Accessing Student Information Systems Using Mobile Connected Devices

Hrvoje Maracic, Iva Bojic, Mario Kusek
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{hrvoje.maracic, iva.bojic, mario.kusek}@fer.hr

***Abstract*—In the competitive environment of international higher education-related services, the concept of students as "customers" becomes a competitive imperative. Seeing students as "customers" increases a dialogue-based relationship between universities and students having a positive impact on students' satisfaction. And students' satisfaction has a significant positive effect on their loyalty which even can promote their success. In order to increase students' satisfaction with their university, in this paper we propose a mobile application Fer Droid that can be used for data aggregation from different student information systems. Our application saves students' precious time leaving them more time for other activities (e.g. studying, training and dating). We made both subjective (using students' surveys) and objective analyses of Fer Droid and concluded that students are more satisfied and can retrieve information much quicker when using our solution than when using complex web pages on phones.**

***Index Terms*—Fer Droid, Android, data aggregation**

I. Introduction

Process of globalization does not affect only politics and economy but also higher education. Since student mobility is becoming something that is very common, there is a dramatic increase in global competition in the higher education business. The findings of study presented in [1] have shown the need for a more comprehensive, involved, and proactive strategy to developing, managing, and maintaining relationship between university and students. Thus, universities worldwide are undergoing fundamental shifts in how they operate and interact with their "customers": students, alumni, donors, faculty members, and staff members [2]. Namely, viewing students as customers provides a competitive advantage for higher education and enhances a university's ability to attract, retain and serve its customers [3], [4], [5].

In 1995 Kotler and Fox [6] stated that every organization in the world will be ineffective unless it does not focus on its "customers". In context of higher education, first and foremost is the treatment of individual students, alumni, parents, friends, and each other (internal customers). When students are satisfied with their education, they are loyal to it. Moreover, their satisfaction has a positive effect on promotion of their scholar success. Besides their satisfaction, largely accepted way to promote students' success is by implementing mechanisms that allow both students and professors to closely monitor and evaluate students' success [7].

One huge part of the monitoring and the evaluation students' success process is to keep and analyze their activity records. There are generally two ways of keeping student activity records: written and electronic. One of the advantages of keeping electronic record, compared to the traditional written records, is the ability to provide all interested parties (both students and their professors) with real-time queries of their records. Since there are many kinds of records to be kept for every particular student, they are usually located in different systems which can be, but do not have to be interconnected.

For a student or a professor this means that in order to access all of student's records, she/he has to access all of the various components of the student information system. With the advent of always connected smart phones, internet-connected electronic records gained another advantage: the ability to be accessed by the student/professor at any time (i.e. during a lecture, while travelling to/from faculty or while waiting in a lunch line). However, as evidenced by our own attempts to use our University's web applications which were designed for desktop browsers, even the most modern smart phones (e.g. Apple iPhone 4, Google Android based phones, Windows Phone 7 based phones) have issues with complex web pages and web applications (probably due to difference in their JavaScript and DOM engines [8], [9], [10]). In order to take advantage of this benefit, the information system needs to have a simple web interface capable of being rendered on mobile devices.

In this paper we propose a mobile application Fer Droid that can be used at Faculty of Electrical Engineering and Computing, University of Zagreb. Fer Droid serves students when retrieving data from different student information systems developed by different Departments of the Faculty, or even different Institutions. Most of these components have non-overlapping functionality and communicate very little with each other, especially for the purpose of aggregating the displays of their records. Our faculty has three major systems: Ferko [11] which provides student calendars, course group changes and on-line exams for some courses; AHyCo [12] which provides student scores, course attendance logs and also on-line exams for some courses and FERWeb [13], a web content management system (CMS) which provides all course-related and general news and notices, Faculty directory and maps and other general information. Moreover, there is also a fourth system used by all University students which provides overview of the available funds for the monthly food subsidy (X-Card balance [14]).

II. Fer Droid Architecture

Since at our University different student information systems are used, for a student to get her/his calendar, find out whether her/his Math paper has been graded, check whether there has been a change to the homework deadline and verify that she/he has enough subsidy left for today's lunch she/he would have to open four web pages and enter her/his credentials four times. Also, some of these systems have web pages not optimized for mobile device rendering, and some have pages which do not even work in mobile browsers. Therefore, we propose a native mobile application approach to presenting student data to users.

The major constraints of the smartphone computing environment are network bandwidth and traffic costs, interface responsiveness i.e. the time needed to present the user with the data they requested, and making efficient use of the battery power available. Most of the data sources used in the application are available only as standalone web applications, with no APIs which would enable creating a mashup or portal-style application i.e. directly interfacing with the data without the requirements for network traffic and processor intensive data transformations. Taking this limitation into account, there are two possible architectural approaches: thin or thick client.

One approach is a distributed architecture system (Figure 1), using a thin client on the mobile device itself and some form of a gateway or a proxy between the data source(s) and the client. The advantage of this approach is the ability to offload all work-intensive processing to the gateway. This would enable the shift of processing and power usage intensive transformations from a very limited mobile environment to a server with a constant power supply and more easily expandable processing power. It would also enable network traffic reduction on the smartphone by using practices such as pagination of lengthy datasets (e.g. smartcard swipe log), break down datasets into smaller subsets (e.g. separating the newsfeed into a feed containing headings and short summaries, and a feed containing separate whole stories to be served on demand) and applying compression.

Unfortunately, this approach also has some serious drawbacks. In a situation where all the data sources are controlled by a third party, such an architecture proves to be an easy target for IP-based service filtering since, when viewed from the data source side, all of the requests seem to originate from the same IP, the one belonging to the gateway. And since almost all of the data sources require authentication, an even more serious drawback is that this approach requires communicating the user credentials with the gateway. This poses a security risk in the case of an attack on the gateway and a deterrent to a number of users who feel uncomfortable sharing their credentials with the third party over which they have no control.

Due to the aforementioned reasons, Fer Droid has been implemented as a thick client (Figure 2). The user's smartphone directly communicates with each of the data sources and performs all the needed transformations itself. To partially mitigate the processing and power costs, an effort has been made to try to use more efficient tailor-made algorithms instead of readily available generic libraries. Compared to using conventional access methods (i.e. mobile web browser), significant network traffic reductions have been achieved by not loading any non-HTML content (e.g. media files).

Access to each data source has been implemented as a logical module allowing for easy addition of new data sources and reducing the impact of changes to one of the data sources on other modules (Figure 3). Our application has eight modules that fetch data from different sources:

- Calendar Module fetches data from Ferko;
- Directory Module fetches data from FERweb;
- X-Card Balance Module fetches data from a web page which provides overview of the available funds for the monthly food subsidy;
- Newsfeed Module fetches data from FERweb;
- Course Results Module fetches data from AHyCo;
- Course Attendance Module fetches data from AHyCo;
- Smartcard Swipe Log Module fetches data from AHyCo;
- Course List Module fetches data from AHyCo.

III. Fer Droid Implementation

In this section we describe the implementation of a native mobile application Fer Droid together with its developed application modules shown in Figure 3.

A. Calendar Module

The calendar module shows a student's calendar, including their classes, exams, laboratory exercises, etc. The iCal calendar file is retrieved using the HTTP protocol and then imported into Fer Droid. To access the file no authorization is required. Instead, it's URL contains a permanent, randomly generated, unique string token. When the user enters their credentials Fer

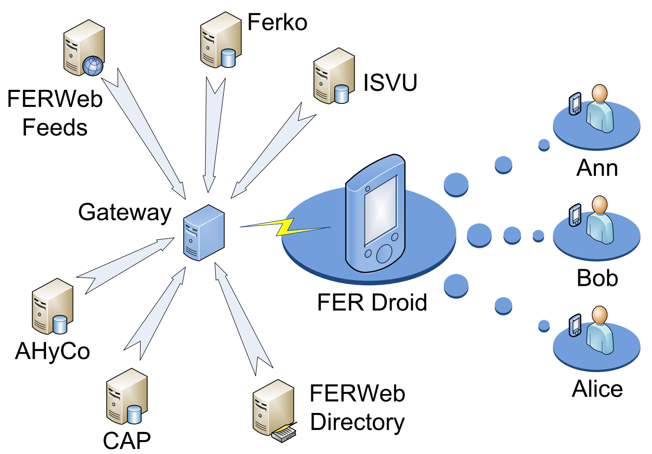

Fig. 1. Thin client architecture

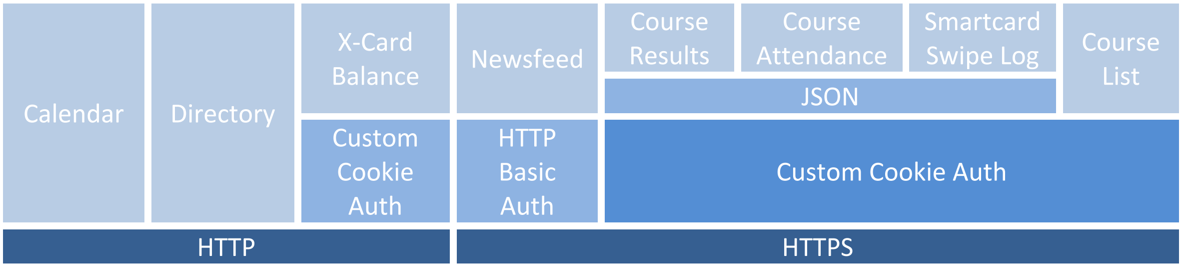

Fig. 3. Fer Droid modules

Droid tries to login to the appropriate SIS server. The link is then extracted by parsing the home page which contains the token as a part of an HTML link. It is then saved on the device for all subsequent calendar update requests.

B. Directory Module

The directory module enables users to quickly find an employee of the Faculty and all the relevant information about them including their telephone number, email address and office location. The module also contains maps of all the floors and buildings of the Faculty so every office location can be displayed highlighted on the appropriate map (Figure 4). The directory itself is downloaded over HTTP as a CSV-type file, parsed and saved to the built-in application database on the device running Fer Droid. The directory can be updated at any time, either by manually selecting the update option or by accepting the update notice which is displayed once after every application start (configurable in options).

C. X-Card Balance Module

Students in Croatia receive monthly food subsidy from the Government. This subsidy is distributed monthly to students' accounts which can be accessed by using the student identification magnet stripe card called the X-Card at selected restaurants. Students can check their balance on the receipt after using the card or by visiting a web site and entering their card number. This module enables users to check the remaining balance by sending an HTTP request that emulates the entry form on the web site, and then parses the response HTML to extract and display the amount to the user.

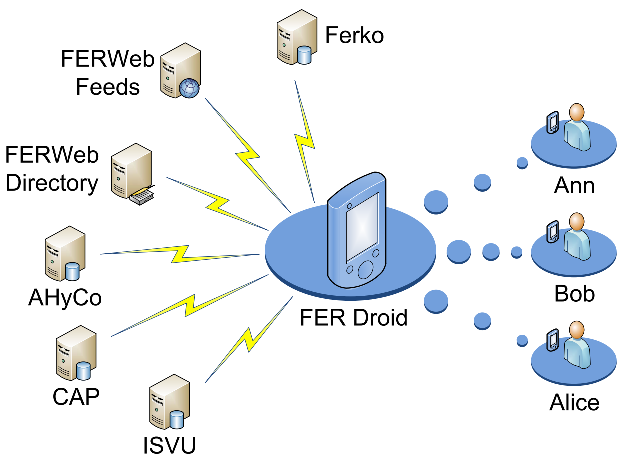

Fig. 2. Thick client architecture

D. Newsfeed Module

The Faculty provides a custom news feed with general news and news from the subscribed classes to each student. This feed is available from the student's home page on the Faculty's site as well as an RSS feed. This module retrieves the student's newsfeed as an RSS feed and displays it.

E. Course Results, Course Attendance and Smartcard Swipe Log Modules

Course Results module displays a list of all student's enrolled courses, present and past. For any selected course a detailed display shows the student their points in various categories (e.g. homework, exams, activity) and the total sum.

Course Attendance module displays whether the student's class attendance has been registered by showing a list of all classes for a course and a plus mark next to the classes for which their attendance has been registered as valid (i.e. they registered and they have not arrived too late or left too early).

Smartcard Swipe Log module is closely related to the Course Attendance module, but instead of showing how their registrations relate to classes, it only shows to students the time and location of their registration, enabling them to verify that the card has been properly detected by the reader in the class room where the lecture is.

All three of these modules retrieve their data from the same Faculty system by emulating a web application. Fer

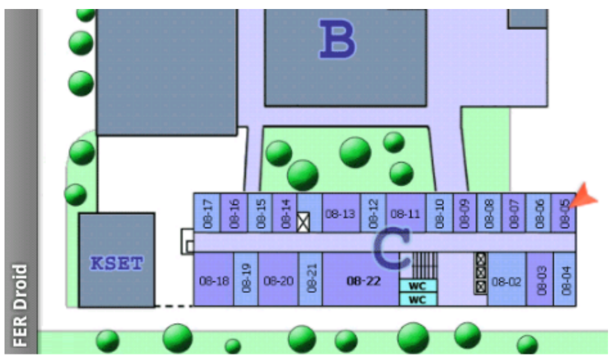

Fig. 4. Office location on a floor map

Droid supports session awareness by using cookies. After authenticating with the web application, it issues a series of HTTPS requests and reads the returned JSON data. The choices and final results are then displayed to the user in a mobile-friendly way.

F. Course List Module

Another information system provided by the University allows students to view information about their courses such as ECTS credits, final grades, lecturers and assigned lecture and exercise groups using a web application. The Course List Module (Figure 5) acquires this data by emulating the HTTP POST and GET requests corresponding to specific user actions in the web application like logging in, requesting the course list or details for a specific course. After logging in the session is maintained using a cookie.

An example of algorithm optimization applicable to this module is parsing of the response HTML pages. The course list is returned as an HTML page with the returned list in an HTML table and DIV elements. One easier method of parsing this would have been to use an HTML parsing library to create a Document Object Model and then navigate that tree. Unfortunately, after following this path, it was discovered that it takes over 8 seconds on an average mobile device for the library to perform all the required tasks. The second try was to search through the page as a string counting elements passed and keeping references of the indices of interest. This approach, although requiring more work to implement, delivered the parsed results in about 60 milliseconds, representing a performance increase of about 130 times.

G. Versioning and Error Reporting

Android Market provides very convenient error collection and reporting functionality, as well as an infrastructure for deployment of application updates. Due to some administrative constraints, Fer Droid was not distributed using Android Market. Therefore, some custom methods were used to overcome these shortcomings. Namely, to collect and report any uncaught exceptions, a Handler was written implementing the UncaughtExceptionHandler interface. The default Android handler for uncaught exceptions was substituted with this custom version which sends the context of the error (i.e. device hardware information, OS and Fer Droid version and stack trace) to a predefined HTTP POST endpoint.

To facilitate simpler application upgrades, every time Fer Droid is started (unless disabled by the user) it requests, using an HTTP GET request, the latest Fer Droid build number. If this number is greater than its own build number, the user is prompted to update the application. If the user chooses to do so, the new version is automatically downloaded and the update process is started using Android's own mechanisms.

IV. Evaluation

To claim that a native mobile application, in this case Fer Droid, will achieve better user acceptance than a more traditional web site user interface (with little or no mobile access optimization) we believe that it should satisfy the following criteria:

- it needs to improve the users' productivity by enabling them to complete their task in less time;
- it should also require less effort from the user by being more user friendly i.e. having a simple and intuitive interface and automating repetitive and boring task; and
- these improvements should not incur any additional expenses for the user, by keeping the network traffic volume comparable to the volume generated by accessing the same information using a mobile web browser.

In the following sections, we will prove that under given conditions, our application is better than traditional web site user interfaces.

A. Network traffic efficiency

Fer Droid accesses various on-line systems to retrieve the selected data. Considering there was no possibility to influence these on-line systems and the ways in which they present the data, and that the majority of these systems are web sites, optimization consisted of loading only the HTTP responses. No other resources, like images and various scripts, are loaded since the web page is not displayed or interpreted but parsed to collect the relevant data.

Fig. 5. List of student's courses

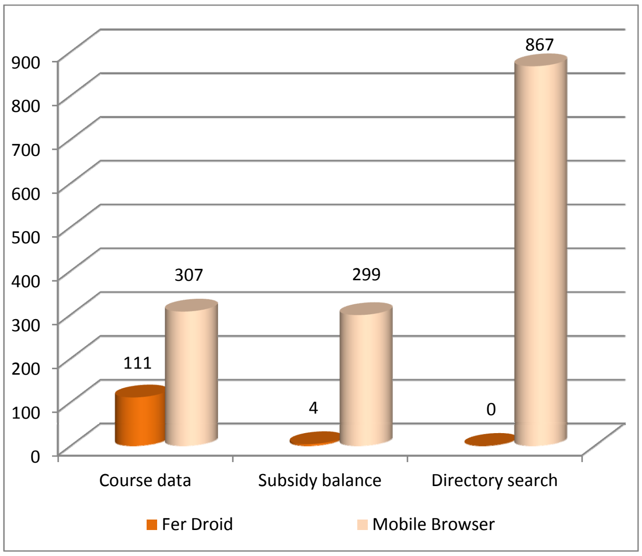

Fig. 6. Network traffic generated for a specific task (in kB)

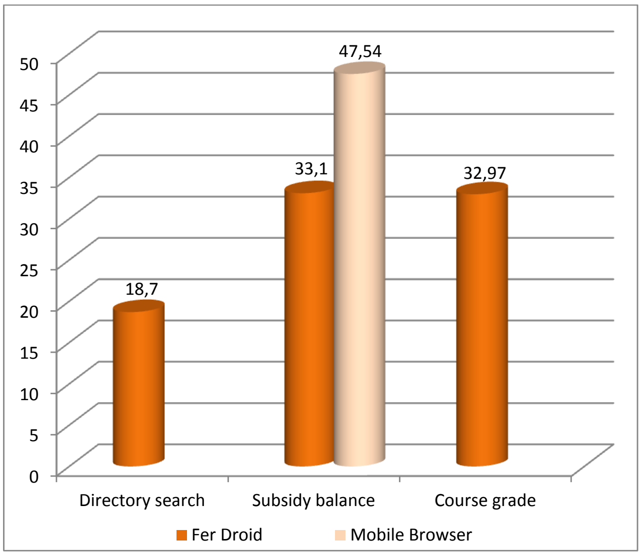

Fig. 7. Time required to complete a task on an HTC Hero (in seconds)

Wireshark, a network protocol analyzer, has been used to capture and measure the traffic volume for three different tasks when done using a mobile web browser and when using Fer Droid with the results shown in Figure 6. Only the data between the device and the target server have been measured in both cases, which means that no DNS or other supporting traffic has been measured.

First task was to retrieve course data for one specific course using the Course List Module. The second task was to check the balance of the monthly food subsidy using the X-Card Balance Module. For these two tasks Fer Droid has generated 2.8 and 74.8 times less traffic than the web browser. The third task was to find a specific person in the directory using the Directory Module. The reason why Fer Droid did not generate any traffic at all for this last task is because it stores this data locally, as described in the section about the implementation of that module.

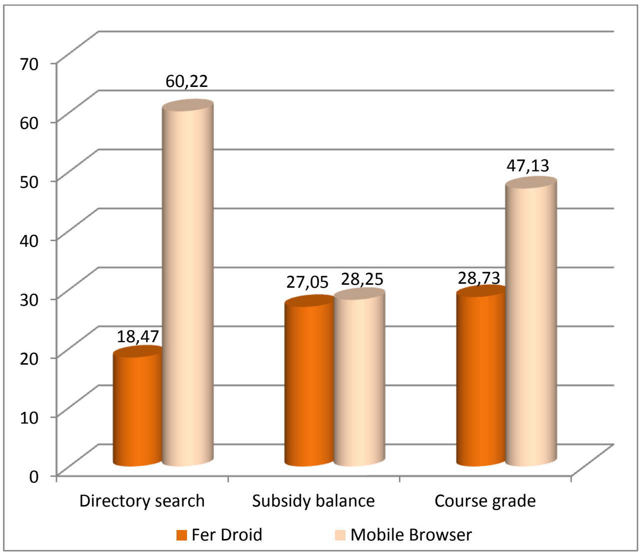

Fig. 8. Time required to complete a task on an HTC Desire (in seconds)

B. *Productivity*

To determine the difference in user productivity a short study was conducted. The study was conducted twice, first using HTC Hero, an older Android device with less processing power and an older version of Android. The version of HTC Hero is 2.1-update1. The second iteration was done on a newer and more powerful device, HTC Desire. The version of HTC Desire is 2.3.3. In the study the participants were timed while performing three tasks using both Fer Droid and a web browser.

1) Study design: The first task was to find a specific person in the directory and call that person. The time measured for FerDroid was from starting the application until selecting the call button. The time measured for the web browser does not include preparing a call to that person since there is no mechanism to automatically transfer the telephone number displayed. Instead, the number needs to be copied or memorized and then entered into the phone's dialer.

The second task was to check the remaining amount of monthly food subsidy. The functionality of Fer Droid and of the web site for this task was identical. The final task was to look up the grade for a specific course. The study was conducted over three days by measuring one person at a time.

2) Participants: Eleven volunteers participated in the study. They were randomly selected volunteers enrolled at the Faculty. They have all described themselves as being both experienced smartphone users and experienced users of the evaluated Student Information Systems. They have not used Fer Droid before and have received no incentives to participate in this study. Due to the small number of participants in the study, the results have only been calculated as a mean, without the standard deviation.

3) Results: As shown in Figure 7, when run on an HTC Hero device, Fer Droid outperforms the browser for the second task, checking the subsidy balance. Due to the version of the operating system on the device and technologies used on the web sites for tasks one and three, it is impossible to complete these tasks by using the web browser. Since there is no official Android upgrade to a version newer than 2.1 available for HTC Hero, it would not be possible to complete these tasks at all without Fer Droid on this device and other devices with older OS versions.

Task completion times when using an HTC Desire continue to demonstrate the same trend, as shown in Figure 8. Due to the newer Android version, all tasks can be completed by using either Fer Droid or the web browser. The first and the third task show Fer Droid's clear advantage, with the completion time for task one when using Fer Droid being only one third of the time when using the web browser. The fact that significantly contributes to this result is the choice of the parsing algorithm, as discussed in the section about the implementation of the Course List Module.

V. User Survey

Since one of the major objectives of this application was to improve the user experience by being more user friendly, measuring only objective and inherently quantifiable performance of the application (i.e. network traffic and task completion time) was not enough to determine our success. A survey was conducted among the users of the application.

A. Survey Design

The survey consisted of fourteen questions regarding the subjective user experience during application use, such as overall satisfaction with the application, ease of use, familiarity of the interface, etc. Participants were able to grade each question with a whole number on a scale from one to five, with five being the highest grade. The questionnaire which was used to conduct the survey was an adapted IBM Post-Study questionnaire, translated into Croatian, the users' native language.

B. Participants

The survey was conducted as an open access on-line questionnaire. The link to the survey was publicized using the same distribution channels as for the application itself: the web site of the Faculty Department, community forum of the Faculty students and the Facebook fan page. The participants received no incentives to participate. Fifty seven people participated in the survey.

C. Results

As shown in Table I, the majority of the survey participants believe that it is easy to use the application. They also agree that the application has a familiar user interface allowing them to quickly and successfully complete their tasks, even at the first attempt. The part which they found needs the most improvement is the applicability. Since the application does not integrate with all Student Information Systems available to the students (only with the ones most commonly used), this lower grade was expected. It also sets out a direction in which to work on improving the results further.

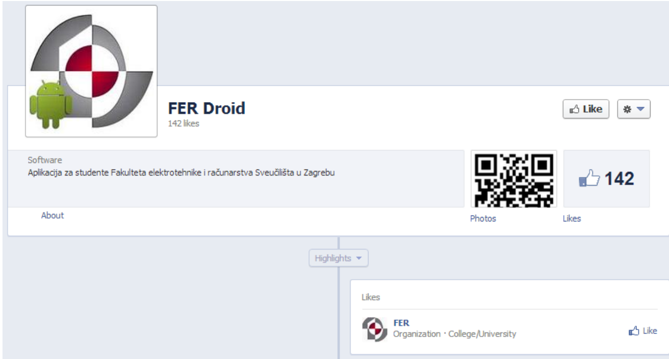

Fig. 9. Fer Droid Facebook fan page screenshot

D. Distribution and Acceptance

After initial testing, the application was made available to the general public. One of the distribution channels was Facebook, where a fan page has been created and 142 Facebook users liked the page as shown in Figure 9.

Another distribution channel was the forum of the student community of the Faculty, where an introductory post with the Fer Droid download link received 104 likes. All of the links to Fer Droid were through the bit.ly service which offers click statistics. It shows that up-to-date there were 1231 downloads of the application, with single day downloads peaking on the first day after the public release when it was downloaded 154 times.

VI. Conclusions

In line with all the objective measurements and subjective survey results, it is to be concluded that an optimized user interface in the form of a native application is better accepted and more convenient to access the data from a Student Information System than having to navigate through desktop version web sites in a mobile browser. It also shows that some older devices with earlier versions of Android, which are still used in large numbers, are completely unable to display some normal web pages due to the way they are written. In such cases there is no substitution for a native application. Possible future work would be to broaden the integration of the system with the smartphone, leveraging other user interface elements available on the Android operating system, such as App Widgets.

Acknowledgment

The authors acknowledge the support of the "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639) project, funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

TABLE I
USER SURVEY QUESTIONS WITH RESULTS

	Mean	Standard Deviation
I can use the application without thinking how to	4.19	0.76
The application fulfils all my requirements	3.84	0.95
It was immediately clear to me how to use the application	4.37	0.83
Interaction with the application is similar to other applications	4.35	0.78
There were no problems while using the application	4.40	0.83
The application is not complicated to use	4.49	0.82
The application enabled me to complete the tasks that I expected	3.98	0.81
It was simple to use the application from the start	4.32	0.88
It was always clear to me what I need to do to get to the information I needed	4.33	0.80
Usage of the application went smoothly	4.30	0.86
Very little effort was needed to use the application	4.32	0.86
The application helped me to fully complete the task which I envisioned	4.14	0.85
I understood right away how to use the application	4.39	0.81
When I wanted to complete some task, I managed to do it at the first attempt	4.19	0.83

REFERENCES

[1] J. L.-H. Bowden, “Engaging the student as a customer: A relationship marketing approach,” *Marketing education review*, vol. 21, no. 3, pp. 211–228, 2011.

[2] L. Thuniki, “Customer relationship management and higher education: A vision,” *Advances In Management*, vol. 4, no. 3, pp. 1–11, 2011.

[3] E. D. Seeman and M. O’Hara, “Customer relationship management in higher education: Using information systems to improve the student-school relationship,” *Campus-Wide Information Systems*, vol. 23, no. 1, pp. 24–34, 2006.

[4] J. J. Vauterin, L. Linnanen, and E. Marttila, “Customer orientation in higher education: the missing link in international student recruitment? A relationship marketing approach,” *Industry and Higher Education*, vol. 25, no. 2, pp. 77–91, 2011.

[5] ——, “Issues of delivering quality customer service in a higher education environment,” *International Journal of Quality and Service Sciences*, vol. 3, no. 2, pp. 181–198, 2011.

[6] P. Kotler and K. F. A. Fox, *Strategic marketing for educational institutions*. Prentice-Hall, 1995.

[7] M. B. Piedade and M. Y. Santos, “Student relationship management: Concept, practice and technological support,” in *IEEE International Engineering Management Conference*, 2008, pp. 1–5.

[8] M. Firtman, *Programming the Mobile Web*. O’Reilly Media, 2010.

[9] P.-P. Koch. (2012) W3C DOM compatibility - core - mobile. [Online]. Available: http://www.quirksmode.org/m/w3c_core.html

[10] ——. (2012) Mobile compatibility tables. [Online]. Available: http://www.quirksmode.org/m/table.html

[11] Faculty of Electrical Engineering, University of Zagreb. (2012) Ferko. [Online]. Available: https://ferko.fer.hr/ferko/

[12] ——. (2012) AHyCo. [Online]. Available: https://ahyco.fer.hr

[13] ——. (2012) FERWeb. [Online]. Available: http://www.fer.unizg.hr/

[14] University Computing Centre Srce. (2012) Centar za autorizaciju prava. [Online]. Available: http://www.cap.srce.hr/ProvjeraSaldo.aspx