1. **INTRODUCTION**

At every level of education, ranging from primary education to higher institution, assignments and class works are being used by the teachers after class session to check the assimilation rate of his/her students in order to ensure that knowledge is impacted into students’ life. Assignments can be defined as tasks allocated to student and related to student course of study.

This project focused on development of android mobile application for assignment tracking system. This project was developed mainly for tracking of student’s provided solutions to course assignments and it was developed as mobile application in order to provide better accessibility of the application and easy tracking of the assignment solutions for the users which are tentatively college students and teachers.

**1.1 Objective**

The objective of this project involved opportunity for college students to perform various activities such as allowing student to register in order to use the application functionalities, enables student to select from list of courses and enroll for the course, provides student with available assignments and his/her submitted assignments on the courses the students have enrolled for and provides opportunity for student to check submission rate of assignments for each courses.

This project also provided opportunity for college teachers to perform various activities such as allowing teacher registration, enables teacher to add new course and add new assignment, provides easy way for teacher to mark assignments for student after teacher has checked that the assignments have been satisfactorily completed by the student and most importantly provides teachers with assignment submission rate and completion rate which are the information about students progress on all the courses’ assignments.

This application data are stored inside MySQL database and the application server side which is implemented in PHP receives data from user interface and sends to MySQL database using android application with internet capability. The data are transferred from MySQL database to server side and from server side transferred to android application using JSON format.

1. **RELEVANT TECHNOLOGIES**

This section describes the structure of the application, application development environment, application development process and the relevant technologies used to build the application.

**2.1 Application Structure**

This application is divided into two main section which are teacher section which is the application used by the teachers and the student section which is the application used by the students. Each of this section of the application can be divided into two main different parts which are client side and server side.

Client side represents android devices which provide users interface for manipulation and visualization of data and server side represents MySQL database which is meant for storing of data.

In this project, communication between android device and MySQL database which is used to store data is facilitated through PHP scripts because android device cannot communicate with MySQL database directly. PHP scripts serve as intermediary between android device and MySQL database.

The detail descriptions of the communication processes between this application client side and server side is described and shown in diagram below.

* The client device makes HTTP POST/GET request to server
* The PHP scripts make connection and queries MySQL server
* MySQL server sends data to PHP
* The PHP scripts write the data in JSON format by assigning keys for the values in JSON array
* Lastly, the application parses the JSON array and displays the data on client side.

**ANDROID DEVICE** **MYSQL DATABASE**

**JSON**

**2.2 Application Development Environments**

The appropriate development environment is required in order to achieve the objectives of this project. The following hardware and software tools are needed to set up appropriate environments for development of this application.

**2.2.1 Hardware**

The appropriate hardware environment for android application development requires a personal computer with either of the following operating system;

* Microsoft Windows 7 or later version
* Mac OS X 10.5.8 or later version with Intel chip
* Linux which includes GNU C Library 2.7 or later.

**2.2.2 Java JDK and JRE**

Android application development environments include essential software components such as Java Development Kit (Java JDK) and Java Runtime Environment (JRE). The minimum versions required are Java JDK 5 and JRE 6. Java JDK is required for building application and one of the most important tools in JDK is the Java compiler which converts Java files into Java bytecode.

**2.2.3 Android SDK**

Android SDK is a collection of API libraries, tools, scripts, and documentation. This component is included in Android Studio IDE and can also be downloaded, installed as stand-alone SDK tool and it is important to set up the path for the location. The new version of SDK and tools are added to Android Studio as they become available.

This provides developers with a packaged set of developer tools and API libraries that enables building of complete application, testing of the applications on virtual devices, and performing debugging and optimization

**2.2.4 Android Studio IDE**

Android Studio is the official Integrated Development Environment (IDE) that is based on JetBrains’ IntelliJ and developed by Google specifically for building Android application. In accordance to the OS of the computer, Android Studio is available and can be downloaded from the official website of Android’s developer (<https://developer.android.com/studio/index.html>). Installation guide can be followed during the installation process after download of Android Studio.

**2.3 Application Development Process**

Application Development Process involved using some specialized tools in each phases of workflow in order to build a well designed Android application. There are five different phases of development process which are setup, write, build and run, iterate and publish. The figure below provides an overview of each phases of process to develop an Android application.

s

**SETUP**

Create a Project

Environment Setup

**WRITE**

Add assets

Write code

Customize your build

Connect to a device or emulator

**BUILD & RUN**

Debug

Profile

Test

**ITERATE**

Sign

Version

**PUBLISH**

Figure: Application Development Process. /3/

* **Setup Phase**: This is the phase where development environment is set up after downloading and installing Android Studio is already accomplished, and creation of project is also carried out in this phase.
* **Write Phase**: This phase includes writing of quality code, designing of UI, creating resources and adding assets for different types of devices.
* **Build and Run Phase**: This phase enhances android project to be built into a debuggable APK package that can be installed and run on the emulator or on an Android-powered device. It’s also includes build customization such as creating various build that produce different types of APKs from the same project.
* **Iterate Phase**: This is an iterative phase which involves writing, building and testing of Application in order to detect, eliminate bugs and optimize application performance.
* **Publish Phase**: This is the phase that makes Android application available to the users and two main tasks are carried out in this phase, which are preparing application for release and releasing application to the users. Application release preparation involves building release version of your application that users can download and install on their Android-powered devices while Application release to users involves publicizing selling, and distributing the release version of Android application to users. /4/

**2.4 Application Technologies**

This section describes the technologies required to build this project. It is important to understand the following technologies and programming languages in order to achieve the objectives of this project, Android, PHP, JSON and MySQL Database.

**2.4.1 PHP**

PHP is a general-purpose open source scripting language which is primarily designed as server-side scripting language for web development and it is an important tool for building simple, dynamic and interactive web applications. First version of PHP was produced by Rasmus Lerdorf in 1994, the PHP code is usually processed by PHP interpreter and its recursive acronym is *Hypertext Preprocessor.* /5/

PHP is freely available and can be downloaded from its official website which is <https://secure.php.net/> and can run on various operating systems such Windows and Unix-like OS. The latest stable version which is version 7.0 with release date of December 2015 is supported until December 2018. It is important to know that PHP can perform the following functionalities; /6/

* PHP can be used to create dynamic page content.
* PHP can be used to perform data encryption.
* PHP can be used for sending and receiving cookies.
* PHP can be used to manage user access.
* PHP can be used for database data manipulation such as add, delete and modify data.
* PHP can be used to collect forms data.
* PHP can perform system functions such as to create, open, read, write, delete, and close files on server.
  + 1. **JSON**

JSON which means JavaScript Object Notation is a lightweight text-based open standard that uses human-readable text to transmit data in name-value pairs. JSON is a language-independent format which was originally specified by Douglas Crockford and it is used with various modern programming languages such as PHP, Python, PERL and Java. JSON filename extension is represented by .json. /7/

JSON syntax which is a subset of JavaScript syntax specifies that JSON data is in name-values pairs, JSON data should be separated by commas, JSON objects are held inside curly braces {} and that JSON arrays are held inside square brackets [ ]. JSON values can be represented in Number, String, Boolean, Array, Object or Null data types. The example below describes JSON syntax rules; /8/

{

“teachers”: [

{“id”: “T100”, “firstName”: “Jerome”, “lastName”: “Bernard”},

{“id”: “T200”, “firstName”: “James”, “lastName”: “Colman”}

]

}

* + 1. **MySQL Database**

Database can be defined as collections of data that are well organized in order to easily access, manage and update the data and database management system (DBMS) is a software application that is designed for managing database activities. DBMS functions can be mainly classified into four which are data definition, update, retrieval, and administration. MySQL is one of the most popular DBMS. /9/

MySQL is an open source relation SQL database management system that can be used with modern programming languages such as PHP, PERL, and JAVA, and can be used on various operating systems such as Windows, Linux, Solaris, OS X and FreeBSD. It is originally created by Swedish company MySQL AB, but now owned by Oracle Corporation and can be freely downloaded from its official website <http://www.mysql.com/>. /10/

* + 1. **Android**

Evolution of Android has important impact in the area of technology in everyday life of today’s world. Android is an operating system that is based on Linux kernel and it is developed by Google. There are different types of androids which are basically named depending on different devices on which the operating system is used, such as Android Mobile which is used on Mobile devices basically mobile phones and tablets, Android TV which is used on televisions, Android Auto which is used for cars and Android Wear which is used for wearable devices. /1/

The first commercial version of Android popularly known as Android 1.0 was released in 2008 and ever since that year, each subsequent versions of Android had been released with major focus on improving performance, the user interface design and providing many features such as voice searching. /2/

Each version of Android released can be identified with code names which are organized in alphabetical order from the first commercial version with code name Alpha to the latest Android version with code name Nougat which was released in August 22, 2016. Open source licensing of Android’s source code is one of the android’s competitive advantages over competitor.

Android applications are developed with Java programming language which its platform independence made it different from other programming languages. Java is a general-purpose programming language that is concurrent, class based and object-oriented and can be run various platforms with an installed Java Runtime Environment (JRE). There is major difference between compiling and running an Android Java programs compare to compiling and running a non-Android Java program which is illustrated in the figures below; /2/

Java Files .java

JDK Compiler

Bytecode .class

Windows

Virtual Machine

Linux

Virtual Machine

MAC OS

Virtual Machine

JRE

Java Files .java

JDK Compiler

Bytecode .class

Dalvik dx

Compiler

Dalvik Code

.dex

Dalvik VM

Virtual Machine

**Non Android Java Compiled Android Java Compiled**

**3 APPLICATION DESCRIPTION**

This section provides the detailed description of the application and the expected requirements from the application. The application description is achieved by providing information related to quality function deployment, use case diagram, sequence diagram, class diagram, and component diagram of the application. This project is divided into two separate applications which are assignment tracking application for teachers used by teachers and assignment tracking application for students used by students.

Teacher application allows user which are teachers to register and login to their home page. Teacher home page with teacher’s first name and last name provides information about the teacher courses, the available assignments for each course and the students that have enrolled for each course. This application allows teacher to mark assignment for student by clicking on the course name there by providing assignment marking page for marking assignment for each student. This application allows teacher to add new course, add new assignment to each course, provides assignment completion rate for each course and provides assignment submission rate for each student.

Student application section of this project allows users which are students to register and login to their home page. Student home page with student’s first name and last name provides information about the courses the student has enrolled for and allows student to enroll for new course by clicking of enroll link there by providing course enrollment page for student to enroll for available courses. This application provides students with information about available assignments, his/her submitted assignments and assignment submission rate for each courses by clicking on the course name from his/her home page.

**3.1 Quality Function Deployment (QFD)**

Quality Function Deployment (QFD) provides means of maximizing applications users’ satisfaction and it is a quality management technique that converts needs of the customer into application technical requirements. It emphasizes on the requirements that are valuable to this project and these requirements are classified in accordance to the project overall goals and objectives. OFQ describes three types of requirements which are Normal requirements with priority level 1, Expected requirements with priority level 2, and Exciting requirements with priority level 3 and the table below provides information about this project Quality Function Deployment.

|  |
| --- |
| **Normal Requirement with priority level 1** |
| * Application must be able to display login page for students. * It must be able to display login page for teachers. * User must be able to register by providing username, password and other information. * User must be able to login with the username and password provided during registration. * User must be able to logout from his/her home page. * Student must be able to enroll for available courses * Student must be able to see courses he/she has enrolled for. * Student must be able to see available assignments on courses he/she has enrolled for. * Student must be able to see his/her submitted assignments on courses he/she has enrolled for. * Teacher must be able to add new course. * Teacher must be able to add assignment for each course. * Teacher must be able to mark assignment for student. * Teacher must be able to see all courses he/she has added * Teacher must be able to see all assignments for his/her course * Teacher must be able to see completion rate for each course assignment. * Teacher must be able to see all students that have enrolled for his course. * Teacher must be able to see submission rate of enrolled students for each course. |
| **Expected Requirement with priority level 2** |
| * User should be able to see registration response if registration fails. * User should be able to see login response if login fails. * Teacher should be able to see assignment marking feedback message. * The application should be interactive, easy to use and user friendly. * The application should be able to install and run on all Android Mobile devices with OS minimum version 3.0 (Honeycomb) to latest version 7.0 (Nougat). |
| **Exciting Requirement with priority level 2** |
| * The application can allow users to reset their password. * The application can provide student with teacher information such as teacher email address and phone number. * The application can provide information about the educational institution using the application such as school website and school address. |

Table: Quality Function Deployment.

**3.2 Use Case Diagram**

Use Case diagram provides information about the functions of the application and the interaction between actors in the application. It shows the relationship between functions in the application. The actors in this project are student and teacher.

**3.2.1 Student Use Case**

The student use case provides the functionalities that involve student interaction with the system. The figure below represents application student use case diagram.

**STUDENT**

Figure: Student Use Case Diagram

**3.2.2 Teacher Use Case**

The teacher use case provides the functionalities that involve teacher interaction with the system. The figure below represents application teacher use case diagram.

**TEACHER**

Figure: Teacher Use Case Diagram

**3.3 Sequence Diagram**

Sequence diagram provides detailed step by step description of the application use case diagrams functionalities and it helps to provide better understanding of the application functionalities.

Sequence diagram is an interactive diagram that shows how and the order in which objects in an application interact with one another in time sequence. The sequence diagrams for this project are shown in the figures below;

**3.3.1 Registration Sequence Diagram for Users (Student and Teacher)**

**DBHandler**

**Object**

ELSE (Write Data to DB)

Display error message

Return Response

Check data in DB

Enter registration data and Click REGISTER

Redirect

Click on Register link

Call DBHandler

**USER**

**Login Page**

**Activity**

**Registration Page**

**Activity**

**<<Database>>**

**DB**

IF (Data Exists)

Return Confirm Message

Display Confirm Message

**3.3.2 Login Sequence Diagram for Users (Student and Teacher)**

Else Login success and Redirect

Display login error message

Return Response

Checks data in DB

Enter Login Data and click LOGIN button

Call DBHandler

**USER**

**Login Page**

**Activit**

**DBHandler**

**Object**

**<<Database>>**

**DB**

IF (Data Does not exists)

**User Home Page**

**Activity**

**3.3.3 Logout Sequence Diagram for Users (Student and Teacher)**

Redirect to Login Page

Enter Login Data and click LOGIN button

Redirect

**USER**

**Login Page**

**Activity**

**User Home Page**

**Activity**

Create a Projec

Click on Logout Link

**3.3.4 Course Enrollment Sequence Diagram (Student)**

Click on Enroll Link

**DBHandler**

**Object**

ELSE Write Date to DB

Display error message

Return Response

Checks data in DB

Enter Course Enrollment Data and Click ENROLL

Redirect

Enter Login data and click LOGIN button

Call DBHandler

**STUDENT**

**Login Page**

**Activity**

**Home Page**

**Activity**

**<<Database>>**

**DB**

IF (Data Exists)

Return Confirm Message

Display Confirm Message

**Enrollment Page**

**Activity**

Redirect

**3.3.5 Add Assignment Sequence Diagram (Teacher)**

Redirect

Click on Add New Course Link

**DBHandler**

**Object**

ELSE Write Date to DB

Display error message

Return Response

Checks data in DB

Enter Course Data and Click Add Course button

Redirect

Enter Login data and click LOGIN button

Call DBHandler

**TEACHER**

**Login Page**

**Activity**

**Home Page**

**Activity**

**<<Database>>**

**DB**

IF (Data Exists)

Return Confirm Message

Display Confirm Message

**Add Course Page**

**Activity**

**3.3.6 Add Course Sequence Diagram (Teacher)**

Click on Add New Assignment Link

**DBHandler**

**Object**

ELSE Write Date to DB

Display error message

Return Response

Checks data in DB

Enter Assignment Data and Click on Add Assignment button

Redirect

Enter Login data and click LOGIN button

Call DBHandler

**TEACHER**

**Login Page**

**Activity**

**Home Page**

**Activity**

**<<Database>>**

**DB**

IF (Data Exists)

Return Confirm Message

Display Confirm Message

**Add Assign. Page**

**Activity**

Redirect

**3.3.7 Mark Assignment Sequence Diagram (Teacher)**

**DBHandler**

**Object**

**Login Page**

**Activity**

**Home Page**

**Activity**

**<<Database>>**

**DB**

**Marking Page**

**Activity**

Click on Course Name

ELSE Write Date to DB

Display error message

Return Response

Checks data in DB

Enter Required Data and Click on MARK ASSIGNMENT

Redirect

Enter Login data and click LOGIN button

Call DBHandler

**TEACHER**

IF (Data Exists)

Return Confirm Message

Display Confirm Message

Redirect

**3.3.8 Display Student Submission Rate Sequence Diagram (Teacher)**

Click on STUDENTS button

Click on Student Number to Display his submission rate

Redirect

Enter Login data and click LOGIN button

**TEACHER**

**Login Page**

**Activity**

**Home Page**

**Activity**

**Course Students Page**

**Activity**

Redirect

**Student Assignment Page**

**Activity**

Redirect

**3.3.9 Display Student Submission Rate Sequence Diagram (Student)**

Select Course

Redirect

Enter Login data and click LOGIN button

**STUDENT**

**Login Page**

**Activity**

**Home Page**

**Activity**

Redirect

**Student Assignment Page**

**Activity**

**3.3.10 Display Assignment Completion Rate Sequence Diagram (Teacher)**

Click on ASSIGNMENTS button

Redirect

Enter Login data and click LOGIN button

**TEACHER**

**Login Page**

**Activity**

**Home Page**

**Activity**

Redirect

**Course Assignment Page**

**Activity**

**3.4 Class Diagram**

Class diagram provides information about the classes in the application with their responsibilities and shows application collaboration which is the interaction between classes. Figure below shows this application class diagram;

-id: String

-name: String

-description: String

-startDate: Date

-endDate: Date

-password: String

+getId(): String

+setId(): void

+getName(): String

+setName(): void

+getDescription: String

+setDescription(): void

+getStartDate(): Date

+setStartDate(): void

+getEndDate: Date

+setEndDate: void

+getPasswrod: String

+setPassword: String

**COURSE**

-id: String

-description: String

-startDateTime: Date

-endDateTime: Date

-completionRate: Double

+getId(): String

+setId(): void

+getDescription(): String

+setDescription(): void

+getStartDateTime(): Date

+setStartDateTime(): void

+getEndDateTime(): Date

+setEndDateTime(): void

+getCompletionRate: Double

+setCompletionRate: void

**ASSIGNMENT**

-userName: String

-password: String

-firstName: String

-lastName: String

-submissionRate: Double

+getStudentNumber(): String

+setStudentNumber(): void

+getPassword(): String

+setPassword(): void

+getFirstName(): String

+setFirstName(): void

+getLastName(): String

+setLastName(): void

+getSubmissionRate: Double

+setSubmissionRate: void

**STUDENT**

-userName: String

-password: String

-firstName: String

-lastName: String

+getUserName(): String

+setUserName(): void

+getPassword(): String

+setPassword(): void

+getFirstName(): String

+setFirstName(): void

+getLastName(): String

+setLastName(): void

**TEACHER**

-courseId: String

-courseName: String

+onCreate(): void

+onAddAssignment(): void

+setDateTimeParameter(): void

+onClick(): void

+onLogout(): void

+onHomePage(): void

**ADDASSIGNMENT**

+onCreate(): void

+onAddCourse(): void

+setDateTimeParameter(): void

+onClick(): void

+onLogout(): void

+onHomePage(): void

**ADDCOURSE**

-username: String

-password: String

-firstname: String

-lastname: String

+onCreate(): void

+onRegister(): void

**REGISTER**

-courseId: String

-assignmentId: String

-studentNumber: String

-teacherId: String

+onCreate(): void

+onMark(): void

**MARK ASSIGNMENT**

-json: String

-jObj: JSONObject

-urlLink: String

-jArray: JSONArray

+doInBackground(): void

+onPreExecute(): void

+onPostExecute(): void

+onProgressUpdate(): void

**ASYNC TASK**

-courseId: String

-coursePwd: String

-courseName: String

-studentNumber: String

-firstname: String

-lastname: String

+onCreate(): void

+onEnroll(): void

**ENROLLMENT**

-id: String

-password: String

+onCreate(): void

+onLogin(): void

+onRegistration(): void

**LOGIN**

**Class Diagram**

**3.5 Component Diagram**

Component diagram describes how an application system is divided into sub-systems; show the services these sub-systems provide and their relationships with other sub-systems. This Android application was built on Model-View-Controller architecture and the components involved in this application are Model, View and Controller. The figure below represent mode of communication between Model, View and Controller objects. /2/

**MODEL**

**(Data)**

**CONTROLLER**

**VIEW**

**(UI elements)**

Notify

Update data

User Action

Update UI

* + 1. **Model**

This part of the components is used to define the logic and computation that manipulate and process data in an application. It provides the objects that are involved in this application such student, teacher, course and assignment objects.

* + 1. **View**

This part of the components represents visual display objects and provides the user interface that enable users to interact with the application, such as teacher views and student views. For instance, Text View element represent View object that displays text on user interface and Button element represent View object that can respond to user actions.

* + 1. **Controller**

This part of the components provides the main control of the application. It receives user’s information from user interfaces and sends the information to the objects for processing of the information. Activities in Android application are an example of controller objects.

Figure below shows this application component diagram;

**VIEW**

**MODEL**

**CONTOLLER**

**Student**

**DBHandler**

**Course**

**Assignment**

**Teacher**

**Teacher** **View**

**Student View**

**Teacher Activity**

**Student Activity**

**4 DATABASE**

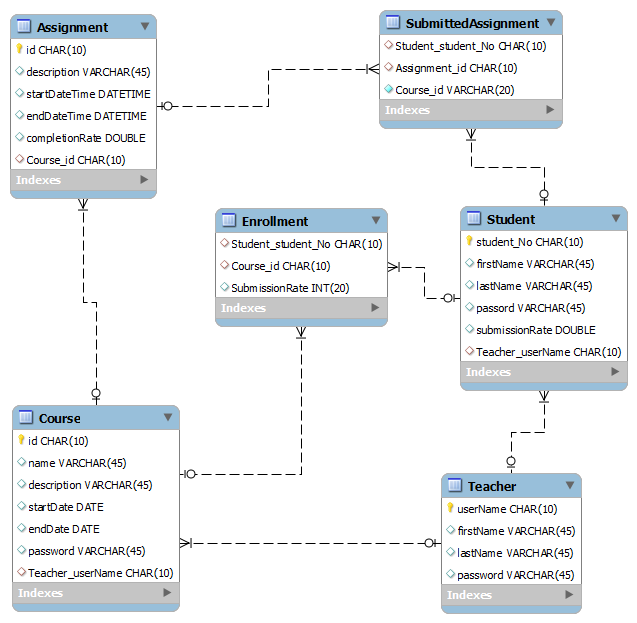
This application is using remote online database server and this section provides information about the design of the application’s database, the effort required for effective design of database and the ways of communication between the database systems.

**4.1 Design of Database**

The design of MySQL database used for this project was achieved with Database Management System (DBMS) tool which is MySQL workbench and after the design of database; the database with the name e1100617\_AssignmentTrackingApp is located inside remote school server of VAMK and can be checked on the website address ([www.mysql.cc.puv.fi](http://www.mysql.cc.puv.fi)).

The design of MySQL database basically involved creating the tables that will be needed for effective performance of the application, identifying the meta-data, unique identity data and data type required by each tables and identifying the type of relationships that exist between the tables in the application. There are situation where by the type of relationships between tables lead to creation of another table, such as the relationship between student table and course table in this application lead to creation of enrollment table. This application database consists of six tables which student, teacher, course, assignment, submitted assignment and enrollment tables. The relationship between tables is presented in entity relationship diagram and the figure below represents this project ER diagram;

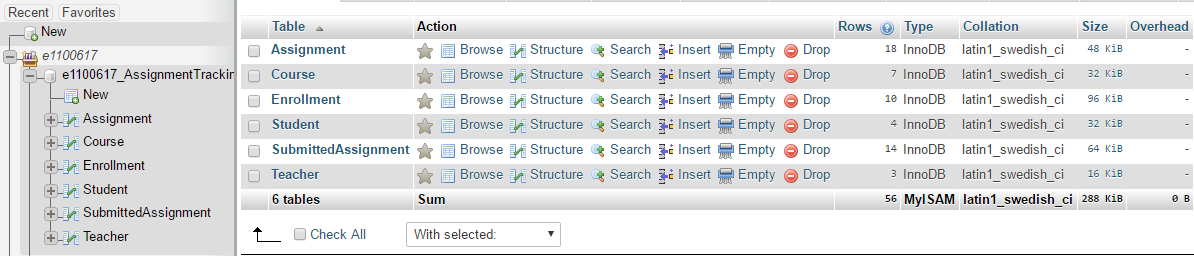
The figure below shows that course table with primary key id has one-to-many relationship with the assignment table with primary key id; this means that one course can have many zero to many assignments and not vice-versa.

****

The figure above shows that the teacher table with primary key userName has one-to-many relationship with student table with primary key of student\_No, while teacher table has one-to-many relationship with course table which has primary key as id and this means that in this project, one teacher can have many zero to many students and zero to many courses and not vice-versa.

The course table has many-to-many relationship with the student table, which means that one student can have zero to many courses and one course can have zero to many students. This type of relationship is not acceptable in relational database design and this lead to creation of bridge table which is enrollment table that created one-to-many relationship between course table and enrollment table and one-to-many relationship between student table and enrollment table. In the same way, submitted assignment table represents the bridge table which was created due to many-to-many relationship that existed between assignment table and student table.

The figure below shows the application database tables created from the ER diagram after the design of database.

****

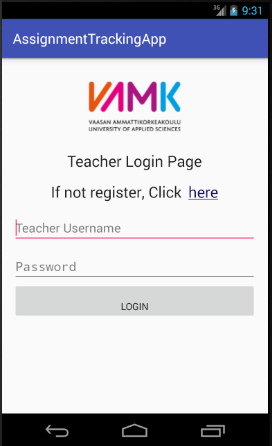
The server side of this application is implemented with PHP scripts and these PHP files are stored inside directory called AssignmentTrackingApp in the public\_html directory (<http://www.cc.puv.fi/~e1100617/AssignmentTrackingApp/>). Making connection to database with PHP scripts requires the database hostname, username and password. All tasks that require communication to database are written with PHP scripts, For example, the PHP script that communicates with database when teacher add new course is addCourse.php and after teacher has entered required data and press register button, the android application with post method calls URL: <http://www.cc.puv.fi/~e1100617/AssignmentTrackingApp/addCourse.php> to communicate with database and return the result as JSON format to the application.

**5 GRAPHICAL USER INTERFACE DESIGN**

This section describes the user interfaces involved in this application and XML language is used for designing user interface in android application.XML files that represent all the user interfaces are located inside layout directory of the application package. The following figures represent all UI in this application with their names.

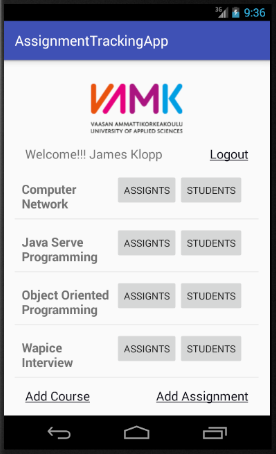
**5.1 Teacher Login Page**

The teacher login page which was designed with xml, allows teachers to login to teacher home page by providing their registered teacher id, password and click on login button. This page provides clickable link which enables teacher that has not registered to register, and by clicking on the link, the application provides teacher with registration form to fill in his/her data and register. The login page did not accept null data input and there is an invisible textbox that will be visible and show message to the user incase of login error. The figure below represents teacher login page.



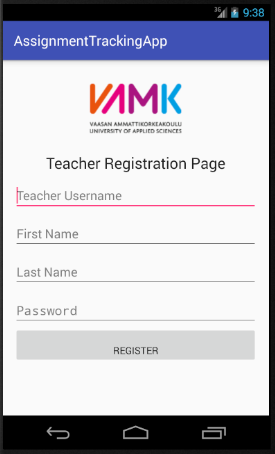
**5.1.1 Teacher Home Page**

After teacher has successfully login from teacher login page, the application provides teacher with teacher home page which was designed with xml and this interface provides the teacher first name and last name to show that user is working on the appropriate page. The teacher home page provides information about the teacher’s courses names presented in clickable text, assignments for each course presented in clickable button and students for each course presented in clickable button. There are also clickable text named add new course which allows teacher to add new course, clickable text named add new assignment which allows teacher to add new assignment to course and clickable logout button that enables teacher to logout. The figure below represents teacher home page.



**5.1.2 Teacher Registration Page**

This page allows teacher to provide required data and click on register button to register. If teacher login attempt was not successful in login page and it happens that the teacher has no registered teacher id and password, teacher can click on clickable registration link from login page which will take the teacher to teacher registration page. There is an invisible textbox that will be visible and show message to the user incase of registration error. The figure below represents teacher registration page.



**5.1.3 Assignment marking Page**

This page allows teacher to mark assignment for student. All courses in teacher home page has clickable names which provides assignment marking page when teacher clicks on the course name in order to mark assignment for students. This page provides student id of all students that have enrolled for the particular clicked course and allows teacher to select student id, provides all assignments for the clicked course and allows teacher to select assignment to mark and click on mark assignment button in order to mark assignment. It has an invisible textbox that will be visible and show message to the user incase of marking error. The figure below represents assignment marking page.



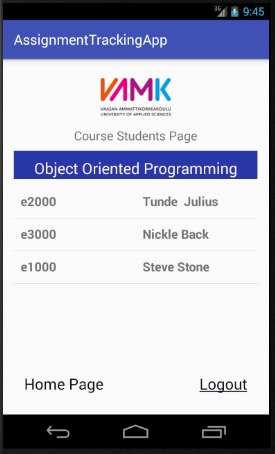
**5.1.4 Course Assignments Page**

This page appears to teacher when teacher clicks on assignment button on his/her home page and it provides teacher information about course name, list of assignments for each course, the due date time for each assignment and the completion rate for each assignment. There is clickable logout text which allows teacher to logout. The figure below represents course assignments page.



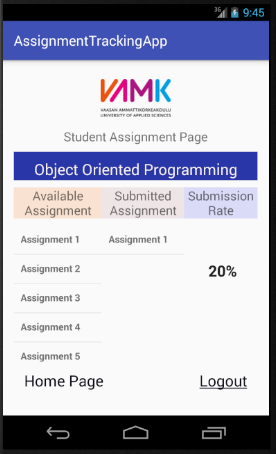
**5.1.5 Course Students Page**

This page appears to teacher when teacher clicks on student button on his/her home page and it provides teacher information about course name, student id, student first name and student last name of all students that have enrolled for particular course. Each row of student information on this page links to a page which is student assignment page that gives information about particular student assignments when teacher clicks on the student id. There is clickable logout text which allows teacher to logout. The figure below represents user interface for course students.



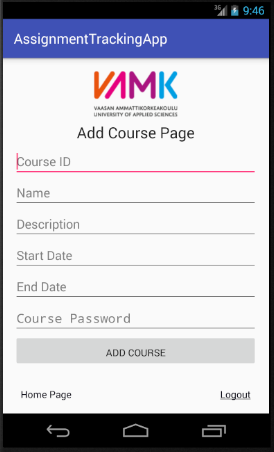
**5.1.5 Student Assignment Page**

Student assignment page provides teacher with information about course available assignments, submitted assignments and submission rate for particular clicked student id, student first name and last name on user interface for course students. The figure below represents student assignment page.



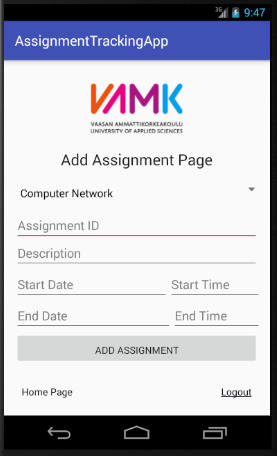
**5.1.6 Add Course Page**

It appears to teacher when teacher clicks on add new course clickable text on his/her home page. This page allows teacher to add new course to his/her course lists by providing required data such as course id, course name, description, start date, end date, course enrollment password and click on add course button in order to add new course. There is an invisible textbox that will be visible and show message to the user incase of error.



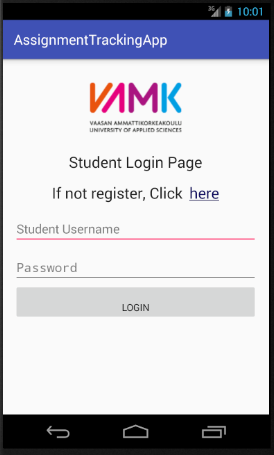
**5.1.7 Add Assignment Page**

This page appears to teacher when teacher clicks on add new assignment clickable text on his/her home page. It allows teacher to add new assignment to his/her assignment lists for selected course name by providing required data such as course id which is automatically selected in the background when user select course name, assignment id, description, start date time, end date time and click on add assignment button in order to add new assignment. There is an invisible textbox that will be visible and show message to the user incase of error.



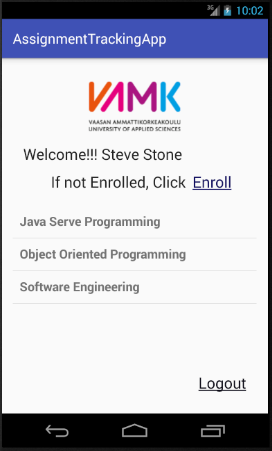
**5.2 Student Login Page**

This page was designed with xml and allows students to login to student home page by providing their registered username, password and click on login button. This page provides clickable text link which enables student that has not registered to register, and by clicking on the link, the application provides student with registration form to fill in his/her data and click on register. The login page did not accept null data input and there is an invisible textbox that will be visible and show message to the user incase of login error. The figure below represents student login page.



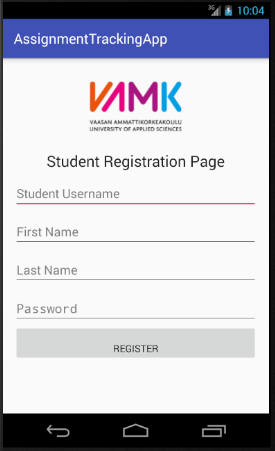
**5.2.1 Student Home Page**

This page appears to student after student has successfully login from student login page and this page was designed with xml. This interface provides the student first name and last name to show that user is working on the appropriate page. The student home page provides information about clickable courses names which student has enrolled in. There is clickable enroll text that enables student to enroll for courses, and by clicking on the link, the application provides student with course enrollment form to fill in his/her data and click on enroll. There is clickable logout text that enables student to logout and the figure below represents student home page.



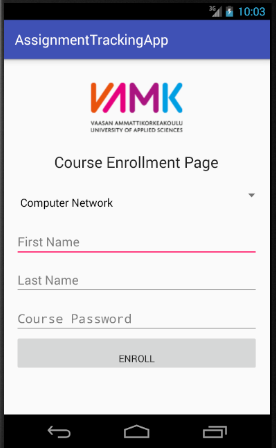
**5.2.2 Student Registration Page**

This page allows student to register by provide required data and click on register button. If student login attempt was not successful in login page and it happens that the student has no registered username and password, student can click on clickable registration link from login page which will take the student to student registration page. There is an invisible textbox that will be visible and show message to the user incase of registration error. The figure below represents student registration page.



**5.2.3 Course Enrollment Page**

This page appears when student click on clickable enroll text on his/her home page and it was designed with xml. It allows student to enroll for course by providing required data such as course id which is automatically selected as the student select course name, student username which is programmatically provided as username that student used for successful login , student first name, student last name, course password and click on enroll button. There is an invisible textbox that will be visible and show message to the student in case of course enrollment error. The figure below represents course enrollment page.



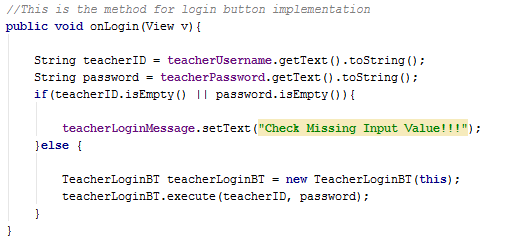
**6 IMPLEMENTATION**

This section describes the implementation of all the graphical user interfaces involved in this application. Implementation is the realization of application specification, idea and design. It involves programming codes written in order to achieve ideas and requirements of each user interface of the application. Implementation of each user interfaces involved XML codes for user interface design, ANDROID JAVA codes for back end and PHP codes for server side. The following figures represent the application code snippets.

**6.1 Login**



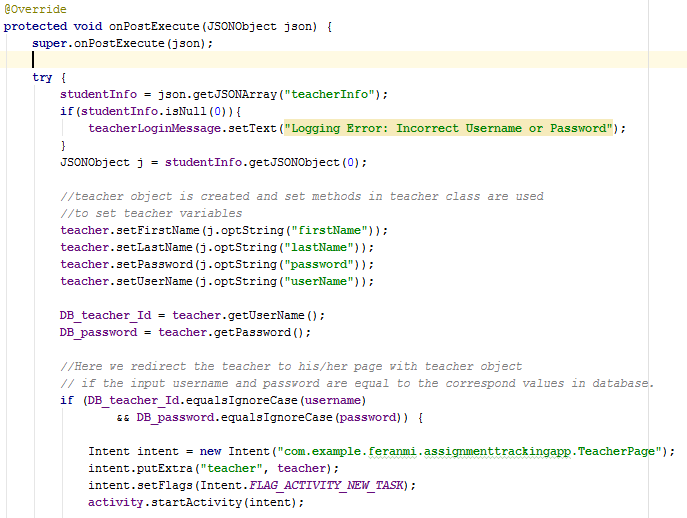
**Code Snippet 1: On create method for login**



**Code Snippet 2: On login method**

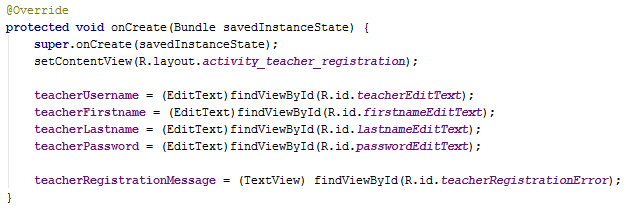


**Code Snippet 3: Login background task class and method**

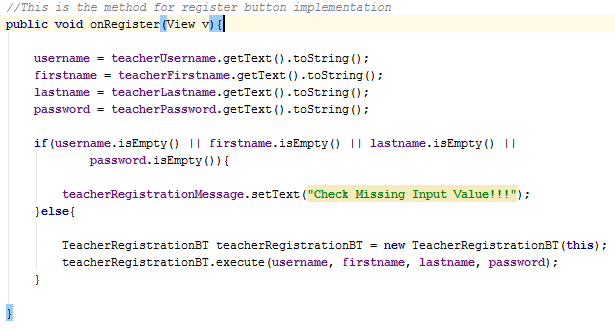


**Code Snippet 4: Login on post execute method**

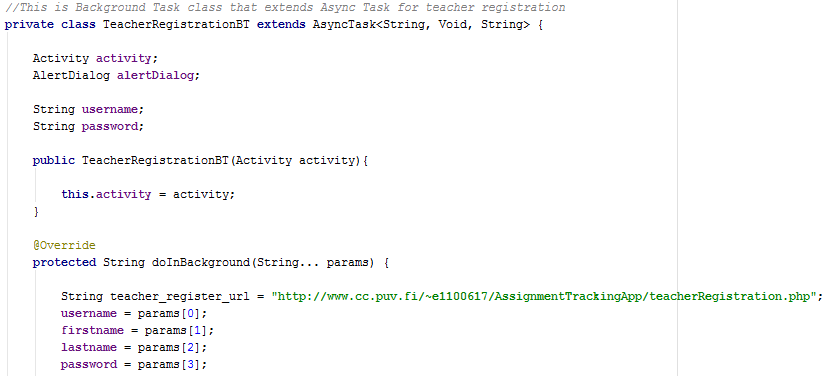
**6.2 Registration**



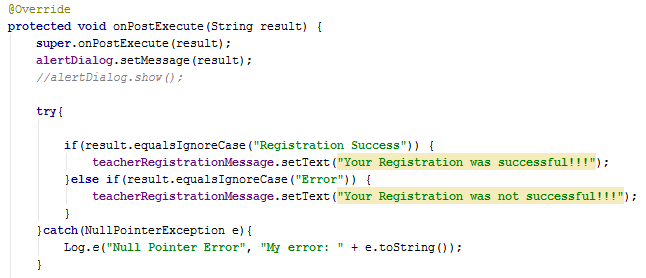
**Code Snippet 5: On create method for registration**



**Code Snippet 6: On register method for register button**

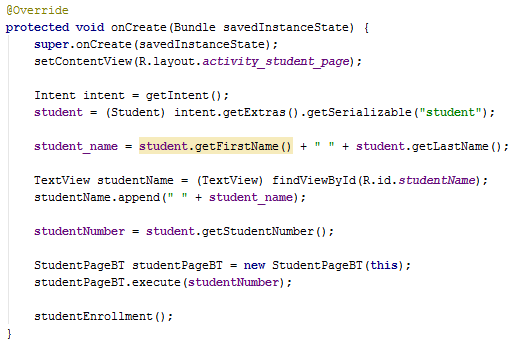


**Code Snippet 7: Registration background task class and method**



**Code Snippet 8: Registration on post execute method**

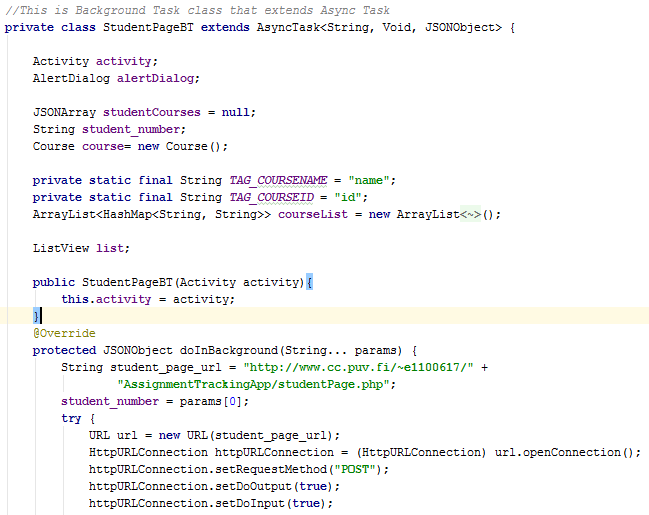
**6.3 Student Home**



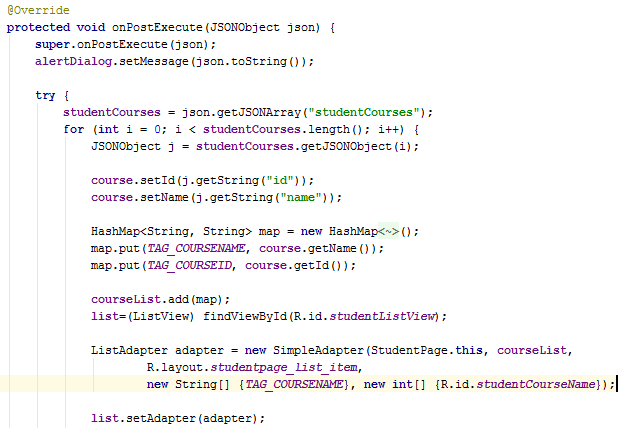
**Code Snippet 9: On create method for student home page**



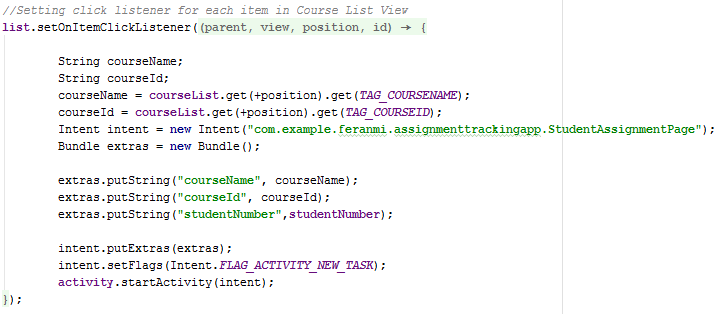
**Code Snippet 10: Method that redirect to student enrollment page**



**Code Snippet 11: Student page background task class and method**



**Code Snippet 12: Student page on past execute method**

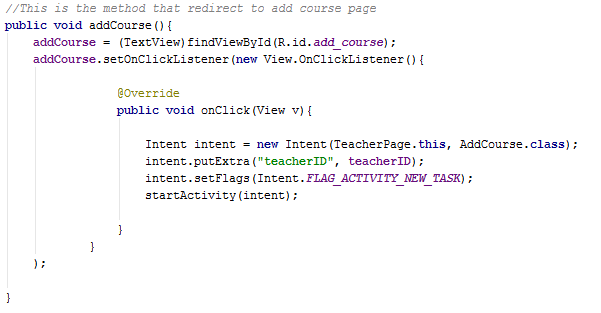


**Code Snippet 13: On click listener for each course on student page**

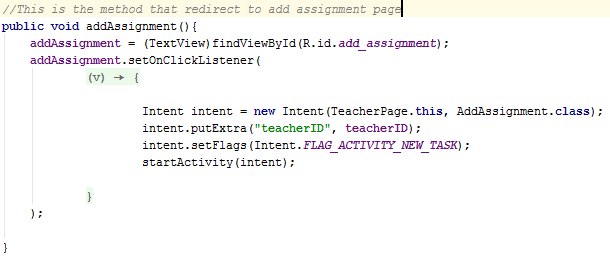
**6.4 Teacher Home**



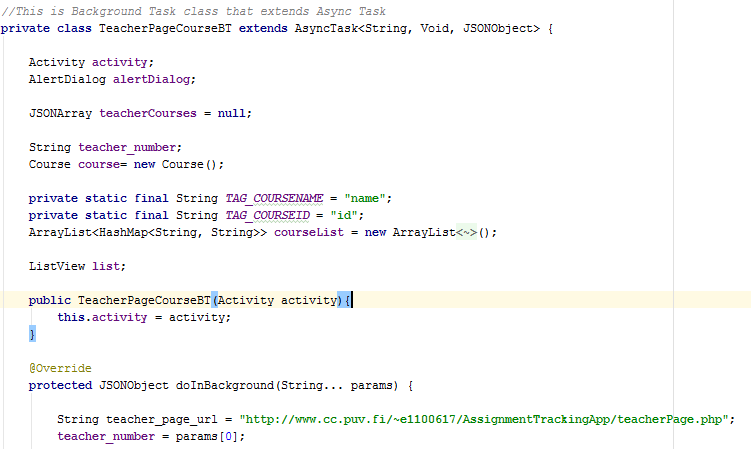
**Code Snippet 14: On create method for teacher home page**



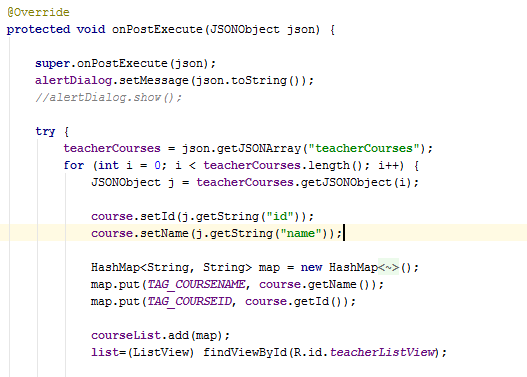
**Code Snippet 15: Method that redirect to add course page**



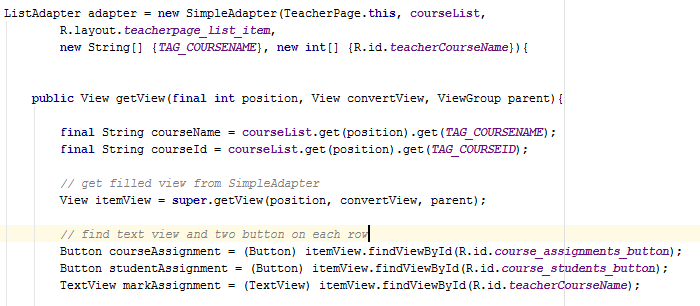
**Code Snippet 16: Method that redirect to add assignment page**



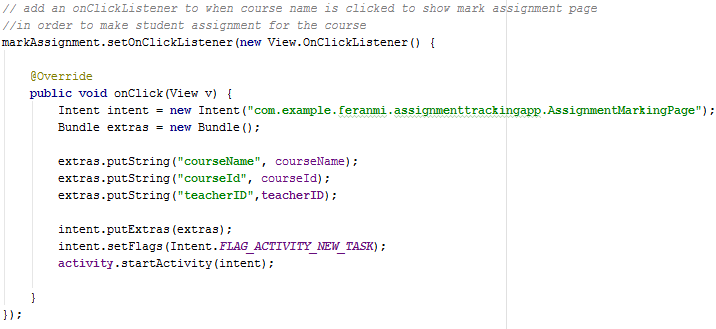
**Code Snippet 17: Teacher page background task and method**



**Code Snippet 18: On post execute method and handling JSON array**



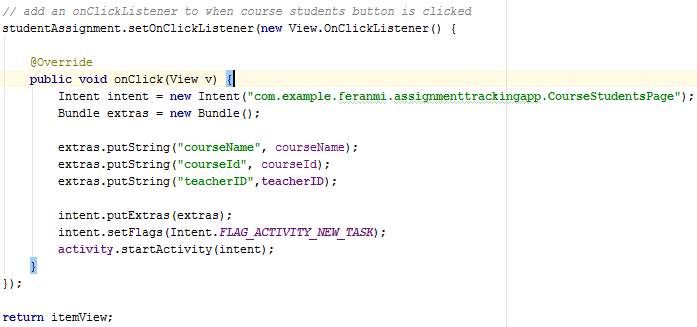
**Code Snippet 19: Setting list adapter for each row in teacher page**



**Code Snippet 20: Setting click listener for course and redirect to mark page**



**Code Snippet 21: Setting click listener and redirect to assignment page**

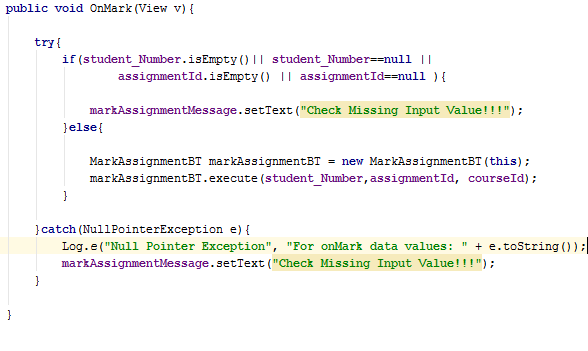


**Code Snippet 22: Setting click listener and redirect to student page**

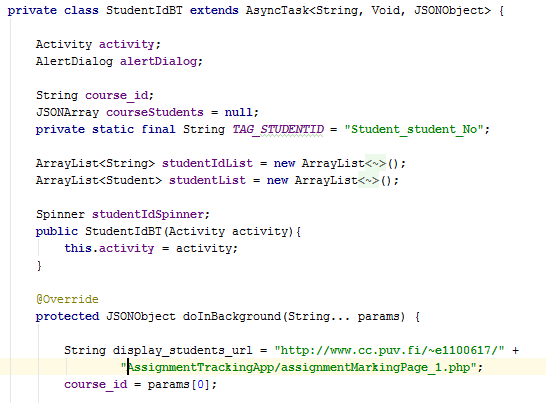
**6.5 Assignment marking**



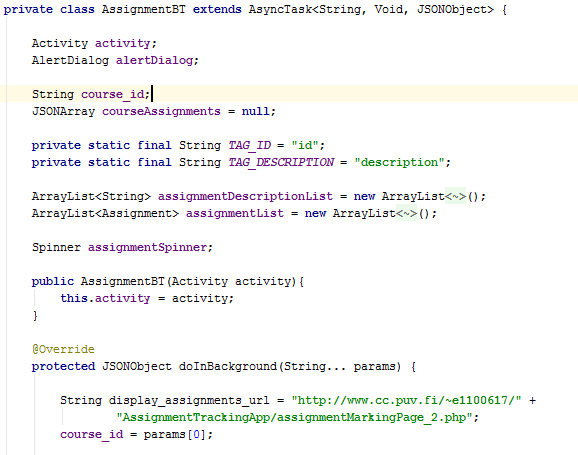
**Code Snippet 23: On create method for assignment marking**



**Code Snippet 24: Method for marking assignment**



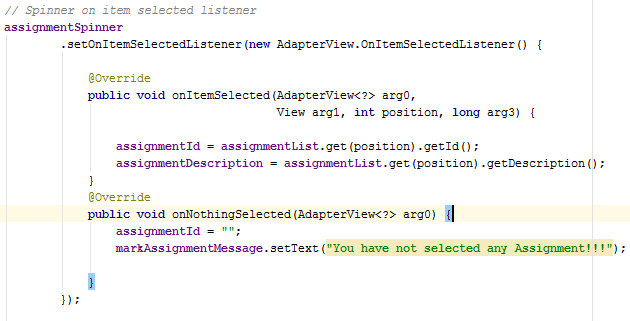
**Code Snippet 25: Background task class and method that show student id**



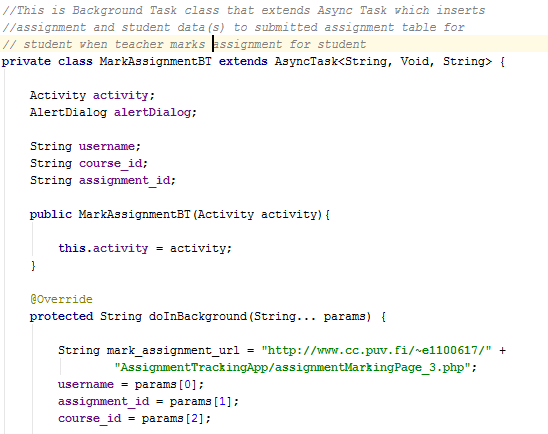
**Code Snippet 26: Background task class and method that show assignments**



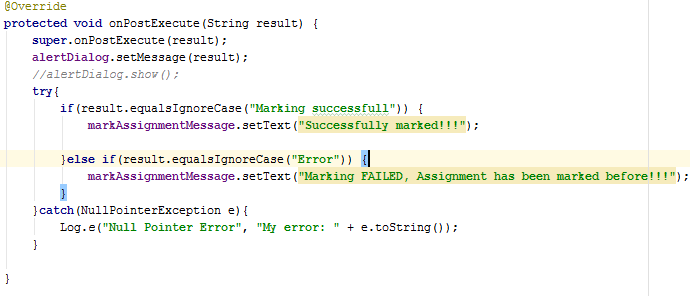
**Code Snippet 27: This populate spinner with assignment description**



**Code Snippet 28: Spinner for selected assignment**

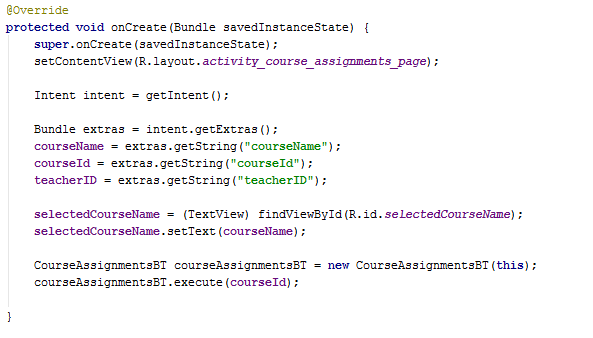


**Code Snippet 29: Background task class and method for marking assignment**

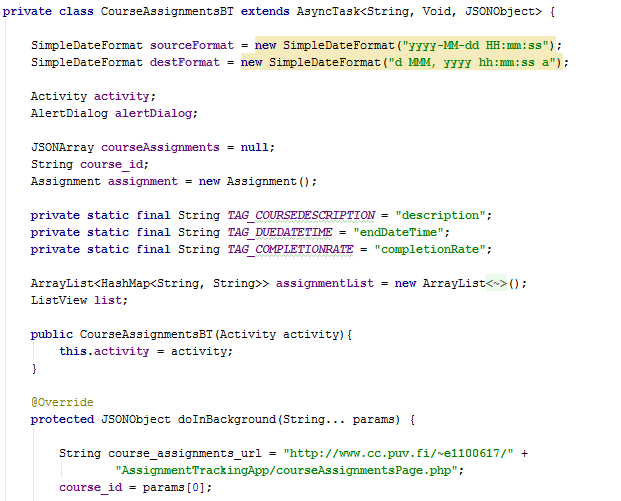


**Code Snippet 30: On post execute method for marking result**

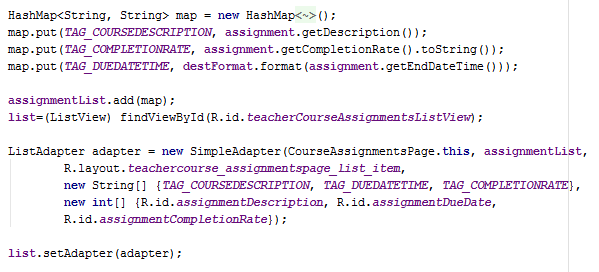
**6.6 Course Assignments**



**Code Snippet 31: On create method for course assignments**

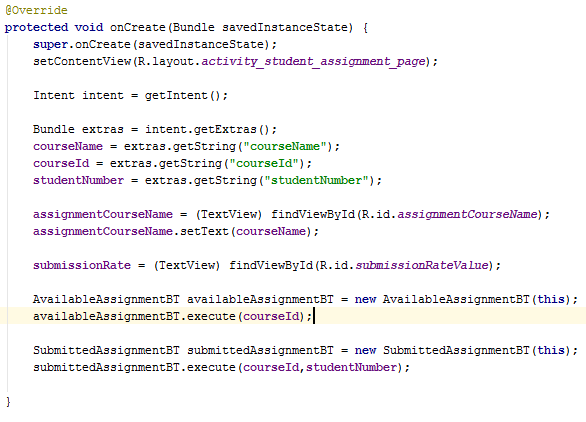


**Code Snippet 32: Background task class and method for course assignment**

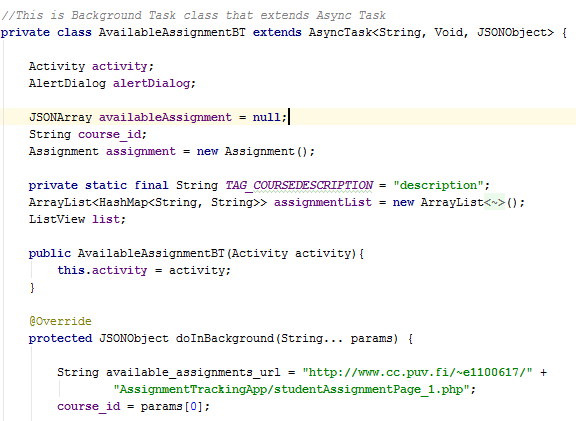


**Code Snippet 33: Setting list adapter for course assignment**

**6.7 Student Assignment**



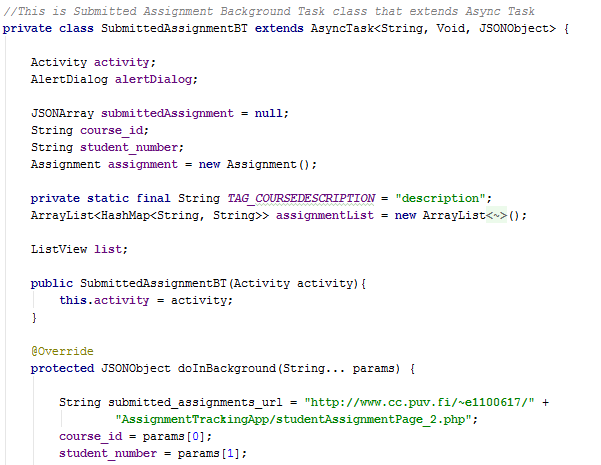
**Code Snippet 34: On create method for student assignment**



**Code Snippet 35: Background task class and method for available assignment**



**Code Snippet 36: Setting list adapter for available assignment**



**Code Snippet 37: Background task class and method for submitted assignment**



**Code Snippet 38: setting list adapter for submitted assignment**

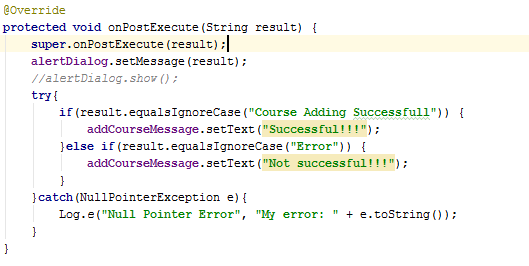
**6.8 Add Course**



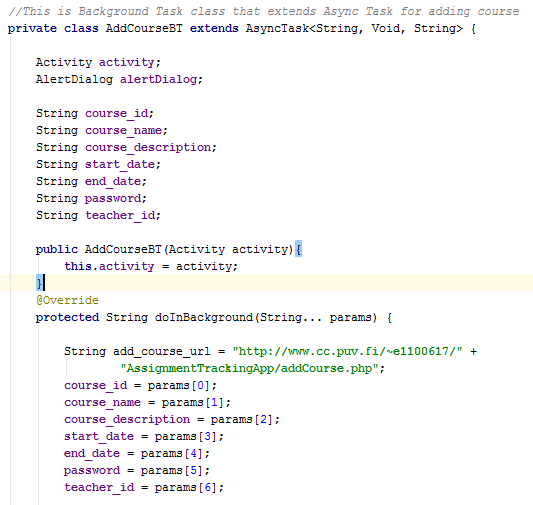
**Code Snippet 39: On create method for adding course**



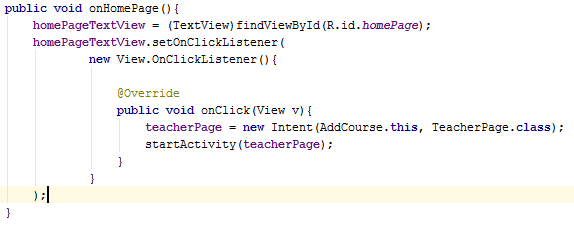
**Code Snippet 40: Method for add course button**



**Code Snippet 41: On post execute method for add course result**



**Code Snippet 42: Background task class and method for add course**

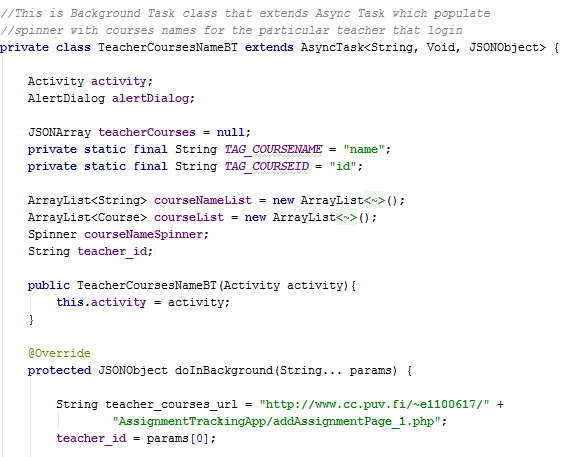


**Code Snippet 43: Method for returning to home page**

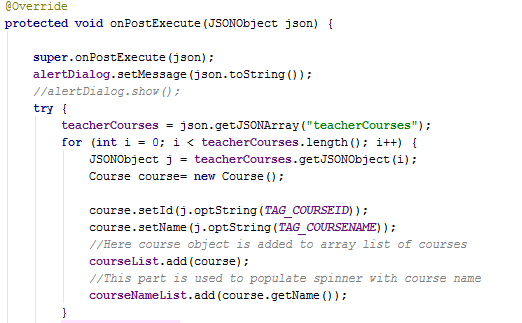
**6.9 Add Assignment**



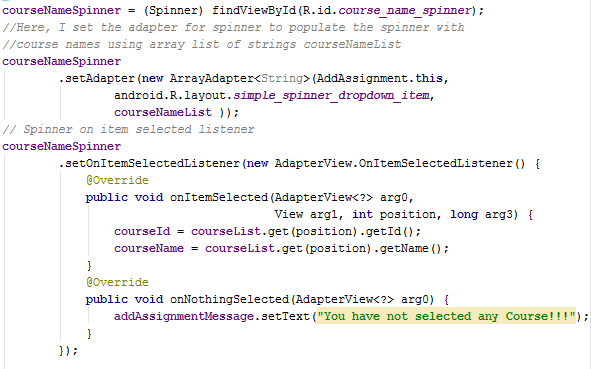
**Code Snippet 44: On create method for add assignment**



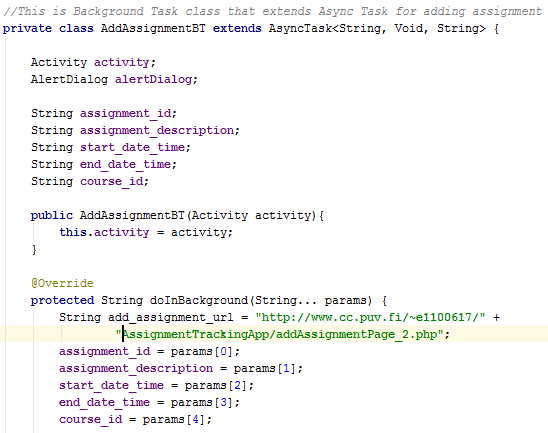
**Code Snippet 45: Background task class and method for course names**



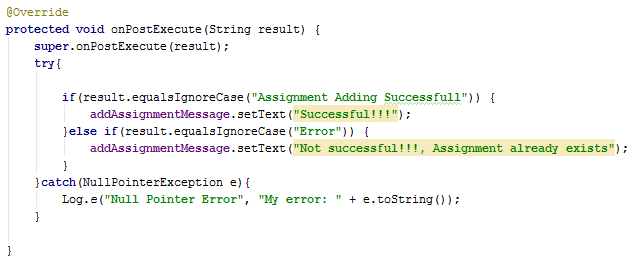
**Code Snippet 46: On post execute method for teacher courses**



**Code Snippet 47: Spinner for course names**



**Code Snippet 48: Background task class and method for add assignment**

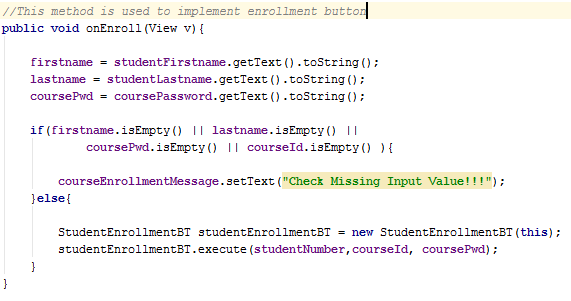


**Code Snippet 49: On post execute method for add assignment result**

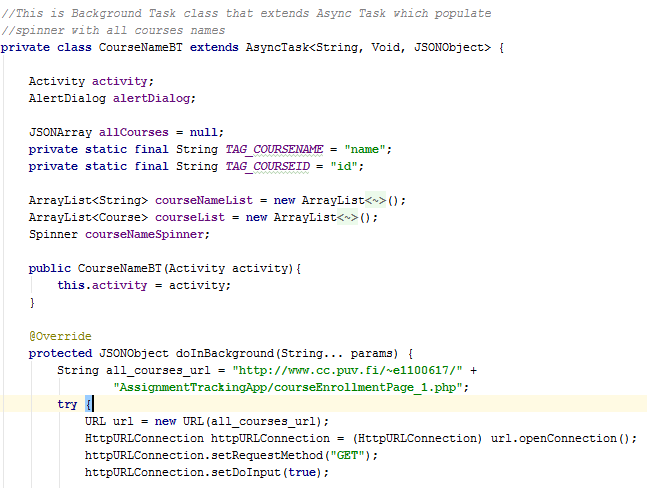
**6.10 Course Enrollment**

****

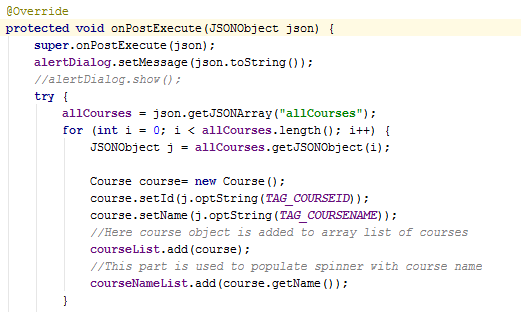
**Code Snippet 50: On create method for course enrollment**

****

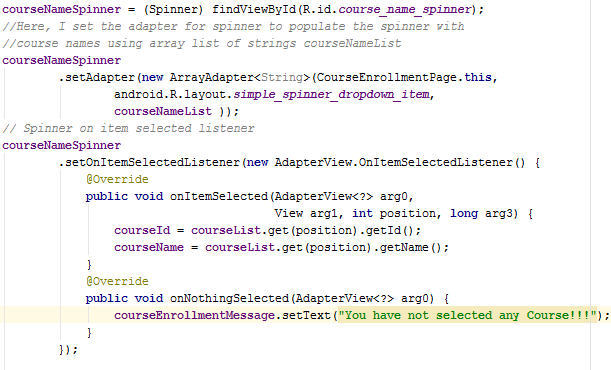
**Code Snippet 51: Method for enroll button**

****

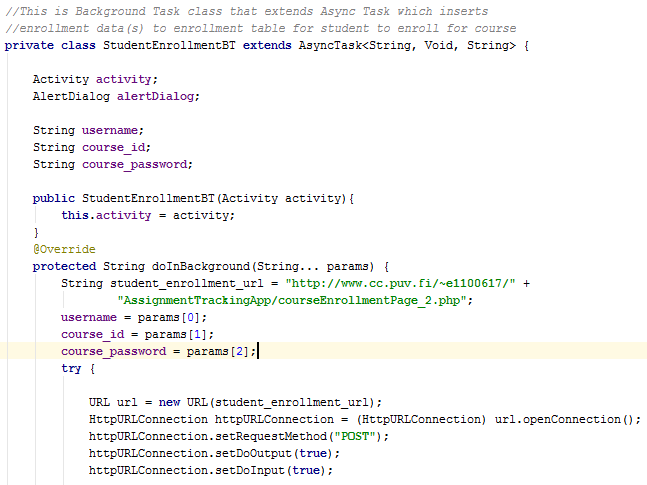
**Code Snippet 52: Background task class and method for course names**

****

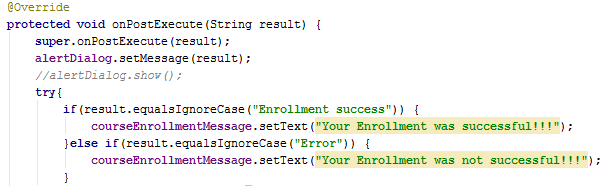
**Code Snippet 53: On post execute method for all courses**

****

**Code Snippet 54: Setting adapter for course names spinner**

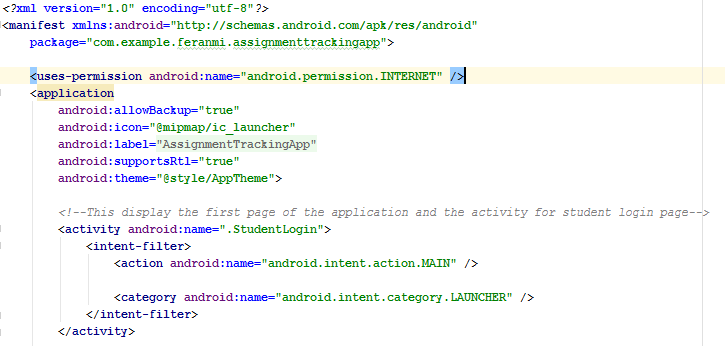
****

**Code Snippet 55: Background task class and method for enrollment**

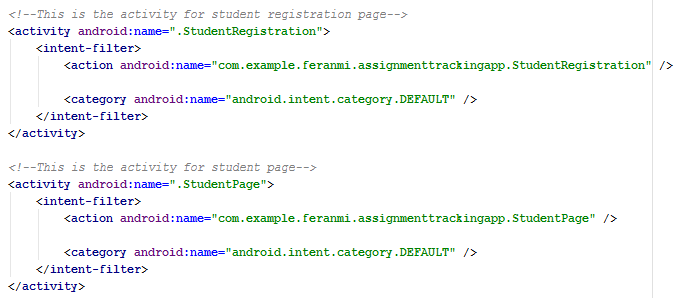
****

**Code Snippet 56: On post execute method for enrollment result**

**6.10 Android Manifest**

****

**Code Snippet 57: Manifest with internet permission and launcher**

****

**Code Snippet 58: Manifest with activities names and intents**

**7 TESTING**

This describes the process of running and executing software application or software product in order to find bugs in the software. Software testing can be defined as the process of validating and verifying that software program meets business needs and technical requirements that guided the design and development of the software product. The client side testing of this project was carried out on Galaxy Express Samsung mobile phone and the following figure represents the testing template used and the result of the testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Test Case Description** | **Steps** | **Expected Result** | **Actual Result** | **(Pass/Fail)** |
| 1. | Check for necessary fields and buttons on login page. | 1. Installed the application. 2. Runs the application and view login page. | The login page should have text fields for username and password and has login button and clickable text for redirect to registration page. | OK | Pass |
| 2. | Check for empty validation on login page. | 1. Click on login button without entering any values for username and password. | The login page should display error; check missing inputs. | OK | Pass |
| 3. | Check for Incorrect inputs on login page. | 1. Enter incorrect value for either username or password 2. Click on login button. | The login page should display error; Incorrect username or password. | OK | Pass |
| **No** | **Test Case Description** | **Steps** | **Expected Result** | **Actual Result** | **(Pass/Fail)** |
| 4. | Check for correct inputs on login page. | 1. Enter correct value for username and password 2. Click on login button. | The login page should redirect to home page. | OK | Pass |
| 5. | Check for clickable text on login page. | 1. Click on register link on login page. | The user should be redirected to registration page. | OK | Pass |
| 6. | Check for empty inputs on registration page. | 1. Click on register link on login page. 2. Click on register button on registration page without inputting all necessary data. | The registration page should display error; check missing inputs. | OK | Pass |
| 7. | Multiple entries of data for registration. | 1. Click on register link on login page. 2. Enter existing registration data 3. Click on register button. | The login page should display error; registration error, already exist. | OK | Pass |
| 8. | Check for logout clickable text. | 1. User login to his/her home page. 2. Click on logout link. | The user should be redirected to login page. | OK | Pass |
| **No** | **Test Case Description** | **Steps** | **Expected Result** | **Actual Result** | **(Pass/Fail)** |
| 9. | Check for homepage clickable text. | 1. User login to his/her home page and navigate. 2. Click on home page link. | The user should be redirected to home page. | OK | Pass |
| 10. | Check for empty inputs for enrollment. | 1. User login to his/her home page and click on enroll link. 2. Click on enroll button without inputting all necessary data. | The enrollment page should display error; check missing inputs. | OK | Pass |
| 11. | Multiple entries of data for enrollment. | 1. User login to his/her home page and click on enroll link. 2. Enter existing enrollment data. 3. Click on enroll button. | The enrollment page should display error; enrollment error, already exist. | OK | Pass |
| 12. | Student checking course assignments submission rate. | 1. User login to his/her home page. 2. Click on course to check. | The user should be redirected to page that displays necessary data such as course name, available assignment, submitted assignment and submission rate. | OK | Pass |
| **No** | **Test Case Description** | **Steps** | **Expected Result** | **Actual Result** | **(Pass/Fail)** |
| 13. | Check for empty inputs for adding course. | 1. User login to his/her home page and click on add new course link. 2. Click on add course button without inputting all necessary data. | The add course page should display error; check missing inputs. | OK | Pass |
| 14. | Multiple entries of data for adding course. | 1. User login to his/her home page and click on add new course link. 2. Enter existing course data and click on add course button. | The add course page should display error; error, already exist. | OK | Pass |
| 15. | Checks for empty inputs for adding assignment. | 1. User login to his/her home page and click on add new assignment link. 2. Click on add assignment button without inputting all necessary data. | The add assignment page should display error; check missing inputs. | OK | Pass |
| 16. | Multiple entries of data for adding assignment. | 1. User login to his/her home page and click on add new assignment link. 2. Enter existing assignment data and click on add assignment button. | The add assignment page should display error; error, already exist. | OK | Pass |
| **No** | **Test Case Description** | **Steps** | **Expected Result** | **Actual Result** | **(Pass/Fail)** |
| 17. | Checks for empty inputs for marking assignment. | 1. User login to his/her home page and click on course name. 2. Click on mark assignment button without inputting all necessary data. | The mark assignment page should display error; check missing inputs. | OK | Pass |

**8 CONCLUSION**

The development of this application was successful with achievement of all the compulsory and necessary technical requirements which enable easy tracking of course assignments on mobile phone by the users of this application.

This project improved my technical and programming skills through working with Android studio and other mobile application development technologies such PHP, XML, MySQL and JSON.

**8.1 Challenges**

The challenges in development of this project include creating responsive user interface with tools such as list views and their adapter and spinner and their adapter. Also getting familiar with android studio development tool was challenging at the beginning such as software debugging, creating and running of application on virtual device.

Requirement of skills and knowledge from other programming technologies which I was not familiar with in development of this project also contributed to the challenges faced and these challenges are part of milestones in success of the project.

**8.2 Improvements**

There are possible improvements that can be included in this project in the future by increase functionalities of the project in order to improve users’ satisfaction. Such improvement includes providing functionality that enable user to reset his/her password.

**REFERENCES**

/1/ Android (Operating System). Accessed 11.10.2016 <https://en.wikipedia.org/wiki/Android_(operating_system)>

/2/ Trish Cornez and Richard Cornez. (2017) *ANDROID Programming Concepts*. Cathy L. Esparti

/3/ Developer Workflow Basics. Accessed 11.10.2016 <https://developer.android.com/studio/workflow.html>

/4/ Publish Your App. Accessed 13.10.2016 <https://developer.android.com/studio/publish/index.html?hl=id>

/5/ PHP. Accessed 13.10.2016

<https://en.wikipedia.org/wiki/PHP>

/6/ PHP 5 Introduction. Accessed 14.10.2016 <http://www.w3schools.com/php/php_intro.asp>

/7/ JSON Wikipedia. Accessed 14.10.2016 <https://en.wikipedia.org/wiki/JSON>

/8/ JSON Introduction. Accessed 14.10.2016 <http://www.w3schools.com/js/js_json_intro.asp>

/9/ Database Wikipedia. Accessed 15.10.2016 <https://en.wikipedia.org/wiki/Database>

/10/ MySQL Wikipedia. Accessed 15.10.2016 <https://en.wikipedia.org/wiki/MySQL>