

## **USC Course Registration Mobile Application**

### **1. Introduction**

Course registration is a crucial part of students' campus lives. In mobile era, it is highly important to take a revolution of the traditional PC based web registration. The new registration system should be loaded in every one's pocket, fact located courses you want, and even smart to give you recommendations.

Our Course Registration Mobile App is an echo of these requirements. This app is iOS and Android based, and designed in content focus style. We help people in two ways:

1. Fast cookbook method to complete course registration. If people know exactly what courses they want to take, they can just follow a few steps to secure choices of courses of the semester.
2. Immersing exploration of courses. If people are doubtful with different options in trying to find out an optimal combination, or without any options, they can explore the courses by taking advantage of several efficient tools:
  - a. Filtering. People can browse courses by filtering school, department, major, start time, etc.
  - b. Multi-Calendar. People not only have a calendar to see or edit their course combination in a high level, but can create more calendars to store different combinations to compare among.
  - c. Course Details. Forget about google. Now people can view course detail information in the app, including brief introduction, prerequisite courses, sections, time, professors, etc. We are also considering to add some interesting features, such as course difficulty, into it.
  - d. Recommendations. People are lazy now. We know it, and we would like to feed them course recommendations based on their course taking history and similar people's history.

In sum, although our app is a honed system focusing on providing burden-less registration experience, people can also explore our app on different layers (check course details, create calendars, etc) if they want to.

### **1.1 Purpose of this document**

This document is a written description of our Course Registration Mobile App. This document gives us a overall guidance to the architecture of this project. Thus, it is a stable reference, outlining all parts of our app and demonstrating how they will work. This document gives a fairly complete description, while maintaining a high-level view of the app. Furthermore, we will discuss our design decision and trade-offs

### **1.2 Scope of the development project**

Student will have a fairly convenient choice to register courses other than sitting at a table. People can really focus on the important content - courses. They swiftly complete registration, conscientiously compare course combinations, and thoroughly explore details of courses. Moreover, lazy bones can take course recommendations with blink.

### 1.3 Definitions, acronyms, and abbreviations

RESTful Services - Representational State Transfer Service

UI - User Interface

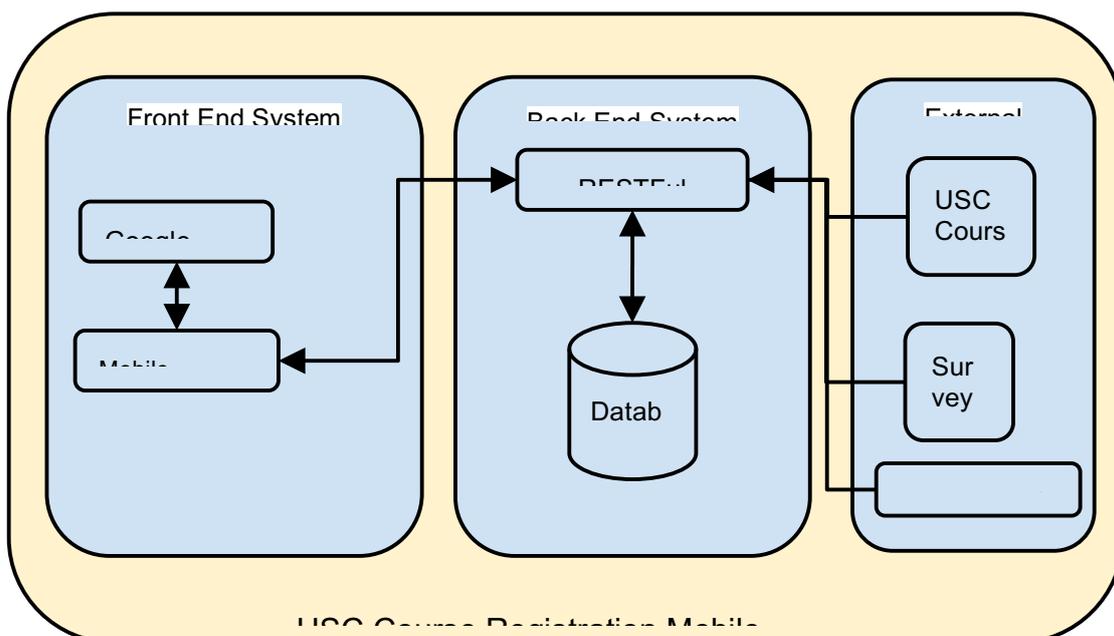
### 1.4 References

Not applicable.

### 1.5 Overview of document

This document firstly gives the description of system architecture in high-level view. Secondly, it dives into components detail design. We will describe our consideration of each component, and demonstrate what those components are. After that, we will talk about reuse and relationship between our app and other systems. Our app is not build from nothing. Thus in this part, we will analyze what other systems or components we take advantage of. Furthermore, we will discuss our design decision and trade-offs. Finally, pseudo code of each component will be described.

## 2. System architecture description



The system architecture can be broadly divided into three subsystems. They are front-end, back-end and external systems. In this section we will take a deeper look into each subsystem and their components.

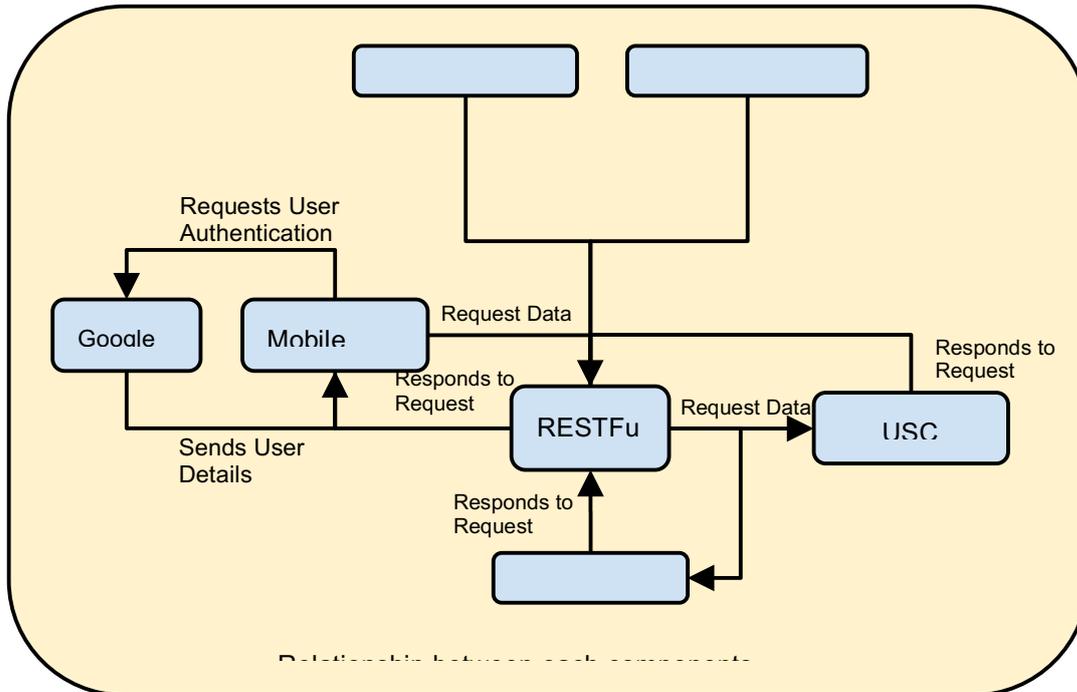
### 2.1 Overview of modules / components

As seen from the above figure, each subsystem is further divided into different components based on their category. In addition with traditional front-end and back-end system, the system also have external data sources which are integrated in different ways. The front-end system is divided into two user-facing components, they are Google services and the Mobile application. The back-end system is also divided into two components, one which acts as servicing and computation component, the

RESTful service. The other component is the storage unit, backed by a MySQL Server. Finally, we rely on certain external components to fetch data about the courses from the university, and other external sources for additional features.

In the following subsection, the components are analyzed with respect to how they are related to each other to perform the required tasks.

## 2.2 Structure and relationships



The above diagram describes the relationship between each component in the system architecture. The central unit of the system is the REST based web server which handles and process all the requests made by the user. It also handle the incoming data from external sources and interacts with database. The mobile application authenticates the user by using “Sign In with Google” feature where user can use USC’s google application account to sign in and authenticate his credentials. Then, the REST services handles the user profile request by creating a new profile for the user in the database. The mobile application also sends other requests to fetch data from the database and also from USC course API to the REST services.

The RESTful service is backed by a MySQL database which stores information about the users, their activity in the application and other incoming data from external sources. There are two external data component that we are planning to use in our application, which are surveys and RateMyProfessor service. Surveys are used to collect certain courses information from the user and use it to provide intelligent suggestions to the user. RateMyProfessor is used to gauge the rating and satisfaction of the students for a course offered in USC.

## 2.3 User interface issues

The major issue with the user interface will be to come up with a intuitive way to display the vast information in a device such as mobile phones with small screen sizes.

The user interface is usually developed over time, with constant iteration of development with feedback from users with different constraints. Therefore, the short development period is also one of the issues with development of UI.

### 3. Detailed description of components

As the overview of components shown above, this chapter will present the detailed information of each components, including how it works and creative features used in this mobile application.

#### 3.1 Component overview description

This application involves three main ends, which are mobile-end, server-end, and usc-api-end. Here, we will summarize the main components of each end below.

Component	Brief Description
RESTFul Service	Worked as a middle layer between USC Course API and Mobile Application
Database	Provide data acquisition and storage service for some specific features of mobile application
Data Process	Dig valued information behind data to do some advanced work, such as course recommendation using external data.
USC API	Provide JSON format data for mobile apps
Mobile Application	It's a mobile app deployed in iOS platform, which provide well-designed user interface and functions to improve user experience during course registration
Google Services	Based on these service provided by Google, this mobile system could implement user log on and other creative features.

#### 3.2 RESTFul Service Component

The REST based web server which handles and process all the requests made by the user. It also handle the incoming data from external sources and interacts with database. This component is backed by MySQL database to store the data.

REST Call examples:

GET: /courses?dept=XXX&day=XXX&time=XXX

GET: /ratings?pid=XXX

POST: /user/add?cid=XXX&term=XXX

#### 3.3 Database

In this component, we construct a database based on mysql to store valued data. There are many kinds of data we need to store for our mobile application system, like there unprovided data by USC API and these fetched data from website. The main data we have, need to get, and need to store is shown as follows.

Entity	Description
Course	Data fetched from USC Courses API, which may be stored in Database for further processing
Section	
Term	
Session	
School	
Department	
Detail	
Professor	
Student	
Calendar	There are many choices for students to choose their courses, and we can store and update students' different calendar choice for the same semester. Furthermore, we could use this data to conduct some machine learning algorithm so as to find some interesting data we can provide for mobile app.

### 3.4 Data Process

Data from external sources will be extracted and processed to give additional information to the students. Such as the professor rating from RateMyProfessor.com. Additionally, collect data using survey forms to determine the course choices pattern from previous student. Thus, allowing the system to suggest course options to new/junior students.

### 3.5 USC API

USC API is provided by the registration department of USC to give access to course information.

### 3.6 Mobile Application

The mobile application is developed in both iOS and Android platforms, as most of the students use either of these platforms. The user interface are designed to be simple, basic and precise for the student to pick and filter the courses.

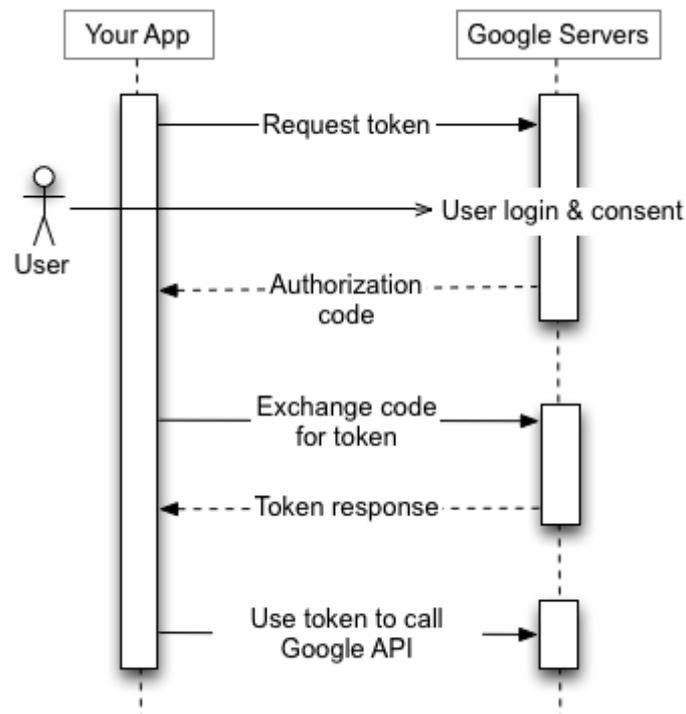
There are some main features we could provide in this mobile app, which would be discussed as follows:

Feature	Description
User Authentication	This application provides the basic authentication to the student, so that courses picked can be picked uniquely. As the authentication is google server based, the extra necessary security can be provided to the students.
Register Guider	This part of the application will guide the students, based on the data the other students register. This is also merged with the rateMyProfessor API, so that the students get more detailed idea about which courses should be picked and under which professor.
Information Viewer	This section shows the filterable results in an expandable list view adapter, from the drawer layout. This will list all the course data and the details can be expanded as per the priorities of the user. This will mock the Amazon app, information filter.
Course Chooser	Like the process of shopping, users would use wish list and checkout list to add the course they desired. This process would happen during course guider or users can choose courses by specific school and department.
Course Register	This feature will record the courses user register in this semester, and submit this to use server
Course Calendar	In a semester, students, especially undergraduate students who have many course need to take in one semester, may have many choices of their course calendar. It's about which course they want to choose and when they want to take them. This feature provides one more calendars for students to organize their arrangement.
Course Reminder	We know all courses in our school have limited seats for students, and thus many students would enter cases where their desired course didn't have any seats available. So this feature is designed for students to know how many course they are interesting will be not available.
DC Asker	This feature will collect and organize the way to deal with D-Clearance for different school of USC, by which students will

	know how to get their DC and when their DC is available.
FAQ Helper	There are a lot frequently asked questions by students, we can collect and organize them for users, by which students don't have to ask course assistants or search information on the internet.
App tutorial	This feature is designed as a tutorial for users to get familiar with our mobile app. It provides guidance pages when app starts, and guide users how to use this app step by step.

### 3.7 Google Services

Google APIs use the OAuth protocol for authentication and authorization. Google supports common OAuth scenarios such as those for web server, installed, and client-side applications. In this component we are using the Google OAuth protocol to authenticate the student. The students can sign in with the gmail account, so that the application get the details of the student and create the profile, as we are not granted the access to the user data. This is how the exchange of token works between the USC mobile application and the Google Servers.



### 4.0 Reuse and relationships to other products

Currently, the process of brainstorming is going on to identify the best possible approach for the given requirement. Therefore, there is no confirmed reuse of existing components. Although, the purpose of the application is to enhance the current web registration portal, there is no relationship to or reusing of its components.

### 5.0 Design decisions and trade-offs

Not Applicable at this stage

## 6.0 Pseudo-code for components

Not Applicable at this stage

## 7.0 Appendices (if any)

**Prototype:**

Login Screen:

