POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione

POLO REGIONALE DI COMO

Master of Science in
Computer Engineering

SOCIAL NETWORKS:
analysis and statistics for integrated social profiles

Supervisor: Prof. Marco Brambilla

Master Graduation Thesis by: Denis Musall
Student Id. number: 735338

Academic Year 2009/2010

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione

POLO REGIONALE DI COMO

Corso di Laurea Specialistica in
Ingegneria Informatica

RETI SOCIALI:
analisi e statistiche per profili sociali integrati

Relatore: Prof. Marco Brambilla

Tesi di Laurea di: Denis Musall
Matricola: 735338

Anno Accademico 2009/2010

INDEX

SOMMARIO

Negli ultimi anni, la diffusione delle reti sociali è stata favorita dall'introduzione del Web 2.0, che non solo ha permesso agli utenti di servirsi dei contenuti presenti su Internet, ma anche di concorrere nel crearne nuovi.
L'enorme successo ottenuto da alcune reti sociali, ha contribuito alla diffusione di molti *"aggregatori di contenuto"*, applicazioni che permettono di osservare gli aggiornamenti provenienti da più reti a cui l'utente è iscritto, nonché di interagire con altri membri della stessa rete.
Nonostante le applicazioni di questo tipo siano molteplici, nessuna è in grado di visualizzare statistiche riguardanti un insieme di profili sociali forniti dall'utente.

La tesi nasce con l'intento di studiare il problema legato alla reputazione di un utente all'interno delle reti sociali, di definire metriche e modelli d'analisi, nonché di creare un'applicazione web, attraverso la quale gli utenti registrati hanno la possibilità di visionare, non solo gli aggiornamenti relativi a più reti, ma anche statistiche riguardanti i dati raccolti quotidianamente, relative a più profili sociali.
A tal proposito, è stata creata una pagina in grado di dare una visione generale in merito all'attività sociale dell'utente, riassumendo le informazioni quantitative raccolte, visualizzando indicatori di *"visibilità"* e *"produttività"* e dando la possibilità di visionare diagrammi temporali relativi all'attività sociale dell'utente.

Partendo da una descrizione sulle reti sociali, sono state fornite le loro principali caratteristiche, introducendo esempi concreti, utilizzati in seguito per la realizzazione del progetto di tesi.

Successivamente sono state presentate le scelte operate durante la fase implementativa, descrivendo come i dati forniti dalle API delle diverse reti sociali sono stati analizzati e hanno influenzato la struttura dati creata. Sono inoltre state presentate le principali caratteristiche dell'applicazione web realizzata.

Infine, sono stati analizzati i risultati ottenuti attraverso la realizzazione del progetto e suggeriti alcuni interessanti sviluppi da attuare in futuro.

ABSTRACT

The online social networks diffusion, in the last years, has been facilitated by the introduction of the Web 2.0 that, not only enables users to retrieve information on the web, but also to contribute creating new ones.
The enormous success obtained by some of them, has contributed to the diffusion of many *"content aggregators"*, applications that enable users to see in real-time social updates coming from different social networks to which they are subscribed, providing them, the possibility to interact with others as if they are really connected to the social website.
Despite there are many applications of this type, no ones allows to visualize integrated statistics related to a set of social profiles provided by the user.

The main objectives of this thesis, are the study of the reputation problem within social networks, the definition of analysis metrics and models, and the creation of a web application through which registered users may view, not only social updates in real-time, but also statistical data daily collected, related to all their social profiles. In particular, it has been created, a page containing a general overview related to the user selected, resuming all his quantitative information collected, as well as, providing indicators of *"visibility"* and of *"productivity"*, and the possibility to visualize temporal diagrams about his social activity carried out over different social networks.

This paper begins providing a background over online social networks, describing which are their main characteristics and introducing some example of them taking

into considerations those used during the implementation phase. Subsequently it draws, the main contribution provided through the realization of the thesis project, describing how the analysis over the data provided by the different social network APIs has been conduced, presenting also the main important features of the web application realized and a detailed description about the implementation phase.
Finally, it provides some conclusions about the project realized, evaluating the results obtained and suggesting some interesting future developments that could be done.

1

INTRODUCTION

In the last five years, thanks to the diffusion of Web 2.0 [1] that enable users not only to retrieve information on the web but also to contribute creating new ones, online social networks have become one of the major phenomena over Internet.

People, through the development of new social media technologies, have begun to connect with each others via Internet, sharing information, idea, common interest, membership in particular groups (i.e. friends, professional colleagues) [2], enabling them to meet strangers but in many cases also to make visible all *"hidden tie"* shared into the real world with known people.

Despite there are many different types of online social network, focusing on individuals (like Badoo, Facebook) or on contents (like SlideShare, YouTube), commonly they share a number of the following concepts [3]:

- *Profiles*: allow each user to be identified over a social network.
- *Relationships*: are connections between a user and others subscribed to the same social network.
- *Contents*: depending on the type of social network, they may be information exchanged between users through simples messages or

others types of them that the social platform enable to share, like videos, images, links, etc.

It's on the basis of these main social concepts that the SNA [4] (Social Network Analysis), applying the network theory, tries to understand how relations are created and how information and contents flows on networks, this, in order to improve existing systems or to create new ones and to make considerations useful for the users themselves (suggesting future interactions or new possible relationships) or for the marketing needs.

Recently, always thanks to the social networks success, many content aggregators like TwitterDeck[1], has been created. These applications enable users to see in real-time social updates coming from different social networks to which they are subscribed, providing them, through previous authorization, the possibility to interact with others as if they are really connected to the social website.
Despite there are many applications of this type, no ones allows to visualize integrated statistics related to a set of social profiles provided by the user.

The main objectives of this thesis project are the study of the reputation problem within social networks, the definition of analysis metrics and models, and the creation of a web application through which registered users may view, not only social updates in real-time, but also statistical data daily collected, related to all the social profiles provided.
In particular, during the project implementation, different types between the most common social networks, have been chosen on the basis of the information made available by their API and, through the definition of a specific data structure, all the daily variations of the social values observed (like content items, posts, views, etc.), have been memorized. Moreover, a page containing a general overview

[1] *TwitterDeck*: is a client application implemented through Adobe Air. It allows users to manage their accounts over different social networks like for example Twitter, Facebook, FourSquare, etc.

related to the user selected, resuming all the quantitative information collected, and providing indicators of *"visibility"* and of *"productivity"*, has been created. Always through this page, users have the possibility to visualize temporal diagrams about their social activity carried out over the different social networks.

The second chapter describes in detail what online social networks, focusing the attention on the main characteristics and introducing those used for the implementation phase of the thesis project. It provides also some examples on how enterprises can benefit adopting them. Finally it draws some real studies conduced nowadays over social networks.

The third chapter describes the main contributions to which a user, subscribed to at least one social network, may benefit thanks to the implementation of the thesis project. In particular, it defines and describes the *"unified visibility panel", a page* containing an overview related to the user's profiles that resumes all the quantitative information collected, as well as provides indicators of *"visibility"* and *"productivity"*. It presents also, through some example, how influencing variables and the related weights, for both types of indicators, have been chosen.

The fourth chapter describes the main requirements on which the implementation of the application is based, showing how the database has been structured in order to contain all the quantitative information daily collected about the different social networks. Since not all the value collected have temporal references, it also explains how the variation related to this types of values over time, have been memorized into the database.

Sequence diagrams are also presented with the aim to show how the main features of the application, like the visualization of the temporal charts, are managed.

The fifth chapther draws how the application has been tested using different categories of social profiles. Moreover, it presents some example of temporal graphs that show the main aspects highlighted during the period observed, with some considerations made on them.

Finally, the last chapter provides a recap over the thesis work, describing limits highlighted during the project implementation and the main final objectives achieved.

The conclusions also outline some features that, through future developments, will improve the application realized, like for example the introduction of graphical indicators of *"productivity"* and *"visibility"* or new types of graphs to visualize data collected.

2

BACKGROUND

2.1 Online social network

An online social network is a web service that allows individuals to construct a public or semi-public profile within a bounded system, to build up a list of other users with whom they share a connection, and to view and to traverse their list of connections and those made by others within the system [5].

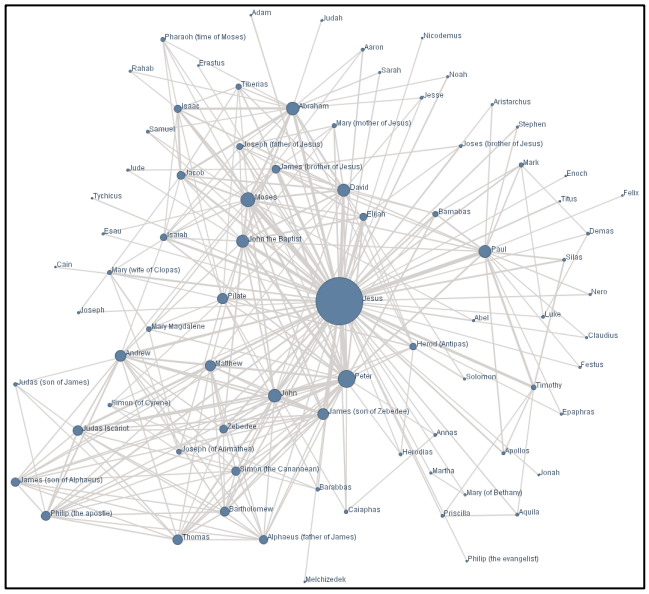

Figure 1 - Example of Social Network Graph showing the friendship connections of a user

Motivations that thrust users to create these connections may vary from site to site, as well as names given to them.

They can be considered unique sites that allow subscribed users to meet strangers, but also to make visible their existing social networks. This can result in connections between individuals that would not otherwise be made, but that is often not the goal, and these meetings are frequently between *"latent ties"* who share some offline connection [5].

It's infact common, nowadays, that people subscribed to online social networks, are not necessarily *"networking"* or looking to meet new ones, but instead that they are primarily communicating with others already known in their extended social network, the real world.

Main characteristics

Despite online socials networks may focus on individuals or on contents and have also a wide variety of technical features, their backbone consists of visible profiles that display an articulated list of Friends[2] who are also users of the system themselves [5].

In order to join an online social network, a user must fill out forms containing a series of questions about, for example, his name, surname, age, email, location, interests, etc., encouraging also in some cases to upload a profile photo. It's according to the given answers that the user's social profile is generated.

Thanks to the possibility to define privacy's constraints, its visibility may vary from site to site and according to the discretion of the user.

Structural variations around visibility and access are one of the primary ways that online social network differentiate themselves from each other.

[2] *Friends*: the term is used in order to generalize the different labels that can be associated to the social relationships.

After the subscription to an online social network, a user is prompted to identify others subscribed to the same system with whom he may start to have an online social relationship. These ties may be called with different names, depending on the type of social network (popular terms include *friends*, *fans*, *followers*, *contacts*, etc.).
Most social network requires bi-directional friendship confirmation, but some do not. One-directional ties are sometimes labelled as "*fans"* or "*followers"* but many sites call these "*friends"* as well, using a misleading term.

Display in public connections, is a crucial component of an online social network. Friends list contains links to Friend's profile, enabling viewers to traverse the network graph by clicking through different Friends lists [5].
On most sites, these lists are visible to anyone who is permitted to view the profile, although there are exceptions like MySpace, YouTube and Facebook, where a user can decide to set his profile in order to hide this information.

Most online socials network provides mechanism for users in order to leave public messages on their Friends' profiles as *"comments"*. In addition, some of them provide the possibility to send to others profiles, private messages, providing a feature similar to a webmail.

Beyond profiles, friends, comments and private messaging, online social network vary greatly in terms of features and uses. For example, some of it are focused on photo-sharing or video-sharing capabilities, others built-in blogging and instant messaging technology.

In general, since online social network are built to encourage the communication between people [6], they typically emphasize some combination of the following social attributes:

1. ***Identity****: Who are you?*
2. ***Reputation****: What do people think you stand for?*
3. ***Presence****: Where are you?*
4. ***Relationships****: Who are you connected with? Who do you trust?*
5. ***Groups****: How do you organize your connections?*
6. ***Conversations****: What do you discuss with others?*
7. ***Sharing****: What content do you make available for others to interact with?*

They don't need to exhibit all of these features, but the more attribute areas they cover, the more engaging they are likely to be.

2.2 Socials networks: some examples

In order to provide concretes examples on how socials networks may vary greatly in terms of features and uses, an overview over different types of them, used also during the realization of this thesis project, is presented.

2.2.1 Facebook

Facebook is a social network website intended to connect friends, family, and business associates [7].

It's the largest one of the networking sites with more than 500 millions[3] of active users, which is about one person every fourteen in the world.

It has been created as a college networking website in 2004 and two years later has been expanded in order to offer the possibility to everyone to join it [8].

One of the main important characteristics, that also has contributed to spread so much the social network, is certainly given by the possibility to search users profiles specifying their real name/surname.

Between the main features provided by Facebook are:

- ✓ *Post status messages on the wall*
- ✓ *Comment posts of others users*
- ✓ *Share user's photos, videos and links*
- ✓ *Friends list*
- ✓ *Tag comments, videos, photos and links*
- ✓ *Possibility to express likely on post of others users*

[3] *Information source*: http://www.facebakers.com (10/2010)

- ✓ *Instant messaging*
- ✓ *Create, share and join groups*
- ✓ *Send private messages to others profiles*
- ✓ *Create, invite and join events*
- ✓ *Set different privacy levels*
- ✓ *Search user by name or email address*
- ✓ *Connection through a mobile device*

2.2.2 Twitter

Twitter is an online social network based on micro blogging. Since its creation in 2006, it has gained popularity worldwide and currently has more than 160 millions of users[4], mainly distributed in USA and Australia, less in Europe [9].

It enables its users to send and read other users' messages called *"tweets"*, text-based posts of up to 140 characters, displayed on the user's profile page.
Posted messages are publically visible by default, however senders can decide to restrict the delivery to their friends list.
Users may also subscribe to other tweets author putting them into their *"following"* list; subscribers are known as *"followers"*.

An important feature provided by Twitter, in order to establish the authenticity of well known account, is the *"verified badge"*, an icon that is associated to all the verifies accounts, so users can trust that a legitimate source is authoring their tweets.

[4] *Information source*: http://sites.google.com/a/twitter.com/twitter-press-site/metrics (10/2010)

Other important features provided by the social network are:

- ✓ *Send public tweets mentioning a specific user (syntax to be used @username)*
- ✓ *Create lists specifying following users*
- ✓ *Send private tweets to followers*
- ✓ *Insert shorts links into tweets*
- ✓ *Connection through mobile device*
- ✓ *Search user by name or email address*

2.2.3 YouTube

YouTube is a video-sharing website on which users can upload, share, and view videos. It has been created in 2005 and today counts more than 152 millions[5] of users.

It uses Adobe Flash Video technology to display a wide variety of user-generated video content, including movie clips, TV clips, and music videos, as well as amateur content such as video blogging and short original videos [10].

Most of the contents on YouTube have been uploaded by individuals, although some media corporations like CBS, BBC and other organizations offer some of their material through the site.

Registered users can upload an unlimited number of videos, while unregistered can only watch them.

[5] *Information source*: Google's search engine, typing "site:www.youtube.com/user" (11/2010)

The main features provided by the social network are:

- ✓ *Tag videos uploaded*
- ✓ *Leave comments on a video*
- ✓ *Vote a video*
- ✓ *Answer to a published video with another one*
- ✓ *Friends list*
- ✓ *Subscribe video channels*
- ✓ *Add to favourites video*
- ✓ *Set different levels of privacy for each video*
- ✓ *Search users by username*

2.2.4 SlideShare

SlideShare is the largest online community for sharing business presentations and documents [11]. Introduced in 2006, the website counts about 23 millions of unique visitors a month[6].

Registered users can upload PowerPoint presentations, Word documents and share them into a blog or send an email containing the URL to friends.
Without registration they can find presentations and documents on almost any topic, searching them on the site, finding them through tags, or navigating through documents proposed of the same topic requested by the user.

All transcripts of the files uploaded are indexed by search engines, enhancing the ranking of the user's presentations and of its documents.

[6] *Information source*: http://www.quantcast.com/ (11/2010)

Others mains features provided by the social network are:

- ✓ *Tag presentations*
- ✓ *Download presentations*
- ✓ *Embed presentations into users' website or blogs*
- ✓ *Share presentations over others socials networks*
- ✓ *Followers list*
- ✓ *Share documents privately (ex. over an Intranet)*
- ✓ *Join groups*
- ✓ *Upload content from others socials networks/devices*
- ✓ *Add to favourites presentations*
- ✓ *Upload and share videos*
- ✓ *Browse presentation by category*

2.3 Socials networks utility in industries

2.3.1 Social BPM

A Social BPM is a Business Process Management to which new collaborative possibilities provided by the Web 2.0, are made available in order to improve processes, taking also into consideration knowledge and experience of a community [12].

Recently, BPM has started to move from a technical way of modelling organizations and automatized processes called science-based, to a human-centric discipline, this due to the fact that, despite the combination of implementing BPM technology and best practices disciplines leads to the creation of highly successful industry, analysts start to recognize a critical missing component provided by the human collaboration [13].

Even if BPM is designed to provide the most optimal solutions, it has the tendency to ignore variance of opinions, ideas and work styles within the organization. The adoption of social networking tools into BPM, despite some criticisms, enables to access a large amount of new resources that may contribute to meet the ever-changing needs of companies.

Due to the fact that each person has unique characteristic, knowledge and skill that enable them to perform tasks, assigning to every one identical series of process, hinders the ability of the organizations to optimize their workflow capacity. This problem may be solved with the introduction of social tools into BPM, enabling workers to express their unique abilities in real time and in an interactive way, promoting collaboration and coordination into the workplace and giving them the opportunity to continuously improve and re-designs their specific business models.

Introduce social BPM into a company, changes the way to identify, manage and solve processes problems [13]. Through the utilization of social tools, an unlimited number of users are able to provide their suggestion giving contribution during the process modelling, creating the most successful and efficient one, interacting each other during the process execution to leverage specific skill sets as needed in order to take the best decisions.

Nowadays, between the main emerging social BPM patterns [14], there are:

- ✓ *Process Reflection*: during the execution of the process, users provide direct feedback on possible improvements to be done.
- ✓ *Extended Discovery*: extends the process discovery taking into consideration bi-directional communication between the company and the customers.
- ✓ *Runtime Process Guidance*: modifies in real-time business processes through the observation of internal and external socials networks.

2.3.2 Enterprise 2.0

The term Enterprise 2.0 indicates all technologies and business practices that make free the workforce from the constraints of legacy communication and productivity tools, like emails [15].

The adoption of new Web 2.0 collaborative tools integrated into the office environments makes accessible the collective intelligence of workers, enabling the information to flow "everywhere", without the necessity to follow hierarchical constraints, translating this into a huge competitive advantage in terms of increased innovation, productivity and capability to adapt the company to the frequently change of the market needs.

Enterprise 1.0	Enterprise 2.0
Hierarchy	Flat Organization
Friction	Ease of Organization Flow
Bureaucracy	Agility
Inflexibility	Flexibility
IT-driven technology / Lack of user control	User-driven technology
Top down	Bottom up
Centralized	Distributed
Teams are in one building / one time zone	Teams are global
Silos and boundaries	Fuzzy boundaries, open borders
Need to know	Transparency
Information systems are structured and dictated	Information systems are emergent
Taxonomies	Folksonomies
Overly complex	Simple
Closed/ proprietary standards	Open
Scheduled	On Demand
Long time-to-market cycles	Short time-to-market cycles

Figure 2 - Main differences between Enterprise 1.0 and the new ones 2.0

In order to clarify which tools are take into consideration with Enterprise 2.0, some examples are provided [16]:

Wiki

Business wiki is one of the most popular forms of Enterprise 2.0. As a collaborative system it is useful for both small tasks, like creating a dictionary about the industry jargon (wiki documentation), and large tasks, like holding

online meetings (wiki meeting).
Due to its easy creation, it is one of the easiest ways to begin implementing Enterprise 2.0 into a workplace.

Blog

A blog can be used in order to keep employee informed about the main events concerning the company or on what happen inside a department. It has a great role in an organization realizing the so call *"top-bottom"* communication between company management and employees that, thanks to the possibility to leave comments on each post, can easily ask for clarification or give suggestions.

Micro-blogging

Micro-blogging, like Twitter, can be used as collaborative tool in order to maintain update team members on what they are working on and to quickly communicate within a group of employee.
A blog network, for example, could use micro-blogging to let writers notify others on what they are working on, this in order to avoid the possibility to public articles providing the same information.

Social Networking

Social networking is another important form of Enterprise 2.0.
Despite the efforts to implement Enterprise 2.0 into the company intranet grow, traditional interfaces based on it can became insufficient.
Social networking provides the possibility to be used as a traditional intranet interface, but also to add some utilities that can help the communication flow through different networks.
Into larger companies, for example, it can provide a great way to find people with

specialized skills and knowledges looking through the employees' profiles, enabling managers to assign the best person for a specific task.

2.4 Analysis on social network

The Social Network Analysis [4] is an approach based on a set of mathematical techniques (in particular on graph theory) used in social psychology, sociology, ethology, and anthropology, that permit to analyse social network focusing on relationships, both casual acquaintance and close bonds, between nodes of people and/or groups.

Figure 3 - Between people and content there is a reciprocal relationship

Among its main objectives there are:

1. Visualization through diagrams of communications and other relationships between nodes.
2. Study factors that influence relationships and their correlations.
3. Draw out implications of the relational data, including bottlenecks where information flows through one person or section and situations where information flows does not match formal group structure.
4. Improve communication providing recommendations based on the studies conduced.

Follow descriptions about some examples of social networks analysis conduced in order to clarify how they can be performed.

2.4.1 Measurements and analysis of online Social Networks

Thanks to the popularity of some online social network that provides a powerful means of sharing, organizing, and finding content and contacts, it was possible to study the characteristics of the social network graphs at large scale.

The analysis has been done taking into considerations publics data obtained from different social networks (YouTube, LiveJournal, Flickr and Orkut), this due to the fact that data gathered from multiple sites enables to identify common structural properties [17].

One of the objectives was that to try to understand if the study of the social network structure, might lead to algorithms that could detect trusted or influential users, much like the study of the Web graph led to the discovery of algorithms for finding authoritative sources in the Web.

An important aspect observed, was the high degree of reciprocity in directed user links, leading to a strong correlation between user indegree and outdegree, providing some difficulties in identifying valid information through the analysis of the only structure of the network (this differs from content graphs like the graph formed by web hyperlinks, where the popular pages, *"authorities"*, and the pages with many references, *"hubs"*, are distinct).

The analysis has highlighted that online social networks contains a large, strongly connected core of high-degree nodes, surrounded by many small clusters of low-degree nodes, suggesting that high-degree nodes in the core are critical for the connectivity and the flow of information.

The high coefficient of local user clustering has confirmed that in social network, normally people tend to be introduced to other people via mutual friends, increasing the probability that two friends of a single user are also friends.

Finally, on groups it has highlighted that members of smaller user groups tend to be more clustered than those of larger groups.

Studies conduced can be view as a stating point in order to improve current system or to design new application, to test current theories on offline social network and to improve the marketing campaigns.

2.4.2 Predicting positive and negative links in Social Networks

Normally social network analysis is done taking into consideration positive relations between users, without considering the negative ones.

This analysis has been conduced on three different social networks: Wikipedia (encyclopaedia updated collectively by users), Epinions (web site through which users can leave reviews on products purchasable online) and Slashdot (news website on new technologies).

Relations among subscribed users, have been considered doing a clear distinction between those positives (like for example friendships between users or appreciation of a link published by another user) and those negatives (like considering users put into a blacklist by others or disapproval on a link published by others) [18].

Studies performed have confirmed that, using both relations' types enables to make more accurate predictions.
Another important aspect that has been highlighted is that, employing information about negative relationships, can be useful even for tasks that involve only the positive ones, such as, the problem of links prediction for positive edges.

Thanks to the methodologies adopted during the analysis and to the different relations' meaning of the social networks considered, results obtained have identified principles that can be generalized across multiple domains.

2.4.3 Modelling relationship strength in online Social Network

Nowadays, analysis performed on socials networks, tend to consider binaries relations between friends, and to make on them appropriate observations.
Often, these relations, include close friendships and casual acquaintances that permit the realization of models affected by noise, making necessary the adoption of new methodologies able to narrow it, using techniques to identify similarities between users [19].

The analysis has been performed taking into consideration not only information concerning the friendship's relations, but also the interaction's activities between users (e.g. through the exchange of information using a messages' board or even observing the tagging activities), identifying in this way similarities between them and trying to improve the accuracy of the prediction model.

Methods used, based on the sociological theory of the *"homophily"* (people tend to stay connected with other having the same characteristics), in contrast with the previous, has not created a clear distinction *"strong/weak"* between friendships but, it has provided a classification scale for them, ranging results from *"weak"* to *"strong"*.

In order to try to make prediction on the future users' interactions, a statistical analysis was carried out, considering two main type of information: the users' profiles (static information) and the interactive activities related to them (dynamic, time distributed).

Methodologies adopted during the analysis of the friendship's relations, can be used also to make consideration on other aspects, like:

- ✓ *Links*
- ✓ *NewsFeeds* (e.g. showing as first that related to strong friendships)
- ✓ *People searching* (e.g. making a ranking on the results)

3

MAIN CONTRIBUTIONS

Nowadays, the successes of online social networks have fostered the creation of many contents aggregators like TwitterDeck[7]. These applications enable users to see in real-time social updates coming from different social networks to which they are subscribed, providing them, through previous authorization, the possibility to interact with others as if they are really connected to the social websites.
Despite there are many applications of this type, no ones allows to visualize statistics related to the social profiles provided by the user.

This thesis project has as a main objectives the creation of a web application, through which users, previous registration, can visualize all real-time contents updates related to the different social profiles provided, as well as to render temporal charts showing quantitative data (like video views, video comments, etc.) daily collected.

Between the main project's contributions there are:

- ✓ A section providing real-time updates on the social activities performed by the user with his different social profiles.

[7] *TwitterDeck*: is a client application implemented through Adobe Air. It allows users to manage their accounts over different social networks like for example Twitter, Facebook, FourSquare, etc.

- ✓ A *"unified visibility panel"*, enabling to resume through quantitative data daily collected the social activity of the user registered. In particular indicators of social *"productivity"* and *"visibility"*, for each social profile, are provided.

- ✓ The possibility to render temporal charts showing quantitative information collected related to each social profile provided by the users.

3.1 User's real-time updates

Through the main page of the web application developed, that provides the list of the person registered to the application, a user can decide to see his social updates by clicking the *"updates"* button associated to his profiles (see fig. 4).

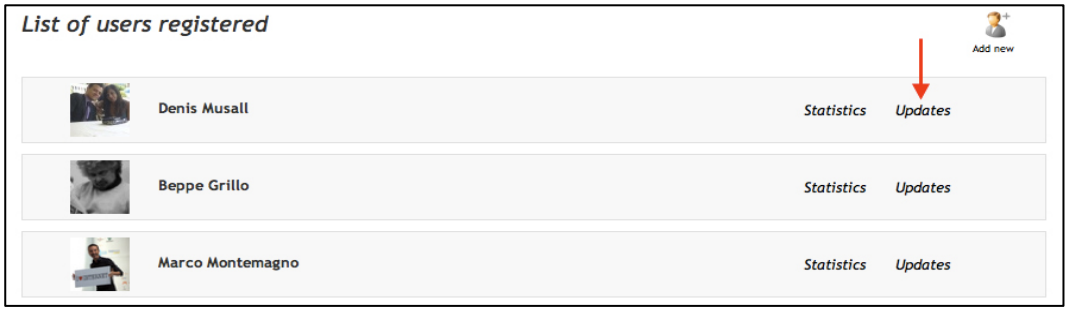

Figure 4 - Example of subscribed users' list

In general, if the user is subscribed to all the social networks considered by the web application and he has provided all his social identifiers during the registration phase, the update's page visualized is composed by four sections (see fig. 5), one for each social network.

Each section contains the recent social activities carried out by the user over the related social network, in particular, for that focusing on contents, like SlideShare and YouTube, are showed the thumbnails related to last four presentations/videos and others detail's information like the upload date and the total views, while for that focusing on individuals, like Facebook and Twitter, are visualized the last four status updates.

Clicking on a thumbnail related to a social item, the user can leave, previous confirmation, the web application, visualizing the chosen content details on the related social website.

If the updates to be visualized are more than four, a button *"see more"* (1) is added to the bottom of the related section, providing the possibility, depending on the cases, to leave the application towards the user profile page of the social network website, in case of Twitter and Facebook or, to visualize another page

and navigate between all the historical uploads, in order to choose a content, in case of SlideShare and YouTube.

It also may happen that a user is subscribed to a social network but that he has no social activity over that. In this case, into the related section, a warning message (2) is visualized informing that no social user updates has been founded.

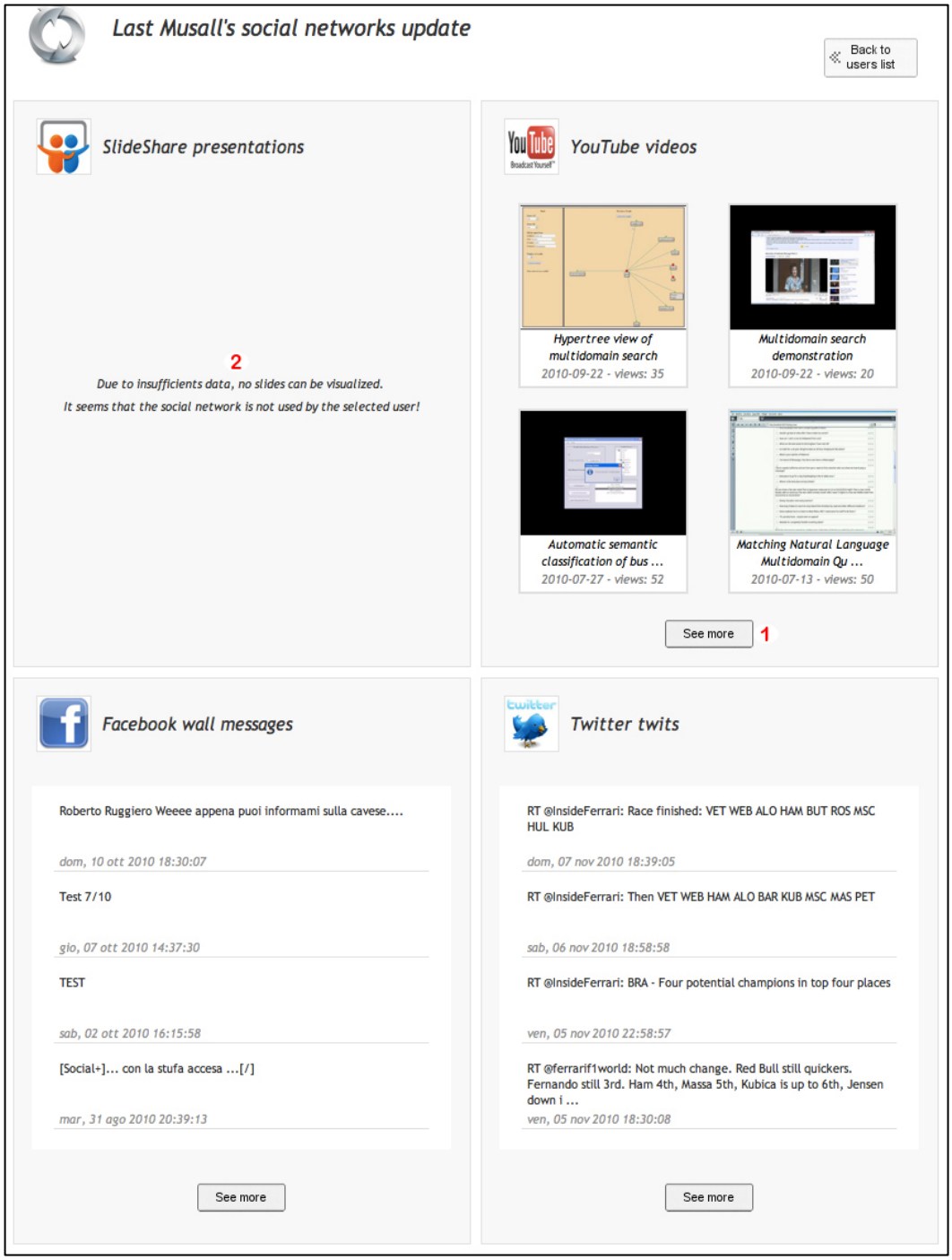

Figure 5 - Example of social updates' page related to a User subscribed to all the social networks considered by the application

3.2 The *"unified visibility panel"*

Navigating from the main page of the web application a user, clicking on the *"Statistics"* button associated to his profiles, can visualize a page containing the *"unified visibility panel"*.

This panel (see fig. 6) shows, for each social profile identifier, provided by the user during the registration phase, an historical general prospectus of his social activities over the different social networks.

Quantitative data visualized on it, may vary, depending on the type of social network and on the information made available by its API (in some cases, like for Twitter, it may depend also on the queries limit per hour).

In general, for that focusing on contents, like SlideShare and YouTube, information such as the number of total items uploaded, views or comments, as well as a thumbnail with the title of the top item (enabling users to click on it in order to see its historical charts), are provided. For that focus on individuals, such as Facebook and Twitter, information like the total number of status updates, friends, or groups joined by the user, are visualized.

Always through the *"unified visibility panel"*, the user has the possibility to visualize his main social history over a specific social network joined by him, through the rendering of temporal graphs, clicking on the related *"Main History"* button (see 2.3 "Rendering temporal charts").

If the user has provided the social identifier of SlideShare or YouTube, he has also the possibility to visualize historical charts related to a single item uploaded.

Clicking on the button *"Select video"*, in case of YouTube, or *"Select presentation",* in case of SlideShare, and navigating through the different items crawled by the application developed, the user can choose the ones of which he wants to visualize the temporal graphs of the quantitative data daily collected.

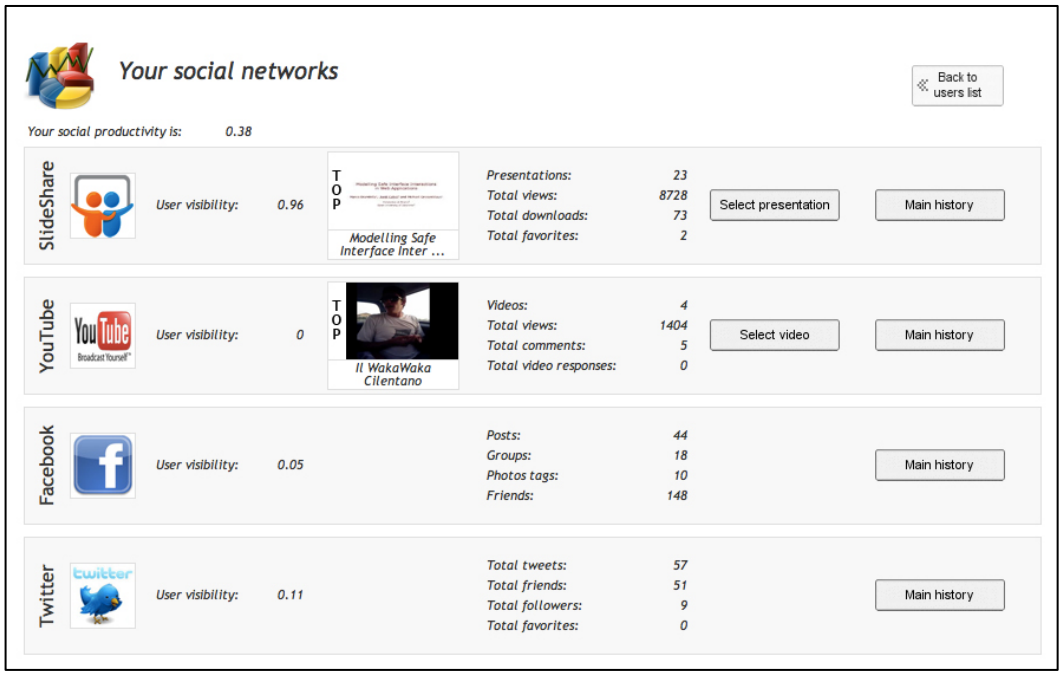

Figure 6 - Example of *"Unified visibility panel"* that resumes all the social activities conduced by a user with his different social profiles provided during the registration phase

Another important characteristic, introduced by the *"unified visibility panel"*, is provided by two indicators: the first of *"visibility"*, associated to each social network joined by the user, and the second of *"productivity"*.

Both indicators, calculated considering the data crawled during the last month by the application (Δ=1 month), have the main objective to give evaluations on the social user activity, but also to enable them to make comparisons with others, through the observation of the values computed.

The *"productivity"* indicator, one for each user, takes into consideration the main object on which, every social network considered, focus its attention; in the specific case, the number of items uploaded by the user for YouTube and SlideShare (that focusing on contents), and the number of posts for Twitter and Facebook (that focusing on individuals). The final value of the indicator is obtained summing all the variables previously descripted and opportunely weighted, dividing the result for the number of social profiles provided by the

user during the registration phase (fig. 7 shows the formula utilized in order to compute it).

$$Productivity = \left(\frac{\Delta\, num.\, tweets}{tt\_tweet\_pro\_wg} + \frac{\Delta\, num.\, videos}{yt\_video\_pro\_wg} + \frac{\Delta\, num.\, slides}{ss\_slide\_pro\_wg} + \frac{\Delta\, num.\, posts}{fb\_post\_weight}\right) \Big/ num.\ user\ profiles$$

Figure 7 - Formula utilized in order to obtain the value of the *"productivity"* indicator of a user

The *"visibility"* indicator, one for each social profile, describes the user popularity over a social network, considering different factors, on the basis of the information that it's possible to collect using the API. Like for the *"productivity"* indicator, the *"visibility"* value is computed as a mean of the sum of multiple variables opportunely weighted.

In order to clarify, how the values influencing it, have been chosen, an example of *"visibility"* indicator, is provided.

Figure 8 shows the formula utilized in order to compute the visibility of a user on Twitter.

$$Twitter\ Visibility = \left(\frac{\Delta\, num.\, followers}{\text{tt\_follower\_vis\_wg}} + \frac{\Delta\, num.\, status\ update}{tt\_state\_wg}\right) / 2$$

Figure 8 - Formula used in order to compute the *"visibility"* indicator of Twitter

The variables have been chosen on the basis of the following consideration:

- ✓ *num. followers*: increases the visibility not only because the user is inserted into the *"following"* list of the *"follower"* but also, because any time that the user posts a new tweet, the message is visualized on the profile's home pages of all the *"follower"* users.

- ✓ *num. status update*: influences the user visibility because tweets posted by him are visualized also on the *"public timeline"*, a sort of public wall where anyone can see messages posted by all the others (not necessarily present into his *"follower"* list).

Always considering the Twitter's *"visibility"* indicator, an important variable that was not possible to consider, due to the queries limit per hour of the API, was the number of *"retweets"*, that is the number of times that a message of a user is re-posted by *"followers"*.

The others *"visibility"* indicators, for the remaining social networks, has been conceived doing, on the quantitative information available, more or less the same considerations done for that explained above.

For both, the *"productivity"* and the *"visibility"* indicators, weights utilized into the formulas (see fig. 9), are calculated taking into considerations the average values, for each variable, related to a "common user" and a delta-time of a month. In particular, the default's weights values, have been obtained through the observation of the social activity related to a sample of ten profiles, each social network considered, computing every variable, as the mean value of the data collected.

	Indicators weights (Δ=1 month)			
	Variable	*Value*	*Variable*	*Value*
Productivity	tt_state_pro_wg	80	yt_video_pro_wg	4
	ss_slide_pro_wg	3	fb_post_pro_wg	100
Visibility	tt_follower_vis_wg	40	fb_page_fan_vis_wg	25
	tt_state_vis_wg	80	fb_friend_vis_wg	25
	yt_video_vis_wg	4	fb_group_vis_wg	50
	yt_view_vis_wg	400	fb_like_vis_wg	60
	yt_comment_vis_wg	30	fb_link_vis_wg	20

	ss_slide_vis_wg	3	fb_photo_tag_vis_wg	10
	ss_comment_vis_wg	15	fb_photo_vis_wg	7
	ss_download_vis_wg	50	fb_post_vis_wg	100
	ss_friend_vis_wg	8	fb_post_tag_given_vis_wg	20
	ss_view_vis_wg	300	fb_post_tag_received_vis_wg	15
	fb_album_vis_wg	1	fb_status_vis_wg	30
	fb_comment_vis_wg	50	fb_video_vis_wg	0,1
	fb_event_vis_wg	0,2	fb_video_tag_vis_wg	0,2

Figure 9 - Default weights used by the application in order to calculate the values of the *"visibility"* and *"productivity"* indicators

Nothing prevents that, default's weights and delta-time considered, can be changed by the administrator of the web application, modifying the values memorized into the configurations file *"application.properties"*.

In general, values showed by the indicators, can be interpreted as follow:

- ✓ If the value is between "0" and "1", it means, depending on the cases, that the user has less social *"productivity"* or *"visibility"*.
- ✓ If it's near "1" it means that he has a social *"productivity"* or *"visibility"* that is in line with the expectations for a "common user".
- ✓ More than "1" it means that the user has a social *"productivity"* or *"visibility"* that is higher than the normal expectation, meaning that he dedicates a lot of time to social networks.

3.2.1 Improvements on *"visibility"* and *"productivity"* indicators

At the current state, *"visibility"* and *"productivity"* values, introduced by the *"unified visibility panel"*, allow users to check their social activity performed through the profiles provided during the registration phase, and to make comparisons, accessing the same panel of others users.

With the aim to improve the indicators' readability, it would be interesting, through future developments, to replace the numerical values, displaying graphical bars (see fig. 10).

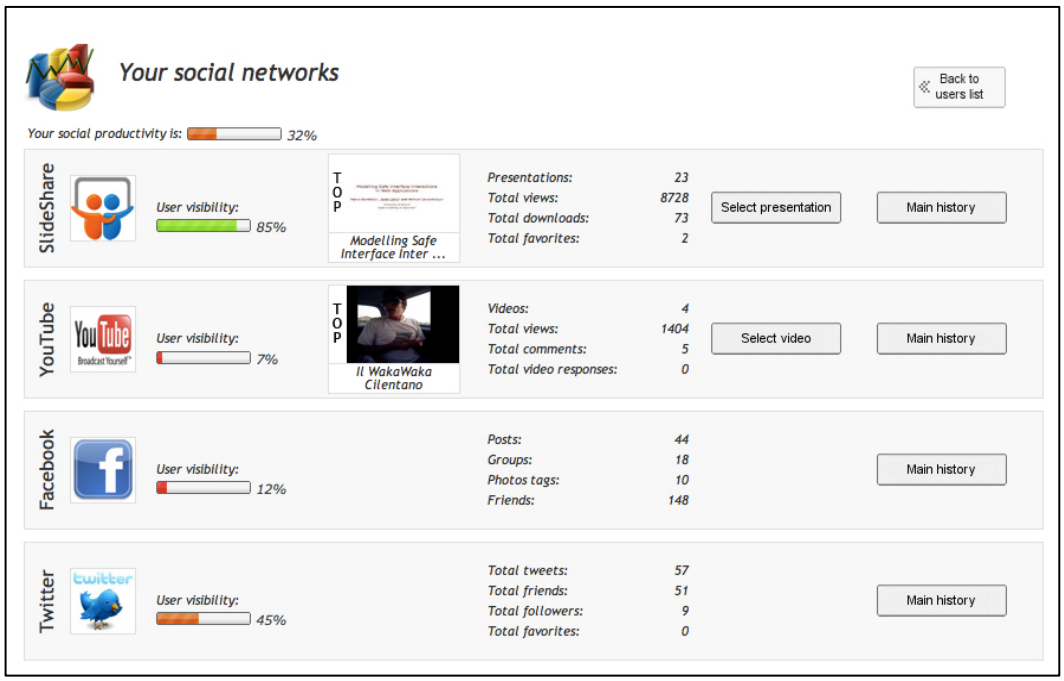

Figure 10 - The *"unified visibility panel"* after the improvements described on the indicators of *"visibility"* and *"productivity"*.

In order to apply this improvement, it's necessary to modify the structural data that contains information related to the users subscribed to the application, asking and memorizing, during the registration phase, information related to their job profession (student, professor, comedian, politic, etc.).

The following table (fig. 11) shows how, even the weights' values, are subjected to a changes, introducing the modification.

In particular, are shown the default weights used in order to calculate the user's *"productivity"* indicator, taking also into consideration its membership's category (job profession).

As for the case implemented, the values have been obtained observing, for each user's category, the social activity during a delta-time period (default Δ=1month), related to a sample of five profiles every social network considered.

Productivity indicator's weights (Δ=1month)	***Twitter*** *(tweets)*	***Facebook*** *(posts)*	***SlideShare*** *(presentations)*	***YouTube*** *(videos)*
Politics	80	250	0	60
Singers	130	100	0	5
Comedians	100	180	0	30
Professors	80	130	4	0,7
Students	50	200	0	0,2

Figure 11 - Default weights that will be used by the application in order to calculate the "productivity" indicator according with the user's category.

By introducing the modification explained above, the computation of an indicator, can be performed using as before the values collected, properly weighed, but making also a final normalization, considering all the data collected related to the others users of the same category, getting as a result, a value between "0" and "100", which can be displayed through the new graphical indicator.

The improvement proposed, not yet implemented, not only will permit to avoid sudden changes of the indicators (values too high or too low) due to possible inhomogeneity between subscribed users, but also, it will provide to them, the possibility to make a first comparison between the others subscribed of the same category, just looking, thanks to the normalization, to the graphical indicators.

Moreover, the *"accuracy"* of the indicators, will be increased, this due to the fact that, their computation will be based both, on the variable's weights, provided for each user's category, and on the final normalisation, performed taking into consideration, data daily collected, related to all the users of the same category.

3.3 Rendering temporal charts

Temporal charts gives the possibility to users registered to the application, to render their online social network history, through the visualization of quantitative data daily collected.

Navigating from the *"unified visibility panel"*, each user has the possibility to visualize charts in two ways: the first one, available for each type of social network, permits to see the general history of a social profile, clicking on the *"Main history"* button, while the second one, only for social network focusing on contents, enables to visualize historical charts related to a single item, through its selection.

The figure 12 shows an example of main history visualization, in particular charts rendered are related to a user subscribed to *"YouTube"*.

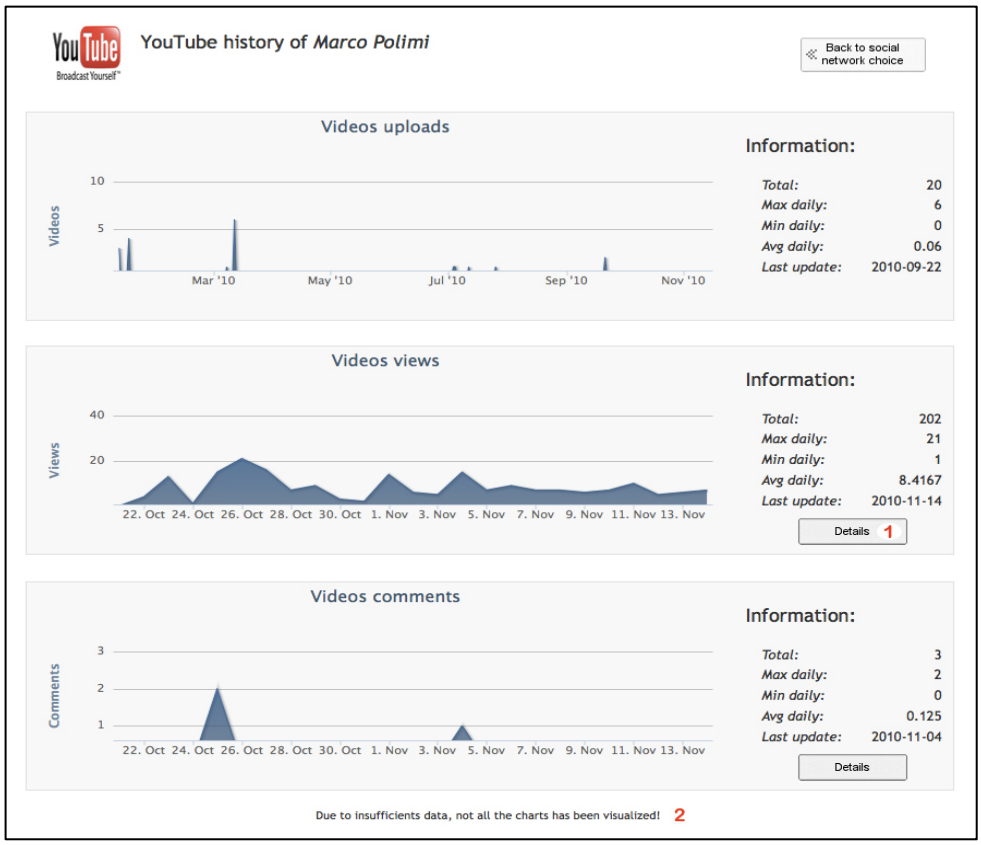

Figure 12 - Example of *"Main History"* page related to a user subscribed to YouTube

As it's showed, the user has the possibility to see different charts, everyone referring to quantitative data collected through the API and stored into the database of the application.

Temporal axis, utilized in order to render charts, are not always the same, in fact, depending on the availability of time information, it's possible to visualize the complete history about a user's value over the social network, like in this case for *"Videos uploads"* or, without temporal references, to render graphs considering as start time the registration date of the user to the application, providing for each day, the upgrades of the value observed, like for *"Videos views"*.

If the time frame showed into a graph, especially when time information are available, is too wide, the user can decide to zoom a part of the chart using the mouse, reducing in this way the temporal frame showed.

Depending on the level of the information details provided by the API, there is also the possibility to visualize advance details concerning a specific value, like for *"Videos views"* that, clicking on the related *"details"* button, they are rendered temporal graphs (see fig. 13), concerning the same value, for each video.

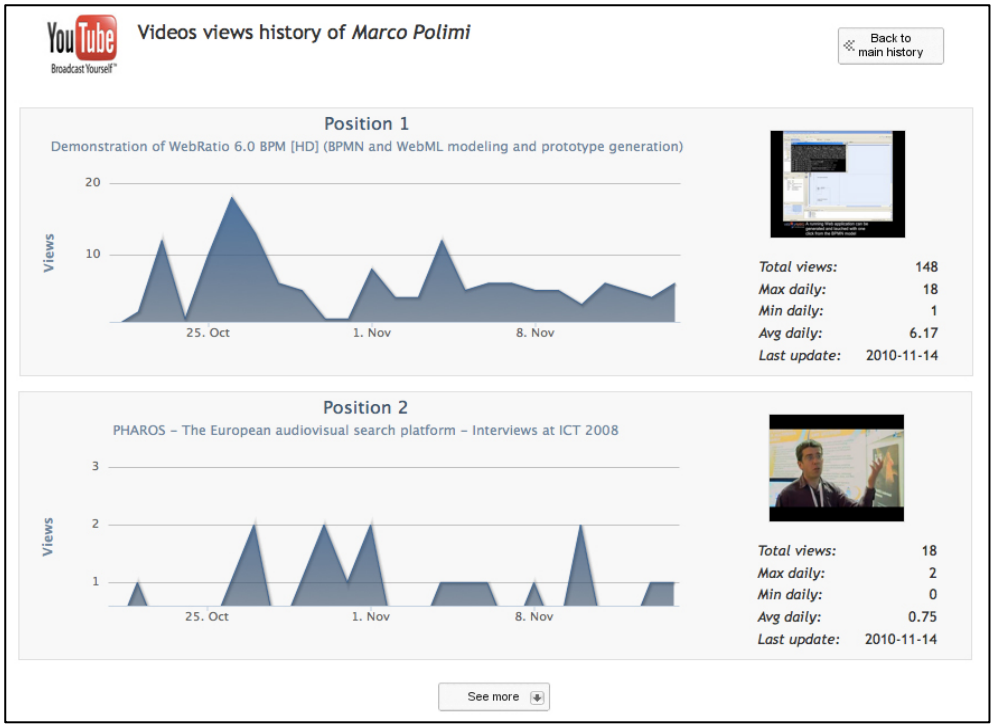

Figure 13 – Example of YouTube's detailed history, that shows charts rendered on the basis of the number of the views obtained for each video uploaded by a user.

As mentioned before, another way in order to visualize temporal charts, provided for social network focus on content, it's through the selection of an item, uploaded by the user. This can be done from the *"unified visibility panel"*, clicking on the button *"Select video"*, if the social network is YouTube or, *"Select presentation"* if it's SlideShare.

Fig. 14 shows how information and temporal charts related to a single item, in this case to a SlideShare presentation, are visualized.

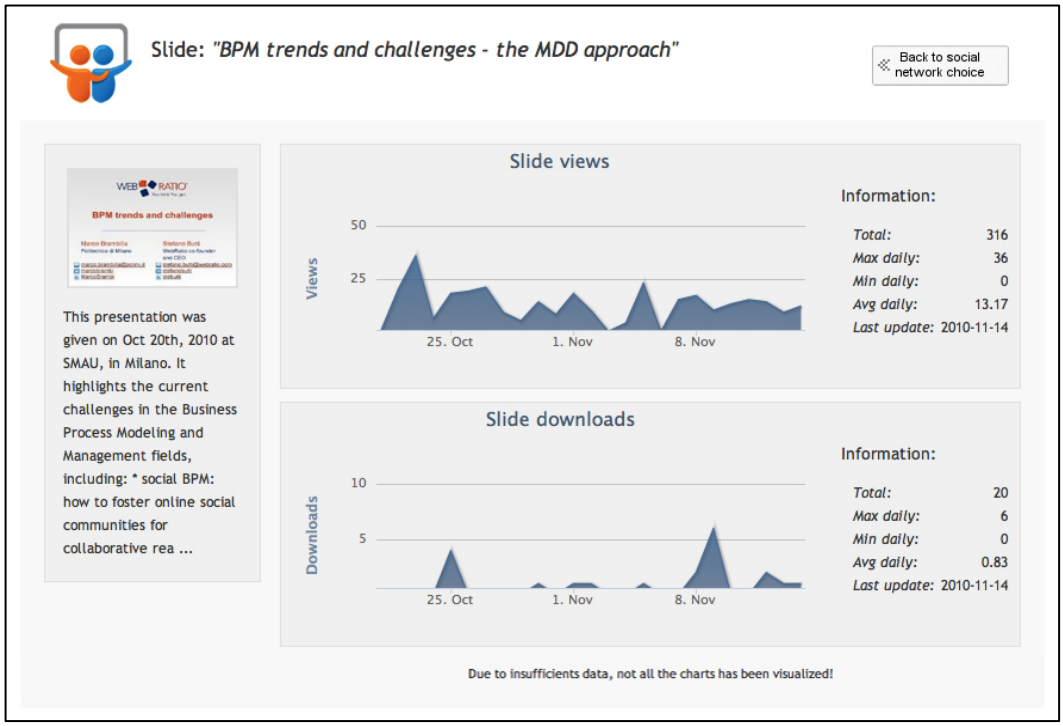

Figure 14 - Example of temporal charts rendered through the selection of a SlideShare's presentation uploaded by the user

It may happen, that a user is subscribed to a social network but, that he doesn't use his profile at all or, that he doesn't like some features provided by it; in both cases, a warn message, informing about the impossibility to render all or some charts, due to insufficient crawled data, is visualized.

4

PROJECT'S REALIZATION

4.1 Main requirements

The project has been implemented on the basis of the following main requirements:

Requirement	***Priority****	***Description***
Application type	1	The application has to be reachable by anyone over Internet through a common web browser, without the necessity of installing any program or plugin.
User registration page	1	A registration link has to be put on the main page in order to enable users that want to join the application to enrol their social profiles giving username/user id.
Real-time user ids check	3	During the new user registration all socials username/user id provided has to be checked in real time in order to avoid the enrolment of data that are not valid, providing in both cases visual feedback to the user.

Unified visibility panel	1	A page, with a panel indicating the main quantitative socials information collected, about the user selected, has to be provided.
Updates and Statistics sections	1	For each user the application has to be able to visualize two sections: the first one called *"update"*, that in real time must visualize the content update related to the user social profiles enrolled and the second one, called *"statistics"*, that must provide a section that summarize all the user's activities (in terms of quantitative data) over the different socials networks.
Social main history page	1	From the section *"statistics"*, the user has to be able to navigate through the page *"social main history",* that must show quantitative data related to the specific social network, visualizing different temporal charts.
Social detailed history page	2	Depending on the quantitative data available for each social network, also detailed charts related to the *"social main history"* has to be provided.
Socials data updates	1	Social quantitative data updates, related to each subscribed user, has to be daily collected and stored into a database in order to be able to show their complete history, starting from the subscription date.

* Priority goes from 1 (most important) to 5 (less important)

4.2 Implementation

Before starting the implementation of the application an analysis on the APIs of the main social networks has been conduced, trying to understand which of them provided important quantitative information necessary to the realization of the project, paying also particular attention to the privacy constraints.

In order to consider different types of socials networks, the following has been chosen:

- ✓ *SlideShare: a social network used to upload and share presentations with others.*

  Through a valid user account, an *API "key/secret"* have been requested and utilized in order to connect the application to SlideShare and to get users public information (like presentations, contacts, favourites, etc.), available without requests limits.
  Since the JSlideShare[8] API provided was based on the version 1.0, the oldest one, an upgrade to the new ver. 2.0 has been done in order to get more information.

- ✓ *YouTube: a video-sharing network on which users can upload, share, and view videos.*

  After the request of a valid *"client id"* and *"developer key"*, the Java YouTube's Data API have been used to retrieve YouTube's information (like comments, videos, friendships…) in the form of Google Data API feeds, without requests limits.

[8] *JSlidesShare*: is a Java wrapper around the SlideShare API. It can be used for mashing up slides a Java Web application.

- ✓ *Twitter: a micro blogging social network, enabling its users to send and read other users' messages, called tweets.*

  Through the request of a *"consumer key/secret"* associated to the user application developer profile, the java Twitter4J API have been used in order to get users public information (like followers, following, tweets, etc.). Due to the call-rate limitation and to the impossibility to put the application into the white-list (that mean no rate limit) detailed information (like tweets content, retweets, friendships details…) was not considered.

- ✓ *Facebook: social network intended to connect friends, family, and business associates.*

  In order to get information with the RestFB[9] library, the *"API key/secret"* associated to the application was not sufficient; an access token for each user, released after his authorization, has been used in order to get all the detailed profiles information.

[9] *RestFB*: is a powerful Facebook Graph API and Rest API client written in Java.

4.3 Main APIs utilized

In order to realize the web application, implementing all the requirements specified above (see 4.1 *"Main requirements"*), it has been necessary to utilize different Java's libraries. In particular, between the main APIs adopted, there were those that have permitted to collect the information related to the users' profiles over the different social networks and to store them into the database, as well as, those that have allowed to schedule the daily collection of the social quantitative information.

A list, presenting the main Java's libraries utilized in order to implement this thesis project, with a small description that justifies their utilization, is provided.

- ✓ *Database connection and XML files management*: the following APIs have permitted respectively to memorize into the database the crawled information related to the social updates of the users' profiles, and to manage the XML file containing all the social profiles of the users registered to the web application.
  In particular:
  - ***MySQL (mysql-connector-java.jar)***: through the JDBC driver has provided the possibility to connect the application to a MySQL database [20].
  - ***JDOM (jdom_1.1.1.jar)***: it has enabled the application to access, manipulate and output XML data [21].

- ✓ *Log of information and errors*: the subsequent API has enabled to log all the information concerning the working status of the application developed, proving textual feedback in both cases, during the normal functioning, as well as, in case of execution errors.

    - ***Apache Log4j (apache-log4j-1.2.1.jar)***: it has permitted to log all the application behaviour [22].

- ✓ *Daily collection of the quantitative information*: the following API has permitted to set a daily trigger in order to update automatically the social information, memorized into the database, related to all the social profiles registered to the application.

    - ***Quartz (quartz_1.8.3.jar)***: it has provided a full-featured, open source job scheduling service [23].

- ✓ *APIs used in order to connect the application to the different social networks*: the subsequent APIs have permitted to get all the information made available by the different social networks taken into consideration by the application developed.
  In particular:

    - ***Google Data APIs (gdata-src.java-1.41.0.jar)***: they have enabled the application to get information related to YouTube using the YouTube's Data APIs included into them [24].

    - ***Twitter4J (twitter4j-core-2.1.6.jar)***: unofficial Java library that has permitted to get, through the Twitter APIs, information related to Twitter [25].

- ***RestFB (restfb_1.5.4.jar)***: it has permitted to the application to get information about users subscribed to Facebook using the Facebook Graph API [26].

- ***JSlideShare (jslideshare.jar)***: it has enabled the application to get information from SlideShare. Since it has been provided only an old version of the library, based on ver. 1.0 of the APIs, an update to the version 2.0, that gives more information, has been done [27].

4.4 Structure of the web application

Figure 15 shows how the web application created is structured and the main communications between client and server that permits to visualize information.

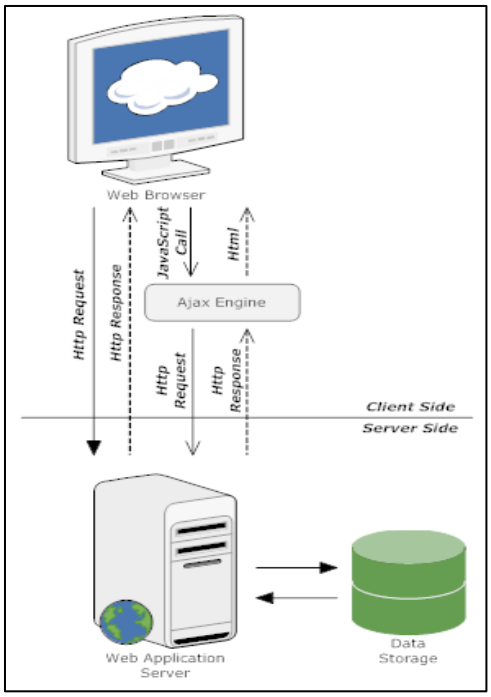

Figure 15 - Structure of the web application and main client/server communications

Http requests coming from a client web-browser can be synchronous or asynchronous, depending on the page requested and on the type of content to be visualized in it.

The Ajax Engine is utilized in particular during the user registration phase in order to check in real-time the provided profiles information through the servlet *"SocialUserDataValidation"*. It's also used during the visualization of pages containing graphs in order to get the render settings (like charts titles, data, etc.), by forwarding an asynchronous request to the servlet *"ChartsDataGeneratorDispatcher"*.

In order to manage and support jsp pages and servlets, Apache Tomcat has been used as *"Web Application Server"*.

"Data Storage", contains the profiles information of the users registered to the application and the social data, daily collected through a scheduler, initialized during the start up of the web server.

4.5 Data storage

In order to store all the information collected with the application, two type of data container has been used:

- *XML (Extensible Markup Language)*: a markup language, produced by the W3C, for documents containing structured information, designed to store and transport data [28], adopted to memorize the social profiles of the users enrolled to the application. This solution in order to be able to get and reuse easily the profiles information for other future objectives.

- *MySQL*: an open source relational database management system (RDBMS) that relies on SQL for processing data, commonly used in web application [29], adopted with the creation of a database, called *"socialDB"*, containing specific tables enabling the application to store all the socials quantitative data collected.

4.6 Users' list structure

The *"UsersList.xml"* file contains all the social profiles information related to the users that decide to join the application (see fig. 16). It is structured as follow:

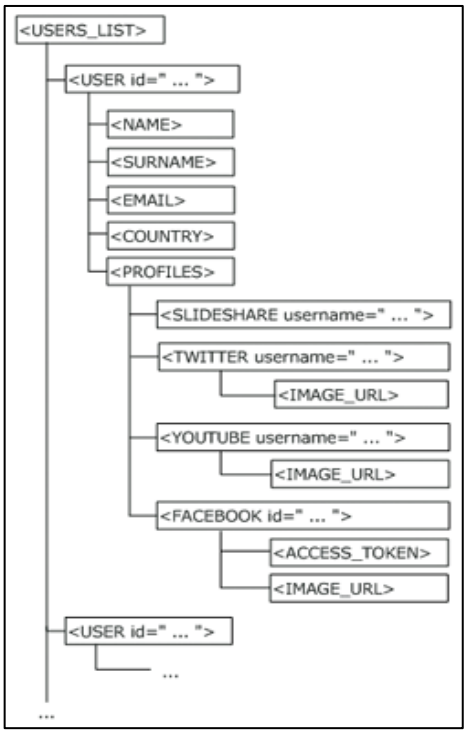

Figure 16 - Structure of the UsersList.xml file

- ✓ *Tag USERS_LIST*: is the root element containing all the users' information.
- ✓ *Tag USER*: node element that contains data related to a single user, both personal information (like name, surname…) and social profiles.
    - *Attribute id*: it's a unique key identifier generated with a hash function applied on the value contained into the *"EMAIL"* tag.
- ✓ *Tag NAME*: contains the name of the user.
- ✓ *Tag SURNAME*: its value is related to the user's surname.
- ✓ *Tag EMAIL*: contains the user's email.
- ✓ *Tag COUNTRY*: element value related to the user's nationality.

- ✓ *Tag PROFILES*: node element that contains all the user's social profiles.

- ✓ *Tag SLIDESHARE*: element that contains information related to the SlideShare user's profile.

    - *Attribute username*: its value is related to the username that identify the user on SlideShare.

- ✓ *Tag TWITTER*: node element containing information related to the Twitter user's profile.

    - *Attribute username*: contains the Twitter user's identifier.

- ✓ *Tag YOUTUBE*: node element that contains information related to the YouTube user's profile.

    - *Attribute username*: its value is related to the YouTube user's identifier.

- ✓ *Tag FACEBOOK*: node element containing information related to the Facebook user's profile.

    - *Attribute id*: contains the unique identifier of the Facebook user's profile.

- ✓ *Tag ACCESS_TOKEN*: element containing the Facebook's access token related to the user's profile. This value is present only if the user grants the permission to the application.

- ✓ *Tag IMAGE_URL*: element value that contains the URL related to the user's profile thumbnail.

4.7 Structure of the database

In order to memorize the quantitative data daily collected from the different socials networks, a database called *"socialDB"* has been created.

The definition of the tables has been done taking into consideration the information that the APIs was able to extract.

During the data analysis, taking into consideration temporal aspects, more important in order to construct the history charts, two main type of information has been identify:

1. Data with temporal references
2. Data without temporal references

For the first category thanks to the availability of the creation time, the application simply memorizes the date with the related information into the specific table, while, for the second, a particular schema has been adopted in order to memorize temporal variations of the value observed (see fig. 17).

id *	object_id	date	value	type_value

Figure 17 - Schema of the table adopted for memorize data without temporal references

Tuples, into this type of tables, are inserted as follow (see also fig. 18):

- ✓ The fist time the object is crawled, the field *"type_value"* is set to *"init"*, and *"value"* contains the real numerical data collected.
- ✓ For each subsequent updates, related to the same object, the field *"type_value"* is set to *"update"* and *"value"* contains the upgrade (difference between the new numerical data collected and the sum of all the previous values inserted with the same object id).

id	slide_id	value	date	type_value
9	3905590	858	2010-10-19	init
14	3905590	3	2010-10-20	update

Figure 18 - Example of table containing data without temporal references

Despite more data collected has been required the same table schema, due to the large amount of tuples inserted in it, the solution adopted was that to keep separate tables containing different types of information.

4.7.1 *"socialDB"* tables

Analysing the APIs found for the different socials networks, the following tables has been created in order to memorize all the social quantitative information available.

YouTube

- ✓ *yt_video_info*: contains detailed information about the videos uploaded by the users that decide to join the application. Due to the fact that the insertion time is available, the complete user's uploads history can be memorized.
- ✓ *yt_view*: since for video views no temporal references are available the table contains the *"init"* view value and the subsequent upgrades related to each video.
- ✓ *yt_comment*: contains information about the number of comments received by each crawled video.
- ✓ *yt_video_response*: memorizes quantitative information about the number of video responses received by each video.
- ✓ *yt_favorite*: contains the *"init"* count value of the favourite's videos related to each user that is registered to the application and the subsequent upgrades.
- ✓ *yt_friend*: memorizes information about the number of friendships joined by each registered user.

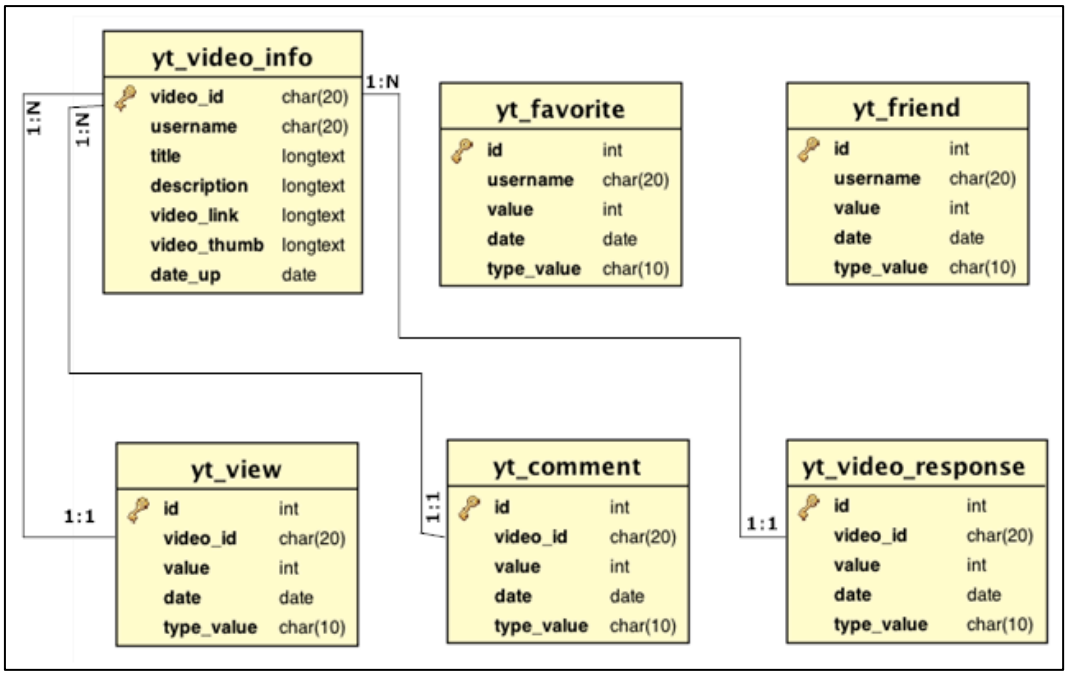

Figure 19 – YouTube's content tables created into *"socialDB"*

SlideShare

- ✓ *ss_slide_info*: contains detailed information about all the presentation uploaded by registered users. Thanks to the availability of the creation time, the complete user's uploads history can be reconstruct.
- ✓ *ss_download*: contains information about the initial number of downloads and subsequent values upgrades related to each presentation.
- ✓ *ss_comment*: memorizes the number of comments received by each presentation and the relatives values upgrades.
- ✓ *ss_favorite*: contains information about how many users consider a presentation as favourite.
- ✓ *ss_view*: memorizes the initial views' values and their subsequent upgrades related to each presentation.

- ✓ *ss_friend*: contains information about the number of friends presents in the friendships' list and the relative value's upgrades.

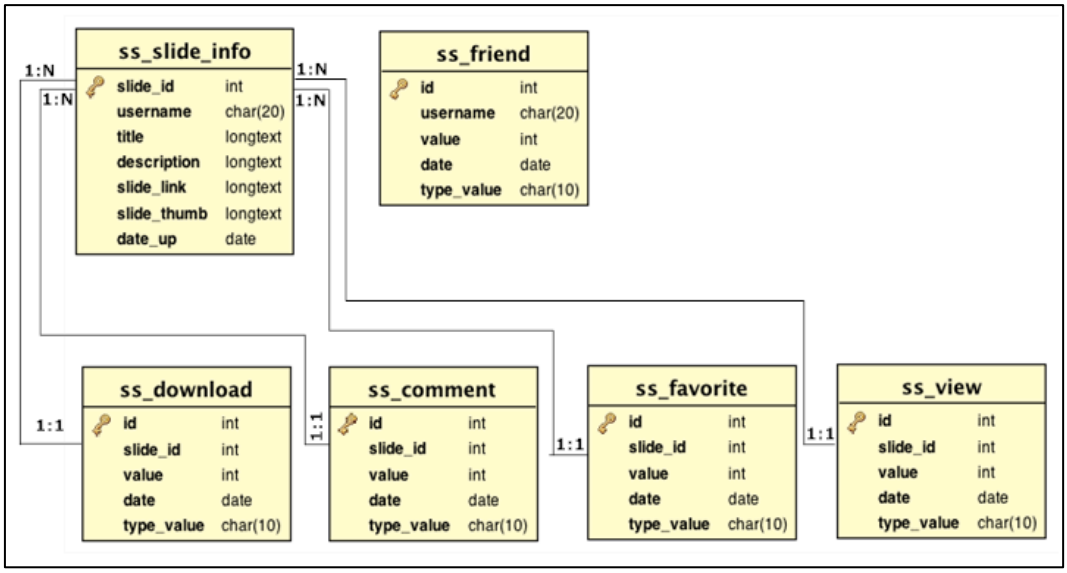

Figure 20 - Tables related to SlideShare contained into *"socialDB"*

Twitter

- ✓ *tt_user*: memorizes the main profiles' information about the users registered to the application.
- ✓ *tt_state*: contains the tweets *"init"* value for each registered user and the subsequent upgrades.
- ✓ *tt_friend*: memorizes the number of the user's *"following"* people and the relatives upgrades' values.
- ✓ *tt_follower*: contains, for each user registered to the application, information about the "init" value of the users' *"followers"* and of the subsequent upgrades.
- ✓ *tt_favorite*: memorizes the number of tweets that the user mentions as favourite and the relative upgrades.

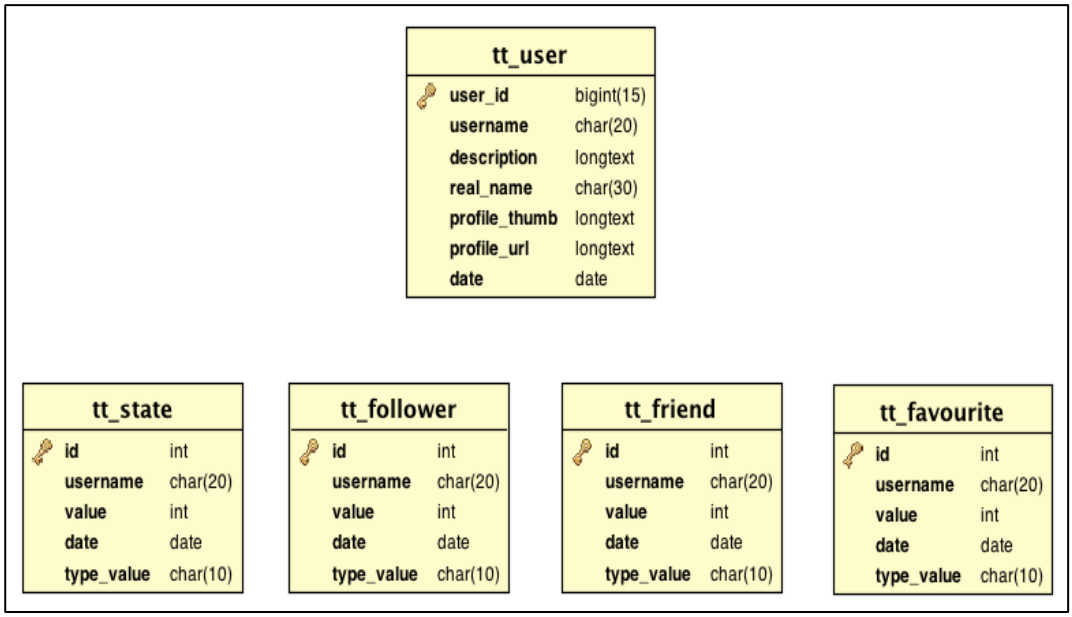

Figure 21 – Twitter's tables created into *"socialDB"*

4.7.1.1 Facebook

- ✓ *fb_user*: contains the main profile's information related to the users registered to the application.
- ✓ *fb_status*: memorizes all the status' changes done by the users through the Facebook wall. Since the creation time is available the complete status' history can be reconstruct.
- ✓ *fb_post*: contains information related to all the posts done by the users, visible looking the user's profile page. Due to the possibility to get the creation time it's possible to reconstruct all the users' posts history.
- ✓ *fb_post_tag_given*: memorizes, for each user registered to the application, information about the *"init"* value of the posts tagged by the user and the subsequent upgrades.

- ✓ *fb_post_tag_received*: contains information related to the number of tags received by other users. Since no time references are available, the table memorizes, for each user, the *"init"* value and the subsequent upgrades.

- ✓ *fb_event_joined*: memorizes information related to all the events joined by the user. The availability of the creation time permits to reconstruct all the user's events joined history.

- ✓ *fb_video*: contains information about videos uploaded by the user. Since the creation time is available the complete videos' uploads history can be reconstruct.

- ✓ *fb_video_tag*: memorizes information about videos in which the user is tagged. The presence of the creation time enables the complete reconstruction of the user's video tags history.

- ✓ *fb_photo*: contains information about the number of photo uploaded by the user. No temporal references are available.

- ✓ *fb_photo_tag*: memorizes the photo tags received by the user. Due to the availability of the creation time is possible to reconstruct the complete photo's tags history.

- ✓ *fb_page_fan*: contains information about the number of pages which the user is fan and the subsequent upgrades.

- ✓ *fb_link*: memorizes the links published by the user on the social platform. The presence of the creation time permits the complete reconstruction of the user's links history.

- ✓ *fb_like*: contains the number of *"I'd like"* received for the post published by the user. Temporal references are implicitly given by the date associated to the posts.

- ✓ *fb_group*: memorizes the number of groups joined by the user and the relative upgrades of the value.

- ✓ *fb_friend*: contains information about the total friends of the user. No temporal references are available.

- ✓ *fb_album*: memorizes the albums uploaded by the user. Due to the presence of the creation time the complete history can be reconstruct.

- ✓ *fb_comment*: contains all the comments' information about the posts done by the user. The creation time permits the reconstruction of all the comments history.

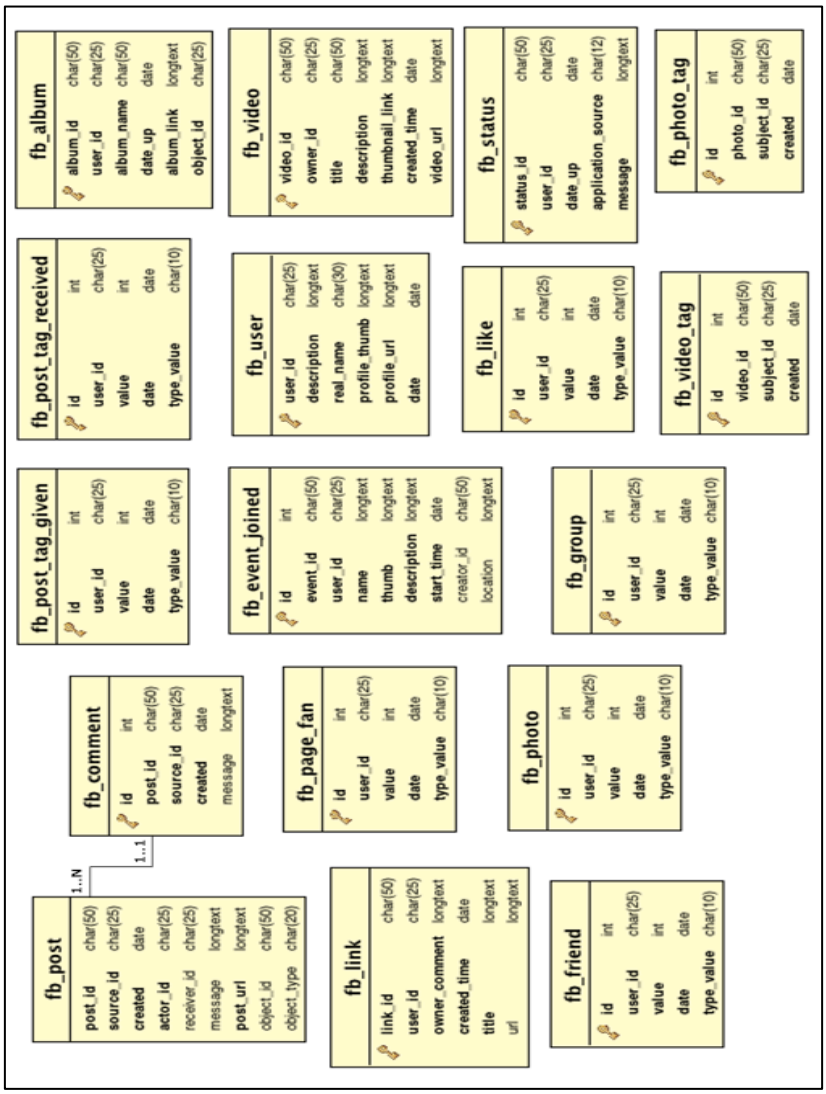

Figure 22 – Facebook's tables created into *"socialDB"*

4.8 Classes packages

A UML diagram, that shows the default package of the application developed, is presented in figure 23. It defines packages containing definitions for all utility classes and servlets that compose the core of the application.

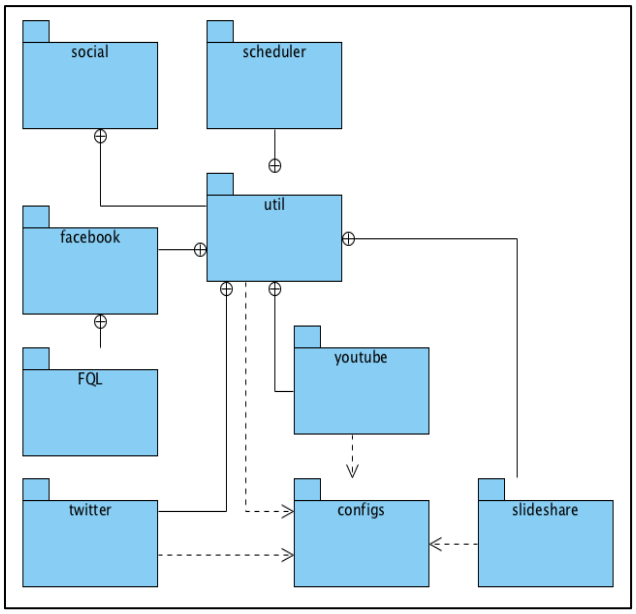

Figure 23 – Application's packages structure

Although it does not directly define any classes, nested sub-packages called *"social"* and *"configs",* are defined within the scope.

Inside the package *"configs",* there is a class called "*ConfigProperties"* providing, to classes of others packages (see image), connections information related to the database or to social networks, stored into the file *"application.properties"*.

The package *"social"*, does not define any class but, it contains a sub-package called *"util"* in which important servlet, like *"SocialUserDataValidation"*, responsible to validate the social ids of a new users and, *"SocialApplicationContextListener"*, that initialized the connections to the social networks, are defined.

Into *"util",* others sub-packages are presents: *"scheduler"*, in which classes reliable to start up the job related to the daily social information updates, are defined, and furthers sub-packages, one for each social network, defining two main classes, the first one providing methods used by the scheduler during the social information update, and the second one, used by the servlet *"ChartsDataGeneratorDispatcher",* in order to reconstruct social histories requested by the user.

Due to the necessity to get detailed information concerning the users' Facebook profiles, the FQL[10] [30] language, enabling to read objects of the Facebook social graph, has been used.

Its adoptions has required the creation of a "*facebook"* sub-package, called *"FQL"*, containing classes used in order to hold the results of the queries.

[10] *FQL (Facebook Query Language)*: it enables to use a SQL-style interface to query data of the Facebook Social Graph.

4.9 Main features

Descriptions on the main features of the application, as well as, on the most important interactions with the end user, are presented.

4.9.1 New user registration

The registration of a new user can be divided into two phases. The first one, where the user requires the registration page and fills the apposite form providing both, personal information (like name, surname, etc.) and social information (like id, username, etc.), checked in real time before the submit, and the second one, in which the application receives the registration's request, and proceed with the memorization of the user information, asking, if the user has provided also the Facebook's identifier, the authorization[11] for the application.

[11] *Facebook Authorization*: by default, the application can access all general information contained in a user's profile, including name, profile picture, gender. If the application needs to access other parts of the user's profile that may be private, the application can request extended permissions.

First registration phase

The sequence diagram of figure 24, describes how the first registration phase is performed.

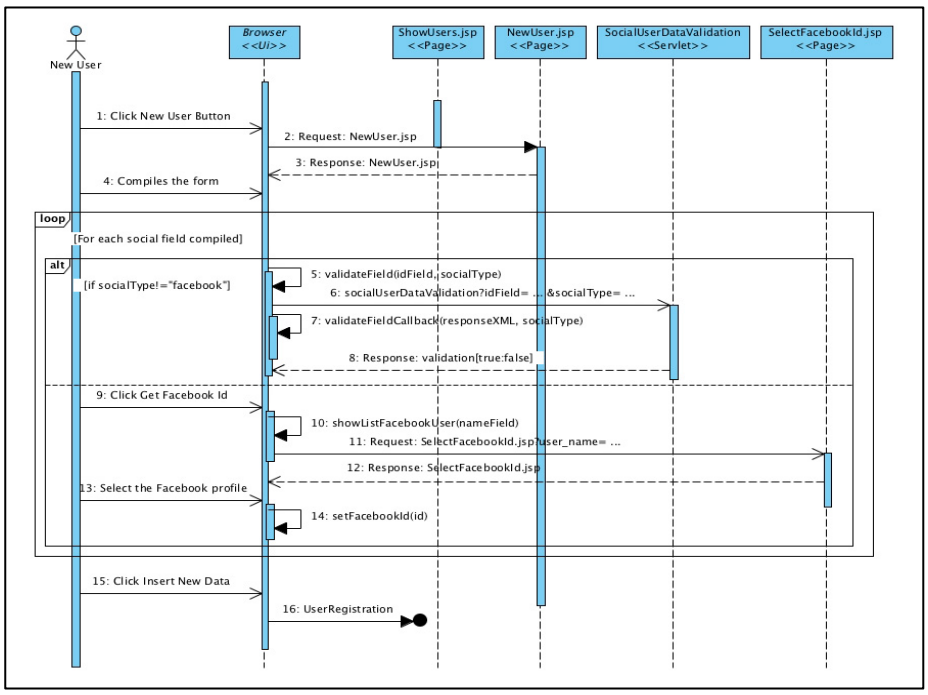

Figure 24 - Steps performed during the first registration phase

Follow a briefly description of the sequences of operations done:

1. The user from the main page of the application clicks the "*Add New User*" button.

2./3. The browser forwards the request to the application server that response providing the html code containing the form to be filled by the user.

4. The user compiles the form fields.

[for each social field different from "Facebook"]

5. The JavaScript function *"validateField(...)"* is called.

6. An asynchronous POST request, is forwarded to the application server in order to call the "*SocialUserDataValidation"* servlet, that check if the social information provided exists.

7. The JavaScript function *"validateFieldCallback(...)"* is called and waits the response of the validation servlet.

8. The validation response is catch by the *"validateFieldCalback(...)"* function that, depending on the result visualizes a graphical feedback to the user.

[field=="Facebook"]

9. The user, after the specification of his Facebook's name, clicks the *"Get Facebook Id"* button in order to select his profile on the social network.

10. The JavaScript function *"showListFacebookUser(...)"* is called.

11. A frame is opened and the page "*SelectFacebookId"* is requested to the application server.

12. The page "*SelectFacebookId"*, containing all the users with the name previously specified, is visualized inside the frame.

13. The user selects his Facebook's profile.

14. The JavaScript function *"setFacebookId(...)"* is called. The id of the user is set into a hidden field presents in the parent window *"New User Page"* and the frame is closed.

15. The user clicks the *"Insert New Data"* button.

16. The new user registration is performed.

Figure 25, shows the sequences of operations done by the servlet *"SocialUserDataValidation"*.

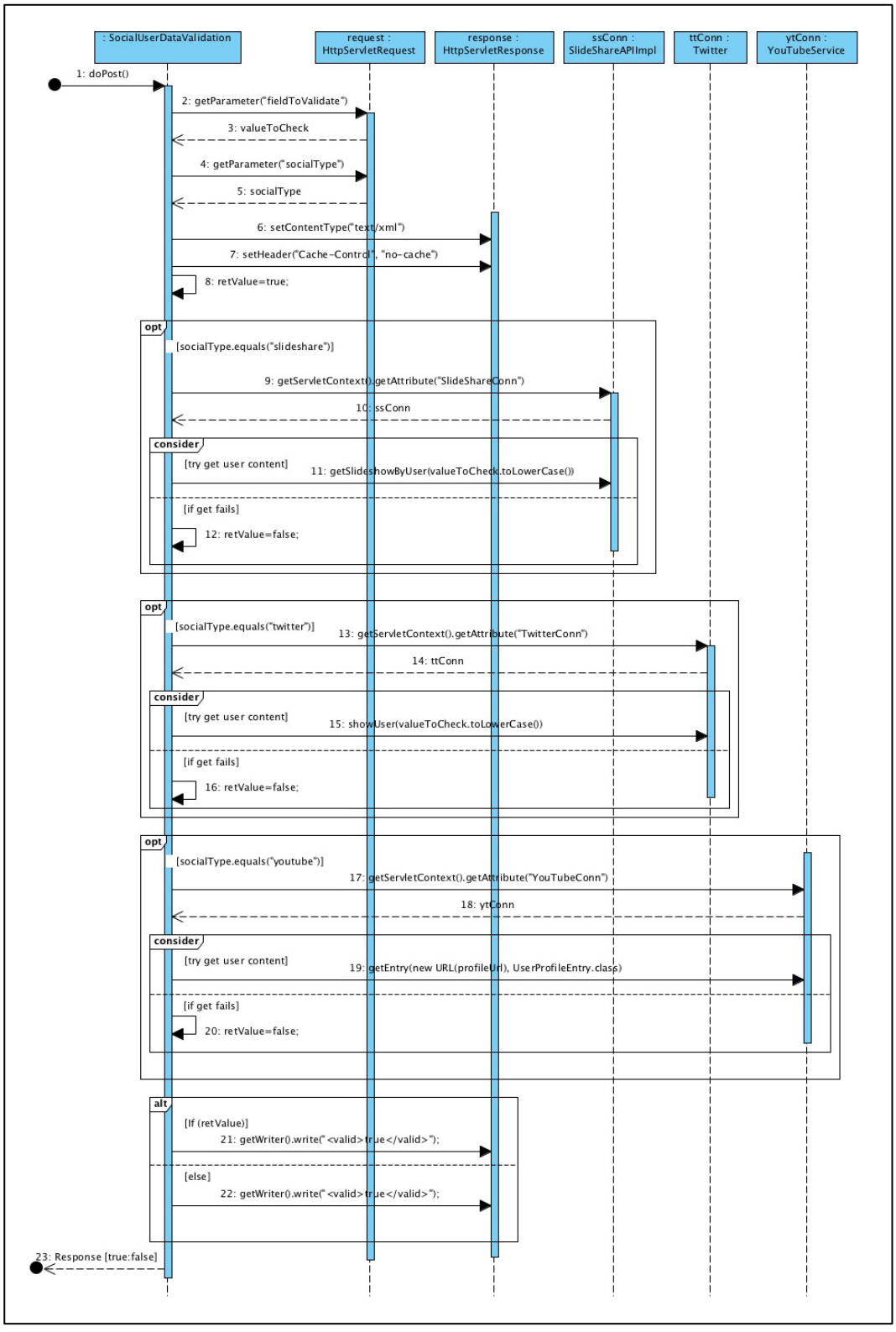

Figure 25 - Steps performed in order to validate the social profiles

The APIs used in order to get information from the different social networks, doesn't provide methods that give the possibility to check if a social username/user id, really exists.

In order to solve this problem, some tests have been done, identifying that, information requested through invalid usernames/users id, raises exceptions.

Due to this consideration, the implemented validation servlet (see fig. 25), try to get social content with the specified username/user id, if the method fail, the exception handler sets *"retValue"* (previously set to true) to false, and the validation's result, is sent back to the client.

Second registration phase

The memorization of the new user's information is performed by the servlet *"UserRegistration"* (fig. 26), that receives data through a client POST.

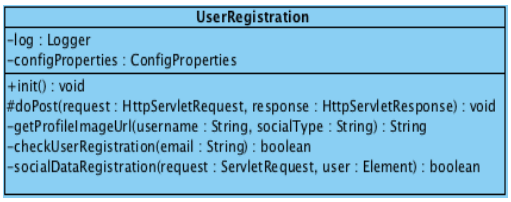

Figure 26 - Class diagram related to the servlet *"UserRegistration"*

The method *"checkUserRegistration()",* evaluates if the user is already present calculating a hash function over the email's value (in this way is created the user's identifier), and making a comparison with all the ids of the users registered.
If the user is not present, the registration of the new data is performed, and a confirmation page is showed, else, the user is redirect to a fail page, showing the failure's motivations.

Whether the user provides also his Facebook's id, between the registration phase and the confirmation page visualization, some additional steps has to be computed, this due to the fact that with Facebook, by default, an application can access all the user's profile general information (including name, profile picture, gender, etc.) but, if it needs to access others information, that may be private, an *"extended permissions"* request, has to be performed.
In order to authorize the application to get access to the user's id, the Facebook platform uses the OAuth 2.0[12] protocol, associating to each user an access token, used to get information from his profile.

[12] *OAuth 2.0*: is a simpler version of OAuth that leverages SSL for API communication instead of relying on complex URL signature schemes and token exchanges.

Figure 27, shows how the Facebook application's authorization is performed.

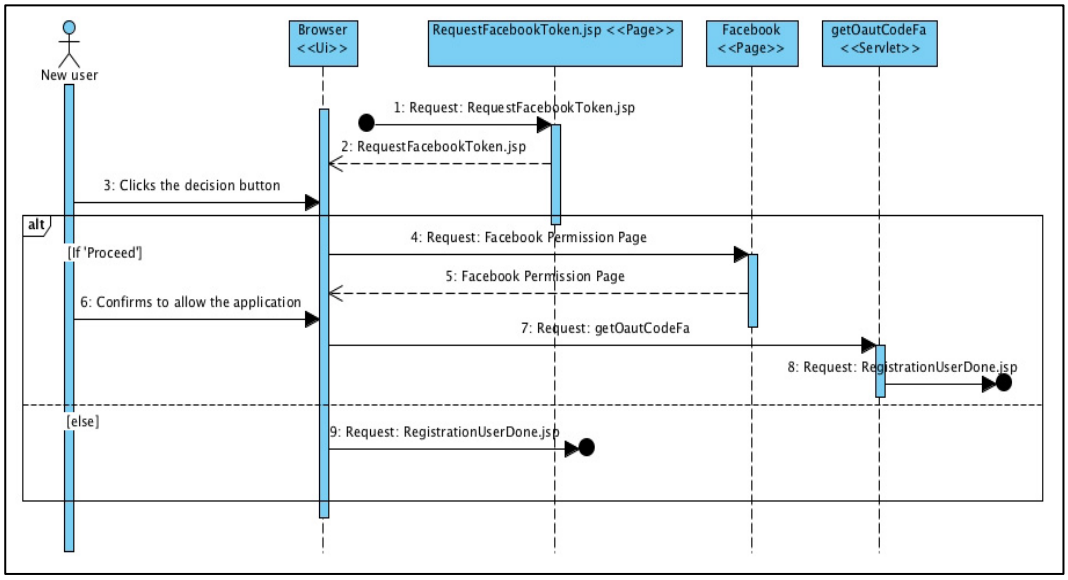

Figure 27 - Sequence Diagram of the main steps done during the Facebook application authorization

Follow a briefly description about the sequences of operations carried out:

1./2. The *"UserRegistration"* servlet makes a redirect to the page *"RequestFacebookToken"* that is visualized on the client's browser.

3. The user clicks the decision button in order to grant or not the Facebook's authorization.

   [if "Proceed"]

   4. The Facebook's authorization page is requested using the URL shows in figure 28 that specify the *"client_id"* associated to the application, the *"redirect_uri"* that is the URL where the user should be redirect after the authorization process, the user's *"facebook_id"* and the *"scope"* argument including a comma separated list of all the permissions requested.

```
https://graph.facebook.com/oauth/authorize?
client_id=...&
redirect_uri=http://musall.webhop.org/Tesi2Level/getOautCodeFa?facebook_id=...&
scope=user_events,read_friendlists,read_stream,...,offline_access
```

Figure 28 - URL utilized in order to request the Facebook authorization page

5. The Facebook's server responses with a page visualized on the client's browser, through which the user can confirm the authorization to access the specific information concerning his profile.

6./7. The user confirms the authorization to the application. A redirect to the servlet *"getOautCodeFa"*, previously defined into the *"redirect_uri"*, is performed. A verification string parameter, called *"code"*, necessary to get the OAuth access token, is attached to the request.

The servlet *"getOautCodeFa",* opens a connection through the URL shows in figure 29, passing as request parameters the *"cliend_id"* and *"client_secret"* related to the application, the same *"redirect_uri"* used in the step 4 and the *"code"* value received has parameter.

```
https://graph.facebook.com/oauth/access_token?
client_id=...&
redirect_uri=http://musall.webhop.org/Tesi2Level/getOautCodeFa?facebook_id=...&
client_secret=...&
code=...
```

Figure 29 - URL utilized in order to request the OAuth access token

8. Once the user access token returned by the request above is received, the servlet proceed with its memorization into the *"usersList.xml"* file, as a child element of the user's *"FACEBOOK"* node and a redirect to the *"RegistrationUserDone"* page, is performed.

[else]

9. A redirect to the *"RegistrationUserDone"* page is performed but, since the user doesn't provide the Facebook's authorization, the social profile is not considered by the application.

4.9.2 Social user's history visualization

In order to visualize the social history of a user, a pure JavaScript library called HighCharts, enabling to add interactive charts to websites, has been used.

Highcharts is solely based on native browser technologies and doesn't require client side plugins like Flash or Java [31]. It needs only two JavaScript files to run: the *"highcharts.js"*, that is the core of the library, and *"jQuery"*, a library that makes quicker and easier to write JS code.

Data to be visualized into charts, are provided by the servlet *"ChartsDataGeneratorDispatcher"*, called through a jQuery GET method, that depending on the types of charts requested, instantiates a specific data history class constructor.

Figure 30, shows the steps performed, when a page containing histories charts, is requested.

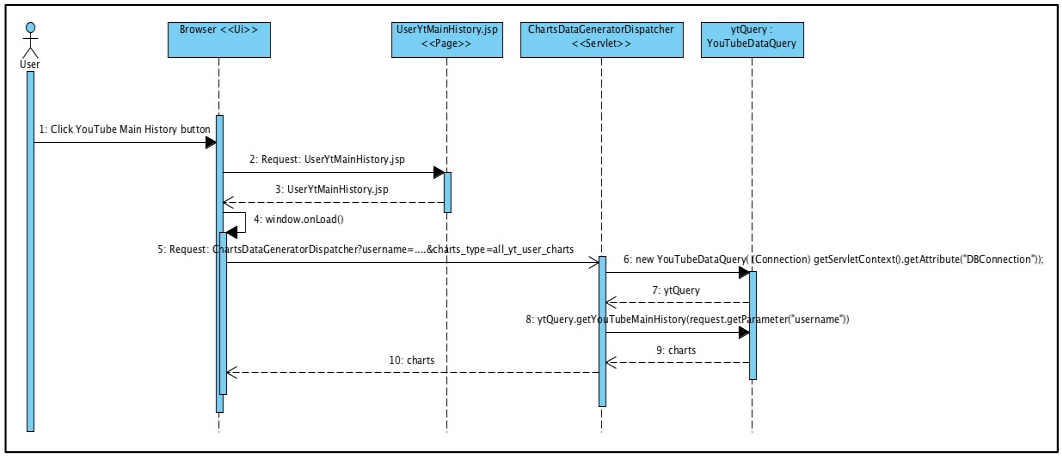

Figure 30 - Sequence Diagram representing the main steps done in order to show a page containing histories charts

Follow a briefly description about the sequences of operations performed:

1. The user clicks on the browser the YouTube's *"Main History"* button.

2. A request for the page *"UserYtMainHistory"* is forwarded to the Application Server.

3. The page *"UserYtMainHistory"* is returned to the client's browser.

4. The JavaScript function "*onLoad()"*, is executed.

5. An asynchronous request toward the servlet *"ChartsDataGeneratorDispatcher"* is performed, specifying as parameters the user's identifier and the type of charts needed.

6./7. The servlet, instantiates an object of a specific data history class constructor, depending on the charts' type requested (in the example, instantiate *"YouTubeDataQuery"*).

8./9. The method *"getYouTubeMainHistory()"* is invoked in order to collect data stored into the *"socialDB"*, necessary to visualize the requested user history charts.

10. Data related to the user's history are returned to the client, and through the Highcharts library, the charts are rendered.

4.9.3 Socials information update

The daily update of the social profiles information, necessary in order to reconstruct the social history of the users, is done automatically through the utilization of *"Quartz"*, an open source job-scheduling framework, based entirely on Java [23].

The scheduler, is initialized during the server start up by the *"QuartzInit"* servlet, that create a job called *"UpdateSocialDBJob"* and set-up the hour for the daily trigger.

Figure 31, shows the sequence of operations done by the class *"SocialDBUpdateJob"* in order to update the socials information.

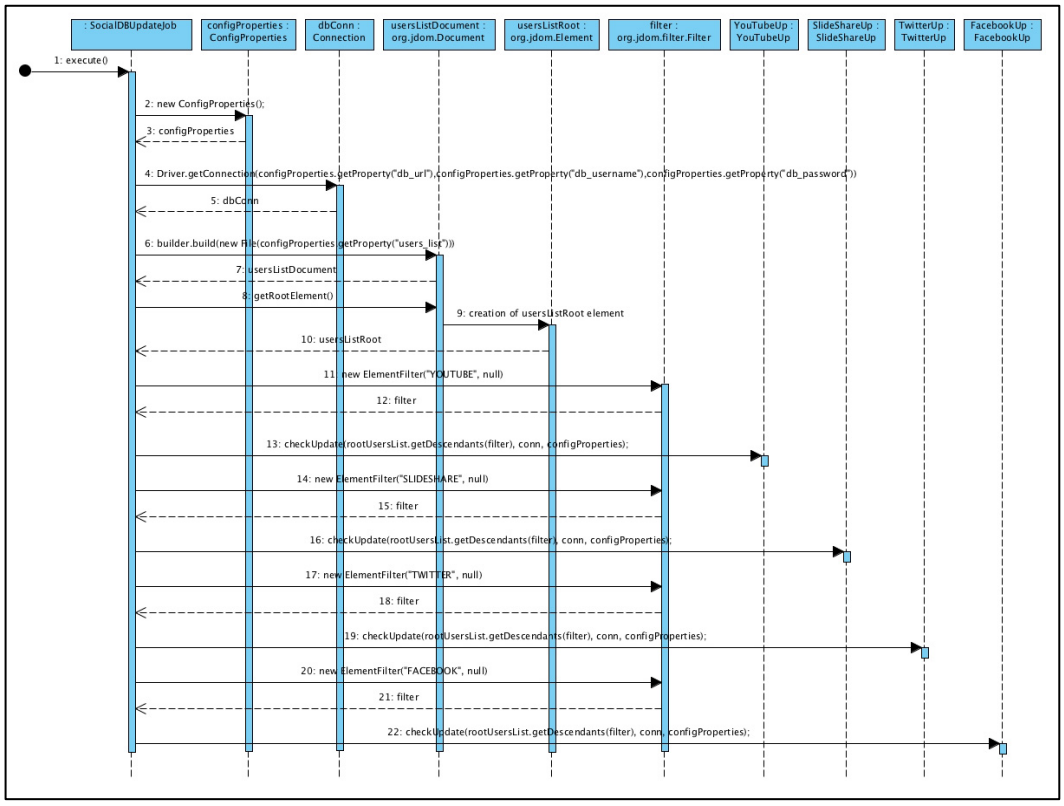

Figure 31 - Sequences of steps performed by the class SocialDbUpdareJob in order to collect the new socials information.

When the job is triggered, the method *"execute()"* of the class *"SocialDBUpdateJob"* is invoked, and the following operations are performed:

2./3. A new object *"ConfigProperties"*, enabling to get all the configuration properties of the application, is created.

4./5. A connection to the *"socialDB"*, necessary to store the new socials quantitative information, is created.

6./7. The *"UsersListDocument"* file, containing all the information related to the users registered to the application, is opened.

8. The method *"getRootElement()"* of the *"UsersListDocument"* is invoked.

9./10. A new object called *"UsersListRoot"*, containing the root element of the *"UsersList.xml"* file, is created.

11./12. An *"ElementFilter"* for the social network YouTube is created and used in order to filter the social profiles.

13. The method *"checkUpdate()"* of the class *"YouTubeUp"* is invoked, passing as parameters the YouTube's profiles, the database connection and the *"configsProperties"* object necessary in order to instantiate the connection with social network. The new information related to the YouTube's profiles are updated.

14./15. The filter is set for the social network SlideShare.

16. The method *"checkUpdate()"* of the class *"SlideShareUp"* is invoked and the update of the social network information is performed.

17./18. The filter is set for Twitter.

19. The method *"checkUpdate()"* of the class *"TwitterUp"* is invoked and new information related to the Twitter's profiles are stored into the database.

20./21. The filter is set for Facebook.

22. The method *"checkUpdate()"* of the class *"FacebookUp"* is called and the social network information are updated.

5

EXPERIMENTS AND EVALUATIONS

In order to verify if the *"unified visibility panel"* and the available types of history charts provide to users useful information in relation to their social activities, the web application has been tested for about three weeks.

During this period, quantitative data concerning different users' categories, including students, professors, comedians, politicians and singers, have been collected, previously memorizing their different social profiles, into the application.

Unfortunately, for certain categories, not all the social networks considered by the application, have been taken into account during the test's phase, this due to some privacy limitations and to the fact that some social tools are not typically adopted by them.

The collection of data relating to the Facebook's profiles, for example, has been carried out only on some students, this because Facebook requires the utilization of a personal authorization token in order to access the profile information through the API, that is released only after the user's authorization.

Also for SlideShare, due to the nature of the contents shared (generally only slide's presentation), have been identified only some social profiles related to the professors' category.

Follow some example of temporal graphs, that show the main aspects highlighted during the period observed, with some considerations made on them.

5.1 Politics and social networks

The figure 32 provides an example of *"unified visibility panel"* related to a politic. In particular, it visualizes information about all the quantitative data collected concerning the social profiles of the U.S. President, Barack Obama, crawled by the application.

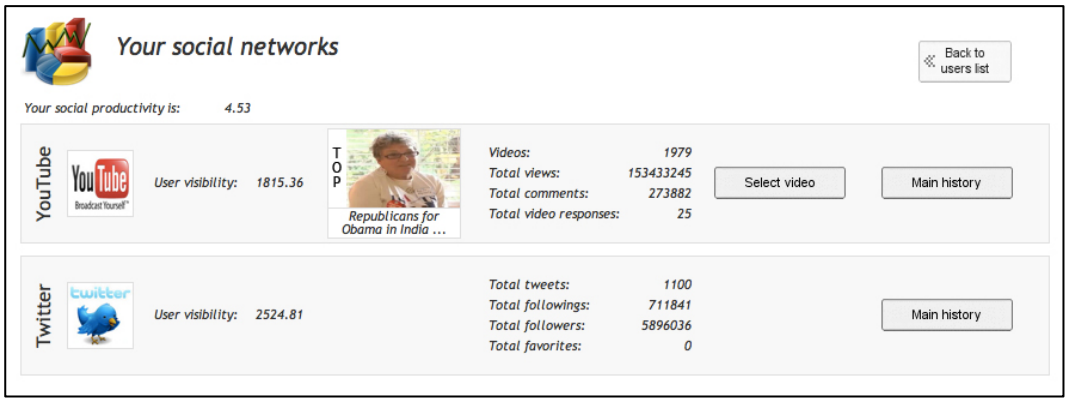

Figure 32 - "*Unified visibility panel*" containing the information of the profiles of Barack Obama, crowled during the test phase by the application

The high value of the *"social productivity"* indicator, suggest that, the president (or in any case their collaborators) uses much the social networks in order to stay connect and keep informed their electors on his political activity, both, using videos, shared on YouTube, and posting short messages, through Twitter.

As a confirmation of his elevated social activity over Internet, there are the *"visibility"* indicators that, for both, YouTube and Twitter, present values very high.

The figures 33 and 34, represent temporal charts rendered by the application, taking into consideration the main content's types, on which, the two social networks utilized by the user, are focused.

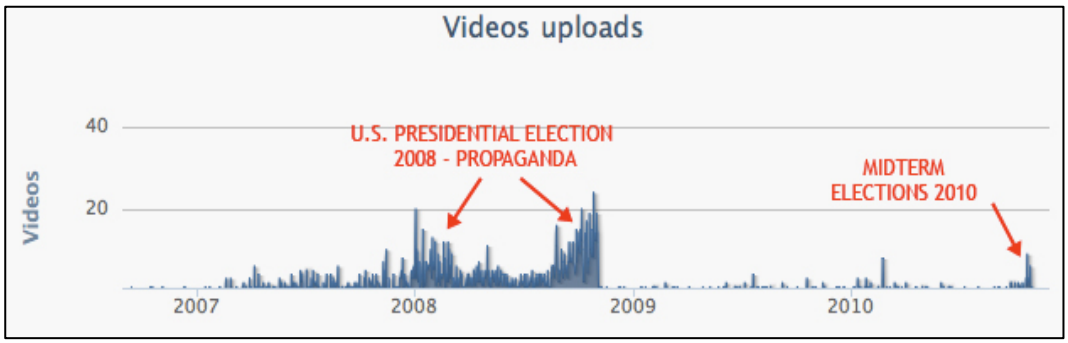

Figure 33 - Complete history about the video's upload performed on YouTube through the *"barackobamadotcom"* profile

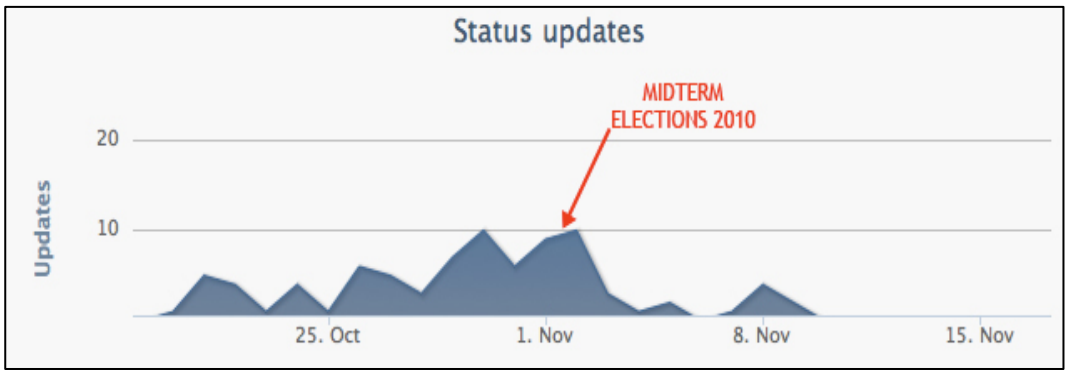

Figure 34 - Daily tweets performed through the Twitter's profile of Barack Obama during the observed period

In particular, the first chart shows the complete history about the video's uploads performed through the profile "barackobamadotcom", over YouTube, starting from its creation date, thanks to the availability of the temporal information concerning the upload's date of each video; the second, represents the history of the tweets performed, through the twitter's profile of Barack Obama, starting from the date of registration to the application developed, this, due to the fact that, the Twitter's API rate-limit doesn't provide the possibility to get detailed information. Both charts, despite the time span considered is different, suggest that social networks are used widely for propaganda. In fact, observing the figure 33, it's possible to notice how, the United States Presidential Elections of the 2008, has been conduced also through social networks and how, the same methods has been adopted, even if in lesser extent, for the recent Midterm Elections.

Evidence on the use of social networks, as effective means for the politics, may be obtained observing the graphs of figure 35, representing the number of "views" and "comments" received by videos posted immediately after the last U.S. Primary Elections (November, 2010).

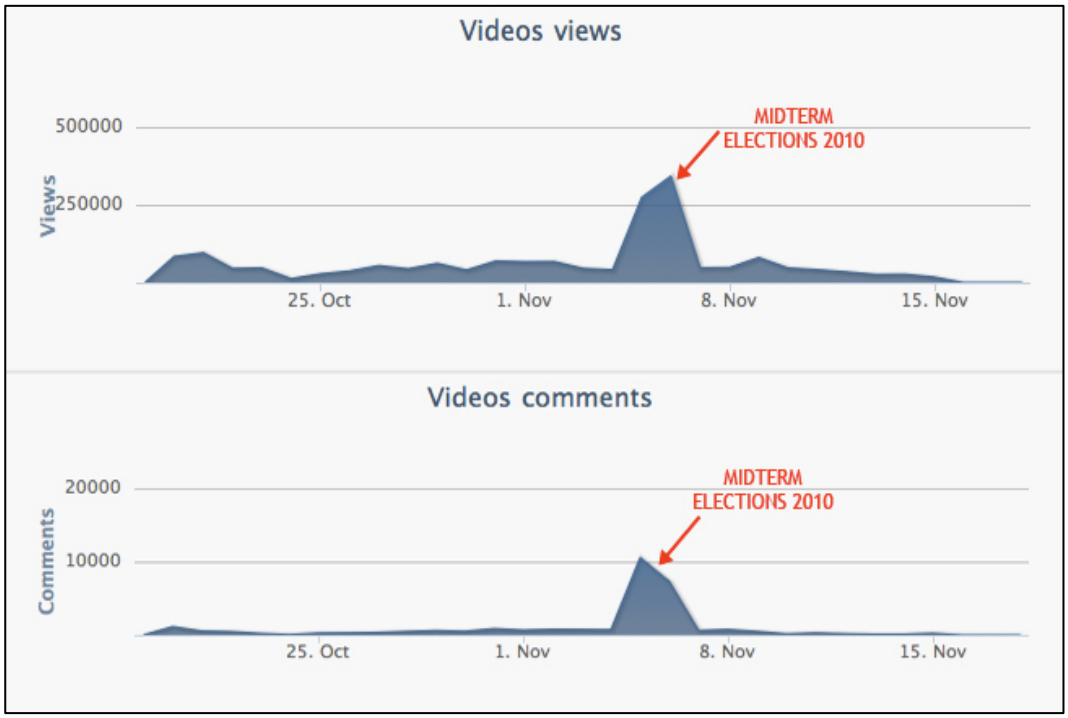

Figure 35 - Variation of views and comments about the *"barackobamadotcom"* profile, observed during the data collection

Both temporal diagrams, suffer a notable increase in proximity of the subsequent days of the Primary, suggesting that, the videos uploaded concerning the Primary Elections results have been, not only observed by many users than normal, but also object of numerous comments, indicating that a social network can be a good instrument to try to reduce distances between voters and elected, accepting suggestions and constructive criticisms.

Always observing graphs obtained through the application, by crowling some Italian political figures, it has been noted that, even though with less enphasis, the use of social networks to diffuse political opinions through Internet, in order to receive also some criticism and suggestions by voters, is spreading also in Italy, through the creations of profiles belonging to politicians or political parties.

5.2 Professors on SlideShare

The figure 36 visualizes the history charts related to the uploads of presentations on SlideShare performed by three university's professors taken into consideration during the test's phase.

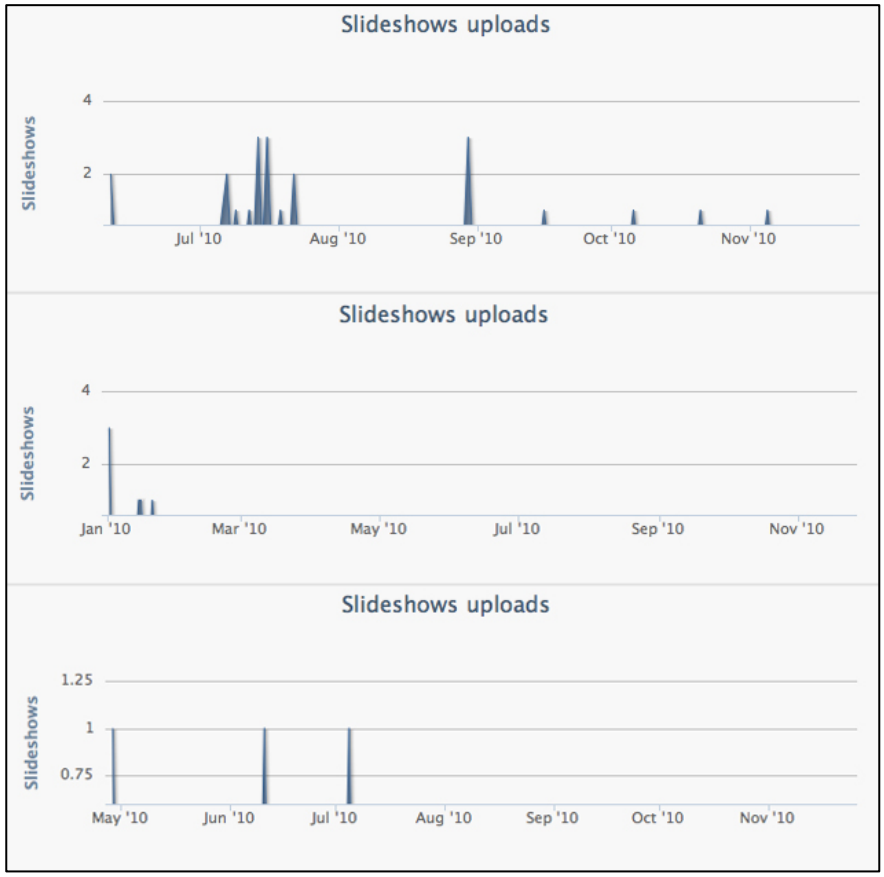

Figure 36 – Temporal charts that visualize the history of the presentations' uploads on SlideShare done by three professors.

The limited number of uploads suggests that professors doesn't like to share presentations related to the lessons topics through SlideShare but, if necessary, they prefer to use others traditional methods, like for example share them through the universities' websites.

Despite the limited number of uploads and the wide period that could be between them, the daily views charts have highlighted that all the profiles' presentations,

independently by the last upload, received a good number of daily views (considering the relationship uploads/views) (see fig. 37).

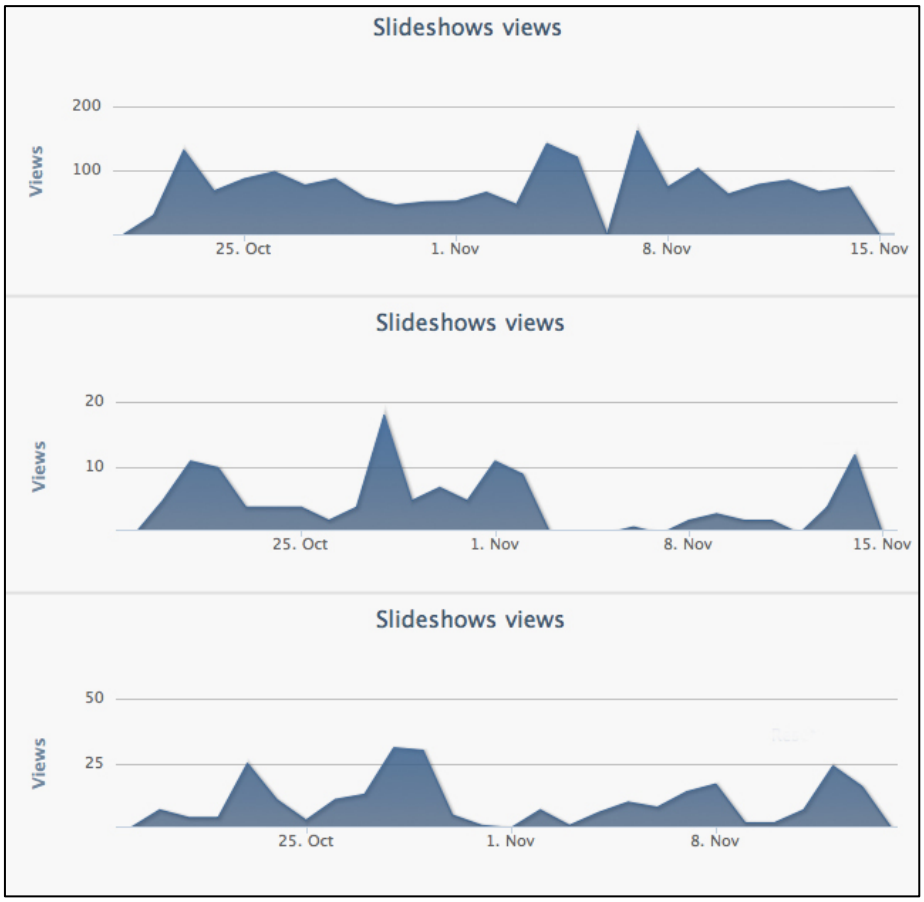

Figure 37 – Temporal charts that visualize the presentations' views of the three professors

The aspect highlighted suggests that professors tend to share presentations that contain information related to innovative aspects, that also after a certain period are still valid and useful for others that utilize the social network.

The consideration presented above, has had an objective verification through the visualization of the some detailed views charts (see fig. 38), that have highlighted that many presentations uploaded are related to meetings about innovations (ex. on Social BPM).

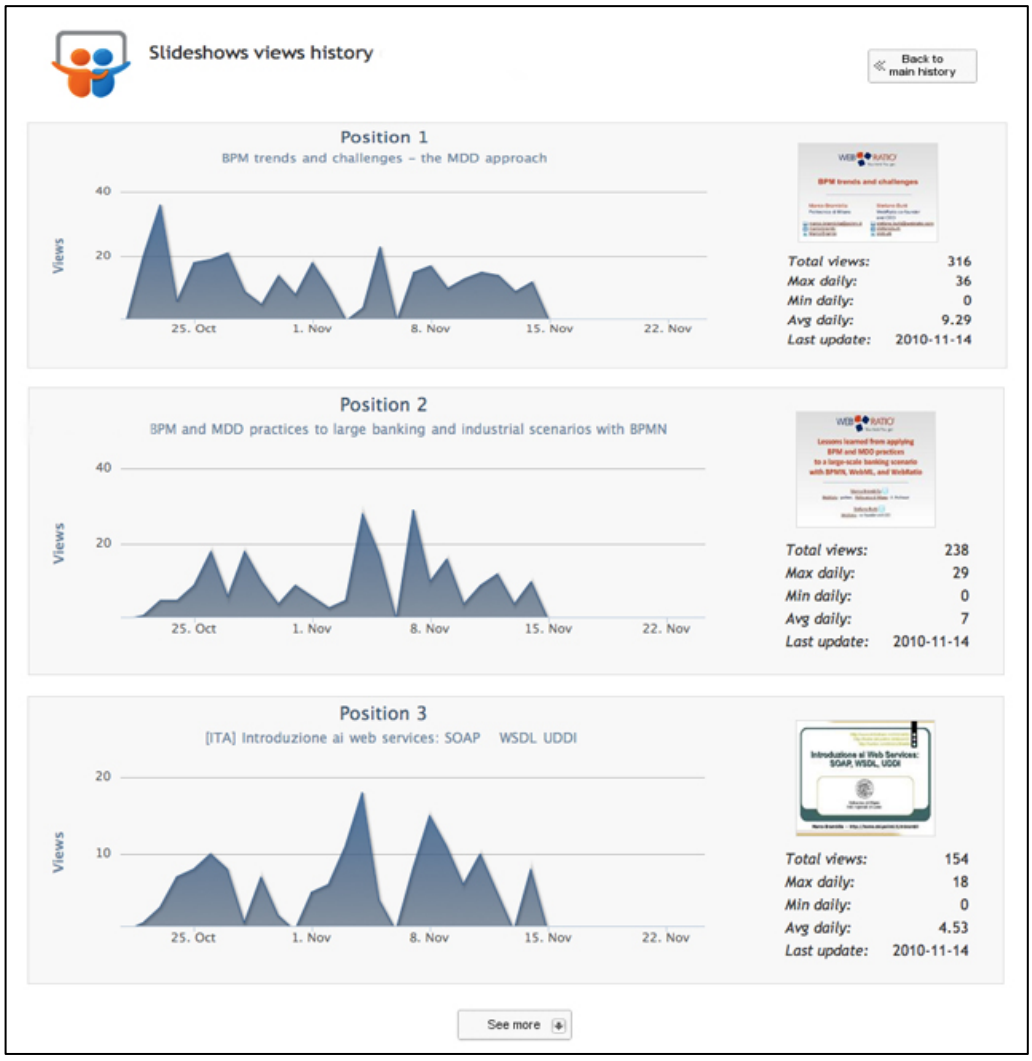

Figure 38 – Temporal charts that visualize the detailed views related to a social profile

5.3 Connection between profiles

Nowadays, the various types of social networks and their continuous expand, as well as the release of APIs that enable the applications to interface with them, have contributed to spread the possibility to link social profiles, created over different social networks, so that they can distribute updates made on one, to all the others connected.

During the analysis period, some profiles, related to the same person, have highlighted values of visibility almost identical (considering the default weights of the application).
In particular, this peculiarity has been highlighted with some profiles of Facebook and Twitter, social networks focusing on individuals, that among their features, allow users to post on the profile's home, public messages.

The figure 39, shows the *"unified visibility panel"* of a user, that during the crawling of the data, has highlighted the characteristic explained above.

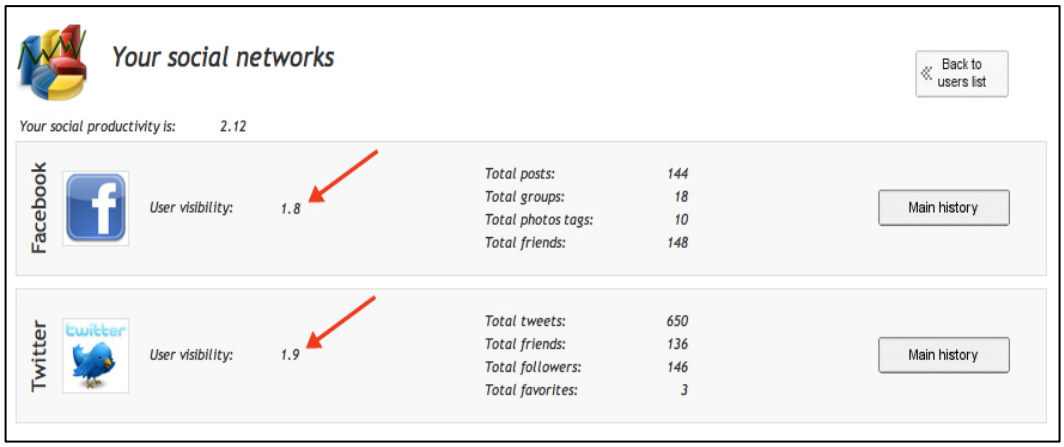

Figure 39 - *"Unified visibility panel"* that shows information collected about the social profiles of a student

Observing the two indicators of visibility, a user can notice that the values are more or less identical. This observation may not be considered to be reliable in order to establish whether the two profiles are actually interconnected.

In fact, each indicator, is influenced by different quantitative data collected and properly weighed, which differ from a social network to a social network, on the basis of the information made available by its API.

In order to know, if the two user profiles are really interconnected, is necessary to observe some temporal charts related to their "main history".

In particular, considering Twitter and Facebook, the comparison must be done visualizing, for the first, the history chart of the tweets (“status updates”), while for the second, the chart of the status updates performed by the user on his wall ("updates of status from wall").

The figure 40, shows the charts necessary to make the comparisons, related to the user of which, the *“unified visibility panel”* is presented above (fig. 39).

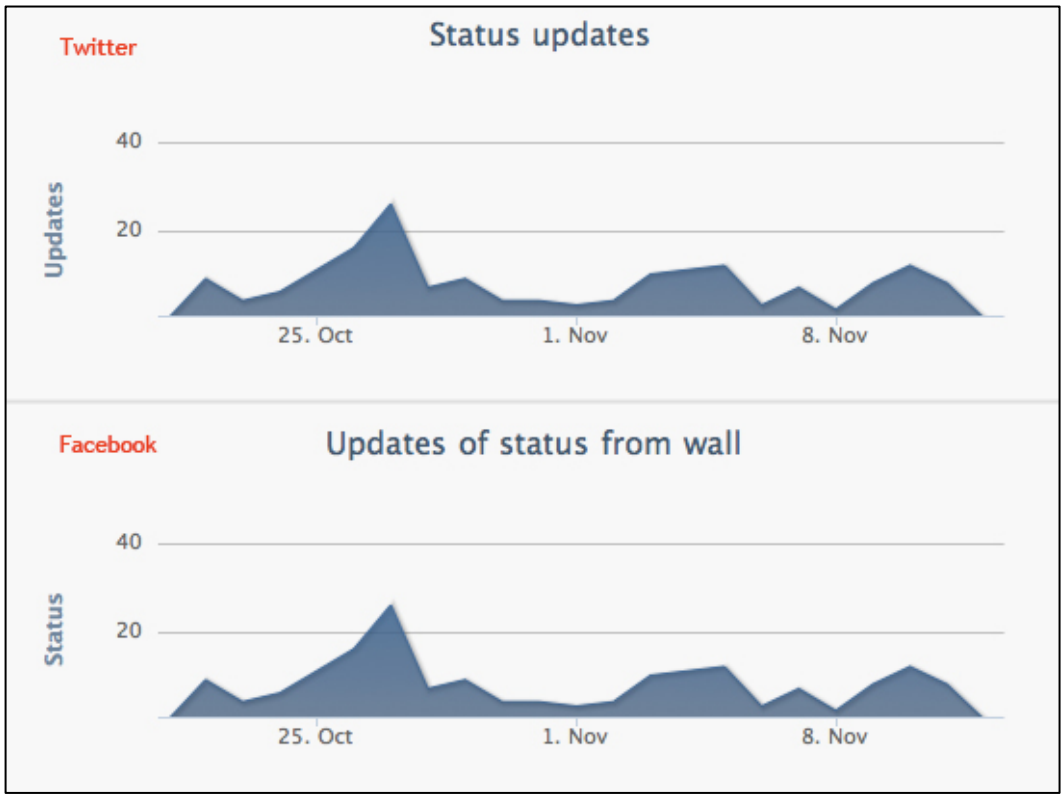

Figure 40 - Posts done with the Twitter's and Facebook's profiles by the same user during the observed period.

The evenness of the two diagrams rendered, concerning the period of the data collection (3 weeks), suggest that there is a very high probability that the user has

interconnected the two social networks, even if, in order to be more certain, a longer period, should be considered.

Observing the only temporal graphs, there is no possibility to understand if connections between social networks are bidirectional or not; it may happen for example that a user use only a single social network and from this, he spread toward all his others social profiles the updates performed, as well as, that he utilizes different social networks, connected all together, and all enabled to propagate updates toward the others.

The highlighted connection into the example, has concerned social networks that focus their attention on individuals; nothing prevents that this type of comparison can be repeated also for evaluate the presence of mixed connections (ex. social network focusing on content toward that focusing on individuals and viceversa) or between more than two networks. In both cases, the verification based only on the observation of the history charts, is more difficult.

6

CONCLUSIONS

The main objectives of this thesis project have been the study of the reputation problem within social networks, the definition of analysis metrics and models, and the creation of a web application capable to render history charts related to quantitative data (like uploads, views, comments, posts, etc.), daily collected, about all the social profiles provided by users that decides to join it.

In particular, during the project implementation, different types between the most common social networks have been chosen, analysing the information made available by their APIs and defining a specific data structure capable to memorize all the daily variations of the social values observed.

Into the web application realized, they have been created two sections, the first one having functionality of *"contents aggregator"*, that visualize real-time updates of the social profiles related to the user selected and, the second one, having the main objective to summarize, through the *"unified visibility panel"*, the online social activity of the user, providing a prospectus for each social network joined by him, through an overview of the quantitative data collected, as well as, indicators of social *"productivity"* and *"visibility"*. Influencing variables, for both indicators, have been chosen conducing an accurate analysis taking into consideration, not only the social information daily collected through the scheduler, but also the type of content on which the social network focus.

By introducing the concept of social network, they have been presented the main characteristics and provided some examples of them, taking also into considerations those used during the implementation phase.

Subsequently, the definition of social network analysis has been provided, describing which are the key points of it, and presenting some examples conduced by universities and expertise.

Finally, before starting the real implementation of the thesis project, an overview on the information made available by the APIs of the different social networks identified, has been conduced, this in order to choose four of them, with different features, able to give enough information, related to the users' profiles.

The final choice is relapsed on Twitter, Facebook, SlideShare and YouTube.

6.1 Evaluations

The implementation phase has highlighted that online social networks, in particular that focusing on individuals, have large limitations in terms of APIs' utilization employed in order to get social information about the user's profiles.
Facebook, for example, permits to collect data only after the user's authorization, through a personal token, independently by the profile's privacy restrictions. Twitter instead, if the application is not *"whitelisted"* (this mean no rate-limitation), allows to perform a limited number of queries per hour, set to 160.

Despite the limitations highlighted, the final objectives of the thesis project can be considered reached.
Analysis and choices of the data to be collected, performed taking into consideration the information provided by the different social network API's, are sufficient in order to show a prospectus, enough detailed, about the social activity of the user that decide to join the application, this, through the *"unified visibility panel"* and the possibility to render temporal graphs about their social history over different online social networks.
"Visibility" and *"productivity"* indicators, not only provide to users a direct feedback on their social activity without the necessity to view in detail temporal graphs of each social network profile, but also enable them to make comparisons with others, subscribed to the platform.

6.2 Future developments

At the current state, analysis metrics, models and data containers, have been defined on the basis of the currently information made available by the APIs provided for Twitter, Facebook, YouTube and SlideShare, social networks taken into consideration by this thesis project. Also the web application realized is able to manage social profiles belonging to the networks listed above.
It would be interesting to expand the number of the social networks analysed, taking also into consideration others types of them, like for example LinkedIn, that enable to share curriculum vitae or Flickr that focus on photo sharing.

The indicators of *"visibility"* and *"productivity"* may be reviewed, substituting numerical values with graphical bars, memorizing during the users' registration phase, information related to their job profession (student, professor, actor, politic, comedian, etc.), in order to be able to perform a normalization of the value to be visualized, considering only users belonging to the same category, and avoiding sudden changes (values too high or too low), due to possible inhomogeneity between subscripted.

Another aspect that could be improved, refers to the visualization of the data collected, introducing new graph's types or providing the possibility to users with more social profiles, to visualize similar voices values, referring to different social networks (such as comments for YouTube and SlideShare), on the same chart, this in order to make comparisons easily.

Finally, considering the registration phase, it would be interesting to introduce a methodology capable to suggest to new users, their possible profiles over the social networks considered by the application, taking into account information just provided by them (like name, surname, location, etc.), as well as, the

possibilities to limit the visibility of the updated and statistical information to itself.

BIBLIOGRAPHY

[1] Wikipedia - WEB 2.0 - http://en.wikipedia.org/wiki/Web_2.0, 09/2010

[2] Harokd Jarche's Blog - Introducing to Social Networking - http://www.jarche.com/2010/06/introduction-to-social-networking/, 06/2010

[3] CERADO - Social Networking for business and associations - http://www.cerado.com/download/Cerado-Haystack-Executive-Briefing-Social-Networking-for-Businesses-and-Associations.pdf, 2010

[4] Wikipedia - Social Network Analysis - http://en.wikipedia.org/wiki/Social_network, 09/2010

[5] Boyd & Ellison - Social Network Sites: definition, history, and scholarship, 2007

[6] Shaun Connolly's Blog - 7 Key attributes of Social Web Applications, 05/2008

[7] Facebook – Facebook definition - http://www.facebook.com, 10/2010

[8] Wikipedia - Facebook - http://en.wikipedia.org/wiki/Facebook, 10/2010

[9] Wikipedia - Twitter - http://en.wikipedia.org/wiki/Twitter, 10/2010

[10] Wikipedia - Youtube - http://en.wikipedia.org/wiki/YouTube, 10/2010

[11] SlideShare - What is SlideShare? - http://slideshare.zendesk.com/entries/55597-what-is-slideshare, 10/2010

[12] Horus - Social BPM definition - http://www.horus.biz/en/horus-iinsidei/performance-range/horus-endeavor/social-bpm.html, 11/2010

[13] MegaStorm Blog - The future of Social BPM - http://www.becauseprocessmatters.com/the-future-of-social-bpm/, 07/2010

[14] Clay Richardson's Blog - Is Social BPM a methodology, a technology, or just a lot of hype? - http://blogs.forrester.com/clay_richardson/10-05-20-social_bpm_methodology_technology_or_just_lot_hype, 05/2010

[15] Enterprise 2.0 Conference's blog - What is Enterprise 2.0? - http://www.e2conf.com/, 06/2009

[16] About.com's blog - What is Enterprise 2.0? - http://webtrends.about.com/od/office20/a/enterprise-20.htm, 10/2010

[17] Alan Mislove, Marcon, Gummadi - Measurements and analysis of online Social Networks, 10/2007

[18] Leskovec, Huttenlocher, Kleinberg - Predicting positive and negative links in Online Social Networks, 04/2010

[19] Xiang, Neville, Rogati - Modeling relationship strength in online Social Network, 4/2010

[20] MySQL – MySQL: open source database - http://www.mysql.com/, 08/2010

[21] JDOM - Accessing, manipulating, and outputting XML - http://www.jdom.org/, 09/2010

[22] Log4J – Logging service - http://logging.apache.org/log4j/1.2/, 09/2010

[23] Quartz – Enterprice job scheduler - http://www.quartz-scheduler.org/, 09/2010

[24] YouTube API – Java Google Data API - http://code.google.com/intl/it-IT/apis/gdata/articles/java_client_lib.html#dependencies, 10/2010

[25] Twitter API - Twitter4J API - http://twitter4j.org/en/index.jsp, 10/2010

[26] Facebook – Facebook RestFB API - http://restfb.com, 10/2010

[27] SlideShare – SlideShare API - http://code.google.com/p/jslideshare/, 10/2010

[28] W3C - What is XML? -http://www.w3schools.com/xml/xml_whatis.asp, 10/2010

[29] Wikipedia - MySQL - http://en.wikipedia.org/wiki/MySQL, 10/2010

[30] FACEBOOK – Facebook Query Language - http://developers.facebook.com/docs/reference/fql/, 10/2010

[31] HighChart – Interactive JavaScript charts for your website - http://www.highcharts.com/, 09/2010