POLITECNICO DI MILANO
Scuola di Ingegneria dell'Informazione (V)
Corso di laurea specialistica in Ingegneria Informatica
anno accademico 2009/2010

"WakeUp!": M-Learning to Stimulate Student Engagement in the Classroom.

Tesi di laurea di
EMANUELE VULCANO
matr. 725196

Relatore: Prof. FRANCA GARZOTTO

Ai miei genitori
che mi hanno aspettato.

Ai miei amici
che mi hanno sopportato.

Ai professori
che mi hanno dato una chance.

A tutti coloro
che mi hanno aiutato.

Ma soprattutto
agli antenati etiopi
dell'odierno popolo Oromo,
i primi a riconoscere le proprietà
del chicco tostato di caffè.

Ringrazio per la enorme disponibilità
la prof. Garzotto, il prof. Cremonesi e il dott. Turrin,
i docenti del Dipartimento di Elettronica e Informazione
e tutti gli studenti che si sono prestati
alla sperimentazione del nostro prototipo.

Table of Contents

Index of Figures

Index of Tables

Index of Graphs

Abstract

Mobile technology has imposed itself in recent years as a transformative force. It's driving the growth of the IT industry, and affecting all those areas of everyday life where some form of communication is important, including the teaching and learning process. This holds especially true in college- or university-level courses of technically-oriented schools such as our own, the Politecnico di Milano.

Despite this, recent innovations in “mobile learning” (or “m-learning” for short) haven't however yet trickled down to students and teachers to become commonplace, even with programmable smartphone more and more often being a fixture for students of all faculties. In this thesis, we'll document an attempt to bring a mobile technological element to courses in our campus, to assist students in everyday life and improve both their engagement within the classroom and their performance outside of it; and how the system was designed with a process that incorporates real-life user testing as an extremely important, iterated step.

This document describes in depth how the resulting distributed system works. Our prototype allows teachers to “broadcast” the lesson to students' mobile phones; students can interact with the teacher by asking questions or expressing their moods. All data is saved both on a web-accessible backend, allowing teachers to answer questions even after the lesson is over and use the data to improve their lectures, and on the students' mobile devices, so that it's always available whenever and wherever they choose to review and study. We will show how the in-class, real-life simulations that make up our user testing were performed, and how student feedback shaped the final prototype and our envisioning of future, better versions of the application.

Estratto (in lingua italiana)

Abstract (in lingua italiana)

La tecnologia mobile si è imposta, in anni recenti, come una fonte di innovazione in grado sia di divenire forza motrice della crescita dell'industria informatica, sia di trasformare la vita di tutti i giorni in ogni ambito in cui la comunicazione è importante, incluso quello dell'insegnamento e dell'apprendimento. Questo è vero in special modo nei corsi universitari di istituzioni a orientamento tecnico, tra le quali è il Politecnico di Milano.

Nonostante ciò, le recenti innovazioni del campo del "mobile learning" (o in breve "m-learning") non sono ancora comunemente disponibili a studenti e docenti; eppure sono sempre più diffusi tra gli studenti smartphone che possono essere programmati allo scopo. In questa tesi, documentiamo un tentativo di inserimento di un elemento tecnologico mobile all'interno dei corsi del nostro campus, in grado di assistere gli studenti nella loro vita quotidiana e migliorare sia il loro coinvolgimento all'interno dell'aula che l'efficacia del loro studio al di fuori. Mostreremo pure come abbiamo incorporato nel nostro processo di sviluppo sessioni frequenti di valutazione con utenti reali.

Questo documento descrive in profondità il funzionamento del sistema distribuito prodotto. Il nostro prototipo consente agli insegnanti di "inviare" la lezione ai cellulari degli studenti, e attraverso questi gli studenti possono interagire inviando domande o esprimendo il loro stato d'animo. I dati sono salvati in un *backend* accessibile attraverso un servizio web, consentendo agli insegnanti di rispondere alle domande anche dopo il termine della lezione; sono inoltre salvati localmente sui cellulari degli studenti, consentendo loro in questo modo di accedervi ovunque e in qualunque momento desiderino rivederli per studiare. Infine, mostreremo come abbiamo condotto le simulazioni reali in aula che sono elemento portante del nostro programma di valutazione, e come il *feedback* degli studenti abbia influenzato profondamente sia il prototipo finale che la nostra visione di versioni future e ancora più utili dell'applicativo.

Una introduzione al progetto

La tecnologia mobile, in special modo quella cellulare, ha fatto passi da gigante negli ultimi decenni. Grazie ad essa, l'informatica ha potuto raggiungere una fetta notevole della popolazione mondiale, e questo ha avuto enormi effetti sulla comunicazione tra persone e, quindi, direttamente o indirettamente, su gran parte degli aspetti della vita quotidiana. Non è quindi una sorpresa vedere come ci sia grande fermento nel campo della ricerca sulle applicazioni della tecnologia mobile sull'apprendimento — un campo spesso indicato con la dicitura "mobile learning" o "m-learning". Nonostante l'interesse, però, la ricerca è ancora agli inizi, e sono pochi i sistemi di m-learning adottati estensivamente dalle istituzioni scolastiche.

Nemmeno nella nostra università, il Politecnico di Milano, sono stati ufficialmente adottati con continuità servizi di supporto allo studio che coinvolgano massivamente l'uso di tecnologie mobili, e questo nonostante la popolazione studentesca del Politecnico, come ci si può aspettare da una università che fa dell'applicazione delle tecnologie il suo obiettivo principale, sia sensibile alle tematiche relative alle novità tecniche.

Si può ragionevolmente supporre che un servizio di questa natura possa essere offerto con profitto nella nostra università, permettendo così agli studenti di sfruttare le caratteristiche di disponibilità continua, trasportabilità e connettività dei propri smartphone a supporto del loro studio. Questo documento descrive design, implementazione e valutazione di un prototipo di questo genere di servizio, che ha ambizione di aiutare lo studente dentro e fuori dall'aula scolastica nel suo percorso formativo.

Il campo dell''m-learning' è molto ampio, e copre un gran numero di possibilità applicative, che vanno dalle guide per museo sensibili alla posizione, a strumenti di gioco per guidare i ragazzi in percorsi didattici, fino a strumenti "logistici" in grado di indicare allo studente orario e posizione delle aule. Il nostro interesse però si è focalizzato su un particolare sottoinsieme di applicazioni nelle quali il dispositivo cellulare è stato usato per aumentare e migliorare la comunicazione e l'interazione in aula tra il docente e lo studente.

Design e Prototipo

Il sistema che abbiamo progettato è stato sviluppato a partire dalla considerazione che molto spesso i docenti sono soliti utilizzare durante la loro lezione la proiezione di presentazioni (blocchi di diapositive). Questa tipologia di materiale si presenta particolarmente adatta ad essere manipolata per la fruizione su un dispositivo mobile, poiché spesso le diapositive (*slide*) sono riconducibili a una struttura a lista puntata (*outline*). Inoltre, l'insieme di tutte le presentazioni prodotte da un docente a sostegno delle proprie lezioni copre spesso l'intero programma di una materia, seppur in forma di riassunto, ed è già disponibile nei corsi di quei docenti che ne fanno uso.

Le presentazioni diventano una infrastruttura che mettiamo alla base della comunicazione tra docente e studente. Grazie ad esse possiamo infatti contestualizzare tale comunicazione: ad esempio, associando una domanda effettuata al punto a cui si riferisce, è possibile inferire quali siano i punti più problematici, informazione questa che è d'aiuto sia ai docenti, che possono migliorare le proprie lezioni, sia agli studenti, che possono così scorrere la presentazione durante lo studio vedendo immediatamente quali punti siano i più interessanti da rivedere.

Il prototipo che abbiamo costruito sfrutta la piattaforma Apple iOS per offrire i propri servizi su dispositivi iPhone e iPod touch, dotati di touchscreen. Lo studente può aprire questa applicazione in aula per "seguire" la lezione: essa rimane infatti sincronizzata con quanto accade sul computer del docente che proietta la presentazione, mostrando sotto forma di lista puntata (*outline*) la stessa diapositiva che viene proiettata in quel momento (figura 4.3). Lo studente può toccare uno qualunque dei punti di cui si compone la diapositiva per fare una domanda relativa a quest'ultimo. Egli può inoltre, in qualsiasi momento, esprimere un *mood* (stato d'animo) che viene sempre associato al contesto corrente. Le domande effettuate vengono condivise sia con il docente che con gli altri studenti in aula, e appaiono sotto forma di indicatori a margine del punto indicato. Per evitare distrazioni, sono disponibili una giusta quantità di *domande predefinite* molto comuni (come ad esempio "Questo punto non è chiaro" oppure "Può approfondire questo punto?") che lo studente può porre al docente con un solo tocco dello schermo.

Lo studente, ruotando il dispositivo in posizione orizzontale, può passare dalla modalità di "visualizzazione *outline*" a una modalità di "visualizzazione estesa" in cui viene mostrata l'immagine della *slide* senza alterazioni, alla stesso modo in cui essa appare nella proiezione del docente. Anche in questa modalità rimangono disponibili sullo schermo controlli che permettono di fare domande relative alla *slide* visualizzata o esprimere stati d'animo. Infine, sia in questa modalità "estesa" che in quella verticale (*outline*) è possibile utilizzare i controlli a schermo o dei gesti di "spinta" a destra o a sinistra per sfogliare la presentazione indipendentemente dal procedere della lezione, funzionalità utile nel caso lo studente abbia "perso" una parte della spiegazione per un qualunque motivo. Mentre lo studente sfoglia la presentazione, la sincronizzazione con la lezione in corso viene sospesa, permettendogli di rileggere con calma. Lo studente può comunque tornare alla diapositiva corrente con un tocco su un apposito tasto di "*fast forward*", riprendendo a seguire la lezione.

Terminata la lezione, lo studente porterà via con sé la presentazione, insieme a tutte le domande poste dagli studenti, scaricate localmente sul suo dispositivo. Questo gli permetterà di fruirla anche senza un collegamento a Internet (che possiamo garantire soltanto all'interno delle aule del Politecnico, grazie al servizio Wi-Fi offerto agli studenti). In tal modo potrà rivedere quanto appreso, incluso l'arricchimento delle domande poste, quando e dove desidera, senza essere costretto a ricercare il materiale sul sito del docente.

Il professore non ha necessità di effettuare lavoro addizionale per permettere al sistema di inviare le presentazioni; basterà installare una seconda applicazione da noi prodotta sul computer utilizzato per proiettare le diapositive. Tale applicazione effettua il monitoraggio del programma di proiezione e automaticamente effettua upload e distribuzione della presentazione agli studenti. L'applicazione visualizza anche una schermata di monitoraggio della lezione, dove il docente ha accesso in tempo reale a tutte le domande poste e agli stati d’animo espressi dagli studenti, raggruppati per diapositiva. Può accedere, inoltre, ad un servizio che gli permette di pubblicare le risposte alle domande effettuate dagli studenti anche dopo il termine della lezione, e, infine, a una schermata di statistiche dove sono visualizzate “a colpo d’occhio” le *slide* più commentate della lezione.

Valutazione e Conclusioni

Il prototipo da noi prodotto è stato valutato sia informalmente, mostrandolo ad alcuni docenti, sia più formalmente attraverso simulazioni in aula. Durante queste simulazioni il nostro sistema è stato utilizzato da docenti e studenti in un certo numero di lezioni reali, durante le quali tutti gli attori coinvolti sono stati sottoposti a osservazione e poi intervistati mediante un questionario per sondarne le opinioni. Gli studenti si sono dimostrati entusiasti del sistema, pure quando problemi tecnici ne hanno impedito il perfetto funzionamento, anche se comunque hanno dato segnali di gradire solo alcune delle funzionalità e hanno espresso dubbi rispetto alle possibilità di distrazione. Anche i docenti hanno espresso le proprie opinioni, specie per quanto riguarda la differenziazione di questo sistema rispetto a esperimenti simili precedenti e la sua integrazione con sistemi di gestione della didattica (Learning Management Systems, LMS) già in essere.

Le valutazioni sono state integrate nel nostro processo di sviluppo. Grazie a una politica di brevi iterazioni, con fasi di design e implementazione seguite rapidamente da una valutazione, siamo stati in grado di migliorare il sistema rapidamente anche in presenza di obiettivi generici e una conoscenza del dominio applicativo non perfettamente completa.

In conclusione, siamo rimasti notevolmente soddisfatti dei risultati raggiunti con questo prototipo, pur trattandosi di un’implementazione solo parziale delle nostre ambizioni iniziali. Abbiamo infine progettato i prossimi passi, scegliendo di migliorare notevolmente alcuni aspetti di questa applicazione, specialmente per quanto riguarda l’accesso alle presentazioni anche fuori dall’aula, la comunicazione tra docente e studenti dopo il termine della lezione e la possibilità per gli studenti di annotare il materiale con note personali, non condivise con gli altri come lo sono invece domande e stati d’animo. Questo è, appunto, solo un passo sulla strada che vogliamo percorrere per rendere il nostro sistema più utile senza sacrificare la *user experience* di studenti e docenti.

1. Introduction

As any student of this university can attest, learning can be quite hard. Pedagogy is often considered to be one of the most important fields of study in sociology, especially since the quality of education plays a pivotal role in managing and making sense of a world of rising complexity; however, even though there are pedagogical methods that are widely employed and are proven to yield good results, there is no “silver bullet” for improving education — the act of learning remains a highly personal experience. Even if the student is exposed to the course material in conventionally proven ways, each student still has to memorize, internalize and comprehend the subject matter in their own, highly personal way.

Things become even more complex as we stray from the beaten path to introduce innovative elements. The pervasive use of computing technology in education is a field still in its infancy, lagging behind the widespread adoption enjoyed by the same technology in everyday life. Use of technology in education can range from the simple replacement of certain traditional tools with software (e.g. using a word processor instead of pen and paper), to the integration of computers in the learning environment (in laboratories or classrooms), all the way up to replacement of the classroom itself with a computer and a communication network (in online learning systems).

Yet the use of computers as tools, while often useful, has a profoundly different psychological impact. Desktop computers, in particular, have a tendency to *replace*, rather then enrich, other tools; and due to their sheer size, interactivity and more, computers *demand attention* from the user, sometimes stealing it from the process at hand [1].

However, technology hasn’t stopped at the personal computer paradigm; indeed, *mobile* technology is becoming more and more popular, as cell phones turn from simple embedded systems to full-fledged computing platforms. In 2010, the number of global mobile phone connections surpassed 5 billion lines [2], and a growing fraction of these phones has computing capabilities on par with low-end desktop or laptop computers, along with unsurpassed mobility and potential access to a communication network at basically any time. The potential impact of this trend has not been ignored among education researchers, and indeed much attention has been given to the nascent field of “mobile learning”, or “m-learning”.

“M-learning” is a blanket term that covers a number of philosophies and application of technologies to the process of learning, all revolving around the concept of “mobility” — not just of the technology, but of the learner, too, both in space (learning anywhere) and in time (learning anytime). Indeed, a large portion of the research goes into ensuring this is a viable option for *lifelong learning*, that is, acquisition of skills even while engaged in some other activity, such as work, that would not let the learner follow a traditional classroom-based course. However, just like all other learning technologies and paradigms, m-learning can also apply to higher education, where its role may move from that of a centerpiece to that of a logistical support, helping students make the most out of the classroom experience by providing them the tools to review, study and check their progress in an assisted fashion at any time and in any place.

Technological progress is always felt first at technical universities. Our *alma mater*, the Politecnico di Milano, like most other campuses around the world, does have several initiatives that fall under the umbrella of “e-learning”. The campus offers service sites to its students, allowing them, for instance, to receive notices from their respective faculties; to find out the schedule for any course provided by the university, or obtain free time charts for every classroom; to show a map to lead the user to any classroom or laboratory; to let a student manage his career, including study plan presentation and vetting, exam enrollment and evaluation delivery, and all that is related to the thesis and final exam, entirely online. The campus even offers a degree course entirely online, allowing access to people who cannot attend lessons at the university structures in Milan, in Como or abroad.

In addition to the above, several services are available to users that directly pertain to didactic work, rather than university logistics. For example, the CorsiOnLine service allows users to access professor-managed areas with material downloads and communication facilities (such as forums) for those courses that use the platform. Additionally, faculties often provide their own centralized services, and professors, especially in the Engineering department, may set up their own communication boards on Politecnico servers, tailoring them to their needs.

Despite such an abundance of services, and the Politecnico also being involved in the advancement of mobile technology through partnerships with industry firms, there is a dearth of mobile-aware services. Recently, the Poliself self-service website, arguably the most important logistic service of the platform (handling exams, student careers, *etc.*) launched a limited version of the site tailored for mobile devices. However, aside from this and other isolated attempts like the Calepoli J2ME-based schedule and calendar organizer, the Politecnico has shied away from implementing services exclusively targeted at mobile phones.

It is no surprise, then, that no m-learning system is available as a centralized service from the university. This occurs despite the tremendous uptake of mobile technology, and smartphones in particular, is often felt in a more pronounced way in technical universities like our own — nowadays, students often have a mobile device that can be programmed and can connect to a data network, be it a cellular network or Politecnico’s ubiquitous free-to-students Wi-Fi service.

It is in this context that we believed we could add value by investigating this line of work. Having experience from other mobile technology projects, we have observed first-hand the benefits that this particular innovation provides in the areas of ease of communication, availability and simplification of the user-device interaction model; mobile technology is less of a barrier and more of an enabler when compared with the traditional desktop model. We therefore resolved to try to make these advantages available to Politecnico students, approaching this project in a *novel* way when compared to most other m-learning projects that can be observed in literature: where most other projects try to focus on a pedagogical theory basis, or propose models that require heavy involvement of all parties to change regular habits to adapt them to the new system, we instead wanted a system that would require *minimal* adaptation from all parties involved. That does not mean we overlooked the pedagogical effects of such a system in favor of other factors; on the contrary, by allowing proven methods to keep applying in the classroom — ensuring the experience is richer, without sacrificing any proven methodology — we allow the system to be *accepted* easily from all parties, and then *augment* traditional methods to streamline them or, once we are sure mobile technology does actually positively impact the study process, experiment with larger changes in the long term.

The objective of the project described by this document was, then:

- to *imagine* how such a system could possibly be introduced in Politecnico classrooms;
- to *design* it in such a manner that it could be implemented with minimal disruption of current services, and provide *real,* immediate value to all those using it;
- to *build* a prototype of the service, and
- to *evaluate* it to see how well it performs in regards to usefulness and user satisfaction.

The prototype system we built allows teachers to “broadcast” the lesson to students' mobile phones; students can interact with the teacher by asking questions or expressing their moods. All data is saved both on a web-accessible backend, allowing teachers to answer questions even after the lesson is over and use the data to improve their lectures, and on the students' mobile devices, so that it's always available whenever and wherever they choose to review and study. We will show how the in-class, real-life simulations that make up our user testing were performed, and how student feedback shaped the final prototype and our envisioning of future, better versions of the application.

As mentioned, we chose to emphasize the human-computer interaction side of this project, focusing on making the system *acceptable* for use by students, even at the cost of initially providing just a subset of our intended functionality. Our focus on human-computer interaction is a necessity if we look at the evolving context of mobile software in recent years; usability has become a differentiator and a key to success on the smartphone market, allowing Apple, whose smartphone has been critically praised mainly for its attention to interaction detail, to capture 51% of the profits of the entire market in 2010 [0]. If the market is any indicator, users now expect user experiences on mobile phones to be refined, a clear signal for any application that aims for acceptance, such as ours.

Another reason for this “small-steps” approach was our limited resources, especially in regards to manpower. This project’s main implementation phase, which included several evaluations of the prototype, ran from November 2010 to early January 2011, with a single person working on all aspects of a system that, while conceptually simple, has a number of components running on several different platforms (Apple iOS and Mac OS X, Microsoft Windows and the Google App Engine Python-based infrastructure on the web). By defining our work this way, we could deliver tangible, quantifiable results in two months from conception to finished prototype, a system that:

- handles material delivery seamlessly without requiring user interaction, and ensures the latest study material is always available on the student’s mobile device;
- provides streamlined, distractionless student-to-teacher “backtalk” communication during a lesson, which is captured to be made available alongside, and enrich, the study material above;
- contextualizes captured interaction so that inferences can potentially be made by the system, helping the student spot “at a glance” what are the most important things to review;
- provides a teacher with data in structured form to aid in reviewing and improving lectures;
- has a conceptually strong and extensible technological foundation that allows for future work to easily continue from the final prototype on.

In addition to building the prototype, we also envisioned how it would grow, detailing the medium and long term challenges we think the system will have to solve if it is to meet our objectives: an always-available study aid that we hope could, some day, assist students in every moment of their Politecnico careers.

In the following chapters, we will see how the system was conceived, what we chose to implement in our initial design, *how* we implemented what we chose, and how we evaluated our implementation in a real-world context. In particular:

- Chapter 2, **Background and Related Work**, shows how this system compares to other m-learning systems discussed in literature, especially in the area of in-class interaction and material delivery.

- Chapter 3, **Eliciting Requirements**, provides the results of our search for "starting points" to build our system on, discussing the results of an open Internet questionnaire distributed in Fall 2010 to students, both within and outside the Politecnico.
- Chapter 4, **"WakeUp!" Design**, discusses what grew from these starting points and what design process we chose to follow, and shows how the final prototype works from the perspective of students and teachers.
- Chapter 5, **Enabling Technology**, examines the actual implementation of the system, highlighting how we solved technological problems and how we designed the system so as to allow both ease of prototyping and future expansion.
- Chapter 6, **Evaluation**, recounts how we integrated user testing in our process, how it was performed, underlining how this testing was performed iteratively as an integral part of the project's lifecycle, and how this shaped the final prototype.
- Chapter 7, **Preliminar Design of Future Functionality**, shows how we envision the near- and medium-term growth of the device, recounting what functionalities we believe are going to be most important to have and how current functionality may evolve.
- Chapter 8, **Conclusions**, will describe the current status of the project and underline long-term challenges that we know must be solved before the prototype can become a full-featured service for Politecnico students.

2. Background and Related Work

2.1 An Overview of Other M-Learning Projects

As we've seen, m-learning is a research field that attracts a growing number of projects. However, it is also an overloaded term; m-learning projects may span anything from PDA-guided museum tours, to downloadable language courses for portable music players, to in-class interactive implements such as student polling or asking questions via SMSs, to entire courses delivered exclusively via cell phone video streaming. In this vast array of possibilities, applying some project taxonomy is necessary, so that we can find more easily those projects whose conclusions we can build upon to make ours succeed.

This categorization can be hard indeed. There have been various approaches to this in literature; a very interesting one is found in [3], where we can see a categorization attempt that looks at several possible axes — context of use, tools employed, subject and object of the learning process — and then evaluates the position on such axes of a large number of published m-learning projects. Although this helps in finding interesting projects to compare ours to, it also has some fundamental drawbacks; for instance, the classification given in the appendix only gives a single value to the context in which the use of the tool occurs, whereas as we've seen in the introduction our system has the ambition of eventually becoming a student companion, able to accompany the student and assist him to learn *both* within a classroom *and* outside of it — that is, using the terminology of [3], not just in a "formal" context (the classroom), but even possibly later within "independent" contexts as well, outside of any formalized location that has the express function of enabling learning.

However, assigning a single value for 'context' (as well as the other axes) is not without basis. Most projects indeed tend to "stick" to a single context or a single mode of operation, so much that we can easily discern some broad-strokes tendencies according to the context:

- In a *physical* context, the application often "lives" on a device that is used within a specified location, for instance to provide content tied to a particular position (museum guides, tools to guide children in localized learning activities) or to the context in general (field guides).
- In a *formal* context, the application is mostly used to improve and promote user participation. Many of these systems focus themselves on *immediate* interaction only — the student participates in that particular moment, but has no lasting tangible benefit from the participation other than what momentarily increased engagement can provide. (Polling software, question-asking software and lesson streaming all fall into this category.)

- In an *independent* context, applications often turn into *logistical* items: they deliver information to the user, be it course material or schedules; or retrieve information from the user, such as homework delivery; or provide mostly freeform two-way communication among teacher and students (*eg*. forums). This may seem familiar; most projects in this area tend to resemble traditional Internet-based learning management systems (LMSs) simply transported to a mobile application or website form.

There are several examples we can make in each of these categories:

- An extremely simple example of a m-learning system are automated, location-sensitive tour guides employed by museums (running in a physical context, in [3]'s classification). One such example in literature is the Tate Modern Multimedia Tour [8], where visitors to the Tate Modern art gallery in London were provided with a PDA able to sense its location within the exhibition, which provides information on the nearest piece of art. Such information can be either audiovisual in nature, or may only possess an audio track. The project is interesting as it notes how mobile devices can distract from the task at hand (for example, presenting audiovisual content from the PDA's screen distracts the user from admiring the work of art on their own); interaction design must be careful to let mobile devices augment the experience, rather than supplant it.
- A slightly different project is LOCH [9], where learners experience a foreign language directly in a community of native speakers (in the project's case, Japanese learners in Japan). In this case, a PDA accompanies the learner while on the field, providing the users with tasks to be performed within the town (for instance, interviewing a person, or having a conversation at a convenience store). Proof of completion of the task is sent back to the instructor via the PDA, who can then provide another task; PDAs can locate themselves via GPS, allowing teachers to monitor the learners from afar.
  This project, like the Tate Modern example above, confines a device within a specified context (although one where the user has full freedom of movement, instead of the enclosed space of an art gallery), allowing users to experience outside the classroom in a structured way.
- Software-mediated collaboration and communication, as we've seen, are powerful tools in the hand of a mobile learner and often used in m-learning projects. One project that eschews tying the user to a specific context is xTasks [10], where students can collaborate online on the composition of a text; users can write by inserting and working on individual paragraphs in parallel, both from a desktop computer and from PDAs that were loaned to students for the occasion. The collaboration software is available for use on mobile devices, and indeed, the system does not constrain users to

mobile terminals, allowing them to move from them to PCs seamlessly. One of the most interesting conclusions of this project is the realization that a multichannel application need not expose its entire feature set on every channel, but can adapt itself depending on the kind of device in use.

Several other such examples are available (in [3] and other references). However, given the design angle we used (that of a project able to engage students in the classroom while being useful outside of it), we kept our focus on projects whose aim was to allow communication and interaction within the classroom, to see what kind of inputs are worthwhile to exchange and how we can use them to improve a user's post-lesson review experience.

2.2 Classtalk

One of the earliest in-class systems aiming to increase in-the-classroom user engagement was Classtalk, developed in 1996 at the University of Massachusetts at Amherst [7]. From a technical point of view, the system is similar to the way we want to structure ours: the teacher has a computer at their disposal, whereas students have mobile hardware on hand. The approach chosen in this system is that of *pushing interactions*: the teacher defines a set of *tasks* that they want the student to pursue, taking the form of a set of polls and quizzes, and the lesson periodically pauses to allow students to perform these tasks using the mobile devices. Students are encouraged to collaborate on giving answers; indeed, the system is built upon a constructivist philosophy of education, which underlines how actual, collaborative application of newly acquired skills is beneficial to learning in general.

These early results have informed much later research on mobile learning systems; however, we felt that they were not applicable entirely to our case, because:

- The interactions performed by this system are *punctual*, in the sense of being only relevant to the moment at hand. This was an entirely reasonable choice at the time, because the state of communication networks was not such that users had the possibility of accessing the system after the lesson was over in a convenient way (now easy thanks to the success of home and mobile Internet connections), nor the pricey devices could be entrusted to the students (which is no longer a factor today, as students may be able bring their own devices to participate in the system).
- Lessons have to be restructured entirely to accommodate for the new format: teachers have to change their approach and reorganize lessons to allow for tasks to be solved, and students no longer have a possibility to remain passive during the lesson. We felt that this, while not undesirable per se, would however hamper the gradual adoption we envisioned for our system — one of our aims is *acceptance* of the system on the part of students and professors, and to do so we felt we should start in a way that has a minimal initial burden, due to change, on both classes of actors. In Classtalk, the

burden is maximal; this is even explicitly noted in the original whitepaper as a possible adoption-hampering factor.

- The system requires collaboration between students in a very short, enforced period of time (a few minutes per task). The interaction is timed, and students may, as observed by the original research, feel pressured to perform choices they might later regret or feel frustrated by the lesson not going “at their own pace”. We felt a system should not frustrate users this way, and allow them to make mistakes that do not have any impact past that particular instant, by preserving anonymity and associating their actions to a well-defined context.

2.3 Shangai Jiaotong University

A more recent project that caught our attention was the m-learning system introduced at Shangai’s Jiaotong University [4] [5] with the expressed goals of:

- Increasing in-class interactivity to attempt to reduce the lack of engagement that, according to the authors, is endemic in Chinese universities during traditional classroom lessons; and
- Allowing non-present users to attend the lesson from afar.

The system they produced essentially allows users to access a number of services during the lesson through an application for their mobile phones, including a video stream of the lesson (both of the professor and the material) to be displayed on cell phones of absent users, text messaging to the instructor for questions, immediate feedback (*eg*. “This point needs explanation”), class polling and a forum for student communication that is open to students during the lesson (both for present and remote students).

Students were surveyed during the evaluation phase of this project for their opinion on the system and its impact on their classroom attendance habits. Among the positive qualities of the project, those surveyed underlined how they appreciated the fact they could access the lesson stream at their leisure and repeat its viewing anywhere, and also how they could influence the lecturer by asking questions or making remarks on the lesson via text messaging, especially for non-present users who are not otherwise able to communicate with the teacher.

Although the results are impressive, we felt some points were deserving of further examination. In particular, the three top complaints from the surveyed users were that [4]:

- The pace was too fast;
- The users could not keep up with the speed of text messaging;
- The users were not accustomed to “such interactive activities”

Indeed, very little time is devoted in the study to an examination of the human-computer interaction occurring between students and the system, leading us to think these complaints stem from insufficient examination of this aspect; the feeling of “pace being too fast” might stem from some of the interactions between the system and the users being sources of distraction. Indeed, text composition on a mobile device (especially a device that does not have a full keyboard) requires learned skills and devoting attention to the activity even for the short time required for a text message, which can severely hurt overall concentration, especially if any important lesson content is missed as an effect of the distraction.

Another observation we made is how the different parts of the system are essentially disjoint. For example, a description of the question text messages subsystem describes how the teacher must manually enable or disable such system, as there is no provision for the system to associate both the question and the lesson content to the particular moment in time where it was made; the student has the additional burden of having to fit questions in a time window, and the teacher the burden of managing this time window, due to a lack of automation on the part of the system.

2.4 MIT.EDU

Another project that caught our attention did so exactly because it addressed some of our objections to the Shangai Jiaotong University m-learning system. In the MIT.EDU system [6], users are outfitted with PDAs; a homegrown software architecture (dubbed “MIThril”) allows quick prototyping and deployment of distributed applications that can use the device’s capabilities to transmit, elaborate and infer information from the PDAs’ embedded sensors, or from specific peripherals paired with them.

Even though this system is technically very interesting, it is but a framework; our attention is attracted by the human-computer interaction angle, which requires examining the applications deployed on this system. Some of those applications are indeed rather interesting; for example, one such application, named OpinionMetrics, is intended to be used in the classroom as a feedback system to the lecturer, to record the ongoing mood of the students and potentially notify the teacher that interest is waning; it can also be used to propose instant polls to the classroom. This system is interesting because it also correlates these mood reports to the lecture, in the form of an audio recording of the same, and uses this correlation both as an input for the teacher for further improving future lectures, and also automatically to, for instance, extract an audio summary of the most interesting moments of the lesson (as inferred by the clustering of positive mood reports). Several experiments have been performed building upon this basic structure, such as the use of biofeedback and motion sensors (for example, to capture nodding), to try to automatically capture some the input without requiring the user to look at the device.

As mentioned, this system is extremely interesting from a technical standpoint, and also introduces a number of ideas that, while slightly invasive, underline the notion that one of the main objectives of such an in-class system is to integrate itself into the existing situation without distracting the students from their learning objectives. The system however does little usability testing, aside from mostly positive polls (which, as mentioned in [7], must take into account the fact that experience of a novel system may produce a positive reaction because it is novel, rather than actually satisfying). The focus on the system was *how* to obtain the information from the students, but does not envision scenarios where this data is used to actually change the students' learning model, rather than just enriching a lesson with data for the teacher.

As we'll see, all of these points were on our minds when we designed our own system. We haven't reached all objectives; indeed, as we'll detail, our prototype only has limited usefulness outside the classroom. We however think that, by employing a simple but appropriate technological base and carefully testing interactions between students, teachers and the system, the latter will have the potential to actually have a lasting, beneficial effect on their way the former organize their learning, rather than just remaining an interesting experiment they'll forget when they leave the classroom and return home.

3. Eliciting Requirements

As we’ve seen, adding mobile technology to the study process seems to yield a possibility of improvement. Still, we were doubtful exactly on *what* to build.

Since the very beginning, we focused on students more than on other possible stakeholders, as they would be the ones who we’d subject to the innovative (mobile) element to the system. Due to this, we felt important to involve them from the very beginning, eliciting their needs with an online survey (Appendix A). The survey aims to find out the habits of students both inside and outside the classroom, the technology they have at their disposal and how much that technology is already part of their learning activities.

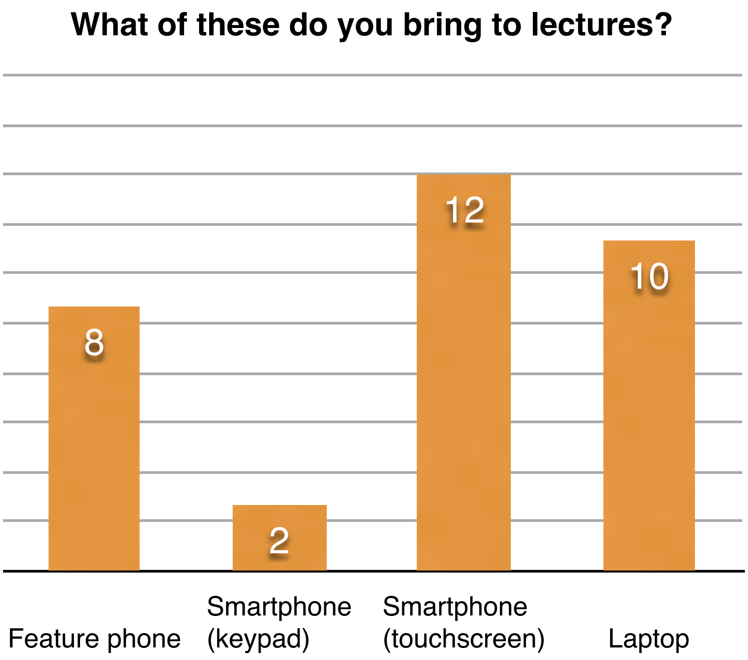
We performed this surveying through the Internet in Fall of 2010, gathering 24 responses from both Politecnico students and students of other universities. The results are shown in tables 1 to 6 (in Appendix A). The small sample size and the potential for bias discouraged us from making general assumptions based on the results, but we still examined them for potential trends of interests, finding that:

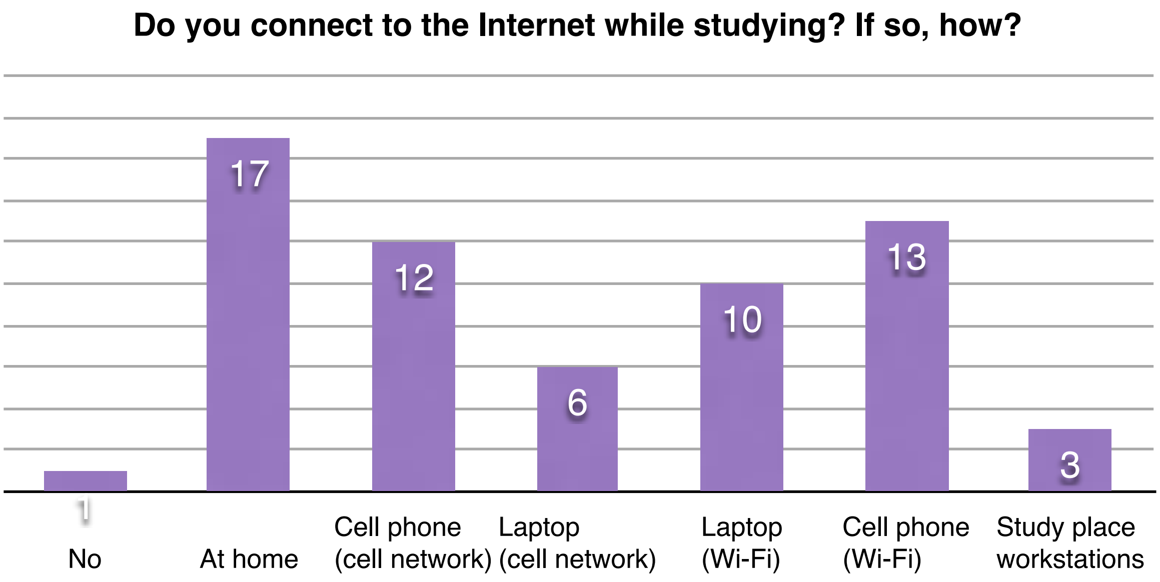
- Among our interviewees, the most common on-campus activity is attending lectures; during lectures, interviewees predominantly take their own notes (20 out of 24), whereas the second most common activity is to follow or annotate professor-provided material (12 out of 24). Notes are predominantly taken with pen and paper (19 out of 24, versus only 7 who claimed to use a keyboard, such as a laptop’s).
- The interviewees predominantly cite not being able to capture everything as the main problem with their notes (11). Interestingly, 9 students also stated that they do not later re-read their notes, using them only as mnemonic aids to understand the lesson at the moment. We interpreted this as an interesting focus on the “here and now”, a (somewhat frustrated) wish to capture the essence of the moment during the lesson.
- 12 of our interviewees out of 24 stated that they owned a touchscreen smartphone which they bring with themselves during lectures. Using the cell phone to ask a question does not drive interest by itself (11 out of 24 state their question-making habits would not vary, against 4 that would be more inclined and 9 that would be less inclined to ask); there’s a much more positive response when faced with the prospect of sharing these questions automatically through a website (10 interviewees stated this would make them more likely to ask, along with 10 who stated they would have asked anyway, versus only 4 students who stated they would feel less inclined to ask in this case).

- 15 out of 24 students stated they used the Internet to share notes with others; 19 out of 24 stated they would be fine with automatically sharing notes taken during the lesson with their fellow students.
- The home is the most popular place to study, with 20 out of 24 students stating they spend a significant amount of time engaged in learning activities there; spaces meant for studying come next, at 13 out of 24. During study, most students stated they reread existing material — textbooks, professor-provided material such as study notes or slide printouts, and their own notes; fewer students stated they exercise themselves, with the most common source of exercises being the professor (through his own study notes, or mock exam sheets).
- 21 out of 24 interviewees have their cell phone with themselves while they study; 20 out of 24 have a laptop. Most students can use the Internet connection of the place where they study at — the top choices being their home's connection (for 17 of them), or through their cell phone via a Wi-Fi network in their chosen place of study (for 13 of them). It's worth underlining, for context, that Politecnico classrooms and study rooms are covered with a free Wi-Fi service available to all students.
- IT technology is mostly used to download material (19), read material (17), consult logistical details such as exam and course schedules (16) or research the course's subject (14). Surprisingly few use a computer to annotate their material (5), ask the professor about something (3) or asking fellow students about the subject (5). Somewhat unsurprisingly, when asked to speculate about performing these activities on a cell phone, the relative numbers do not change much; users expect the same activities to be useful on a cell phone as they are on a computer.
- Most students would be interested in trying to carry out the aforementioned activities with a cell phone application (21).

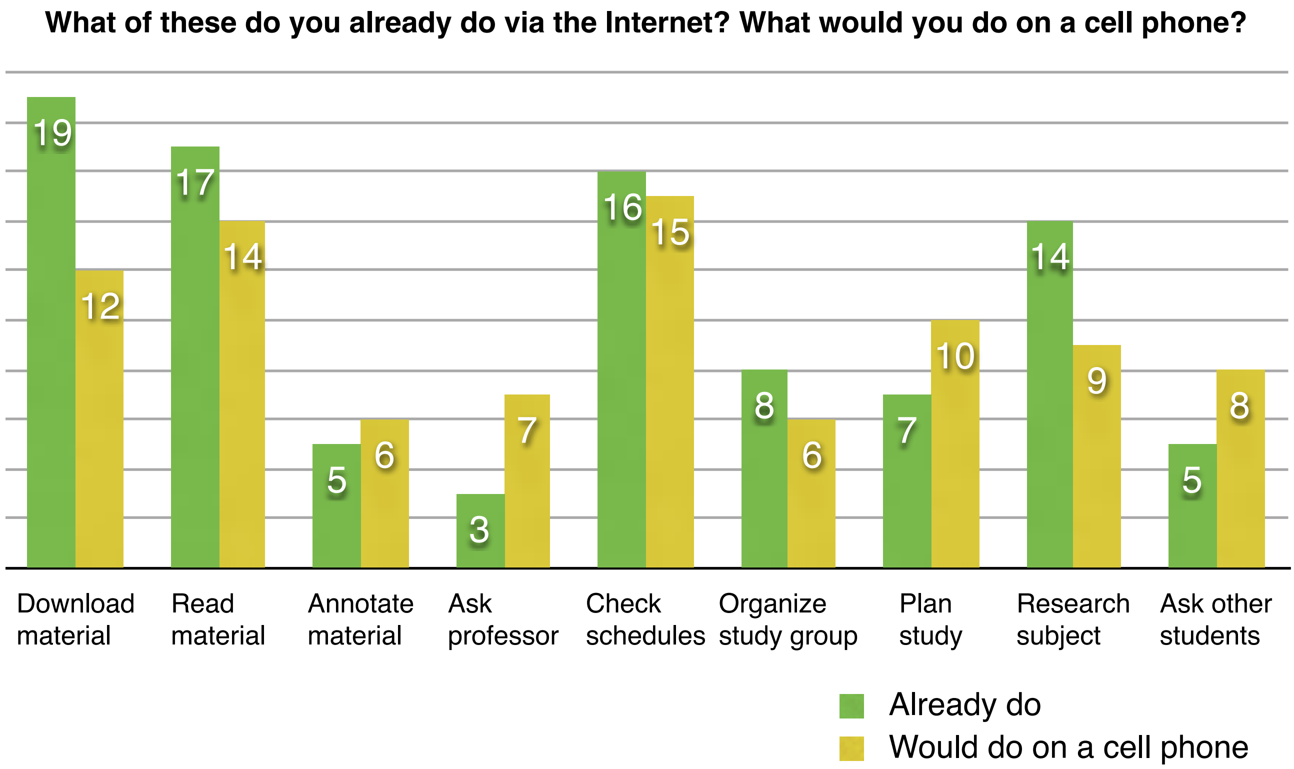
So, interesting take-aways on the survey were: we had reached a tech-savvy group who nonetheless recognized the practical limits of their technology. The interest in speculative questions helped us realize that, although there may not be strictly speaking a *need* for mobile technology as a study helper or enabler, students would however still be willing to experiment with bringing changes to their usual process.

Despite these results, we started out in a vacuum: a relative lack of "solid" data prevented us from verifying our hypotheses. We could count on our experience as student and professor in this same university; still, we decided that "playing it safe" was a sound choice. To this end, we resolved to be innovative, but not radical: unless we could gather solid proof that some advancement was strictly better than the traditional alternative, we would not force ourselves on the study process more than strictly required. The aim would be *augmenting* it, rather than *replacing* it outright.





Graph 3.1: Technological highlights from the survey.



Graph 3.2: Highlight of common tasks, already performed on computers and speculatively performed on mobile.

4. "WakeUp!" Design

4.1 The User Experience Design Process

In software engineering, it is extremely rare that a project can be brought to fruition without a process to accompany it and ensure its growth occurs in a context of continuous improvement. This becomes especially true in cases like ours, where initial requirements are generic and the domain is understood only in general terms.

The solution to this is to ensure for timely, continuous feedback, so that we can be sure that whatever thing we're building actually does perform as we intended. By treating our domain like a "black box", we essentially perform experiments on it by:

- making *hypotheses* based on the initial elicited requirements and our perception of the best-suited solution to the problems we faced;
- building a series of *incremental prototypes* to test these hypotheses;
- performing *evaluations* of the prototype at every step of the way (in real-world environments);
- *understanding* the result of the evaluations, which can be used to better predict user behavior and therefore update the design to be better performing in those areas where we feel it has been lacking.

To avoid "wasting" work, raising the cost of retrying, iterations of this process have been kept short; after an initial period where the initial prototype was designed and build, we performed evaluations, be they informal or in-class simulations involving dozens of people, no less than every two weeks. We felt that an empirical approach to software design such as this one — inspired by "agile" methodologies [11] and HCI design processes — is essential in explorative projects such as our own; this kind of "tweened" testing allows us to treat users' expectations as a "black box" and to rapidly act on feedback to correct any problems we observed.

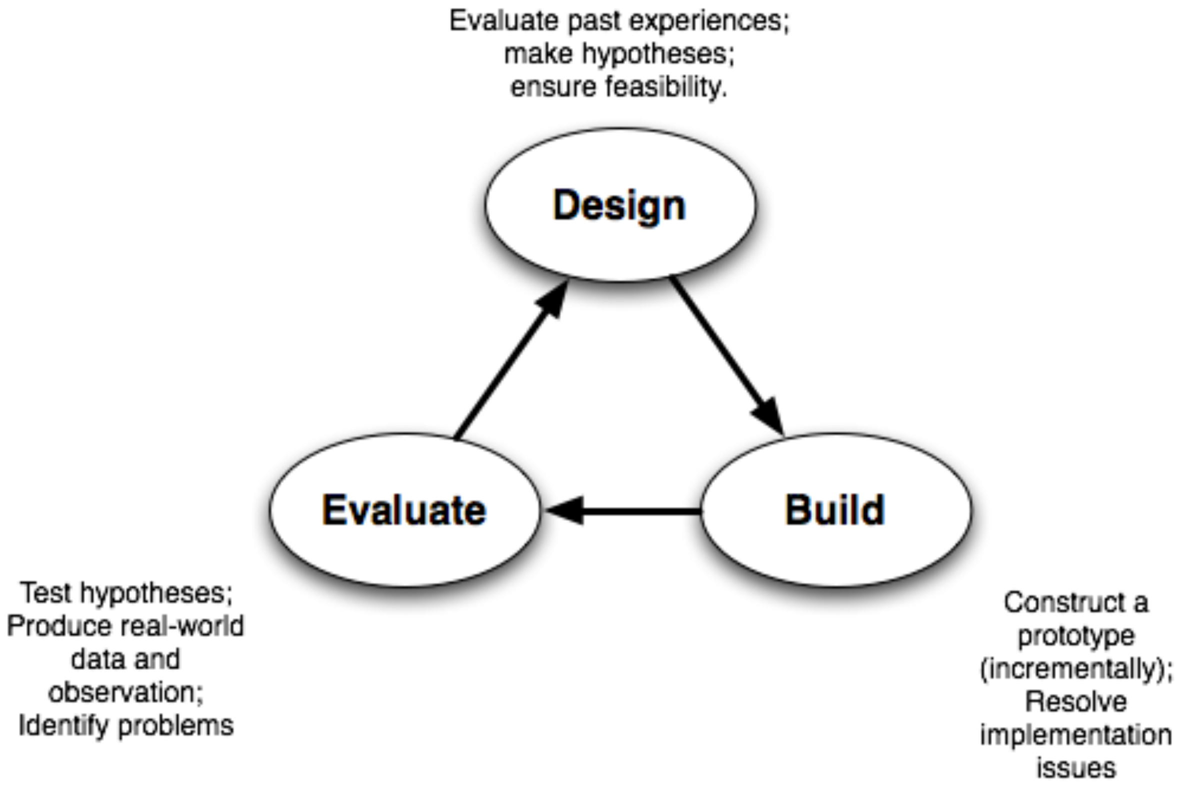


Figure 4.1: The simplest process that could possibly work.

In this chapter, we will discuss the *design* portion of this process, showing how we chose to attack this problem. The subsequent two chapters will discuss the *implementation* and *evaluation* stages, outlined above; but it must be underlined that these three activities were performed iteratively multiple times, rather than linearly, ensuring that the results of an evaluations were used to course-correct the design and, subsequently, improve the implementation.

4.2 Designing the Workflow

One of the things that were apparent even since the very beginning of our exploration was that we'd need to define a narrow scope for our exploration to fit into time and resource constraints. This was particularly important due to our generic premises: "adding mobile technology to the study process", while a good starting point, risks being vague enough that, unless precautions were taken, we could be tempted to try doing too much and delivering too little, or nothing at all. On the other hand, we also knew that our resource constraints made the delivery of a complete system unwieldy; we would need to carefully find the right approach to this trade-off.

We approached this problem from an end user’s point of view, trying to produce user scenarios that exemplified what we aimed to obtain. Therefore, we had to consider:

- *who* would be the actors of our system. Obviously students are involved, and we decided early we wanted to let professors be directly part of the system so that they could benefit from any data gathered by the system, too.
- *when* and *where* the application would be used. This is especially important for a mobile application, as the context of use can be an essential part of the system. We understood that there are basically three moments when the Politecnico student approaches the subject of learning:
  - *during formal learning activities*, including lessons, laboratories, etc. Most of this time is spent *in the classroom*, attending lessons.
  - *during self-study*, which can occur either *at home* or *in a facility designed for studying* (a study room on campus, the public library, *etc*). At times, the student may also perform self-study in *places not designed for it* (for instance, in a park during a sunny day; the Politecnico itself offers tables in the open in several spaces).
  - *in unplanned moments*. The student may have a momentary doubt, or be asked a question he knows is in the material contained in the application, for instance.
- *what* to provide to the user.

This last point is maybe the most interesting. We struggled finding potential candidates, because even though a student’s life revolves around a large number of things that can be helpfully represented by a computer (course schedules, books, etc.), only a few of those are suitable for display, consumption or editing on a small screen. For example, reading a A4 or US Letter-sized PDF file — one of the most common formats for the delivery of course notes, articles and textbook excerpts — is often difficult on a typical cell phone touchscreen of 3.5” to 4.5”, due to the amount of zooming and panning required to produce readable text.

At this point, we focused on one important observation: that, anecdotally, many (though not all) of the professors lecturing at our university does so with the aid of projected slide decks, typically built with Microsoft PowerPoint. Slide decks have several interesting properties:

- Their content is automatically accessible through the scripting commands provided by the projecting application;

- Slides can be processed automatically to produce outlines and other semi-structured representations of the same; PowerPoint itself can produce such a representation as an “Outline” sidebar during editing.
- Slides often represent in a detailed, semi-structured way the entirety of the course’s content. That is, they often expose the entirety of the subject in some detail, and are often meant to be kept by the students for aiding in later review.
- Slides are already available for current courses where the lecturer employed them, and typically professors that employ them in their courses have already planned for the time and effort required to produce new slides or adapt previous slides to the needs of new courses. This means that by using them, we do not impose an additional burden on any of the actors in the process.

We decided to use slide decks as a “backbone” of our user experience, as a proxy for the entirety of the course’s program. We would help the user retrieve them, access them at any time, and use them as starting points for discussion, review and the prioritization of study.

Given our decision, it was obvious at this point that the main interactions we were going to design all centered around the slide deck concept. Users we surveyed already hinted at the fact they followed the lesson using professor material, including slide deck printouts, so it was natural for us to focus on possible in-the-classroom interactions; we hypothesized that the user could annotate, or ask questions on, a certain slide, or prioritize slides according to his own annotations and other contextual information (for example, whether others had found that particular slide important or difficult to understand). At this point, with annotation in mind, another part of the survey caught our eye: students were lamenting that they could not capture all of the lesson in their notes. This lead us to think about one thing machines could do well in the classroom: being able to "capture the moment", that is, maintain information after the lesson that would normally missed by the student, or that may not be obvious, and make this available after the lesson is over.

These were the basis on which we started designing a workflow that could cover the most useful among these possibilities.

Given the above, it was a simple decision for us to limit the scope of the prototype system to:

- Material delivery;
- Mobile fruition; and
- In-class interactions.

These points were immediately identified to be the most challenging parts of the project. Material delivery has to efficiently handle contextual delivery slide decks — which can have a size of several megabytes — to a typical Politecnico course class, which typically ranges from 30 to 100 students; the material delivery must also happen in a way that does not interfere with in-class interactions, even in the worst case where the student did not allow a download until the lesson already started. (Requiring students to perform the download beforehand is an additional burden, which we aimed to avoid.)

Aside from the justifications above, in-class interaction is also important because that's one of the few times in which we can *guarantee* wireless connectivity to all students (as opposed to encouraging a student to use outside connectivity or his own data plan); most Politecnico classrooms are covered by a Wi-Fi network all students can access, making it a good moment to perform the aforementioned material delivery, in concert with other in-classroom activities.

The final workflow we designed is shown in the diagram in figure 4.2. In this workflow, the applications can be used to access a shared experience in the classroom, where the material is available to all students and the lesson can effectively be "followed" with the aid of the device. We augment this experience with the concept of annotating, and asking and answering questions, both during the lecture and later when the lecture ends.

In the following, we'll walk through each point of entry to our system, discussing why they were made this way and how they interact with each other to ensure the desired experience.

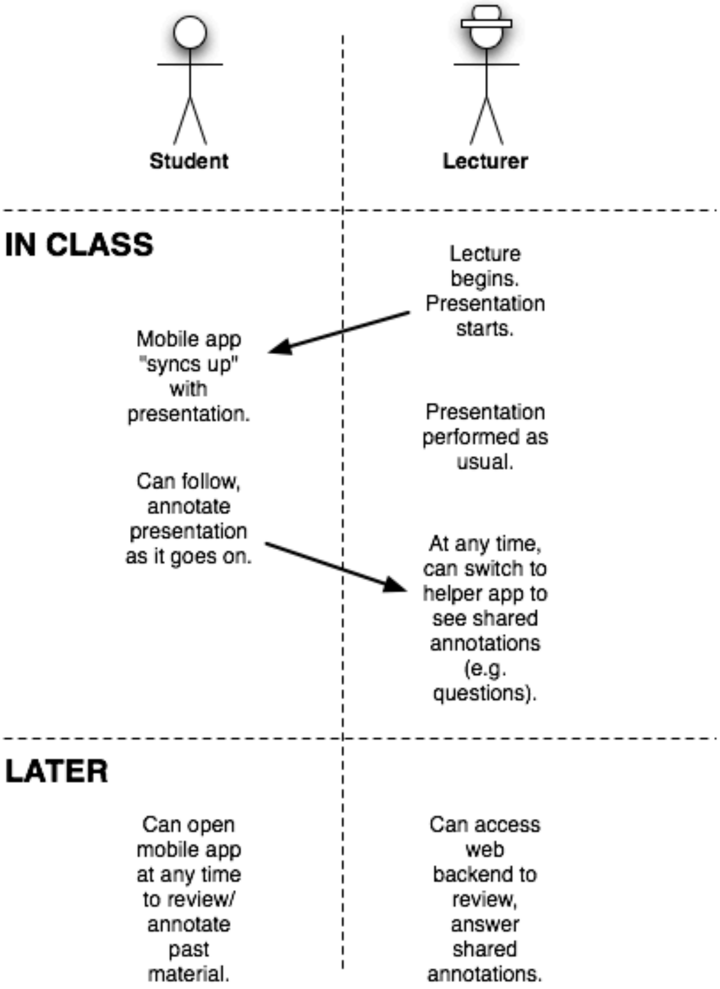

Student
Lecturer
IN CLASS
Lecture begins. Presentation starts.
Mobile app "syncs up" with presentation.
Presentation performed as usual.
Can follow, annotate presentation as it goes on.
At any time, can switch to helper app to see shared annotations (e.g. questions).
LATER
Can open mobile app at any time to review/ annotate past material.
Can access web backend to review, answer shared annotations.

Figure 4.2: A bird's eye view of the final in-class workflow.

4.3 Features of the Mobile Application (for Students)

4.3.1 Initial View

Students install the mobile application on their phones at the beginning of the academic year. They can access it at any time, but the application behaves differently depending on the context in which it finds itself: *inside the classroom*, or *outside.*

When the user opens the application inside the classroom, during a lecture, the app automatically synchronizes itself to the lecture. The student sees an outline of the current slide if he holds the device in portrait mode, or an image of the slide if he rotates the device in landscape (figure 4.3).

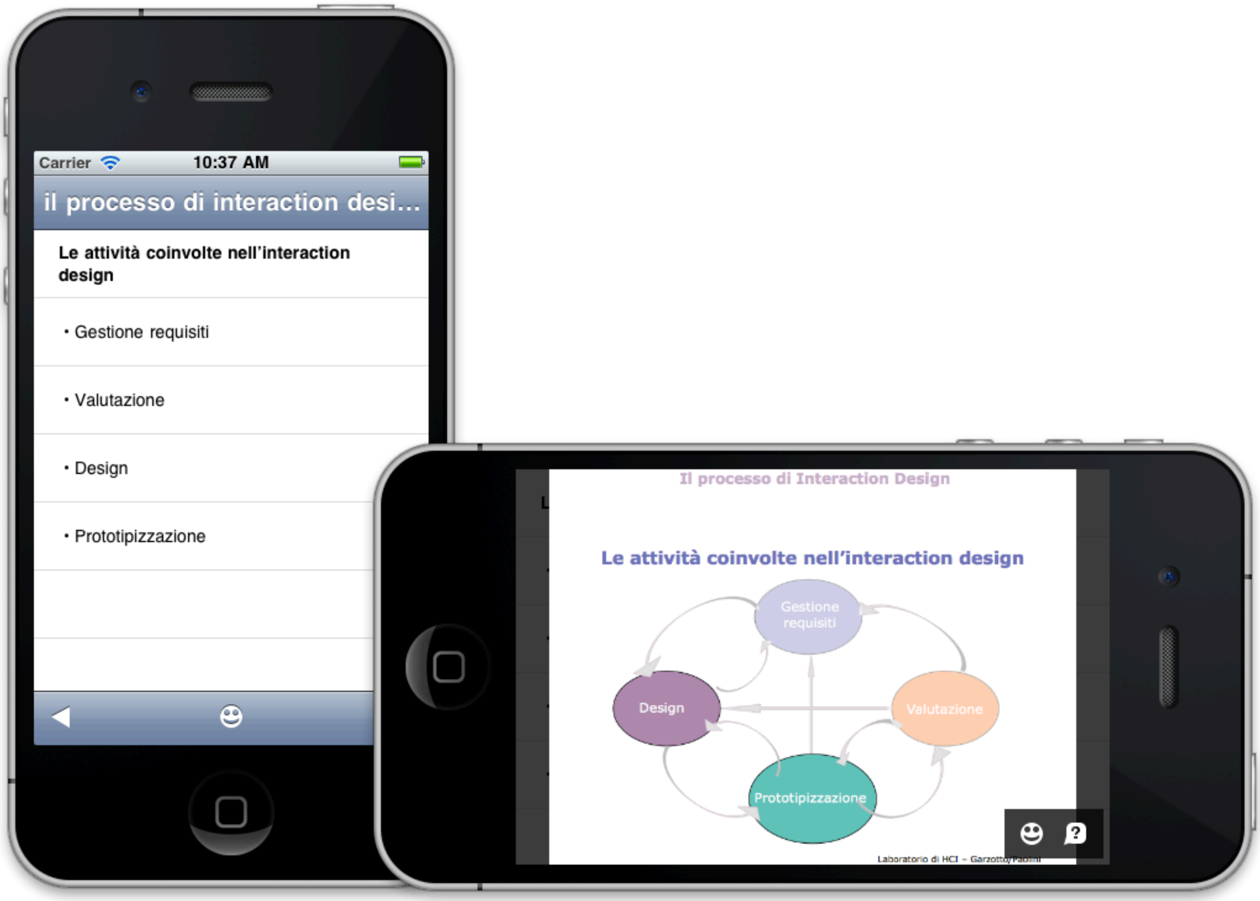

Carrier
10:37 AM
il processo di interaction desi...
Le attività coinvolte nell'interaction design
• Gestione requisiti
• Valutazione
• Design
• Prototipizzazione
Il processo di Interaction Design
Le attività coinvolte nell'interaction design
Gestione requisiti
Design
Valutazione
Prototipizzazione
Laboratorio di HCI – Garzotto, Paolini

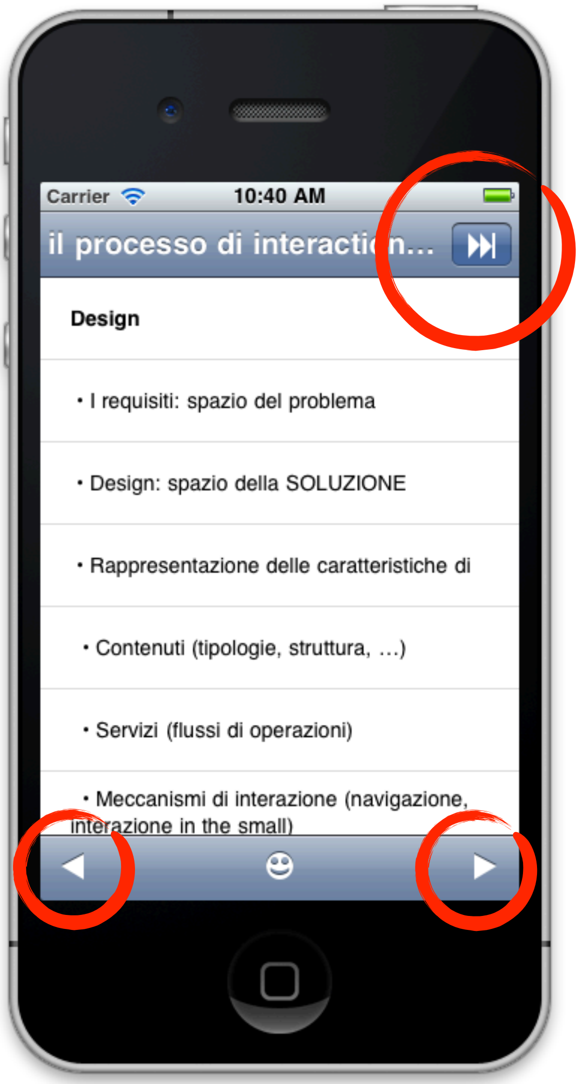
Figure 4.3: The slide display as the user opens the application (outline in portrait, slide image in landscape).

4.3.2 Following & Browsing

If the user does not interact with the device, it stays synchronized to the lecture — whenever the teacher changes slide, the device will display the new slide, sliding it in from the right. This works even if the teacher skips slides or returns to a previous slide; the mobile application will continue in this mode, allowing the user to see the outline or image of the current slide.

The user can at any time *browse the lecture* by either using the left and right arrow toolbar buttons in portrait mode, or swiping the screen left or right. Browsing allows students to return to a previous slide or skip ahead in the presentation. When the user starts browsing, the application does not visibly synchronize to the lecture; the user can continue looking at other slides at their leisure.

Browsing does not change presentation — the user still sees an outline or slide image — but adds a "fast forward" button on the top bar (figure 4.4); touching it will allow the user to return to the current point in the presentation. If at any time the user hits this button, or returns to the current slide, and then stops interacting with the application, the app resumes synchronizing and following the lecturer again.



Design

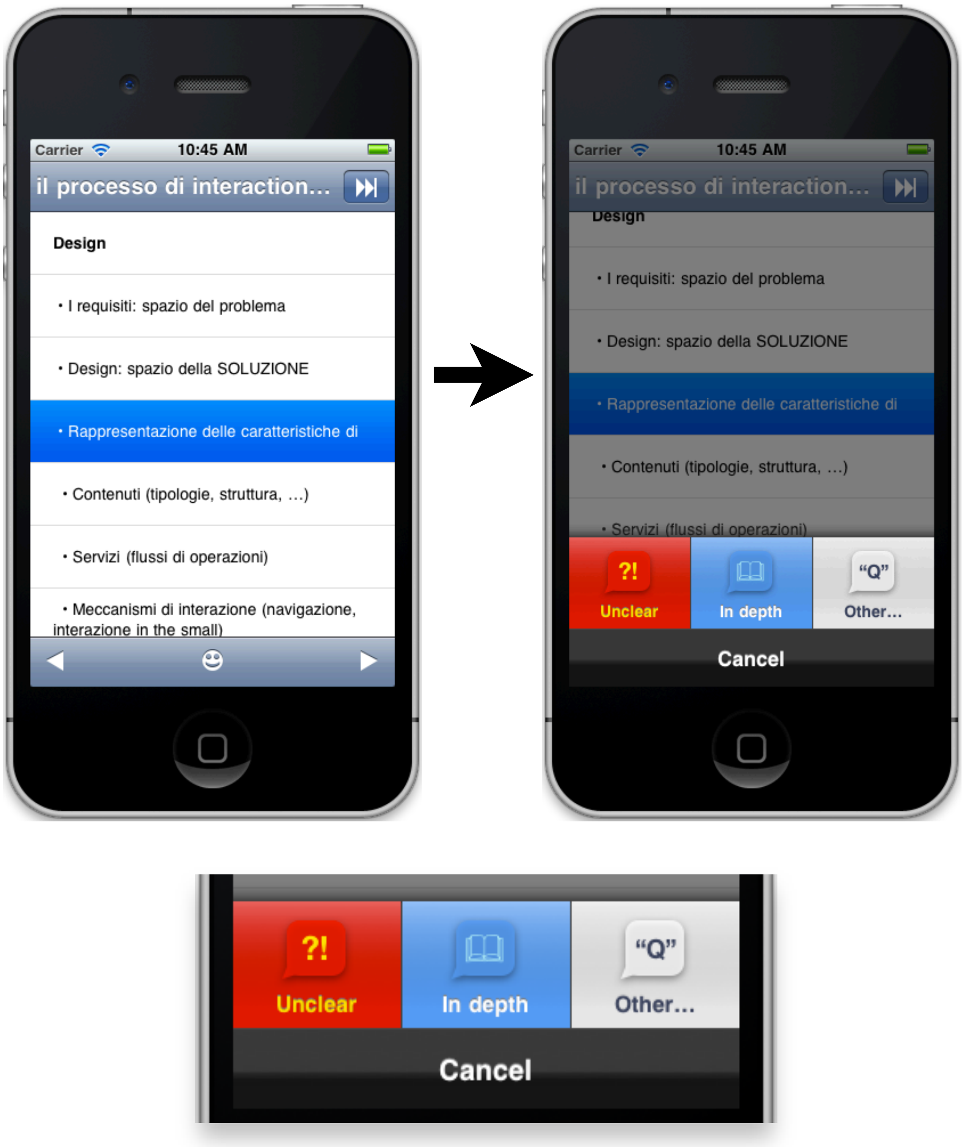
- I requisiti: spazio del problema
- Design: spazio della SOLUZIONE
- Rappresentazione delle caratteristiche di
- Contenuti (tipologie, struttura, …)
- Servizi (flussi di operazioni)
- Meccanismi di interazione (navigazione, interazione in the small)

Figure 4.4: Browsing to another slide during a lecture. Top right circle: the "fast forward" button, which synchronizes again with the lecture by returning to the current slide. Bottom circles: the next and previous slide buttons. The slide can also be changed by swiping left or right on the screen.

4.3.3 Asking Questions

Students can also interact with the lecturer by asking questions, or expressing their moods. Both interactions are very streamlined, to reduce distraction as much as possible.

A student can ask a question regarding a *point* or a *slide* by touching either the point or the slide's title, respectively, from the outline. Doing so shows an *iconographical menu* (figure 4.5) that lists a number of predefined question options; our prototype presents two of these options, "Unclear" (as in: 'this point is unclear') and "In depth" (as in: 'can you go in depth on this point?'). The user can chose any of them, or "Other…" to write a full-text question using the keyboard (figure 4.6).



Design

• I requisiti: spazio del problema

• Design: spazio della SOLUZIONE

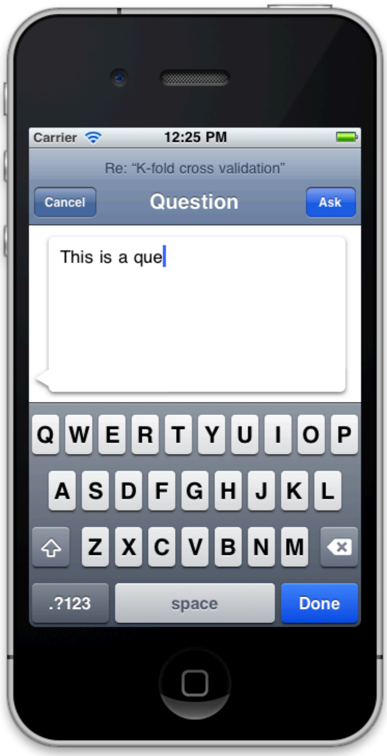
• Rappresentazione delle caratteristiche di

• Contenuti (tipologie, struttura, …)

• Servizi (flussi di operazioni)

• Meccanismi di interazione (navigazione, interazione in the small)

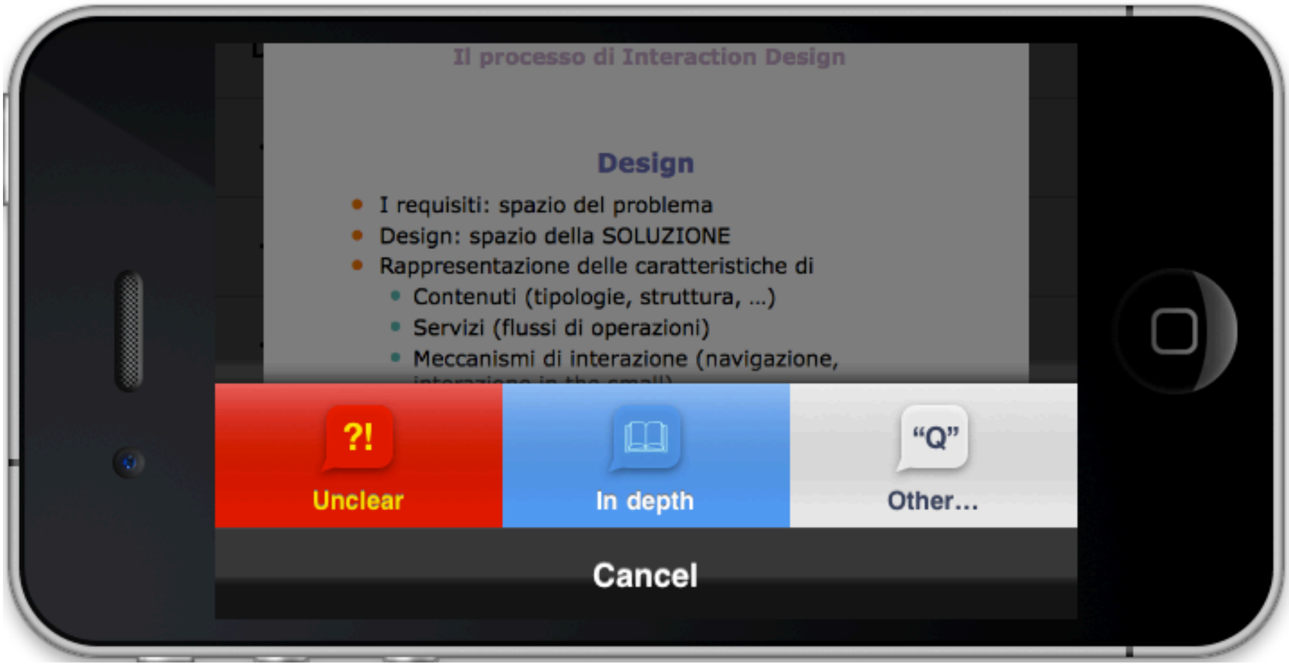
Figure 4.5: Touching a point in the outline view displays the iconographical predefined questions menu.



This is a que

Figure 4.6: Full-text question entry screen (displayed using the "Other..." option in the questions menu).

The same menu is also available in landscape mode by touching the question mark button on the lower right of the slide image view; this displays the same menu (figure 4.7). To simplify interaction, while in this mode the question will be asked in the context of the entire slide, just as if the user had touched the slide's title in portrait mode.

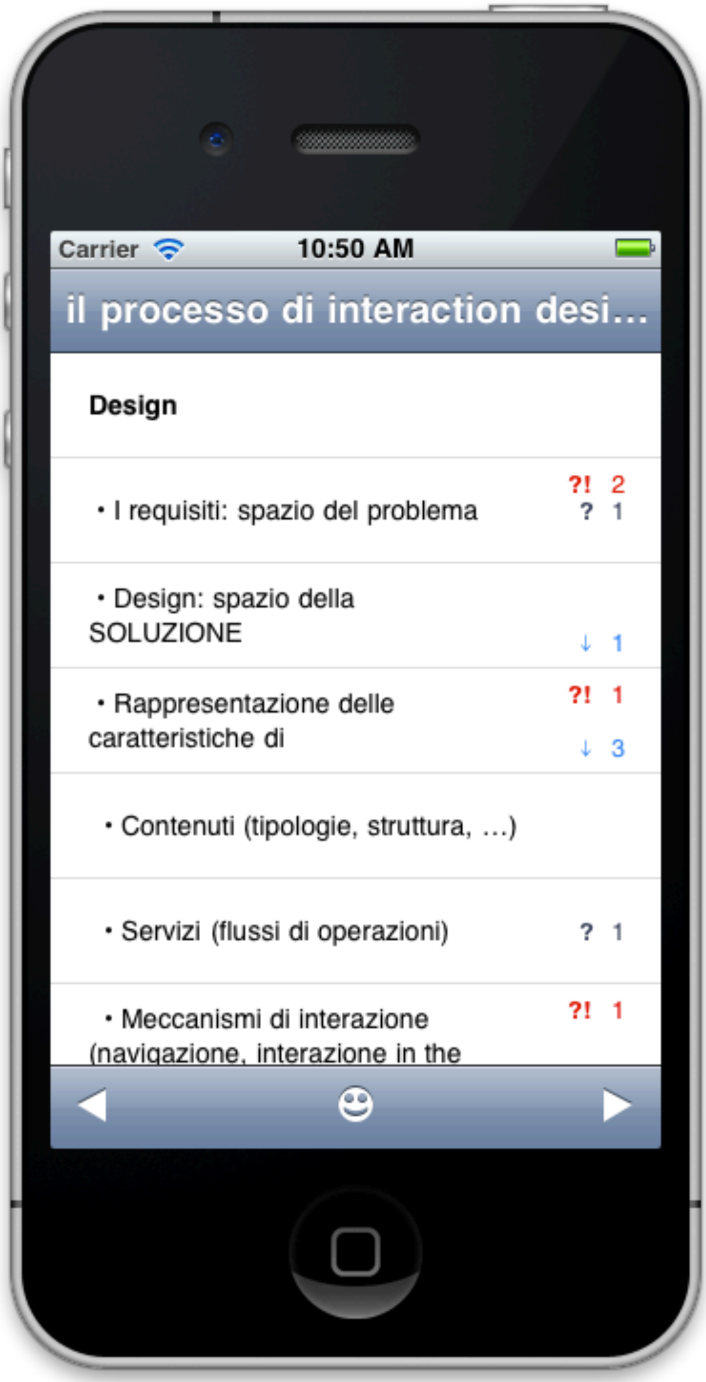

Il processo di Interaction Design

Design

- I requisiti: spazio del problema
- Design: spazio della SOLUZIONE
- Rappresentazione delle caratteristiche di
  - Contenuti (tipologie, struttura, ...)
  - Servizi (flussi di operazioni)
  - Meccanismi di interazione (navigazione,

Figure 4.7: The iconographical menu for asking questions on a slide in landscape mode.

Touching a predefined question kind dismisses the menu; in portrait mode, that question's kind's marker will be added to the point's right gutter, to indicate that the question was asked. When any student in the class asks a question, a marker for that question appears on all student's devices (figure 4.8), allowing them to see the more controversial or important points at a glance.



il processo di interaction desi…

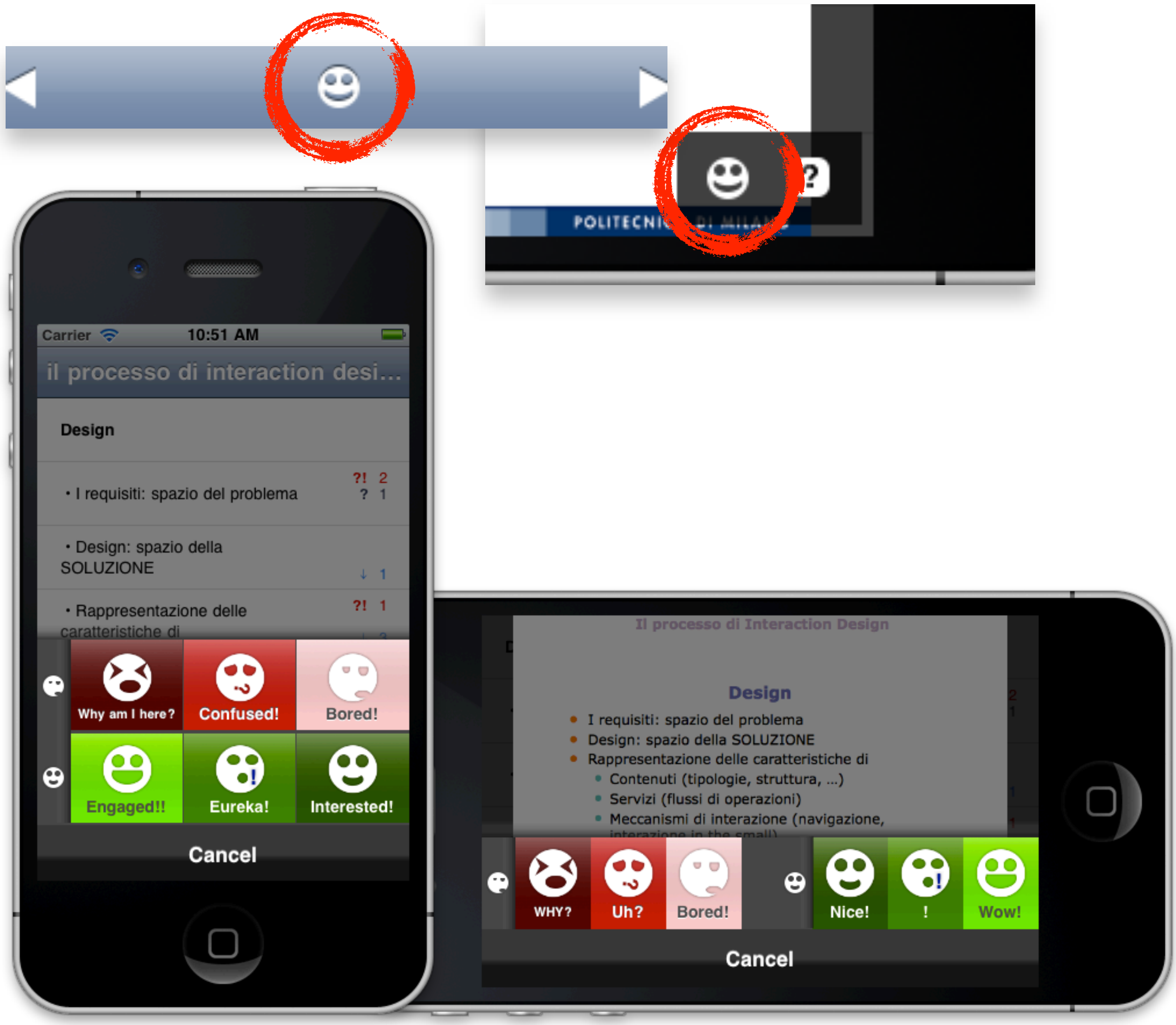
Design

- I requisiti: spazio del problema ?! 2 ? 1
- Design: spazio della SOLUZIONE ↓ 1
- Rappresentazione delle caratteristiche di ?! 1 ↓ 3
- Contenuti (tipologie, struttura, …)
- Servizi (flussi di operazioni) ? 1
- Meccanismi di interazione (navigazione, interazione in the ?! 1

Figure 4.8: Several question markers in the right gutter for several different points on a slide. Red markers correspond to "Unclear", blue to "In depth" and black to free-text questions.

4.3.4 Expressing Moods

The user can also express their mood. He does so with the “emoticon” button in the toolbar (in portrait mode) or on the lower right of the slide (in landscape mode), which invokes another iconographical menu (figure 4.9). Moods are always associated to the entire slide displayed when the user expresses them, and have no markers — they’re only visible by the lecturer through his helper application (again, below). The prototype presents six possible moods, three negative (extreme disappointment, or “why am I here?”; confusion; boredom) and three positive (engagedness; interest; and sudden understanding, or “eureka!”). Red and white color cues, and sorting order (more extreme emotions on the left, milder emotions on the right) attempt to convey the difference visually.



Carrier 10:51 AM

il processo di interaction desi...

Design

- I requisiti: spazio del problema ?! 2 ? 1
- Design: spazio della SOLUZIONE ↓ 1
- Rappresentazione delle caratteristiche di ?! 1

Il processo di Interaction Design

Design

- I requisiti: spazio del problema
- Design: spazio della SOLUZIONE
- Rappresentazione delle caratteristiche di
  - Contenuti (tipologie, struttura, ...)
  - Servizi (flussi di operazioni)
  - Meccanismi di interazione (navigazione, interazione in the small)

Figure 4.9: The “emoticon” buttons for expressing moods (top left, in the portrait mode toolbar; top right, floating over the slide in landscape mode) open an iconographical menu with several predefined choices.

Once the lesson ends, the prototype application stops synchronizing. The slides and question markers, however, are downloaded to the device, and remain available for browsing even outside the classroom. In fact, opening the application outside of a lesson starts it in this mode, allowing the user to review the material. Synchronization ensures the user will not need a network connection outside the classroom (where we can otherwise guarantee the device has connectivity thanks to the free-to-students Wi-Fi service that blankets most buildings of the Politecnico).

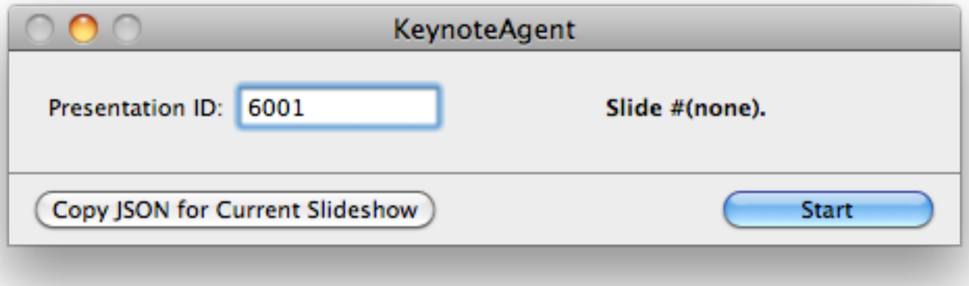
4.4 Features of the Lecturer Helper Application

The student application is all about receiving content and sending "inputs" in the form of questions and moods. The lecturer application, on the other hand, is all about receiving them, allowing the lecturer to answer to questions and adjust to the mood of the class.

The lecturer application is composed of a number of parts. The lecturer begins by launching the helper application itself, which is installed on the projecting laptop (our prototype is in figure 4.10). Clicking its "Start" button for a presentation does two things:

- It starts the projection in the presentation application (for example, in PowerPoint);
- It opens the *lesson monitoring screen* in the web browser.

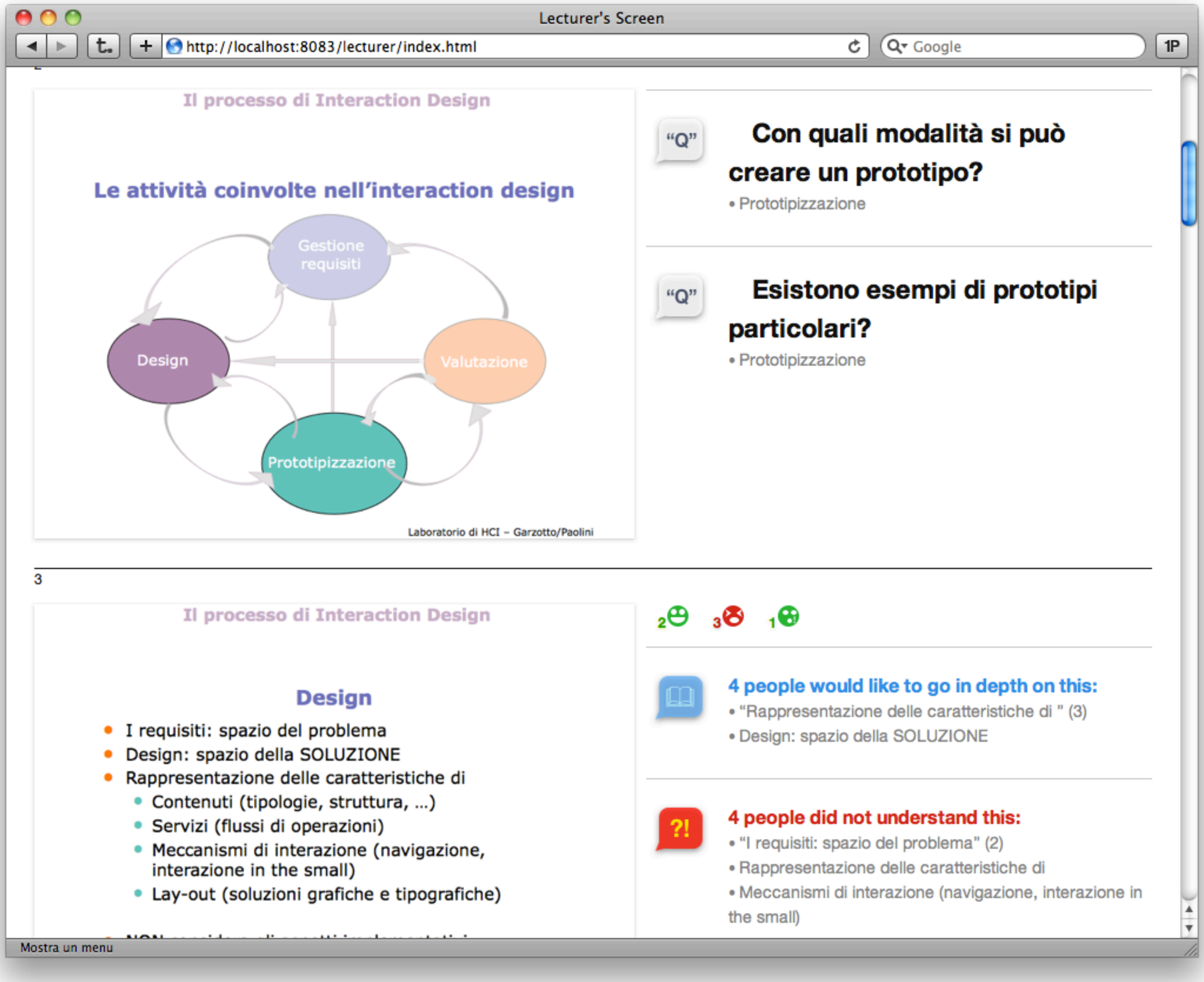
From this moment on, the lecturer interacts with these two, while the helper application broadcasts events to student devices in the background.



Presentation ID: 6001

Slide #(none).

Figure 4.10: The actual helper application. The lecturer does not interact with it other than pressing "Start"; most of the interaction occurs in the projection application and the lesson monitoring screen.



Il processo di Interaction Design

Le attività coinvolte nell'interaction design

Laboratorio di HCI – Garzotto/Paolini

"Q" **Con quali modalità si può creare un prototipo?**
- Prototipizzazione

"Q" **Esistono esempi di prototipi particolari?**
- Prototipizzazione

3

Il processo di Interaction Design

Design

- I requisiti: spazio del problema
- Design: spazio della SOLUZIONE
- Rappresentazione delle caratteristiche di
  - Contenuti (tipologie, struttura, …)
  - Servizi (flussi di operazioni)
  - Meccanismi di interazione (navigazione, interazione in the small)
  - Lay-out (soluzioni grafiche e tipografiche)

2 ☺ 3 ☹ 1 ☺

4 people would like to go in depth on this:
- "Rappresentazione delle caratteristiche di " (3)
- Design: spazio della SOLUZIONE

?! **4 people did not understand this:**
- "I requisiti: spazio del problema" (2)
- Rappresentazione delle caratteristiche di
- Meccanismi di interazione (navigazione, interazione in the small)

Figure 4.11: The lesson monitoring screen, showing the questions and moods for a slide. Note the mood emoticon strip, the free-text question, and the multiple predefined question groupings.

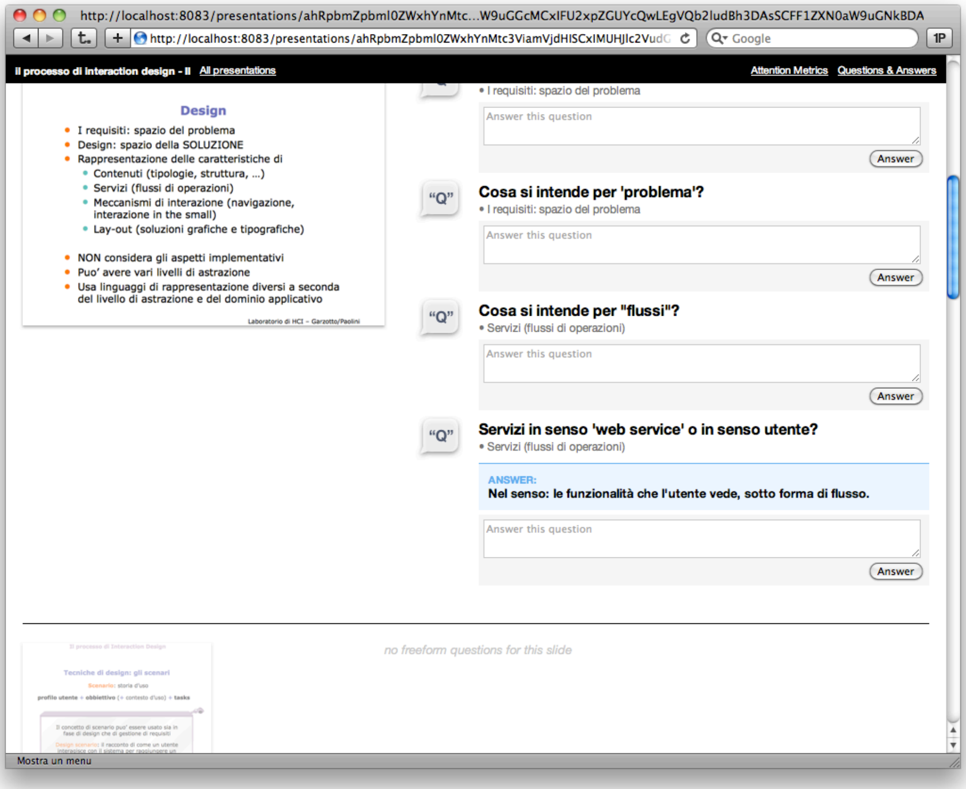
Among these two, the lesson monitoring screen (or "lecturer's screen") is by far the most interesting. The screen is a web page that displays the current state of question and mood annotations, grouped by slide and further reordered for at-a-glance monitoring.

- Moods are displayed in a thin strip near the top of the slide, with colored emoticons (green for positive moods, red for negative ones);
- Then, all questions are displayed. Free-text questions are displayed first. Then, those using a *predefined* kind are shown, each kind making its own group and containing the quantity and text of the points they refer to.

This grouping, the large font and consistent colors and iconography aid in keeping track of what is going on. Having questions be displayed like this allows the professor to choose exactly when to ask; if they decide to answer questions only periodically, they may quit the presentation software periodically to reveal the window beneath; or they may choose to monitor the lesson in real time by placing the window on another screen (for example, the laptop's built-in screen while the presentation runs on the external projector).

The monitoring screen does not allow interaction; it is a static display of content (although updated in real time). However, professors can, after the lesson is over, access a website front-end to retrieve data about the lecture. Our prototype front-end provides two interesting features (shown in figure 4.12 and 4.13): a "Questions & Answers" section allowing the lecturer to respond to freeform questions after the lesson is over, and a Metrics screen that displays, for each slide in the presentation, the number of shared annotations — both moods and questions — in bar chart form, divided by kind, allowing the teacher to immediately notice controversial or unclear points in the presentation. These two features are just very small samples of what can be done by our application's backend.

As our work has been student-centric, lecturer features in the finished prototype are somewhat lacking; for instance, the prototype is unable to deliver answers sent through the "Questions & Answers" page to students, even though it does record them. We foresee that lecturer and student web access would be an important part of the project's future work, and one of the major focus areas of future work, which we will discuss in this document's conclusive chapter.



http://localhost:8083/presentations/ahRpbmZpbml0ZWxhYnMtc...W9uGGcMCxIFU2xpZGUYcQwLEgVQb2ludBh3DAsSCFF1ZXN0aW9uGNkBDA

http://localhost:8083/presentations/ahRpbmZpbml0ZWxhYnMtc3ViamVjdHlSCxIMUHJlc2VudG

Il processo di interaction design - II All presentations — Attention Metrics | Questions & Answers

Design

- I requisiti: spazio del problema
- Design: spazio della SOLUZIONE
- Rappresentazione delle caratteristiche di
  - Contenuti (tipologie, struttura, ...)
  - Servizi (flussi di operazioni)
  - Meccanismi di interazione (navigazione, interazione in the small)
  - Lay-out (soluzioni grafiche e tipografiche)
- NON considera gli aspetti implementativi
- Puo' avere vari livelli di astrazione
- Usa linguaggi di rappresentazione diversi a seconda del livello di astrazione e del dominio applicativo

Laboratorio di HCI – Garzotto/Paolini

• I requisiti: spazio del problema

Answer this question

Answer

“Q” **Cosa si intende per 'problema'?**

• I requisiti: spazio del problema

Answer this question

Answer

“Q” **Cosa si intende per "flussi"?**

• Servizi (flussi di operazioni)

Answer this question

Answer

“Q” **Servizi in senso 'web service' o in senso utente?**

• Servizi (flussi di operazioni)

ANSWER:
Nel senso: le funzionalità che l'utente vede, sotto forma di flusso.

Answer this question

Answer

no freeform questions for this slide

Mostra un menu

Figure 4.12: The Questions & Answers screen from the final prototype, with free-text questions and a published answer (in blue).

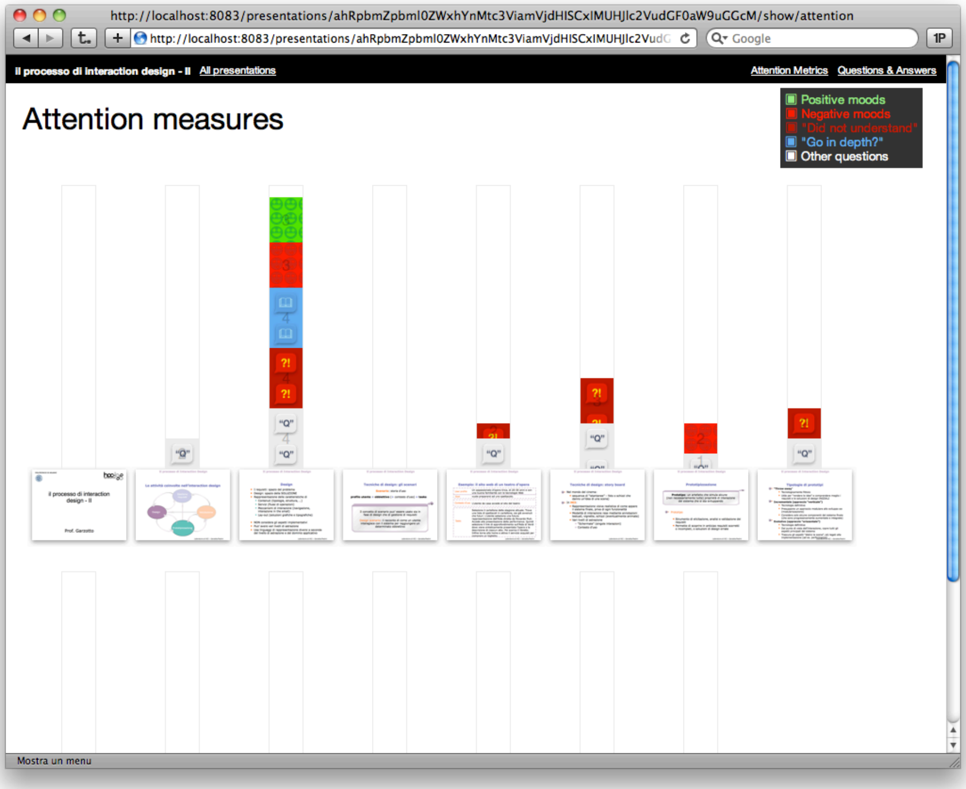


Figure 4.13: The Metrics screen from the final prototype, a bar chart showing the number of annotations — moods and questions, grouped by kind — have been made at each point in the presentation.

5. Enabling Technology

5.1 The Technological Context

The mobile technology market has been a center of innovation in the last few years. Thanks to the intervention of several firms — such as Nokia, Apple, Microsoft, Google and others — smartphones have taken the place of PDAs and are now commonplace fixtures of everyday life. Especially since the introduction of the iPhone in 2007, smartphones are slowly becoming synonymous with capacitive touchscreens, accelerometers, location awareness (often in the form of GPS receivers), and simple finger-friendly interfaces with rich animation and support for intuitive gestures.

Despite the impact it had on the market, choosing the iPhone's platform (Apple iOS) for this project was not a foregone conclusion; the platform, while popular, is not as widespread within our campus as one might think, as we found out during both the initial requirements survey and the subsequent in-the-field evaluations. In fact, Politecnico students we worked with for evaluation seemed to favor lower-cost smartphones running the Symbian or Windows Mobile operating systems to the higher-cost iPhones and iPhone-like Android or Windows Phone 7 phones; this might change as time goes on and touchscreen smartphones reduce their cost to meet the cost-conscious segment of the market.

However, phone availability was not the only (or even a major) factor in our decisions. The main factor was *familiarity*: due to the restricted time and resources we had, we chose to use tools and platforms on which we already had vast previous experience, to reduce the potential impact of having to learn about a new platform even as we tackle a complex new project. The author of this thesis has had experience of use of Apple developer tools since 2003, and was among those selected for pre-release testing the iPhone OS developer toolchain during its release in early 2008; he also publishes and maintains commercial software on the platform. It was then natural for us to use a platform for which we had not only ready tools and developer experience, but also a good understanding of its potential, capabilities and interface guidelines.

Another point in favor of iOS is its extensive user interface definition library. Where other systems often require long hours of work to produce even remarkably, if deceptively, simple interfaces, iOS has a very powerful user interface library providing complex behaviors, such as memory-efficient inertially scrolling list views, rotating windows that match the interface orientation and extensive, customizable built-in support for potentially complex animations using the Core Animation framework. All of these would prove to be essential in crafting the finished prototype in an acceptable amount of time.

A similar reasoning applies to the choice of web server software and hosting. We chose to use Google App Engine as our provider because the author was already very familiar with the Python-based web framework it offers, and because the simplified model it offers allowed us to focus on providing functionality rather having to prepare and configure hosting ourselves.

Neither choice was free of downsides. For example, the Apple iOS environment is notoriously so user-focused, writing native applications for it may prove challenging; for example, the iOS model does not allow the application control over its lifecycle, and something like a phone call or responding to a SMS may cause the application to enter the background, where it stays suspended without executing, or even being entirely killed and removed from memory. This is on top of problems typical of mobile environments, such as the fact phones may sometimes lose connectivity unexpectedly (due to the movement reducing available signal, or even the phone turning off radio hardware to save power). This isn't entirely negative; it required us to think about a system where clients could come and go at any time, adding reliability provisions to our architecture.

Similarly, the Google App Engine hosting provider, while known, is geared towards web application that execute in a web browser; in particular, it did not allow long-lived connections, or other efficient, low-latency mechanisms for "pushing" of information to clients, for instance[1]. Since our system relies on delivering realtime updates to mobile devices when in the classroom, we had to make the polling for updated information — the only mechanism allowed by Google's infrastructure — especially efficient. Fortunately, the judicious application of caching allowed us to stay well within Google's maximum usage quotas during both development and evaluation.

[1] This situation was remedied somewhat by Google producing a "Channel API" in December 2010, when our project was already well underway. We did not alter the design to accommodate for this new feature, although a future version of the application certainly might.

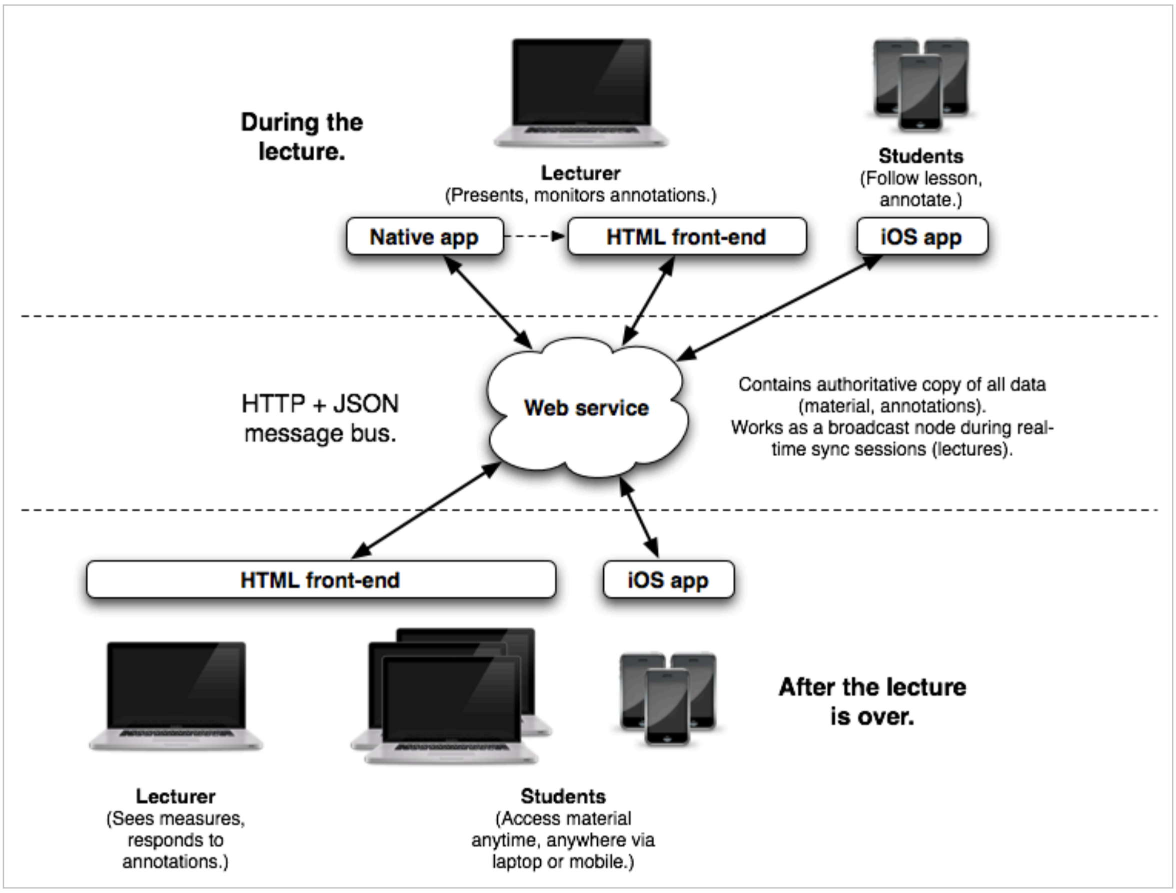

During the lecture.
Lecturer
(Presents, monitors annotations.)
Students
(Follow lesson, annotate.)
Native app
HTML front-end
iOS app
HTTP + JSON message bus.
Web service
Contains authoritative copy of all data (material, annotations).
Works as a broadcast node during real-time sync sessions (lectures).
HTML front-end
iOS app
After the lecture is over.
Lecturer
(Sees measures, responds to annotations.)
Students
(Access material anytime, anywhere via laptop or mobile.)

Figure 5.1: Bird's eye view of the main technological components of the prototype system.

5.2 Foundations

Producing the system detailed above — a system providing real-time sync of data over an IP network, with seamless delivery of content — is not trivial. Fortunately, we could rely on proven technologies and patterns that made our job much simpler that it could have been otherwise, with an eye to allowing future extensibility and porting to other platforms.

A bird's eye view of the system is available in figure 5.1. The system is built out of four applications:

- One instance of a web service running on the Google App Engine infrastructure;
- One instance of a web front-end, which may serve lecturers (in real-time during a lesson) and students, which in our prototype is served by the same infrastructure also handling the web service;
- Any number of mobile application clients (in our prototype system, iOS native applications); and
- Any number of lecturer helper applications (in our prototype, Windows or Mac native applications).

Most of these applications do not communicate directly; all communication is mediated by the web service. The choice of a web server was dictated by our desire for communication to be reliable and always available, inside and outside the Politecnico network, and provides several advantages:

- A web service, using HTTP as its transport protocol, can be accessed by the strictly regulated Wi-Fi network available at our university. The network requires traffic to pass through a proxy, and only traffic to certain ports (including the HTTP 80 port and HTTPS 443 port) is allowed. Additionally, all traffic between devices connected to that network is disallowed, requiring us to deploy the web service on the Internet rather than on the student-accessible internal network.
- By first uploading material to a web service, the material becomes available independently of the lecturer's host being available or reachable. This means that any disruption that could occur during the lesson (for instance, the lecturer's laptop crashing or losing connectivity) does not prevent the material streamed up to that point from remaining available for those students that retain connectivity.
- The material is available both inside and outside the classroom. If a student does not attend the lesson, or has to leave early, they need only later open the mobile application to synchronize any material they haven't yet obtained, which will also conclude any partial download of existing material (due to interruptions like losing connectivity, the OS demanding network use, or the user simply closing the application and leaving).
- By "splitting up" the presentation in its basic semantic components (slides and points), we can stream all of it or just the not-yet-downloaded part of it to a device with ease. This allows for progressive download, easy addressing of each part within the system and user interface needs-based prioritization of fetching, among many other advantages. The device can also set its own download pace, allowing us to better manage available bandwidth. Finally, the entire set of HTTP features is potentially available for each portion of the presentation, including encryption (via HTTPS), authentication, content type negotiation, automatic language selection, proxy caching, *etc*.

In the following, we will describe and observe the system "in motion" in several contexts, and we'll go in depth on both the internal design of each component and the way the component coordinate themselves through communication with the web service.

5.3 Before the Lecture: the web service and the lecturer helper application

As we've seen in the workflow design section, we expect the lecturer to prepare a slide deck and provide it to the system by alerting the native helper application on their laptop, either in advance or as part of starting the presentation as the lecture begins. At this point, the native application must provide the slide deck to the web service.

The application does not upload the presentation to the service as-is (that is, by uploading the document file). This would require the service to parse the document format, which is a very nontrivial operation, especially with the plethora of formats and in-format variants supported by the most popular presentation application, Microsoft PowerPoint. Instead, we enlist the application's help through its *scripting interfaces*, querying them to discern the structure and content of the document beforehand, and uploading the extracted data instead, sidestepping any format incompatibility issues. The process extracts:

- An *outline* of the presentation, composed of a set of *slide outlines* each containing a set of *points* (which may have varying indentation, alongside their text content); and
- An *image representation* of each slide.

The web service helps in this regard: each of the above concepts (presentation, slide, slide image, point) is assigned a *unique URL* within the web service. For example, a presentation may correspond to a URL `http://service.host/presentations/123`, and a point within a slide may be available at `http://service.host/presentations/123/slides/12/point/3`. The intention was for our finished service to perform as a sort of dataflow component as in figure 5.2: it inspects the slides, obtains the data, and pushes the obtained that to the web service by performing appropriate HTTP PUT requests on the URLs corresponding to the desired resource, at any granularity level. Our prototype application was unfortunately not so refined, as the web service only supported pushing content at a entire-presentation (coarse) granularity, but the design remains the same and can be easily extended to perform as intended.

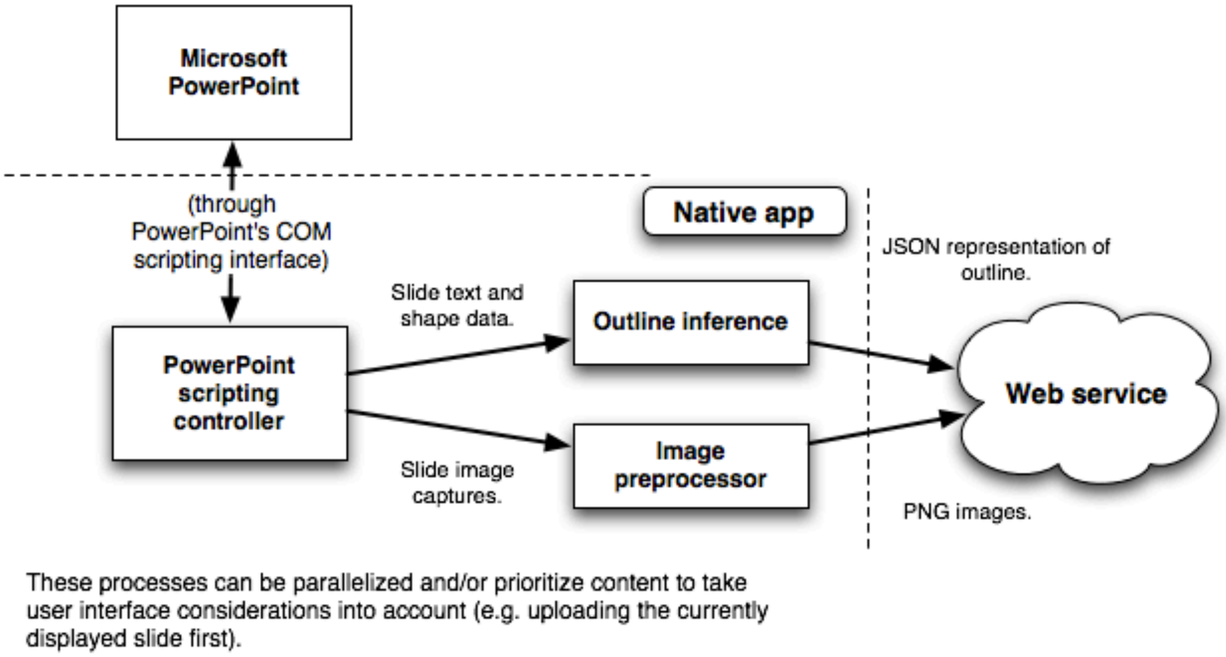

Microsoft PowerPoint
(through PowerPoint's COM scripting interface)
Native app
JSON representation of outline.
Slide text and shape data.
Outline inference
PowerPoint scripting controller
Web service
Image preprocessor
Slide image captures.
PNG images.
These processes can be parallelized and/or prioritize content to take user interface considerations into account (e.g. uploading the currently displayed slide first).

Figure 5.2: The lecturer helper application's presentation import pipeline. It was only partially implemented in our prototype application.

The native application also manages another HTTP resource, whose address is at `http://service.host/live`, which contains the information regarding the current lecture. Whenever the presentation state changes (for instance, by hitting the "Start" button in the native application, or moving between slides, or closing the presentation software) the application performs a PUT request on this address to notify the web service of the change to be broadcast to users. This triggers mode changes in all clients as the lesson starts and goes on.

5.4 During the Lecture: the iOS and web clients

While the professor sets up his laptop and starts the presentation, students will arrive and possibly open the mobile application on their phones. The mobile application polls the web service's `/live` resource at fixed intervals, noting any change from the last known recorded state and the state as reported on the server.

This polling, like all other network-using activities inside the iOS application, is handled by a graph comprising several controller-layer objects. The root of this graph is an object called a *sync coordinator*, sitting at the center of a hub-and-spoke model where the spokes are *sync controllers*, each handling a specific kind of resource that may arrive, downloaded, from the web service. The application categorizes every potential response from the server into a taxonomy of *schemas*, each of which represents the expected format of a particular class of server resources (presentations, single slides, single points within slides, slide images, live presentation state, questions and moods); most of these are expected to be in the JSON interchange format, containing specified values and value types. For instance, a presentation may be returned from the server, if downloaded, as a JSON payload of the form:

```
{
        "title": "Titolo della presentazione",
        "slides": [
             { "URL": "/presentation/1/slide/1" },
             { "URL": "/presentation/1/slide/2" },
             { "URL": "/presentation/1/slide/3" },
        ],
        ... etcetera ...
}
```

In this case, the schema class responsible for validating a presentation payload will know that it must be parsed to a JSON object containing `"title"`, `"slides"`, *etc.* keys whose values must be strings, arrays of JSON objects, *etc.* In fact, schemas can nest: the items in the `slides` array are actually of a different schema (one used for slides), which may specify just the URL to the slide or contain the actual contents of the same. This allows us to fine-tune how much content is provided during a single HTTP GET request — for instance, downloading just the currently-displayed slide for immediate display to the user, or downloading an entire presentation in one request in case there's no need to display just one part of it, but the app needs to cache it in its entirety for later.

When such a payload arrives, the sync coordinator's controllers come into play. Each controller is associated to a schema, and the coordinator routes each download to the associated controller; the controller then performs whatever job is typically associated with that schema and can notify the coordinator of nested schemas so that they are routed to another controller that knows how to handle them. There are also additional provisions to allow controllers to know when a requested nested schema has been processed, allowing for instance to set relationships between the model object that represent the main schema and the one for the nested schema that the other controller has created. The nested schema may contain more subschemas, which are then handled by other controllers; the controllers do not invoke each other directly, avoiding tight coupling, but always communicate in terms of *update requests* they send to the sync coordinator for routing.

Sync controllers are not the only sources of update requests; indeed, typically the coordinator needs an update request from the outside of the graph to start. However, once for instance a polling timer produces an update request for the live presentation resource, this produces a cascade of related subrequests. (For example: since the current slide has changed, the live presentation controller may produce an update request for the new slide; if the new slide refers to a never-before-seen presentation, the slide controller can produce an update request for the presentation, which may produce update requests for all other slides in the deck, and so on.) This organization keeps the update logic confined to the coordinator graph, but allows us some flexibility in choosing how to trigger it.

This system, while slightly complex, has several advantages:

- Redundancy in parsing code is minimal. Each coordinator performs actions for its own schema type and is not tightly coupled with other classes, allowing for modularity.
- One "funnel" (the coordinator) is responsible for performing all downloads, and is provided priority information as part of the update requests it routes. It can then prioritize downloads that are for immediate resource display over those that are performed for optimistic caching. Additionally, it can monitor the Internet connection, enqueuing or disallowing requests, or retrying failed ones, transparently, or interrogate controllers to determine if a download is necessary or if its result has already been cached, avoiding unnecessary network traffic.
- The library that provides the sync coordination mechanism can also provide built-in sync controllers, for example to standardize the polling behavior we describe above, while allowing the application to define its own.

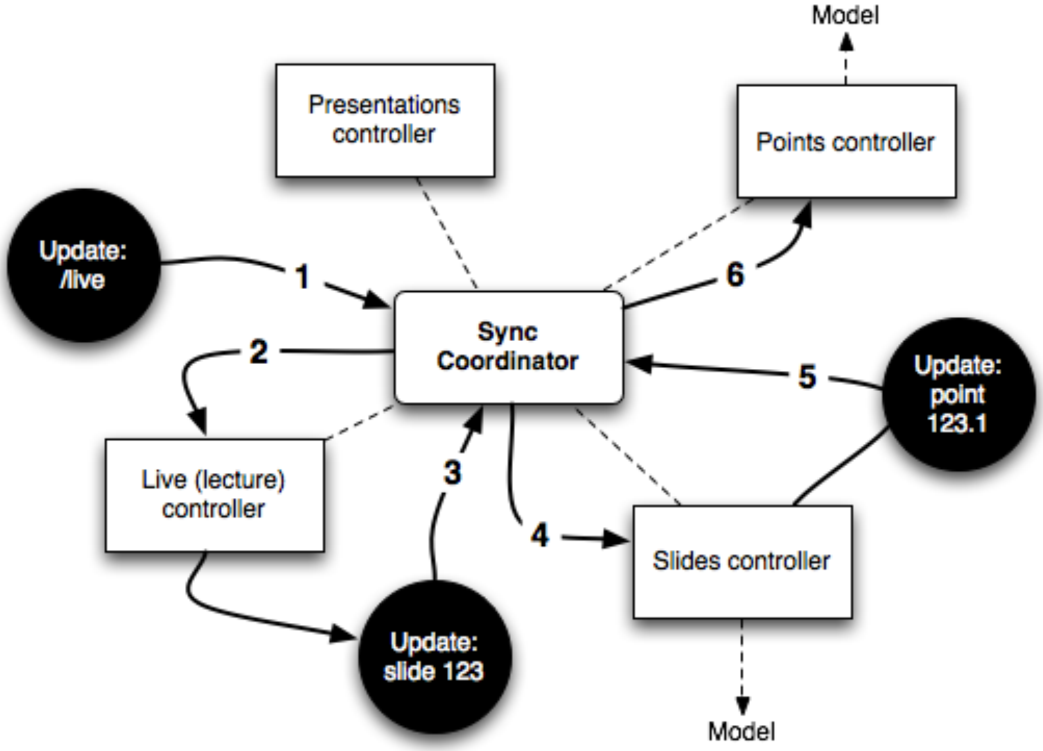


Figure 5.3: The sync coordinator and controllers in action. An original request is routed, and produces a cascade of requests that update the application's model.

As we mentioned, this hub-and-spoke graph is conceptually part of the controller layer, and influences the model layer, typically by caching the result of the downloads. The application uses Apple's powerful Core Data object-oriented database to store these results, and uses its built-in notification mechanisms to update the view layer accordingly, providing the user with content and updating any onscreen control as the content is processed, piece by piece, by sync controllers. The controllers that manage the view layer also send update requests to the sync coordinator, ensuring that slides the user wishes to see (e.g., while browsing the presentation) are prioritized over other downloads. The Core Data database is saved to mass storage on the device, ensuring the user always has access to whatever was previously downloaded.

Although the view layer forms the core of the iOS application, this part is, from a technical point of view, the least complex of the project. Apple provides very powerful built-in behaviors and prepackaged views which we used to produce a simple, but aesthetically pleasing and smooth user interface. For example, the user can swipe using a finger over the outline or slide view, and an animation will swap the old slide in with the new one entering from the appropriate direction; the swipe recognition is performed by a `UISwipeGestureRecognizer` component, encapsulating Apple's swipe-detection heuristics, and the powerful Core Animation subsystem is asked to perform a swipe-typed `CATransition` just before we change the content of the screen. Both of these classes are system-provided, and produce simple but powerful and well-tuned effects with a handful of lines of code. The pattern of built-in, powerful, simple-to-use view components repeats itself to form the core upon which every interaction in the application is constructed.

5.5 Technology Used in the Web Service

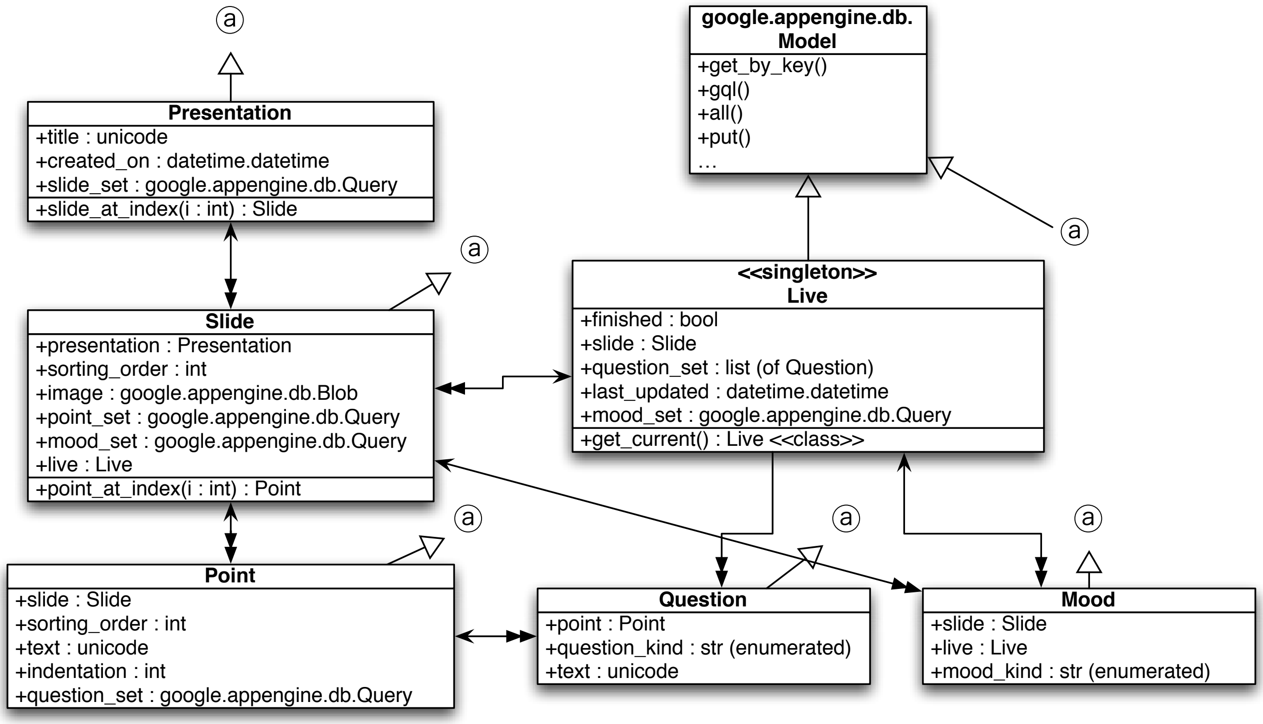
Despite being the centerpiece of this set, the web service is much simpler than most other components of this system. We avoid the complexity often inherent in the server portion of a client-server system by:

- Pushing most of the view logic to clients, such as the iOS application and the JavaScript code running in the user's browser in the case of the lecturer's screen and web front-end; the business logic on the server only pertains to storage and display of structured information (and, in a non-prototype system, would also include such things as authentication and use fairness policies).
- Using the tools provided to us by the Python Google App Engine SDK, in particular the App Engine datastore, an object-oriented database backed by Google's powerful BigTable [12] storage system and the bare-bones web application framework ("`webapp`").
- Applying the principles of representational state transfer, or REST [13], so that there is little or no "impedance mismatch" between the HTTP transport protocol and the way we use it. In particular, each instance in the persistent datastore is associated to a

unique URL, and refers to other entities by URL; each of these URLs respects the semantics of the main kinds of HTTP requests (GET, PUT, DELETE etc.). In particular, a GET request performed on that URL will return a JSON payload describing the resource and 'linking' to, or providing the URL to, other related resources (such as a slide JSON description containing a link to its image representation).

As mentioned, we then get all the benefits of a "RESTful" web service, such as the fact we can freely address our datastore with simple, well-defined semantics, allowing flexibility in how we construct clients; the possibility of accessing the server from code running in a web browser with ease; and all the features we inherit from the HTTP protocol (content type and language negotiation, caching semantics, *etc*). While we use very little in our prototype system, several of these features — especially those related to caching — would be very useful in a finished system.

Of particular note is the peculiar structure of the data in the web service, which is also mirrored in all client applications at some level of detail, but is particularly simple to display in the App Engine datastore model. A pseudo-UML diagram of the associated model classes is shown in figure 5.4.



ⓐ
Presentation
+title : unicode
+created_on : datetime.datetime
+slide_set : google.appengine.db.Query
+slide_at_index(i : int) : Slide
google.appengine.db.
Model
+get_by_key()
+gql()
+all()
+put()
…
ⓐ
ⓐ
<<singleton>>
Live
+finished : bool
+slide : Slide
+question_set : list (of Question)
+last_updated : datetime.datetime
+mood_set : google.appengine.db.Query
+get_current() : Live <<class>>
Slide
+presentation : Presentation
+sorting_order : int
+image : google.appengine.db.Blob
+point_set : google.appengine.db.Query
+mood_set : google.appengine.db.Query
+live : Live
+point_at_index(i : int) : Point
ⓐ
ⓐ
ⓐ
Point
+slide : Slide
+sorting_order : int
+text : unicode
+indentation : int
+question_set : google.appengine.db.Query
Question
+point : Point
+question_kind : str (enumerated)
+text : unicode
Mood
+slide : Slide
+live : Live
+mood_kind : str (enumerated)

Figure 5.4: Partial diagram of model classes used by the web service for persistent storage.

To better understand the diagram, it should be mentioned that the persistent datastore provided by App Engine will persist instances of all subclasses of google.appengine.db.Model; class methods on this common superclass, inherited by subclasses, also provide a way to retrieve instances and query the datastore efficiently. In the diagram, a one-to-many relationship has been represented in a simplified fashion with a arrow-to-double-arrow line, where the single arrow points to the "one" side of the relationship and the double-arrow to the "many" side. The two sides are accessible through object properties (for example, Slide exposes a presentation property that fetches the corresponding Presentation instance, whereas the Model superclass provides Presentation subclasses with a slide_set property containing a query that returns all Slide instances that refer to the Presentation). "<<class>>" annotations indicate that the item is a class method or property; additionally, for implementation simplicity of the prototype, <<singleton>> marks how the system only allows one Live instance to exist at any given time, supporting a single lecture being handled by this backend at any one time. (The system can be easily extended to remove this limitation, but would need a way to control lesson start and end — naturally tied to the lecturer helper application's monitoring — which we did not implement here.)

In this diagram, we can see how annotations are handled. First of all, the Live object (that represents the currently running lecture) keeps track of all question and mood annotations that have been shared during the lecture. You can also see how questions and moods are handled differently; questions have been conceptually tied to the presentation (so that they remain in successive lectures using it as accumulated knowledge within the web service), whereas moods are an attribute of the particular lecture they are associated to. Accordingly, the Live object has a one-way relationship with Questions that were asked during the lecture, but a regular two-way relationship with Moods.

One of the areas that concerned us was the response of the server to polling the /live resource, which exposes the contents of the Live object above to clients; as mentioned, at the time the system was designed, Google App Engine provided no viable way to "push", or efficiently provide clients with material. Fortunately, however, the hosting is explicitly optimized to serve a large number of "small" (less than 1 MB), "fast" (indicatively less than 100ms) requests, and provides developers with access to an installation of the popular memcached in-memory caching engine. By aggressively caching the /live resource in memory, we had more than acceptable request roundabout times (with the initial request taking around 200ms, with all subsequent, cache-enabled request taking less than one millisecond).

Although the Google infrastructure helped us getting started quickly, it comes with its own share of problems. Even with caching, use of the infrastructure noticeably uses up a significant part of the daily quota allowed to applications (for example, the second evaluation — a 30' lesson with five devices polling the web service — depleted roughly 5%

of the quota). Given the number of lectures occurring at Politecnico di Milano every day, using this application in production for even a fraction of them would probably quickly deplete the entire quota. Additionally, moving *off* the infrastructure is made hard by the fact that our use of the built-in Google APIs ties us to the particular, not easily replicable object-oriented datastore, and that the built-in “webapp” framework cannot be easily executed outside of the Google infrastructure (either the production one, or the simulated one provided by Google’s tools for development), and it is almost certain that a brand new version of the backend would be required going forward, so that it could be made independent of the Google dependencies it currently possesses.

6. Evaluation

Evaluation was central to our work. Due to our limited knowledge of the problem domain, small initial sample for surveying, and the naturally occurring unpredictability when adding a technology to an existing context, it was essential that we could test our assumptions in contexts as similar as possible to those where the application would be used.

The possibility of controlled evaluation was also one of the many reasons for limiting our scope to in-class interactions in the initial prototype. Although out-of-class interactions are an essential missing piece, it is however obvious that testing them requires an infrastructure with a modicum of maturity, the creation of testing and monitoring tools able to run 24/7 (since the application has to be available anytime), and professors committing to use our system throughout a prolonged period of time. All of these required an engineering effort we were unable to provide in our limited time.

What we chose to implement had testing advantages out-of-class interaction lacks:

- We could observe users running our application in a controllable environment.
- We could provide users with our own devices, enlarging the testing pool; indeed, we could provide devices to user that did *not* previously own an iPhone or iPod touch, so that we could gauge the reactions of a newbie to the platform and contrast them with those of a previous owner that is already able to parse iOS's visual and gestural language.
- We could make testing span a single lesson, requiring little commitment from teachers and students alike.
- We could easily show the application to professors in simple, one-on-one sessions to gauge their response.

Our testing focused on a limited number of one-on-one, ten minutes informal sessions with teachers, and on three different instances in which the application prototype, at various stages of development, has been provided to users to follow a lecture. In the following, we will focus on the latter, as they were the source of a number of important insights that shaped the final prototype in substantial ways.

6.1 The Logistics of Evaluation

Our in-class evaluations spanned three different dates, courses and two different faculties (and campuses) of the Politecnico. In all, a subset of 40 students participated by either using a device or observing a classmate using a device; of these, 17 actually used the application executing on a device. We presented the students with a survey after (and in one case also before) the lesson, and informally observed their behavior without affecting them while they used the application, except in case of technical problems.

For our evaluation, we used the following devices:

- Three iPhones (of the 3G, 3GS and 4 models respectively) and one iPad in the author's possession. The iPad was only used in a single evaluation.
- Two iPhones which are property of the Politecnico's Hypermedia Open Center laboratory (HOC).
- Students were encouraged, in the days before the lesson took place, to bring their own iPhones and iPod touches to the lecture. These phones were registered with Apple as beta testing devices and used alongside those we brought ourselves.

The evaluations took place as the product was being developed; the prototype was in different stages of completeness. Since the iOS application was the one we felt most critical to the whole experience, we prioritized that over lecturer backend features, and for all three evaluations the upload features of the helper application were entirely unavailable. Therefore, we asked the professors participating in our evaluations to provide us their slide deck in advance of the lecture, and we manually uploaded them to the backend before the lesson began. The helper application was still used, however; not only it did broadcast the presentation to mobile devices during the lesson, but we also used a preliminary version of the outline inference subsystem to produce the JSON payloads expected by our backend. To allow for the partially produced helper application to perform correctly, we did not install it on the lecturer's computer. Instead, we brought a development machine with the helper application installed and the copy of the slide deck, and instructed teachers to use that machine to perform the presentation instead of their usual laptop. The way the helper works, by monitoring the state of the presentation software rather than replacing its UI, allowed teachers to be immediately familiar and productive with the development machine, averting any problems.

We will now examine the three sessions in detail, relating our own observation, the data, and what we did to address issues raised by each session.

6.2 Evaluation one — December 2nd, 2010

Our first evaluation took place on December 2nd, 2010, replacing a lesson of prof. Garzotto's Human-Computer Interaction course to Computer Science Engineering students on the Leonardo campus. Less than 20 students attended the lesson; of these, 7 received a device for testing or had their own device provisioned with the application.

This session was intended to be a test run — not just for the application, which was in a very preliminary stage, but for the entire in-class simulation in general. We decided to follow this outline:

- For the first 15 minutes, we would present the application and provide surveys (shown in Attachment B) to the students, instructing them to complete parts 1 to 3. During the presentation, we offered a small tutorial on the basic functionality of the application.
- Following that, a 15 minutes presentation was held by Prof. Garzotto. The presentation was broadcast by the system; students were encouraged to ask questions and express moods using the application.
  During the presentation, we also informally observed students using the application, in order to evaluate their behavior. We did not interfere with any of their actions, except to provide technical support if needed.
- Once the presentation was done, we allowed 10 minutes for the students to finish Part 4 of the survey.

6.2.1 Evaluation Results

10 of the students decided to turn in their survey at the end of the lesson. Again, the smallness of the pool and the bias in having only Computer Science Engineering students prevented us from thinking observations based on this data as 'general', but again allowed us to spot interesting trends we recount below.

The most interesting results of Part 1 have to do with the availability of technology in those students' life:

- All the ten students characterized their courses as providing them with online material; 6 said "Most (70-100%)" did, whereas 4 said "Some (40-70%)" did.
- 7 out of 10 students are "constantly" connected to the Internet while studying; downloading and reading material (10 and 8 out of 10) are two of the most popular activities performed with it. Only 6 out of 10 use the Internet to pose questions to the professor.
- 7 out of 10 remarked they have an iPhone-like (full-body touchscreen) phone. The remaining 3 indicated they have a traditional smartphone (with a keypad). 7 out of 10 also "always" bring their phone to lectures.
- 5 of the interviewed connect to the Internet "sometimes" with their cell phone while studying (as opposed to 2 that do it "often" and 3 that "never" do it). The place where it's most used is in a study room or another structure built for studying, such as a library (6 out of 10).

- The most popular activity performed with a cell phone is “finding schedules” (6 out of 10). Downloading, reading and annotating material (1, 4 and none out of 10), and asking questions to the professor (2 out of 10) are not activities performed with a cell phone by the group.

In Part 2, we ask students questions about their habits during a lesson. We learn that this group has no qualms asking questions face-to-face during a lesson, either immediately when having a doubt (6), during a break or at the end (5), or privately by approaching the professor at the end of the lesson (7). 8 out of 10 prefer the answer to be given immediately, rather than later.

Additionally, 9 out of 10 take notes during lectures, typically with pen and paper (8). On printouts, the most used kind of annotations are writing on the margins (8) and underlining (7). By far the most popular use for notes and annotated printouts is review during study (9 and 8 respectively).

Part 3 was designed to inquire about student’s initial expectations about interacting with the lesson using software. In particular, students sound intrigued about having questions asked during the lessons on a website (9 out of 10), and would participate actively in discussions on that site (7). Also, students seem divided on using a cell phone to ask questions (5 out of 10 both for “yes, I would use it” and “no, I would not”), with 4 out of 10 preferring asking vocally. Additionally, it is notable that when asked what they would like to do using a cell phone (as opposed to what they actually do, above) 6 out of 10 expressed a desire for the ability to download material, and there’s a slight increase in the number of students who would like to ask questions to the professor, while other activities are substantially unchanged.

Finally, part 4 asks students to describe how they liked the software and how would they think of it becoming a part of their daily life:

- There is a marked increase in the number of students that would ask questions using the app (9 out of 10), with most students putting an emphasis on how they’d like the professor to respond immediately (5 out of 10 rank it as “High” priority) or as the lesson goes on, but not later (5 out of 10 rank it as “Mid-high” priority).
- The application did not significantly change the group’s expectations of what they would like to do with a cell phone (asked in Part 3).
- 9 out of 10 would say the app is “usable”; only 5 out of 10 would characterize it as “useful”, though. Among the features, asking questions and browsing the presentation ranked the highest (7 out of 10 participants said they’d characterize them as “very useful”).

- When asked if the application made the lesson more “engaging”, “effective” or “stimulating”, there was a marked skepticism, with only a minority indicating they’d thought this wholeheartedly. However, we find there is a low number of people who disagree with any of these descriptors (1, none and 2 respectively), most instead choosing to say they “somewhat” agree with them.
- Moods and questions are expressive enough for most interviewees (8 and 9 respectively). When asked if they’d like more information, the majority of those who responded to the question indicated they’d find it distracting (3 out of 10).

This was not all of the data we collected; during our informal observation, we observed behavior that made us rethink a number of UI approaches. We discuss these observation point by point below.

6.2.2 Discussion & Observations

Circumstances we describe below forced us to conclude that this evaluation was, in the end, largely unsuccessful.

This was not entirely unexpected: the application was at a very early stage of development, and we had designed this session to be more like a test run for evaluations than an evaluation proper. However, a number of easily avoidable errors were highlighted by this trial run, causing us to immediately reprioritize work to correct them.

In particular:

- We had not yet deployed the application to the Google production infrastructure, instead relying on the development simulator Google provides, running on the projection laptop. The simulator proved unable to withstand the polling traffic coming from 7 devices, each set with a polling interval of 1.5s, causing noticeable lag that was reported from most testers, and caused massive distractions as users requested assistance for the nonworking application.
- In this evaluation, we provided one of the students with an iPad rather than an iPhone or iPod touch. iPad devices can run regular iPhone applications, either at their original resolution in a small window in the middle of the screen, or by zooming them to fill the entire display. However, the larger screen proved more distracting; the user specifically reported, later, how it was unwieldy to use and how the larger, more immersive screen caused him to ignore the lesson in favor of experimenting with the application. This echoed our initial worries that large-screen devices would attract more attention to themselves (as per [1]), rather than the lesson.

- The survey compilation took a long time, taking more than 20 minutes when including all the four parts.
- Several users, including users not familiar with the platform (whom we had provided with a device ourselves) attempted to perform gesture-based interactions, such as swiping the screen left or right, or rotating the device. The early prototype application did not have any gesture-based interactions available at the time, failing to respond to these inputs; in particular, it was missing the slide image landscape display mode.
- The early prototype application provided limited feedback: once a question was picked, the application waited for server acknowledgment before showing a question indicator in the right margin (which would lag, due to the aforementioned server problem), and picking a mood showed no feedback at all. Users attempting these operations were seen repeating them, unsure if they worked at all.

After this test run, we decided to correct these major issues in several ways:

- We reprioritized backend work; in particular, we started deploying on Google's infrastructure and doing informal load testing by simulating a large number of devices (at one time up to 50) accessing the service from a limited number of IP addresses. This allowed us to highlight several bottlenecks. Additionally, polling intervals were raised to 3s without any appreciable loss in responsiveness, and eager caching behavior in the iOS application was tuned to avoid interfering with live display or polling.
- We decided to no longer use devices larger than iPhones or iPod touches during evaluation, to focus on our intended use case.
- The survey was considerably shortened for subsequent evaluations; we would give students part 4 after the lesson only.
- Swipe and device rotation gesture-based interactions were implemented, including the landscape slide display mode.
- Feedback was subtly reworked: the application now showed a question indicator in the gutter immediately, and a slide-in panel was added for sending moods that confirmed that the mood was being processed.

6.3 Evaluation two — January 12th, 2011

Our second evaluation was very different from the first. We collaborated with prof. Paolo Cremonesi and Dr. Roberto Turrin to bring our system into a real-world lesson of their "Recommender Systems for Interactive Television" course; they would hold a 30' segment of their lecture, involving a slide deck they had prepared, while allowing the system to run, and answer questions accordingly at the end of the segment. This evaluation was held on January 12th, 2011, at the building in via Golgi, 20 of the Città Studi campus, with students of the Computer Science Engineering course. About 40 students were in attendance that day.

We only had an availability of 5 devices, as we decided to no longer use the iPad and very few students brought their own. We observed the application crashing on an additional student-brought device; however, the device had OS-level alterations and appeared to be malfunctioning due to these alterations (for example, where all iOS devices generate detailed crash logs whenever an application aborts, this device generated none). We were unable to replicate this behavior on other, unaltered devices, and did not investigate further.

We followed a different schedule for this evaluation:

- As for the first evaluation, students were asked to bring their own devices in advance. However, some of the devices proved not usable because they were running an old version of Apple iOS, or had received software modifications that appeared to prevent the application from running. All devices with a recent version of iOS and no software modification ran the software without problems, suggesting this was due to the modifications rather than any technical problem of the application itself.
- Device setup was done, as much as possible, before the start of the lesson and the arrival of Dr. Turrin, to avoid interfering with the lesson proper.
- After the lesson started, we explained how the system worked much like we did in the first evaluation. Unlike that evaluation, however, we did not ask user to fill parts 1, 2 or 3 of the survey before evaluation began, shortening this section to roughly 5 minutes. We also encouraged students to try out both asking questions and expressing moods.
- We then allowed Dr. Turrin to hold the slide-deck-assisted segment of his lecture as usual. During this time, we observed students, recording some of our observations on photographs and video. This segment of the lecture took around 30 minutes. We did not interact with or influence students during this segment, except in case of technical problems.
- We presented both Dr. Turrin and the class with the lesson monitoring screen, allowing Dr. Turrin to answer a number of questions asked through the system.
- Finally, we asked participating students to fill in just part 4 of the survey before letting the lesson continue as usual. This took around 10 minutes.

6.3.1 Evaluation Results

The pool of results was again small, with 13 people handing in a completed survey. There were few variations between the first evaluation and this one, except for the much lower number of reported lag incidents; however, we saw some significative variations in certain questions:

- This particular audience seemed more open to the possibility of receiving questions from a website, with 5 ranking it as “High”, in contrast to a single person doing so in evaluation one.
- A significant amount (7 and 8 out of 13) said they would download and read material on their cell phone after application use.
- A lower amount of people found the app “very usable” (4). However, no votes were cast for saying the app was “not usable”, with the rest of the interviewees opting for “somewhat usable”. A notable increase was in the number of people that would not describe the lesson with the app as “more stimulating” (5), with slight increases for the negative response even for the other descriptors. Still, the number of user who that would somewhat or fully agree that the lesson was “more effective” or “more engaging” still outnumber those who don’t.

Our informal observations indicate that many of the frustration factors present in the first evaluation did not return in this one. In particular, users were not seen trying to repeat a just-performed action, and some of them adopted our intended usage model of placing the device on their desk reducing interactions with it to the minimum (figure 6.1). Some students, however, verbally communicated their disappointment at being unable to interact with the application (except for browsing) while it was held in landscape mode, as the prototype at the time had no controls in the slide image display mode other than gestures for backward and forward browsing.

6.3.2 Discussion & Observations

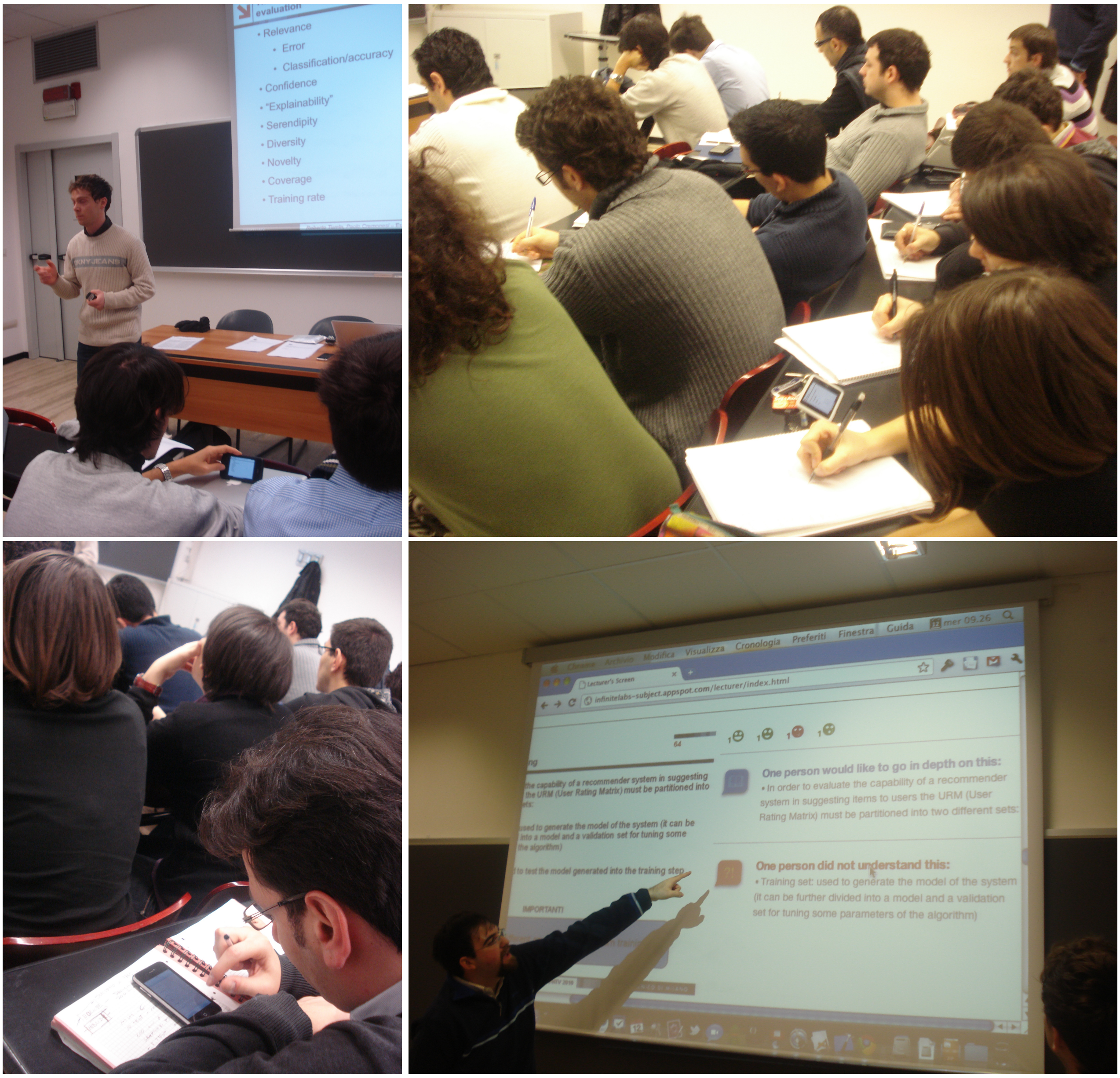

evaluation
Relevance
Error
Classification/accuracy
Confidence
"Explainability"
Serendipity
Diversity
Novelty
Coverage
Training rate
Lecturer's Screen
infinitelabs-subject.appspot.com/lecturer/index.html
One person would like to go in depth on this:
In order to evaluate the capability of a recommender system in suggesting items to users the URM (User Rating Matrix) must be partitioned into two different sets:
One person did not understand this:
Training set: used to generate the model of the system (it can be further divided into a model and a validation set for tuning some parameters of the algorithm)
IMPORTANT

Figure 6.1: From left to right: Dr. Turrin lecturing while a student follows the presentation on his device; the device, unobtrusively placed on the table; a student, interacting with the application; displaying questions on the lesson monitoring screen.

From a technical point of view, this was the first evaluation where the application performed as intended. The lag was eliminated, and no student reported being unable to follow the presentation due to it; indeed, some even adopted our intended model of allowing the app to be "just another tool on their desk" without prompting on our part. However, surveys indicate that the students are still critical of the application, although they again appreciated its two more prominent features (presentation browsing and question asking) and would see value in using the collected data after the lesson, e.g. through a website.

6.4 Evaluation three – January 20th, 2011

A third and final evaluation session was held on January 20th, 2011 as part of the Interaction Design course of prof. Garzotto. Unlike previous evaluations, this one was held at the Design faculty, in the Durando campus (in a different part of Milan than the main offices, and the Engineering and Architecture faculties, of the Città Studi campus, where the first two evaluations were held). This evaluation followed the same pattern as the previous ones:

- We introduced the system to students, including a tutorial, for roughly 5 minutes;
- A lesson was performed, taking approximately 25 minutes.
- The lesson monitoring screen was introduced, and a few questions posed through the tool were explained. This took approximately 5 minutes.
- Finally, students who participated in the experiment completed Part 4 of the survey, requiring about 10 minutes.

Despite a large number (more than 40) of students being in attendance, we again had only 5 devices available for this demonstration; only 16 students provided a survey.

During this evaluation, we again had technical problems that introduced lag in the mobile application. Investigation occurring immediately after the evaluation seemed to identify the culprit in the limited bandwidth available from the Wi-Fi network in that particular building; informal testing revealed how the network could only provide a laptop with 20 to 50 KB/s of bandwidth in that particular moment. This is an anomalous condition for the Politecnico Wi-Fi, but is known to occur in case of network congestion for a particular wireless access point. Students also informally lamented that slowness or unreliability occur with noticeable frequency for the Wi-Fi service on the Durando campus, although we were unable to substantiate this empirically. It is our hypothesis that the limited bandwidth available from the classroom's wireless access point was "taken up" by the eager caching behavior of the application, interfering with other devices receiving updates on the state of the live presentation.

6.4.1 Evaluation Results

Although basic patterns repeat, there are several differences in survey results that we'd like to highlight:

- Students overall found the application made the lesson more "engaging", (somewhat) more "effective", and more "stimulating". In addition, students found the app "usable" (7 very much so, whereas 9 only somewhat), and overwhelmingly agreed that the app was "useful" (12, versus 4 only agreeing somewhat). No student disagreed with any of these descriptors.

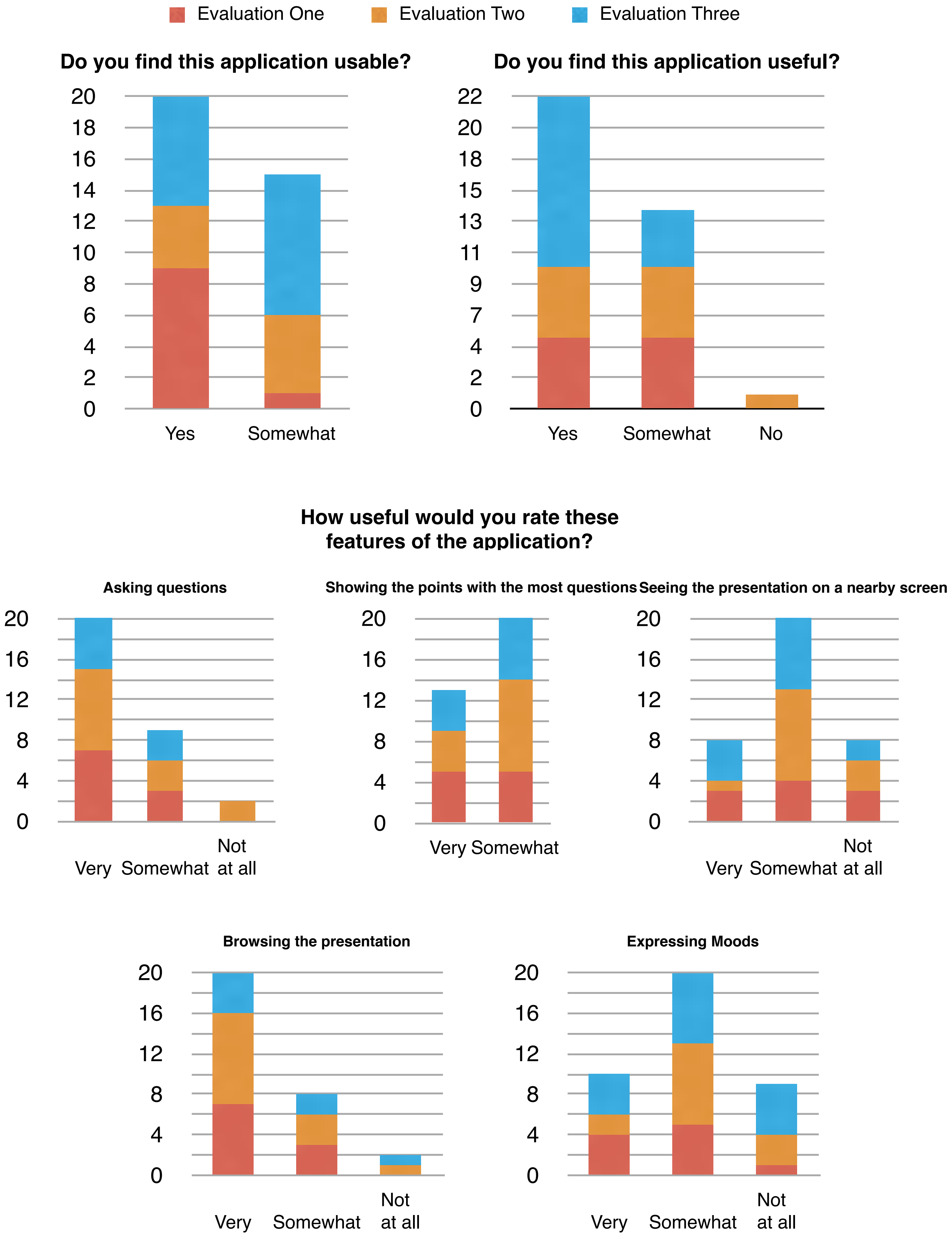
- A higher number of students requested more information to be shown through the mobile application. 5 out of 16 remarked they would have preferred to have access to the text of freeform questions posed by other students.
- There seems to be a somewhat different preference in professor answer timing; whereas Engineering students in the past evaluations preferred an immediate or almost-immediate response, students assigned a higher average priority than Engineering students to having the professor respond at the end of the lesson. Students still prefer the professor to respond within the classroom as opposed to using an online communication system.

6.4.2 Discussion & Final Observations

Again, patterns repeated themselves: users were happy to use the application, although not entirely satisfied with it, and found it a useful addition. We believe we can explain the differences between evaluation surveys by noting that Design students appreciated the idea and the potential ramifications, whereas Engineering students might have looked more closely at the current implementation.

We were happy to see that, in general, after using the application students ranked activities it performed — downloading and reading material, for instance — high among those they said they'd do using a cell phone. Additionally, the majority of the features we asked to evaluate were evaluated as being "very" or "somewhat" useful, again with a particular emphasis on asking questions and browsing the presentation. We hypothesized the former was due to being the centerpiece of interaction between student and app, whereas the latter is appreciated because it brings an innovative element that is not possible during lectures currently without interrupting the teacher entirely.

Students also felt that moods and predefined questions were expressive enough; we were especially impressed by the latter kind, since there are only two predefined question kinds provided by the application. Our hypothesis here is that these two are somewhat "archetypical" — that is, most student questions that do not have a specific issue with the content and are posed during lessons are often a paraphrasing of one or the other question. (Which predefined questions exactly to add was the subject of much debate, and even we were surprised on the relatively low number of possibilities.)



Graph 6.1: Highlights from Part 4 of all three evaluations; application descriptors and per-feature usefulness.

6.5 Informal Presentations

Alongside in-class simulations of actual use, we also presented this application to a number of teachers at the Dipartimento di Elettronica e Informazione (Department of Electronics and Information). Each presentation was one-on-one and informal, introducing the application idea and showing a short lesson simulation where one or two devices followed a presentation displayed on a laptop, adding questions and sending moods. The prototype was shown at different stages of completion, and results from past evaluations were also discussed and remarked upon.

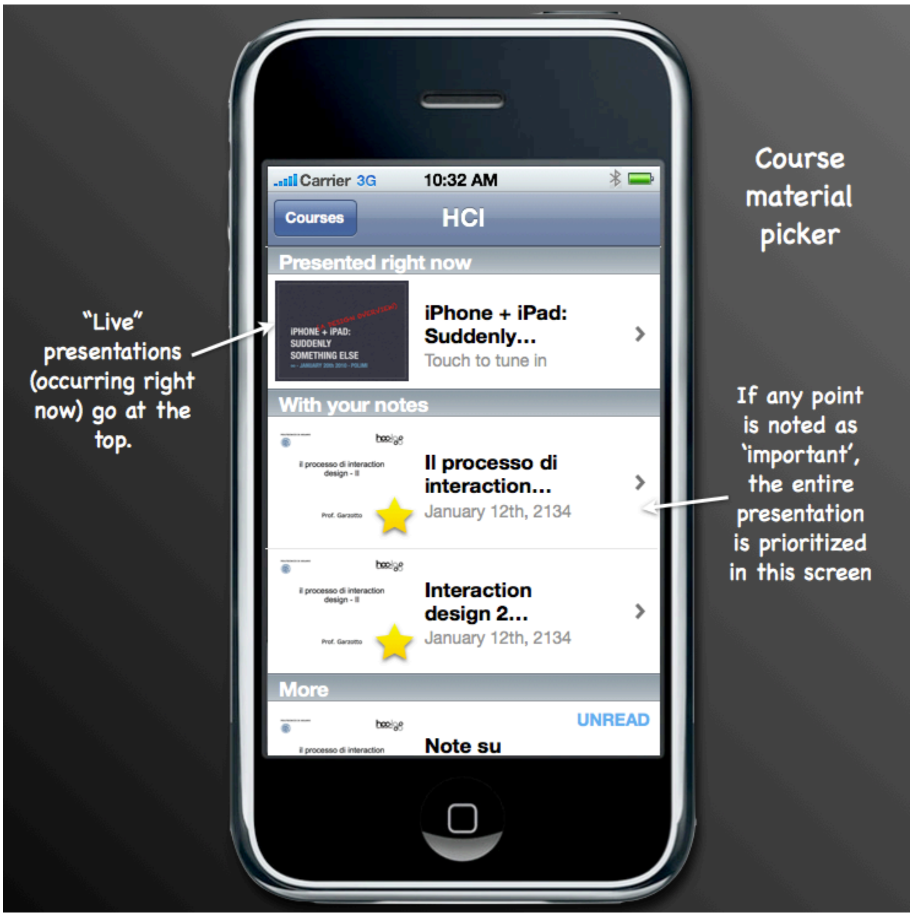
Invaluable observations have come from this activity. Several teachers remarked their possible disappointment at having a system that *only* operated within the class, as they felt it would duplicate existing systems that already “poll” the opinion of the class. Some criticism was also expressed at the fact that our system was “isolated”, so to speak: some professors already deploy Learning Management System or other communication tools (*eg*. web forums) on their own, and felt that the question-and-answer style of communication would be redundant unless integrated with existing tools somehow. However, the possibility of preserving the context of lesson this system has was noted as being something not usually found in systems with similar aims. We also felt that the focus on a satisfying mobile user experience was uncommon, enough to make this project worthwhile in and of itself.

7. Preliminary Design of Additional Functionality

As we mentioned in the introduction, we worked iteratively: design, then build, then evaluate. Our third in-class evaluation, mentioned above, was the last step of our final full iteration on this project in January 2011. However, even though work on the project was reduced, we did not stop; an extra design step was performed before winding down work, in which we speculated on what changes could concretely be achievable in the “next phase” of the project, limiting ourselves to the next few *design-build-evaluate* iterations. We will conclude our coverage of this work by showing and commenting our preliminary mock-ups of those new features we strongly feel should become part of the application; in particular:

- the beginnings of a *material review & study* user interface;
- an extension of the concept of *annotating* points and slides to allow for *private* annotations;
- and a cleaner, more featured web-based user interface for lecturers to extend our “Questions & Answers” and “Metrics” experiments.

In our updated iOS application, the application would feature a home screen containing all available presentations (figure 7.1); this screen shows prominently the “live” lecture, if one is occurring right now, and allows the user to choose what exact material to review. Slide decks in this screen are grouped so that the current lecture, if any, and any slide decks that contain points marked as “important” are displayed prominently. (We will discuss marking points as “important” momentarily.)



Carrier 3G 10:32 AM

Courses

HCI

Presented right now

iPhone + iPad: Suddenly...
Touch to tune in

With your notes

Il processo di interaction...
January 12th, 2134

Interaction design 2...
January 12th, 2134

More

UNREAD

Note su

“Live” presentations (occurring right now) go at the top.

Course material picker

If any point is noted as ‘important’, the entire presentation is prioritized in this screen

Figure 7.1: Slide deck selection screen for a course, with “smart” groupings.

Touching the “live” presentation returns to a screen very similar to that in our final prototype (figure 7.2); however, there are a number of important changes:

- Mood sending has been made easier to understand by reducing the number of moods to only those we judged non-redundant. The user can now express a generally negative, neutral (bored), or generally positive mood. Help text has also been added.

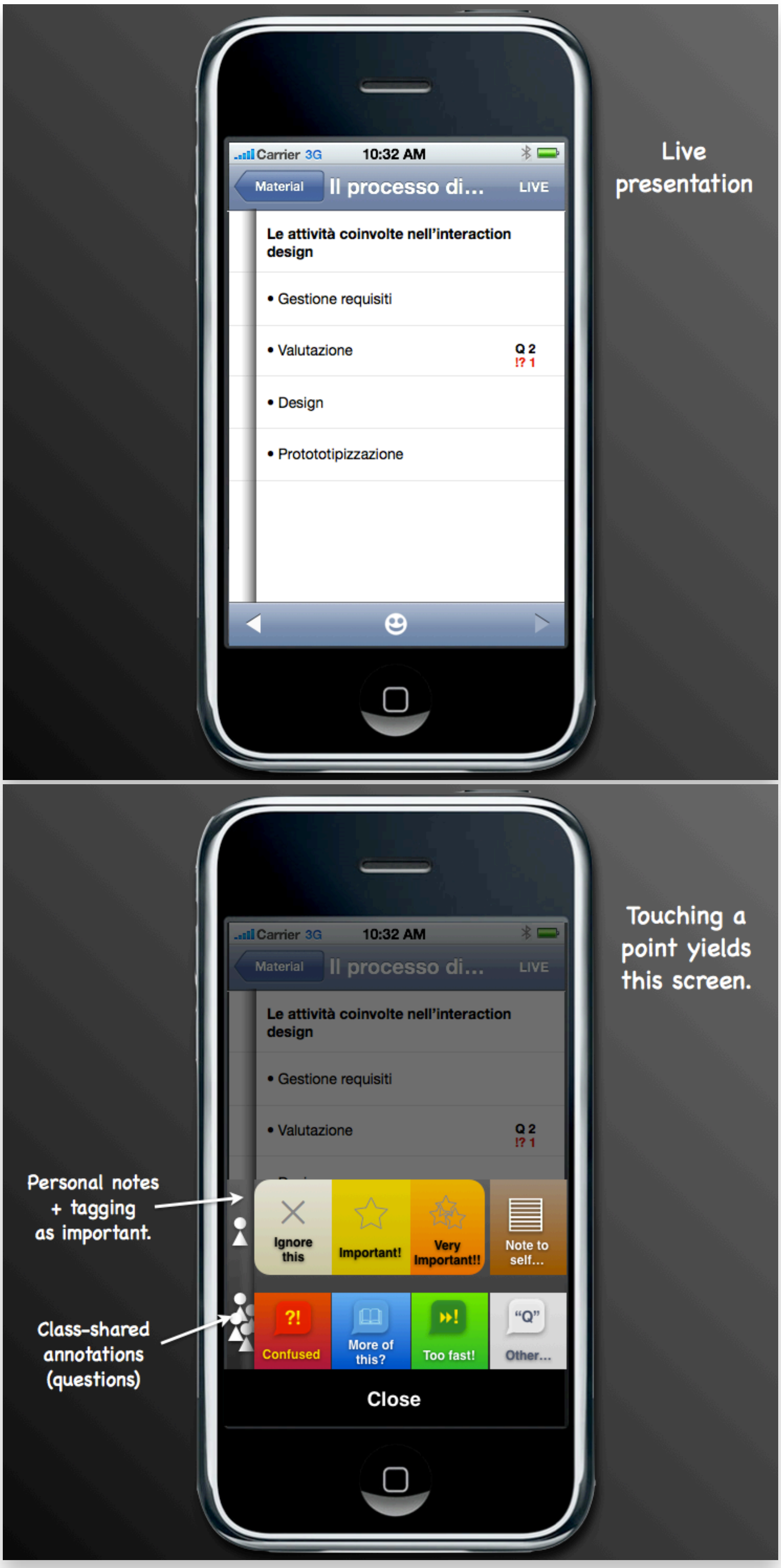
- Touching a point brings up a new iconographic menu that has two rows — an updated bottom row for questions, and a new top row for *personal annotations*. Personal annotations are not automatically shared with other students, and work within the metaphor of “scribbling over the slide” — highlighting, underlining, crossing away, *etc*. In our mock-up, three degrees (other than no highlighting) are provided: “ignore this”, “this is important” and “this is very important”. Additionally, the user can type free-text notes to themselves. Important points are shown as though they were highlighted, whereas ignored points are stricken out and made lighter to reduce their visual prominence.

- As a shortcut to the above, the slide screen prominently features a sunken gutter on the left side. Touching a row in the gutter causes the row to be highlighted (marked as important); touching again causes the row to toggle between highlighted and normal. This allows for one-tap marking of important items, which the application can then use to help prioritize study.

- Rotating the device in landscape mode reveals vertical controls that do not interfere with the slide's content in the corner, and a new "ribbon" that serves the same function as the gutter; like the question and mood controls, it applies an "important" personal annotation to the whole slide.

The user interface changes slightly between slides shown during live presentations and those shown while reviewing already-downloaded presentations. In the latter mode, the interface also provides a way to read the text of free-text questions asked during the lecture, and only allows for private annotations, removing shared-with-class questions and moods. Notes, including those taken previously during the lecture, are entirely preserved, and the presentation is always available regardless of connectivity once downloaded.

Mock-ups of all these functions are shown in figures 7.2 to 7.6.



Le attività coinvolte nell'interaction design

- Gestione requisiti
- Valutazione
- Design
- Protototipizzazione

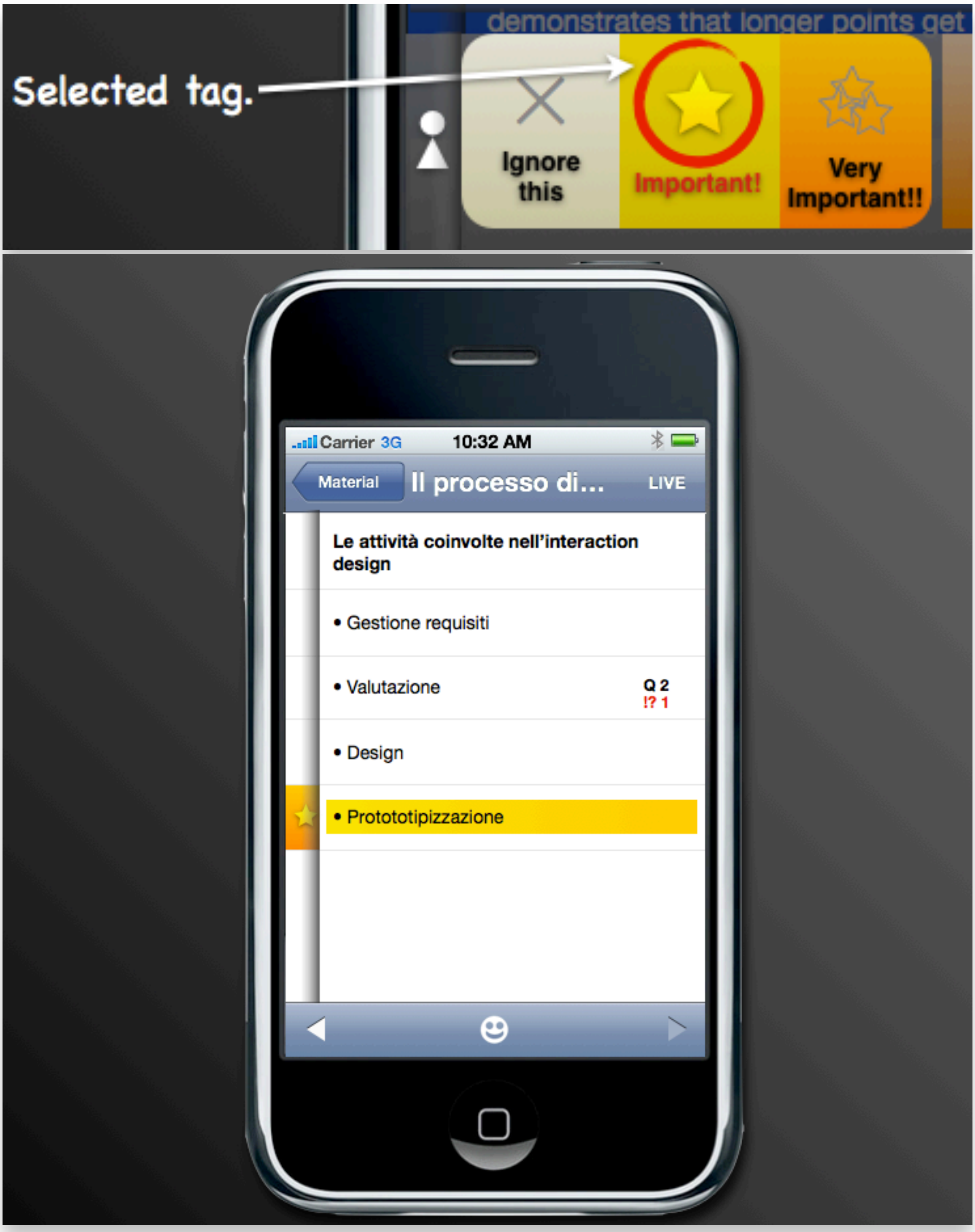
Live presentation

Touching a point yields this screen.

Personal notes + tagging as important.

Class-shared annotations (questions)

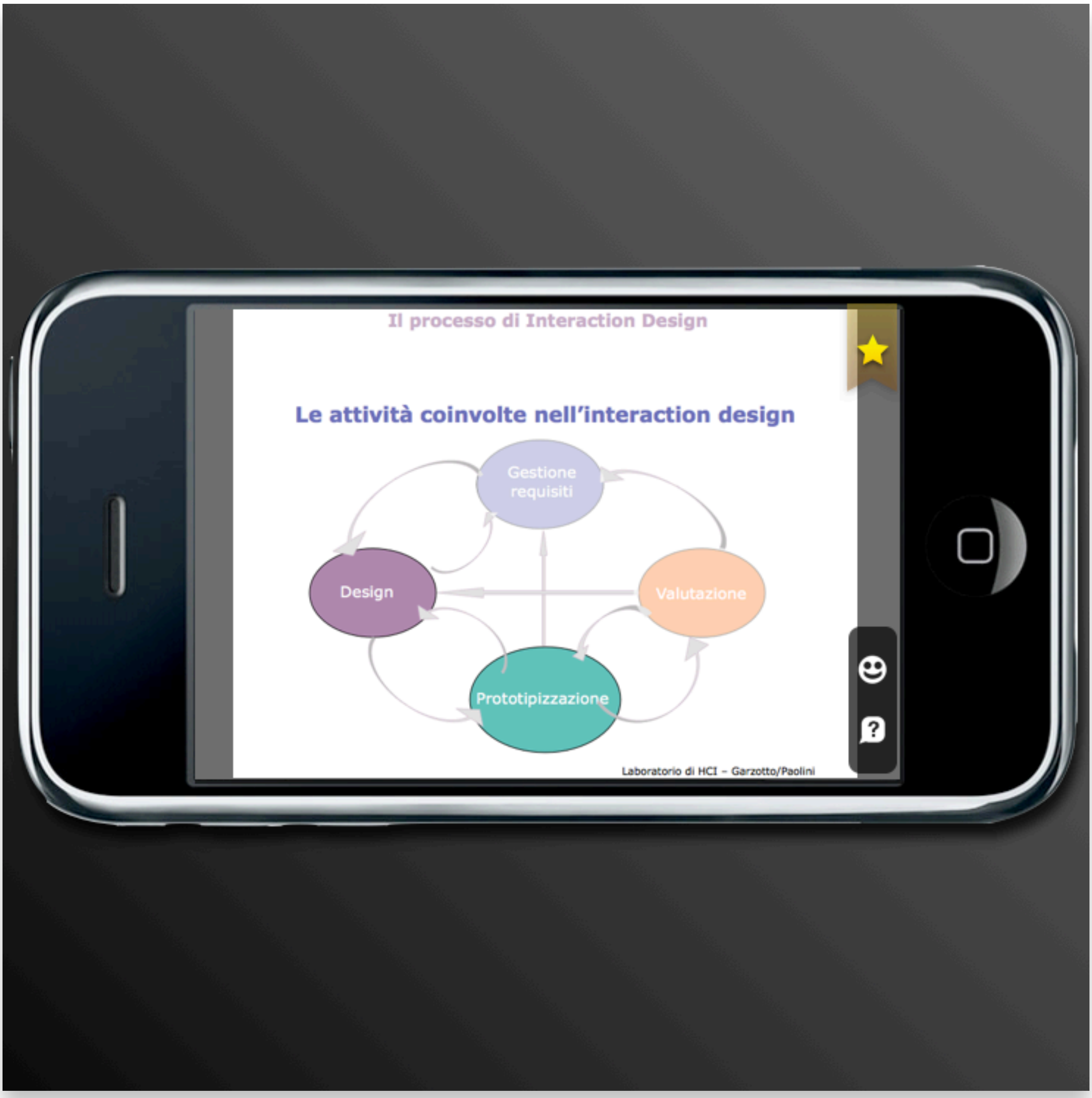
Figure 7.2: New slide display view (with gutter) and new iconographic menu with personal annotation facilities.



Le attività coinvolte nell'interaction design

- Gestione requisiti
- Valutazione
- Design
- Protototipizzazione

Figure 7.3: Selecting "Important" causes the point to become highlighted. Iconography and color indicates the importance of the highlight. Touching the gutter serves as a shortcut to toggle highlight of a point without invoking the iconographic menu.

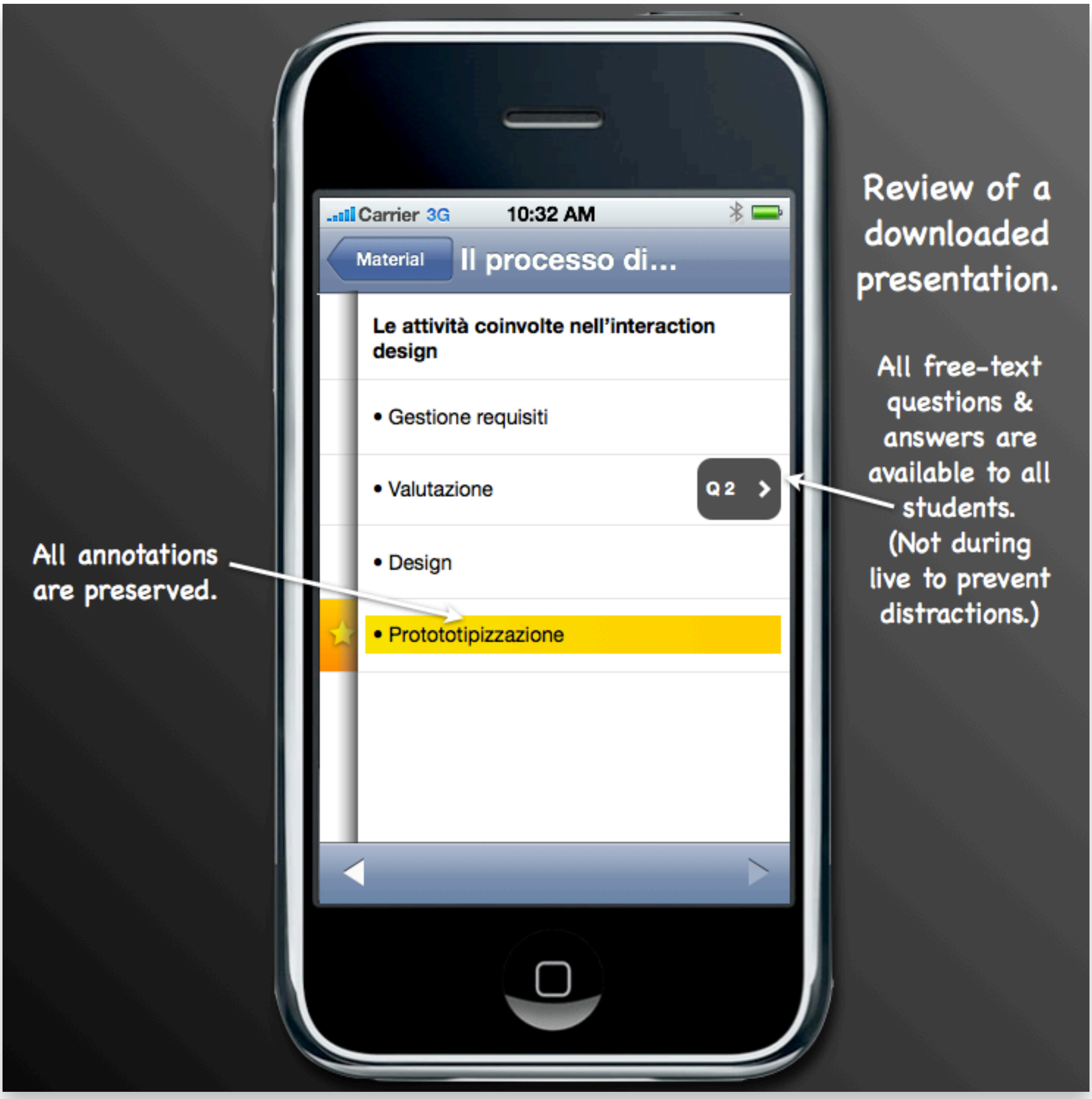


Il processo di Interaction Design

Le attività coinvolte nell'interaction design

Laboratorio di HCI – Garzotto/Paolini

Figure 7.4: In landscape mode, controls are made vertical to avoid content interference. A "ribbon" control allows for quick highlighting of the entire slide as "important".



Le attività coinvolte nell'interaction design

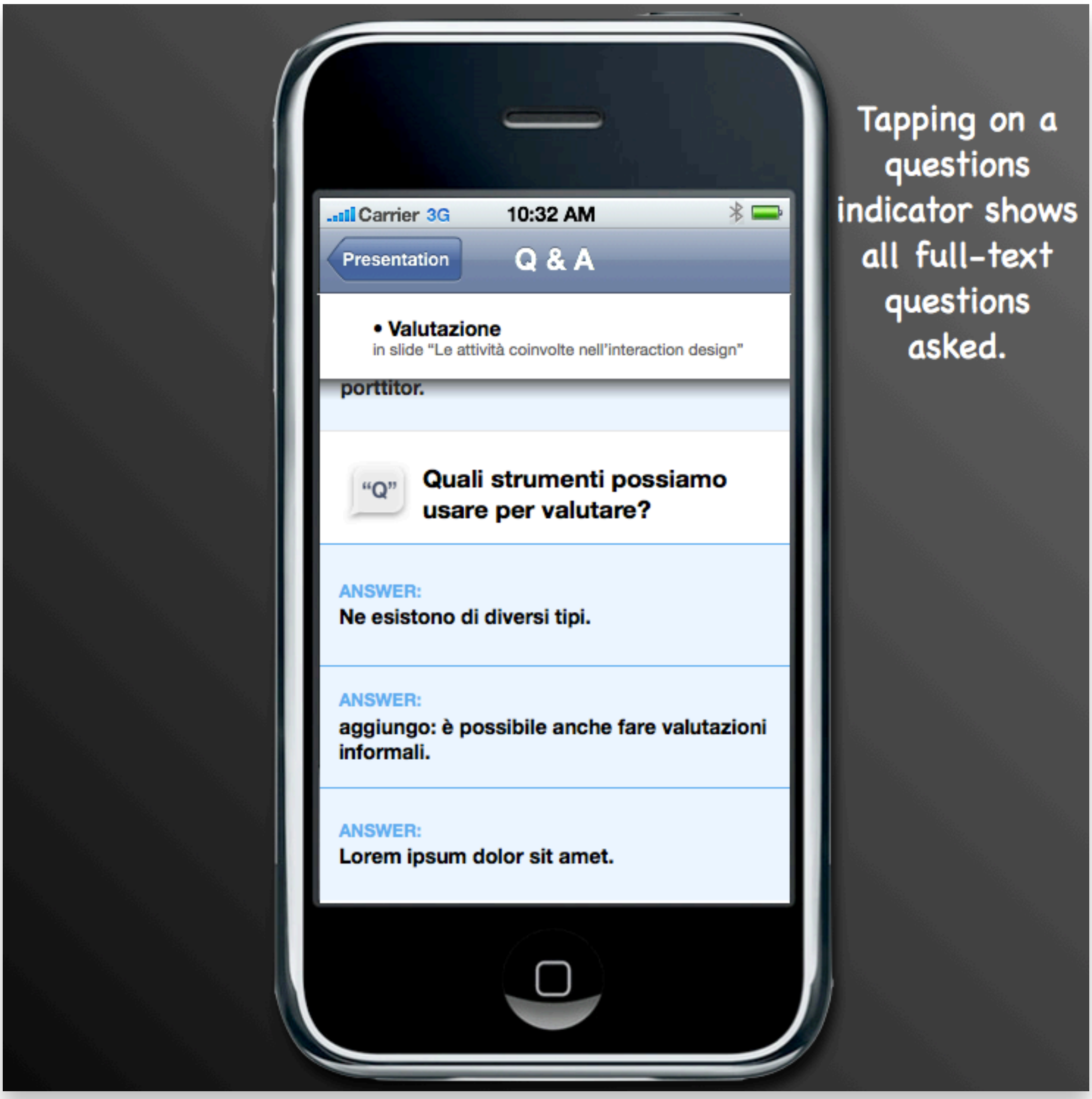
- Gestione requisiti
- Valutazione
- Design
- Prototipizzazione

Review of a downloaded presentation.

All free-text questions & answers are available to all students. (Not during live to prevent distractions.)

All annotations are preserved.

Figure 7.5: Slight variations of the user interface in review mode. The question markers can be touched to show free-text questions and professor-provided answers (coming from the "Questions & Answers" site).



• Valutazione
in slide "Le attività coinvolte nell'interaction design"

porttitor.

"Q" Quali strumenti possiamo usare per valutare?

ANSWER:
Ne esistono di diversi tipi.

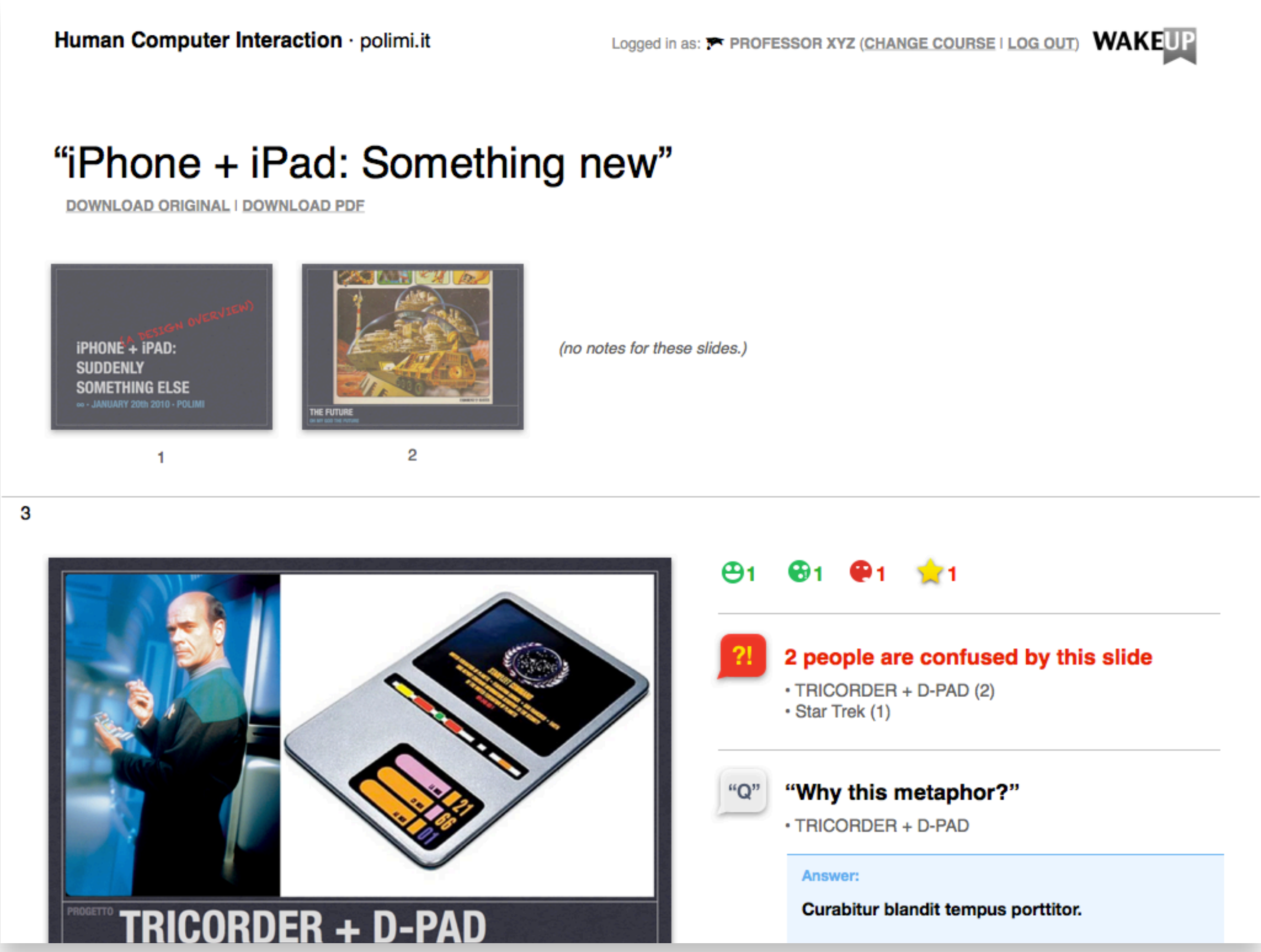
ANSWER:
aggiungo: è possibile anche fare valutazioni informali.

ANSWER:
Lorem ipsum dolor sit amet.

Tapping on a questions indicator shows all full-text questions asked.

Figure 7.6: Questions & Answers view for a point.

For lecturers, we would instead focus on a better user interface for controlling the presentation, both during and after the lecture. We would merge the current lesson monitoring screen and the section known as "Questions & Answers" into a single entity, which shows all notes and allows posting answers to free-text questions in real time; it would also show statistical information on points the users marked as "important", as a new marking alongside moods. We would also shift the current native user interface to a management screen from the lecturer web site, allowing for upload management and starting a presentation directly from the web browser. These features are shown in figures 7.7 and 7.8.



Human Computer Interaction · polimi.it

Logged in as: PROFESSOR XYZ (CHANGE COURSE | LOG OUT) WAKEUP

"iPhone + iPad: Something new"

DOWNLOAD ORIGINAL | DOWNLOAD PDF

1

2

(no notes for these slides.)

3

☺1 ☺1 ☹1 ★1

?! **2 people are confused by this slide**

- TRICORDER + D-PAD (2)
- Star Trek (1)

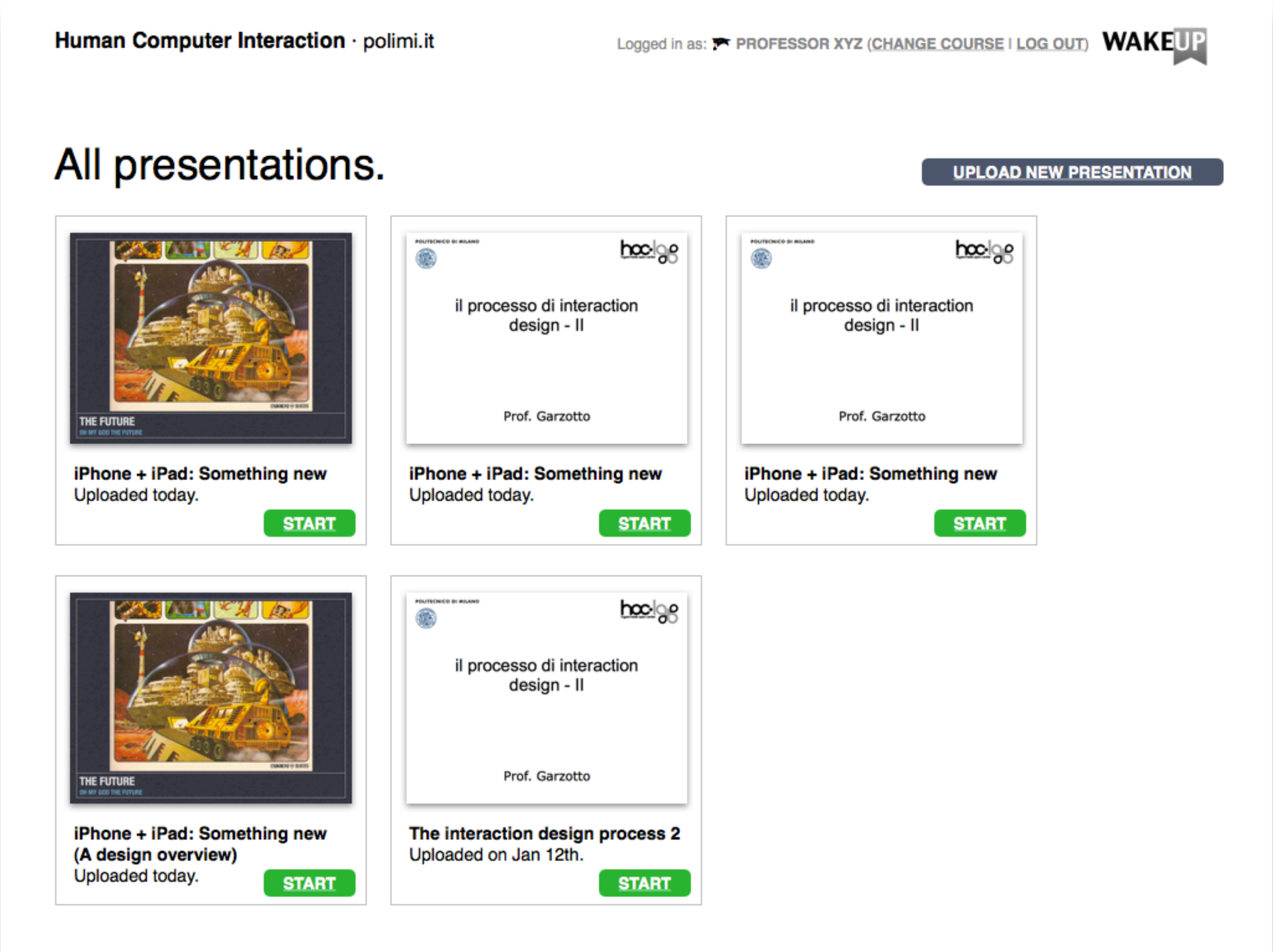
"Q" **"Why this metaphor?"**

- TRICORDER + D-PAD

Answer:

Curabitur blandit tempus porttitor.

Figure 7.7: The new lesson monitoring screen, streamlined.



Human Computer Interaction · polimi.it

Logged in as: PROFESSOR XYZ (CHANGE COURSE | LOG OUT)

All presentations.

iPhone + iPad: Something new
Uploaded today.

iPhone + iPad: Something new
Uploaded today.

iPhone + iPad: Something new
Uploaded today.

iPhone + iPad: Something new
(A design overview)
Uploaded today.

The interaction design process 2
Uploaded on Jan 12th.

Figure 7.8: The new "All presentations" screen, showing 'Start' buttons for all presentations. Clicking on a presentation opens tools, such as the monitoring screen or the "Measures" page.

We believe these changes are steps in the right direction. They may appear rather limited, but we prefer making small steps and evaluating them, opening to more complex scenarios (such as full communication between students, or integration with preexisting learning management systems) once we are sure of the changes we have brought. This, we hope, will help us produce a system that avoids the pitfalls often associated with such efforts — complexity, lack of use, or inattention to overall user experience.

8. Conclusions

When we started this project, we didn't know exactly what we'd end up with, or how useful the end result would be. However, we did know that, whatever the approach we'd chosen to take, we'd only find acceptable to build software that is approachable and may have utility. We can't definitely prove the latter, but we feel that the data we collected vindicates us on the former, and that the project, while incomplete, is a good platform to build upon — not just as a proof of concept, but even as a foundation for later production work.

The surveys, in particular, indicated that students:

- are interested in the system (even just in the very small subset of the intended we've been able to build);
- see some value in our application, especially in regards to better interaction with their professor and the ability to browse the lesson's slides at their leisure;
- claim to be eager to participate in a online extension of our existing system.

In addition to this, we were pleased with our own informal observations of user behavior during evaluations: rarely students seemed stumped, and none of them required any assistance aside from our short and purposefully very high-level tutorial. Through what we attribute to a careful, minimal design, many of them naturally adopted the workflow we designed: having the device on the desk (possibly propped-up in landscape, to resemble a small passive screen), using it momentarily during lesson, with little distraction. We feel justified in saying that, in spite of limited time and a very small scope, the project was overall successful in its aims of user acceptance, increased engagement and in the performance and soundness of design of its technological components.

Finally, we think that our system is original in respect to many other described in literature, in at least three important areas:

- Student involvement not just as an afterthought or final validation, but as *an integral, iterated part of the development process*. This is atypical even in larger software projects, where testing is often performed internally (by a quality assurance team) due to the relative cost and effort involved in user testing. We felt this approach was justified, as it allowed us to enter a tight feedback loop, reducing the time to an acceptably-performing system.
- On the same note, a singular focus on the *student*. It is not our desire to limit our "m-learning" system to a single context or even to just learning in the strictest sense of instances of studying and class attendance. Our final aim is to produce a set of integrated *tools* the user can employ to study in an easier, assisted way — tools that

manage material, prioritize what to study, allow the student to convert "wasted time" into useful review time, and potentially even serve as a logistical aid or, in the far future, integrate with existing learning management systems and organizational services to provide what is needed for all university-related tasks that are useful in context.

- Speaking of *context*, our intended system also avoids another staple of m-learning, that of being *tied* to a particular place or time of use. We want the application to be *always* useful (for instance, allowing the user to review material wherever he is), although to be so we also take the current context into account (for instance, synchronizing with a lesson if opened during one); our application is therefore *context-sensitive* but never *context-constrained*.

Despite all these important results, we however cannot say the project has succeeded in all its aims. In particular, professors we informally interviewed underlined their concerns about the actual overall usefulness of the system, comparing it to similar systems that they felt did not add much to their lessons; we did not address these concerns, therefore possibly causing an acceptance problem from the *other* stakeholders we wanted to involve. In addition, we can also hypothesize there might be the psychological and management problem of adopting "yet another system", in addition to the plethora both the university and single faculties require to use. Any future work would need to address this, as we'll discuss below.

Another problem lies in our approach: the reduced scope did indeed help to meet certain expectations, but — obviously — resulted in only a minimal subset of our intended functionality being implemented; in particular, the prototype does little to nothing to address the problem of aiding study and review *outside* the classroom — we can deliver the material, but not help the student make use of it.

Therefore, our work isn't finished. If we are to bring this software outside the "proof-of-concept" stage, we have to face a number of issues that in our early prototypal stage we could afford to ignore, including:

- study aiding: helping the student review material, annotate it for personal use (as opposed to sharing questions with the class) and prioritize the most interesting, controversial (or otherwise annotated) material;
- support for a broader number of mobile devices and/or other channels (such as the web) that may make sense for students;
- lecturer aid, and a much deeper look at the teacher side of a course experience to identify and address their specific needs as well;

- support for other forms of material that are appropriate for mobile devices, as well as logistical information (such as course schedules);
- long-term effects on student performance, to check whether the application is actually beneficial;
- software installation, delivery and updating for those parts of the system that are not managed (such as the lecturer helper application);
- support and help to users;
- the actual hosting of the backend server software, including any necessary work that needs to be performed on measuring and reducing bandwidth consumption, ensuring availability, *etc*.

None of these is a particularly trivial task, and each of these has the potential to impact the user experience potentially in a deep way. Therefore, any future work involving these tasks must be handled with care, ensuring — as we did for the first few steps — that small changes with quantifiable effects can be applied to produce, step by step, a more usable, more reliable and more useful system.

Attachment A: Requirements Survey

This is the text of the requirements elicitation survey mentioned in the “Eliciting Requirements” section. The survey was originally in Italian, but is translated in English here for consistency. Below the survey, highlights from the results can be found.

This is a very quick survey on your study habits, your lesson participation habits and the tools you use, both during a lesson (e.g., to take notes) and during review afterwards.

Almost all questions are optional. (Required questions are marked with an asterisk and are only found in Part 1.) If you don’t want to answer a question, just skip it.

The survey will remain entirely anonymous.

A.1 Questions about you (Part 1)

- Which academic year are you studying for? *
  - ○ First
  - ○ Second
  - ○ Third
  - ○ Late (three-year course or bachelor’s degree)
  - ○ Fourth (First year of master’s degree)
  - ○ Fifth (Second year of master’s degree)
  - ○ Late (more than two years in a master’s degree)
- Are you a student of Politecnico di Milano? *
  - ○ Yes, I study at Politecnico di Milano
  - ○ No, I’m enrolled elsewhere
- In what kind of school or faculty are you enrolled in? *
  - ○ Engineering School
  - ○ Architecture School
  - ○ Design School
  - ○ Other: …

- What course do you follow in your faculty? (For example: Computer Science, Telecommunications, MBA, Chemistry…) *
  - ○ _______________

A.2 In the classroom (Part 2)

- How long do you generally spend in activities at your university? (Pick the actual time you spend in university structures, regardless of your course schedule.)
  - ○ Almost all the time (every day, 5 or more hours per day)
  - ○ A lot of time (almost every day, or less than 5 hours per day)
  - ○ A little time (fewer than half of the days of the week, or less than 3 hours per day)
  - ○ None (I never come to the university)
- What kind of activity do you attend at the university? (Pick all that apply.)
  - ○ Lessons
  - ○ Guided exercises
  - ○ Hands-on labs
  - ○ Other: …
- If you attend lessons: Among these activities, what do you do during a lesson? (Pick all the activities you do often; disregard those you do rarely or not at all.)
  - ○ I take notes.
  - ○ I record the professor's voice.
  - ○ I highlight or annotate, or just follow, on the textbook.
  - ○ I highlight or annotate, or just follow, on the professor's material (e.g., study notes, slide printouts).
  - ○ I ask questions to the professor.
  - ○ I ask questions to my fellow students.
  - ○ I don't follow the lesson.
  - ○ Other: …
- If you take notes: What do you take notes with? (Pick all that apply.)
  - ○ Pen and paper

  ○ Keyboard (e.g. on a laptop)

  ○ Other: …

- If you take notes: Have you ever had problems using or re-reading them? (Pick all you noticed.)

  ○ I “lose parts”, can’t write everything down.

  ○ I can’t use them because I lend them to other students

  ○ They’re disorganized (I often have to reorder them, and sometimes I lose them).

  ○ I don’t re-read notes, I only use them as an immediate memorization aid.

  ○ Other: …

- If you record the lesson: Have you ever had problems with making or later using your recording?

  ○ The professor won’t allow recording.

  ○ I can’t comprehend the recording due to room acoustics.

  ○ The recording is too long, it’s difficult to find highlights.

  ○ Other: …

- Sometimes, professors try to keep the lesson interesting by using their lecturing time to do things other than a traditional lesson. Do your courses’ lecturers use any of the following during their lectures? Which ones? (Pick all that apply.)

  ○ Videos or music

  ○ “Polling” the classroom

  ○ On-the-fly testing of audience

  ○ Answering student’s questions

  ○ Other: …

- Do you like these variations during a lesson? (If you don’t, skip this question.)

  ○ Yes, they make the lesson more memorable

  ○ Yes, they break up the rhythm and help keep my attention

  ○ Other: …

- What do you dislike of these variations during a lecture? (If you like them, skip this question.)

○ I don’t like them, it’s embarrassing to stand out in the classroom

○ I don’t like them, they’re distractions from the lesson proper

○ Other: …

- Do you usually have a cell phone or a similar device with you during lessons? If so, which one? (Pick all of these that apply.)

○ Normal cell phone

○ Smartphone (no touchscreen)

○ Smartphone (with touchscreen; e.g. iPhone)

○ Laptop

○ PDA (or iPod touch)

○ Tablet

○ Other: …

- If you could use your cell phones to ask questions, and the professor then responded to these questions, would that make asking questions easier?

○ Yes, I would feel more at ease asking questions

○ Nothing much would change

○ No, I prefer an immediate response

○ Other: …

- If questions you send this way would be available after the lesson is over (e.g. on a web site), would you ask them?

○ Yes, I would ask more of them

○ Yes, but this wouldn’t change how many questions I ask

○ No, I don’t want my questions to be recorded

○ No, I wouldn’t have asked using a cell phone anyway

○ Other: …

- Do you often share your notes? If so, how?

○ No, I don’t share my notes

○ Yes, I physically give them to others

○ Yes, I send copies through the Internet to others

○ Other: …

- If you could share automatically your notes with others during a lesson, would you do so?

  ○ Yes, I would share

  ○ No, I wouldn't share

  ○ Other: …

A.3 Your study habits (Part 3)

This is the last section of this survey.

- In what locations do you usually study? (Pick all those where you go often.)

  ○ At home

  ○ In the library, or in a study room

  ○ In another public place (e.g., a park, a pub)

  ○ Other: …

- What do you usually do while studying? (Pick all those you do most often.)

  ○ I review the textbook

  ○ I review the professor's material (e.g. study notes, slide printouts)

  ○ I review my notes

  ○ I review other people's notes

  ○ I do exercises taken from the textbook

  ○ I do exercises my professor gave me (e.g. from study notes or mock exam sheets)

  ○ I invent and do my own exercises

  ○ I discuss the subject with fellow students (in a group)

  ○ Other: …

- What do you bring with you while you study? (Pick all that apply.)

  ○ Laptop

  ○ Cell phone

  ○ PDA (or iPod touch)

○ Tablet

○ Other: …

- Can you access the Internet while you study? (Pick all that apply.)

○ No, I can’t access the Internet.

○ Yes, I can connect using my home connection.

○ Yes, I can connect through my cell phone

○ Yes, I can connect through my laptop (and a cell phone network USB key)

○ Yes, I can connect through my laptop via the network of the place where I study (cable, Wi-Fi, etc.)

○ Yes, I can connect through my cell phone via the network of the place where I study (Wi-Fi, etc.)

○ Yes, I can connect through computers made available by the place where I study

○ Other: …

- If you access the Internet or use a computer while you study, or while you’re preparing to study, what do you use them for? (Pick all that usually apply.)

○ I download study notes or other material

○ I read study notes, books or other material

○ I annotate study notes, books or other material

○ I ask questions the course’s professor

○ I check study and exam schedules for the course

○ I organize my study group

○ I plan what to study (e.g. by scheduling on a calendar)

○ I research the course’s topic

○ I ask questions to other students (e.g. via a forum)

○ Other: …

- If you could do any of the above through a cell phone, what actions do you think would be most useful to have? (Pick all those that you think would be useful, even if you don’t usually perform them.)

○ Downloading study notes or other material

○ Reading study notes, books or other material

○ Annotating study notes, books or other material

○ Asking questions the course's professor

○ Checking study and exam schedules for the course

○ Organizing my study group

○ Planning what to study (e.g. by scheduling on a calendar)

○ Researching the course's topic

○ Asking questions to other students (e.g. via a forum)

○ Other: …

- Would you be interested in trying a cell phone application that does one or more of these things?

○ Yes, I would be interested

○ No, I wouldn't be interested

○ I don't know

A.4 Survey Results

The survey ran until September 2010; we collected a total of 20 survey responses. For simplicity, we omit the (rare) use of “Other” in the questionnaire answers. All answers (except those in Part 1) were optional, and many allowed more than one answer.

What year are you studying for?		***Enrolled at Politecnico?***		***In what school?***		***How much time do you spend at the campus?***	
First year	4	Yes, I’m enrolled at Politecnico di Milano	11	Engineering	12	Almost all the time	5
Second year	2	No, I’m enrolled elsewhere	13	Architecture	1	Some of the time	9
Third year	3			Design	2	A little of my time	5
Late (three-year course or bachelor’s degree)	4					None	0

Fourth (First year of master’s degree)	0						
Fifth (Second year of master’s degree)	4						
Late (more than two years in a master’s degree)	7						

Table A.1: Questions 1-4.

What activities?		*What do you do during a lesson?*		*What do you take notes with?*		*What problems do you have with notes?*	
Lessons	20	I take my own notes	20	Pencil and paper	19	Can't write it all.	11
Guided exercises	11	I record the professor	0	Keyboard (eg. laptop)	7	Away lent to others	1
Hands-on labs	12	I follow professor material (slides)	12			Disorganized	6
		I ask questions to other students	3			Not re-read	9
		I ask questions to the professor	4				
		I don't pay attention	2				

Table A.2: Questions 5-8.

What problems do you have with lesson recordings?		*What variations do you see during lessons?*		*How do you like variations?*		*How do you dislike variations?*	
Not allowed	0	Videos/ music	9	Make lesson more memorable	12	Embarassing to stand out	6
Incomprehensible	7	Polling	8	Help keep attention	14	Distracting	3
Too long	5	Oral tests	5				
		Answers to questions	20				

Table A.3: Questions 9-12.

Do you have any of these with you?		***Would you ask more if you used a cell phone to do so?***		***Would you ask if questions were recorded on a website?***		***Do you share notes?***	
Feature phone	8	Yes	4	Yes	10	No	5
Smartphone (with keypad)	2	No change	11	No change	10	Physically	4
Smartphone (with touchscreen)	12	No	9	No, privacy	1	Via Internet	15
Laptop	10			No, no cell phone questions	3		
PDA or iPod	0						
Tablet	0						

Table A.4: Questions 13-16.

Would you share automatically ?		***Where do you usually study?***		***What do you do while studying?***		***What do you have with you while studying?***	
Yes	19	Home	22	Textbook	19	Laptop	20
No	4	Library, study room	13	Professor material	20	Cell phone	21
		Public place	2	My notes	19	PDA	2
				Others' notes	12	Tablet	1
				Exercises from textbook	7		
				Exercises from professor	14		
				Invent my own exercises	1		
				Discuss subject with others	8		

Table A.5: Questions 17-20.

Can you access the Internet while studying?		***What do you use the Internet for?***		***What among these things would you do on a cell phone?***		***Would you try an app that does any of these?***	
No	1	Download material	19	Download material	12	Yes	21
At home	17	Read material	17	Read material	14	No	1
Cell phone	12	Annotate material	5	Annotate material	6	Don’t know	3
Laptop (cell modem)	6	Ask questions	3	Ask questions	7		
Laptop (Wi-Fi)	10	Check schedules	16	Check schedules	15		
Cell phone (Wi-Fi)	13	Organize study group	8	Organize study group	6		
Computers at study place	3	Plan what to study	7	Plan what to study	10		
		Research subject	14	Research subject	9		
		Ask questions to other students	5	Ask questions to other students	8		

Table A.6: Questions 21-24.

Attachment B: Evaluation Survey

This is the text of the survey mentioned in the "Evaluation" chapter. The survey was originally in Italian, but is translated in English here for consistency. Below the survey, highlights from the results can be found.

This is a short survey on your study habits, both inside and outside the classroom. Your answers will help us design innovative services for students.

The survey is made of four sections and only requires a few minutes. It will remain anonymous.

Thanks for your help!

B.1 Part 1

- 1.1 What faculty are you currently enrolled in?
  - ○ Engineering (school of __________________)
  - ○ Design (school of __________________)
  - ○ Architecture (school of __________________)
  - ○ Other: …
- 1.2 What year are you enrolled in?
  - ○ Bachelor's: First year
  - ○ Bachelor's: Second year
  - ○ Bachelor's: Third year
  - ○ Bachelor's: Late
  - ○ Master's: First year
  - ○ Master's: Second year
  - ○ Late
- 1.3 How long, on average, do you attend activities at the campus?
  - ○ 30-40 hours
  - ○ 20-30
  - ○ 10-20

○ 5-10

○ Less than 5

○ None

- 1.4 Do courses you attend provide you with material through the Internet (such as study notes or slide decks)?

  ○ All or most (100-70% of them)

  ○ Only some (70-40%)

  ○ Very few (40-20%)

  ○ Less than 20%

- 1.5 How long do you remain connected to the Internet, through a desktop or laptop computer, while studying (for didactic purposes)?

  ○ Constantly (I must remain connected, as I study almost exclusively online)

  ○ Enough (More online material than pen and paper)

  ○ Sometimes (I only connect when I need specific material or information)

  ○ Never (no Internet while studying)

  ○ Other: …

- 1.6 If you use the Internet through a desktop or laptop computer for didactic purposes, what do you do on it?

  ○ Finding out course and exam schedules

  ○ Finding classrooms or study rooms

  ○ Looking for information on Politecnico-organized events (e.g. seminars)

  ○ Downloading material, such as study notes or books

  ○ Reading material, such as study notes or books

  ○ Annotating material, such as study notes or books

  ○ Asking for information or questions to the professor

  ○ Organizing my study group

  ○ Planning what to study (e.g. through a calendar)

  ○ Researching the subject

  ○ Asking questions to others in my course (eg. through a forum)

○ Other: …

- 1.7 Do you own a cell phone or PDA?

  ○ No

  ○ Yes, a feature phone (calls and SMSs only)

  ○ Yes, a smartphone with a keyboard

  ○ Yes, a smartphone with a touchscreen

  ○ Yes, an iPod touch

  ○ Yes, an iPad

  ○ Yes, another PDA

  ○ Other: …

- 1.8 If you have it, do you bring the smartphone or PDA with you to lectures?

  ○ Yes, often

  ○ No

  ○ Sometimes

- 1.9 If you have a cell phone or PDA, what brand of phone is it?

  ○ Apple (iPhone)

  ○ Nokia (model: _____________)

  ○ HTC (model: _____________)

  ○ Samsung (model: _____________)

  ○ Other: …

- 1.10 Do you connect to the Internet on your cell phone or PDA?

  - Yes, through a data plan with my carrier

    ○ Regularly

    ○ Occasionally

  - Yes, through a data plan with my carrier

    ○ Regularly

    ○ Occasionally

  ○ No

○ Other: …

Please answer the following questions only if you own a smartphone or PDA and you use it to connect to the Internet.

- 1.11 If you have it, do you connect to the Internet on your smartphone or PDA while you study?

  ○ Yes, often

  ○ No

  ○ Sometimes

- 1.12 If you connect to the Internet via a smartphone or PDA, where do you do so?

  ○ At home

  ○ At the library or in a study room

  ○ At the pub

  ○ In a park

  ○ On a train/On public transport

  ○ Other: …

- 1.13 If you connect to the Internet via a smartphone for didactic reasons, what of the following do you do, if any?

  ○ Finding out course and exam schedules

  ○ Finding classrooms or study rooms

  ○ Looking for information on Politecnico-organized events (e.g. seminars)

  ○ Downloading material, such as study notes or books

  ○ Reading material, such as study notes or books

  ○ Annotating material, such as study notes or books

  ○ Asking for information or questions to the professor

  ○ Organizing my study group

  ○ Planning what to study (e.g. through a calendar)

  ○ Researching the subject

  ○ Asking questions to others in my course (eg. through a forum)

  ○ Other: …

B.2 Part 2

- 2.1 What of the following ways to keep the lesson lively do you think is interesting to you? Rank them “Very interesting”, “A little interesting” or “Not interesting”.

  ○ Videos

  ○ “Live” demonstrations

  ○ Classroom polls

  ○ Questions to students

  ○ Answering student questions

  ○ Other: …

- 2.2 Assume your professor is available to answer students’ questions during a lecture. What of the following do you do?

  ○ Raise your hand and ask during the lecture whenever you have doubts

  ○ Raise your hand and ask during the lecture when the professor asks if anyone has questions

  ○ Raise your hand and ask at the end of the lecture or at the break

  ○ You wait for someone to ask a question, then you ask too

  ○ You queue in front of the podium at the end of the lesson to ask

  ○ You never ask during, or just after, lessons

  ○ You ask by e-mail

  ○ You ask through the course’s forum (assuming it exists)

  ○ You ask by taking an appointment with the professor

  ○ You never ask, ever

  ○ Other: …

- 2.3 If you ask questions during a lesson, when do you prefer to receive an answer?

  ○ Immediately

  ○ As the lesson ends

  ○ Even after the end of the lesson (eg. via e-mail or forum)

  ○ Other: …

- 2.4 If you never ask questions “publicly” to the professor, why is it so?

○ It’s embarrassing to ask in front of others

○ I can’t come up with questions on the spot

○ It feels like a waste of time

○ Questions only come up while studying or reviewing

○ Other: …

- 2.5 Do you take notes during a lesson?

○ Yes

○ No

- 2.6 If you do, how do you take notes during a lesson?

○ Pen and paper

○ Digital tools (eg. laptop)

- 2.7 How do you use notes after a lesson?

○ I reread them as I study

○ I scan them to save them in my computer

○ I retype them at the computer

○ I make photocopies for my friends

○ I give the originals to my friends

○ I send them to my friends through the Internet

○ I don’t use notes afterwards; I only use them to aid memorization during a lesson

○ Other: …

- 2.8 If you annotate books, study notes or slides on paper during a lesson, what kinds of annotations do you make?

○ Underlining

○ Highlighting

○ Doodles (eg. circles, arrows) on the page

○ Written notes on the gutter

○ Drawings (eg. diagrams) on the gutter

○ Folding a corner or otherwise marking important pages

○ Other: …

- 2.9 If you annotate books, study notes or slides on paper during a lesson, how do you reuse the notes after the lesson?

○ I reread them as I study

○ I scan them to save them in my computer

○ I retype them at the computer

○ I make photocopies for my friends

○ I give the originals to my friends

○ I send them to my friends through the Internet

○ I don't use notes afterwards; I only use them to aid memorization during a lesson

○ Other: …

B.3 Part 3

- 3.0 Would you find interesting being able to reread questions asked by your colleagues during a lesson on a website (for example, because you were absent or to review)?

○ Yes

○ No

- 3.1 If you could ask the professor questions (anonymously) during a lesson using a smartphone, would you do that?

○ Yes, I would

○ No, I would not

- 3.2 If you answered no: why?

○ I want to ask him in person, from my desk or face-to-face

○ I can't come up with questions on the spot

○ It feels like a waste of time

○ Questions only come up during study or review

○ Other: …

- 3.3 Would you find it useful to see questions asked through a smartphone later on a website, with the possibility to discuss them and a clear indication of the context (eg. at what point during a lesson they were posed)?

  ○ Yes

  ○ No

- 3.4 If yes: how would you use this site?

  ○ I would participate in the discussion

  ○ I would just read answers and questions

  ○ Other: …

- 3.5 If you could do any of the following through a cell phone, what of these would you do?

  ○ Finding out course and exam schedules

  ○ Finding classrooms or study rooms

  ○ Looking for information on Politecnico-organized events (e.g. seminars)

  ○ Downloading material, such as study notes or books

  ○ Reading material, such as study notes or books

  ○ Annotating material, such as study notes or books

  ○ Asking for information or questions to the professor

  ○ Organizing my study group

  ○ Planning what to study (e.g. through a calendar)

  ○ Researching the subject

  ○ Asking the teacher about the slides

  ○ Asking the teacher about some other part of the course

  ○ Asking questions to others in my course (eg. through a forum)

  ○ Other: …

B.4 Part 4

Now that you've seen, or tried, our system on iPhone, please complete this part of the survey assuming you had a smartphone with the application you just tested on it. Please, be sincere!

- 4.1 If you could ask the professor questions (anonymously) during a lesson using this application, would you do that?

    ○ Yes, I would

    ○ No, I would not

- 4.2 If you answered no: why?

    ○ I want to ask him in person, from my desk or face-to-face

    ○ I can't come up with questions on the spot

    ○ It feels like a waste of time

    ○ Questions only come up during study or review

    ○ The application doesn't seem like a good way to ask questions

    ○ The application has problems that do not allow to ask questions the way I want

    ○ Other: …

- 4.3 If you answered yes: when would you prefer the professor to respond to your questions? Rank each possibility on a scale of 1 (very low priority) to 5 (very high priority).

    ○ As soon as possible

    ○ As the lesson progresses (even if not immediately)

    ○ At the end of the lesson

    ○ Through the course website

    ○ Other: …

- 4.4 Would you find it useful to see questions asked through this application later on a website, with the possibility to discuss them and a clear indication of the context (eg. at what point during a lesson they were posed)?

    ○ Yes

    ○ No

- 4.5 If yes: how would you use this site?

    ○ I would participate in the discussion

    ○ I would just read answers and questions

    ○ Other: …

- 4.6 If you could do any of the following through a cell phone (outside of the classroom), what of these would you do?

  ○ Finding out course and exam schedules

  ○ Finding classrooms or study rooms

  ○ Looking for information on Politecnico-organized events (e.g. seminars)

  ○ Downloading material, such as study notes or books

  ○ Reading material, such as study notes or books

  ○ Annotating material, such as study notes or books

  ○ Asking for information or questions to the professor

  ○ Organizing my study group

  ○ Planning what to study (e.g. through a calendar)

  ○ Researching the subject

  ○ Asking the teacher about the slides

  ○ Asking the teacher about some other part of the course

  ○ Asking questions to others in my course (eg. through a forum)

  ○ Other: …

- 4.7 Would you say the application was easy to use?

  ○ Yes

  ○ Somewhat

  ○ No

- 4.8 Please list any difficulties you have found while using the application:

  ○ ___________________________

- 4.9 Would you say the application seemed useful?

  ○ Yes

  ○ Somewhat

  ○ No

  Would you comment on the usefulness of the application?

  ○ ___________________________

- 4.10 What of the following features do you find most useful? Rank them as “Very useful”, “Somewhat useful” or “Not useful”.

  ○ Asking questions

  ○ Seeing on what points others asked questions

  ○ Reading the presentation on a nearby screen

  ○ Browsing the presentation

  ○ Expressing my mood

  ○ Other: …

Would you say the application made the lesson…

- 4.11 … more engaging?

  ○ Yes

  ○ Somewhat

  ○ No

- 4.12 … more effective?

  ○ Yes

  ○ Somewhat

  ○ No

- 4.13 … more stimulating?

  ○ Yes

  ○ Somewhat

  ○ No

- 4.14 Do you think the moods list is expressive enough?

  ○ Yes

  ○ No

- 4.15 Would you like to add other moods to the list?

  ○ No

  ○ Yes: ___________________________

- 4.16 Do you think the predefined questions were expressive enough?

○ Yes

○ No

- 4.17 Would you like to add other predefined questions?

  ○ No

  ○ Yes: ____________________________

- 4.18 Would you like further information the application’s slide display, such as question texts, or moods expressed by your colleagues, or would you find it distracting?

  ○ No, it’s fine as it is

  ○ No, it would distract otherwise

  ○ Yes, I’d like the texts of free-text questions.

  ○ Yes, I’d like to know what moods others have expressed.

  ○ Yes, I’d like _________________________

- 4.19 Would you like to comment on anything else?

  ○ ____________________________________

Thank you for your time!

B.5 Survey Results

This survey was proposed three times, during our three in-class evaluation simulations. It was proposed in two different ways: during the first simulation, we proposed the entire survey, whereas in subsequent evaluations we proposed only Part 4 instead. The three instances have been kept separate, and results from each are below. For discussion, see the "Evaluation" chapter.

B.5.1 Evaluation One

What year are you enrolled in?		**How much time do you spend attending lectures?**		**How many courses provide you with online material?**		**How much do you stay connected to the Internet while studying?**	
B — First	0	30-40 h	1	Most (70-100%)	6	Constantly	7
B — Second	0	20-30	3	Some (40-70%)	4	Somewhat	3
B — Third	0	10-20	4	A few (20-40%)	0	Rarely	0
B — Late	0	5-10	2	Less than 20%	0	Never	0
M — First	6	Less than 5					
M — Second	4	None					
M — Late	0						

Table B.1: Evaluation One, Questions 1.1 to 1.4

What do you use the Internet for?		**Do you have a cell phone or a PDA?**		**Do you bring it to lectures?**		**Do you connect to the Internet with it while studying?**	
Find schedules	10	No	0	Always	7	Often	2
Find classrooms	4	Featurephon e	0	Sometimes	0	Never	3
Know about events	5	Smartphone (keyboard)	3	Never	1	Sometimes	5
Download material	10	Smartphone (touchscree n)	7				
Read material	8	iPod touch	0				
Annotate material	3	iPad	0				
Ask the professor	6	Other PDA	0				
Organize study group	4						
Plan what to study	1						
Research the topic	5						
Ask other students	3						

Table B.2: Evaluation One, Questions 1.5 to 1.9

Where do you use it?		What do you do with it?	
At home	2	Find schedules	6
Study room/Library	6	Find classrooms	3
Pub	2	Know about events	2
Park	1	Download material	1
Public transport	4	Read material	4
		Annotate material	0
		Ask the professor	2
		Organize study group	2
		Plan what to study	0
		Research the topic	3
		Ask other students	3

Table B.3: Evaluation One, Questions 1.10 and 1.11

How interesting are videos during lectures?		How interesting are live demos during lectures?		How interesting are polls during lectures?		How interesting are questions to students during lectures	
Very	5	Very	10	Very	2	Very	4
Somewhat	4	Somewhat	0	Somewhat	5	Somewhat	4
Not at all	1	Not at all	0	Not at all	3	Not at all	2

Table B.4: Evaluation One, Question 2.1 (partial)

How interesting are the lecturer's answers to student questions during lecture?		What do you do when you want to ask questions?		When do you prefer to receive an answer?		If applicable, why no questions at all in public?	
Very	2	Raise hand whenever in doubt	6	Immediately	8	Embarassin g	1
Somewhat	7	Raise hand when professor solicits	1	Can wait end of lesson or break	1	Can't come up with q. on the spot	0
Not at all	1	Raise hand at end of lesson or at break	5	Can wait after lesson (eg. mail, forum)	1	Waste of time	1
		Wait for someone else to ask first	1			Q.s come up only during review	3
		Ask privately at the end of the lesson	7				
		Never ask at lessons	0				
		Send a mail	4				
		Ask in course's forum	1				
		Get an appointment	2				
		Never ask, ever	0				

Table B.5: Evaluation One, Questions 2.1 (cont) to 2.4

Do you take notes during lectures?		**How do you take notes?**		**How do you reuse notes?**		**What kind of notes do you take on printouts ?**		**How do you reuse annotations on printouts ?**	
Yes	9	Pen and paper	8	Reread during study	9	Underlining	7	Reread during study	8
No	1	Digital tools	2	Scanned to computer	0	Highlighting	3	Scanned to computer	0
				Retyped	0	Doodles on page	5	Retyped	0
				Photocopied to lend	1	Write on margin	8	Photocopied to lend	0
				Originals lent	2	Doodle on margin	4	Originals lent	1
				Sent via Internet	2	Page corner fold/mark as important	1	Sent via Internet	2
				Not reused	0			Not reused	0

Table B.6: Evaluation One, Questions 2.5 to 2.9

Would you like seeing questions on a website?		**Would you like to ask questions via smartphone?**		**If no: why?**	
Yes	10	Yes	5	Prefer talking	4
No	0	No	5	Can't come up with q. on the spot	0
				Waste of time	1
				Only come up with q.s while reviewing	2

Table B.7: Evaluation One, Questions 3.0 to 3.2.

Would you like discussing questions on a website with lesson context?		If yes, how would you use the website?		What of these would you do on a cell phone?	
Yes	9	I'd participate	7	Find schedules	9
No	1	I'd just read	2	Find classrooms	8
				Know about events	4
				Download material	6
				Read material	5
				Annotate material	0
				Organize study group	4
				Plan what to study	4
				Research the topic	3
				Ask the professor re: slides	5
				Ask the professor re: other things	5
				Ask other students	5

Table B.8: Evaluation One, Questions 3.3 to 3.5.

Would you ask questions using our app?		If no: why?	
Yes	9	Want to ask in person	1
No	1	Can't come up with q.s on the spot	1
		Waste of time	1
		Q.s only come up during review	0
		Doesn't seem right	0
		Doesn't allow me to ask properly	0

Table B.9: Evaluation One, Questions 4.1 and 4.2.

If yes: how would you like the professor responding immediately? [priority]		If yes: how would you like the professor responding as the lesson goes on? [priority]		If yes: how would you like the professor responding at end of lesson? [priority]		If yes: how would you like the professor responding via website? [priority]	
High	5	High	2	High	1	High	1
Mid-high	1	Mid-high	5	Mid-high	3	Mid-high	0
Mid	2	Mid	1	Mid	2	Mid	2
Mid-low	0	Mid-low	0	Mid-low	0	Mid-low	2
Low	1	Low	0	Low	3	Low	4

Table B.10: Evaluation One, Question 4.3

Would you like discussing questions on a website with lesson context?		If yes, how would you use the website?		What of these would you do on a cell phone?	
Yes	9	I’d participate	7	Find schedules	7
No	1	I’d just read	5	Find classrooms	5
				Know about events	2
				Download material	6
				Read material	5
				Annotate material	0
				Organize study group	2
				Plan what to study	3
				Research the topic	2
				Ask the professor re: slides	4
				Ask the professor re: other things	2
				Ask other students	2

Table B.11: Evaluation One, Questions 4.4 to 4.6

Is the app usable?		Was the app useful?	
Yes	9	Yes	5
Somewhat	1	Somewhat	5
No	0	No	0

Table B.12: Evaluation One, Questions 4.7 and 4.9

How useful did you find: asking question s?		How useful did you find: seeing on what other points q.s are made?		How useful did you find: reading the presenta tion on a nearby screen?		How useful did you find: browsin g the presenta tion?		How useful did you find: expressi ng moods?	
Very	7	Very	5	Very	3	Very	7	Very	4
Somewh at	3	Somewh at	5	Somewh at	4	Somewh at	3	Somewh at	5
Not at all	0	Not at all	0	Not at all	3	Not at all	0	Not at all	1

Table B.13: Evaluation One, Question 4.10

Is the lesson more engaging?		Is the lesson more effective?		Is the lesson more stimulating?	
Yes	4	Yes	3	Yes	4
Somewhat	5	Somewhat	7	Somewhat	4
No	1	No	0	No	2

Table B.14: Evaluation One, Questions 4.11 to 4.13

Are moods expressive enough?		Are predefined questions expressive enough?		How about seeing other people's questions and moods?	
Yes	8	Yes	9	Fine like this	1
No	1	No	0	It'd be distracting	3
			0	I'd like to see free-text questions	2
				I'd like to see moods	1

Table B.15: Evaluation One, Questions 4.14 to end.

B.5.2 Evaluation Two

Would you ask questions using our app?		If no: why?	
Yes	8	Want to ask in person	1
No	5	Can't come up with q.s on the spot	1
		Waste of time	0
		Q.s only come up during review	1
		Doesn't seem right	0
		Doesn't allow me to ask properly	0

Table B.16: Evaluation Two, Questions 4.1 and 4.2

If yes: how would you like the professor responding immediately? [priority]		If yes: how would you like the professor responding as the lesson goes on? [priority]		If yes: how would you like the professor responding at end of lesson? [priority]		If yes: how would you like the professor responding via website? [priority]	
High	3	High	4	High	0	High	5
Mid-high	4	Mid-high	4	Mid-high	3	Mid-high	2
Mid	3	Mid	1	Mid	3	Mid	1
Mid-low	0	Mid-low	1	Mid-low	3	Mid-low	1
Low	1	Low	0	Low	1	Low	1

Table B.17: Evaluation Two, Question 4.3

Would you like discussing questions on a website with lesson context?		If yes, how would you use the website?		What of these would you do on a cell phone?	
Yes	11	I'd participate	5	Find schedules	9
No	2	I'd just read	6	Find classrooms	8
				Know about events	2
				Download material	7
				Read material	8
				Annotate material	6
				Organize study group	0
				Plan what to study	1
				Research the topic	4
				Ask the professor re: slides	6
				Ask the professor re: other things	6
				Ask other students	5

Table B.18: Evaluation Two, Questions 4.4 to 4.6

Is the app usable?		Was the app useful?	
Yes	4	Yes	5
Somewhat	5	Somewhat	5
No	0	No	1

Table B.19: Evaluation Two, Questions 4.7 and 4.9

How useful did you find: asking questions?		How useful did you find: seeing on what other points q.s are made?		How useful did you find: reading the presentation on a nearby screen?		How useful did you find: browsing the presentation?		How useful did you find: expressing moods?	
Very	8	Very	4	Very	1	Very	9	Very	2
Somewhat	3	Somewhat	9	Somewhat	9	Somewhat	3	Somewhat	8
Not at all	2	Not at all	0	Not at all	3	Not at all	1	Not at all	3

Table B.20: Evaluation Two, Question 4.10

Is the lesson more engaging?		Is the lesson more effective?		Is the lesson more stimulating?	
Yes	4	Yes	3	Yes	4
Somewhat	6	Somewhat	9	Somewhat	4
No	3	No	1	No	5

Table B.21: Evaluation Two, Questions 4.11 to 4.13

Are moods expressive enough?		Are predefined questions expressive enough?		How about seeing other people’s questions and moods?	
Yes	10	Yes	11	Fine like this	2
No	2	No	1	It’d be distracting	1
				I’d like to see free-text questions	2
				I’d like to see moods	2

Table B.22: Evaluation Two, Questions 4.14 to end

B.5.3 Evaluation Three

Would you ask questions using our app?		If no: why?	
Yes	14	Want to ask in person	1
No	2	Can’t come up with q.s on the spot	0
		Waste of time	0
		Q.s only come up during review	1
		Doesn’t seem right	0
		Doesn’t allow me to ask properly	0

Table B.24: Evaluation Three, Questions 4.1 and 4.2

If yes: how would you like the professor responding immediatel y? [priority]		If yes: how would you like the professor responding as the lesson goes on? [priority]		If yes: how would you like the professor responding at end of lesson? [priority]		If yes: how would you like the professor responding via website? [priority]	
High	6	High	5	High	5	High	3
Mid-high	2	Mid-high	7	Mid-high	1	Mid-high	1
Mid	3	Mid	1	Mid	4	Mid	4
Mid-low	2	Mid-low	1	Mid-low	4	Mid-low	3
Low	0	Low	0	Low	0	Low	3

Table B.25: Evaluation Three, Question 4.3

Would you like discussing questions on a website with lesson context?		If yes, how would you use the website?		What of these would you do on a cell phone?	
Yes	12	I’d participate	8	Find schedules	14
No	4	I’d just read	4	Find classrooms	10
				Know about events	3
				Download material	10
				Read material	7
				Annotate material	4
				Organize study group	5
				Plan what to study	2
				Research the topic	4
				Ask the professor re: slides	10
				Ask the professor re: other things	10
				Ask other students	6

Table B.26: Evaluation Three, Questions 4.4 to 4.6

Is the app usable?		**Was the app useful?**	
Yes	7	Yes	12
Somewhat	9	Somewhat	4
No	0	No	0

Table B.27: Evaluation Three, Questions 4.7 and 4.9

How useful did you find: asking questions?		**How useful did you find: seeing on what other points q.s are made?**		**How useful did you find: reading the presentation on a nearby screen?**		**How useful did you find: browsing the presentation?**		**How useful did you find: expressing moods?**	
Very	13	Very	4	Very	4	Very	13	Very	4
Somewhat	3	Somewhat	11	Somewhat	10	Somewhat	2	Somewhat	7
Not at all	0	Not at all	0	Not at all	2	Not at all	1	Not at all	5

Table B.28: Evaluation Three, Question 4.10

Is the lesson more engaging?		**Is the lesson more effective?**		**Is the lesson more stimulating?**	
Yes	9	Yes	7	Yes	12
Somewhat	7	Somewhat	9	Somewhat	4
No	0	No	0	No	0

Table B.29: Evaluation Three, Questions 4.11 to 4.13

Are moods expressive enough?		Are predefined questions expressive enough?		How about seeing other people's questions and moods?	
Yes	12	Yes	12	Fine like this	2
No	3	No	2	It'd be distracting	2
				I'd like to see free-text questions	5
				I'd like to see moods	3

Table B.30: Evaluation Three, Questions 4.14 to end

Bibliography

[0]: “The iPhone share: 17.25% of smartphones, 4.2% all phones”, H. Dediu (http://www.asymco.com/2011/02/01/the-iphone-share-17-25-of-smartphones-4-2-all-phones/), February 2010

[1]: “Addressing computer-supported collaborative learning in the classroom: Experiences in engineering education”, I. Casas, C. Collazos *et al.*, from Procedia – Social and Behavioral Sciences, 2010.

[2]: “Snapshot: Global mobile connections surpass 5 billion milestone”, Report by Wireless Intelligence (http://www.wirelessintelligence.com/), July 8th, 2010.

[3]: “Mobile Learning projects – a critical analysis of the state of the art”, D. Frohberg, C. Göth, G. Schwabe, from Journal of Computer Assisted Learning, 25: 307–331, August 2009.

[4]: “Increasing interactivity in blended classrooms through a cutting-edge mobile learning system”, R. Shen, M. Wang, X. Pan, from British Journal of Educational Technology 39 N. 6, 2008.

[5]: “The impact of mobile learning on students’ learning behaviours and performance: Report from a large blended classroom”, M. Wang, R. Shen, D. Novak, X. Pen, from British Journal of Educational Technology 40 N. 4, 2009.

[6] “MIT.EDU: M-learning Applications for Classroom Settings”, M. Sung, J. Gips, N. Eagle *et al.*, from Journal of Computer Assisted Learning, 21: 229–237, June 2005.

[7]: “Classtalk: A Classroom Communication System for Active Learning”, R. J. Dufresne, W. J. Gerace, *et al.*, from Journal of Computing in Higher Education, 7, 3-47, 1996.

[8]: “Tate Modern Multimedia Tour pilots 2002–2003”, N. Proctor, J. Burton, from “Learning with Mobile Devices: Research and Development”, Learning and Skills Development Agency, 2004

[9]: “LOCH: supporting mobile language learning outside classrooms”, H. Ogata *et al.*, from International Journal of Mobile Learning and Organization Vol. 2, N. 3, 2008

[10]: “xTask – an adaptable learning environment”, H. Ketamo, from Journal of Computer Assisted Learning N. 19, 2003.

[11]: “Agile Software Development: Principles, Patterns, and Practices”, R. C. Martin, Prentice Hall, 2003.

[12] “Bigtable: A Distributed Storage System for Structured Data”, F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh *et al.*, from ACM Trans. Comput. Syst. 26, 2, Article 4, June 2008.

[13] “RESTful Web Services”, L. Richardson, S. Ruby, edited by O’Reilly, 2007.

[14] “Mobile Learning — Small devices, big issues”, M. Sharples *et al.*, from Technology-enhanced learning: Principles and products, edited by Springer, 2007.

[15] “What Students Really Need beyond Learning Content: Ubiquitous Shared-connectivity Services to Foster Learning Communities on the Campus”, V. M. Garcia-Barros *et al.*, from the proceeds of the MindTrek conference, 2009.

[16] “Mobile Learning with Bluetooth-based E-learning System”, Y. Zhang *et al.*, from the proceeds of International Wireless Communications and Mobile Computing Conference, 2006.

[17] “Podcasting Enhances Learning”, P. R. Ormond, from Journal of Computing Sciences in Colleges, 2008.

[18] “Mobile learning: A framework and evaluation”, L. F. Motiwalla, from Computers & Education edited by Elsevier, 2007.