Development of eStudent iOS Mobile Application

http://dx.doi.org/10.3991/ijim.v7i1.2295

M. Antic, S. Jovanovic and S. Cvetanovic
Metropolitan University, Belgrade, Serbia

***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 financies, exam scheduling, professors and assistents, 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 explains in details software requirements analysis, system architecture, system modelling, and UI of the eStudent iOS mobile application.**

***Index Terms*—Apple, API, Ruby On Rails, Objective C**

I. INTRODUCTION

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 Oct.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 [1,2]?

- Nonsegmented market
- Excellent user satisfaction (experience) with iOS devices
- Precise and clear advices for developers
- Many students use iOS devices (iPhone, iPad, …)

The Metropolitan University in Belgrade uses the eStudent Web application, which enables students by using Web to get information about their marks, financies, 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 explains the development of the eStudent iOS mobile application. It consists of:

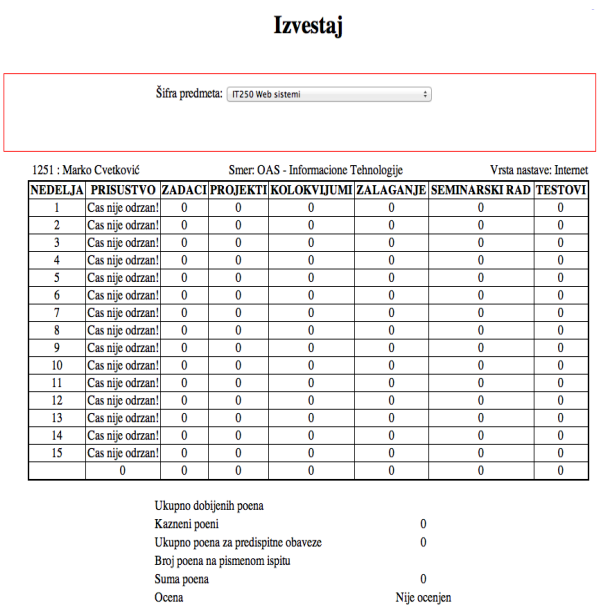
- Software requirements analysis (application functionality analysis)
- Defining system architecture
- Models of eStudent iOS mobile application
- API (Application Programming Interface) development
- Development of User Interface

II. SOFTWARE REQUIREMENTS ANALYSIS

During the analysis, the software requirements of iOS mobile application eStudent are groupped 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 devopers how to present data on the small screen. Fig. 1 shows a "Report" of the eStudent Web application for one of teaching subjects.

(2) Financies**:** This page shows the financies data (financial obligations, debts, and payments during the school year, etc.). Here, it is important to have an easy access to the financies 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. Fig. 2 presents the student financies page in the eStudent Web application.



Izvestaj

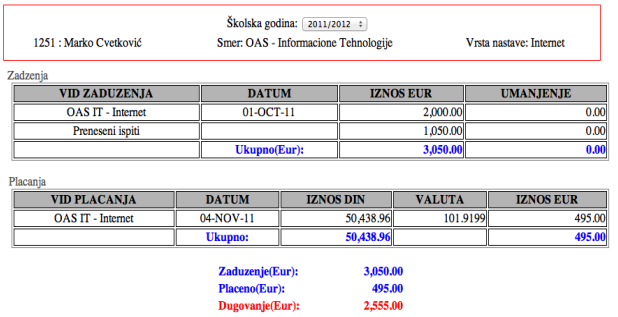
Šifra predmeta: IT250 Web sistemi

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
Ocena Nije ocenjen

Figure 1. Report page in Web eStudent


Školska godina: 2011/2012

1251 : Marko Cvetković | Smer: OAS - Informacione Tehnologije | Vrsta nastave: Internet

Zadzenja

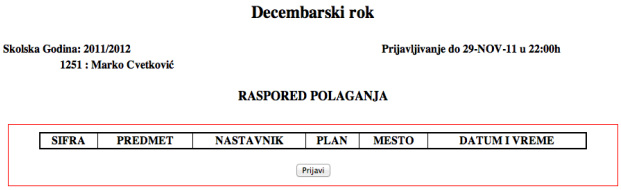
VID ZADUZENJA	DATUM	IZNOS EUR	UMANJENJE
OAS IT - Internet	01-OCT-11	2,000.00	0.00
Preneseni ispiti		1,050.00	0.00
	Ukupno(Eur):	3,050.00	0.00

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


Decembarski rok

Skolska Godina: 2011/2012 **Prijavljivanje do 29-NOV-11 u 22:00h**

1251 : Marko Cvetković

RASPORED POLAGANJA

SIFRA	PREDMET	NASTAVNIK	PLAN	MESTO	DATUM I VREME

Prijavi

Figure 3. Exam registration 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 chosing 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. Fig. 3 shows the present look of the exam registration page in the eStudent Web application.

(4)Professsors**:** The page "Professors" gives relevant information about lecturers and teaching assistents for relevant teaching subjects for every student. It was found that all relevant data can be presented in a single table.

(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 tea\xching 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 Metropoltan 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

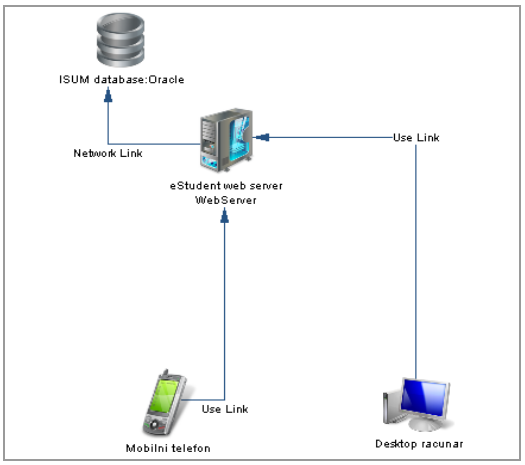


Figure 4. System architecture

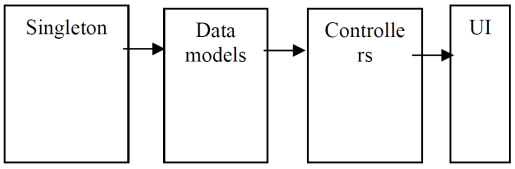


Figure 5. Structure of iOS eStudent

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 the Web server.

(8) Secure autentification 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 autentification, to block any unauthorized use of API. Besides this baic HTTP autentification, for every request, the student ID number is requested in order to get data for the corresponding student.

III. SYSTEM ARCHITECTURE

The system architecture is given in the Fig. 4, resulting from system requirements. It is a distributed mobile application, where the iOS eStudent mobile application communicates with the eStudent Web server by using the HTTP ptotocol (GET, POST), while the data are in th 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 (Fig. 5):

- Data 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

IV. PROCESS MODEL AND OBJECT MODEL

On the highest abstraction level, all processes are shown in a form of a tree, where we can see also subrocesses presented. Five separate processes: report viewing, financies viewing, exam registration, professors viewing, and sending feedback. Fig. 6 shows this process model.

Fig.7 i 8 present object models of the eStudent iOS application. Based on defining system Use Cases, set of program classes has been identified. In Fig.7 the following classes are included: Singleton, Subject, Week, Semester, Course, CourseRegistration, Professor.

In Fig.8 classes are: ReportViewController, DetailedReportViewController, FinanciesViewController, FinanciesYearViewController, RegistrationViewController, RegistrationDetailedViewController, ProfessorsViewController, FeedbackViewController, LoginViewController.

V. USER INTERFACE

Login View Controller **(Fig. 9)** appears after starting (openning) 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.

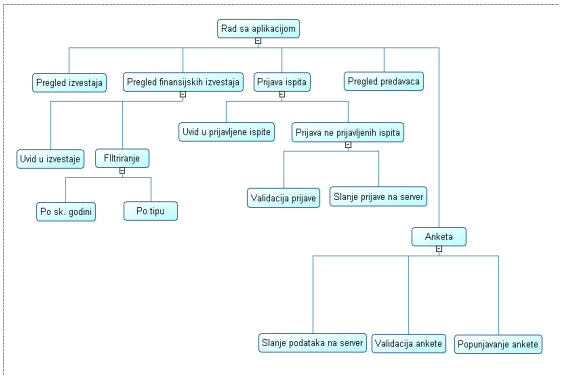


Figure 6. Process model in eStudent iOS

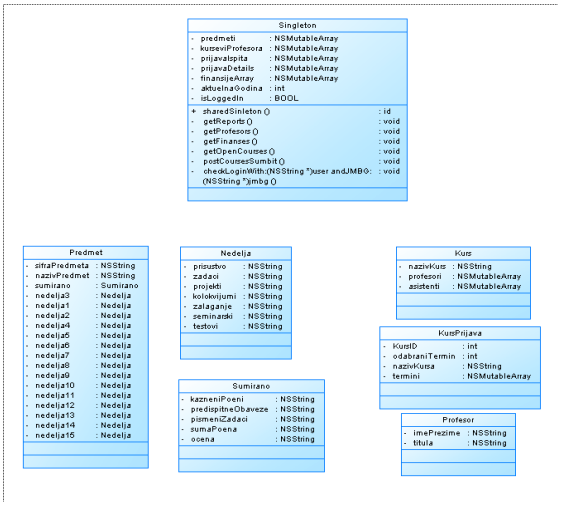


Figure 7. Data object models and Singleton class

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 visable 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 (Fig. 10 and 11). 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 theWeb application usung 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

Fig. 12 shows a *view* with a list of subjects and corresponding examination dates/times. I f any subject is already registrated 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.

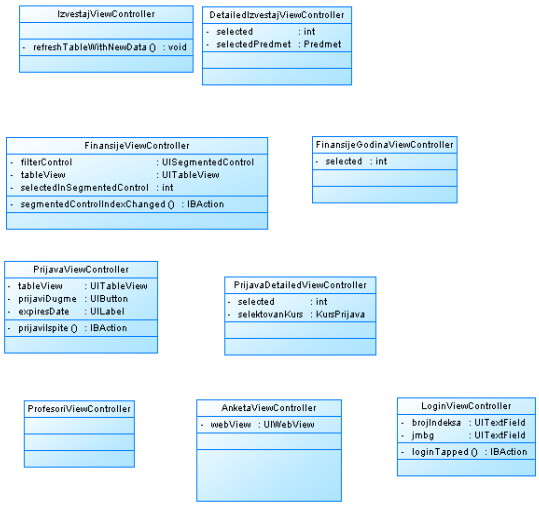


Figure 8. View controllers object model

Feedback ("Anketa") View Controller (Fig.13)**:** For each semester, students have to send a feedback about their satisfaction with teaching (lectures, professors, teaching msaterials). 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 PickerViewControl, enabling a quick movements through the feedback document. This small but useful functionality significantly speeds up the process.

Financies ("Finansijski Izveštaj") View Controller (Fig.14)**:** 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, Oblgations, 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.

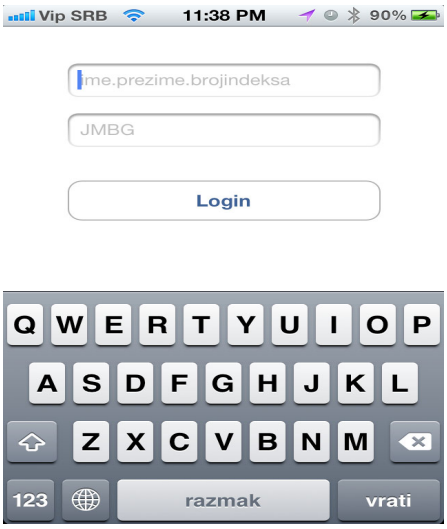


Figure 9. Login View Controller

Figure 10. Report View Controller

Figure 11. Report for IT 250

Figure 12. Exam registration View Controller

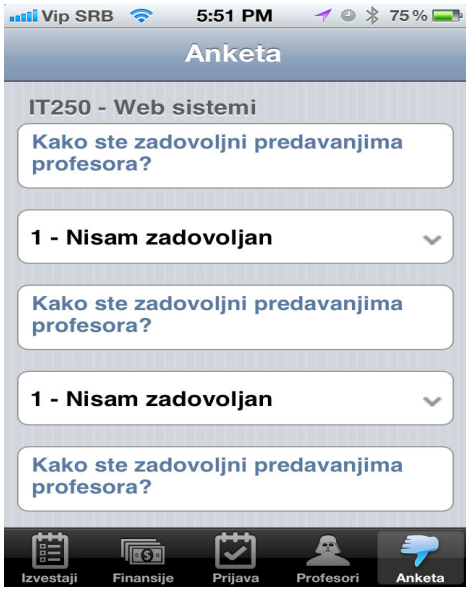


Figure 13. Feedback View Controller

Professors view is a simple view. After a tab "Professors", a request is sent to the Web server, for fresh data. Then, a table is formed with several sections, each section for an exam, and within the section are data about lecturers and assistents for corresponding subject. Fig. 15 shows this view.

Every view controller works independently between each other. Fig. 16 presents view control hierarchy in the eStudent iOS mobile application.

Figure 14. Financies report

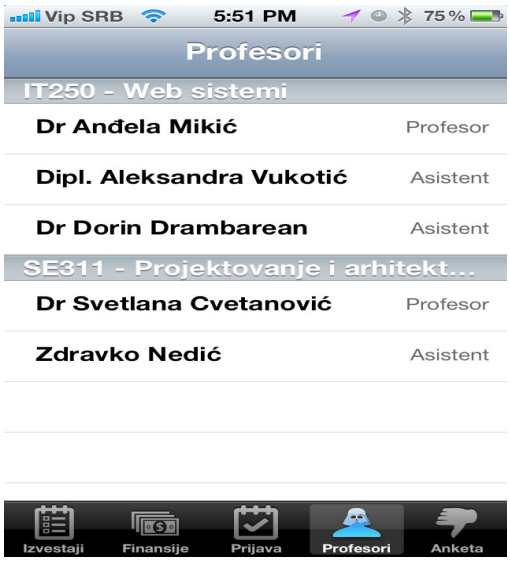


Figure 15. Professors View Controller

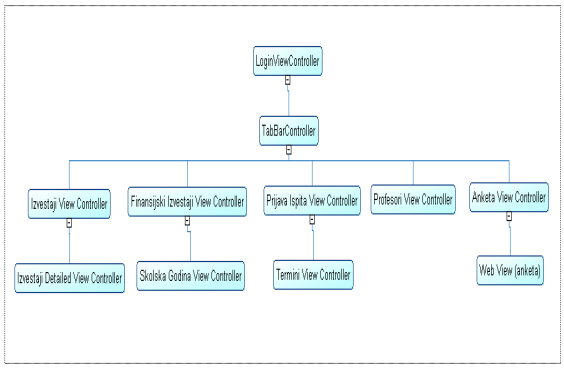


Figure 16. View Controllers hiearachy

VI. API

Many mobile applications communicate with Web servers, making it interactive and interesting. Eg. Twitter, Facebook, etc. communcate 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 tpwards the external world, so it is important to make API safe against any potential security risks. Figure 17 presents the connection between the iOS eStudent mobile application, API and the Web eStudent application, and iOS operating system.

During the development of the iOS eStudent mobile application, it was decided to make a prototype API layer first ("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, a popular Web framework for Ruby, and for every API request it was neccessary 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 Autentification, and, concretely, in this Demo API, the access data are: username: mladjan, password: mladjan.

This is the lst 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).

Format of the JSON file returned by the method is:

```
[ {
"profesori" : [ "Dr An•ela Miki•" ],
"asistenti" : [ "Dipl. Aleksandra Vu-
koti•", "Dr Dorin Drambarean" ],
 "predmet" : "IT250 - Web sistemi"}, {
"profesori" : [ "Dr Svetlana Cvetano-
vi•" ],"asistenti" : [ "Zdravko Nedi•"
],
"predmet" : "SE311 - Projektovanje i
arhitektura softvera"} ]
```

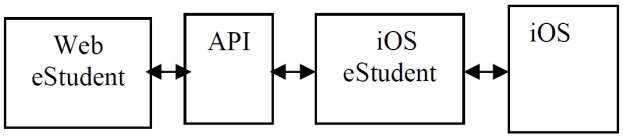


Figure 17. API connection

VII. OBJECTIVE-C PROGRAM LISTING

Objective-C is an object oriented programming language, made by combining the sintax of C and *Smalltalk* [5, 6]. 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.

Here, we have only a small section of this program listing, as an illustration. This is one of simple program classes, titled Kurs:

```
// Kurs.h
// eStudent
//
// Created by Mladjan Antic on 1/3/12.
//       Copyright       (c)       2012
__MyCompanyName__.All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Profesor.h"
/**Model for Course*/
@interface Kurs : NSObject
/**Course name*/
@property nonatomic   retain   NSString
*nazivKursa;
/**Professors*/
@property nonatomic   retain   NSMutab-
leArray *profesori;
/**Assistents*/
@property nonatomic   retain   NSMutab-
leArray *asistenti;
@end
```

And, implementing the class Kurs looks like this:

```
// Kurs.m
// eStudent
//
 // Created by Mladjan Antic on 1/3/12.
//       Copyright       (c)       2012
__MyCompanyName__.All rights reserved.
//
#import "Kurs.h"
@implementation Kurs
@synthesize   nazivKursa,   profesori,
asistenti;
@end
```

VIII. CONCLUSION

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

ACKNOWLEDGMENT

REFERENCES

[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

[6] M.Antic,S.Jovanovic, S.Cvetanovic, eStudent iOS mobile application, 3rd Int. Conf. eLearning-2012, Belgrade, Sept. 27-28.

AUTHORS

Mladjan Antic was with the Belgrade Metropolitan University until 2011, as a software developer. He is now with his own company, which produces mobile applications software (e-mail: mladjan.antic@me.com).

Slobodan Jovanovic, was with Strathclyde University, Glasgow, Scotland during 1993-2008. Since 2008 he is with the Faculty of information technology, Belgrade Metropolitan University, as a professor in Computer science (e-mail: Slobodan.jovanovic@metropolitan.ac.rs).

Svetlana Cvetanovic has been with the Faculty of information technology, Belgrade Metropolitan University, since 2007, as a lecturer in Software Engineering and a director of Master studies (e-mail: svetlana.cvetanovic@metropolitan.ac.rs).

This work was supported by Ministry of Science and Education of Serbia (Project III44006). Received 1 October 2012. Published as resubmitted by the authors 20 December 2012.