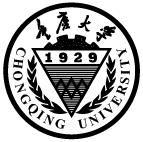
**Course Project Summary**



**Course Name: Mobile Application Development**

**Project Name: CQU Edu. Mobile Platform**

**Student Name: Wu Tongyu / 吴侗雨**

**Student ID: 20161673**

**Advisor: Dr. Feno**

**School of Big Data & Software Engineering**

**Chongqing University**

**March 24th, 2019**

**Content**

[**I.** **Introduction** 3](#_Toc4526719)

[**II.** **Software Functions** 4](#_Toc4526720)

[**2.1** **Login Activity** 4](#_Toc4526721)

[**2.2** **Main Frame Activity** 5](#_Toc4526722)

[**2.3** **Student Information Activity** 5](#_Toc4526723)

[**2.4** **Faculty Scheme Activity** 6](#_Toc4526724)

[**2.5** **Class Schedule Activity** 6](#_Toc4526725)

[**2.6** **Course Scores Activity** 7](#_Toc4526726)

[**2.7** **Score Comparison Activity** 8](#_Toc4526727)

[**III.** **Software Design** 10](#_Toc4526728)

[**3.1** **Drawable images design** 10](#_Toc4526729)

[**3.2** **Theme Color Design** 10](#_Toc4526730)

[**3.3** **Security Settings** 11](#_Toc4526731)

[**IV.** **Software Coding** 12](#_Toc4526732)

[**4.1** **Login Procedure** 12](#_Toc4526733)

[**4.2** **Information Check** 14](#_Toc4526734)

[**4.3** **Faculty Scheme** 21](#_Toc4526735)

[**4.4** **Class Schedule** 24](#_Toc4526736)

[**4.5** **Course Scores** 28](#_Toc4526737)

[**4.6** **Score Comparison** 30](#_Toc4526738)

1. **Introduction**

CQU Edu. Mobile Platform is an extension of existing CQU Educational System web page. It can do most of the searching work over the phone, which is excellent to deal with the inconvenience of logging into the website.

This document introduces the production of the software step by step. Also, it can be used as an instruction of using this application.

Part II describes some major functions of this software, as well as pointing out what kind of problem it has solved to improve user experience.

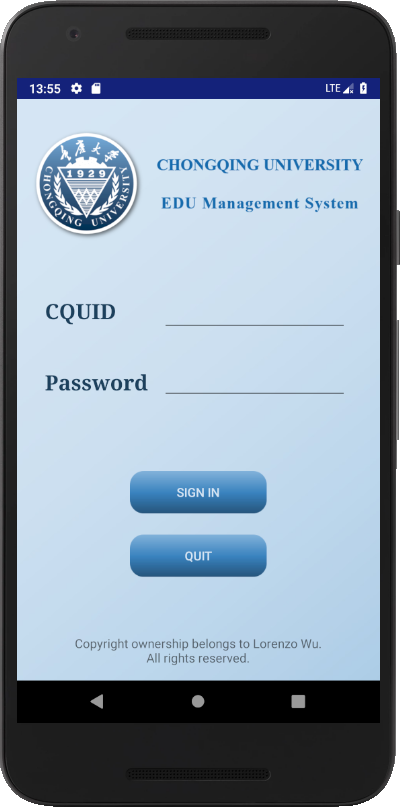
Part III introduces the process of designing the software, the layout design as well as the color design, for a better experience in user interface and graphic elements.

Part IV shows how do these functions work, which will show codes inside.

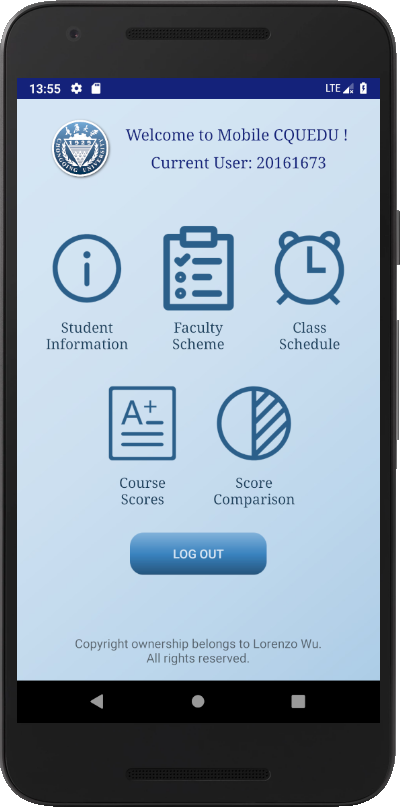
1. **Software Functions**

The application covers eight activities, each of which gives a specific function for student users to use.

* 1. **Login Activity**

The login activity is the main activity of this application, used for checking username and password. Only by logging in with the credentials enables a user to use all of the following activities.

This activity is made up of three major parts: on top is the title bar, showing the logo of Chongqing University and the name of the application for users. In the middle is the login form, consisting of two entries of student’s CQUID as well as his / her password. At the bottom is the buttons, which allows student users to sign in with the written form or just quit the application. Besides the three major parts, copyright information is set at the bottom of the screen, which is a typical way of showing by Chinese students.

* 1. **Main Frame Activity**

The main frame comes after a successful login. This activity serves as a main menu, showing all sub-functions of this application.

On the top of the screen is the signup information, showing the current user. It also contains welcome information. In the middle of the screen are six icons with a text under each icon, all of which leads to six activities of specific sub-functions. At the bottom is the log out button, which will cut out the current connection to the school server and return to the login activity.

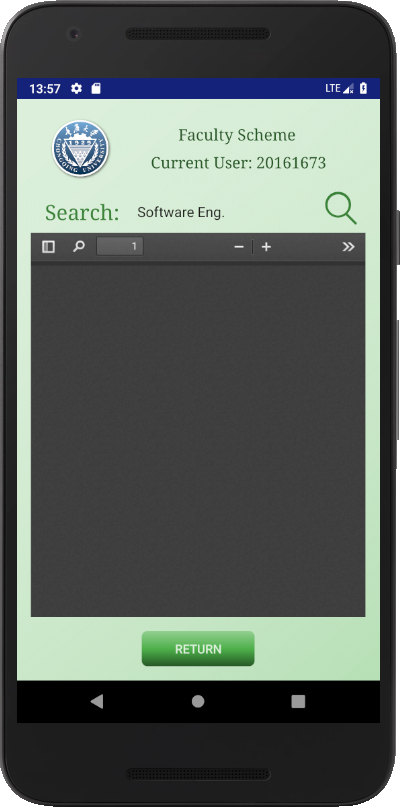
* 1. **Student Information Activity**

This activity shows the basic information of a student via the hidden interface presented by the web version of the education system.

This page gets data from specific AJAX page and parses the data within it into a table. This page can acquire students’ civil ID and judge the gender, date of birth as well as the origin. It also gets the faculty, grade and class from that page.

The search column on top enables users to check other student’s information by changing the sending form to the server.

The bottom shows a return button, which allows student users to go back to the main menu.

* 1. **Faculty Scheme Activity**

This activity engages a web view to acquire the faculty scheme in a format of PDF on the educational website. Since the server itself is not very stable, this function does not always work. This is still a testing function, or generally speaking, a beta version of activity.

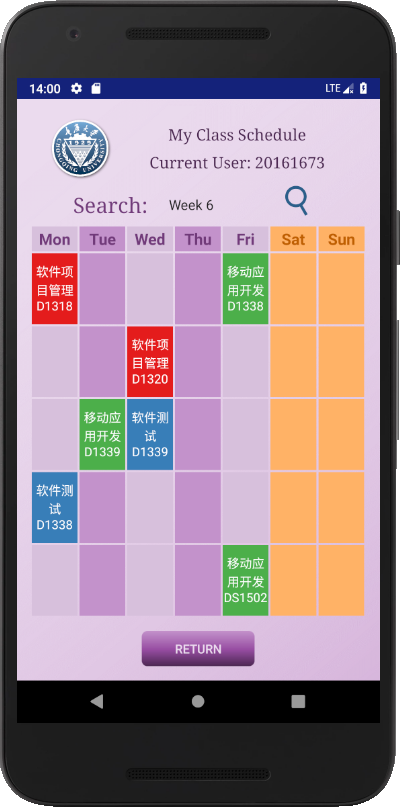
In this version, a user can find the scheme he needs according to the spinner search box. After clicking the search button, the software will generate information from the specific webpage and transform it into the format of PDF. The file will be shown with the assistance of Mozilla "pdf.js".

* 1. **Class Schedule Activity**

This activity shows the table for each student with a layout of a table covering seven days a week and five classes each day (two in the morning, two in the afternoon and one in the evening), showing the name of the class and the location of the class.

The program can move to the current week according to the system time. Different classes will be drawn in different colors, while the same class at different time will be drawn in the same color. Users can also search for classes in other weeks by using the spinner and the search icon.

The images of this activity is in the next page.

* 1. **Course Scores Activity**

This activity displays the information of scores of all the courses one student has learnt over a single semester.

User can change the semester needs to be check by using the spinner on the top of the screen and clicking on the "Check" button. The user is able to see the score of each course of that semester, along with the name of the score and how many credits are gained by passing that course.

If there are so many courses that it not able to be put within a screen, there will be a scroller.

The images of this activity are shown in the next page.

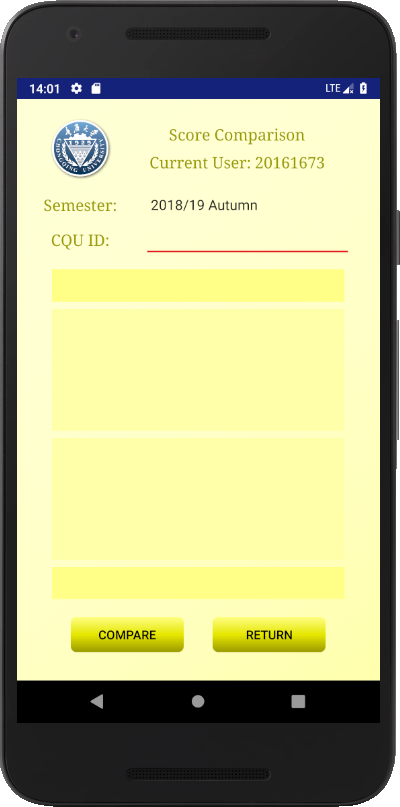
 

* 1. **Score Comparison Activity**

This activity merged the function of calculating Grade Point Average and comparing your score with others in a single semester.

A student user needs to input the semester that he needs and another CQU ID of the student’s score he wants to know. When the user presses the "compare" button, the software will get the scores from a hidden interface in CQU Edu System. Then the program will calculate the GPA automatically and show it on the screen.

The images of this activity are shown in the next page.

1. **Software Design**
   1. **Drawable images design**

Most of the images of this application are drawn via Microsoft PowerPoint and Adobe Photoshop.

1. Background Photos

The background photos use a diagonal layout by using the same hue of different brightness. The design of colors is shown in part 3.2.

1. Buttons

The button photos have two conditions: the original condition and the pressed-down condition. When a button is pressed, it will automatically change into the pressed-down condition. The background photo of buttons uses a horizontal layout by using the same hue of different brightness. The design of colors is shown in part 3.2.

1. CQU Logos

The logos of Chongqing University are acquired via the public resources given on Github.com. In this application, we use the blue fading theme.

1. Functional Icons

This application uses icons from iconfont.cn supported by Aliyun. The colors used in the icon follows the theme set in part 3.2.

* 1. **Theme Color Design**

This application uses 6 major theme color hues. Each color hue has six colors of different brightness. The six theme color hues are:

Theme 1: Color red, RGB E41A1C;

Theme 2: Color blue, RGB 377EB8;

Theme 3: Color green, RGB 4DAF4A;

Theme 4: Color purple, RGB 984EA3;

Theme 5: Color orange, RGB FF7F00;

Theme 6: Color yellow, RGB FFFF33;

For each color, there are 3 light auxiliary colors and 2 dark auxiliary colors. For example, theme 1 has 6 colors in total:

Color 1 Neutral: RGB E41A1C;

Color 1-Dark 1: RGB AB1315;

Color 1-Dark 2: RGB 720C0D;

Color 1-Light 1: RGB EF7576;

Color 1-Light 2: RGB F5A3A4;

Color 1-Light 3: RGB FAD1D1.

* 1. **Security Settings**

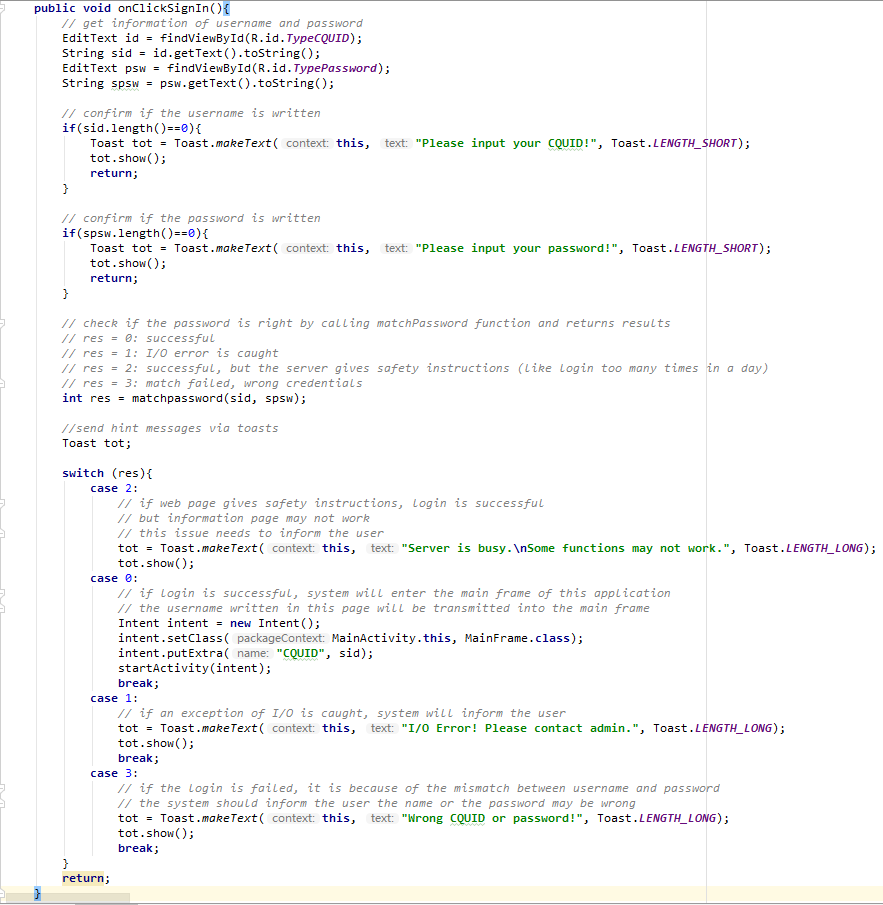
Since Android has some restrictions in system access since version 26, there should be some security configs to solve access problem. This application needs to use some permissions, such as Internet access, external writing access, downloading, read phone state and so on.

In file "AndroidManifest.xml", we can add permissions into the app. For security issues, we need to set in a xml file. We name it "security.xml" and allow the app to have clear text traffic in network security configs.

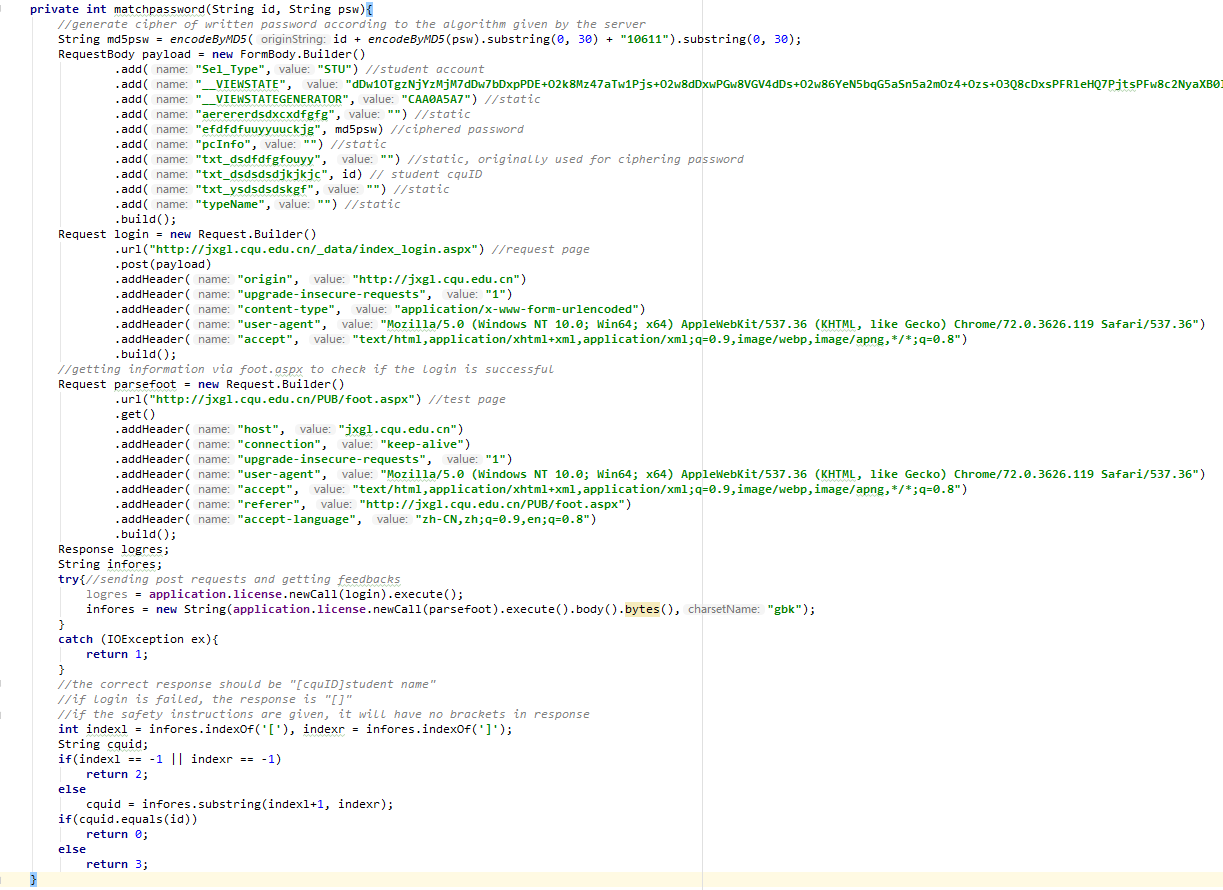
1. **Software Coding**
   1. **Login Procedure**

This application sends login information to school server (<http://jxgl.cqu.edu.cn>), then the school server will deal with the message to check if it matches the username – password pair within the server. If the login is successful, the server will allow the user to use the functions of the website, or none of the pages in the following activities will work.

When the sign in button is clicked, the program will run the "onClickSignIn" function, shown below.

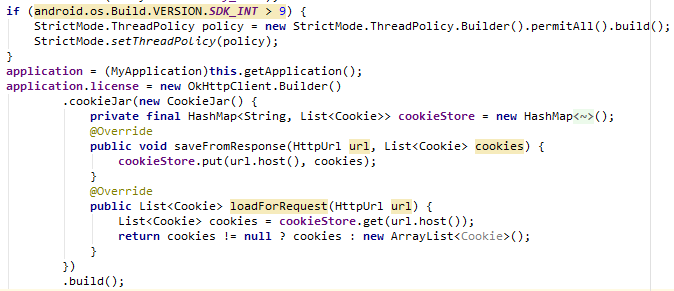


Function matchPassword() is a function checking whether the login is successful, shown below:



In case of lower versions of Android, strict mode should be turned off.

In order to save cookies, we need to write a customized application class including an OKHttpClient. In this activity, we need to build the OKHttpClient and set a customize cookie jar to avoid self-deletion of cookies, containing login info, from the application. The build procedure is in the snapshot below.



MD5 coding is embedded into the application.



* 1. **Information Check**

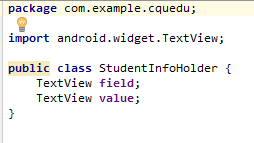
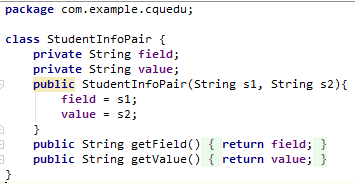
On the website, this interface is hidden by the server, students cannot enter this page by clicking on functions on the website. However, by analyzing the source code, we are able to find this page. The page looks like this:



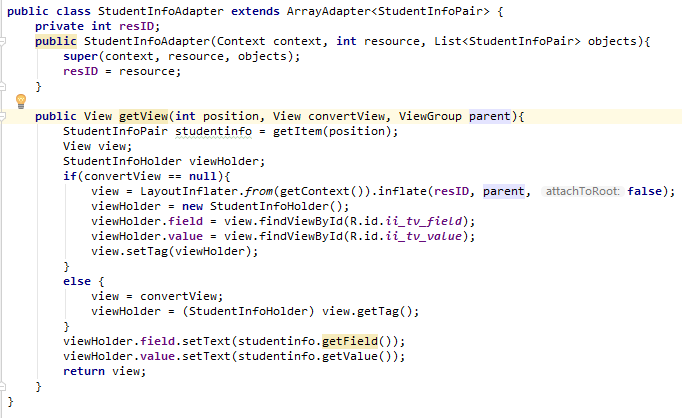
As we see, in this page we cannot find the origin of the student, as the date of birth and gender is not always correct.

In the application, the student info activity contains a customized list view, showing a pair of fields – values. Three new classes need to be set up to show the customized list view.

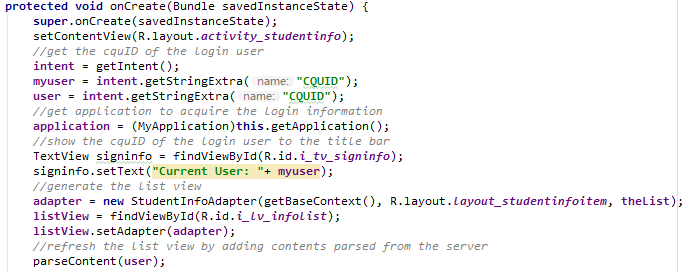
Holder and Info Pair classes:

Adapter class:



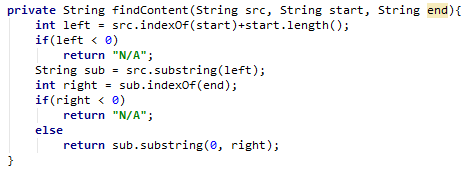
When the page is loaded, it will automatically parse student info of the user logged in.



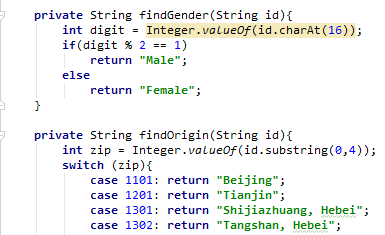
Function parseContent gets name, student id card, faculty, class and grade from the website.

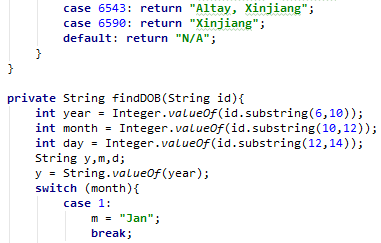


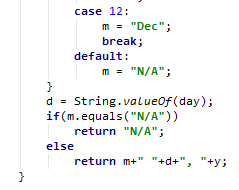
Find content function gets substring with the beginning and ending string as a mark.



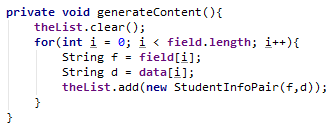
The ID card can be used to get the gender, DOB and origin of a student. The parse from zip code to city is made by a huge list, hidden in this document.







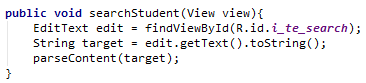
The function to write info in to the list view is shown below.



Since the faculty name is written in Chinese, it needs to be translated into English.



When the user inputs the new CQU ID and click on “Search” button, the system will go to find the information according to the input ID.

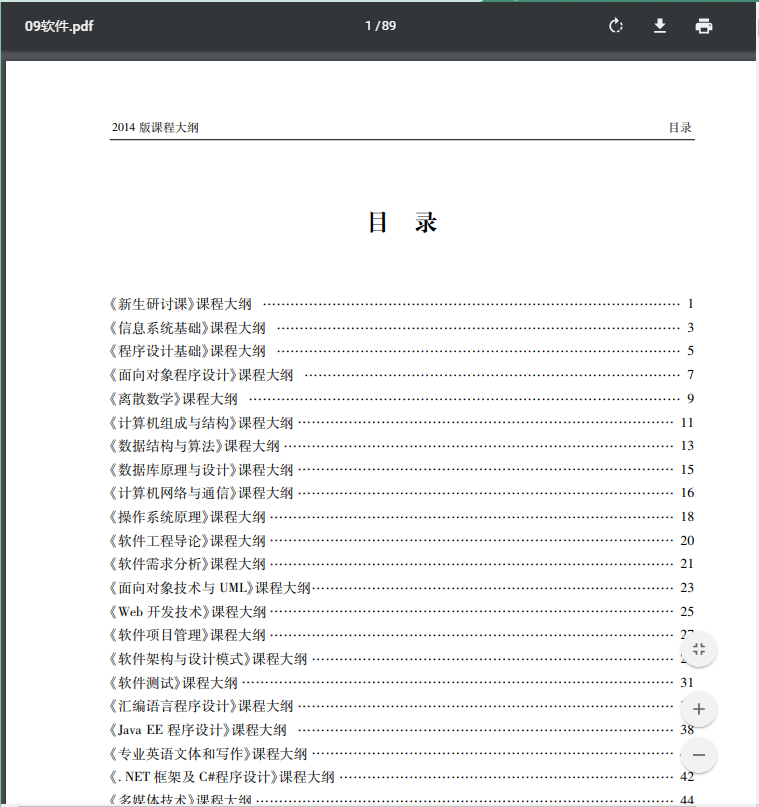


The page on the phone looks like this:

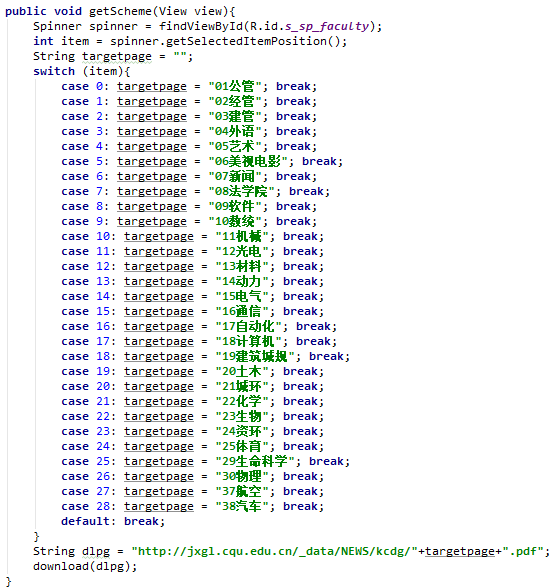


* 1. **Faculty Scheme**

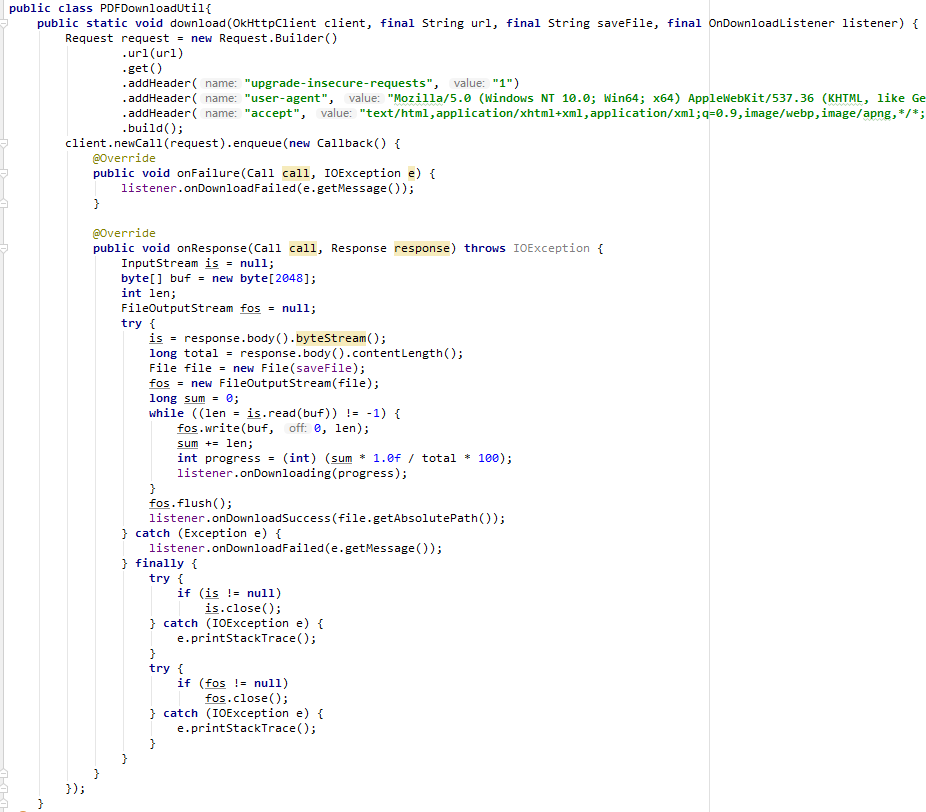
On a website, this page is found like this:

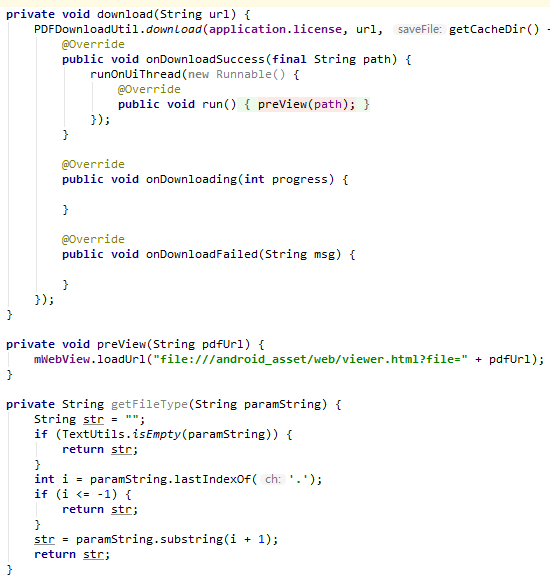


In the application, the page confirms the needed information of a faculty and download it from the target page. Then it is shown in a web view assisted by PDF.js supported by Mozilla.



In this activity, we wrote a customized download utility. Then we do the download by calling it.





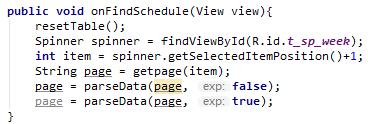
* 1. **Class Schedule**

On a website, the schedule should be displayed like this:

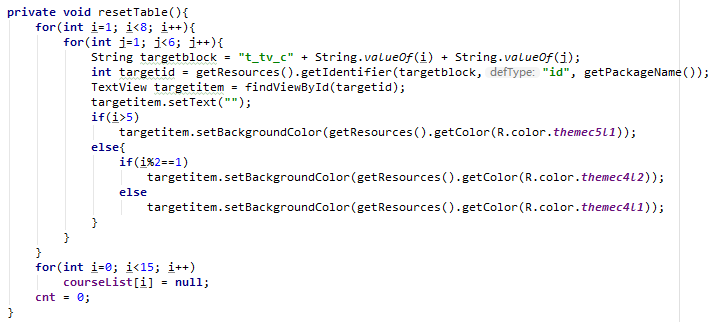


This web page is shown in a form which is not quite clear for students to understand the classes they are about to take. So the activity has to change the form into a table, as we always used as a class table.

The page gets class list from the server and parse it into a table form. Main function is on find schedule function.



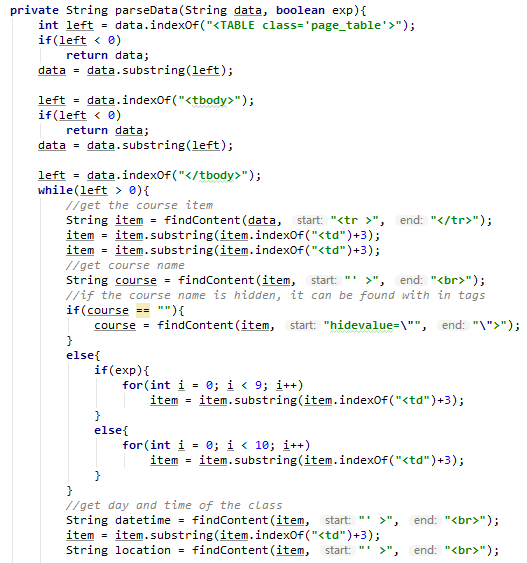
The reset table makes the initialization of the table, make sure the info get before won’t affect further search.

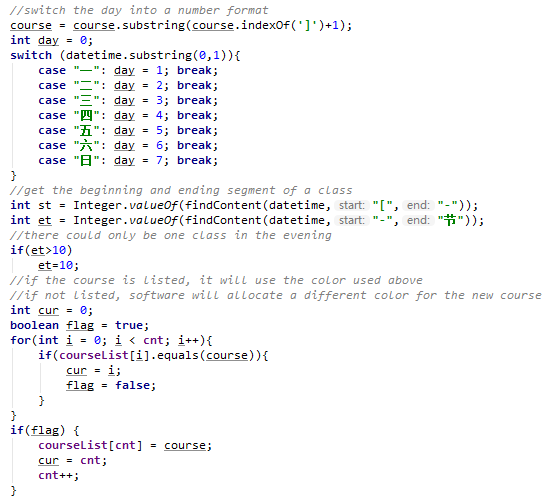


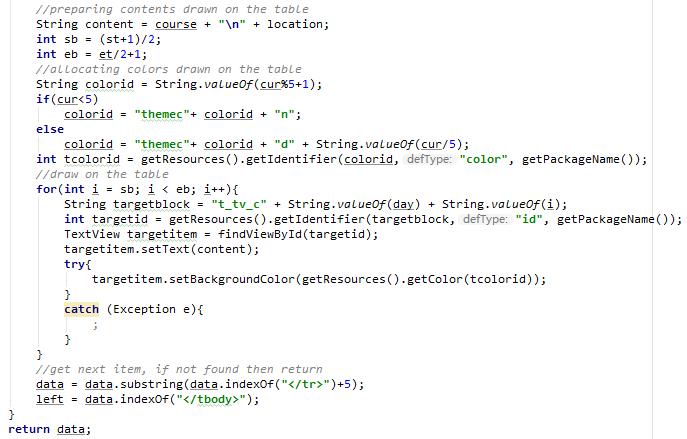
The get page function grabs the table from the website.



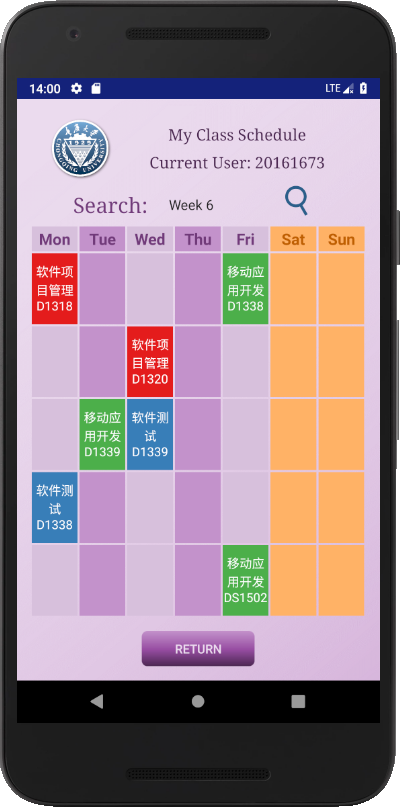
The parse data function analyzes contents of it and put it onto the table according to time order, drawn in different colors for different courses.







The parsed course table looks like this:



* 1. **Course Scores**

On a website, the scores should be displayed like this:

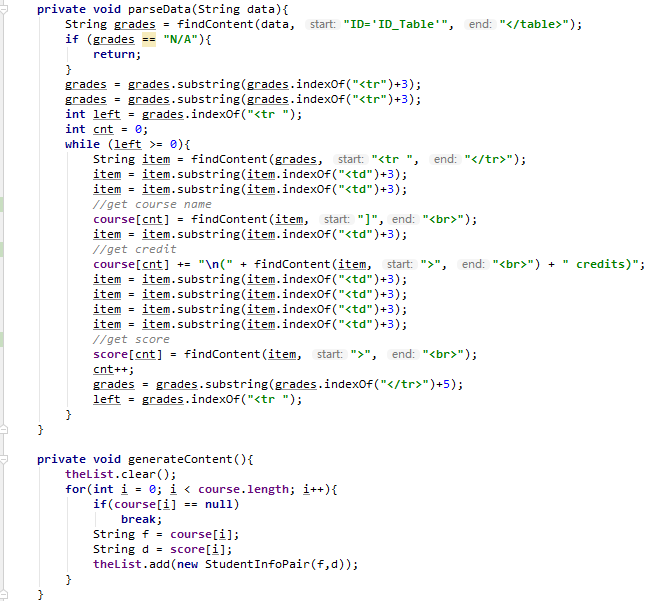


This page is clear enough, we only need to transform it on the phone.

In this activity, we will use the list view set in part 4.2 again. Then the process goes similarly as what we have done in part 4.4.



The functions parse data has some minor changes, while the generate content function has no changes. The changed parse data function is shown below.



This activity looks like this:



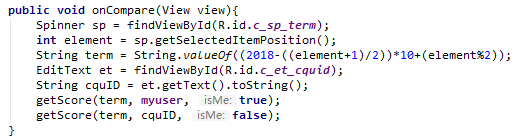
* 1. **Score Comparison**

On a website, students are never able to check other students’ scores, not to mention comparing with them. However, according to the source code, there is a place where I can find other students’ scores. In this example, we are going to use CQU ID 20161684 as an example. The result looks like this (private information has been protected).



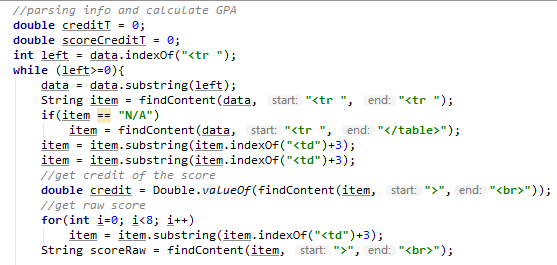
As a student, we do not need to know the actual score of every course of other students (actually it is bad to see because of damage of privacy). A better solution is to get the grade point average (GPA) instead of several scores. This does both calculation and comparison of GPA, which helps students a lot

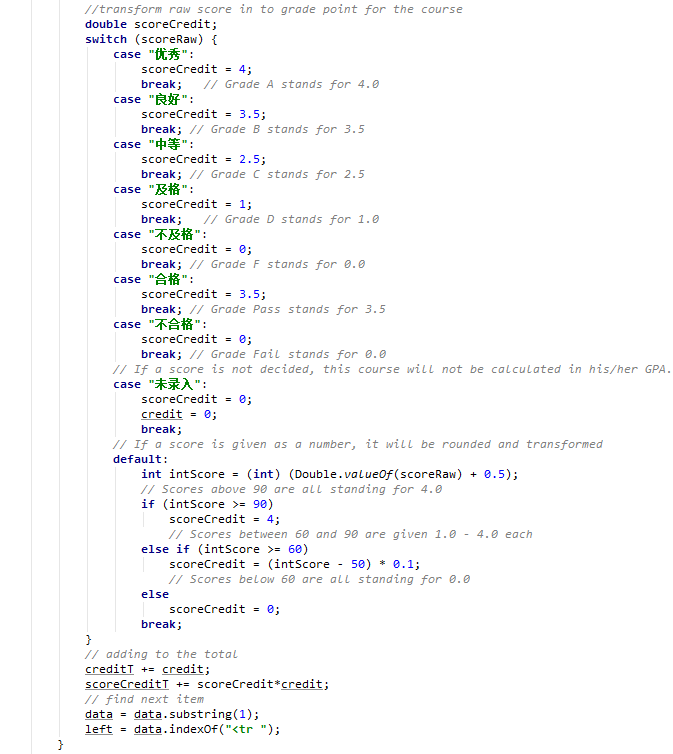
On compare function is the entrance function, generating the form to be submitted to the server.

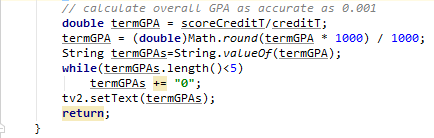


Then is the parsing process, goes like a normal one.









The result looks like this:

