementata attraverso l'uso dei WebSockets, una nuova tecnologia che ha permesso di ridurre i tempi di latenza invogliando l'utente a usufruire del servizio. Infine il riconoscimento dell'individuo è stato fatto adottando l'NFC in grado di rendere trasparente l'intera procedura.

La modularità dell'infrastruttura è senza dubbio uno degli aspetti ingegneristicamente più significativi del lavoro svolto. Il vantaggio che ne deriva ricade innanzitutto sull'inserimento futuro del vero algoritmo di raccomandazione, che può essere fatto senza cambiare la struttura del progetto. In secondo luogo, la parte di front-end, contenente la logica per la creazione delle schermate da mostrare sul grande schermo, risulta indipendente dal sistema operativo installato sul dispositivo mobile. Questo chiaramente facilita il lavoro futuro degli sviluppatori di altri sistemi operativi, come iOS o Windows Phone.

Nonostante i vantaggi menzionati, l'architettura risente di una certa limitazione legata alla singola User-Experience nell'interazione con il grande schermo. Nella parte finale di questo capitolo, sono state inserite una serie di ipotesi in grado di estendere il lavoro alla sfera della Group Recommendations.

1. Problemi riscontrati

I problemi riscontrati in fase di implementazione si possono riassumere essenzialmente in tre categorie. La prima riguarda la connessione Internet, e in particolare la rete WiFi del Politecnico di Milano. Non è stato possibile testare l'apertura della comunicazione tramite WebSockets sfruttando tale rete; le ragioni di questo rifiuto sono riconducibili ad alcune restrizioni inserite nel proxy o alla presenza di firewall adibiti a bloccare richieste di questo tipo. Il problema infatti non si presentava utilizzando le reti WiFi casalinghe o ricorrendo al traffico dati di un gestore telefonico.

La seconda problematica riguarda la mancanza fisica del lettore NFC. Nei primi stadi del lavoro di tesi, si è cercato di capire quale potesse essere il modello più adatto alle esigenze del progetto. In particolare, per dare vita all'integrazione, è richiesto un componente in grado di favorire la comunicazione bilaterale e lo scambio di dati da/verso il grande schermo. I lettori attualmente presenti sul mercato italiano non rispecchiano queste caratteristiche, consentendo un passaggio di informazioni solo in una direzione. Con alcune ricerche sul Web, si è trovato il modello desiderato su un negozio online e, tramite il Politecnico di Milano, ci si è mossi per procedere con l'acquisto. Purtroppo il lungo iter burocratico previsto (non ancora concluso!) e i tempi stretti imposti dall'avvicinarsi della laurea, non hanno permesso agli autori di testare quali potessero essere i reali svantaggi e vantaggi nell'utilizzo di questa tecnologia. Ai fini del progetto, la presenza del lettore NFC viene quindi simulata impostando manualmente l'indirizzo IP del grande schermo. Infine, ma non per questo meno importante, si trova il malfunzionamento di Eclipse come ambiente di sviluppo Android. Ciò ha portato spesso a generare problemi non legati a righe di codice sbagliate ma a errori interni del software. Dopo aver perso tempo a cercare di capire dove potesse essere l'errore, bastava un semplice "Clean" del progetto o un riavvio del programma per risolvere tutti i problemi. Questo evidentemente ha impattato nella fase di sviluppo, sia in termini di tempo che di energie mentali, nell'inutile ricerca di un errore umano nemmeno presente.

2. Challenges affrontate

Oltre allo studio sull'integrazione tra piccoli e grandi schermi, la sfida maggiormente affrontata dagli autori riguarda il criterio da utilizzare per suggerire la lista personalizzata di ristoranti e punti di interesse all'utente finale. Attualmente nel progetto di tesi non si effettua una vera e propria raccomandazione, in quanto gli items mostrati dal grande schermo sono filtrati dal dataset in base alle preferenze dell'utente (esplicite o implicite che siano). Più che di raccomandazione, si parla di ranking dei risultati ottenuto tramite la formula dello Shrinkage Factor.

Il portale di TripAdvisor prevede che gli utenti possano valutare item specifici ma non esprimere una preferenza sulle loro categorie di appartenenza; pertanto, è possibile affermare "mi piace il ristorante Mare e Monti" ma non "mi piacciono i ristoranti di pesce". Mancando queste informazioni, non è possibile scrivere un algoritmo di raccomandazione che colleghi in qualche modo le preferenze su una categoria (in questo caso la cucina di pesce) con le valutazioni espresse su un singolo item (in questo caso il ristorante Mare e Monti).

La soluzione potrebbe essere quella di prevedere all'interno del prototipo una sezione in cui l'utente possa esprimere un feedback su ristoranti e punti di interesse visitati recentemente. E, con queste nuove informazioni disponibili, si può procedere verso due direzioni, utilizzando un algoritmo content-based o un algoritmo di tipo collaborativo. Il primo consente di ricavare i gusti della persona osservando l'item valutato, e propone nuovi articoli in relazione a quanto scoperto. Ad esempio, l'utente esprime un parere positivo sui seguenti ristoranti "Mare e Monti", "Sapori del mare" e "La pescheria"; il sistema deduce una preferenza sulla cucina a base di pesce, e presenta nuovi ristoranti che appartengono a questa categoria.

Il secondo, invece, va a identificare punti in comune tra individui sulla base delle valutazioni da loro espresse. Ad esempio, l'utente esprime un parere positivo sul ristorante "Mare e Monti"; il sistema va a guardare le persone che hanno apprezzato questo ristorante e, accorgendosi che tra le loro preferenze emerge il ristorante cinese "La collina d'oro", lo propone all'utente iniziale.

Le soluzioni sopra proposte non prendono in considerazione le informazioni estrapolate da Facebook, che devono essere opportunamente lavorate per essere utilizzate come input del motore di raccomandazione. Come ampiamente discusso all'interno dell'elaborato, ogni like prelevato dal social network è associato ad una categoria che lo descrive (ad esempio "Restaurant/Cafe" per il ristorante giapponese "Ginko"), e ogni categoria a sua volta viene descritta da un insieme di sottocategorie (ad esempio "Sushi" è una sottocategoria di "Restaurant/Cafe").

Affinché queste informazioni risultino utili ai fini della raccomandazione, bisogna in qualche modo collegarle al dataset di TripAdvisor. In altre parole, è importante trasformare le preferenze implicite nascoste tra i likes di Facebook in preferenze esplicite su ristoranti e punti di interesse. Riprendendo l'esempio del "Ginko", la categoria "Sushi" si può ribaltare in una preferenza dell'utente sul tipo di cucina "Japanese" e "Sushi".

Per effettuare questo collegamento, nel progetto di tesi si è inizialmente utilizzato lo string matching il quale si occupa di confrontare la sottocategoria del like di Facebook con il contenuto delle tabelle *tipoCucina*, *opzioni*, *tipoAttrazione*, *tipoAttivita*, *tipoVitaNotturna* del dataset di TripAdvisor. Ad esempio, se la sottocategoria del ristorante è "Italian Restaurant", si deduce una preferenza dell'utente per il tipo di cucina "Italian" dal momento in cui il matching ha riscontrato una parola in comune tra le due stringhe. Questa soluzione, per quanto facile da implementare, risulta in alcune situazioni ostica perché potrebbe portare all'insorgere di falsi match. In alternativa, si è optato quindi per scaricare l'intera lista delle sottocategorie presenti in Facebook e collegare manualmente ognuna di esse con la relativa informazione dentro il dataset di TripAdvisor.

3. Sviluppi futuri

Come già anticipato nel capitolo iniziale sullo stato dell'arte, fare Group Recommendations è tutt'oggi una sfida ancora aperta. Non esiste un algoritmo da applicare, un modello da seguire o un insieme di regole valide in ogni situazione.

Innanzitutto bisogna capire con che tipo di gruppo si sta lavorando. È un insieme di amici con gusti simili? Sono persone che casualmente si trovano a condividere un'esperienza insieme?

Allo stato attuale, il progetto di tesi supporta la singola User-Experience per quanto riguarda l'interazione tra utente e grande schermo. In altre parole, se una persona sta navigando tra i contenuti mostrati dal display pubblico, nessun altro individuo può aggiungersi/interferire nella comunicazione. Il ragionamento è perfetto dal punto di vista teorico, in quanto consente all'utente di accentrare tutta l'attenzione del sistema e godere di una lista di raccomandazione personalizzata. A livello pratico, però, questa situazione difficilmente si potrebbe verificare, a maggior ragione se si tratta di una manifestazione di una certa portata come l'Expo dove si stima l'arrivo di un afflusso elevato di persone.

Risulta quindi doveroso avanzare delle ipotesi per estendere il progetto, includendo la possibilità di fare Group Recommendations. Tracciando i possibili sviluppi futuri, gli autori hanno delineato due scenari da seguire a seconda del genere di gruppo che si presenta di fronte al grande schermo.

3.1 Scenario 1: gruppi volontari

I gruppi volontari sono costituiti da un insieme di persone che contemporaneamente si presentano di fronte al grande schermo per iniziare l'interazione. A titolo di esempio, si potrebbe pensare ad un gruppo di amici che vuole conoscere i locali più fashion di Milano, oppure colleghi di lavoro che desiderano sapere un buon posto dove fermarsi a mangiare nella pausa pranzo.

In questi casi, la soluzione migliore è delineare un profilo di gruppo che rispecchi il più possibile le esigenze di tutti i partecipanti. Per farlo si è pensato di nominare una persona "capogruppo", alla quale dovranno sopraggiungere gli interessi di tutti i partecipanti. Naturalmente questa operazione non dovrà essere fatta manualmente; piuttosto si è pensato all'utilizzo della tecnologia NFC per il passaggio delle informazioni dai device dei partecipanti a quello del capogruppo. Innanzitutto ogni membro deve registrarsi all'applicazione seguendo la procedura

mostrata nel Capitolo 5. La differenza sta nel recupero delle informazioni di Facebook che viene fatta direttamente in questa fase; i dati prelevati saranno temporaneamente salvati nel database interno all'applicazione.

A questo punto il capogruppo utilizza la nuova funzionalità di "Create group" e il suo Smartphone diventa un *reader*. I device di tutti i partecipanti, funzionando in modalità passiva, possono essere letti e in questo modo trasferiscono le informazioni contenute nel loro profilo.

È compito dell'algoritmo sottostante aggiornare di volta in volta il profilo di gruppo sulla base delle nuove informazioni arrivate. In particolare:

- L'età del gruppo viene calcolata attraverso una media aritmetica rispetto a quella dei partecipanti;
- Il sesso del gruppo è costituito dalla percentuale di donne e di uomini presenti;
- Le preferenze vengono ordinate mettendo al primo posto quelle condivise da più utenti. Per farlo si associa ad ogni preferenza una variabile numerica, che viene incrementata ogni volta che la si ritrova nel profilo sopraggiunto. La lista desiderata si ottiene imponendo l'ordinamento decrescente rispetto a tale variabile.

Una volta costituito il profilo di gruppo, è possibile iniziare l'interazione; chiaramente solo il device del capogruppo viene abilitato a diventare il remote controller per navigare sul grande schermo.

3.2 Scenario 2: gruppi casuali

I gruppi casuali si formano nel momento in cui un insieme di persone si presentano di fronte al grande schermo, anche in istanti temporali differenti. Questo scenario ovviamente presenta un numero di ostacoli maggiore rispetto al precedente. Innanzitutto non si sa il momento preciso in cui il gruppo prende vita: può essere che un utente stia interagendo con il grande schermo e, ad un tratto, si avvicinino alcune persone con l'intenzione di partecipare all'esperienza.

In secondo luogo, gli individui non si conoscono, non c'è alcun rapporto affettivo che li lega; non è quindi possibile collaborare nella creazione di un profilo di gruppo.

Infine, come già anticipato, le persone possono giungere davanti al display pubblico in istanti temporali differenti; pertanto, non si riesce a prevedere il numero di partecipanti al gruppo.

Prima di mostrare le ipotesi avanzate dagli autori, si vuole sottolineare una distinzione sulla modalità di funzionamento del grande schermo. Si parla di "fruitore di contenuti" per indicare il momento iniziale in cui il display si comporta come un televisore, visualizzando un insieme di item. Si parla invece di "fase interattiva" per indicare la situazione in cui l'utente, utilizzando lo Smartphone come remote controller, naviga tra i contenuti mostrati a video.

Supponiamo che un utente si trovi di fronte al grande schermo e che, ad un certo punto, arrivi un individuo intenzionato a iniziare la comunicazione. Finché il display funziona come fruitore di contenuti, non ci sono problemi. Ogni persona intenzionata a partecipare all'esperienza, appoggia il device sul lettore NFC per farsi riconoscere e il display aggiorna la lista dei risultati tenendo conto delle preferenze del nuovo arrivato. In questo caso quindi è l'algoritmo stesso che crea/aggiorna il profilo di gruppo in base agli interessi di coloro che si sono fatti riconoscere tramite NFC.

I problemi nascono nel momento in cui il grande schermo si trova nella fase interattiva. Infatti, come prima cosa, il sopraggiungere di una nuova persona che appoggia lo Smartphone sul lettore NFC non deve interferire nell'interazione già avviata con un altro utente. A tal proposito, il grande schermo può recuperare immediatamente le informazioni del nuovo arrivato ma aggiornare la lista di raccomandazione una volta che torna a funzionare in modalità fruitore di contenuti.

In secondo luogo, vanno definite delle regole per gestire la fase interattiva nel momento in cui diverse persone si trovano di fronte al display pubblico. Secondo quanto ideato dagli autori, si possono intraprendere due differenti direzioni.

La prima fa sì che il grande schermo venga suddiviso in aree, in modo tale da permettere ad ogni utente di godere della propria User-Experience. È una soluzione valida solo se limitata a 2/3 persone, in quanto si devono dividere le risorse (hardware e software) lasciando un tempo di risposta dell'ordine di pochi secondi.

La seconda invece concentra la navigazione tra i contenuti nelle mani di un singolo utente, assegnandogli un tempo massimo pari a 3/4 minuti. Allo scadere del tempo, "il testimone" viene passato ad un altro partecipante secondo l'ordine di arrivo nel gruppo. Anche in questo caso però il numero di persone che possono prender parte allo scenario è limitato; infatti se un nuovo utente vuole partecipare all'esperienza ma si trova quinto nell'ordine di arrivo, non aspetta di certo un quarto d'ora.

In relazione al progetto di tesi, l'importanza della fase interattiva si ritrova nella possibilità di far conoscere al turista i ristoranti e i punti di interesse che maggiormente si adattano a lui. Questa funzionalità è tanto utile nel caso teorico di singolo utente di fronte al grande schermo, quanto dannosa nel momento in cui sopraggiunge un numero elevato di persone.

Un'idea potrebbe essere quella di limitare il display a funzionare come fruitore di contenuti, e trovare una valida alternativa in grado di rimpiazzare la fase interattiva. Gli autori hanno trovato risposta nell'inviare la lista di raccomandazione personalizzata (quindi non quella mostrata a video che è stata aggiornata in base alle preferenze di tutti) allo Smartphone di coloro che hanno effettuato il riconoscimento tramite NFC. La tecnologia utilizzata per l'invio della lista può spaziare dal Bluetooth, al WiFi Direct oppure ancora attraverso l'uso dei WebSockets. È una soluzione meno soddisfacente, ma più flessibile in quanto ipoteticamente permette a molti utenti di intervenire e inserirsi nel processo di raccomandazione.

MANUALE D'INSTALLAZIONE

Il manuale di installazione viene inserito per fornire agli sviluppatori gli strumenti necessari ad estendere il progetto di tesi. Si parte definendo quali sono i tool da utilizzare per andare a lavorare sui sorgenti dell'applicazione mobile. Si passa poi a descrivere la gestione della base di dati e, infine, vengono elencati gli step da seguire pedissequamente per permettere l'integrazione tra personal device e grande schermo. Per facilitare e velocizzare il lavoro agli sviluppatori, si riporta l'indirizzo Web per il download delle diverse componenti da installare.

Prima di entrare nel dettaglio della guida, si vuole spiegare brevemente il contenuto della cartella d'installazione, la quale presenta:

- La cartella *Expo* contenente il codice sorgente, le immagini, gli stili e tutto ciò che serve per il corretto funzionamento dell'applicazione mobile;
- La cartella *LocalServer* contenente le pagine PHP atte alla creazione del server locale;
- Il file *Tripadvisor.sql* che rappresenta il dump della base di dati presente sul Web Server;
- Il file *ItemsDisplay.sql* che fornisce la struttura delle tabelle da utilizzare per salvare gli item mostrati a video e gli eventuali dati recuperati da Facebook;

- La cartella *NodeJs* contenente:
 - *expo – server.js* che costituisce la parte server-side;
 - *expo.php* che costituisce la parte client-side;
 - *frontend.js* che contiene al suo interno la logica della pagina Web mostrata dal grande schermo;
 - Il video di presentazione mostrato dal grande schermo;
 - Altri file come Javascript, immagini, css e font utilizzati.

Per iniziare a lavorare sull'applicazione mobile, occorre:

1. Installare la versione corretta di Java

L'ambiente di sviluppo Eclipse, utilizzato per la scrittura del codice sorgente dell'applicazione, si basa su Java. Per prima cosa quindi si deve scaricare la versione corretta di Java che, in accordo con il progetto, deve essere la 7 o l'ultima versione della Java Development Kit (JDK). In questo modo si avrà una Java Runtime Environment (JRE) indispensabile per lanciare Eclipse.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2. Scaricare Eclipse

Con la versione corretta di Java sul pc, è possibile scaricare l'IDE di sviluppo Eclipse dedicato al linguaggio Java. Si può utilizzare "Eclipse Standard" o "Eclipse IDE for Java Developers" indifferentemente.

<http://www.eclipse.org/downloads>

3. Installazione ADT

A questo punto si accantona momentaneamente l'ambiente Eclipse, per andare a scaricare l'SDK di Android. Dalla pagina linkata sotto, selezionare *Use an Existing IDE* per scaricare solamente l'SDK dato che si è già in possesso dell'IDE di Eclipse.

Completato il download, decomprimere la cartella e aprirla per procedere con l'installazione. Una volta conclusa questa operazione, automaticamente

viene aperto l'SDK Manager nel quale sono presenti gli SDK (e le relative API) da quelli più datati fino all'ultima versione disponibile sul mercato. In accordo con i requisiti del progetto, selezionare tutte le versioni Android uguali e superiori alla 2.3.3 e, nella parte di *Extras*, spuntare "Android Support Repository" e "Google play services". Cliccare su *Install Selected* e attendere l'installazione di tutti i pacchetti che richiederà un po' di tempo.

Terminata questa procedura, lanciare Eclipse e dal menu di navigazione cliccare *Help* → *Install new software*. In questa finestra è possibile installare tutti i plug-in per Eclipse, non solo quello per sviluppare applicazioni Android. In ogni caso, nel campo *Work with* incollare il link "https://dl-ssl.google.com/android/eclipse/" e cliccare *Add*. È sufficiente seguire la procedura di installazione per inserire l'ADT all'interno di Eclipse.

<http://developer.android.com/sdk/index.html>

4. Importare il progetto

Per importare il progetto, dal menu di navigazione di Eclipse cliccare *File* → *Import*. A questo punto si aprirà una finestra, dalla quale selezionare *Android* → *Existing Android Code Into Workspace*. Cliccare su *Browse ...*, selezionare il progetto desiderato e infine fare *Finish*. A questo punto nel *Project Explorer* di Eclipse apparirà il progetto importato con tutti i sorgenti.

5. Testare l'applicazione

Per testare l'applicazione si può utilizzare uno Smartphone/Tablet Android con una versione superiore o uguale alla 2.3.3, oppure un emulatore messo a disposizione dall'ambiente di sviluppo. Dal menu di navigazione di Eclipse cliccare *Window* → *Android Virtual Device Manager*. A questo punto si aprirà una finestra, cliccare *New* per creare l'emulatore con le caratteristiche desiderate.

Per quanto riguarda la gestione del database, occorre:

1. Installare MySQL Workbench

Come spiegato dettagliatamente all'interno dell'elaborato, l'applicazione comunica costantemente con una base di dati presente sul Web Server. Per gestire comodamente la base di dati si consiglia l'utilizzo di MySQL Workbench, che offre strumenti importanti nella creazione, esecuzione e ottimizzazione delle query SQL.

Dalla pagina linkata sotto, andare nella sezione *Generally Available (GA) Releases* e scaricare il software. Terminato il download, procedere con l'installazione guidata.

<http://dev.mysql.com/downloads/tools/workbench/>

2. Importare la base di dati

Avviare MySQL Workbench e dalla schermata iniziale selezionare *Local instance MySQL55*. Per importare il database, dagli strumenti *MANAGEMENT* cliccare *Data Import/Restore*. A questo punto si aprirà una nuova finestra, scegliere *Import from Self – Contained File*, cliccare su ... e selezionare il dump del database che si desidera importare.

Per permettere l'integrazione tra personal device e grande schermo, occorre:

1. Installare EasyPHP

Come specificato all'interno dell'elaborato, il front-end del grande schermo si comporta come un Browser Web che visualizza i contenuti mediante pagine PHP. È indispensabile quindi dotarsi di un tool che permetta la scrittura/modifica di tali pagine.

Dal link sottostante, scaricare indifferentemente una versione di EasyPHP e seguire la procedura guidata di installazione. Una volta scaricato il programma, comparirà l'icona di EasyPHP nella barra delle applicazioni del calcolatore. Passando il mouse sopra di essa si può vedere lo stato del server, per capire se sta funzionando correttamente o meno.

Le pagine PHP realizzate devono essere salvate nella cartella “<percorso_di_installazione>/www”, e saranno successivamente visualizzate all’indirizzo <http://localhost/> oppure <http://127.0.0.1>.

<http://www.easyphp.org/download.php>

2. Installare Python

La creazione del canale WebSocket tra device personale e grande schermo è opera di Node.js, il quale per funzionare correttamente necessita la versione 2.6 o 2.7 del linguaggio di programmazione Python.

<https://www.python.org/download/releases/>

3. Installare Node.js

Scaricato Python, è possibile procedere con l’installazione di Node.js. Dalla pagina linkata sotto, scaricare l’installer e seguire le indicazioni riportate per completare la procedura.

<http://nodejs.org/download/>

4. Installare un editor di testo

L’editor di testo è indispensabile per creare/modificare le pagine PHP e le pagine Javascript. Vista la sua versatilità nel leggere diversi linguaggi di programmazione, gli autori suggeriscono di utilizzare Sublime Text disponibile al link seguente.

<http://www.sublimetext.com/>

A questo punto, dopo aver completato con successo i passi precedenti, si devono fare opportune configurazioni:

1. Impostare indirizzo IP

Per conoscere l’indirizzo IP associato al grande schermo (più precisamente del calcolatore che comanda il grande schermo), aprire il prompt dei comandi e digitare *ipconfig*. Tale indirizzo deve essere settato nella

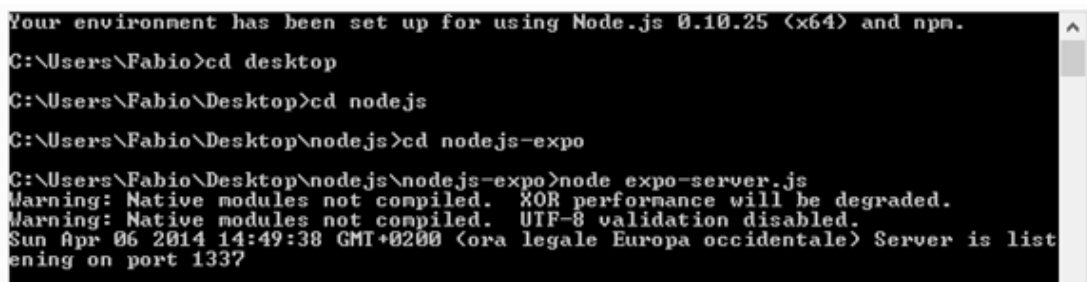
variabile "ip" del file *frontend.js* e all'interno del progetto mobile. Più precisamente, lanciare Eclipse e seguire il percorso *Expo* → *src* → *com.expo* → *IntroController.java*. All'interno della classe *IntroController* cercare il metodo *openWebSocket()* e modificare la variabile *wsuri* mettendo "ws://<newIndirizzoIP>:1337".

2. Lanciare il server locale

Per la creazione del server locale, lanciare il programma EasyPHP, controllando che funzioni correttamente attraverso l'icona presente nella barra delle applicazioni.

3. Costruire il canale di comunicazione

Per costruire il canale dedicato alla comunicazione tra personal device e grande schermo, bisogna aprire il terminale Node.js e spostarsi nella cartella *Nodejs* fornita dagli autori. Utilizzare quindi i comandi *cd* per arrivare nel punto desiderato e lanciare il comando *node expo - server.js*. A questo punto, se l'operazione è andata a buon fine apparirà il messaggio riportato in Figura 152.



```
Your environment has been set up for using Node.js 0.10.25 (x64) and npm.
C:\Users\Fabio>cd desktop
C:\Users\Fabio\Desktop>cd nodejs
C:\Users\Fabio\Desktop\nodejs>cd nodejs-expo
C:\Users\Fabio\Desktop\nodejs\nodejs-expo>node expo-server.js
Warning: Native modules not compiled. XOR performance will be degraded.
Warning: Native modules not compiled. UTF-8 validation disabled.
Sun Apr 06 2014 14:49:38 GMT+0200 (ora legale Europa occidentale) Server is list
ening on port 1337
```

Figura 152. Canale di comunicazione creato con successo.

4. Lanciare il front-end del grande schermo

Dopo aver completato correttamente gli step precedenti, aprire il browser e lanciare il file *expo.php*. A questo punto, il grande schermo mostrerà il video di presentazione contenuto nella cartella d'installazione fornita dagli autori.

5. Integrare Smartphone/Tablet con il grande schermo

Avviare l'applicazione sul dispositivo Android e, dopo essersi registrati o effettuato il login con Facebook, cliccare *Controller* dalla home. Se la comunicazione si instaura con successo, il grande schermo interrompe la visualizzazione del video di presentazione e mostra un messaggio di benvenuto all'utente.

Si può controllare l'esito dell'operazione di connessione direttamente osservando la console che, in caso di esito positivo, dovrà presentare il messaggio riportato in Figura 153.

```
Your environment has been set up for using Node.js 0.10.25 (x64) and npm.
C:\Users\Fabio>cd desktop
C:\Users\Fabio\Desktop>cd nodejs
C:\Users\Fabio\Desktop\nodejs>cd nodejs-expo
C:\Users\Fabio\Desktop\nodejs\nodejs-expo>node expo-server.js
Warning: Native modules not compiled. XOR performance will be degraded.
Warning: Native modules not compiled. UTF-8 validation disabled.
Sun Apr 06 2014 14:49:38 GMT+0200 (ora legale Europa occidentale) Server is list
ening on port 1337
Sun Apr 06 2014 15:20:00 GMT+0200 (ora legale Europa occidentale) Connection fro
m origin null
Sun Apr 06 2014 15:20:00 GMT+0200 (ora legale Europa occidentale) Connection acc
epted.
Sun Apr 06 2014 15:20:40 GMT+0200 (ora legale Europa occidentale) Connection fro
m origin undefined
Sun Apr 06 2014 15:20:40 GMT+0200 (ora legale Europa occidentale) Connection acc
epted.
Received Message: welcome:
Received Message: status:ok
```

Figura 153. Connessione avvenuta con successo.

Nel caso in cui la connessione non dovesse instaurarsi, il problema potrebbe essere dovuto a determinate regole restrittive inserite nel Firewall o nel Proxy. Si può quindi agire in due modi: impostare una delle porte rimaste libere (si può utilizzare anche la porta 80) o impostare il Port Forwarding del router.

Optando per la modifica del numero di porta, è importante aggiornare tutti i file interessati, nella fattispecie la variabile "connection" del file *frontend.js*, la variabile "webSocketsServerPort" del file *expo - server.js* e, infine, la variabile "connection" presente nella classe *IntroController* di Eclipse.

SOFTWARE UTILIZZATI

In questa sezione vengono elencati i software utilizzati nel lavoro di tesi. Per ognuno di essi, si fornisce una breve descrizione e si riporta la sua utilità in merito al prototipo realizzato.

A.1 StarUML

StarUML è una piattaforma open source utilizzata per lo sviluppo di diagrammi UML. È flessibile ed estensibile grazie alla sua architettura a plug-in e alla disponibilità delle apposite API.

All'interno della tesi è stato utilizzato per la composizione di Use Case Diagram, Sequence Diagram e Activity Diagram.

A.2 Microsoft Visio 2010

Microsoft Visio 2010 è uno strumento utilizzato per la creazione semplificata di diagrammi grazie ai suoi efficaci elementi visivi.

All'interno del progetto è stato utilizzato per la realizzazione del modello E/R e per la descrizione delle tabelle presenti nel database di TripAdvisor.

A.3 Eclipse + ADT (Android Development Tools)

Eclipse è un ambiente di sviluppo multiplatforma che abbraccia il mondo dell'open source. L'ADT è un plug-in per l'IDE di Eclipse che estende le sue funzionalità, permettendo lo sviluppo di progetti Android.

All'interno del lavoro è stato utilizzato per permettere il Web Scraping di TripAdvisor, e per la stesura del codice alla base del funzionamento dell'applicazione mobile.

A.4 Adobe Photoshop

Adobe Photoshop è un software proprietario prodotto dalla Adobe System Incorporated specializzato nell'elaborazione di fotografie (fotoritocco) e, più in generale, di immagini digitali.

All'interno del prototipo è stato utilizzato per la creazione dell'icona dell'applicazione mobile e per il ridimensionamento delle immagini presenti al suo interno.

A.5 Sublime Text

Sublime Text è un sofisticato editor di testo, conosciuto per la versatilità di lettura di diversi linguaggi di programmazione.

All'interno del progetto è stato utilizzato per la creazione delle pagine PHP e delle pagine Javascript.

A.6 FileZilla

FileZilla è un software gratuito che permette il trasferimento di file su un server attraverso il protocollo FTP.

All'interno del lavoro è stato utilizzato per caricare le pagine PHP sul server Altvista.

A.7 Fiddler

Fiddler è un “Web debugging proxy” che registra tutto il traffico HTTP tra il calcolatore e Internet. Si usa per l’intercettazione dei dati inviati e ricevuti da qualsiasi applicazione che supporta un proxy come Internet Explorer, Chrome, Safari, Firefox, Opera e altro ancora.

All’interno della tesi è stato utilizzato per il debug e il testing della parte lato server mediante la creazione di pacchetti HTTP personalizzati.

A.8 Node.js

Node.js è una piattaforma basata su Javascript Engine V8 di Google, il cui scopo è realizzare applicazioni di rete veloci e scalabili.

All’interno del progetto è stato utilizzato sia per la realizzazione della parte server-side (in grado di mettere in comunicazione il device personale con il grande schermo) che per la creazione della parte client-side (contenente la logica del front-end del display pubblico).

A.9 EasyPhp

EasyPHP è una piattaforma di sviluppo Web di tipo WAMP¹¹ basata sull’interprete PHP che consente al calcolatore di dotarsi delle funzionalità tipiche di un Web Server. Non è solo un software, ma un ambiente di sviluppo comprendente un server Web Apache, un server di database MySQL, un interprete di script PHP e un amministratore di database MYSQL con interfaccia grafica chiamata phpMyAdmin. Questi componenti sono installati insieme per avere in una sola volta tutto il necessario per iniziare lo sviluppo locale di siti Web in PHP.

¹¹ Windows Apache Mysql Php, indica una piattaforma software di sviluppo web/database che prende il nome delle iniziali dei componenti software con cui è realizzata.

All'interno del lavoro è stato utilizzato per la costituzione del server locale, nel quale sono presenti le pagine PHP che il calcolatore richiama tramite Ajax¹². Queste pagine vengono utilizzate per la creazione del front-end del display pubblico.

A.10 MySQL Workbench

MySQL Workbench è uno strumento visuale di progettazione per database che integra lo sviluppo SQL, la gestione e la modellazione dati, la creazione e la manutenzione di database MySQL.

All'interno del lavoro è stato utilizzato per la gestione della base di dati presente sul Web Server (invocata dall'applicazione per ottenere le info sui punti di interesse), e per la gestione della base di dati risiedente in locale sul calcolatore (interrogata dal grande schermo per ottenere la lista di risultati personalizzati).

A.11 Graph API Explorer

Graph API Explorer è un tool offerto gratuitamente da Facebook che permette di testare le richieste fatte tramite le API proprietarie.

All'interno del prototipo è stato utilizzato per il debug e il testing delle richieste effettuate dall'applicazione, per ottenere i likes e le locations presenti sul diario Facebook dell'utente.

¹² Asynchronous JavaScript and XML, è una tecnica di sviluppo software per la realizzazione di applicazioni web interattive.

RINGRAZIAMENTI

Prima di passare ai ringraziamenti personali, vogliamo mostrare la nostra gratitudine verso la Prof.ssa Franca Garzotto per il supporto nella realizzazione di questa tesi. Un sentito grazie va anche agli Ing. Luigi e Antonella per i loro consigli e per averci seguito nello sviluppo del progetto.

Ringraziamenti Elisabetta

Eccomi qui dopo tre anni a scrivere nuovamente i ringraziamenti, perché non bisogna mai dare per scontato ciò che gli altri fanno per noi, anche semplicemente con un sorriso.

Vorrei iniziare rivolgendo il pensiero al mio caro nonno Mario, scomparso pochi mesi fa e al quale è stata dedicata questa tesi. Avrei voluto completare il mio percorso con te al mio fianco, ma so che sei orgoglioso di me anche da lassù.

Ringrazio i miei genitori per essermi stati vicini, per avermi incoraggiato e sostenuto nelle mie scelte, per avermi permesso di studiare e di conseguire prima una laurea e poi una specializzazione. Se oggi io sono arrivata fin qui è anche grazie a voi!

Ringrazio mio fratello che, oltre ai genitori, è un punto di riferimento nella mia vita. So che nonostante il tuo dire scherzosamente “beh ancora stai lì a giocare con gli elicotteri?”, sei orgoglioso del traguardo che ho raggiunto. Grazie anche a te Terry per la persona che sei. Una concentrazione di simpatia, pazzia e dolcezza: sei la conferma che “nella botte piccola c’è il vino buono”.

Grazie zio Sandro per aver sempre creduto in me; mi hai sempre sostenuta dal primo giorno di università e la tua presenza è stata fondamentale in questo cammino.

Grazie a tutta la mia famiglia, in particolare i nonni che con il loro carisma mi hanno incoraggiata a non arrendermi mai e ai miei cugini Mario, Irene, Marco e Andrea con i quali sono cresciuta condividendo tante ma tante stupidate.

Una persona a cui devo tantissimo sei tu Fabio. Ti ringrazio perché mi ami da 5 anni, perché mi sostieni sempre e soprattutto perché mi capisci e asseconi tutte le mie pazzie. A volte nel litigare ci siamo detti frasi pesanti, ma le nostre giornate non sono mai finite con un muso, anzi era ancora più forte la voglia di abbracciarsi e di ricordarci quanto siamo importanti l'uno per l'altra. Abbiamo condiviso quasi tutto il nostro percorso universitario e ben due tesi di laurea: spero questo sia solo l'inizio di una lunga vita insieme. Grazie di essere al mio fianco e di avermi fatto conoscere il significato della parola amore.

Un sincero ringraziamento va alla mia migliore amica Raffaella, per avermi incoraggiata a combattere per ciò in cui valeva la pena lottare. Ogni volta che penso a te mi vengono in mente le merende speciali a casa mia, dove praticamente non potevi alzarti dalla sedia: ti ricordi il triplo panino big alla nutella, accompagnato da yogurt e brioche al cioccolato pucciata dentro? Scherzi a parte, sei una persona fondamentale, la mia roccia pronta a sorreggermi ad ogni fallimento, ad ogni caduta.

A te Nene riservo un grazie speciale, per tutte le volte che ti ho assillata con le mie insicurezze e i miei scleri post esame. La nostra complicità è unica, ci capiamo solo guardandoci negli occhi. Sei la mia unica pera cu ☺ E grazie anche a te Leo che con i tuoi giochi, i tuoi discorsi strampalati e il tuo volermi rinchiudere a tutti i costi nella tenda mi hai sempre fatto tornare il sorriso, sei fantastico!

Un ringraziamento particolare va ai miei stupendi amici Fede, Oggi, Vinci, Jasmin, Zac, Nico, Riccio e Lauret. La nostra amicizia è nata quasi per caso, quando Oggi mi ha portato una sera a conoscere la sua compagnia. Mi avete subito messa a mio agio, e ora eccoci qui: ogni volta che esco con voi il divertimento non è assicurato, di

più! Soprattutto quando mi fate bere e ve la sghignazzate alle mie spalle, visto che io non ricordo una parola di ciò che dico. Il mio rito di iniziazione all'alcool lo hanno voluto gli amici di Fabio, a cui vorrei dire che ora non vado più fuori con un Bayles. Sono passata a reggere quasi un invisibile: so che siete orgogliosi dei miei progressi e ve lo dimostrerò in occasione della festa di laurea!

Grazie alle mie amiche Bea, Noemi e Dani; alle superiori ci siamo frequentate poco vista la mia insana pazzia di voler fare due scuole contemporaneamente. Devo dire che abbiamo recuperato bene e spero riusciremo a rimanere sempre in contatto nonostante gli impegni di lavoro futuri.

Un ringraziamento speciale va alle mie Squinzy preferite: Fra e Marta. Ci siamo conosciute nei tre mesi di lavoro alle poste, e da lì è nata un'amicizia sincera e molto forte. Ma vi ricordate quanti "Seeenti Raffaele" ho detto in quei giorni? Secondo me li sogna di notte!

Per te Luca non basterebbero nemmeno le duecento pagine della tesi per ringraziarti di tutto quello che hai fatto per me e la mia famiglia. Le tue attenzioni, il tuo affetto e il tuo umorismo sono unici. E poi vuoi mettere tutti i muffin al cioccolato che mi facevi trovare dopo pranzo? Adesso svelo il segreto: è per merito degli zuccheri dei muffin che ho passato gli esami! Scherzi a parte, grazie di essere parte della mia vita e della persona straordinaria che sei.

Un ringraziamento sincero va a tutte le persone che ho incontrato durante la mia esperienza universitaria. Pool e Riky... beh che dire di voi... siete i miei sciocchi loserelli preferiti!! La nostra conversazione su Facebook è passata da "troviamoci a studiare" a "insultiamo lo struzzo Elisabetta"; per non dimenticare delle stupidate che scrivete sulla mia pagina!! *Siete piccolini di testa [cit.]*. Scherzi a parte, la vostra allegria è stata fondamentale in tutti quei momenti in cui l'unica cosa che avevo voglia di fare era lanciare libri e appunti direttamente in un camino. Spero di mantenere la nostra amicizia anche in futuro perché ci tengo.

Aleeee tu ovviamente non potevi mancare in queste pagine. Sei una persona speciale e super intelligente, ma questo lo sai già. Ti ammiro molto davvero, e spero di arrivare ad avere anche solo metà della tua conoscenza informatica. Grazie per

averci prestato il Tablet per testare il progetto (nel senso che in certi momenti lo avremmo proprio preso a testate!), ti meriti un super pranzo insieme a Pool che è stato il nostro guru di Eclipse!

Grazie anche a tutta la famiglia di Fabio, in particolare la mamma, la nonna, i fratelli. Mi avete sempre accolta in casa a braccia aperte, non facendomi sentire un semplice ospite anche quando rimanevo "accampata" per quasi una settimana intera. E parlando di ospitalità Katia sei la numero uno! Per non parlare poi dei tuoi aperitivi... quando hai fatto l'abbinamento crodino e pesche sciropate ti sei proprio superata! Ogni volta che prendo il treno sono contentissima all'idea che presto sarò lì a giocare con Emma, Riky e la piccola Elisa, ma soprattutto per stare con te. Sei una persona stupenda, sempre presente in ogni momento di debolezza e di felicità; avevi sempre la cosa giusta da dire e il tuo affetto arrivava anche da un semplice messaggio. Per tutto questo grazie di cuore, sei speciale.

A tutti i miei amici Alby, Gabry, Laura, Dani... (siete veramente tanti da elencare, ma non per questo meno importanti!) grazie per aver condiviso con me serate divertenti tra disco, bevute, Como (vero Alby?!).

Un ringraziamento particolare va anche a tutte le persone che ho incontrato nella mia esperienza universitaria Tania, le due Marti, Riky, Fra, Eva, i vari Stefano... è stato un piacere conoscervi, mi avete fatto ridere e stare bene soprattutto nei momenti di panico quando non uscivano i voti degli esami.

Per concludere, grazie a tutte le persone citate perché, in un modo o nell'altro, mi avete sostenuto nel raggiungere questo importante traguardo. Con il vostro affetto riuscite a rendermi ogni giorno una persona migliore, e per questo non esistono parole nel vocabolario in grado di ringraziarvi!

“Finché abbiamo ricordi, il passato dura... finché abbiamo speranze, il futuro ci attende... finché abbiamo amici, il presente vale la pena di essere vissuto. Non c'è vera felicità senza un amico con cui dividerla!”



Elisabetta

Ringraziamenti Fabio

Dopo sei anni di Politecnico sono giunto alla conclusione della mia carriera universitaria, fatta di sofferenze e gioie condivise con le persone che mi sono state vicine durante questo lungo e faticoso percorso. Vorrei innanzitutto ringraziare i miei genitori che grazie ai loro sacrifici mi hanno permesso di intraprendere questa strada cercando sempre di spronarmi e di tirarmi su quando magari andavano male gli esami. È grazie a loro se sono riuscito a raggiungere questo traguardo, in particolar modo a mia mamma che mi è stata accanto in tutto il lungo periodo universitario. Un grosso ringraziamento va anche ai miei zii Domenico e Tiziana che sono come dei secondi genitori, è grazie a loro se ho continuato a studiare e non mi sono fermato al primo anno quando ero indeciso se continuare o andare a lavorare perché volevo fare qualcosa di pratico. Entrambi mi hanno consigliato di tenere duro dicendomi che l'università forma la persona che sarai un domani, offrendoti la possibilità di avere un futuro migliore e che i sacrifici verranno ripagati. Ai miei

fratelli Marco e Alessandro che mi cercano sempre quando hanno bisogno di un consiglio o di un aiuto in ambito informatico, anche se mi hanno sempre preso in giro dicendomi che sarebbe ora di andare a lavorare per guadagnare so che alla fine sotto sotto, ma proprio in profondità, sono fieri di me anche se non lo ammetteranno mai. Ringrazio i miei nonni, in particolare quelli che non ci sono più poiché non potrò condividere questo momento insieme a loro anche se so che da lassù sono contenti di tutto quello che sto facendo, per l'unica ancora presente, mia nonna Carolina la ringrazio per avermi fatto sempre trovare quintali di cibo da mangiare a qualsiasi ora tornassi affermando che per studiare bisogna incamerare energie. A mio cugino Luca con cui ho sempre condiviso, dai momenti liberi alle partite a Pes insieme a mio fratello lo ringrazio per avermi fatto svagare quando magari avevi bisogno di staccare un po' la spina dopo ore di studio o meglio di programmazione. Una persona a cui devo molto è Elisabetta, ti ringrazio perché da 5 anni mi sopporti, non solo in ambito universitario ma anche affettivo. È anche grazie a te che sono arrivato a questo punto, dopo esserci fidanzati al secondo anno della triennale non so se sarei riuscito a dare 4 esami compresa la tesi in meno di 5 mesi durante la laurea breve. So benissimo di essere molte volte testardo e per questo ti ringrazio che sei ancora insieme a me anche se ho un carattere particolare che non tutte le ragazze accetterebbero. Ringrazio inoltre tutta la sua famiglia, dai nonni ai cugini che mi hanno accettato e sopportato per tutte le volte che mi sono fermato da lei per studiare o solamente per trascorrere del tempo insieme. Grazie a lei ho anche conosciuto un sacco di amici veri e sinceri con cui ho trascorso parecchie serate tra chiacchiere e alcool, i primi che devo citare insieme alla parola alcool non possono che essere Vinci (detto vino, il che la dice lunga) e Fede, ma non mi dimentico di Oggi, Zac, Nene, Riccio, Lauret, Jasmine, Nico, Albi e Raffa. Non posso non ringraziare i miei amici di sempre con cui ho praticamente trascorso tutti i momenti liberi dall'adolescenza fino ad adesso e con cui ho fatto i viaggi più marci ma anche i più divertenti. Teuz con cui sono andato a gioire e piangere allo stadio San Siro per vedere la nostra magica Inter, Corra con cui ho trascorso parecchio tempo tra partite di poker (giocare in casa sua è come giocare contro la Juve di

calciopoli, impossibile battere lui o suo papà), calcio e soprattutto bevute, Neits e Zappi con cui ho trascorso e trascorrerò parecchie serate al nostro solito vecchio 22100; quando sono in coppia il divertimento è assicurato e non ci si può annoiare.

Trinz che è l'unico del "gruppo tesserete" che non mi ha abbandonato al mio triste destino ma che ha cercato di venire a recuperarmi, anche se con scarsi risultati; uscire con lui è sempre un pericolo, si parte sempre con una serata tranquilla e si finisce con il fegato che chiede aiuto.

Un ringraziamento un po' meno sentito va a Grida, per colpa di una sua scommessa sulla Juve ho dovuto tagliarmi i capelli cortissimi (mai fatto in vita mia); a parte gli scherzi ho sempre potuto contare su di lui per una cigs in compagnia o per bere una birra e fare due chiacchiere.

Ringrazio inoltre Stux, Boch, Ale che anche se adesso li vedo un po' meno, è sempre un piacere rivederli e fare serata con loro in ricordo delle vecchie vacanze madide. Il ringraziamento più importante nel gruppo dei miei amici va sicuramente a Pelle, ci conosciamo ormai da 25 anni, direi un po' troppi per poterlo sopportare ancora; scherzo ovviamente, è la persona con cui ho trascorso più tempo tra calcio, superiori, uscite, bevute, e tutto il resto. È sempre stato presente per darmi supporto morale, o solamente per dire qualche cazzata; anche se adesso non ci vediamo spesso come prima, ogni volta succede sempre qualcosa di particolarmente stupido che rende la serata indimenticabile. Uscire con ognuno di loro è impossibile annoiarsi, quando poi si riesce a riunirsi tutti insieme succedono cose indescrivibili che è meglio non raccontare; per questo non li cambierei con nessun altro gruppo di amici.

Dopo gli amici acquisiti, quelli di lunga data, non posso non ringraziare quelli che hanno trascorso con me gli esami, le ore infinite di studio in segreteria, i mille caffè bevuti. Per primo non posso che citare Paul, la persona che ho conosciuto il primo giorno di università e che è stata con me fino a questo ultimo giorno, con lui ho condiviso il primo progetto pratico in Java (il bellissimo crossfire) e trascorso le ore a imprecare dietro a Jboss e Eclipse per i mille problemi che comportano. L'unica

nostra pecca è non aver vinto abbastanza alle scommesse, complice una sfiga che ci vede benissimo, per aprire una startup con un nostro amico segreto.

Ricky che ho conosciuto meglio nella magistrale, ma che sarebbe stato meglio conoscerlo prima per la bella persona che è, anche se un quasi ingegnere come lui, dato che a breve si laurea non può non essere capace di abbassare una veneziana; se non ci credete ho dei video che lo dimostrano.

Luca che ha trascorso con me il calvario del mio ultimo esame del laboratorio di LFC per ben tre volte, è stato veramente gentile in qualsiasi situazione, se avevi bisogno di un qualsiasi tipo di aiuto era sempre presente; l'unica sua pecca è quella di arrivare costantemente in ritardo e andare in bagno proprio nel momento in cui noi tutti dovevamo andare a prendere il treno (ovviamente per i continui ritardi, scioperi e soppressioni Trenord non viene inserita nei miei ringraziamenti).

Vorrei ringraziare tutti i miei compagni di università come Gerry, Davide, Martina, Ricky, Ale che ci ha prestato il suo Tablet e senza di esso non sarei qui ora a scrivere, Ste e tanti altri.

Quasi sicuramente, a causa della mia poca memoria, mi sarò dimenticato qualche persona importante, ma vorrei comunque dire grazie a tutte le persone che mi sono state vicino anche solo con un piccolo gesto o che hanno trascorso con me anche poco tempo perché l'importante non è il cammino che si compie ma è le persone con cui hai lo hai condiviso che lo rendono unico.

Fabio

REFERENCES

- [1] Burke R.: *Hybrid recommender systems: Survey and experiments*. User Modeling and User-Adapted Interaction 12(4), pp. 331–370 (2002).
- [2] Baltrunas L., Makcinskas T., Ricci F.: *Group recommendations with rank aggregation and collaborative filtering*. In: Proceedings of the fourth ACM conference on Recommender systems, vol. 2, pp. 119–126 (2010).
- [3] Boratto L.: *Group Recommendation With Automatic Detection and Classification of Groups*. Tesi di dottorato, Cagliari (2011).
- [4] Chen Y., Pu P.: *CoFeel: Emotional Social Interface in Group Recommender Systems*. In: Proceeding of RecSys'12 Workshop on Interfaces for Recommender Systems, in conjunction with the 6th ACM Conference on Recommender Systems (RecSys'12), Dublino, Irlanda, (2012).
- [5] Ricci F.: *Mobile Recommender Systems*. In: Proceedings of the First Italian Information Retrieval Workshop, Padova, Italia, pp. 71–75 (2010).
- [6] Schiller J. H.: *Mobile Communications*. Addison-Wesley (2003).
- [7] Gavalas D., Konstantopoulos C., Mastakas K., Pantziou G.: *Mobile recommender systems in tourism*. In: Journal of Network and Computer Applications (2013).
- [8] Nuno L., Anacleto R., de Almeida A.: *Tourism mobile and recommendations systems – a state of the art*. In: CSREA EEE, pp. . 277–283 (2010).
- [9] Baltrunas L., Ludwig B., Peer S., Ricci F.: *Context relevance assessment and exploitation in mobile recommender systems*. In: Personal and Ubiquitous Computing, vol. 16(5), pp. 507–526 (2012).

- [10] Modsching M., Kramer R., Hagen K., Gretzel U.: *Effectiveness of Mobile Recommender Systems for Tourist Destinations: A User Evaluation*. In: Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (2007).
- [11] Jawaheer G.: *Comparison of implicit and explicit feedback from an online music recommendation service*. Presented at the proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems. Barcelona, Spain (2010).
- [12] Claypool M., Brown D.: *Curious Browsers: automated gathering of implicit interest indicators by an instrumented browser*. Computer Science Department – WPI, USA (2003).
- [13] Joachims T., Granka L., Pan B., Hembrooke H., Gay G.: *Accurately interpreting clickthrough data as implicit feedback*. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, USA, pp. 154–161 (2005).
- [14] Moling O., Baltrunas L., Ricci F.: *Optimal radio channel recommendations with explicit and implicit feedback*. In: Proceedings of the sixth ACM conference on Recommender systems, USA, pp. 75–82 (2012).
- [15] Zaslow J.: *Oh no! My TiVo thinks I'm gay*. In: Wall Street Journal (2002).
- [16] Pu P., Chen L.: *Trust building with explanation interfaces*. In: IUI'06, Recommendations I, pp. 93–100 (2006).
- [17] Cramer H., Evers V., Someren M.V., Ramlal S., Rutledge L., Stash N., Aroyo L., Wielinga B.: *The effects of transparency on perceived and actual competence of a content-based recommender*. In: Semantic Web User Interaction Workshop, CHI (2008).
- [18] Czarkowski M.: *A scrutable adaptive hypertext*. Ph.D. thesis, University of Sydney (2006).
- [19] Muller J., Alt F., Michelis D., Schmidt A.: *Requirements and design space for interactive public displays*. In: Proceeding of the International Conference on Multimedia, New York, USA, pp. 1285 – 1294 (2010).

- [20] Kuikkaniemi K., Jacucci G., Turpeinen M., Hoggan E. E., Muller J.: *From Space to Stage: How Interactive Screens Will Change Urban Life*. In: IEEE Computer 44 (6), Los Alamitos, USA, pp. 40 – 47 (2011).
- [21] Coutrix C., Kuikkaniemi K., Kurvinen E., Jacucci G., Avdouevski I., Makela R., *FizzyVis: Designing for Playful Information Browsing on a Multitouch Public Display*. In: Proceeding of the 2011 Conference on Designing Pleasurable Products and Interfaces, New York, USA, art. 27 (2011).
- [22] Maunder A., Marsden G., Harper R.: *Creating and sharing multi-media packages using large situated public displays and mobile phones*. In: Proceedings of the 9th international conference on Human computer interaction with mobile devices and services, New York, USA, pp. 222 – 225 (2007).
- [23] Anzures H., Mendoza S.: *Multi-user Interaction with Public Screens Using Mobile Devices*. In: Electrical Engineering Computing Science and Automatic Control, pp. 1 – 5 (2011).
- [24] Choi J. H.-J., Seeburger J.: *Sapporo World Window urban interaction through public and private screens*. In: Pervasive Computing and Communications Workshops, pp. 508 – 512 (2011).