

CLOVER: MOBILE STUDENT COMMUNITY ON SQUACE

Gayane AZIZYAN

The Royal Institute of Technology, Stockholm

gaya.nuno@gmail.com

Aron HENRIKSSON

The Royal Institute of Technology, Stockholm

aron.henriksson@gmail.com

Miganoush Katrin MAGARIAN KENARAKI

The Royal Institute of Technology, Stockholm

megarlen@gmail.com

Andrei NECULAU

The Royal Institute of Technology, Stockholm

andrei.neculau@gmail.com

Abstract

Azizyan, Gayane. Henriksson, Aron. Magarian Kenaraki, Miganoush Katrin, Neculau, Andrei. 2009. Mobile Student Community on Squace. Pages 12

Keywords: project report, mobile, bookmarking, student community, integration

1. Business Case

1.1. Background

Online communities have become increasingly ubiquitous over the past few years. People are able to stay in touch with childhood friends and new acquaintances alike through a number of social networking sites, which have undeniably become epitomized by Facebook. These virtual communities have greatly affected the way people interact. As university students tend to be among those most receptive to innovations – Facebook, in fact, initially only targeted university students – this phenomenon should be extendable to a more academic setting. It is only slowly beginning to catch on with the use of forums and wikis; however, primarily for the purpose of teacher-student communication. Student-student communication is typically eschewed on the universities' official channels, effectively hampering the full potential of student communities from being realized.

A problem that has arisen over the years is that inadequate systems providing information to students have been supplemented by new systems, which not only tend to provide added functionality but also partially replicate that which already exists. This overlap is of course redundant and confusing; however, the real cause of frustration among students is the result of having to stay up-to-date with multiple websites rather than just one.

Furthermore, universities have generally been apathetic in keeping up with the seemingly inexorable migration to ever more portable devices. People require constant access to applications and information. This has also been the case with virtual communities, to which people now have access through their mobile phones. However, not in academia, and as a result, students suffer: without access to a computer, they become largely paralyzed.

1.2. Opportunity Description

It can thus be stipulated that integrating information from the plethora of academic systems and building up a student community to which mobile access is provided can be of great value for students and universities alike. Universities will be able to disseminate important information to students more rapidly; for students, it will enhance their sense of community and enable them to stay up-to-date through a single interface on their most loyal companion: the mobile phone.

Apart from social networking and communication needs, students require ready access to information regarding courses, class schedules, shared documentation, discussions with professors, etc. If all such resources were available from one web community, Squace (described in 2.4) can be employed to provide a way for students to access certain information from their mobile phones in a centralized and convenient manner.

Students at the Royal Institute of Technology (KTH) are amongst those who suffer from the above-described problem. Students at the DSV department need to keep track of several websites and applications, e.g., the KTH website for course descriptions and grades; Daisy for course schedules at DSV, exam information and booking student rooms; and FirstClass conferences for access to course documentation and discussions. To make matters worse, some course leaders use Moodle course pages, or even independent course websites, instead of FirstClass. Also, courses not belonging to DSV make use of a different system – PingPong. The resulting lack of standardization makes it necessary for students to keep up to speed on all relevant sites, and potentially even create their own customized calendar to incorporate all classes. This situation is not sustainable. Thus, by covering the basic needs of students and simultaneously providing innovative mobile functionality through Squace, a web community can provide the core structure that may later be extended as well as customized for the use of other universities.

1.3. Aim

The aim of this project is to create a web community and add value for Squace and university students.

1.4. Goal

To fulfill the described aims our project's goal is to make a web student community for KTH students which uses Squace.

The main goal can be thought of as consisting of two main parts – integrating data from existing systems and making the data available through a web and mobile interface. While seemingly standalone, these two parts are interconnected and strongly support each other. Without integrating data from existing systems, the created web/mobile community would not be readily usable by real students on real student data. Also, the integrated data would not have much use, if not provided with an interactive, usable interface. For these reasons these two parts have been merged into one main goal.

1.5. Sub-Goals

The main goal can be divided into sub-goals, as follows:

SUB-GOAL1: Conduct pre-development tasks.

- 1) Research – literature, Squace documentation, and development environment review.
- 2) User survey – prepare and send survey, survey result review.

SUB-GOAL2: Integrate students' study information from KTH, FirstClass and Daisy to a single web source.

- 1) Get authentication from students to access their KTH and Daisy accounts – implementation of authentication and registration systems for web application.
- 2) Read information from existing systems and save it in application database – data extraction from event management systems, course management systems, people management systems.

SUB-GOAL3: Add core functionality to new web application.

- 1) Students should be able to view their class schedules in one integrated calendar – data extraction of event management systems.

2) Students should be able to add and edit new schedules to their calendars – data presentation for event management system. It is important to mention that students are not able to edit information integrated from Daisy/FirstClass.

3) Students should be able to check their messages from different FC forums in one integrated mailbox – data integration/presentation of forum systems.

3) Students should be able to search for course information and other students – data presentation for course management system and people management system.

4) Students should be able to make comments and post ratings on the web community – implementation of commenting system.

SUB-GOAL4: Find efficient ways of communication between the web community and Squace.

1) Students should be able to view their schedules from their mobiles – integrate event management system with Squace.

2) Students get notifications on their mobiles when they get new emails on their DSV or KTH accounts – integrate forum management system with Squace.

3) Students should be able to search other students' information through their mobile phones - integrate people management system with Squace.

4) Students should be able read and post comments through their mobiles – implementation of commenting system with Squace.

SUB-GOAL5: Conduct post-development tasks.

1) Test End product – test web application, test mobile application.

2) Refine development.

3) Produce report.

Note that the order and numbering of the steps in the sub-goals corresponds to their prioritization and order of implementation.

1.6. Method

The steps involved in the development of the system can be divided into two parts: (1) initial research and preparation and (2) actual development, including deployment and testing.

The methodology for the first part includes an initial literature review about web and mobile communication. A careful review of the functionality offered by Squace is also going to be carried out, including an analysis of the underlying technical aspects which will be used during actual development. Further, existing systems will be analyzed, including the functionality they provide and the underlying data and data structures, used in later design of the system database. In order to get an initial assessment of usage and acceptance by the student body, a user survey is going to be carried out in the first stages of the project.

The second part – the actual development of the system – will be carried out using certain Agile techniques. The development team, being truly cross-functional with changing roles, will include pair programming between team members at different stages of development and review sessions after the completion of each task. Teamwork organization will include extensive use of collaboration tools, including file sharing and system version support. Web mining and data extraction are going to be used during existing system integration.

All steps of project planning and development are going to be carefully documented, resulting in a final report of the project and the system functionality.

1.7. Disposition

To achieve the goals enumerated above, it is vital to build upon a solid theoretical foundation. Chapter 2 thus provides an analysis of existing research that is pertinent to the cause of the current paper. In Chapter 3, the practical work that was carried out is presented. This is followed by an analysis of the work and a discussion of resulting findings, in Chapter 4. Finally, in Chapter 5, conclusions are drawn and recommendations made.

2. Theoretical Foundation

2.1. Student communities leveraging IT

That which most resembles some form of IT-leveraged community in an academic setting are threaded discussions, i.e. forums. They have proved useful in allowing the continuation of discussions even after and outside of class – and without the supervision of the teacher. As a result, students are more likely to learn from each other. However, despite the asynchronous benefits of forums, their effectiveness are still being hampered by exclusive access via computers and the “passive nature” of the medium. This entails that students have to check for updates; when activity is low, students will check less and less frequently, sparking a vicious cycle that typically ends up in long periods of persistently low activity. Mobile phones have a big potential in countering this dilemma by increasing the sense of connectivity and possibly even allowing students to respond away from the computer (Hill, 2003).

Another reason for advocating close-knit student communities is that they foster learning. It has been suggested that institutionalized education only becomes valuable once the individual student becomes part of a social group. Similarly, it has been shown that there exists a correlation between the sense of community and cognitive learning. However, the closeness of a community is naturally difficult to measure. One approach has been to study forum contributions of students, and the positive correlation between forum posts and the student’s sense of community was confirmed. On the flip side, frequent system posts thwart the endeavor to induce a sense of community amongst students. Computer-aided communication (CMC) has nevertheless proved its pedagogical benefits, in particular by promoting close-knit communities through various collaboration tools (Dawson, 2006).

Online communities that are available on mobile phones have led to the establishment of the term Mobile Virtual Communities (MVCs). However, MVCs have not become as popular as non-mobile online communities for a number of reasons, such as limited content and the fact that MVCs have failed to adapt sufficiently to the distinctive ways in which mobile phones are used. Research has been conducted on the use of MVCs in an academic setting, where they proved helpful in enhancing the educational experience, in particular by making information available to students on the move. It was, however, limited and did not provide collaboration and notification. This is an area in which MVCs have a significant potential to improve (El Morr & Kawash, 2007).

2.2. Technical considerations

With "an estimated two billion subscribers worldwide" (Jin & Lee, 2008, p. 113), mobile technology is ubiquitous in many corners of the world. Availability and penetration of this technology is a strong incentive to develop for the mobile users yet technology considerations must be asserted. That applies especially when the goal is to facilitate social connections and when some argue, "that mobile phones may lead to atomization and privatization among users" (Jin & Lee, 2008, p. 113). Having had to deal with changes in deadlines and in the overall project plan, interaction features have been delayed beyond the scope of the current project.

Several non-social technology considerations have had to be parsed though. "There are multiple markup languages" (Heins & Steiner, 2001, p. 237) for low bandwidth connections and along with "the limited display area and input methods" (Heins & Steiner, 2001, p. 237) this introduces unique obstacles. Given the mobile browser that our product is targeting, HTML and SquaceML are to be addressed when talking

about markup languages. The Squace application takes both as input, while our desktop browser module produces HTML-only content.

Considerations regarding the available bandwidth, and indirectly the size of the page content, are doubled by the proxy interface of the Squace application. Each page requested by a Squace user goes through Squace servers where content is checked for errors and processed. This slows down the content access even further.

The success of a mobile web application "lies on the value and features they offer to the users/clients" (Murugesan & Venkatakrishnan). In order to develop a compelling product, one needs to consider the number of clicks necessary to access needed information and to minimize them. Overall "web pages should be specifically tailored to suit hand-held devices. Besides the problem of the presentation, information content delivered to the mobile devices" (Murugesan & Venkatakrishnan) needs to be adapted.

2.3. Mobile considerations

Existing academic communities have only web interface. Accessing these web applications in mobile phone browser has high load of internet traffic, and the resulting web pages may be depend on mobile phone specific OS and technical details. To cover this gap integrated student community is needed to be develop for mobile browsers, so that student can access their academic information fast and simple through their mobile phones.

2.4. Introduction to Squace

Squace is a Universal Mobile Interface offers an encompassing solution for structuring, accessing and sharing information over the mobile Internet. It does this using Extensible Markup Language (XML) over HyperText Transfer Protocol (HTTP) (Squace AB 2009).

Squace client is a Java application that is installed on mobile phone, Squace content server works as a proxy between client and application server. It get the request from the client and retrieves information from application server, then it translates Squace XML pages by adapting images and content structure for a specific mobile phone, Squace also uses Caching system to cache the application server static pages to reduce internet traffic.

By using Squace as mobile interface for web application it is simple and fast to develop mobile applications without caring about mobile vendor and operating system specific details such as technical inefficiencies of mobile application model and deployment and support issues.

3. Practical Solution

3.1. User survey

Before implementing a system one must always take into consideration the needs and preferences of the end users. Therefore, a user survey was sent out to KTH students currently active in the Engineering and Management of Information Systems Master program. Out of the 59 respondents, 57 wished to see a unified interface to the multitude of academic systems currently in operation. An overwhelming majority of well over 80% also wanted to have mobile access to relevant academic information and expressed a desire to interact informally on that information. The only question prompting some form of deliberation was to what extent interactive features should be provided in consideration of the inherit limitations of mobile phones. With the exception of 2 respondents, there was a consensus that dynamic information would be desirable in addition to that of the static variety. Interesting to note, however, is the fact that approximately 2/3 of the respondents believed that "advanced" interaction – free commenting in addition

to predefined responses – or “full” interaction – integration with current posting features, such as the ability to respond directly on forums – should ideally be provided.

3.2. Data extraction and integration

A simple and attractive interface is nothing without valuable data. As easy as data it is to produce, making it valuable is not. Our business case research leads us to a solution of data extraction from current systems, and integrating it in an uniform format.

Our project is a project that did not fight for informational support. The IT office of the DSV department was not informed of the development of this software. Although a risky decision, the analysis showed that it is not an impossible mission and that the more general applicable, the better. This leads to a software product that can be used in several campuses, even without the involvement and support of the campus officials.

The IT systems made available to students at the DSV department, have a special feature that makes external access, and thus data extraction, a process that requires little innovation. They all have a web interface. This feature allowed our product to simulate the browsing process and perform web-mining routine.

Developed on top of the PHP platform, and using MySQL as a database backend, the extraction module is developed on an abstract level assuming different types of data sources, but implementing only one: a web source. Every IT systems has associated an extraction plug-in that defines how impersonating a student’s account and authentication takes place, that defines how courses, events and messages are extracted, based on each system’s capabilities and features.

Although data extraction proved to be fairly easy to implement through Document Object Model (DOM) parsing and Regular Expressions, data integration is much more of a sensitive issue. The goal of our extraction module is to provide valuable data, yet valuable data means complete and accurate data. The reality shows that each system provides partial data, not to mention in many different formats. Extracting partial data is possible, but prioritizing between data sources’ fields is not. That is why extracted data is taken through SQL views, in order to filter and prioritize information and output clean data.

3.3. Software development process

The layers of Clover mobile and web applications can be represented by the following diagram:

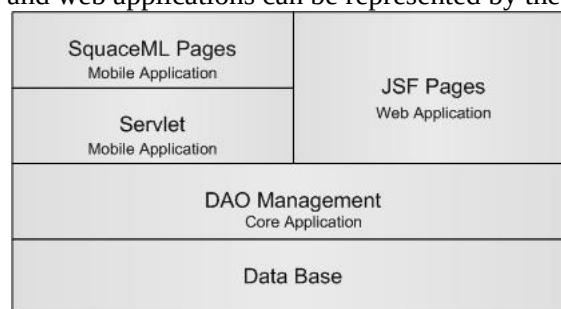


Figure 3-1. Application Layers

Software consists of three layers: Core application with Database Access Objects (DAO), Web application using JavaServer Faces (JSF) and Mobile application. Core application is used to connect to database using JDBC and Tomcat connection pooling. Web and mobile application use the same core functions to interact with the user.

Mobile application uses Servlet technology to get the request from Squace client and generate the response in XML pages with SquaceML tags and user specific information retrieved from core part of application (Figure 3.2).

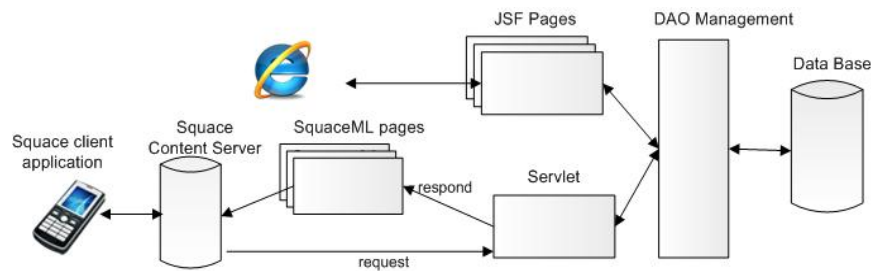


Figure 3-2 *Software Design Components*

3.4. Student community on the web

Clover web is developed using JavaServer Pages (JSP), JavaServer Faces (JSF) and JavaBeans technologies. On the front end, Sun Jsf RI is used, as well as the OpenFaces component library for components such as tabbed pane, popup layer, etc, and the RichFaces library for Ajax functionality. Tomcat 6.0 is used as a servlet container (both for local implementation and actual hosting), and the MySQL database is accessed via Java DataBase Connector (JDBC). OSCache is used for session-caching content retrieved from the database. Given that Clover is in a prototype stage, only very basic validation methods have been implemented as of now (e.g. login information cannot be empty).

The development environment tools used included Eclipse Interface Development Environment (IDE) and Tortoise SVN for code subversioning. One of the main considerations during development was to use the best available open-source libraries.

The web User Interface (UI) consists of a login page, a registration page, and a tabbed main page. The main page includes a home tab which, at this time, contains a folding panel which displays today's events for the logged in user and a Download Sqaunce button. Other tabs include calendar, information, courses, people, and messages. The calendar displays fixed and custom events for the user – the user can see his/her course schedule in popup windows as well as create new events. Messages tab displays a table with the user's messages, appropriately marked read or unread depending on their state in the system they were imported from. The courses tab contains a paginated table with a list of all courses. The table is loaded dynamically and can be searched/filtered by every column field, and external links are provided to the course pages. The people tab contains a table of people, though at this time the table will be empty because data extraction failed to provide a list of people in the database. Finally, the information tab contains links to certain useful information (such as printer locations on campus) which open in new popup windows.

3.5. Student community on the mobile

Students are able to check their calendar which includes course event and custom events added by student from web interface. More over students can read their messages from forums that they are registered in the web application in general mailbox and not read messages mailbox.



Figure 3-3. *Clover mobile application screen shots.*

In the mobile presentation, testing Squace pages on an emulator from localhost server was an issue because it is not possible to test SquaceML pages from localhost.

4. Analysis

4.1. Options available and considered

There are many potential approaches one can assume when attempting to build a mobile student community. A common and sensible starting point is to try to meet an existing need, resolve or mitigate a problem, or bridge a gap. The unfavorable situation that was addressed and – in a way – exploited in this project was the plethora of academic systems and the lack of access on mobile phones to the content contained in these systems. Even with a business case in place, it is still necessary to weigh the alternative ways in which to carry out the project.

One way of doing this could have been to develop a brand new academic system from scratch, and try to establish a more involved student community around it. This would then have been made available on mobile phones through a mobile website or a separate mobile application. This would have necessitated obtaining the status of an official tool by a university, and thus be robust enough to replace the numerous existing systems already in place today. The feasibility of this was highly questionable. Instead, it was decided that a better short- to medium-term solution would be to bridge the existing gap by merely integrating the information from the different systems into a single interface. By appealing directly to the students, who are the greatest beneficiaries, instead of attempting to persuade universities to overhaul their existing infrastructures, the likelihood of success would increase significantly.

The integrated information has thus been made available on mobile phones through Squace. This was a pragmatic solution, as it enabled the application's core to be shared by the web and mobile versions. At one point, the idea of replicating the Squace interface on the web was considered; however, it was discarded in the end in favor of exploiting the advantages of a larger screen. In order to ensure an active student community, the idea was to provide a number of interactive features, even on the mobile version – a wish that was expressed in the conducted survey. Due to the limited time frame of this project, such features are severely limited in this prototype. Custom events can, however, be added on the website. In a fully developed version, adding more interactive features would have a high priority.

4.2. Cost-benefit analysis

The benefit of this project for Squace will be to attract new Squace users and find efficient interactions between the web community and Squace. Students, on the other hand, will benefit from the new application: it will make it easier for them to check their daily schedule, search for courses, get notifications and perform other activities through their mobiles. To meet the student needs we aim to put the web community in real use for KTH students.

The costs of the system development and research are summarized in table 4.1.

BUDGET FOR: Student Community on Squace					
Expenditure code and heading		Monthly figures			Totals
		Sep 2009	Oct 2009	Nov 2009	
A	Project-specific research	6 days			6 days
B	System Development	2 days	22 days	5 days	29 days
C	System Testing			2.5 days	2.5 days
D	Documentation			7.5 days	7.5 days
Monthly totals:		8 days	22 days	15 days	45 days

Table 4.1: Budget analysis for projects

Each day includes 4 hours of development per team member for 4 team members, totaling up to 16 hours of development per day. Assuming that one hour of development costs 125 SEK, the development cost

per day in Swedish crowns amounts to 2000 SEK. The total number of days being 45, the overall costs are $45 \times 2000 \text{ SEK} = 90,000 \text{ SEK}$. Initial costs include price for web hosting, roughly 500 SEK per year. Thus the total cost is 90,500 SEK. Future costs will also include maintenance costs.

The benefits of the end-users has been adequately described above. The principal benefit of the project is associated with appealing to a big user group. Once this has been achieved, we will be in a good position to seek financial benefits. The many options include (1) selling the product to a third-party company, (2) selling the product to universities around the world, and (3) maintaining ownership and selling advertising rights.

4.3. Impact and risks

Our business case mentioned that “the successful completion of this project and subsequent launch of the application would mainly affect the way university students access academic information and interact with each other.” Given the reality that this is a partial success due to the missing interactive component, the impact is clearly diminished. Even so, given the results of the survey, there were slightly more students giving more importance to the integration and the mobile-access feature, than to the interaction feature.

The main risk that was identified before the development started was data security, primarily related to password storage. No learned computer user will easily provide credentials without security assurance. This risk cannot be eliminated, but can be diminished. One of two identified steps has been implemented: the database server cannot be accessed from the exterior, but only from the server that hosts the application. The second step to be taken is to encrypt the credentials, but only so that it is not in clear text because strong encryption algorithms will not allow the credentials to be used later for data extraction.

Our team also had to deal with other risks, primarily related to meeting deadlines, arising team conflicts and to hosting and e-collaboration issues. As a contingency measure, the project plan has been updated in order to take into account new realities and to minimize the impact on the development. Finding the right host was not an easy thing due to the combination of different development platforms, but eventually the issue has been dealt with.

Three other risks have been met by eliminating features: the extraction of people, emails and RSS articles.

5. Conclusions & Recommendations

Building up a student community by simultaneously exploiting and overcoming the existence of a prevalent problem in academia is a sensible means of attracting users. By integrating the most important information from various academic systems, and bringing convenient access to that information by making it available on mobile phones through Squace, we hope to appeal to a large group of potential users.

The assumed approach to deliver the solution was practical, as it allowed the project to be completed within a short time frame and thus keep costs reasonably low. Maintaining a database as an intermediary level facilitated development, as data extraction and presentation could to a large extent be done independently. This design allowed the project team to be divided into two sub-teams, and the development to be done in two different languages – PHP and JSP – effectively taking advantage of the different skill sets of the individual project members. For the presentation layer, Squace was a convenient solution, not only for end users, but also as a means of completing development in a short amount of time: the web and mobile versions are, in fact, using one and the same application core.

In the survey conducted amongst KTH Master students, a strong demand for interactive content on mobile phones was expressed. In a fully-fledged version of the system, it is highly recommended that such features be incorporated. This could present a way for students to interact informally on academic content. The demand for a mobile student community built around such content has more than adequately been demonstrated. The cost of developing the current system, and of a more complete, extended version, will, within a reasonable amount of time, be outweighed by the prospective benefits.

The system also presents a source for more academic research to be conducted. It could, for example, be interesting to analyze and discover different types of interactions that students would enjoy on their mobile phones in general, and on the Squace platform in particular. One could, for instance, investigate what type of information students would find most useful to have access to on their mobile phones. Emphasis could also be placed on how to carry out efficient mobile-to-web communication, for instance by using Squace. Finally, it would be interesting to build on prior research – related above in section 2.1 – and see how the emergence of a mobile student community would affect students' sense of community and learning outcome.

References

Dawson, S. (2006). Online forum discussion interactions as an indicator of student community. *Australasian Journal of Educational Technology* , 22 (4), 495-510.

El Morr, C., & Kawash, J. (2007). Mobile virtual communities research: a synthesis of current trends and a look at future perspectives. *International Journal of Web Based Communities* , 3 (4), 386-403.

Heins, J., & Steiner, D. (2001). Taking Your Information Into the Wireless World: Developing Information for Delivery to Mobile Devices. *Communication Dimensions* , 237-244.

Hill, T. R. (2003). Leveraging Mobile Technology for m-Learning: 3rd Generation Threaded Discussions. *Proceedings of the 36th Hawaii International Conference on System Sciences* (pp. 1-4). Honolulu: IEEE.

Jin, H. J., & Lee, H. C. (2008). Semantic web technology using collective intelligence of mobile social networks. *4th International Conference on Next Generation Web Services Practices* (ss. 113-116). ?: IEEE.

Murugesan, S., & Venkatakrishnan, B. A. Addressing the Challenges of Web Applications on Mobile Handheld Devices. *International Conference on Mobile Business* (s. ?). ?: IEEE.

Cadle, J., & Yeates, D. (2008). *Project management for information systems*. Harlow: Pearson Prentice Hall.

Squace AB. (2009). *Squace Homepage*. Retrieved 09 10, 2009, from Squace.com: <http://www.squace.com>