COURSE REVIEW SYSTEM

By

SHAILAJA MADDALA

B. TECH, JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD, INDIA, 2013

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2015

Approved by:

Major Professor
DR. MITCHELL NEILSEN

Abstract

The Course Review System (CRS) application reviews courses and their feedbacks, thereby providing a platform to learn about the courses being offered at Kansas State University (KSU) before enrollment. This can help current and also prospective students who are planning to enroll in courses at KSU to form an idea about the courses being offered at KSU. For the CRS application graduate and undergraduate level courses from 16 departments at KSU were included. Users can rate a course on a scale of 1 to 5 in three categories namely: Difficulty, Grading, and Learning Curve. Statistical analysis was used to display the top 10 courses in each department for each rating category. A recommendation feature which recommends courses to users based on the courses they are currently viewing was added to provide users with recommendations. Users could post their reviews and comments in the comments section. In addition to this we also have a 'Questions and Answers' section allowed users to ask questions and any interested user could answer them in that forum.

Dimensional Research conducted a survey about the impact of opting courses via online reviews, in which 88% of respondents were influenced by online course reviews when selecting an online course [17]. In addition to the survey however, rational thinking suggests that obtaining an idea about a course involves reading through the experiences of actual users before deciding to select a course. Therefore, the proliferation of various review websites, including software application reviews, plays a major role.

Reviews and ratings of a course provides users with an idea about the course they intend to take up and also helps in effectively planning out coursework for the semester. As users are looking for information to help them choose the most suitable course for their requirements course reviews are playing a larger role than they were in the past. It also helps students to make smart choices in laying out a flow chart for their program. The CRS application was a perfect platform for students to know everything they need to know about courses before they enroll.

Table of Contents

List of Figures

List of Tables

Acknowledgement

I wish to express my deepest gratitude and warmest appreciation to my major professor **Dr. Mitchell Neilsen** who availed me an opportunity to utilize and apply gained theoretical knowledge practically and enhance my knowledge in the field of Android Application Development. He always offered constant support and encouragement through his valuable suggestions and feedback during every phase of this project. I would like to thank **Dr. Torben Amtoft** for being a member of my committee and also for his constant support throughout my Master's program by offering valuable knowledge as well as an opportunity to explore new fields such as Probabilistic Programming. I would also like to thank **Dr. Doina Caragea** for serving on my committee.

I am honored to have the privilege of taking courses under the professors of the CIS department who helped me dig a bit deeper into the depths of Computer Science with this Master's program. I would also like to thank the members of the CIS department staff who were helpful and supporting at every step during my Master's program at K-State. Special thanks to **Dr. Gurdip Singh** and **Dr. Scott Deloach** who gave me an opportunity to be a Graduate Teaching Assistant, which was a new and learning experience for me.

I would like to thank K-State for being a home away from home and for helping me to explore the field of Computer Science by providing me with all the amenities and resources.

I would also thank my friends who have been a constant source of support and love. I would finally like to thank my parents, **Mr. Venkateswara Rao Maddala** and **Mrs. Vanaja Maddala,** for being my biggest strength at every stage of my life and for giving me the wings to fly and saying that sky is the limit.

Chapter 1 - Introduction

Course Review System currently is an essential and crucial application because according to the Dimensional Research regarding the impact of opting courses, 78% of people responded that course reviews are very important factors [17]. When considering training courses people tend to choose those courses that had reviews, received more views and feedback than those that don't have reviews. An additional 20% of users stated that they sometimes considered reviews before they decided which training course was most suitable for them, and only 5% claimed that reviews were not important to them. Online reviewing has revolutionized online marketing since the Internet became a household convenience. By allowing the application to have active, positive participation from consumers and giving consumers an opportunity to create a relationship with those businesses, so it pays to keep in control of it.

Reviews and ratings of a course provides users with an idea about the course they wish to take up and also helps them in effectively planning out their coursework for the semester. As users are looking for information to help them choose the most suitable course for their requirements course reviews are playing a larger role than they were in the past. Also as a student I would prefer to get to know more about a course before taking up so that I can be prepared accordingly. It also helps students to make smart choices in laying out a flow chart for their program. The CRS application is a perfect platform for students to know everything they need to know about courses before they enroll

Dimensional Research conducted a survey about the impact of reviews on opting courses, in which 88% of respondents were influenced by online Course reviews. But even without the survey, rational thinking suggests that getting the most value includes reading through the experiences of actual users before plunging head on to opt. Hence the proliferation of various review websites, including software application review plays a major role.

This application helps in reviewing about courses and also allows users to post their respective feedbacks. Review-compare sites are becoming more important to online consumers, who claim that reviews give them a real sense of try before you buy and a valuable insight into a product before they make a financial commitment. This is also true in the professional training courses world. As users are looking for information to help them choose the most suitable course for their requirements, course reviews are playing a larger role than they were in the past.

The Course Review System (CRS) application provides users with an interactive platform to rate courses they have already taken and also post their reviews and feedbacks about those courses which can help current and prospective students who are planning to enroll in those courses. This can help current and also prospective students to form an idea about the courses being offered at K-State.

The idea of our application Course Review System (CRS) was to provide a single platform for prospective students to gain knowledge of previous student's experiences in order to effectively determine their coursework for a semester.

Course Review System uses App42 for backend as a service for server and App42 Cloud Service as backend and Android Software Development Kit as a front end for clients.

Chapter 2 - Motivation

Motivation for this application stemmed from personal experience in an MS program. Prior to enrollment every semester the author talked to fellow students to determine if they had taken anticipated courses and learn of their experiences within those courses. I would want to know how their experience was taking up that course. Many fellow students reciprocated with similar questions. In order to ascertain difficulty levels of courses thereby allowing accurate judgment when deciding how many credits to take and logical potential pairings to ensure academic success. Course decisions were typically made by identifying difficult, moderate and easy courses and then selecting two hard course and one course within a semester. However the realization was made that if a platform existed in which all course information could be conveniently accessed students would be able to more effectively plan their semester course work. The determination was made that Android would be most advantageous for an application because Android is the most commonly used operating system on mobiles. After discussion with the author's professor the CRS application was created including a recommendation feature.

Chapter 3 - Technical Details of Backend Server (APP42)

3.1 Introduction to APP42

App42 Cloud Platform is a flagship product, on which all other products are built. App42 has includes an ecosystem approach in order to provide mobile, web, social, TV, and gaming app developers an all-inclusive product for all their cloud needs [5].

Whether they are very simple or complex, reliable and robust apps can be developed in a very short time using App42 as a backend service. App42 Platform as a Service was built over the App42 Platform. App42 Paas provides a complete transparency to developers so that developers do not have to manage any servers, software installations or hardware upgrades [5].

3.2 App42 screenshots

3.2.1 App Dashboard

The App42 dashboard contains the app manager which has a list of options to choose from as shown in the screenshot below. It has a Technical Service Manager which helps in managing the collections.

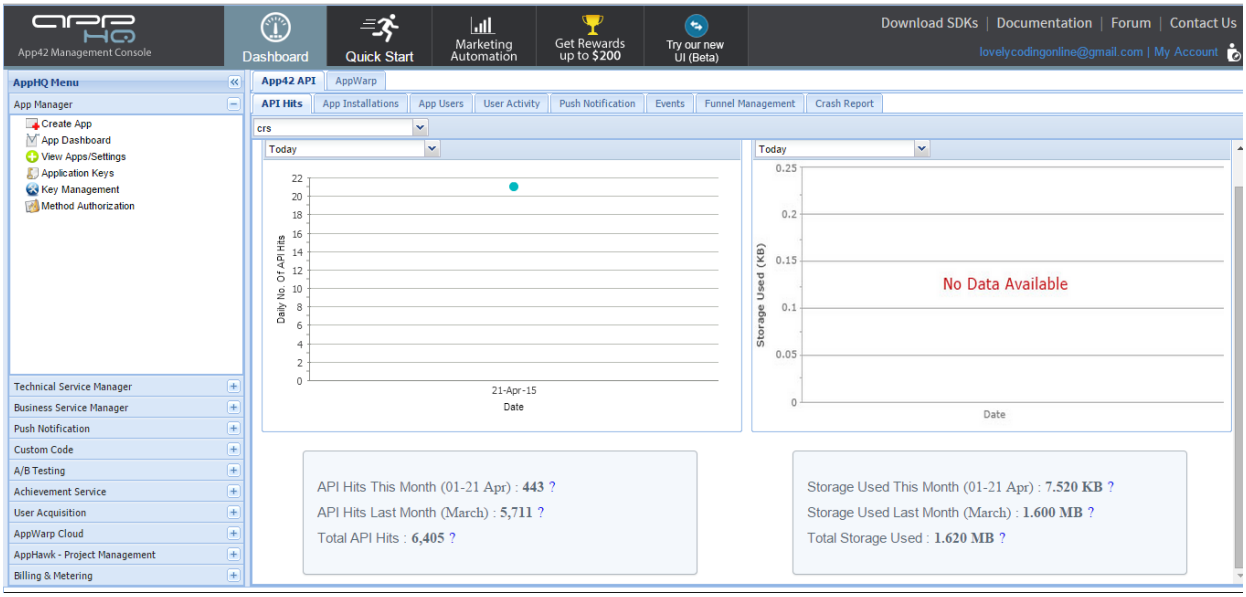

Figure 3.1 App Dashboard

3.2.2 Email Services: Reset User Password Template

In the login page of the application we provide forgot password link to help users reset the password in case they forgot their password. The screen shot in Figure 3.2 below demonstrates how that can be done in App42.

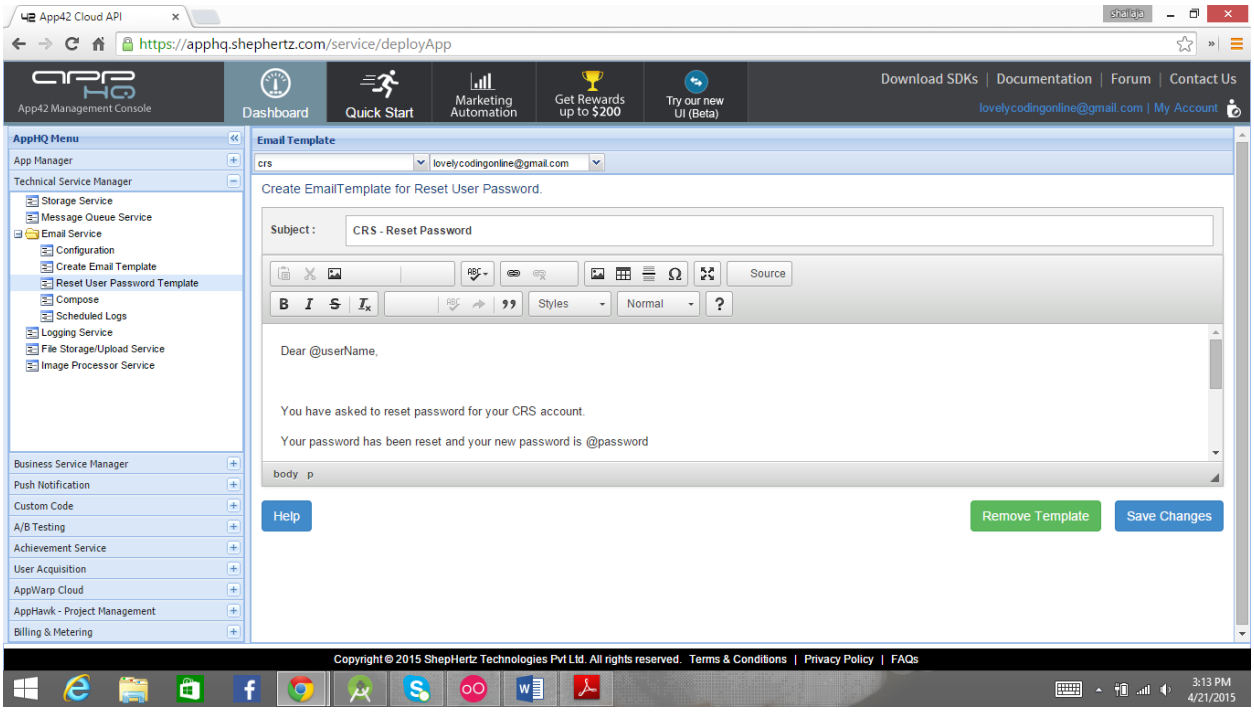

Figure 3.2 Email Services User Password Reset Template

3.3 Database Collections used in CRS

3.3.1 answer_data

This collection stores the answers given to the questions in the Questions & Answers section. It contains two components:

a. text (string)

b. code (string)

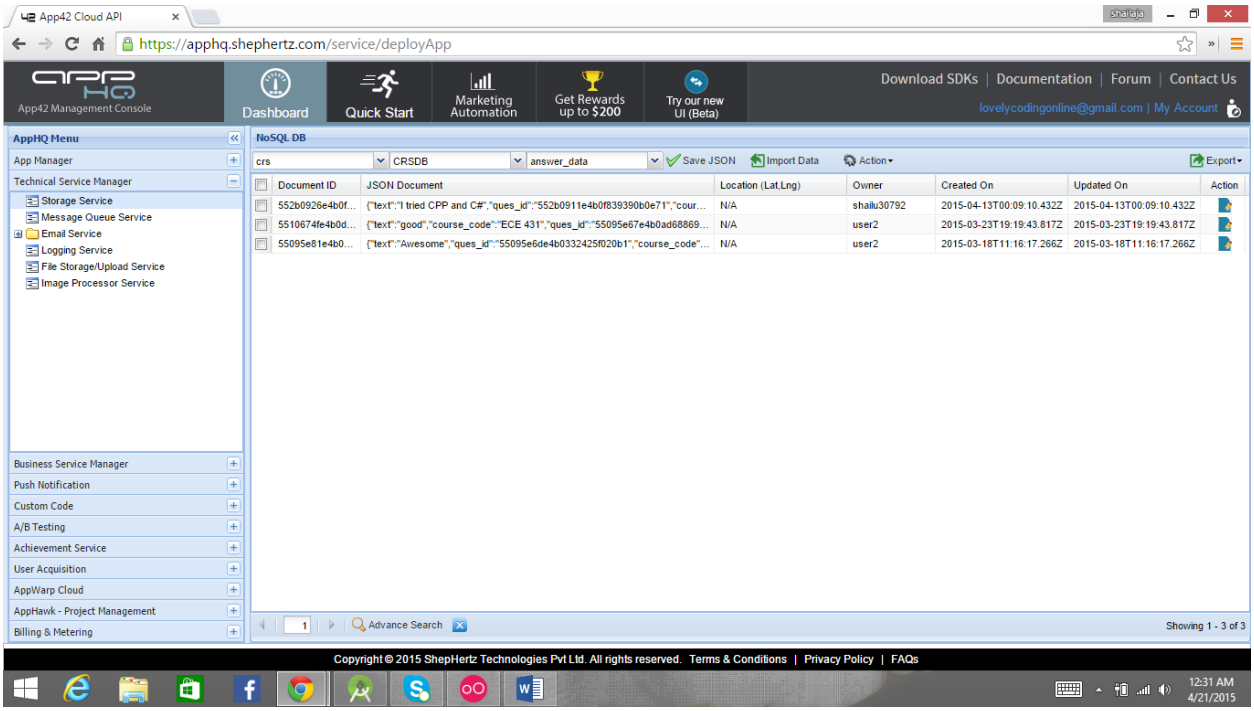

Figure 3.3 anwer_data Collection Screenshot

3.3.2 comments_data

This collection contains the comments which are submitted in the course rating page by the users. It contains two components:

a. text (character)

b. code (string)

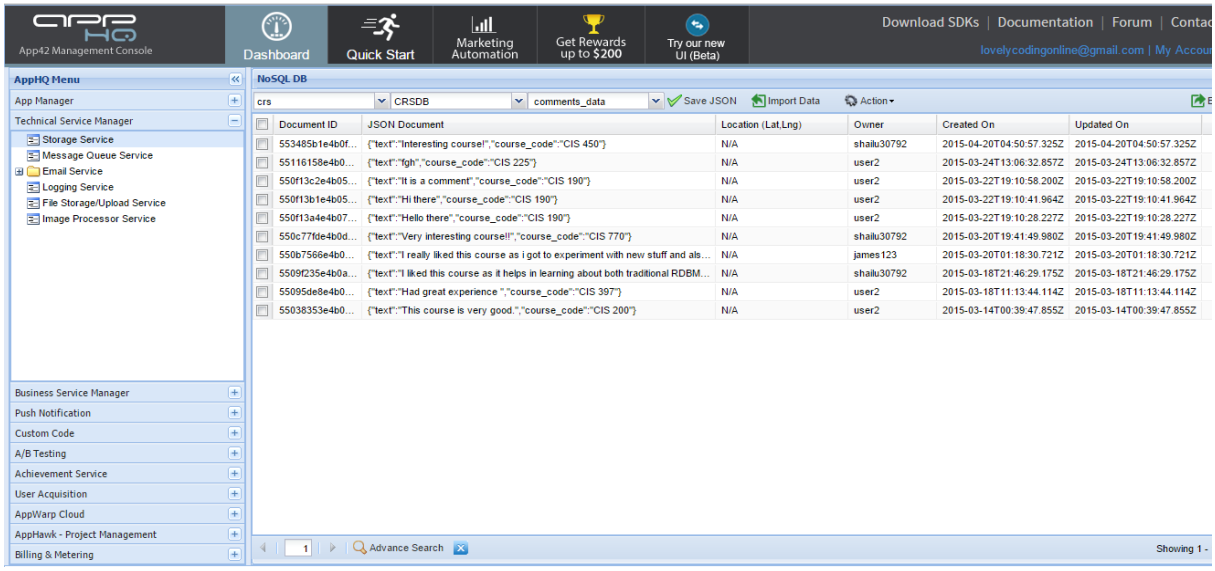

Figure 3.4 comments_data Collection Screenshot

3.3.3 course_data

This collection contains the names of courses from all the departments listed in the application. It contains the components:

a. name (string): Name of the course
b. dept_code (string): Code of the department to which the course belongs
c. code(integer): Code number of the particular course

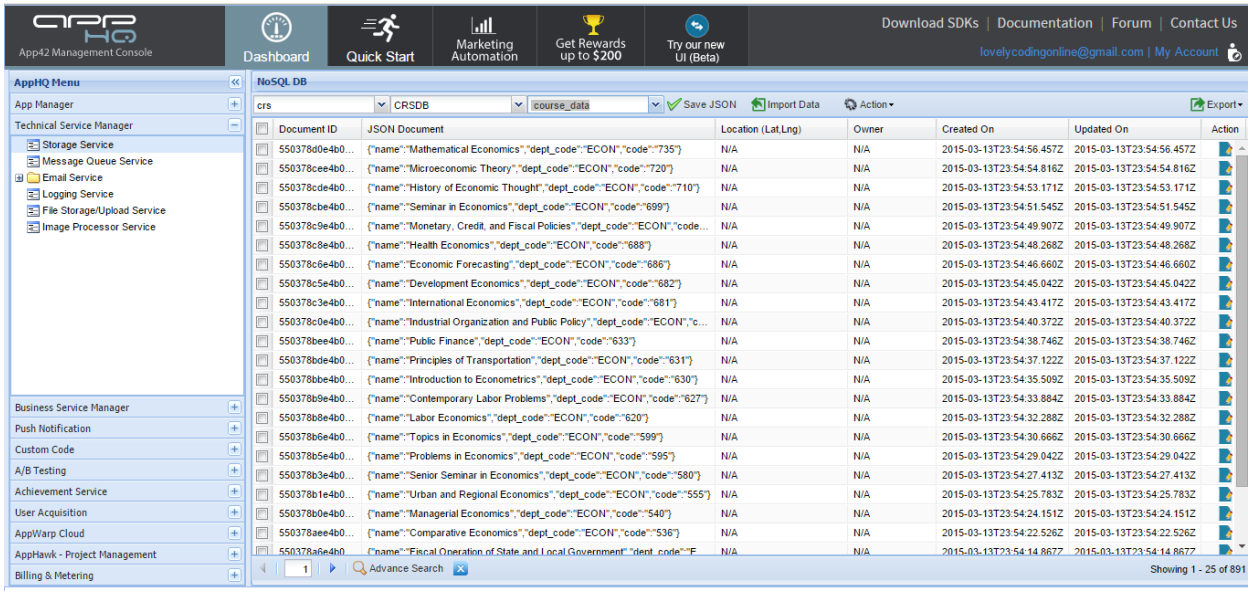

Figure 3.5 course_data Collection Screenshot

3.3.4 department_data

This collection contains the list of departments which are listed in the application. It has the following components:

a. name (string): Name of the department
b. code (string): Code of the particular department

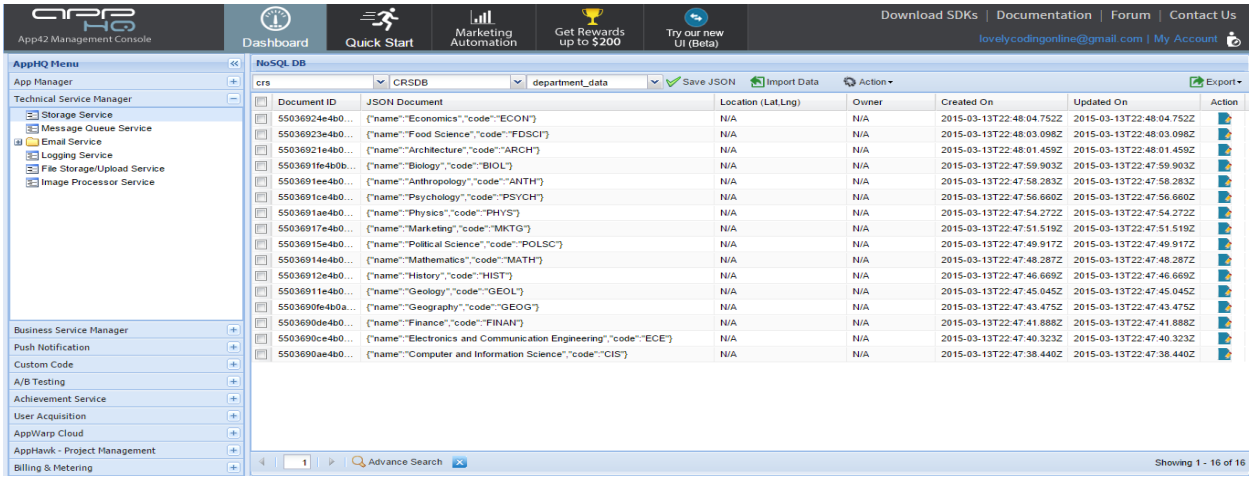

Figure 3.6 department_data Collection Screenshot

3.3.5 question_data

This collection has the questions which are submitted by the users in the Questions & Answers. It has the following components:

a. text(string): Stores the question that is submitted by the user in the Questions and Answers section
b. Course_code(string): This represents the department code and course code of the particular course

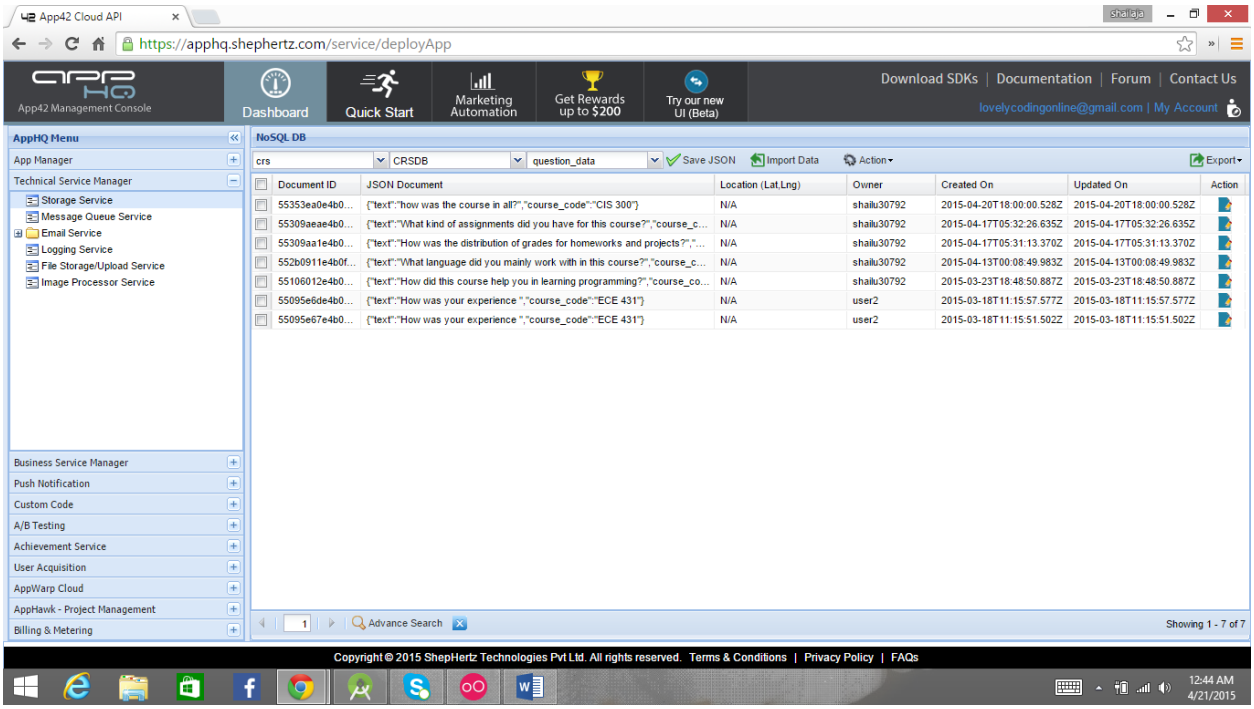

Figure 3.7 question_data Collection Screenshot

3.3.6 rating_average_data collection

This collection contains the average rating for each of the three categories for a particular course which is calculated from the ratings provided by the users. It contains the following components:

a. course_code (string): This field represents the course code number of that particular course
b. dept_code (character): This represents the code of the department
c. course_code (string): This field represents the code of the course
d. number(integer): This field represents the number of ratings received by a particular course
e. all_rating(float): This field represents the combined sum of the three ratings for that particular course
f. difficulty_rating(float): This field represents the average difficulty rating for the course referring to the course_code
g. learning_rating(float): This field represents the average learning rating for the course
h. grading_rating(float): This field represents the average grading rating for the course

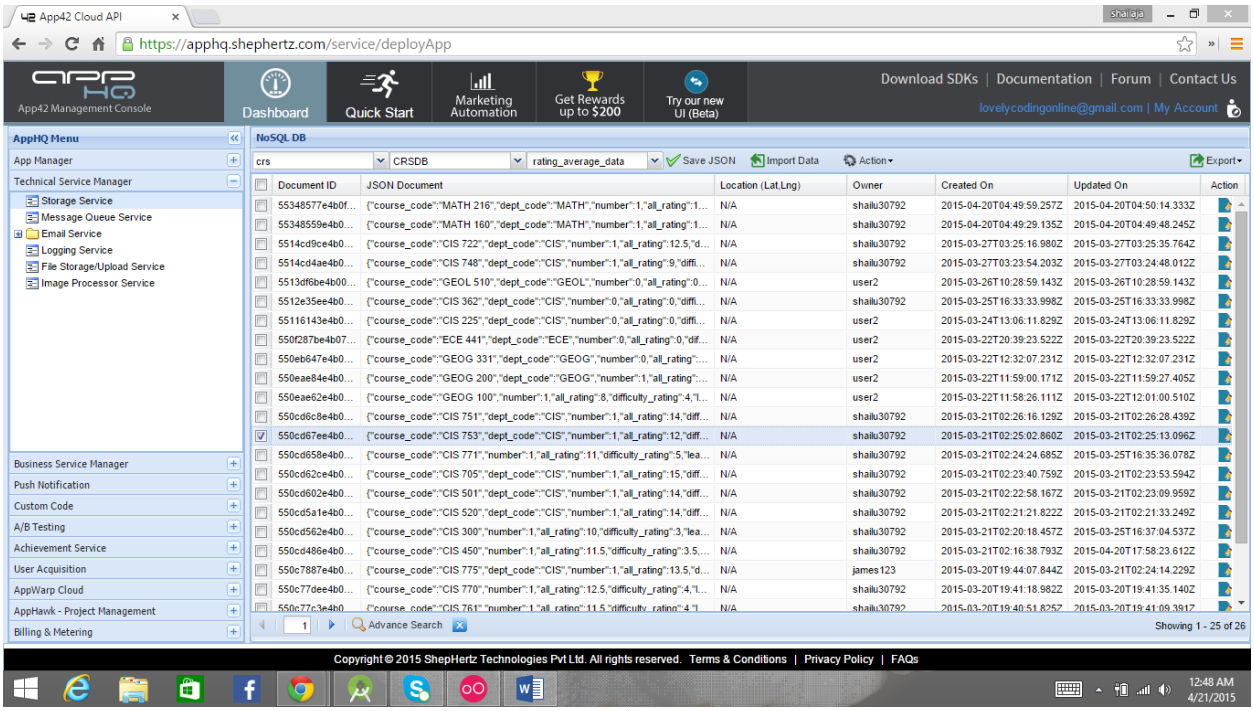

Figure 3.8 rating_average_data collection

3.3.7 rating_data

This collection contains the information about the individual ratings given to each course by an authenticated user in each of the three categories. It contains the following components:

a. time(string): It contains information of when the course was taken up by a particular user
b. course_code(string): The course code of the particular code
c. difficulty_rating (float): The difficulty rating given by a particular user for this particular course is represented by this field
d. learning_rating((float): The learning rating given by a particular user for this particular course is represented by this field
e. grading_rating(float): The grading rating given by a particular user for this particular course is represented by this field

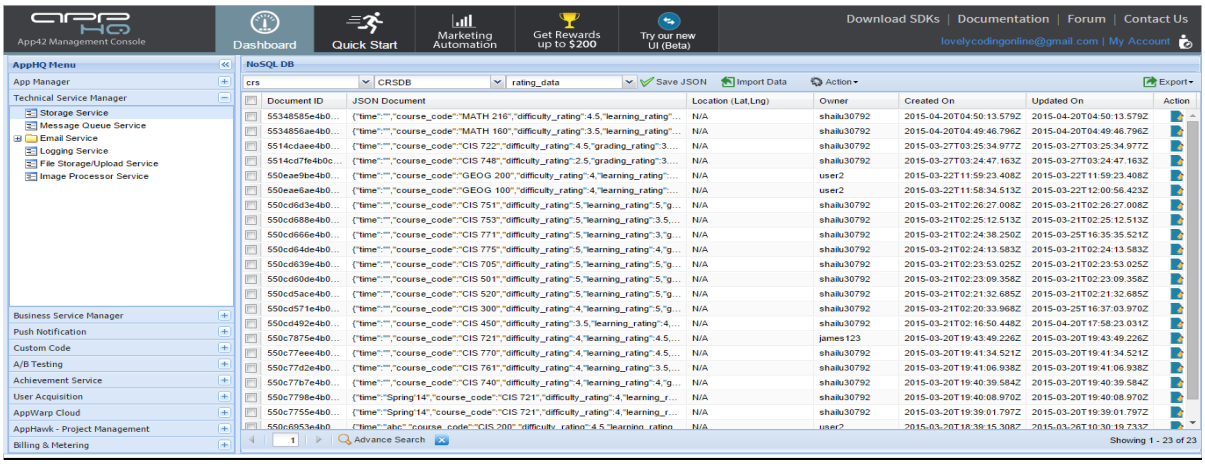

Figure 3.9 rating_data collection

3.3.8 user_data

This collection contains information about the user. It has a single component named student_type (integer) which contains information about student being an undergraduate, graduate or Ph. D student. Each category is represented by a number. 1 corresponds to undergraduate, 2 corresponds to graduate and 3 corresponds to Ph. D.

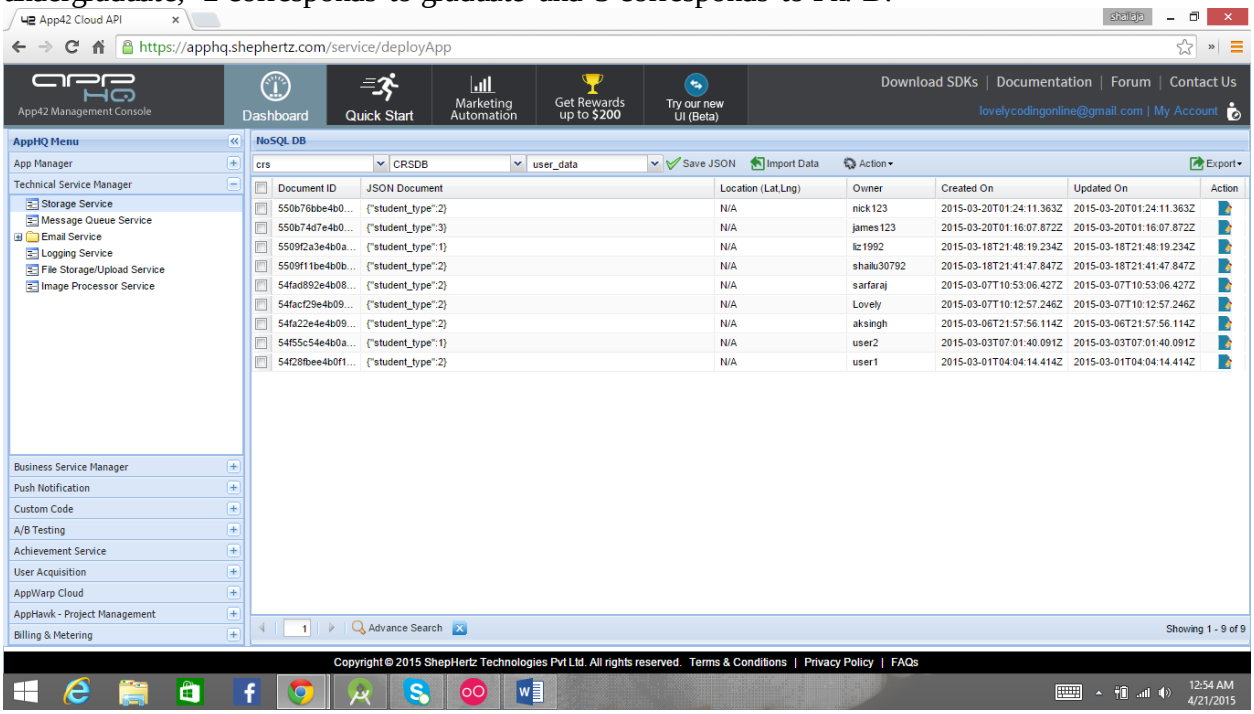

Figure 3.10 user_data collection

Chapter 4 - Android SDK (Front End)

4.1 Introduction to Android

Android, a mobile operating system which is based on the Linux kernel, was developed by Google [3]. Android is designed primarily for mobile devices, with touch screens, such as smartphones and tablet computers with a user interface based on direct manipulation. The OS is also used in game consoles, digital cameras, regular PCs, and other electronics despite being primarily designed for touchscreen input.

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android Software Development Kit (SDK), but other development environments are also available.

As of July 2013, more than one million applications have been developed for Android, with over 25 billion downloads. A June 2011 research indicated that over 67% of mobile developers used the platform, at the time of publication. In Q2 2012, approximately 105 million units of Android smartphones were shipped thereby acquiring a total share of 68% in overall smartphones sales in Q2 2012 [3].

4.2 Background

Android Operating System is a Linux based Operating System which can be used on smartphones and tablets. It is initially developed by Android Inc. and is later acquired by Google. It is an open source development platform powered by a modified Linux 2.6 Kernel [3].

4.3 Android Architecture

Layered Android architecture can be seen in the Figure 4.1 where the modified Linux 2.6 Kernel acts as the Hardware Abstraction Layer (HAL), provides the memory management, device drivers, process management, and networking functionalities.

The libraries layer is interfaced by Java and it contains the Android Specific Bionic libc which is the Android's own c library developed by the Android Community.

Next, the Android Runtime Layer which contains the Core Libraries and the Dalvik Virtual Machine (DVM). .

The next layer is the Android Application Framework which is responsible for the application life cycle.

The top layer contains applications such as Calculator, Clock, and Calendar etc.

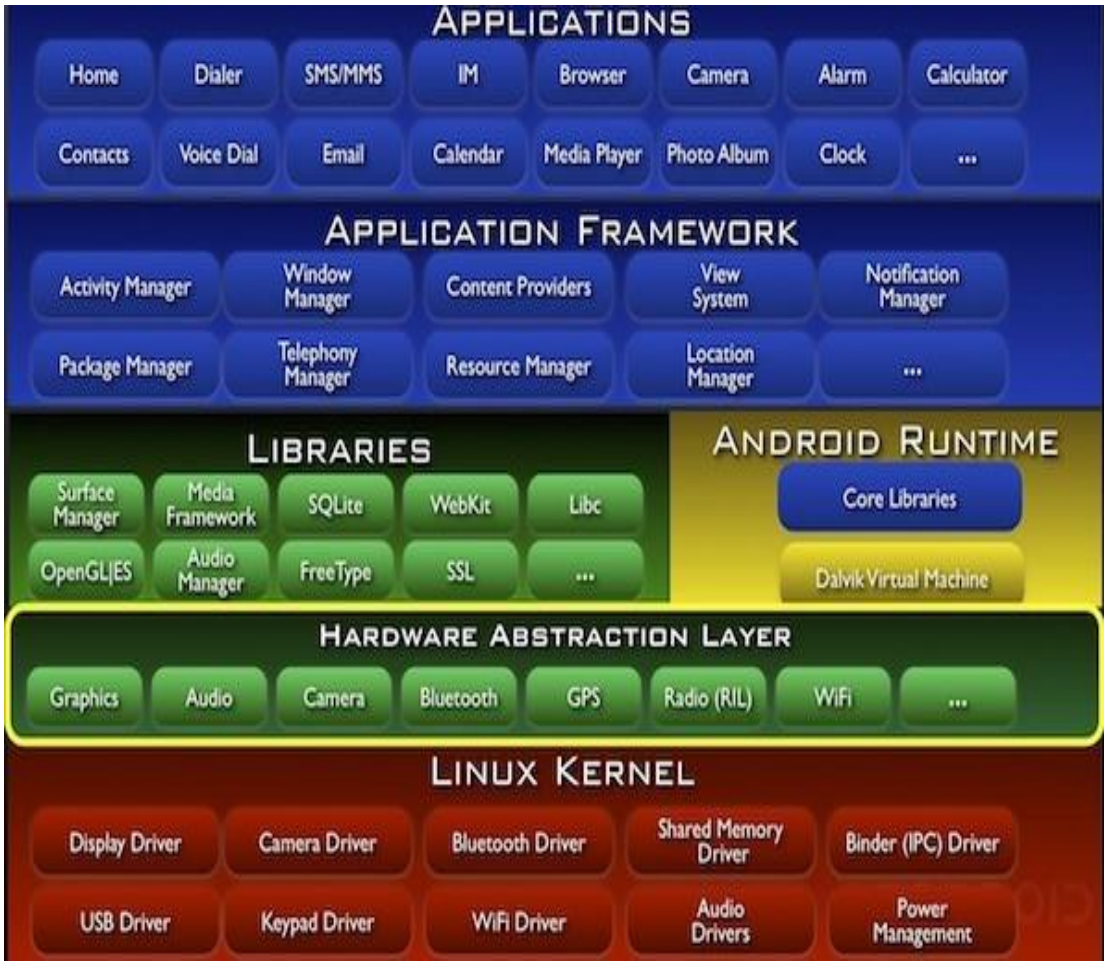

Figure 4.1 Android Layered Architecture [16]

4.4 Data Storage

Android provides various options to save persistent data.

- Shared Preferences - Stores private primitive data in the form of Key-Value pairs.
- Internal Storage - Stores private data directly on the device.
- External Storage - Stores public data on a shared external storage.
- SQLite Database – Stores structured data on a private database.
- Network Connection – Stores data on the web with our own network server.

4.5 Libraries Used in CRS

1) com.android.support: appcompat-v7:21.0. – Compatibility library developed by Google to bring the action bar feature to older Android versions [6].
2) android-saripaar:2.0-SNAPSHOT– Validation Library

   Android Saripaar is a simple, yet powerful rule-based user interface form validation library for Android. It is the simplest and feature-rich validation library available for Android [8].
3) org.roboguice: roboguice: 3. 0– Dependency Injection Framework RoboGuice 3 helps in smoothing out some of the wrinkles in your Android development experience and makes things simple. RoboGuice 3 takes the guesswork out of development. Inject your View, Resource, System Service, or any other object, and let RoboGuice 3 take care of the details. RoboGuice 3 reduces application code which means fewer opportunities for bugs. It also makes code easier to follow [7].
4) com.balysv: material-ripple: 1.0.1 – For ripple effect in android version less than Android 5. You can use this library in addition with Jake Wharton's animation back port changing the imports from import android.animation.*; to: import om.nineoldandroids.animation.*; to import android.util.Property; to import com.nineoldandroids.util.Property; and in

MaterialRippleLayout.java file, calling functionshouldDelayChildPressedState () only if you're using API greater than 14 [10].

5) com.joanzapata.android: android-iconify: 1.0.9– Uses icons from Font AwesomeLibrary. Iconify allows you to include any of the FontAwesome 4.3.0 icons by Dave Gandy in your texts, your ActionBar, and even in your EditTexts. Icons are infinitely scalable, and customizable with shadows and everything you can do on texts [9].

6) com.afollestad: material-dialogs: 0.6.3.1– Provides alert dialogues in material theme. A beautiful, easy-to-use, and customizable dialogs API, enabling you to use Material design themed dialogs across all versions of Android (unlike AppCompat) [11].

7) it.neokree: MaterialNavigationDrawer: 1.3.2 – It is used to make navigation drawer.

Chapter 5 - Requirements Analysis

For requirements analysis requirements were elicited through requirements gathering by analyzing the need of the user obtaining a clear set of unambiguous requirements and recording those requirements.

5.1 Requirements Gathering

Because the CRS is an application intended for students from different levels of technical backgrounds, the graphical design that is the front end is a main aspect and it should be taken care that it will be easy to use for any user and navigation should be clear. CRS application targets both prospective and also current students. Interaction with several students resulted in a set of parameters which were considered to be the most import factors in choosing a course. The first and foremost parameter was the difficulty level. Other requirements were gathered after interacting with current students who are also interesting in knowing about a course before enrolling. Major suggestions included course availability, course grading criteria, learning curve and few others. Three parameters were chosen from that list, namely difficulty level, grading and learning curve. In order for the users to express their views in detail about the course apart from the ratings a comments bar was added which helps users to post their comments and reviews about a course. All the requirements to build an application were gathered and then approved by the author's professor. Provide a ratings in the categories of difficulty, grading and learning curve which are the major factors in choosing a course

- In a particular module the user can comment on the courses and post their in detail reviews or questions
- Offering opportunity for the user to add questions so that his / her doubts based on the reviews received onto it could be cleared

- Presenting a list of all courses once they select a particular department
- Utilizing a tab view to view the course ratings, top ten courses, and a forum to exchange questions and answers about a particular course
- Recommend course based on the ratings of the course the user is currently viewing

5.2 Requirements Specification

5.2.1 Software Requirements

CRS application requires the following software requirements

Development Perspective:

Operating System: Windows 8

Language: Android Studio, Java

Database: NOSQL

Tools: Genymotion

Technologies: Java, NOSQL, Android, XML.

Debugger: Genymotion Simulator, Android mobile device (Android version 4.0.4)

Application Perspective:

Framework: Android SDK Version 4.0.4

Network Required: Mobile network and Internet (cellular or Wi-Fi)

5.2.2 Hardware Requirements

Development Perspective:

Processor: i3 and higher

RAM: 512 MB

Space on disk: 250MB or higher

Application Perspective:

Device: Android phone with version 4.0.4 or higher

5.3 Feasibility Analysis

5.3.1 Economic Feasibility

CRS application is economically feasible because it requires an Android device with Android Studio 4.0.4 or higher which can be downloaded for free. However, in order to download the application, the users must have a Wi-Fi or a cellular network. Hence it is economically feasible as a Wi-Fi or a cellular connection are available easily at economic prices.

5.3.2 Technical Feasibility

Developing this application needed a system to install Android SDK and develop the application and a device to test it. Also the testing can also be done on an emulator. This application has been tested on HTC One M8 and Genymotion emulator. Hence it is technically feasible.

Chapter 6 – Design

6.1 Control Flow Diagram

Control flow diagram is used to show the flow of control in a particular application. It helps us to understand how the navigation is done among the pages in the application. The control flow diagram in Figure 6.1 illustrates the flow of control in the CRS application. It starts with the login and ends with the logout. The user can navigate intermediately according to the selection made. The control flow diagram below thus illustrates how the user can navigate.

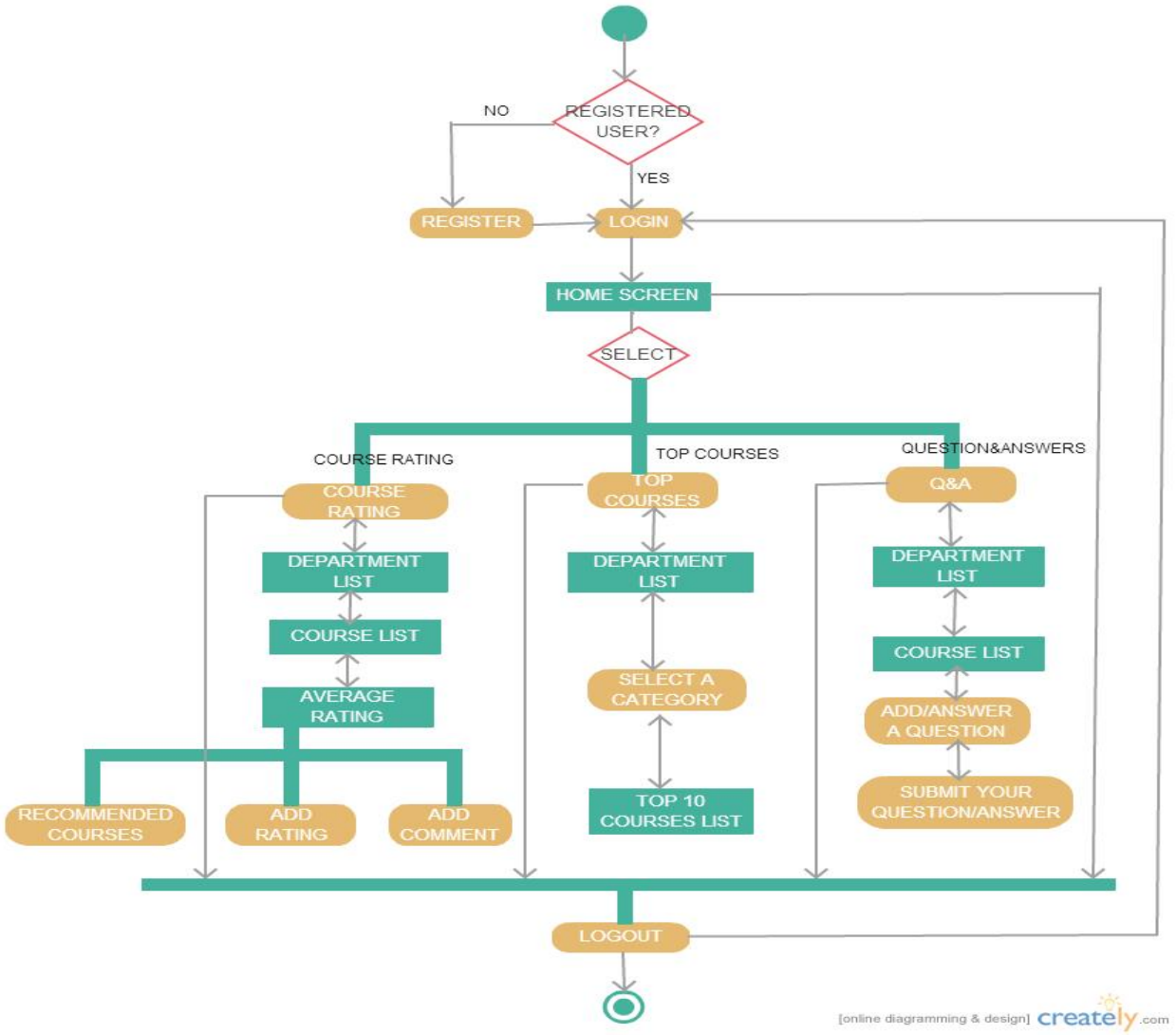

Figure 6.1 Control Flow Diagram

6.2 Use case diagram

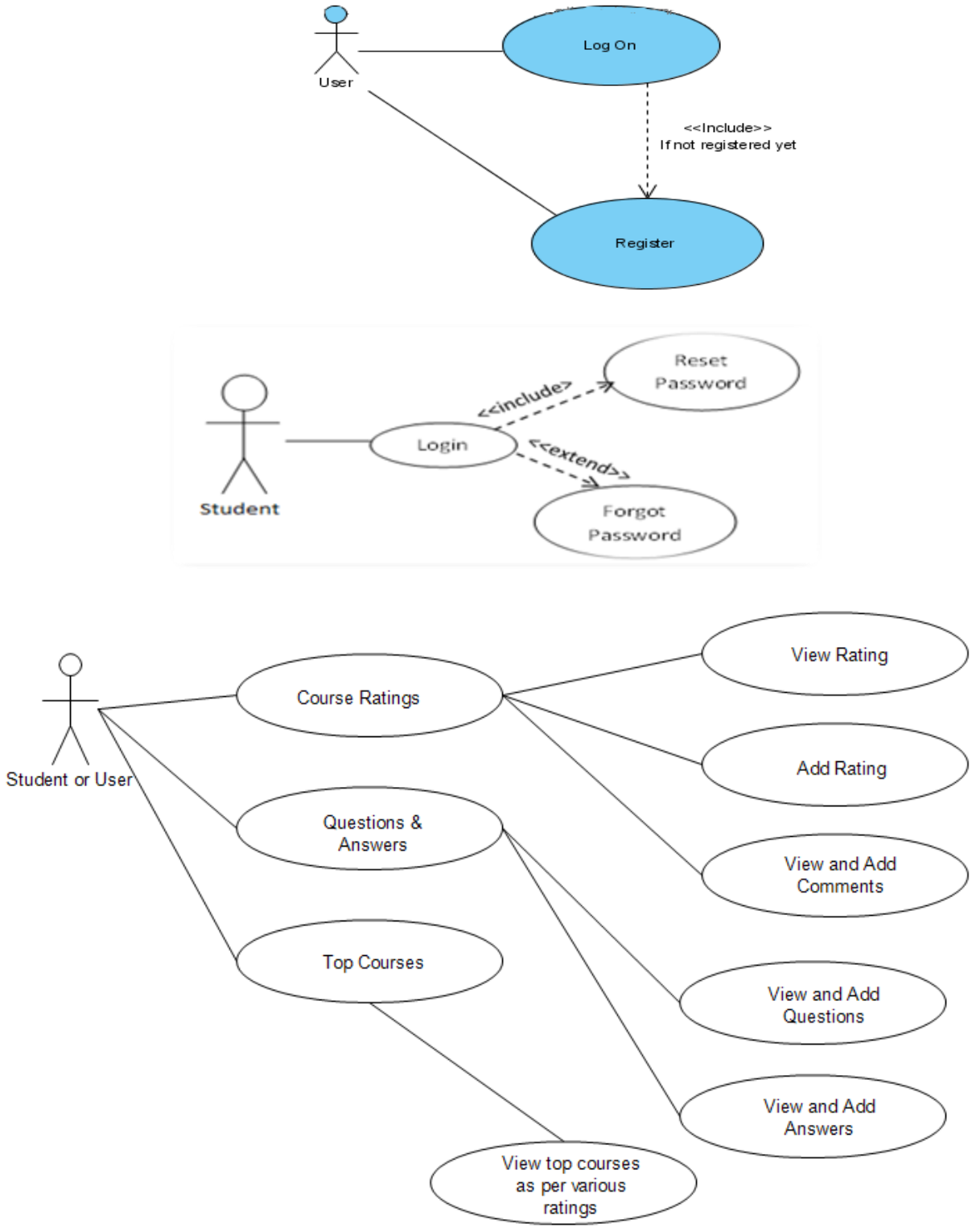

Figure 6.2 Use Case diagram

6.3 Class Diagram

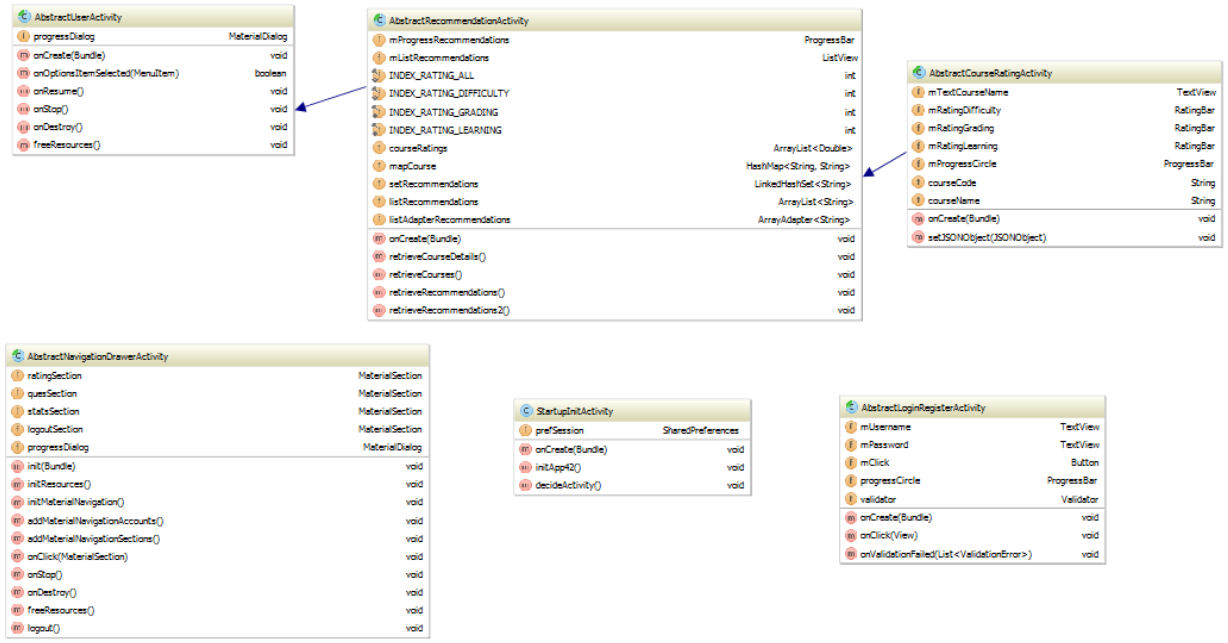

Figure 6.3 Class Diagram

Chapter 7 - Android Framework Components

Android SDK provides us all the required API libraries and developer tools to build, test, and debug apps for Android [3]. Course Review System has been developed in Android studio Integrated Development Environment (IDE) with the Android Developer Tool (ADT) plugin. CRS application can be run on an emulator or generated .apk file can be installed on an android smartphone or tablet.

7.1 AndroidManifest.xml

Every Android application should have an AndroidManifest.xml file to work in its root directory. This file provides essential information about the application to the Android system. It also provides the point from which the application should start, that is

<category android: name="android.intent.category.LAUNCHER" />

Among the other things, the manifest file does the following:

It names the Java package for the application which serves as a unique identifier for the android application. The manifest file also provides a description of components including activities, services, broadcast receivers, and content providers. These declarations enable the Android system to know the list of components and the conditions under which they can be launched.

It tells about the permissions an application has to access protected parts of the API and its permissions to interact with other applications. Similarly tells us about the permissions other applications have to interact with the current application components [12].

It also talks about the minimum sdk version and the target sdk version of the application. The manifest file for CRS can be seen below:

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest package="com.example.crs"
xmlns:android="http://schemas.android.com/apk/res/android"><application
android:theme="@style/AppTheme" android:label="@string/app_name"
android:icon="@drawable/ic_launcher" android:allowBackup="false"><activity
android:theme="@style/AppTheme.NoActionBar" android:label="@string/app_name"
android:name=".base.StartupInitActivity"><intent-filter><action
android:name="android.intent.action.MAIN"/><category
android:name="android.intent.category.LAUNCHER"/></intent-filter></activity><activity
android:theme="@style/AppTheme.NoActionBar" android:label="@string/activity_title_login"
android:name=".ui.LoginActivity" android:windowSoftInputMode="stateAlwaysHidden"
android:launchMode="singleTop"> </activity><activity android:theme="@style/AppTheme"
android:label="@string/activity_title_register" android:name=".ui.RegisterActivity"
android:windowSoftInputMode="stateAlwaysHidden"
android:parentActivityName=".ui.LoginActivity"> </activity><activity
android:theme="@style/NavigationDrawerTheme"
android:label="@string/activity_title_departments" android:name=".ui.DepartmentListActivity"
android:launchMode="singleTop" android:configChanges="orientation|screenSize">
</activity><activity android:theme="@style/AppTheme"
android:label="@string/activity_title_courses" android:name=".ui.CourseListActivity"
android:launchMode="singleTop" android:parentActivityName=".ui.DepartmentListActivity"
android:configChanges="orientation|screenSize"> </activity><activity
android:theme="@style/AppTheme" android:label="@string/activity_title_course_rating"
```

```
android:name=".ui.CourseAverageRatingActivity"
android:parentActivityName=".ui.CourseListActivity"
android:configChanges="orientation|screenSize"> </activity><activity
android:theme="@style/AppTheme" android:label="@string/activity_title_course_rating"
android:name=".ui.CourseAddRatingActivity"
android:parentActivityName=".ui.CourseAverageRatingActivity"
android:configChanges="orientation|screenSize"> </activity><activity
android:theme="@style/AppTheme" android:label="@string/activity_title_ques_ans"
android:name=".ui.QuestionListActivity" android:launchMode="singleTop"
android:parentActivityName=".ui.CourseListActivity"
android:configChanges="orientation|screenSize"> </activity><activity
android:theme="@style/AppTheme" android:label="@string/activity_title_ques_ans"
android:name=".ui.AnswerListActivity" android:parentActivityName=".ui.QuestionListActivity"
android:configChanges="orientation|screenSize"> </activity><activity
android:theme="@style/AppTheme" android:label="@string/activity_title_stats"
android:name=".ui.StatisticsActivity"
android:parentActivityName=".ui.DepartmentListActivity"
android:configChanges="orientation|screenSize"> </activity></application><uses-permission
android:name="android.permission.INTERNET"/></manifest>
```

According to the manifest file, the minimum sdk version for this application is 13.Therefore, the devices below sdk version 13 that is android Honeycomb_MR2 will not be able to run this application. Different activities and their intents are mentioned in this file for CRS.

7.2 Android Dependencies

Android Dependencies is a virtual folder containing JAR files that Eclipse uses for the project. It is a virtual folder and will not be found on the hard disk. For Course Review System android-support-v7-appcompat.jar has been added to enable ActionBar and it in turn depends on v4 Support Library.

7.3 Activity

Activity is an important component which helps provide screen so that the users can interact. All the activities in a system are managed as an activity stack that is the last-in first-out procedure. Therefore, if a new activity is started, it is placed on the top and will be the current activity that is running and the previous activity will be in the background [14].

An activity has essentially four states:

- Active or running when the activity is in foreground and is running
- Paused state when the activity has lost focus but is still visible
- Stopped state if the activity is completely obscured by another activity
- An activity that is paused or stopped can be dropped by the system from memory by either asking it to finish, or by killing its process

7.4 Activities Involved in Course Review System

7.4.1 StartupInitActivity

This activity is mainly about establishing a connection with the back end which in this CRS application is App42. Here we use init () function and set the database which we are going to connect to. App42 has a two way authentication process which provides a security key to be entered while connecting. Another special key which is application specific must also be entered

along with the security key to establish a secure connection with the back end. This activity mainly does that.

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        prefSession = getSharedPreferences (getString (R.string.preference_session),
                MODE_PRIVATE);
        initApp42 ();
        decideActivity ();
}
private void initApp42 () {
        App42API.setDbName (getString (R.string.app42_database_default));
        App42API.initialize (getApplicationContext (), getString (R.string.app42_api_key),
                getString (R.string.app42_secret_key));
}
```

7.4.2 LoginActivity

This activity mainly deals with the login page. The onClick () function mainly deals with forgot password, and register buttons functionality. The onValidationSuccess () function determines what must be done once the correct username and password credentials are entered by the user. The on Success () function verifies the username and password entered to match with those stored in database using user service. The onException () function sets an error dialog to be displayed when the credentials entered by the user does not match those in the database.

7.4.3 RegisterActivity

This activity mainly deals with the registration of the user. The RegisterActivity class extends the AbstractLoginRegisterActivity. We use methods like onCreate (), onValidationSuccess () are used to take the details entered by the users and then to verify that the details are unique and does not exist in the database. We use the onException () method to again raise any errors if any of the details entered by the user are not matching the format set. This displays the error message as a toast.

7.4.4 DepartmentListActivity

This activity mainly deals with displaying the list of departments after the login. The DepartmentListActivity class extends the AbstractNavigationDrawerActivity which adds different sections to the activity based on the things user selects. The init (Bundle bundle) function mainly checks for the login and on success launches the department list. The retrieveDepartments () function gets the list of collections from the database.

7.4.5 CourseListActivity

This activity mainly deals with displaying the list of the course from a particular department selected from the department list. The CourseListActivity class extends the AbstractUserActivity class. It uses the onBundle () similar to DepartmentListActivity. The retrieveCourses () function is used to display the list of courses. This is done by using the findDocumentByKeyValue () function and by passing the dept_code as a parameter.

```
private void retrieveCourses () {
        progressDialog.show ();
        String dbName = getString (R.string.app42_database_default);
        String collectionName = getString (R.string.app42_collection_course);
        String key = getString (R.string.jsonkey_course_deptcode);
```

```
String value = PreferenceHandler.getActivityPreference (this,
                    R.string.prefkey_activity_deptcode);
StorageService storageService = App42API.buildStorageService ();
storageService.findDocumentByKeyValue (dbName, collectionName, key, value, new
App42CallBack () {
public void onSuccess (Object response)
                    {
Storage storage = (Storage) response;
System.out.println ("dbName is" + storage.getDbName ());
System.out.println ("collection Name is" + storage.getCollectionName ());
ArrayList<Storage.JSONDocument> jsonDocList = storage.getJsonDocList ();
        for (int i=0; i<jsonDocList.size (); i++)
                    {
                      System.out.println ("objectId is" + jsonDocList.get (i).getDocId ());
                       System.out.println ("Jsondoc is" + jsonDocList.get (i).getJsonDoc ());
                        try {
                            JSONObject obj = new JSONObject (jsonDocList.get (i).
                                                getJsonDoc ());
                            listCourse.add (obj.getString (getString
                            (R.string.jsonkey_course_deptcode))
                             + " " + obj.getString (getString (R.string.jsonkey_course_code))
                             + " - " + obj.getString (getString (R.string.
                             jsonkey_course_name)));
```

```
                    } catch (JSONException e) {
                    e.printStackTrace ();
                        }
                }
            runOnUiThread (new Runnable () {
@Override
              public void run () {
                    listAdapter.notifyDataSetChanged ();
                    progressDialog.hide ();
              }
                    });
 }
```

7.4.6 CourseAverageRatingActivity

This activity is mainly concerned with displaying the average ratings of the difficulty_rating, grading_rating and learning_rating parameters. The CourseAverageRatingActivity class extends the AbstractCourseRatingActivity class. The submitComment () function deals with the comments section where users can add their comments through a dialog box. The retrieveComments () function retrieves the comments of the particular course page being displayed from the database. The retrieveAverageRating () function retrieves the average ratings of the particular course for the three different parameters from the database using StorageService object which used findDocumentByKeyValue () function.

7.4.7 CourseAddRatingActivity

This activity is mainly concerned with adding new ratings and also updating the ratings previously submitted by a user. For this we use two different functions. The code below is used to update a particular rating previously submitted by the user.

```
int number = 0;
double grading = 0;
double learning = 0;
double difficulty = 0;
double all = 0;
System.out.println ("jsonData before calc update: " + jsonData);
try {
        number = jsonData.getInt (getString(R.string.jsonkey_rating_average_number));
        grading = ((jsonData.getDouble (getString (R.string.jsonkey_rating_grading)) * number)
                - oldRetrievedData.getDouble (getString (R.string.jsonkey_rating_grading))
                + mRatingGrading.getRating ()) / number;
        learning = ((jsonData.getDouble (getString(R.string.jsonkey_rating_learning)) * number)
                - oldRetrievedData.getDouble (getString (R.string.jsonkey_rating_learning))
                + mRatingLearning.getRating ()) / number;
        difficulty = ((jsonData.getDouble(getString(R.string.jsonkey_rating_difficulty))*number)
                - oldRetrievedData.getDouble (getString(R.string.jsonkey_rating_difficulty))
                + mRatingDifficulty.getRating ()) / number;
        all = grading + learning + difficulty;
```

In order to add a new rating by the user and then to update the average rating of the course accordingly the following code is used.

```
int number = 0;
double grading = 0;
double learning = 0;
double difficulty = 0;
double all = 0;
try {
        number = jsonData.getInt (getString (R.string.jsonkey_rating_average_number));
        grading = ((jsonData.getDouble(getString(R.string.jsonkey_rating_grading)) * number)
                + mRatingGrading.getRating()) / (number + 1);
        learning = ((jsonData.getDouble(getString(R.string.jsonkey_rating_learning)) * number)
                                + mRatingLearning.getRating ()) / (number + 1);
        difficulty = ((jsonData.getDouble(getString(R.string.jsonkey_rating_difficulty))*number)
                + mRatingDifficulty.getRating ()) / (number + 1);
        all = grading + learning + difficulty;
        number += 1;
    } catch (JSONException e1) {
            e1.printStackTrace ();
        }
```

7.4.8 QuestionListActivity

This activity mainly deals with the Questions section of Questions and Answers module of CRS. We can add new question to this section by pressing on the '+' button at the top right corner of the page. The submitQuestion () function is mainly used to submit new questions. The onException () method is used to display any error message if a problem is encountered.

7.4.9 AnswerListActivity

This activity deals with the Answers section of the Questions and Answers. Here the user can select a particular question and view any answers already existing or can also add his/her own answer using the submitAnswer () function.

7.4.10 StatisticsActivity

This activity is mainly used to calculate and display the top 10 courses in each of the five categories. The StatisticsActivity class extends the AbstractUserInterface class. In order to calculate and display the top ten courses list in each of the five different categories, five different functions retrieveTopAll (), retrieveTopDifficulty (), retrieveTopGrading (), retrieveTopLearning (), retrieveTopNumber () are used.

7.4.11 AbstractRecommendationsActivity

This activity deals with the recommendations of the courses in the course rating page. The recommendations are generated using a compound query in the retrieveRecommendations2 () class as shown in the code below.

```
protected void retrieveRecommendations2 () {
        final String courseCurrent = PreferenceHandler.getActivityPreference (this,
                                        R.string.prefkey_activity_coursecode);
        String dbName = getString (R.string.app42_database_default);
        String collectionName = getString (R.string.app42_collection_rating_average);
        String key1 = getString (R.string.jsonkey_rating_average_deptcode);
        String value1 = PreferenceHandler.getActivityPreference (this,
                                R.string.prefkey_activity_deptcode);
        String key2 = getString (R.string.jsonkey_rating_difficulty);
        double value2 = courseRatings.get (INDEX_RATING_DIFFICULTY);
```

```
String key3 = getString (R.string.jsonkey_rating_learning);
double value3 = courseRatings.get (INDEX_RATING_LEARNING);
String key4 = getString (R.string.jsonkey_rating_grading);
double value4 = courseRatings.get (INDEX_RATING_GRADING);
Query query1 = QueryBuilder.build (key1, value1, QueryBuilder.Operator.EQUALS);
Query query2 = QueryBuilder.build (key2, value2, QueryBuilder.Operator.EQUALS);
Query query3 = QueryBuilder.build (key3, value3, QueryBuilder.Operator.EQUALS);
Query query4 = QueryBuilder.build (key4, value4, QueryBuilder.Operator.EQUALS);
Query queryCompound1 = QueryBuilder.compoundOperator (query3,
                                QueryBuilder.Operator.OR, query4);
Query queryCompound2 = QueryBuilder.compoundOperator (query2,
                                 QueryBuilder.Operator.OR, queryCompound1);
Query query = QueryBuilder.compoundOperator (query1, QueryBuilder.Operator.AND,
                    queryCompound2);
String keySort = getString (R.string.jsonkey_rating_average_number);
int max = 10;
int offset = 0;

StorageService storageService = App42API.buildStorageService ();
storageService.findDocsWithQueryPagingOrderBy (dbName, collectionName, query,
max, offset, keySort, OrderByType.DESCENDING, new App42CallBack () {
public void onSuccess (Object response) {
                    Storage storage = (Storage) response;
```

```
System.out.println ("dbName is" + storage.getDbName ());
System.out.println ("collection Name is" + storage.getCollectionName ());
ArrayList<Storage.JSONDocument> jsonDocList =
storage.getJsonDocList ();
for (int i = 0; i < jsonDocList.size (); i++) {
System.out.println ("objectId is" + jsonDocList.get (i).getDocId ());
        System.out.println ("Jsondoc is" +
        jsonDocList.get (i).getJsonDoc ());
         try {
              JSONObject obj = new
                        JSONObject (jsonDocList.get (i).getJsonDoc ());
               if (! courseCurrent.equals (obj.getString (getString
               (R.string.jsonkey_rating_coursecode))) {
                        setRecommendations.add (obj.getString (getString
                        (R.string.jsonkey_rating_coursecode))
                        + " - " + mapCourse.get (obj.getString (getString
                        (R.string.jsonkey_rating_coursecode))));
                           }
                } catch (JSONException e) {
      e.printStackTrace();
    }
}
```

7.5 Intent

Intent is an android component which provides an abstract description of an operation which is to be performed [15]. The two primary forms of intents are

- Explicit Intents**:** These intents have a specified component that is via setComponent (ComponentName) or setClass (Context, Class), thus providing the exact classes to run
- Implicit Intents**:** These intents will not have specified a component. Thus they include enough information for the system to determine which of the available components suits the best to run for that intent

7.6 Layout Inflator

Layout Inflator android component helps loading the layout XML file into its view objects such as ProgressBar, TextView etc. It is used in conjunction getLayoutInflater () or getSystemService (String) to retrieve a standard LayoutInflater instance which is already hooked up to the current context.

"LayoutInflater li = (LayoutInflater) context.getSystemService (Context.*LAYOUT_INFLATER_SERVICE*);"

Chapter 8 – Graphical User Interface

The front end of the CRS application was developed using XML. Each and every component is briefly described in the following sections.

8.1 Logo

The logo for CRS application was designed in Photoshop and viewed using App Drawer. The App Drawer is also called the App Tray. App Drawer is a collection of all the apps and widgets installed on your device. The logo in Figure 8.1 is used mainly in the login page and also in the registration page. Purple was used mainly to keep up the K-State theme which was used throughout the application.

Figure 8.1 Logo of CRS

8.2 Login Page

Figure 8.2 shows the login screen of the CRS application. A login application is the screen asking your credentials to login to this particular application. The Login page also contains "forgot password" option through which a user can easily retrieve his/her password using their respective username.

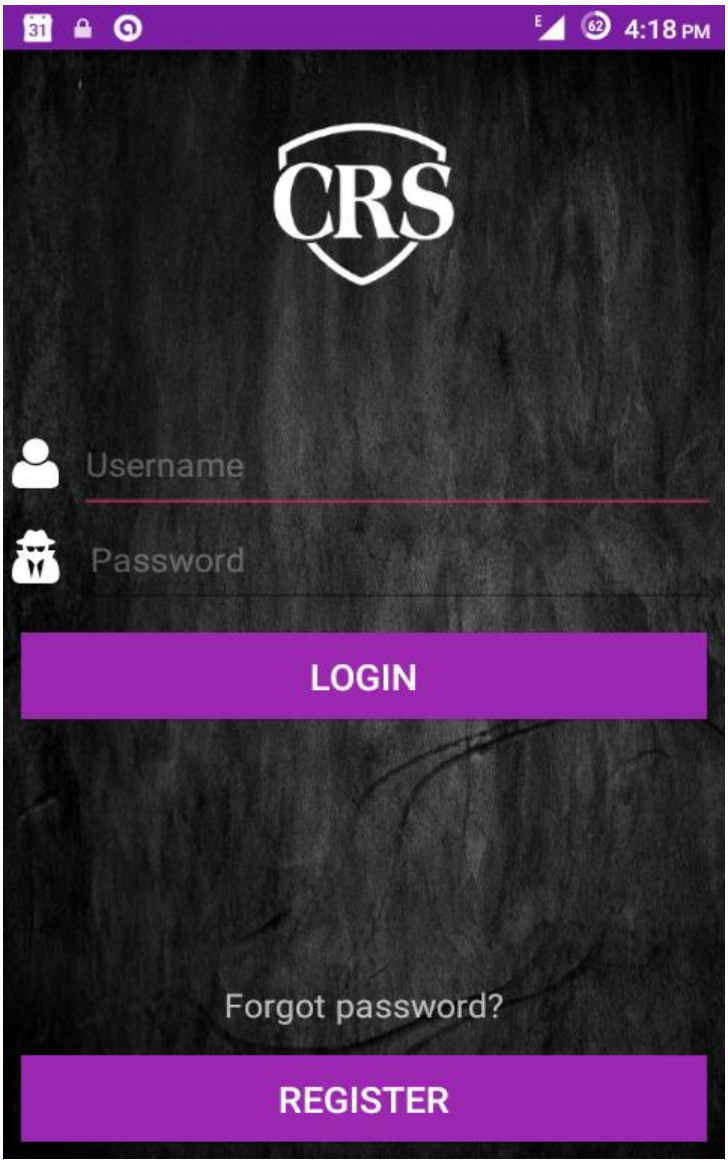

Figure. 8.2 Login Page

8.3 Registration Page

Course Review System contains a login and registration process in order to authenticate a user. The user can enter his/her details which will are saved in the database, and thereby authenticating the user into the respective application and use it on accordingly. Figure 8.3 shows the Registration page of CRS including username which can be selected by the user, the user's email address, a password containing a minimum of 8 alphanumeric characters, a confirm password field matching the password and a student category which is a drop-down menu containing the options of Undergraduate, Graduate and Ph. D.

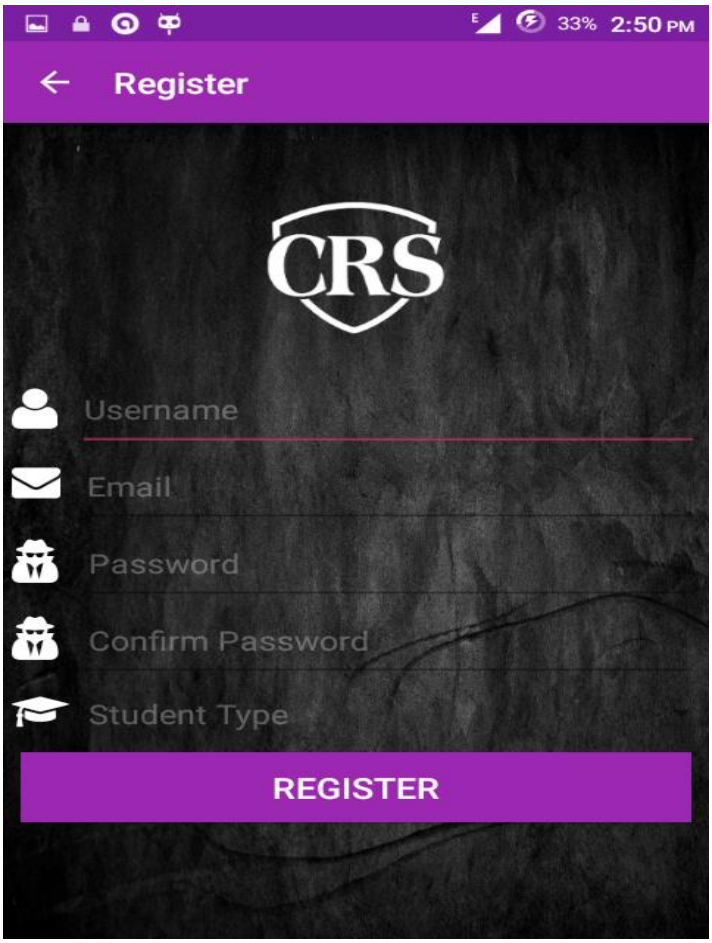

Figure 8.3 Registration Page

8.4 Home Screen

Figure 8.4 shows the Home screen for CRS application. The left side of the screen contains a fragment with the username, and student category. Under this segment three different options are available: Course Ratings, Questions & Answers, and Statistics. The Logout option is located at the bottom of this fragment.

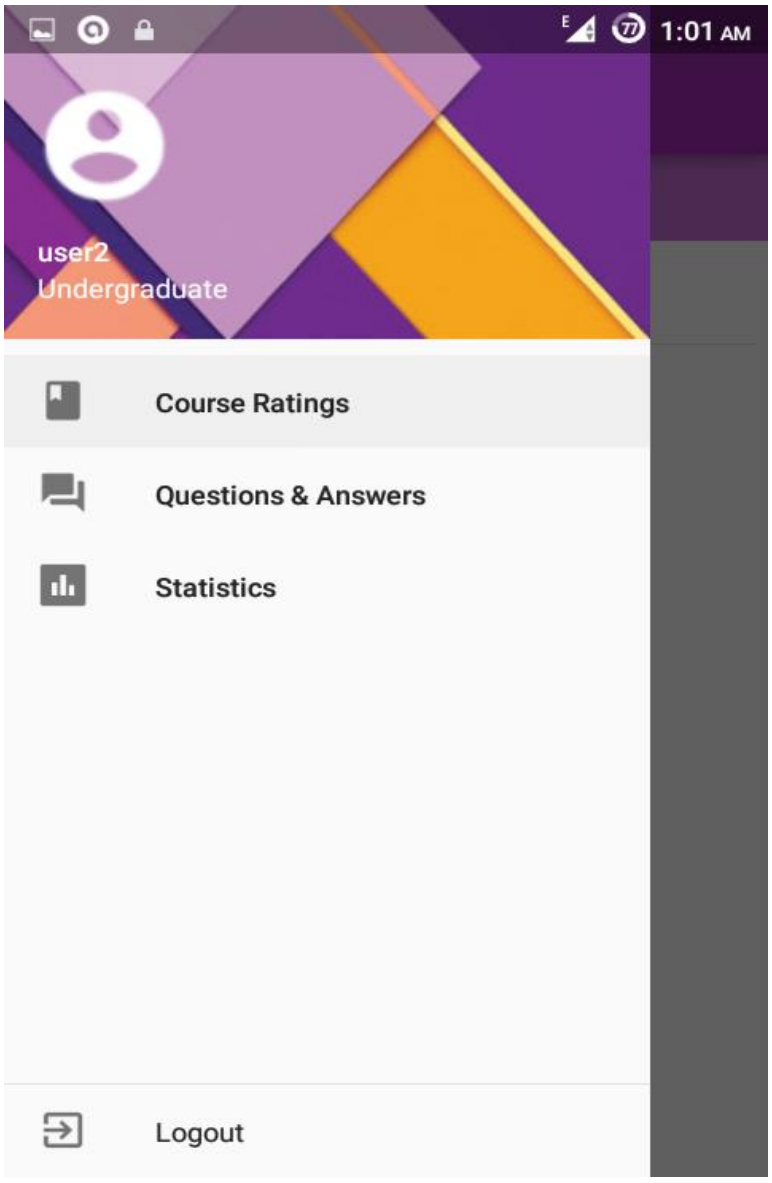

Figure 8.4 Home Screen

8.5 Course Ratings

When a user selects the course ratings option a screen appears as shown in Figure 8.5 which contains the list of departments. When a department is selected, user navigates to a page containing a list of courses in that department as in figure 8.6. Once a course is selected the user is directed to the course ratings page as in Figure 8.7.

Figure 8.5 Course Ratings/Departments

Figure 8.6 Course Ratings/Courses

The Course Ratings page of a particular course contains the course title followed by average ratings for that course in the categories of difficulty level, grading, and learning curve. The action bar on this page contains two buttons: the ‘add ratings’ button which allows a user to add your ratings for that particular course and the ‘add comment’ button which allows a user to add a comment in the comments bar below. The comments bar displays user comments. The bottom of the Course Ratings page contains a pull up tab of recommended courses which on clicking displays the recommended courses for the user based on the course the user is viewing.

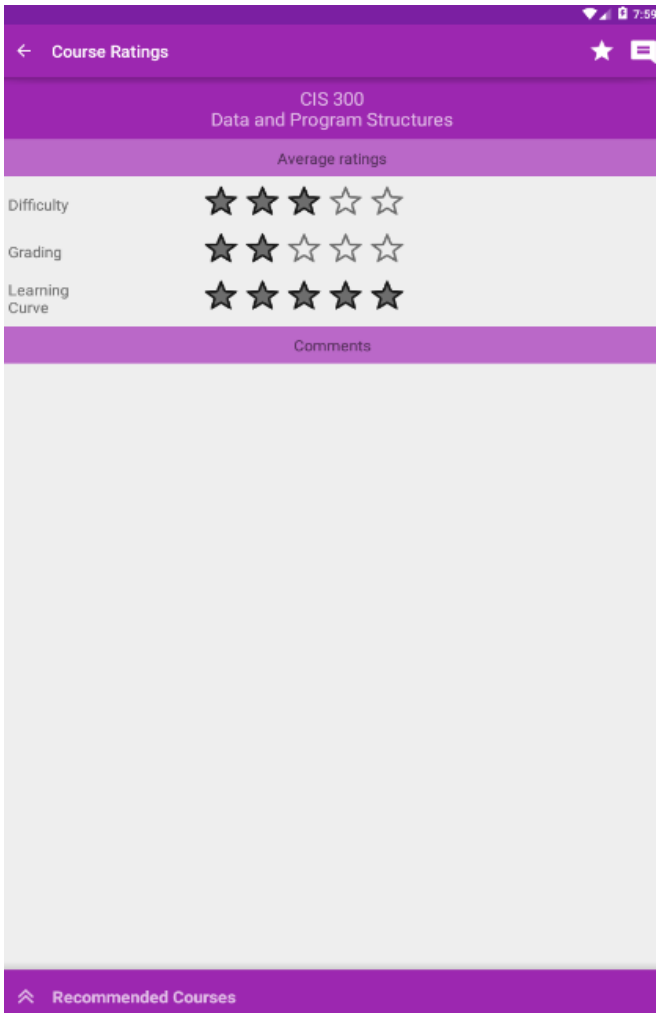

Figure 8.7 Course Ratings

8.6 Recommended Courses

The pull up tab at the bottom of course ratings page displays a list of recommended courses for the user based on the ratings of the course the user is currently viewing as shown in Figure 8.8.

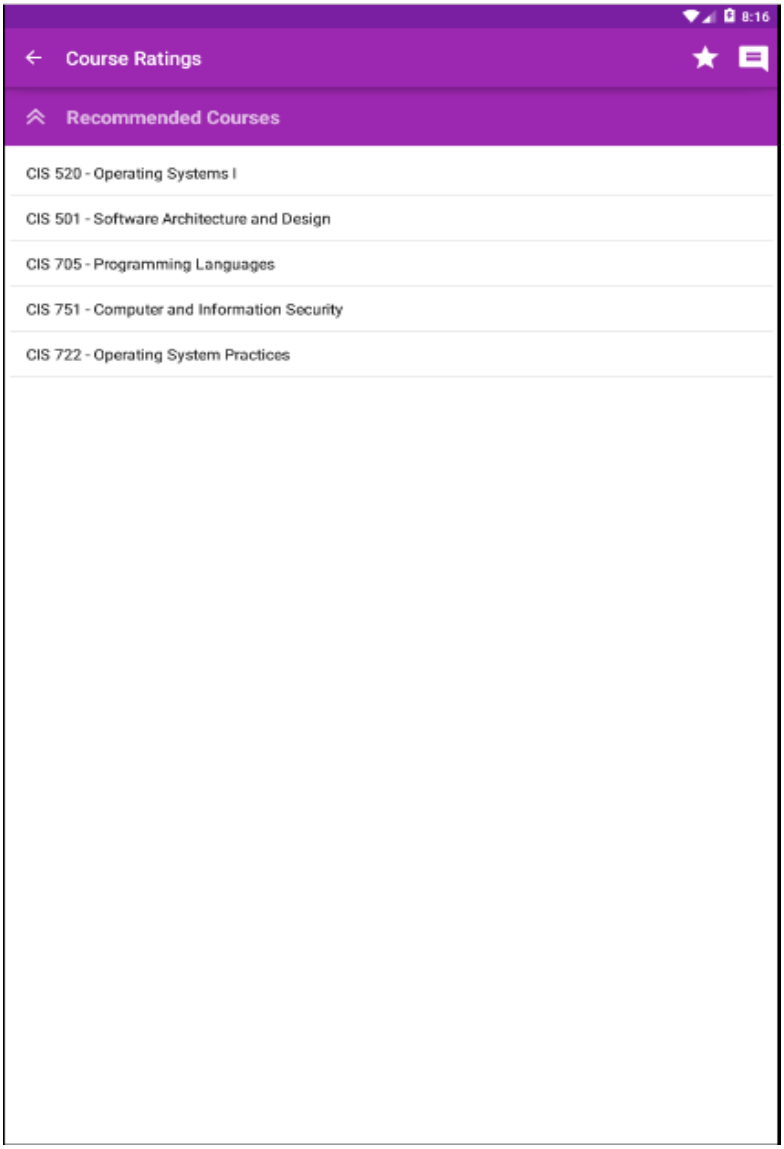

Figure 8.8 Recommended Courses

8.7 Top Courses

When user selects the top courses option the user is directed to a page containing a list of departments. When a department is selected the user is redirected to a page containing a dropdown menu containing categories for the Top courses display as in Figure 8.9. When a category is selected the user is presented with a list of the top 10 courses in that category as in Figure 8.10.

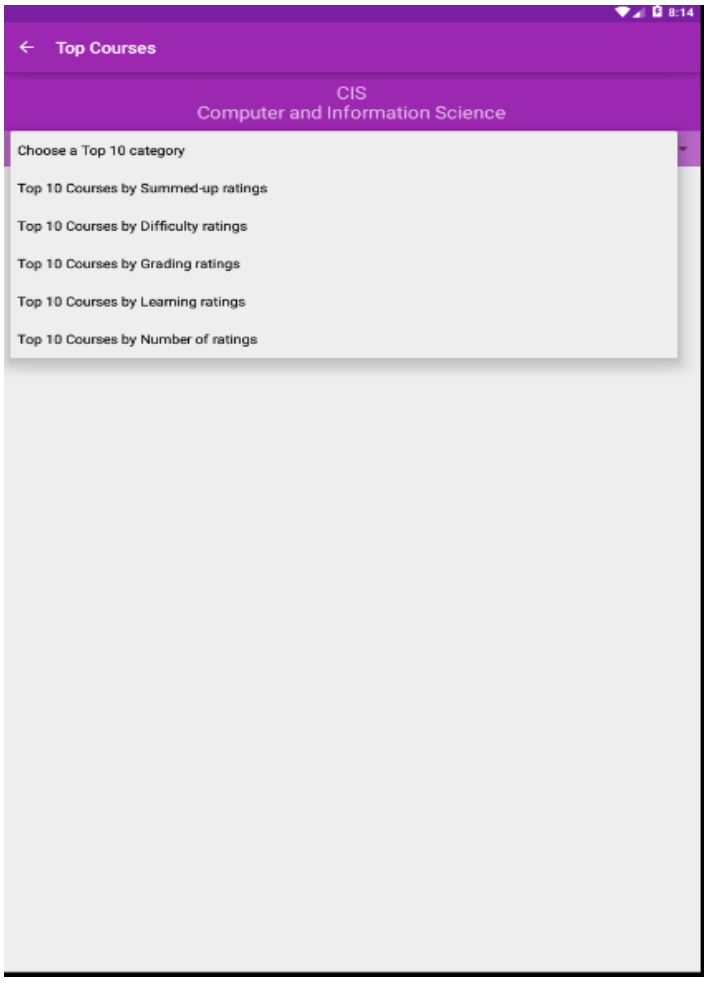

Figure 8.9 Top Courses/Category

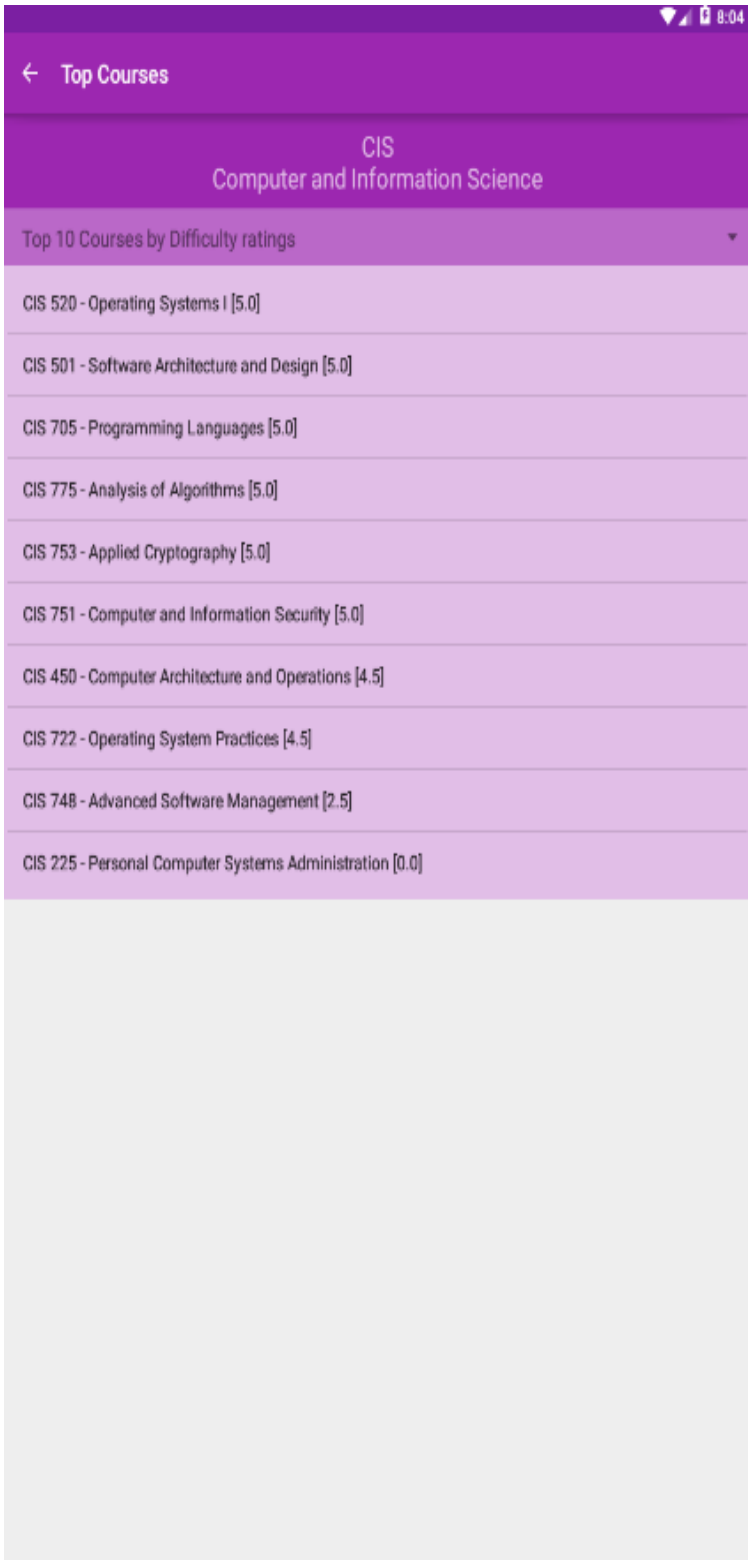

Figure 8.10 Top Courses

8.8 Questions and Answers

The Questions and Answers section allows the user to select from an available list of questions or add his/her own questions as in Figure 8.11. Once a particular question is selected, the user can either view the answers already available or add his/her answer to that question there as in Figure 8.12.

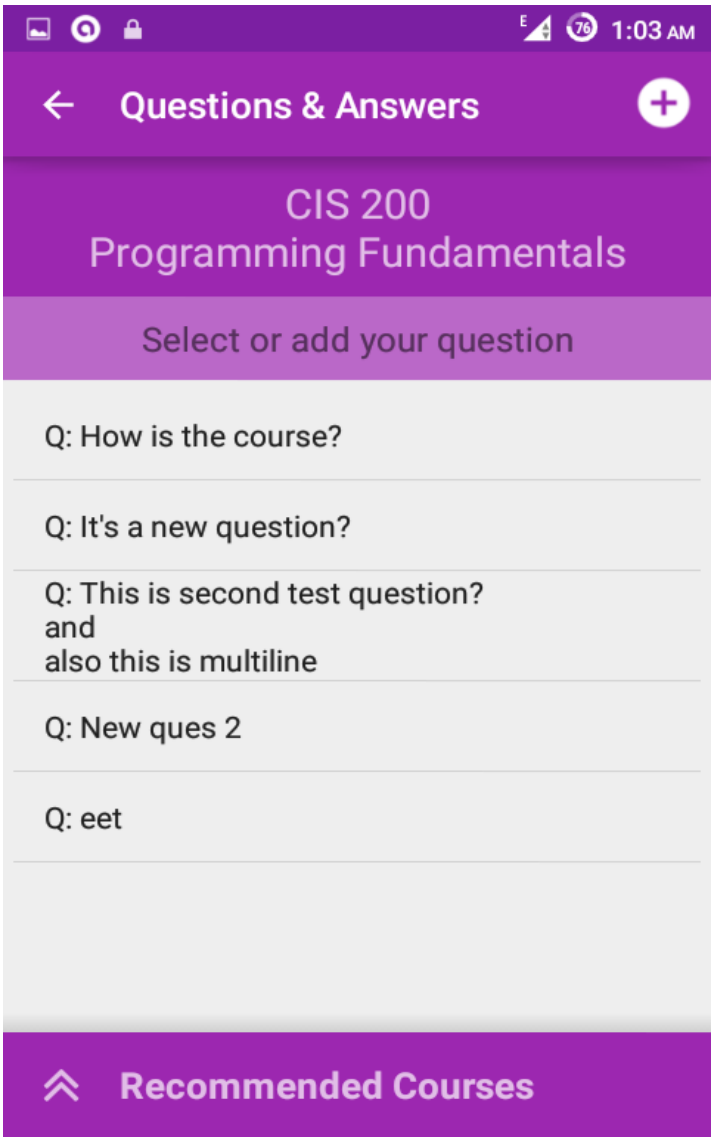

Figure 8.11 Questions and Answers/Questions

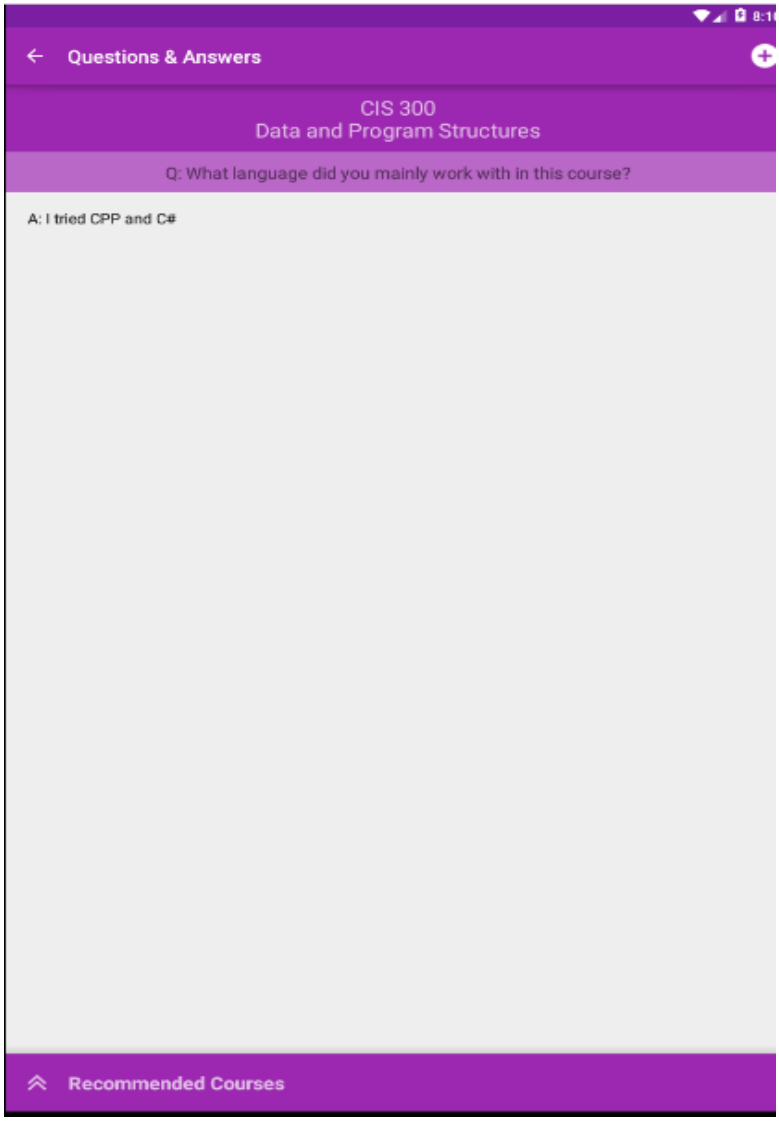

Figure 8.12 Questions and Answers/ Answers

Chapter 9 – Testing

The primary goal behind testing an application is to identify the defects in that application. By verifying if the system met all the requirements such as functionality, reliability, and usability. Testing also confirms validate if the product developed was in accordance with user requirements leading to product improvements. Various types of testing performed for CRS application and are explained in the following sections.

9.1 Unit Testing

Unit testing is the process by which individual units of the source code are tested to verify if they function as desired. Unit tests were performed for CRS and their results are tabulated below in Table 9.1.

9.1.1 Unit Test Cases

S.no	Test Case	Expected Result	Result
1	On load Start up screen	Display Login Page	Pass
2	On click of Registration	Display the screen with the registration details of the user.	Pass
3	On click of Forgot Password link	Display a dialog box which asks for your username to reset the password	Pass
4	On click of Course Rating	Display the average rating in all 3 categories for that course	Pass

5	On click of Add rating button on top-right corner	Open the screen where you can add the user ratings for that course in rating module.	Pass
6	On click of comments button	Open up the dialog to enter your comments about that particular course in the course ratings page	Pass
7	On click of Recommended courses	Display a list of recommended courses for that particular course	Pass
8	On click on Question and Answers	Save the questions and answers given by the user to database and show the list of them.	Pass
9	On click on + buttons in Questions And Answers	Display the dialog box which lets user add their questions	Pass
10	On click on a question in Questions and Answers	Open a dialog box to let user add their answer to that specific question	Pass
11	On click of a top Courses category	Display the list of top 10 courses in that particular category	Pass

Table 9.1 Unit Test Cases

9.2 Compatibility Testing

CRS application was installed on various devices such as HTC One M8, Google Nexus 4, Samsung Galaxy 4 and Samsung Galaxy Tab 10.1 in order to test the compatibility. The application ran with the expected resolution. In order to provide proper image resolution, images are stored with various resolutions in folders hdpi, xhdpi, and xxhdpi and so on.

9.3 Usability Testing

Three test subjects installed the CRS application on their phones. They tested all the modules and identified some defects such as missed error checking, course not saved after pressing save, and various other bugs and suggestions. All of these have been incorporated and the application has been re coded to include all these inputs received from Usability testing.

9.4 Battery consumption

The battery consumption has been tested using a Samsung Galaxy Grand device. The application was tested for the time where the battery percentage started at 100% and reduced to 10 %. In the first case the phone was used to perform normal operations like audio playback, voice calls, WhatsApp texting and voice calls. In the second case along with the normal operations, the CRS application was running in the background. In both the cases, the phone was using Internet with Wi-Fi or 4G/LTE Networks.

S. No	Applications Running	Time taken for the battery to reduce to 10%
1	Normal operations like audio playback, voice calls, WhatsApp texting and voice calls	117 minutes
2	CRS Application along with other normal operations	105 minutes

Table 9.2 Battery consumption

Chapter – 10 Conclusion

The CRS application was proven to help students learn about a course, which they might want to enroll, from other experienced users, by providing a friendly user interface. CRS application provides the user with a feasibility to add course reviews, rate them in different aspects and write comments on it. The recommendations make the search of courses easier for the users and keeps them on the track. Overall it is an application which allows students to gain all the information about a course they need to know before enrolling and helps them plan their coursework more effectively.

The CRS application followed a complete software development life cycle with analysis, followed by requirements gathering, implementation which is done using Android Studio with ADT plugin and testing which was done on real devices and also using Genymotion emulator. Throughout this process I have learnt android development and understood its various components and the functionality. It also helped me gain knowledge about App42 cloud service.

References

[1] ShepHertz Technologies Pvt Ltd., "APP42 Cloud API", retrieved from URL: http://api.shephertz.com/, on Apr. 4, 2015.

[2] ShepHertz Technologies Pvt Ltd., "APP42 Cloud AppHQ", retrieved from URL: https://apphq.shephertz.com/, on Apr. 4, 2015.

[3] Google Inc., "About Android", retrieved from URL: http://developer.android.com/about/index.html on Mar. 5, 2015

[4] ShepHertz Technologies Pvt Ltd., "About APP42 PaaS", retrieved from URL: http://app42paas.shephertz.com/about-us/ on Apr. 4, 2015

[5] ShepHertz Technologies Pvt Ltd., "Features of APP42 Cloud API", retrieved from URL: http://app42paas.shephertz.com/dev-center/features/ on Apr. 6, 2015

[6] Google Inc., "Android Libraries", retrieved from URL: https://developer.android.com/tools/support-library/features.html on Mar. 24, 2015

[7] roboguice, "Roboguice 3.0", retrieved from URL: https://github.com/roboguice/roboguice on Mar. 24, 2015

[8] Ragunath Jawahar, "Android Saripaar library", retrieved from URL: https://github.com/ragunathjawahar/android-saripaar on Mar. 24, 2015

[9] Joan Zapata, "Android iconify library", retrieved from URL: https://github.com/JoanZapata/android-iconify on Mar. 25, 2015

[10] Balys Valentukevicius, "Android balysv Library", retrieved from URL: https://github.com/balysv/material-ripple on Mar. 24, 2015

[11] Aidan Follestad, "Android material dialogs", retrieved from URL: https://github.com/afollestad/material-dialogs on Mar. 25, 2015

[12] Google Inc., "Manifest", retrieved from URL: http://developer.android.com/guide/topics/manifest/manifest-intro.htmlon Mar. 25, 2015

[13] Google Inc., "Android Framework Components", retrieved from URL: http://developer.android.com/guide/components/fundamentals.html on Mar. 25, 2015

[14] Google Inc., “Android Activity”, retrieved from URL: http://developer.android.com/reference/android/app/Activity.html on Mar. 25, 2015

[15] Google Inc., “Android Intent”, retrieved from URL: http://developer.android.com/reference/android/content/Intent.html on Mar. 25, 2015

[16] Wikipedia Foundation., “Android Architecture”, retrieved from URL: http://commons.wikimedia.org/wiki/File:Android-System-Architecture.svg on Mar. 26, 2015

[17] Dimensional Research, “Dimensional Research survey”, retrieved from URL: http://dimensionalresearch.com/ , https://www.zendesk.com/resources/customer-service-and-lifetime-customer-value/ on Apr. 21, 2015