

E-STUDENT IOS MOBILE APPLICATION

MLADJAN ANTIĆ, SLOBODAN JOVANOVIĆ, SVETLANA CVETANOVIĆ
Belgrade Metropolitan University, www.metropolitan.edu.rs

Abstract: iOS is the Apple mobile operating system, for Apple mobile devices. Initially developed for iPhone, and later for iPod Touch, iPad, AppleTV. The Metropolitan University has a Web application titled eStudent, which enables students by Web to get information about their marks for all subjects, their finances, exam scheduling, professors and assistants, and send exam registration and feedback about teaching, etc. This paper explains the development of the mobile application eStudent on the iOS platform. This application enables students whenever they want, by using their iPhone mobile phone, to get access to the information from the eStudent Web application, and to present it on their iPhone user interface UI. This paper describes software requirements analysis, system architecture, and user interface UI of the iOS mobile application eStudent.

Keywords: Apple, API, Ruby

1. INTRODUCTION

The Metropolitan University in Belgrade owns the eStudent Web application, which enables students by using Web to get information about their marks, finances, to send exam registrations, and send feedback about teaching quality, etc. However, since more and more students have a smartphone in their pockets, often iPhone, there is a need for eStudent mobile application, to enable students to use eStudent Web application via their iPhone mobile phone, whenever they want.

This paper consists of:

- Software requirements analysis (application functionality analysis)
- Defining system architecture
- Short description of chosen iOS technology
- Description of iOS mobile application eStudent and its API (application programming interface)
- Description of User Interface
- Conclusion

2. SOFTWARE REQUIREMENTS

During the analysis, the software requirements of iOS mobile application eStudent are grouped in separate parts, and the main purpose is to explain how to adapt eStudent Web application to be presented on iPhone mobile device. The eStudent Web application has five separate parts, so consequently, the iOS eStudent mobile application is also divided in five parts (five tabs).

(1)**Reports:** "Reports" show student marks for all current teaching subjects. The main goal here is to adapt as better as possible the presentation of information to the size of the mobile phone screen and to the "native" controls of iOS mobile device (designed by Apple). During this process, the experience of developer is an important factor, and guidelines from Apple Human Interface Guidelines, which advices designers and developers how to present data on the small screen. Figure 1 shows a "Report" of the eStudent Web application for one of teaching subjects.

Izvestaj

Šifra predmeta:

1251 : Marko Cvetković		Smer: OAS - Informacione Tehnologije				Vrsta nastave: Internet	
NEDELJA	PRISUSTVO	ZADACI	PROJEKTI	KOLOKVIJUMI	ZALAGANJE	SEMINARSKI RAD	TESTOVI
1	Cas nije odrzan!	0	0	0	0	0	0
2	Cas nije odrzan!	0	0	0	0	0	0
3	Cas nije odrzan!	0	0	0	0	0	0
4	Cas nije odrzan!	0	0	0	0	0	0
5	Cas nije odrzan!	0	0	0	0	0	0
6	Cas nije odrzan!	0	0	0	0	0	0
7	Cas nije odrzan!	0	0	0	0	0	0
8	Cas nije odrzan!	0	0	0	0	0	0
9	Cas nije odrzan!	0	0	0	0	0	0
10	Cas nije odrzan!	0	0	0	0	0	0
11	Cas nije odrzan!	0	0	0	0	0	0
12	Cas nije odrzan!	0	0	0	0	0	0
13	Cas nije odrzan!	0	0	0	0	0	0
14	Cas nije odrzan!	0	0	0	0	0	0
15	Cas nije odrzan!	0	0	0	0	0	0
	0	0	0	0	0	0	0

Ukupno dobijenih poena	
Kazneni poeni	0
Ukupno poena za predispitne obaveze	0
Broj poena na pismenom ispitu	
Suma poena	0
Oцена	Nije ocenjen

Figure 1: Report page in Web eStudent

(2)**Financies:** This page which shows the finances data (financial obligations, debts, and payments during the school year, etc.). Here, it is important to have an easy access to the finances data both from the current and from the previous year. So, in the navigation bar, it will be enabled to access easily the previous year, not just the current year. Also, it is convenient to separate payments and obligations into two separate tables, so some data filtering patterns can be applied. Figure 2 presents the student finances page in eStudent Web application.

Školska godina: 2011/2012	1251 : Marko Cvetković	Smer: OAS - Informacione Tehnologije	Vrsta nastave: Internet
---------------------------	------------------------	--------------------------------------	-------------------------

Zaduzenja

VID ZADUZENJA	DATUM	IZNOS EUR	UMANJENJE
OAS IT - Internet	01-OCT-11	2,000.00	0.00
Preneseni ispiti		1,050.00	0.00
	Ukupno(Eur):	3,050.00	0.00

Placanja

VID PLACANJA	DATUM	IZNOS DIN	VALUTA	IZNOS EUR
OAS IT - Internet	04-NOV-11	50,438.96	101.9199	495.00
	Ukupno:	50,438.96		495.00

Zaduzenje(Eur): 3,050.00
 Placeno(Eur): 495.00
 Dugovanje(Eur): 2,555.00

Figure 2: Financies page in Web eStudent

(3) Exam registration: The exam registration page is very important for students, because it is one of the most important functions of eStudent application. It must show all relevant information, eg.

- confirmation about exam registrations
- registration deadline
- data about dates and times of scheduled exams, etc.

After the requirements analysis, it was decided to present the whole exam scheduling list on the same screen, with the possibility to reset the exam date and time by choosing the relevant date from the list. Also, it is decided to enable the exam registration to be performed easily by a single tap (pressing a single button). And, the server response will be showed in a corresponding notifying window. Figure 3 shows the present look of the exam registration page in eStudent Web application.

Decembarški rok

Školska Godina: 2011/2012
 1251 : Marko Cvetković

Prijavlivanje do 29-NOV-11 u 22:00h

RASPORED POLAGANJA

SIFRA	PREDMET	NASTAVNIK	PLAN	MESTO	DATUM I VREME
<input type="button" value="Prijavi"/>					

Figure 3: Exam registration page in Web eStudent

(4) Professors: The page “Professors” gives relevant information about lecturers and teaching assistants for relevant teaching subjects for every student. It was found that all relevant data can be presented in a single table. Figure 4 presents the page “Professors” in the eStudent Web application.

Profesori

CS230 Distribuirani sistemi	
dipl.ing. Mateja Opačić	Asistent
Andrej Luneski	Asistent
Lovro Ilijašić	Profesor
IS201 Osnove informacionih sistema(Prenet)	
Dr Svetlana Cvetanović	Profesor
Boško Mijin	Asistent
Valentina Paunović	Profesor
Valentina Paunović	Asistent
IT250 Web sistemi(Prenet)	
dipl.ing. Mateja Opačić	Asistent
Stefan Mirkovic	Asistent
Dušan Tatić	Asistent
Lovro Ilijašić	Profesor
IT320 Savremene tehnološke platforme	
dipl.ing. Mateja Opačić	Asistent
Dr Nikola Milivojević	Profesor

Figure 4: Professors page (in Web eStudent)

(5) Student’s feedback: Students send their feedback about teaching quality (their satisfaction and impression about teaching quality for every teaching subject) twice a year (once during a teaching semester). After the requirements analysis, it was decided to present relevant data on the mobile screen as a HTML document, created by the University Web server and sent to the mobile device after a request sent by student’s mobile phone.

Besides the previous five requirements regarding the data presentation, we have another three requirements: (6) Analysis of API:API - Application programming interface, has the goal to provide input data for eStudent mobile application. The input data will be taken from the Web server. For each Web server page, a HTTP request for getting data (from the Web server) will be sent, and the Web server will sent the required data (to the iOS eStudent mobile application) in the JSON format. However, since the Information system of the Metropolitan university (ISUM) is in the process of upgrading and renovating, and that the final version of API is not ready in the moment, a prototype API was created. This prototype API will be used as a good representative, during the process of finalizing the future API in the final ISUM environment, which will enable iOS eStudent to work online in the real environment.

(7) Number of API requests: In order to maximize the user satisfaction, for every server page (for every of five basic screens), data will be sent separately. In this way, we minimize the size of JSON files, for 3G (or Edge) connection, and the user will not wait much during the process of inputting data sent from the Web server. (8) Secure authentication and authorization: This requirement is resolved simply by requesting from the user to input

his/her name, family name and a password. After this, the user can access the Web application. During every HTTP request sent to the Web server, it is necessary to do the basic HTTP authentication, to block any unauthorized use of API. Besides this basic HTTP authentication, for every request, the student ID number is requested in order to get data for the corresponding student.

3. SYSTEM ARCHITECTURE

The system architecture is given in the Figure 5, resulting from system requirements. It is a distributed mobile application, where iOS eStudent mobile application communicates with the eStudent Web server by using the HTTP protocol (GET, POST), while the data are in the JSON format. Native applications for iOS are written using the Apple framework Cocoa Touch (for mobile interfaces). The structure of the iOS eStudent mobile application is divided in four parts (Figure 6):

- Models, describing data
- Presentation layer - .xib files, describing user interface UI
- Controllers – Business logic which controls UI and manipulates data
- Singleton object, responsible for communicating with API

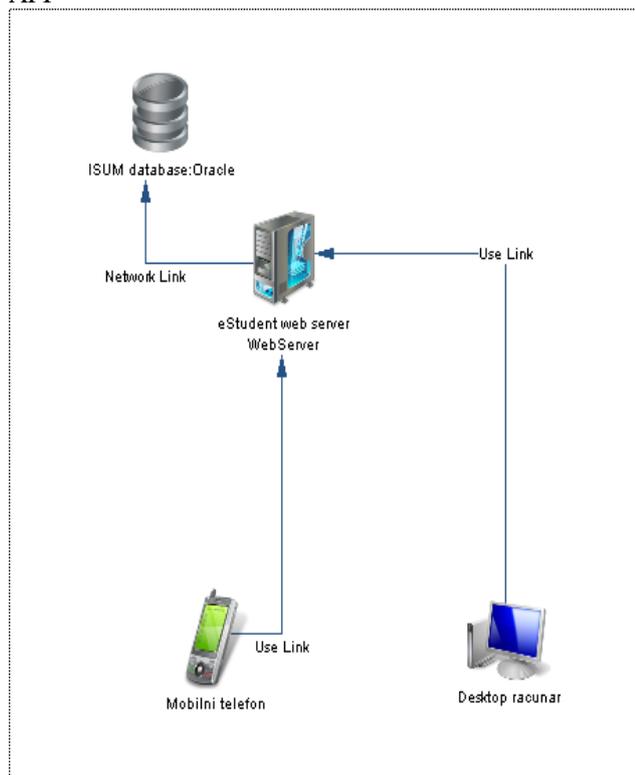


Figure 5: System architecture

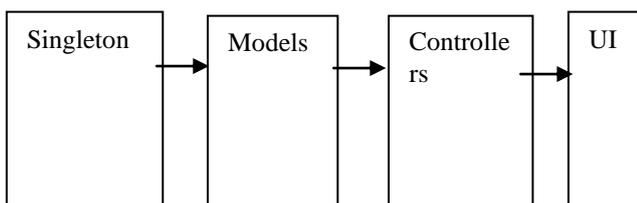


Figure 6: Structure of iOS eStudent

4. IOS PLATFORM

iOS (iPhone OS) is the Apple mobile operating system (operating system for Apple mobile devices). Initially developed for iPhone, and later for iPod Touch, iPad, AppleTV. Regarding statistics from October 2011, AppStore (Apple shop of iOS applications) had more than 500,000 iOS applications, downloaded more than 18 billion times! Couple of arguments why to use iOS platform?

- Nonsegmented market
- Excellent user satisfaction (experience) with iOS devices
- precise and clear advices for developers
- many students use iOS devices (iPhone, iPad, ...)

A very important reason for recommending the iOS operating system, is that from the beginning iOS (version 1.0) had a support for “direct manipulation” concept (touch screen) and *multi-touch gestures*. On the opposite, the Android operating system was developed initially for using a keyboard, and without a touch screen support/vision, več je planirano da osnovni unos podataka bude preko tastature, what was logical at that time when touch screen was not so popular. After appearing of iPhone-a in 2007, Google bought the whole Android team, and decided to develop Android further, to be able to use touch-screen. Today, it is clear that Android devices, although with more powerful hardware, compared with Apple devices, have worse performances regarding touch-screen capabilities [1].

From the beginning, Apple established some rules and restrictions for writing their applications. Their applications are written in Objective C language only (similar with C++), and a developer licence is required, and the final application must satisfy a “review” process specified by Apple, before appearing in AppStore. Also, there is no a GarbageCollector for iOS, which forces developers to take care about memory, on the device with limited hardware capabilities, so, the final product is an application which is executing fluently on all Apple mobile devices. UI must satisfy criterions from HIG (Human Interface Guideline), resulting in high quality of applications in AppStore. Developers know they must cooperate with a good UI and UX designer, because it is regarded as crucial for creating a good product [2].

Android devices use many different screens, of different size, and quality. This is a problem for developers and designers. On the other side, Apple has strict standards. All iPhone mobile phones have the same screen size (3.5”), and only two different resolutions, which is very convenient for developers. Many students of the Metropolitan university have an iOS device (iPhone, iPad, iPod Touch), so iOS is logical choice of the mobile operating system for eStudent mobile application.

5. USER INTERFACE

Login View Controller appears after starting (opening) the application, in order to enable the access of the user to the system, and to send necessary data to the Web server.

After the tap of the Login button, the iOS eStudent mobile application sends a HTTP POST request to the Web server, to check entered data via eStudent Web application and API. If entered data are correct, Login View is removed and further steps are allowed. If entered data are incorrect, in Login View, the view is unchanged, and the user can try to enter data again.

Apple's iOS UI is very good, and other big IT companies are trying to copy it. One of UI controls on iOS, very practical, and used in all applications with several views (more than 2-3 views) is TabBar. TabBar is the control which enables quick change between views, it is always visible and is placed on the bottom of the screen. TabBar controller provides five different view controllers:

- Report View Controller
- Financies View Controller
- Application View Controller
- Professor View Controller
- Feedback View Controller

Report ("Izveštaj") View Controller at its first appearance takes fresh data from eStudent Web application using GET API request. After parsing the JSON file (obtained from Web eStudent), the table is filled in with the list of all teaching subjects relevant for the particular student. Title and ID for every relevant teaching subject is shown in the single table. Every box in the table has an arrow, indicating to the user to tap the box in order to get more information about the particular subject (Figures 7 and 8). This new set of data are shown immediately, they are already in the mobile phone, taken from Web eStudent in the current JSON file. There are 16 sections, the first section is summing up, showing data for the whole semester for the chosen teaching subject, and the rest of 15 sections represent 15 weeks of the teaching semester, and the student marks for each week.

Exam registration ("Prijava ispita") View Controller: After a tapping tab "Prijava", the iOS application contacts the Web application using HTTP GET method, and collects all the information needed for this view:

- List of subjects which are enabled to the student to try to pass in the current examination period
- list of dates and times for every subject
- Deadline information

Figure 9 shows a view with a list of subjects and corresponding examination dates/times. If any subject is already registered for examination, this is then clearly marked and the examination date/time is given. Also, if the student changes his mind about exam registrations, he can make changes in the examination registration.

Financies ("Finansijski Izveštaj") View Controller: Financies View shows student's financial obligations, payments and debts, to the Metropolitan university. Different data are presented in separate boxes, enabling a good visibility on a small space. There are three sections: Sum, Obligations, and Payments. Using segmented control, students can filter presented data, depending what is interesting for them in the moment, to choose either a summing report, or a detailed report with obligations and payments.

Feedback ("Anketa") View Controller: For each semester, students have to send a feedback about their satisfaction with teaching (lectures, professors, teaching materials). Since it is more practical to send this data in HTML format, there is a WebView embedded in Feedback View Controller, enabling creating HTML code, and this HTML code will be sent by Web server (ie. by Web eStudent application). Feedback is filled in as a multiple choice document. During this process, students can use the Next button, located in the navigation bar UIPickerViewControl, enabling a quick movements through the feedback document. This small but useful functionality significantly speeds up the process.



Figure 7: Report View Controller



Figure 8: Report for IT 250



Figure 9: Exam registration View Controller

6. API

Many mobile applications communicate with Web servers, making it interactive and interesting. Eg. Twitter, Facebook, etc. communicate with Web services using HTTP protocols. But, the question is how such mobile applications communicate with Web applications? Answer is, they do it using Application Programming Interface (API) [3,4].

API enables developers of mobile applications (or any other application) to connect it easily with a Web application, and enable using all functions defined by API. In most cases, API is the only open point of an application towards the external world, so it is important to make API safe against any potential security risks. Figure 10 presents the connection between iOS eStudent mobile application, API and Web eStudent application, iOS operating system and UI.

During the development of iOS eStudent mobile application, it was decided to make first a prototype API layer ("Demo API"), which enables the application to work and to be fully developed, and the final (real) API can be developed in some later stage, after finalizing the ISUM information system. This Demo API will enable an easy development of the final API, as an integral part of ISUM (Information system of University Metropolitan), which will operate on a production data base. Demo API is produced in Ruby On Rails-u, a popular Web framework for Ruby, and for every API request it was necessary to define details regarding input parameters and return information. API was tested by RestClient application (<http://code.google.com/p/rest-client/>). API layer is protected by BASIC HTTP Authentication, and, concretely, in this Demo API, the access data are: username: mladjan, password: mladjan.

This is the list of API methods:

- login_client
- get_reports
- get_profesors
- get_open_courses
- getFinance

- makeCoursesSignup

API program listing has approximately around 10 pages. For example, the method `get_profesors`. This method returns data about professors and corresponding teaching subjects, in the JSON format. Expected parameters are: name, surname, student ID number. These parameters must be sent as parameters of a POST method! If the request is successful, this method will return data in the JSON format. In case of any error, the method will return ERROR (in text format).

Objective-C is an object oriented programming language, made by combining koji je nastao the syntax of C and *Smalltalk* [5]. Today, Objective-C is well known as the language of OSX and iOS applications, and supported by Apple Inc. Like C/ C++, the program code in Objective-C is separated in two separate files, for every program class(the header file, and implementation fajl). The iOS eStudent application is written in Objective-C. The program listing in Objective-C has around 60 pages.

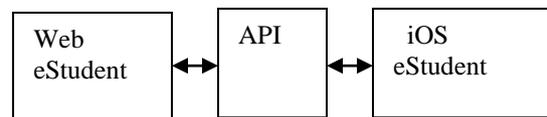


Figure 10: API connection

7. CONCLUSION

This paper describes the iOS eStudent mobile application, developed to enable the students of the Metropolitan University, via their Apple mobile phones, to access data from the Web application eStudent. The paper explains some important aspects of planning, analysis, development of an iOS mobile application. The focus of the paper is the design and organization of the mobile application, and its communication via API. Design simplicity of the iOS eStudent, intuitive UI, and its speed, are its advantages. iOS eStudent is dependent of the corresponding Web servis, ie. Web application eStudent, which means that any new function of iOS eStudent must be followed by development of Web eStudent.

Acknowledgment: This work was supported by Ministry of Science and Education of Serbia (Project III44006).

LITERATURE

- 1] <http://www.asktog.com/columns/070iPhoneFirstLook.html>
- 2] <http://developer.apple.com/appstore/guidelines.html>
- 3] **CS193P iPhone Application development**, Stanford University, <http://www.stanford.edu/class/cs193p/cgi-bin/drupal/>
- 4] **Vandad Nahavandipoor**, iOS 5 Programming Cookbook, O'Reilly Media 2011, <http://oreilly.com/iphone/>
- 5] **Stephen G. Kochan**, Programming in Objective-C, Sams Publishing, 2011, ISBN 0672325861, www.amazon.com