



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Feranmi Timothy Akanni

# ASSIGNMENT TRACKING ANDROID APPLICATION

Information Technology  
2016

## **ACKNOWLEDGEMENT**

I would want to express my sincere gratitude to all my teachers in VAMK, most especially to my programming teacher and my supervisor for this project, Dr. Ghodrat Moghadampour for his knowledge impact ability, support and guidance during my years of study in VAMK and for his effort in successful completion of this project.

My appreciation goes to my family members for their unending understanding and encouragement during years of my study and my stay in Finland. I will also like to express my gratitude to Finland as a country and VAMK as an educational institution for providing me opportunity to study.

I would like to thank Vaasa Summer University for organizing courses like Android programming course during summer because I used Android programming skills which I learned during summer courses in development of this project. Finally, I thank almighty God for seeing me through years of my study.

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Degree Program in Information Technology

## ABSTRACT

Author	Feranmi Timothy Akanni
Title	Assignment Tracking Android Application
Year	2016
Language	English
Pages	78
Name of Supervisor	Dr. Ghodrat Moghadampour

---

One of the common ways of checking that knowledge is impacted into students at every level of education is by giving various tasks to students and part of the responsibilities of the teacher is to give assignments to students and check the solution provided by the students. Increase in technology development involves a number of mobile applications that are being developed and released on a daily basis, out of which Android operating application is one of the dominant mobile application.

The main idea of this study was to develop an Android operating mobile application that allows the teacher to provide assignments to students and to easily track student involvement in providing complete and required solutions to the assignments. The application can be used anywhere because it's mobile application.

The application allows the teacher to register, to add a course, add an assignment, update and delete an assignment, mark the assignment for the student with the correct assignment solution, check student assignment submission rate and check assignment completion rate. While the application allows the student to register, to enroll for the course, check available and submitted assignments and check assignment submission rate. The targeted group tentatively involved VAMK teachers and students.

The development of this application was carried out on Android Studio IDE using Java programming language and it involved designing and creating working database which is MySQL database for storing application data. Development of this application also involved writing PHP scripts that are located at the server side and these PHP scripts are used for communication between the application and database.

---

Keywords                      Android, PHP, VAMK, MySQL

## CONTENTS

ACKNOWLEDGEMENT.....	II
ABSTRACT.....	III
1 INTRODUCTION.....	12
1.1 Background.....	12
1.2 Motivation.....	12
1.3 Objective.....	12
1.4 Description of the Topic.....	13
2 RELEVANT TECHNOLOGIES.....	14
2.1 Application Structure.....	14
2.2 Application Development Environments.....	15
2.2.1 Hardware.....	15
2.2.2 Java JDK and JRE.....	15
2.2.3 Android SDK.....	16
2.2.4 Android Studio IDE.....	16
2.3 Application Development Process.....	16
2.4 Application Technologies.....	18
2.4.1 PHP.....	18
2.4.2 JSON.....	19
2.4.3 MySQL Database.....	20
2.4.4 Android.....	20
3 APPLICATION DESCRIPTION.....	23
3.1 Quality Function Deployment (QFD).....	23
3.2 Use Case Diagram.....	25
3.2.1 Student Use Case.....	25
3.2.2 Teacher Use Case.....	26
3.3 Sequence Diagram.....	27
3.3.1 Registration Sequence Diagram for Users.....	27

3.3.2	Login Sequence Diagram for Users .....	28
3.3.3	Logout Sequence Diagram for User .....	29
3.3.4	Course Enrollment Sequence Diagram (Student) .....	30
3.3.5	Add Assignment Sequence Diagram (Teacher).....	31
3.3.6	Add Course Sequence Diagram (Teacher) .....	32
3.3.7	Mark Assignment Sequence Diagram (Teacher).....	33
3.3.8	Display Student Submission Rate Sequence Diagram (Teacher) .....	34
3.3.9	Display Student Submission Rate Sequence Diagram (Student).....	35
3.3.10	Display Assignment Completion Rate Sequence Diagram (Teacher)...	36
3.4	Class Diagram.....	37
3.5	Component Diagram.....	38
3.5.1	Model.....	38
3.5.2	View.....	39
3.5.3	Controller.....	39
4	DATABASE.....	40
4.1	Design of Database .....	40
5	GRAPHICAL USER INTERFACE DESIGN .....	43
5.1	Teacher Login Page .....	43
5.1.1	Teacher Home Page .....	44
5.1.2	Teacher Registration Page .....	45
5.1.3	Assignment marking Page .....	45
5.1.4	Course Assignments Page.....	46
5.1.5	Course Students Page.....	47
5.1.5	Student Assignment Page .....	48
5.1.6	Add Course Page.....	49
5.1.7	Add Assignment Page.....	50
5.2	Student Login Page.....	51
5.2.1	Student Home Page.....	52
5.2.2	Student Registration Page .....	53
5.2.3	Course Enrollment Page .....	54

6	IMPLEMENTATION .....	56
6.1	Login.....	56
6.2	Registration.....	58
6.3	Student Home .....	59
6.4	Teacher Home.....	61
6.5	Assignment marking.....	63
6.6	Course Assignments .....	64
6.7	Student Assignment.....	65
6.8	Add Course .....	66
6.9	Add Assignment .....	67
6.10	Course Enrollment.....	68
6.11	Android Manifest.....	68
7	TESTING.....	71
8	SUMMARY.....	76
9	CONCLUSION .....	77
9.1	Challenges .....	77
9.2	Improvements .....	77
	REFERENCES.....	78

## LIST OF FIGURES, TABLES AND CODE SNIPPETS

Figure 1: Application Architecture.....	15
Figure 2: Application Development Process. /3/.....	17
Figure 3: Compilation Process between Non Android and Android Java. /3/ .....	22
Figure 4: Student Use Case Diagram .....	26
Figure 5: Teacher Use Case Diagram.....	27
Figure 6: Registration Sequence Diagram for Users.....	28
Figure 7: Registration Sequence Diagram for Users.....	29
Figure 8: Logout Sequence Diagram for User .....	30
Figure 9: Course Enrollment Sequence Diagram.....	31
Figure 10: Add Assignment Sequence Diagram .....	32
Figure 11: Add Course Sequence Diagram.....	33
Figure 12: Mark Assignment Sequence Diagram .....	34
Figure 13: Assignment Submission Rate Sequence Diagram (Teacher) .....	35
Figure 14: Student Assignment Submission Rate Sequence Diagram.....	35
Figure 15: Assignment Completion Rate Sequence Diagram (Teacher) .....	36
Figure 16: Class Diagram.....	37
Figure 17: Mode of Communication .....	38
Figure 18: Component Diagram.....	39
Figure 19: ER diagram .....	41
Figure 20: DB Tables .....	42
Figure 21: Teacher Login Page .....	43
Figure 22: Teacher Home Page .....	44
Figure 23: Teacher Registration Page .....	45
Figure 24: Assignment Marking Page.....	46
Figure 25: Course Assignments Page.....	47
Figure 26: Course Students Page.....	48

Figure 27: Student Assignment Page .....	49
Figure 28: Add Course Page .....	50
Figure 29: Add Assignment Page.....	51
Figure 30: Student Login Page.....	52
Figure 31: Student Home Page.....	53
Figure 32: Student Registration Page.....	54
Figure 33: Course Enrollment Page .....	55
Table 1: Quality Function Deployment.....	24
Table 2: Application Testing Template.....	71
Code Snippet 1: On create method and on login Method .....	56
Code Snippet 2: Login background task .....	57
Code Snippet 3: Login on post execute.....	57
Code Snippet 4: On register method for register button .....	58
Code Snippet 5: Registration background task .....	58
Code Snippet 6: Registration on post execute method.....	59
Code Snippet 7: Method that redirect to student enrollment page .....	59
Code Snippet 8: On click listener for each course on student page .....	60
Code Snippet 9: Method that redirect to add course page.....	61
Code Snippet 10: Method that redirect to add assignment page .....	62
Code Snippet 11: Setting list adapter for each row in teacher page.....	62
Code Snippet 12: Method for marking assignment.....	63
Code Snippet 13: Spinner for assignment selection.....	64
Code Snippet 14: List adapter for course assignment .....	64
Code Snippet 15: List adapter for available assignment .....	65
Code Snippet 16: List adapter for submitted assignment.....	66



Code Snippet 17: Method for add course button.....	67
Code Snippet 18: Method for add assignment button .....	67
Code Snippet 19: Method for enroll button.....	68
Code Snippet 20: Manifest with internet permission and launcher.....	69
Code Snippet 21: Manifest with activities names and intents.....	70

## LIST OF ABBREVIATION

JSON	JavaScript Object Notation
OS	Operating System
PHP	Personal home page Hypertext Preprocessor
SQL	Structured Query Language
GUI	Graphical User Interface
XML	Extensible Markup Language
IDE	Integrated Development Environment
SDK	Software Development Kit
ADT	Android Development Tools
AVD	Android Virtual Device
JRE	Java Runtime Environment
API	Application Programming Interface
VAMK	Vaasan Ammattikorkeakoulu
DBMS	Database Management System
PC	Personal Computer
GNU	GNU's Not Unix
PERL	Practical Extraction Reporting Language
VM	Virtual Machine
OOP	Object Oriented Programming
JSP	Java Server Programming

JDK	Java Development Kit
HTTP	Hypertext Transfer Protocol
ER	Entity Relationship
DB	Database
OFD	Quality Function Development
APP	Application Program

# **1 INTRODUCTION**

The main objective of this study is to develop an assignment tracking mobile application that allows a teacher to provide assignments to students and to easily track student involvement in providing complete and required solutions to the assignments. This section describes the background, motivation and objective of this study.

## **1.1 Background**

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 the students' life. Assignments can be defined as tasks allocated to students and related to the course of study.

## **1.2 Motivation**

The development of this project was motivated during the process of checking and marking of programming assignments by my programming teacher in my fourth year of study in VAMK. Programming assignments have to be checked manually by teacher in order to ensure that the assignments meet the requirement before the teacher will mark it completed for each student and this is done on paper list that contains list of students with their student number and names. The numbers of assignments completed by each student is written against his/her student number and during one of the classes, an idea came that it could be good to have this tracking on mobile phone. I started with a web development and moved on to building the mobile development.

## **1.3 Objective**

The objective of this project involved an 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 an opportunity for college teachers to perform various activities such as allowing teacher registration, enables the teacher to add a new course and add a new assignment, provides an easy way for a teacher to mark assignments for the student after the teacher has checked that the assignments have been satisfactorily completed by the student and most importantly provide teachers with assignment submission rates and completion rates which provide the information about students progress on all the course assignments.

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

#### **1.4 Description of the Topic**

This project focused on development of the 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 a 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.

## 2 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

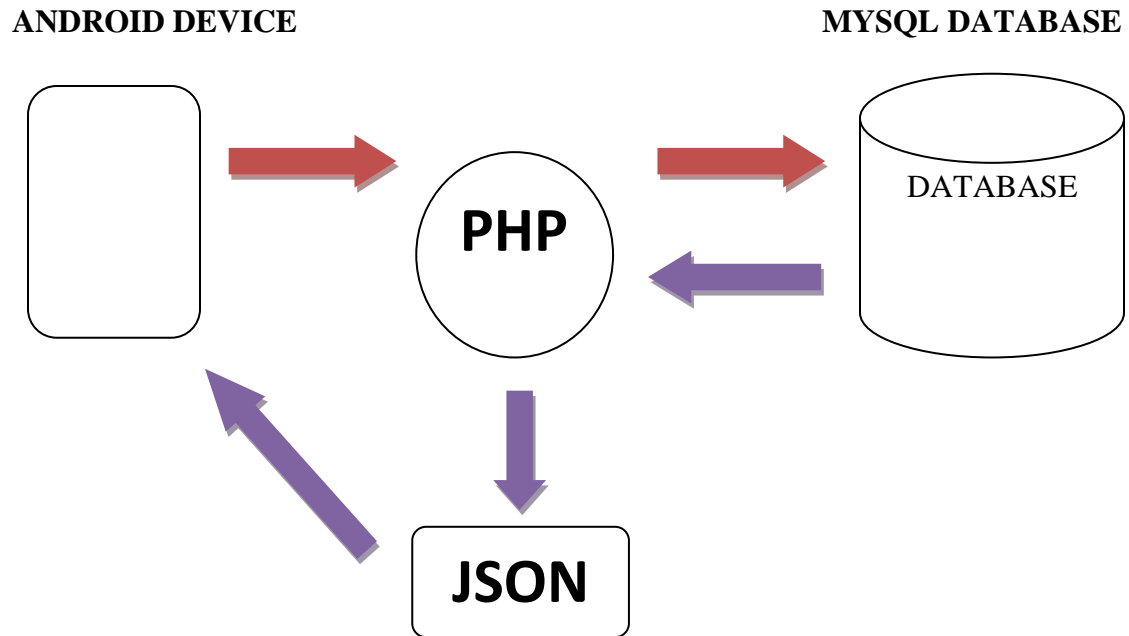
This application is divided into two main sections which are the teacher section which is the application used by the teachers and the student section which is the application used by the students. Each of these sections of the application can be divided into two main different parts which are client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

The Client side represents Android devices which provide user interfaces for manipulation and visualization of data and the server side represents MySQL database which is meant for storing of data.

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

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

- ❖ The client device makes a HTTP POST/GET request to server
- ❖ The PHP scripts make connection and send queries to 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.



**Figure 1: Application Architecture**

## 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

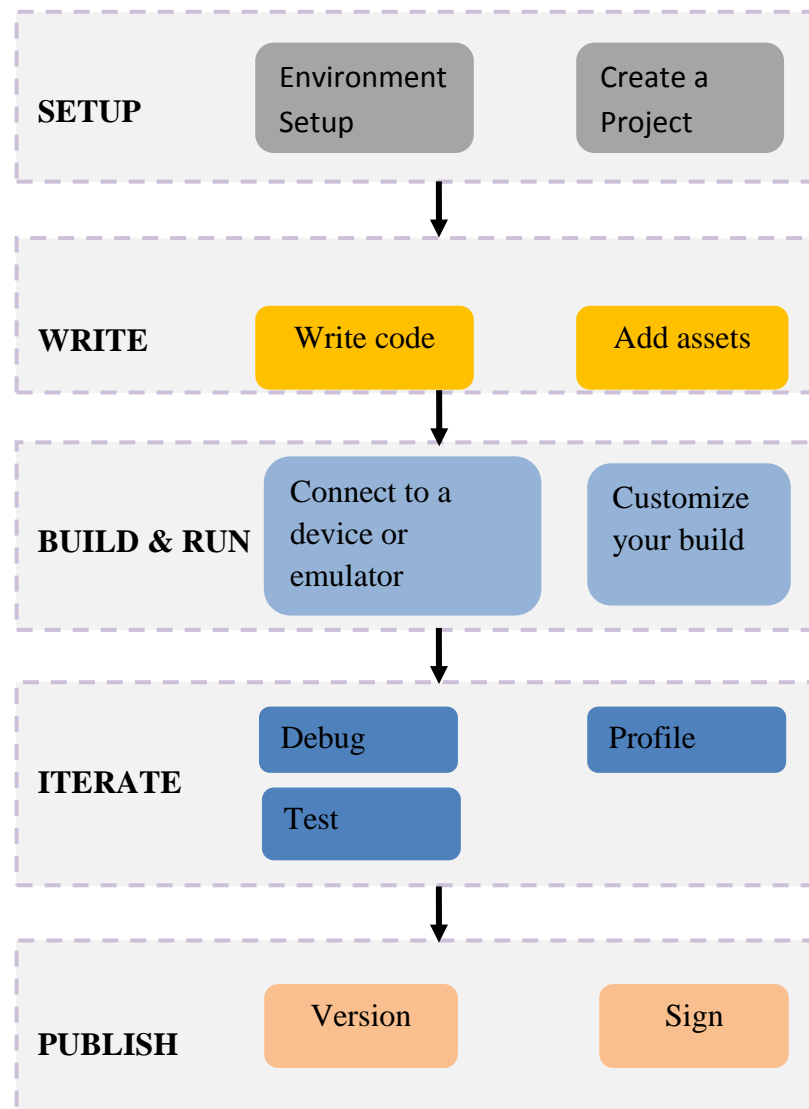
### **2.2.4 Android Studio IDE**

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**

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





**Figure 2: Application Development Process. /3/**

- ❖ **Setup Phase:** This is the phase in which the development environment is set up after downloading and installing Android Studio is already accomplished. Creation of the 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 the 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. The 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.

#### 2.4.2 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"}
    ]
}
```

### 2.4.3 MySQL Database

A 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. The DBMS functions can be mainly classified into four types, 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/

### 2.4.4 Android

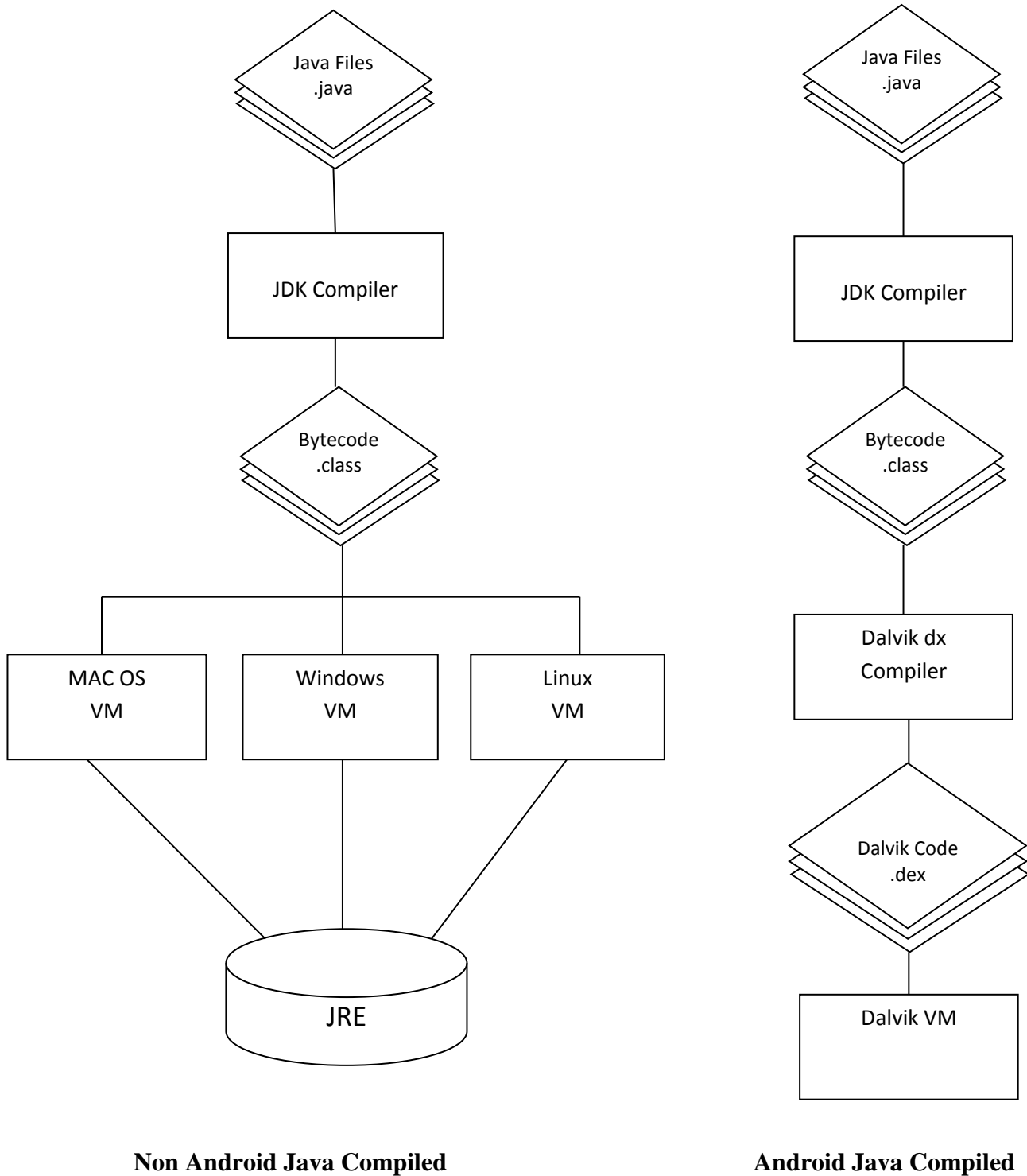
The evolution of Android has an important impact on the area of technology in today. Android is an operating system that is based on Linux kernel and it is developed by Google. There are different types of Androids which are basically named depending on different devices on which the operating system 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 whose platform independence makes 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 on 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/



**Figure 3: Compilation Process between Non Android and Android Java. /3/**

### **3 APPLICATION DESCRIPTION**

This section provides a detailed description of the application and the expected requirements of 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.

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

The student application section of this project allows the users which are students to register and login to their home page. The 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 a 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 the requirements that are valuable to

this project and these requirements are classified in accordance to the project overall goals and objectives. QFD 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.

**Table 1: Quality Function Deployment.**

<b>Normal Requirement with priority level 1</b>
<ul style="list-style-type: none"> <li>▪ The application must be able to display login page for students.</li> <li>▪ It must be able to display login page for teachers.</li> <li>▪ The user must be able to register by providing username, password and other information.</li> <li>▪ The user must be able to login with the username and password provided during registration.</li> <li>▪ The user must be able to logout from his/her home page.</li> <li>▪ The student must be able to enroll for available courses</li> <li>▪ The student must be able to see courses he/she has enrolled for.</li> <li>▪ The student must be able to see available assignments on courses he/she has enrolled for.</li> <li>▪ The student must be able to see his/her submitted assignments on courses he/she has enrolled for.</li> <li>▪ The teacher must be able to add new course.</li> <li>▪ The teacher must be able to add assignment for each course.</li> <li>▪ The teacher must be able to mark assignment for student.</li> <li>▪ The teacher must be able to see all courses he/she has added</li> <li>▪ The teacher must be able to see all assignments for his/her course</li> <li>▪ The teacher must be able to see completion rate for each course assignment.</li> <li>▪ The teacher must be able to see all students that have enrolled for his course.</li> <li>▪ The teacher must be able to see submission rate of enrolled students for each course.</li> </ul>



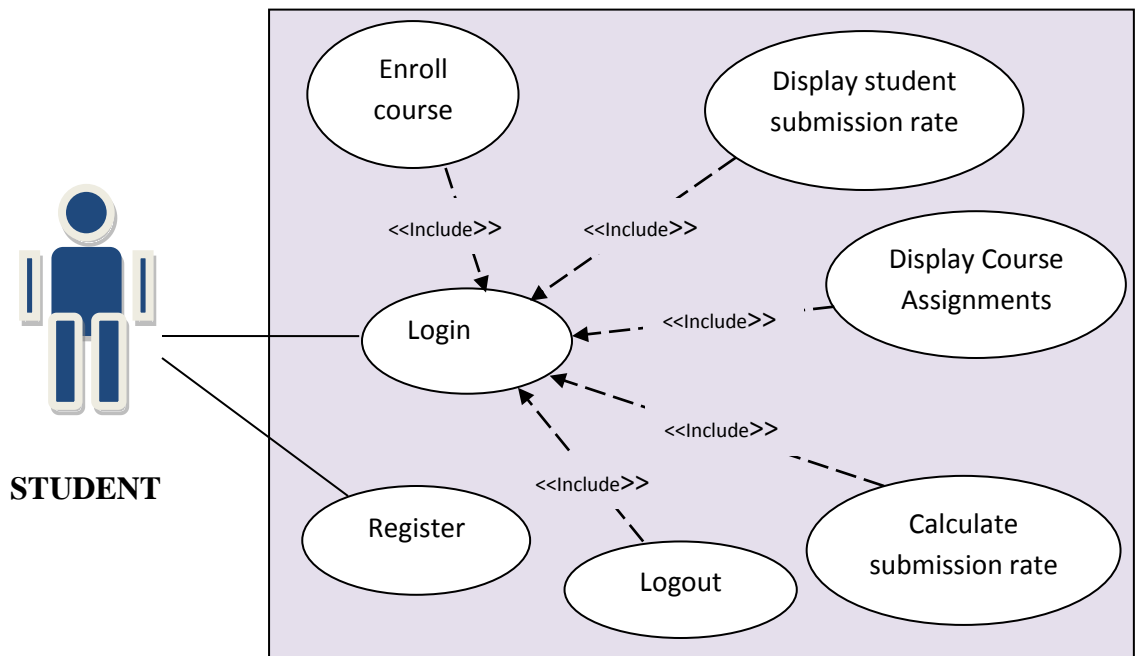
<b>Expected Requirement with priority level 2</b>
<ul style="list-style-type: none"> <li>▪ The user should be able to see registration response if registration fails.</li> <li>▪ The user should be able to see login response if login fails.</li> <li>▪ The teacher should be able to see assignment marking feedback message.</li> <li>▪ The teacher should be able to unmark assignment for student.</li> <li>▪ The application should be interactive, easy to use and user friendly.</li> <li>▪ 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).</li> </ul>
<b>Exciting Requirement with priority level 2</b>
<ul style="list-style-type: none"> <li>▪ The application can allow users to reset their password.</li> <li>▪ The application can provide student with teacher information such as teacher email address and phone number.</li> <li>▪ The application can provide information about the educational institution using the application such as school website and school address.</li> </ul>

### **3.2 Use Case Diagram**

The 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**

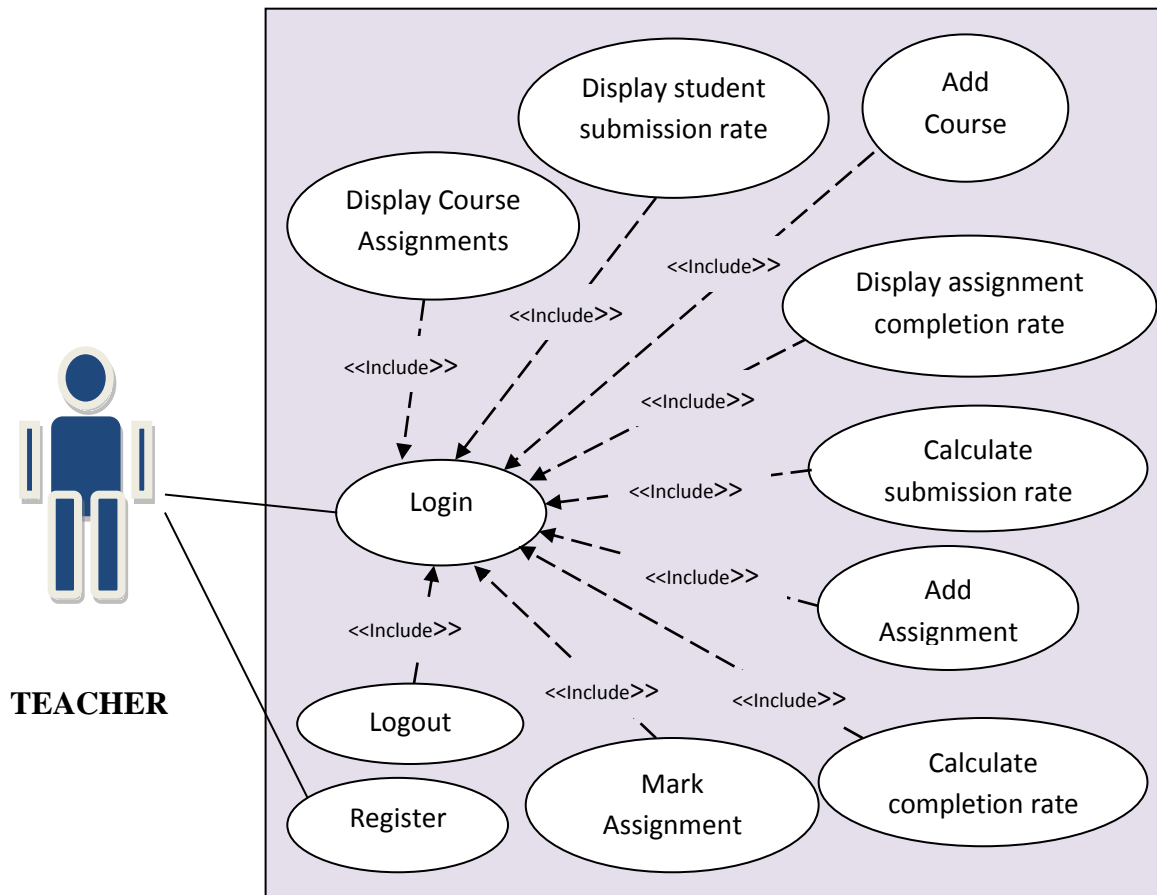
The student use case provides the functionalities that involve student interaction with the system. The student needs to register before login to the system and after login; the student can view available courses in the system, enroll for the course and log out with logout link. The student can view available assignments for the course he/she has enrolled for and can also view the submission rate. The figure below represents the application student use case diagram.



**Figure 4: 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 teacher needs to have an account in the system by register before he/she can login and use the application. The teacher can add course, add assignments for each course and can also mark assignments for students that have enrolled for the course. The teacher can view all course assignments, view the completion rate for each assignment, and also view each student assignment submission rate for each course. The teacher can log out his/her account with logout link. The figure below represents the application teacher use case diagram.



**Figure 5: Teacher Use Case Diagram**

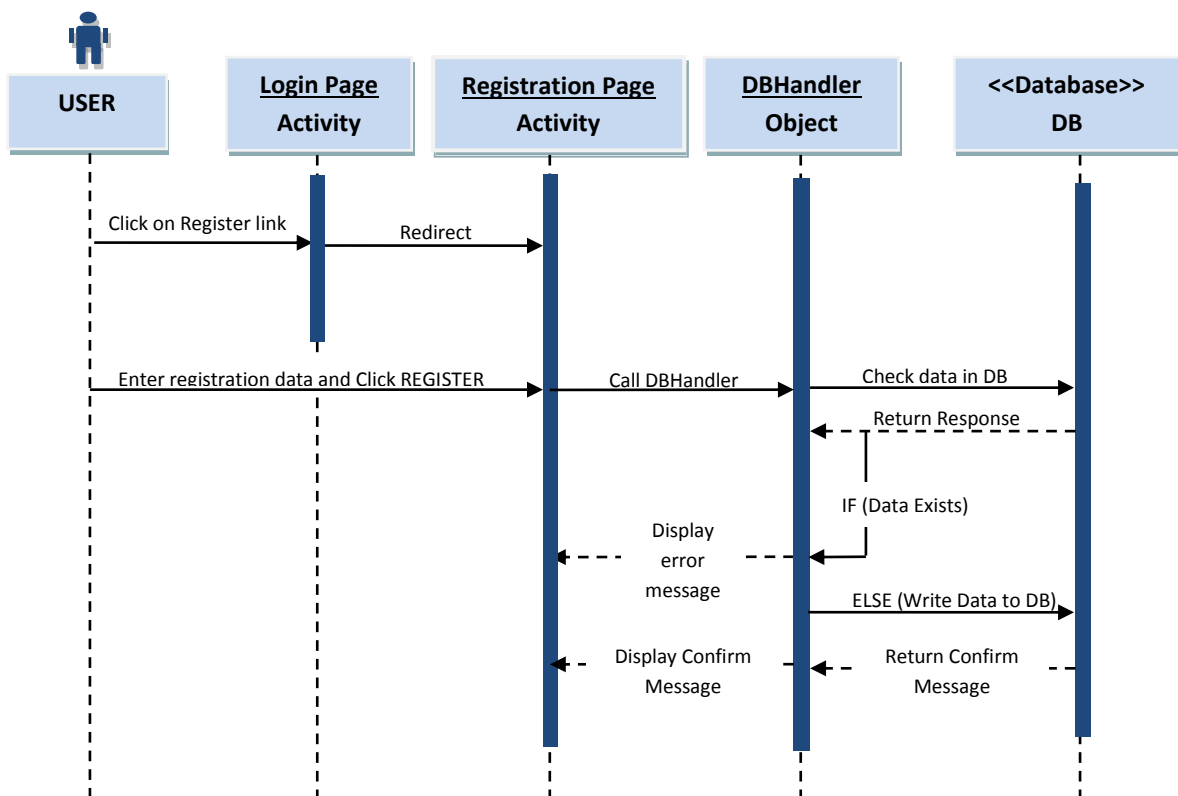
### 3.3 Sequence Diagram

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

This describes the sequence of steps required for the users to register before using the application. The user will click on the registration link at the login page of the application and will be redirected to the registration page. The user would provide the required

registration data and click register button. The DB Handler will check if the provided data already exist in the database and display error message if already existed. If the provided data does not exist in the database, the DB Handler will write the provided data into application database and display successful registration message to the user. The figure below represents the user's registration sequence diagram.

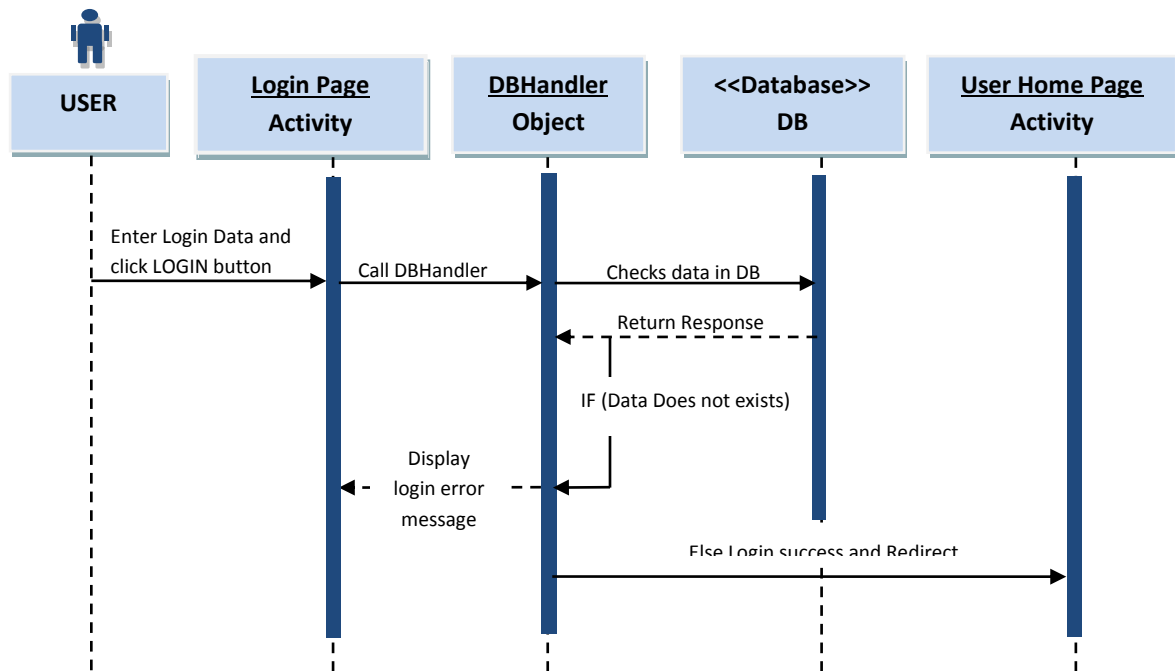


**Figure 6: Registration Sequence Diagram for Users**

### 3.3.2 Login Sequence Diagram for Users

This describes the sequence of steps required for the application users to login into the application. Before the user can login to the application, the users must have an account in the database through registration process. On the login page, the user would provide login data which are username and password and click on login button. Then the DB Handler would check for existence of provided login details in the database and if login details existed, the user would be redirected to his/her home page. If the login details provided are

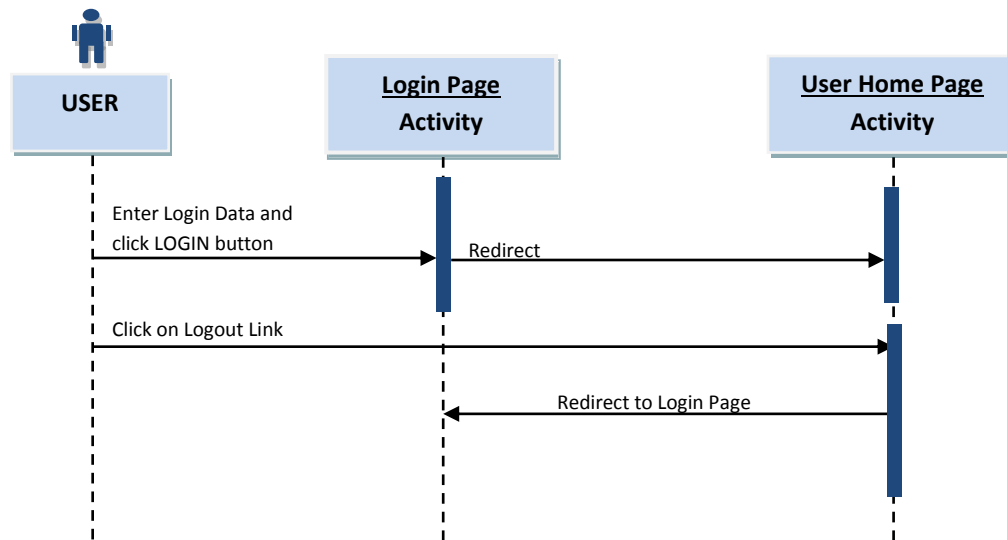
not found in the database, login error will be displayed on login page. The figure below represents user's login sequence diagram.



**Figure 7: Registration Sequence Diagram for Users**

### 3.3.3 Logout Sequence Diagram for User

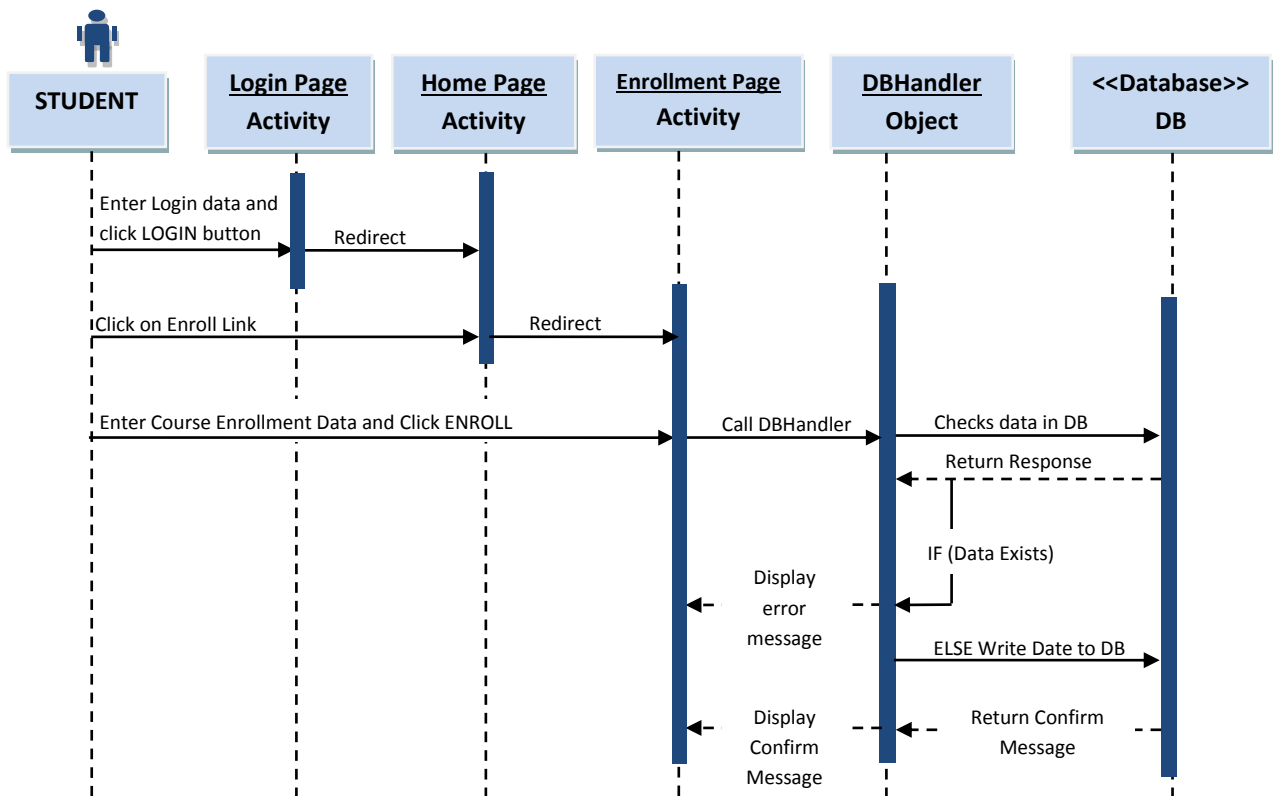
This describes the sequential steps required for user to logout of the application. On the login page, the user would need to log in into the application by providing login details and after successful login, the user would be redirected to his/her home page. The user can log out of the application by clicking on logout link and on pages of the application asides from login and registration page and the user would be redirected to login page. The figure below represents user logout sequence diagram.



**Figure 8: Logout Sequence Diagram for User**

### 3.3.4 Course Enrollment Sequence Diagram (Student)

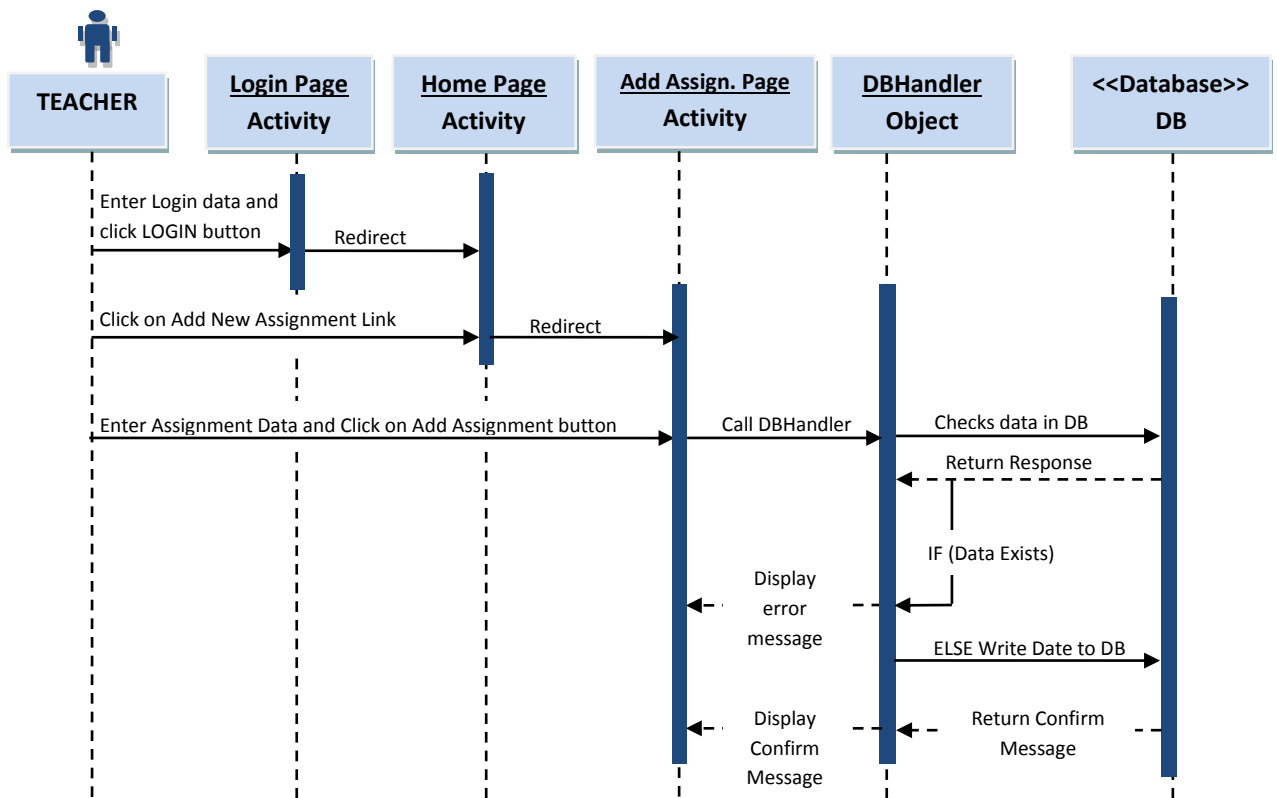
This describes the sequence of steps required for the student to enroll to a course. The student needs to provide valid login details on student login page and click on login button and after which the student will be redirected to his/her home page. From the student home page, the student will click on enroll link in order to enroll for course and then he/she will be redirected to course enrollment page that shows drop down list of available courses in the system. The student will select course and provide required data for enrollment such as course password and click on enroll button. Then the DB Handler will check if the provided data already exist in the database and display error message if already existed. If the provided data does not exist in database, the DB Handler will write the provided data into application database and display confirmation message on enrollment page. The figure below represents student course enrollment sequence diagram.



**Figure 9: Course Enrollment Sequence Diagram**

### 3.3.5 Add Assignment Sequence Diagram (Teacher)

This describes the sequence of steps required for the teacher to add an assignment. The teacher needs to provide valid login details on teacher login page and click on login button and after which the teacher will be redirected to his/her home page. From the teacher home page, the teacher would click on add assignment link in order to add assignment and then be redirected to add assignment page. The teacher will provide required data for the assignment and click on add assignment button. Then the DB Handler will check if the provided data already exist in the database and display error message if already existed. If the provided data does not exist in database, the DB Handler will write the provided data into application database and display confirmation message on add assignment page. The figure below represents add assignment sequence diagram.

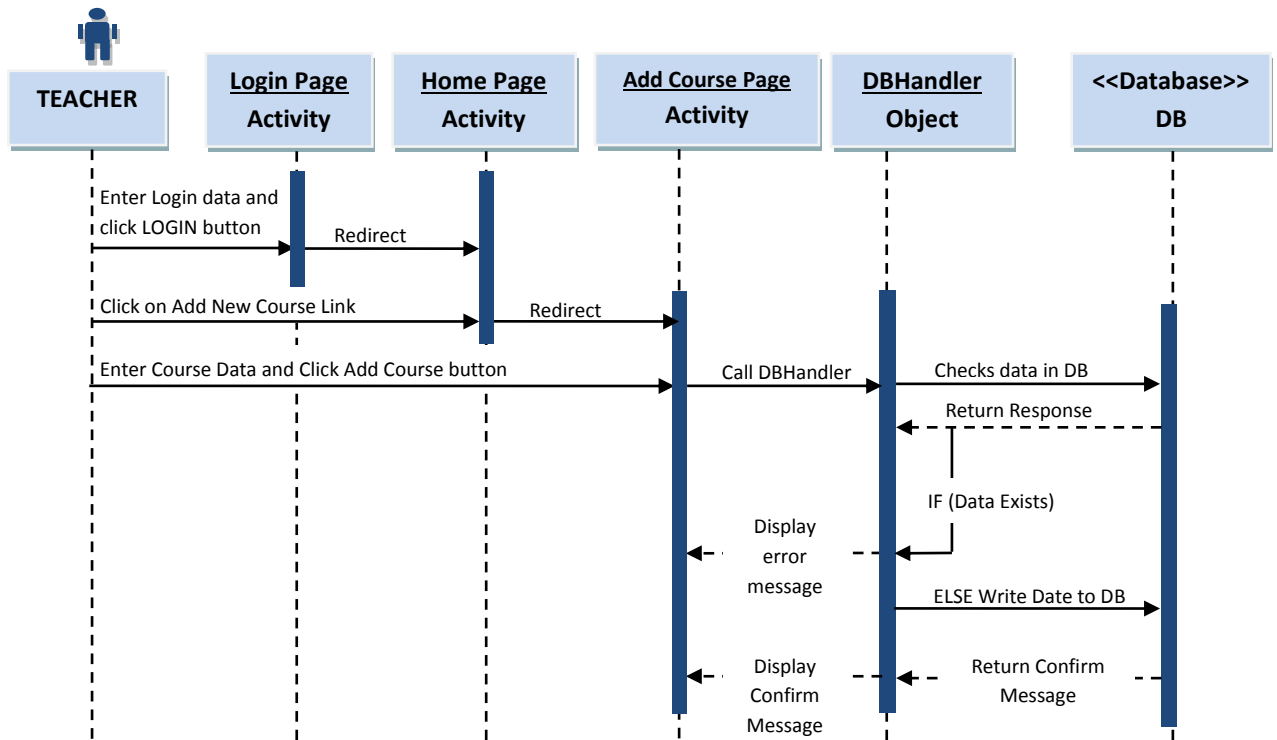


**Figure 10: Add Assignment Sequence Diagram**

### 3.3.6 Add Course Sequence Diagram (Teacher)

This describes the sequence of steps required for the teacher to add a course. The teacher needs to provide valid login details on teacher login page and click on login button and after which the teacher will be redirected to his/her home page. From the teacher home page, the teacher will click on add course link in order to add course and then be redirected to add course page. The teacher will provide required data for the course and click on add course button. Then the DB Handler will check if the provided data already exist in the database and display error message if already existed. If the provided data does not exist in database, the DB Handler will write the provided data into application database and display confirmation message on add course page. The figure below represents add course sequence diagram.

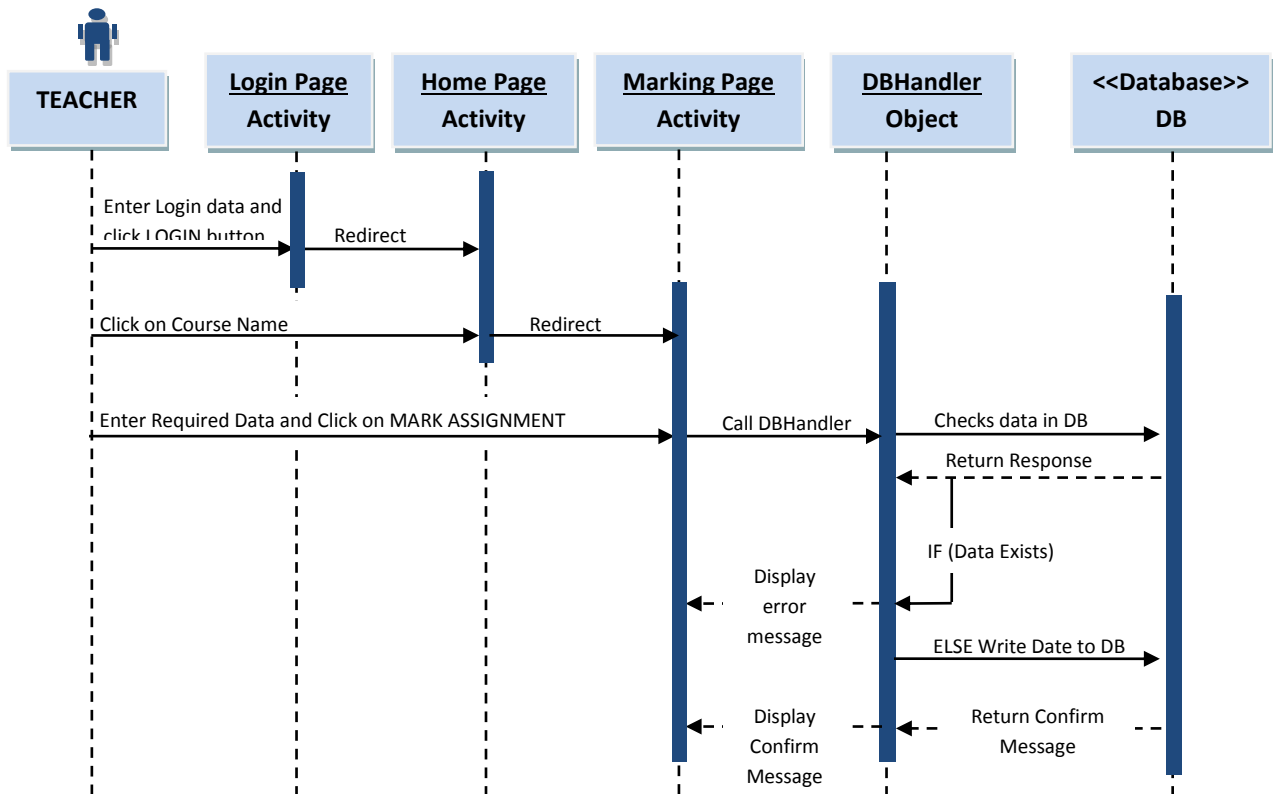




**Figure 11: Add Course Sequence Diagram**

### 3.3.7 Mark Assignment Sequence Diagram (Teacher)

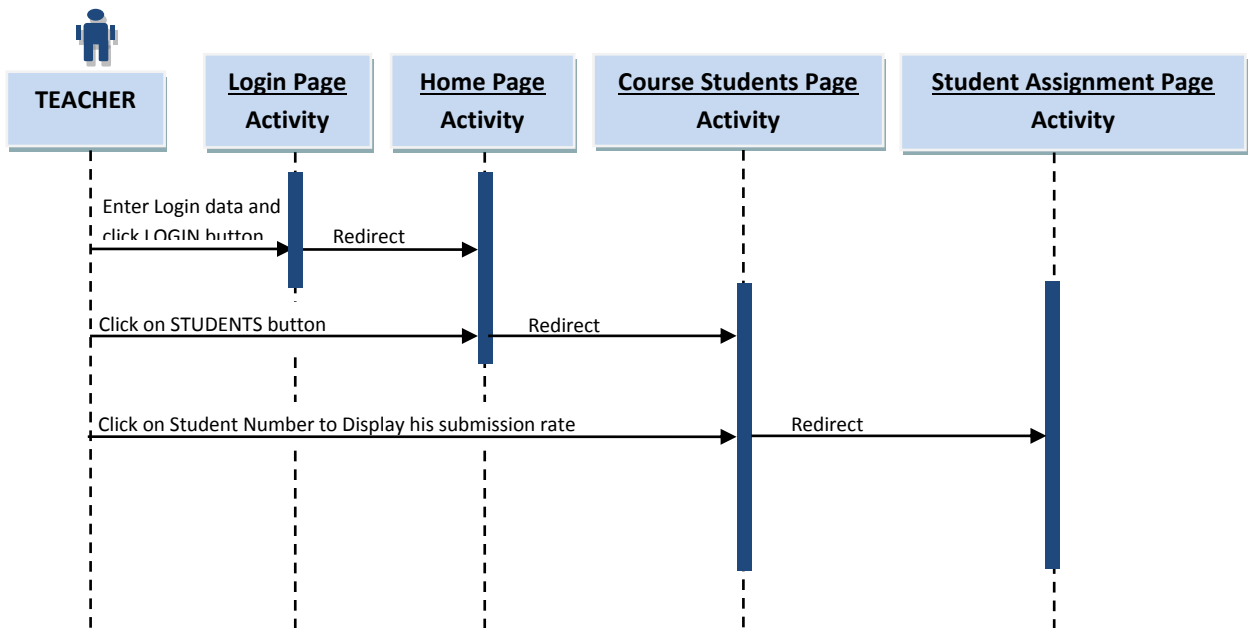
This describes the sequence of steps required for the teacher to mark an assignment for student. The teacher needs to provide valid login details on teacher login page and click on login button and after which the teacher will be redirected to his/her home page. From the teacher home page, the teacher will click on course link in order to mark course assignment and then be redirected to the marking page for that particular course. The teacher will provide required data and click on mark assignment button. Then the DB Handler would check if the provided data already exist in the database and display error message if already existed. If the provided data does not exist in database, the DB Handler will write the provided data into application database and display confirmation message on marking page. The figure below represents mark assignment sequence diagram.



**Figure 12: Mark Assignment Sequence Diagram**

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

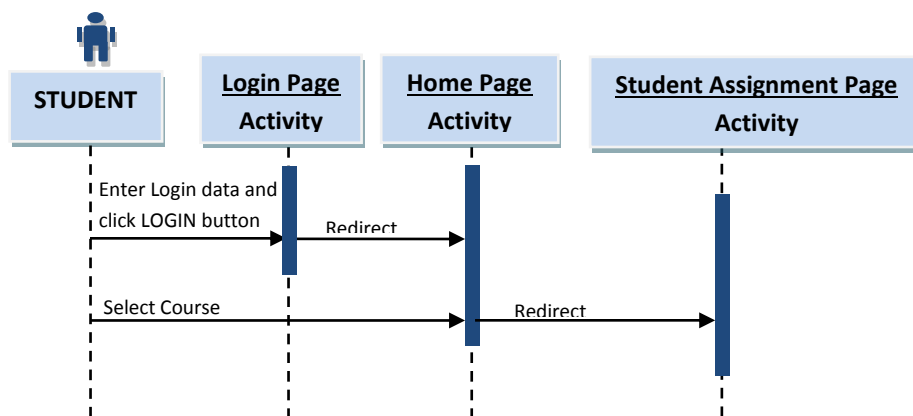
This describes the sequence of steps required for the teacher to check assignment submission rate for student. The teacher needs to provide valid login details on teacher login page and click on login button and after which the teacher would be redirected to his/her home page. From the teacher home page, the teacher will click on students' button for particular course and then be redirected to the course's student page which displays student number, first name and last name of all students that have enrolled for the course. The teacher can click on student number for each student and then be redirected to the student assignment page where the teacher can view the available assignment, submitted assignment and the assignment submission rate of the student. The figure below represents assignment submission rate sequence diagram.



**Figure 13: Assignment Submission Rate Sequence Diagram (Teacher)**

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

This describes the sequence of steps required for the student to check his/her assignment submission rate. The student needs to provide valid login details on student login page and click on login button and after which the student will be redirected to his/her home page. From the home page, the student will select course and will be redirected to student assignment page for the selected course where student can view his/her submitted assignment and the assignment submission rate. The figure below represents assignment submission rate sequence diagram.



**Figure 14: Student Assignment Submission Rate Sequence Diagram**

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

This describes the sequence of steps required for the teacher to check assignment completion rate. The teacher needs to provide valid login details on the teacher login page and click on login button and after which the teacher will be redirected to his/her home page. From the teacher home page, the teacher will click on assignments button for particular course and then be redirected to course assignments page which displays all assignments for the course. The figure below represents assignment completion rate sequence diagram.

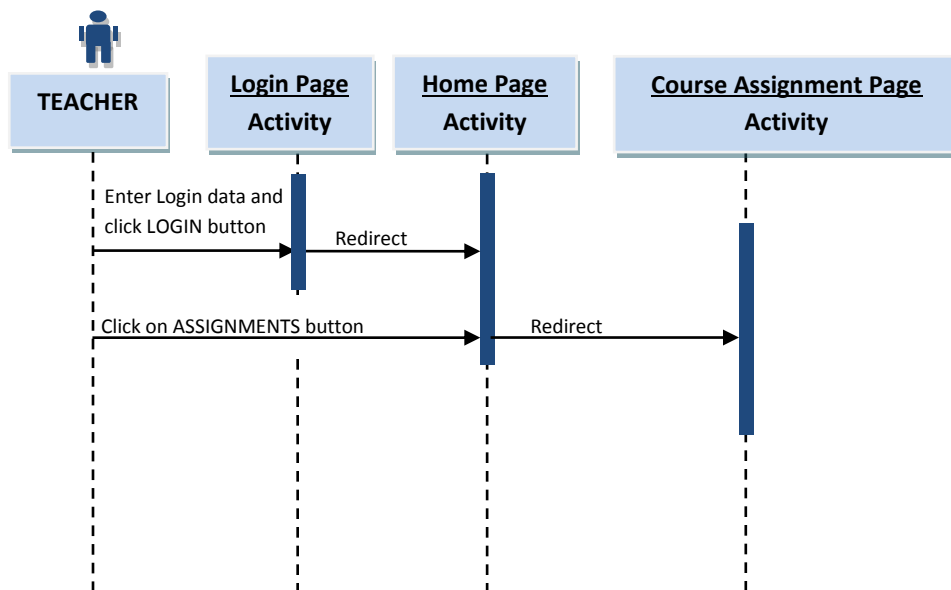


Figure 15: Assignment Completion Rate Sequence Diagram (Teacher)

### 3.4 Class Diagram

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

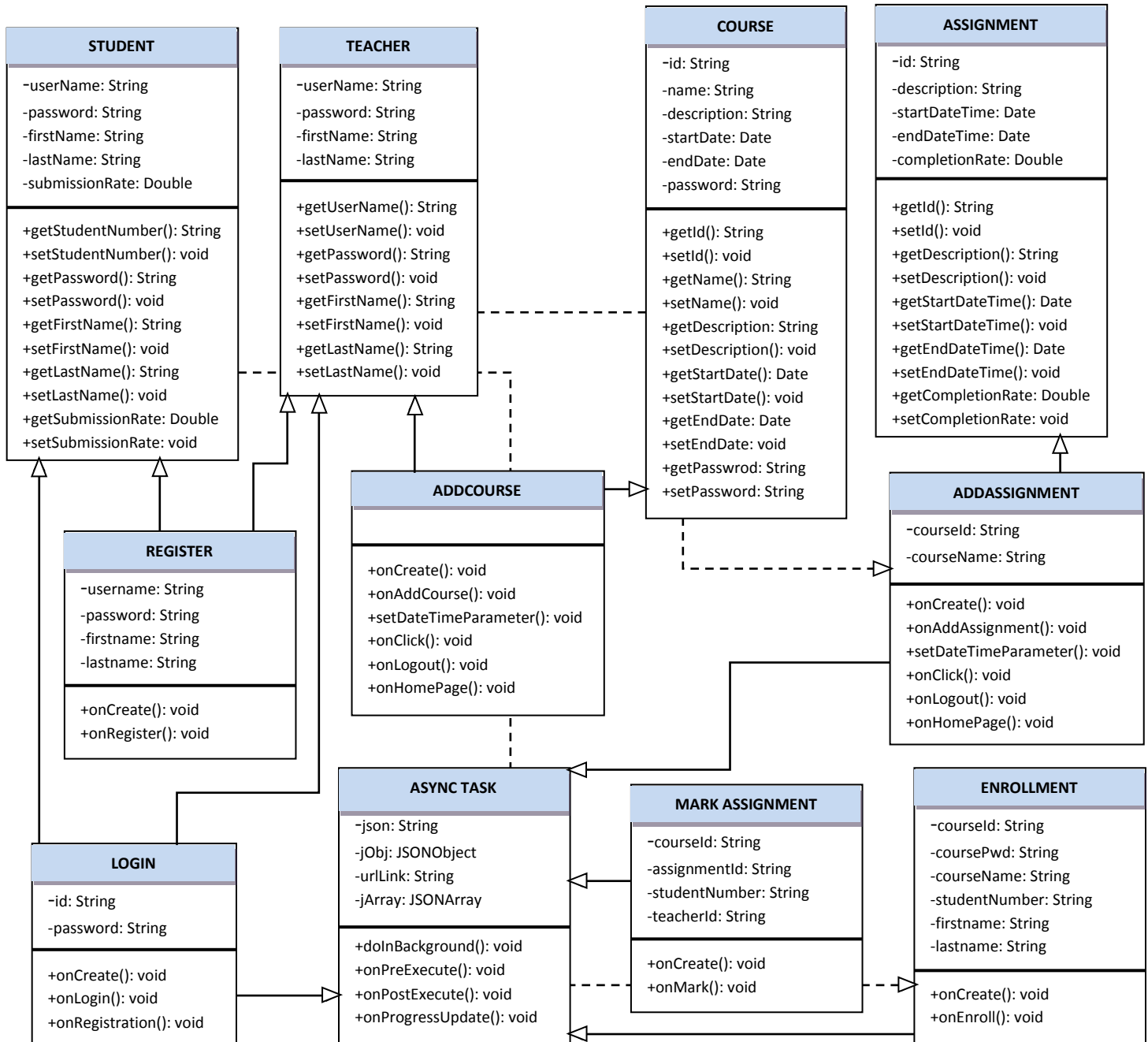
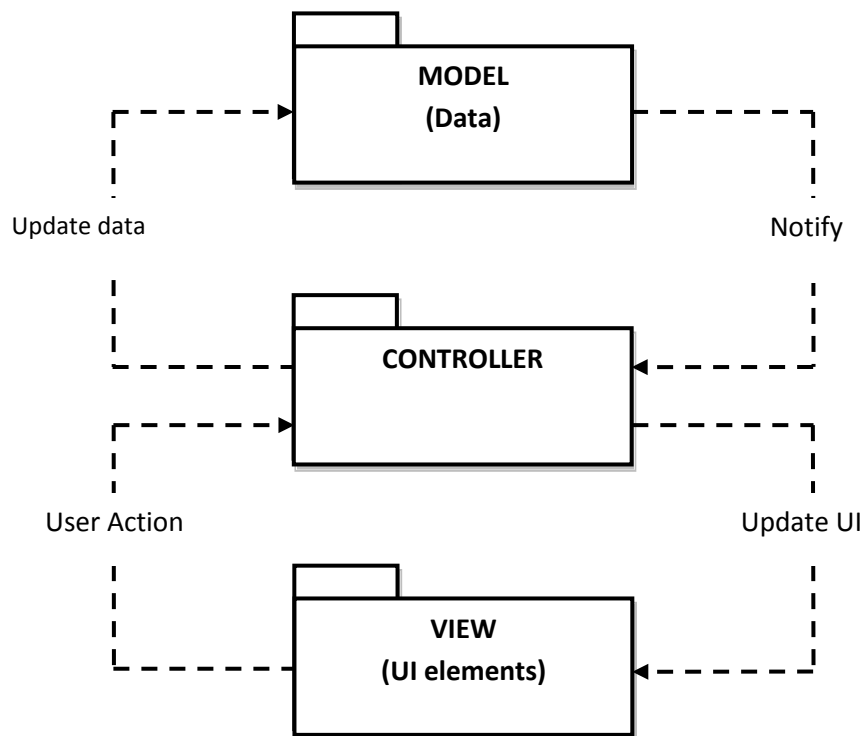


Figure 16: Class Diagram

### 3.5 Component Diagram

The component diagram describes how an application system is divided into sub-systems; shows 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/



**Figure 17: Mode of Communication**

#### 3.5.1 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 as student, teacher, course and assignment objects.

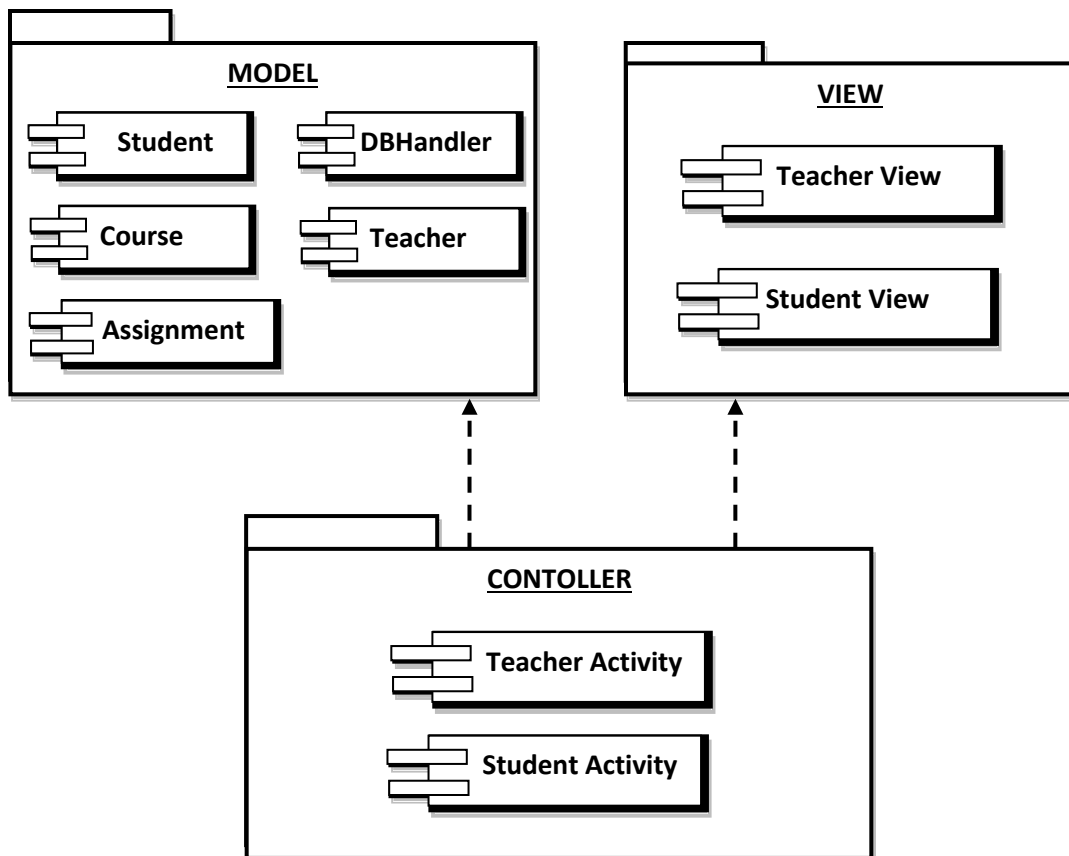
### 3.5.2 View

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.

### 3.5.3 Controller

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

The figure below shows this application component diagram;



**Figure 18: Component Diagram**

## 4 DATABASE

This application is using a 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.

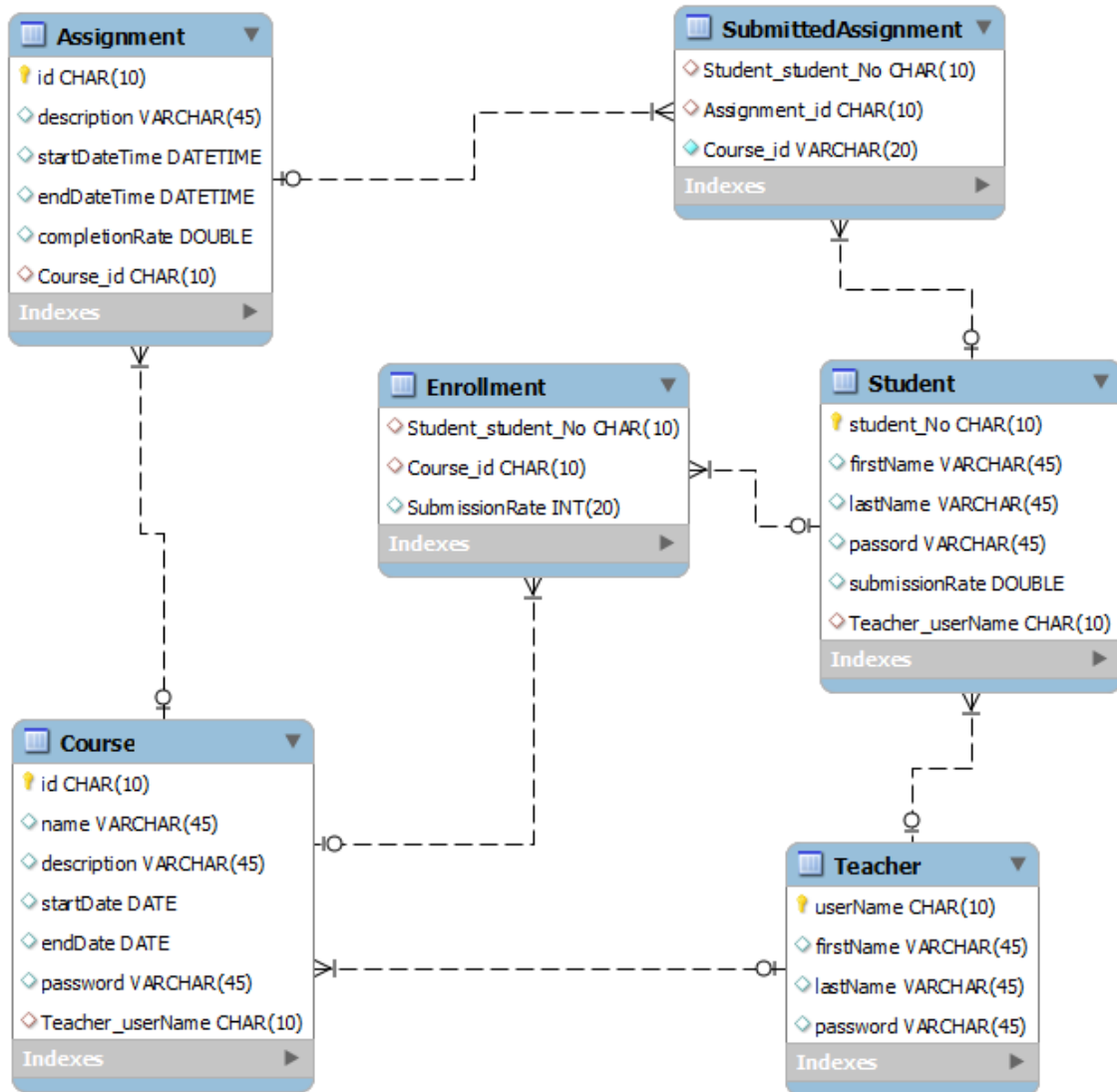
### 4.1 Design of Database

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 situations in which the type of relationships between tables leads to creation of another table, such as the relationship between a student table and a course table in this application leads to creation of an enrollment table. This application database consists of six tables which are 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 the 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.





**Figure 19: ER diagram**

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 a 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 leads to

the creation of bridge table which is an 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, the 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.

Table	Action	Rows	Type	Collation	Size	Overhead
Assignment	Browse Structure Search Insert Empty Drop	18	InnoDB	latin1_swedish_ci	48 KiB	-
Course	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	32 KiB	-
Enrollment	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	96 KiB	-
Student	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	32 KiB	-
SubmittedAssignment	Browse Structure Search Insert Empty Drop	14	InnoDB	latin1_swedish_ci	64 KiB	-
Teacher	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
6 tables	Sum	56	MyISAM	latin1_swedish_ci	288 KiB	0 B

**Figure 20: DB Tables**

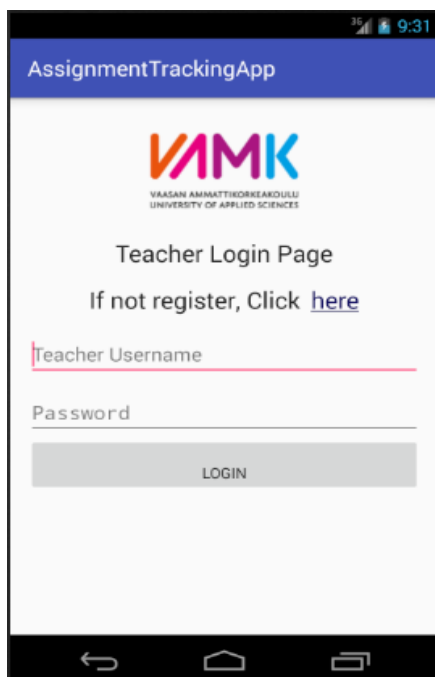
The server side of this application is implemented with PHP scripts and these PHP files are stored inside directory called AssignmentTrackingApp in the public\_html directory (<http://www.cc.puv.fi/~e1100617/AssignmentTrackingApp/>). Making connection to database with PHP scripts requires the database 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 the teacher home page by providing their registered teacher id, password and click on the login button. This page provides clickable link which enables the teacher that has not registered to register, and by clicking on the link, the application provides the 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 the message to the user in case of login error. The figure below represents the teacher login page.



**Figure 21: Teacher Login Page**

### 5.1.1 Teacher Home Page

After the teacher has successfully logged in 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 course names presented in a clickable text, assignments for each course presented on a clickable button and students for each course presented on a clickable button. There is a clickable text named add new course which allows the teacher to add a new course, a clickable text named add new assignment which allows the teacher to add a new assignment to the course and a clickable logout button that enables the teacher to logout. The figure below represents the teacher home page.

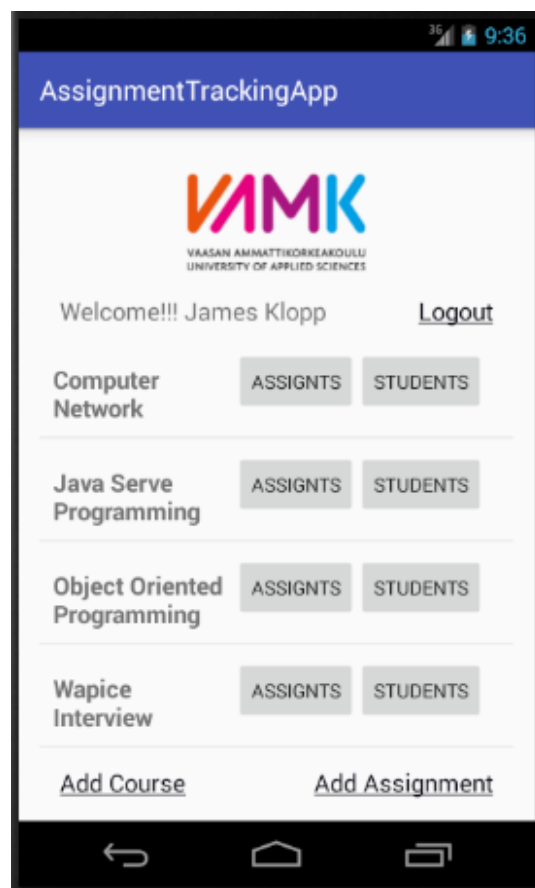


Figure 22: Teacher Home Page

### 5.1.2 Teacher Registration Page

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

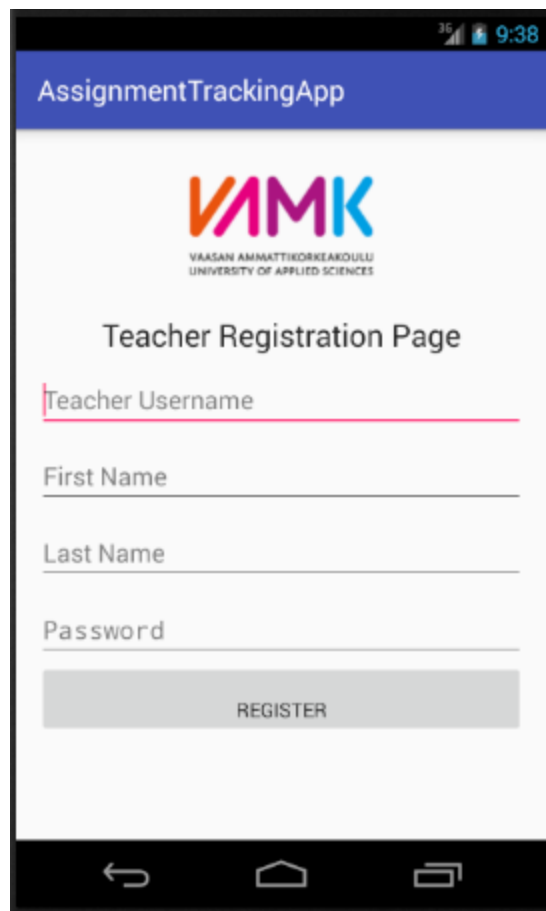


Figure 23: Teacher Registration Page

### 5.1.3 Assignment marking Page

This page allows the teacher to mark assignments for student. All courses in the teacher home page has clickable names which provides assignment marking page when the

teacher clicks on the course name in order to mark an assignment for students. This page provides the student id of all students that have enrolled for the particular clicked course and allows the 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 in case of marking error. The figure below represents the assignment marking page.



**Figure 24: Assignment Marking Page**

#### **5.1.4 Course Assignments Page**

This page appears to the teacher when the teacher clicks on the assignment button on his/her home page and it provides the teacher information about the course name, list of

assignments for each course, the due date time for each assignment and the completion rate for each assignment. There is a clickable logout text which allows the teacher to logout. The figure below represents the course assignments page.

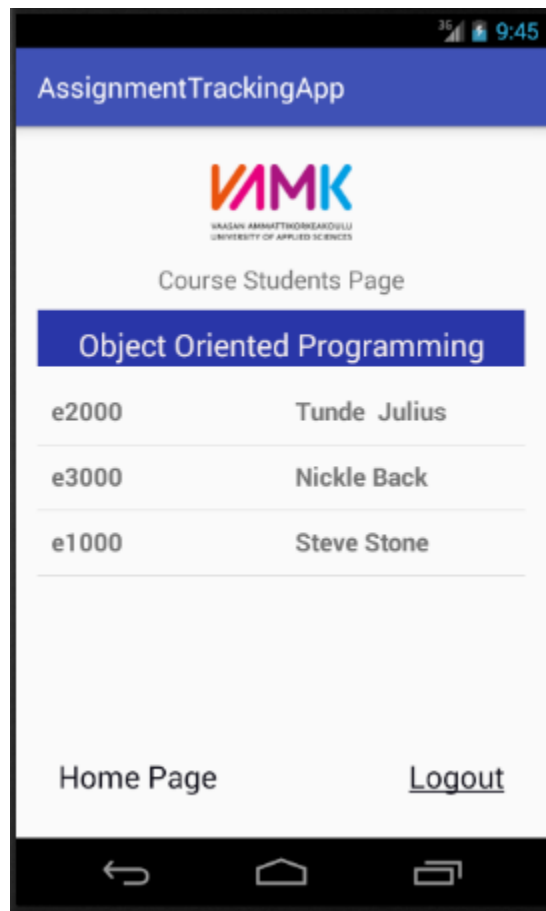


**Figure 25: Course Assignments Page**

### 5.1.5 Course Students Page

This page appears to the teacher when the teacher clicks on the student button on his/her home page and it provides the teacher information about the 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 a clickable logout text which allows the teacher to logout. The figure below represents the user interface for course students.

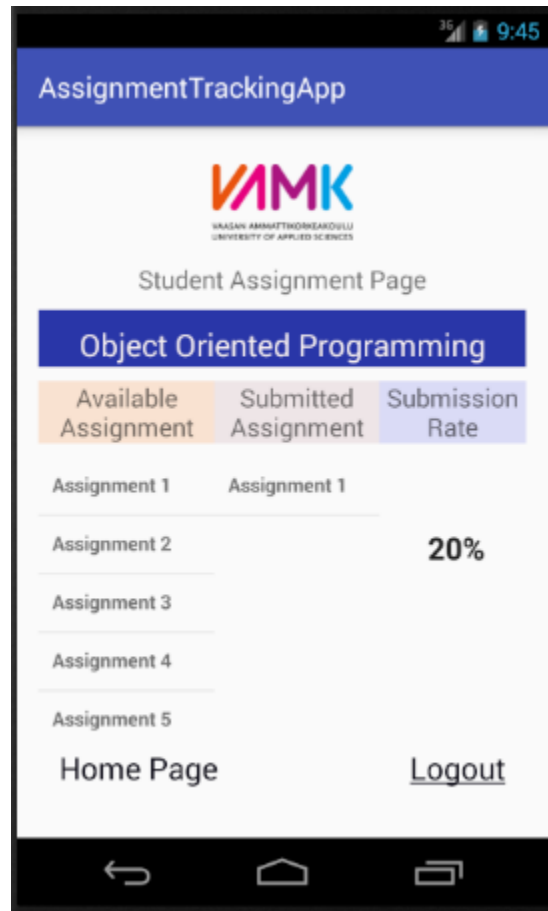


**Figure 26: Course Students Page**

### **5.1.5 Student Assignment Page**

The student assignment page provides the 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 the student assignment page.





**Figure 27: Student Assignment Page**

### 5.1.6 Add Course Page

It appears to the teacher when the teacher clicks on add new course clickable text on his/her home page. This page allows the teacher to add a new course to his/her course list 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 a message to the user in case of error.

AssignmentTrackingApp

VAMK  
VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Add Course Page

Course ID

Name

Description

Start Date

End Date

Course Password

ADD COURSE

Home Page Logout

**Figure 28: Add Course Page**

### 5.1.7 Add Assignment Page

This page appears to the teacher when the teacher clicks on add new assignment clickable text on his/her home page. It allows the teacher to add a new assignment to his/her assignment list for selected course name by providing required data such as course id which is automatically selected in the background when the user selects 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 a message to the user in case of error.

AssignmentTrackingApp

VAMK  
VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Add Assignment Page

Computer Network

Assignment ID

Description

Start Date Start Time

End Date End Time

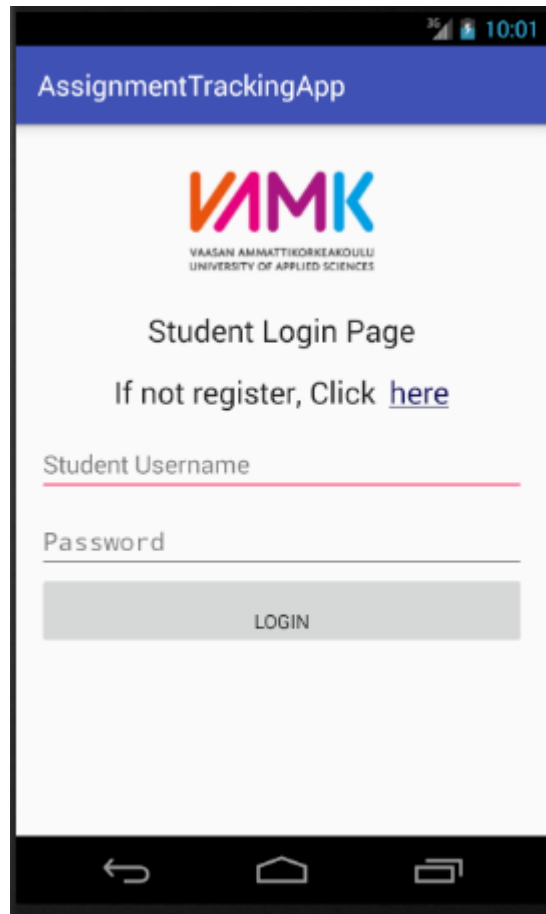
ADD ASSIGNMENT

Home Page [Logout](#)

**Figure 29: Add Assignment Page**

## 5.2 Student Login Page

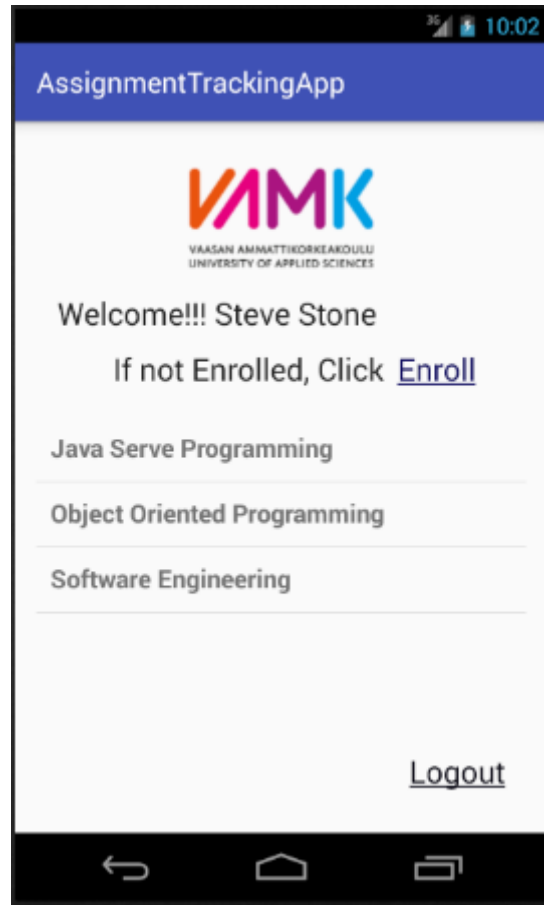
This page was designed with xml and allows students to login to the student home page by providing their registered username, password and click on the login button. This page provides a clickable text link which enables student that has not registered to register, and by clicking on the link, the application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



**Figure 30: Student Login Page**

### 5.2.1 Student Home Page

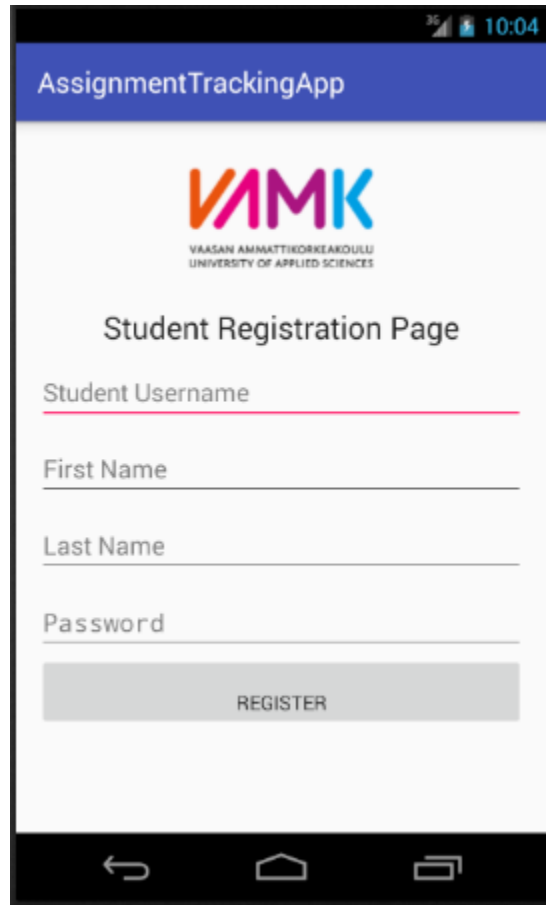
This page appears to the student after the student has successfully logged in from the student login page and this page was designed with xml. This interface provides the student with first name and last name to show that the user is working on the appropriate page. The student home page provides information about clickable courses names which student has enrolled in. There is a clickable enroll text that enables the 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 a clickable logout text that enables the student to logout and the figure below represents the student home page.



**Figure 31: Student Home Page**

### 5.2.2 Student Registration Page

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



The screenshot shows a mobile application interface for 'AssignmentTrackingApp'. At the top, there is a blue header with the app name. Below the header is the VAMK logo, which consists of the letters 'VAMK' in a stylized, multi-colored font, with the full name 'VAASAN AMMATTIKORKEAKOULU UNIVERSITY OF APPLIED SCIENCES' underneath. The main title of the page is 'Student Registration Page'. Below the title, there are four input fields: 'Student Username', 'First Name', 'Last Name', and 'Password'. Each field has a horizontal line below it. At the bottom of the form is a grey button labeled 'REGISTER'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

**Figure 32: Student Registration Page**

### **5.2.3 Course Enrollment Page**

This page appears when the student clicks on the clickable enroll text on his/her home page and it was designed with xml. It allows the student to enroll for the 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 a message to the student in case of course enrollment error. The figure below represents the course enrollment page.

AssignmentTrackingApp

VAMK  
VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Course Enrollment Page

Computer Network

First Name

Last Name

Course Password

ENROLL

**Figure 33: 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 interface 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

The user input data from login activity is declared and the on-create method of the teacher login class initialized the variables data provided by the users. The on-login method responded to user clicking on login button on login page and this method ensures that necessary parameters are provided and then call the teacher login background task.

```
public class TeacherLogin extends AppCompatActivity {
    EditText teacherUsername;
    EditText teacherPassword;
    TextView registerLink;
    TextView teacherLoginMessage;
    static JSONObject jsonObj = null;
    static String json = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_teacher_login);
        teacherUsername= (EditText)findViewById(R.id.teacherEditText);
        teacherPassword= (EditText)findViewById(R.id.passwordEditText);
        teacherLoginMessage = (TextView)findViewById(R.id.teacherLoginError);
        teacherRegistration();
    }
    public void onLogin(View v){
        String teacherID = teacherUsername.getText().toString();
        String password = teacherPassword.getText().toString();
        if(teacherID.isEmpty() || password.isEmpty()){
            teacherLoginMessage.setText("Check Missing Input Value!!!");
        }else {
            TeacherLoginBT teacherLoginBT = new TeacherLoginBT(this);
            teacherLoginBT.execute(teacherID, password);
        }
    }
}
```

#### Code Snippet 1: On create method and on login Method



This involves do-in-background method that communicates to and from the database with the URL link which contains the PHP codes, and the necessary parameters for communication are represented by params which are the parameters provided from the on-login method.

```
@Override
protected JSONObject doInBackground(String... params) {
    String teacher_login_url =
        "http://www.cc.puv.fi/~e1100617/AssignmentTrackingApp/teacherLogin.php";
    username = params[0];
    password = params[1];
```

### Code Snippet 2: Login background task

This is the login on-post-execute method where the response from the database is collected in form of json array and based on the type of response from the database; the user is redirected to another page through intent.

```
studentInfo = json.getJSONArray("teacherInfo");
if(studentInfo.isNull(0)){
    teacherLoginMessage.setText("Logging Error: Incorrect Username or
    Password");
}
JSONObject j = studentInfo.getJSONObject(0);

teacher.setFirstName(j.optString("firstName"));
teacher.setLastName(j.optString("lastName"));
teacher.setPassword(j.optString("password"));
teacher.setUsername(j.optString("userName"));

DB_teacher_Id = teacher.getUserName();
DB_password = teacher.getPassword();

if (DB_teacher_Id.equalsIgnoreCase(username)
    && DB_password.equalsIgnoreCase(password)) {
    Intent intent = new
        Intent("com.example.feranmi.assignmenttrackingapp.TeacherPage");
    intent.putExtra("teacher", teacher);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    activity.startActivity(intent);
}
```

### Code Snippet 3: Login on post execute

## 6.2 Registration

The on-register method responded to the user clicking on register button on registration page and this method ensures that the necessary parameters are provided and then call the teacher registration background task.

```
public void onRegister(View v){

    username = teacherUsername.getText().toString();
    firstname = teacherFirstname.getText().toString();
    lastname = teacherLastname.getText().toString();
    password = teacherPassword.getText().toString();

    if(username.isEmpty() || firstname.isEmpty() || lastname.isEmpty() ||
        password.isEmpty()){
        teacherRegistrationMessage.setText("Check Missing Input Value!!!");
    }else{
        TeacherRegistrationBT teacherRegistrationBT = new
TeacherRegistrationBT(this);
        teacherRegistrationBT.execute(username, firstname, lastname,
password);
    }
}
```

### Code Snippet 4: On register method for register button

This represents registration do-in-background method that communicate to and from the database with the URL link which contain the PHP codes, and the necessary parameters for communication are represented by params which are the parameters provided by the user from registration page.

```
@Override
protected String doInBackground(String... params) {

    String teacher_register_url =
"http://www.cc.puv.fi/~e1100617/AssignmentTrackingApp/teacherRegistration.php";
    username = params[0];
    firstname = params[1];
    lastname = params[2];
    password = params[3];
}
```

### Code Snippet 5: Registration background task

This is the registration on-post-execute method where the response from the database is collected in form of string which informs the user if the registration was successful or not successful.

```
@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    try{

        if(result.equalsIgnoreCase("Registration Success")) {
            teacherRegistrationMessage.setText("Your Registration was
            successful!!!");
        }else if(result.equalsIgnoreCase("Error")) {
            teacherRegistrationMessage.setText("Your Registration was not
            successful!!!");
        }
    }catch(NullPointerException e){
        Log.e("Null Pointer Error", "My error: " + e.toString());
    }
}
```

### Code Snippet 6: Registration on post execute method

## 6.3 Student Home

This is the student enrollment method which redirects user from student home page to the student enrollment page along with the student identity which is student number.

```
public void studentEnrollment(){
    enrollLink = (TextView)findViewById(R.id.studentEnroll);
    enrollLink.setOnClickListener(
        new View.OnClickListener(){

            @Override
            public void onClick(View v) {
                Intent intent = new
                Intent("com.example.feranmi.assignmenttrackingapp.CourseEnrollment
                Page");
                Bundle extras = new Bundle();
                extras.putString("studentNumber",studentNumber);

                intent.putExtras(extras);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
            }
        }
    );
}
```

### Code Snippet 7: Method that redirect to student enrollment page

This section of code snippet shows setting of list adapter for all course names that student has enrolled for on the student home page and click listener is also set for each course name on the list of courses on the student home page.

```

HashMap<String, String> map = new HashMap<String, String>();
map.put(TAG_COURSENAME, course.getName());
map.put(TAG_COURSEID, course.getId());

courseList.add(map);
list=(ListView) findViewById(R.id.studentListView);

ListAdapter adapter = new SimpleAdapter(StudentPage.this, courseList,
    R.layout.studentpage_list_item,new String[] {TAG_COURSENAME}, new int[]
    {R.id.studentCourseName});

list.setAdapter(adapter);

//Setting click listener for each item in Course List View
list.setOnItemClickListener(new AdapterView.OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
    position, long id) {

        String courseName;
        String courseId;
        courseName = courseList.get(+position).get(TAG_COURSENAME);
        courseId = courseList.get(+position).get(TAG_COURSEID);
        Intent intent = new
        Intent("com.example.feranmi.assignmenttrackingapp.StudentAssignmen
        tPage");
        Bundle extras = new Bundle();

        extras.putString("courseName", courseName);
        extras.putString("courseId", courseId);
        extras.putString("studentNumber", studentNumber);

        intent.putExtras(extras);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        activity.startActivity(intent);
    }
});

```

### Code Snippet 8: On click listener for each course on student page

## 6.4 Teacher Home

When the user clicks on add course link on the teacher home page, this section of code snippet shows implementation of add course method which redirects user from the teacher home page to add course page along with the teacher identity which is teacher id.

```
public void addCourse(){
    addCourse = (TextView)findViewById(R.id.add_course);
    addCourse.setOnClickListener(new View.OnClickListener(){

        @Override
        public void onClick(View v){

            Intent intent = new Intent(TeacherPage.this, AddCourse.class);
            intent.putExtra("teacherID", teacherID);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        }
    });
}
```

### Code Snippet 9: Method that redirect to add course page

When the user clicks on add assignment link on the teacher home page, this section of code snippet shows implementation of add assignment method which redirects the user from the teacher home page to add assignment page along with the teacher identity which is teacher id.

```

public void addAssignment(){
    addAssignment = (TextView)findViewById(R.id.add_assignment);
    addAssignment.setOnClickListener(
        new View.OnClickListener(){

            @Override
            public void onClick(View v){

                Intent intent = new Intent(TeacherPage.this,
                AddAssignment.class);
                intent.putExtra("teacherID", teacherID);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
            }
        }
    );
}

```

### Code Snippet 10: Method that redirect to add assignment page

This section of code snippet shows setting of list adapter for each row of data on the teacher home page. The course assignment button, course student button and mark assignment text view for each course name are attached to the list adapter and initialized.

```

ListAdapter adapter = new SimpleAdapter(TeacherPage.this, courseList,
    R.layout.teacherpage_list_item,
    new String[] {TAG_COURSENAME}, new int[] {R.id.teacherCourseName}){
    public View getView(final int position, View convertView, ViewGroup
    parent){
        final String courseName = courseList.get(position).get(TAG_COURSENAME);
        final String courseId = courseList.get(position).get(TAG_COURSEID);

        View itemView = super.getView(position, convertView, parent);

        Button courseAssignment = (Button)
        itemView.findViewById(R.id.course_assignments_button);
        Button studentAssignment = (Button)
        itemView.findViewById(R.id.course_students_button);
        TextView markAssignment = (TextView)
        itemView.findViewById(R.id.teacherCourseName);
    }
}

```

### Code Snippet 11: Setting list adapter for each row in teacher page

## 6.5 Assignment marking

This represents the implementation of on mark method which responded to the user clicking on mark assignment button on assignment marking page and this method ensures that necessary parameters are provided and then call mark assignment background task.

```
public void OnMark(View v){
    try{
        if(student_Number.isEmpty()|| student_Number==null ||
            assignmentId.isEmpty() || assignmentId==null ){

            markAssignmentMessage.setText("Check Missing Input Value!!!");
        }else{

            MarkAssignmentBT markAssignmentBT = new MarkAssignmentBT(this);
            markAssignmentBT.execute(student_Number,assignmentId, courseId);
        }
    }catch(NullPointerException e){
        Log.e("Null Pointer Exception", "For onMark data values: " +
            e.toString());
        markAssignmentMessage.setText("Check Missing Input Value!!!");
    }
}
```

### Code Snippet 12: Method for marking assignment

This section of code snippet represents setting of spinner for selection of the assignment when the teacher is marking the assignment for the student. It shows how an on-item listener is set for selected assignment and using the position of the assignment to get the assignment id and the assignment description.

```

// Spinner on item selected listener
assignmentSpinner
    .setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> arg0,
            View arg1, int position, long arg3) {

            assignmentId = assignmentList.get(position).getId();
            assignmentDescription =
                assignmentList.get(position).getDescription();
        }
        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
            assignmentId = "";
            markAssignmentMessage.setText("You have not selected any
                Assignment!!!");
        }
    });

```

### Code Snippet 13: Spinner for assignment selection

## 6.6 Course Assignments

The below section of code snippet shows the setting of list adapter for course assignment presented through the teacher course assignments list view and this is achieved by attaching the assignment description, assignment due date and assignment completion rate to adapter for each assignment.

```

HashMap<String, String> map = new HashMap<String, String>();
map.put(TAG_DESCRIPTION, assignment.getDescription());
map.put(TAG_COMPLETIONRATE, assignment.getCompletionRate().toString());
map.put(TAG_DUEDATETIME, destFormat.format(assignment.getEndDateTime()));
map.put(TAG_ID, assignment.getId());

assignmentList.add(map);
list=(ListView) findViewById(R.id.teacherCourseAssignmentsListView);

ListAdapter adapter = new SimpleAdapter(CourseAssignmentsPage.this,
assignmentList,
    R.layout.teachercourse_assignmentspage_list_item,
    new String[] {TAG_DESCRIPTION, TAG_DUEDATETIME, TAG_COMPLETIONRATE},
    new int[] {R.id.assignmentDescription, R.id.assignmentDueDate,
        R.id.assignmentCompletionRate});
list.setAdapter(adapter);

```

### Code Snippet 14: List adapter for course assignment



## 6.7 Student Assignment

This section of code snippet shows the setting of list adapter for the available assignment presented to the users. The available assignment data is received from the database in form of json array and the assignment description part is received through json object. The get count method of list adapter is used to get the number of the available assignment.

```
try {
    availableAssignment = json.getJSONArray("availableAssignment");
    for (int i = 0; i < availableAssignment.length(); i++) {
        JSONObject j = availableAssignment.getJSONObject(i);
        assignment.setDescription(j.getString("description"));

        HashMap<String, String> map = new HashMap<String, String>();
        map.put(TAG_COURSEDESCRIPTION, assignment.getDescription());

        assignmentList.add(map);
        list=(ListView) findViewById(R.id.studentAvailableAssignmentListView);

        ListAdapter adapter = new SimpleAdapter(StudentAssignmentPage.this,
            assignmentList, R.layout.available_assignment_list_item,
            new String[] {TAG_COURSEDESCRIPTION}, new int[]
            {R.id.studentAvailableAssignment});

        list.setAdapter(adapter);
    }
    availableAssignmentCount = list.getAdapter().getCount();
}
```

### Code Snippet 15: List adapter for available assignment

This section of code snippet shows the setting of list adapter for submitted assignment presented to the users. The submitted assignment data is received from database in form of json array and the assignment description part is received through json object. The get count method of list adapter is used to get the number of the submitted assignment.

```

try {
    submittedAssignment = json.getJSONArray("submittedAssignment");
    for (int i = 0; i < submittedAssignment.length(); i++) {

        JSONObject j = submittedAssignment.getJSONObject(i);
        assignment.setDescription(j.getString("description"));

        HashMap<String, String> map = new HashMap<String, String>();
        map.put(TAG_COURSEDESCRIPTION, assignment.getDescription());

        assignmentList.add(map);
        list=(ListView) findViewById(R.id.studentSubmittedAssignmentListView);
        ListAdapter adapter = new SimpleAdapter(StudentAssignmentPage.this,

            assignmentList, R.layout.submitted_assignment_list_item,
                new String[] {TAG_COURSEDESCRIPTION}, new int[]
                    {R.id.studentSubmittedAssignment});

        list.setAdapter(adapter);
    }
    //This is where submission Rate Value is calculated and set
    submittedAssignmentCount = list.getAdapter().getCount();
}

```

### **Code Snippet 16: List adapter for submitted assignment**

## **6.8 Add Course**

The below section of code snippet represents implementation of on add course method which responded to user clicking on add course button on add course page and this method ensures that the necessary parameters are provided and then call add course background task.

```

public void onAddCourse(View v){
    courseId = idEditText.getText().toString();
    courseName = nameEditText.getText().toString();
    description = descriptionEditText.getText().toString();
    startDate = startDateEditText.getText().toString();
    endDate = endDateEditText.getText().toString();
    coursePwd = coursePasswordEditText.getText().toString();

    try{
        if(courseId.isEmpty() || courseName.isEmpty() || description.isEmpty()
           || startDate.isEmpty() || endDate.isEmpty() || coursePwd.isEmpty()){

            addCourseMessage.setText("Check Missing Input Value!!!");
        }else{
            AddCourseBT addCourseBT = new AddCourseBT(this);
            addCourseBT.execute(courseId,courseName,description,
                               startDate,endDate,coursePwd,teacherID);
        }
    }catch(NullPointerException e){
        Log.e("Null Pointer Error", "My error: " + e.toString());
        addCourseMessage.setText("Check Missing Input Value!!!");
    }
}

```

### Code Snippet 17: Method for add course button

## 6.9 Add Assignment

This represents the implementation of an on-add assignment method which responded to the user clicking on the add assignment button on add assignment page and this method ensures that the necessary parameters are provided and then call add assignment background task.

```

try{
    if(assignmentId.isEmpty() || description.isEmpty() || startDate.isEmpty()
       || startTime.isEmpty() || endDate.isEmpty() || endTime.isEmpty() ||
       courseId.isEmpty()){
        addAssignmentMessage.setText("Check Missing Input Value!!!");
    }else{
        AddAssignmentBT addAssignmentBT = new AddAssignmentBT(this);
        addAssignmentBT.execute(assignmentId,description,
                               startDate,endDate,courseId);
    }
}catch(NullPointerException e){
    Log.e("Null Pointer Error", "My error: " + e.toString());
    addAssignmentMessage.setText("Check Missing Input Value!!!");
}

```

### Code Snippet 18: Method for add assignment button

## 6.10 Course Enrollment

This represents the implementation of an on-enroll method which responded to the user clicking on the enroll button on the enrollment page and this method ensures that the necessary parameters are provided and then call student enrollment background task.

```
public void onEnroll(View v){

    firstname = studentFirstname.getText().toString();
    lastname = studentLastname.getText().toString();
    coursePwd = coursePassword.getText().toString();

    if(firstname.isEmpty() || lastname.isEmpty() ||
        coursePwd.isEmpty() || courseId.isEmpty() ){

        courseEnrollmentMessage.setText("Check Missing Input Value!!!");
    }else{

        StudentEnrollmentBT studentEnrollmentBT =new StudentEnrollmentBT(this);
        studentEnrollmentBT.execute(studentNumber,courseId, coursePwd);
    }
}
```

### Code Snippet 19: Method for enroll button

## 6.11 Android Manifest

The below section of code snippet shows the use of internet permission for the application which means that the availability of internet on the user's mobile phone is important for the application to work. It shows all the activities involved in the application with intent filter embedded in each activity. The activity with main and launcher inside the embedded intent filter indicates the first page activity of the application that is presented to the users when the application is launched.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.feranmi.assignmenttrackingapp">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".TeacherLogin">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```

### **Code Snippet 20: Manifest with internet permission and launcher**

The below section of code snippet represents some other activities involved in the application. It shows the activity with the embedded intent filter for each activity. The intent filter contains information for referring to each page of the application and apart from first page activity which is categorized as launcher, other activities of the application are categorized as default.

```
<!-- This is the activity for teacher page -->
<activity android:name=".TeacherPage">
    <intent-filter>
        <action
            android:name="com.example.feranmi.assignmenttrackingapp.TeacherPage" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>

<!-- This is the activity for add course page -->
<activity android:name=".AddCourse">
    <intent-filter>
        <action
            android:name="com.example.feranmi.assignmenttrackingapp.AddCourse" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>

<!-- This is the activity for add assignment page -->
<activity android:name=".AddAssignment">
    <intent-filter>
        <action
            android:name="com.example.feranmi.assignmenttrackingapp.AddAssignment" />

            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
```

### **Code Snippet 21: 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 table represents the testing template used and the result of the testing.

**Table 2: Application Testing Template**

No	Test Case Description	Steps	Expected Result	Actual Result	(Pass/Fail)
1.	Check for necessary fields and buttons on login page.	<ol style="list-style-type: none"> <li>1. Installed the application.</li> <li>2. Runs the application and view login page.</li> </ol>	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.	<ol style="list-style-type: none"> <li>1. Click on login button without entering any values for username and password.</li> </ol>	The login page should display error; check missing inputs.	OK	Pass
3.	Check for Incorrect inputs on login page.	<ol style="list-style-type: none"> <li>1. Enter incorrect value for either username or password</li> <li>2. Click on login button.</li> </ol>	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.	<ol style="list-style-type: none"> <li>1. Enter correct value for username and password</li> <li>2. Click on login button.</li> </ol>	The login page should redirect to home page.	OK	Pass
5.	Check for clickable text on login page.	<ol style="list-style-type: none"> <li>1. Click on register link on login page.</li> </ol>	The user should be redirected to registration page.	OK	Pass
6.	Check for empty inputs on registration page.	<ol style="list-style-type: none"> <li>1. Click on register link on login page.</li> <li>2. Click on register button on registration page without inputting all necessary data.</li> </ol>	The registration page should display error; check missing inputs.	OK	Pass
7.	Multiple entries of data for registration.	<ol style="list-style-type: none"> <li>1. Click on register link on login page.</li> <li>2. Enter existing registration data</li> <li>3. Click on register button.</li> </ol>	The login page should display error; registration error, already exist.	OK	Pass
8.	Check for logout clickable text.	<ol style="list-style-type: none"> <li>1. User login to his/her home page.</li> <li>2. Click on logout link.</li> </ol>	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.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and navigate.</li> <li>2. Click on home page link.</li> </ol>	The user should be redirected to home page.	OK	Pass
10.	Check for empty inputs for enrollment.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and click on enroll link.</li> <li>2. Click on enroll button without inputting all necessary data.</li> </ol>	The enrollment page should display error; check missing inputs.	OK	Pass
11.	Multiple entries of data for enrollment.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and click on enroll link.</li> <li>2. Enter existing enrollment data.</li> <li>3. Click on enroll button.</li> </ol>	The enrollment page should display error; enrollment error, already exist.	OK	Pass
12.	Student checking course assignments submission rate.	<ol style="list-style-type: none"> <li>1. User login to his/her home page.</li> <li>2. Click on course to check.</li> </ol>	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.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and click on add new course link.</li> <li>2. Click on add course button without inputting all necessary data.</li> </ol>	The add course page should display error; check missing inputs.	OK	Pass
14.	Multiple entries of data for adding course.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and click on add new course link.</li> <li>2. Enter existing course data and click on add course button.</li> </ol>	The add course page should display error; error, already exist.	OK	Pass
15.	Checks for empty inputs for adding assignment.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and click on add new assignment link.</li> <li>2. Click on add assignment button without inputting all necessary data.</li> </ol>	The add assignment page should display error; check missing inputs.	OK	Pass
16.	Multiple entries of data for adding assignment.	<ol style="list-style-type: none"> <li>1. User login to his/her home page and click on add new assignment link.</li> <li>2. Enter existing assignment data and click on add assignment button.</li> </ol>	The add assignment page should display error; error, already exist.	OK	Pass

<b>No</b>	<b>Test Case Description</b>	<b>Steps</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>(Pass /Fail)</b>
17.	Checks for empty inputs for marking assignment.	<ol style="list-style-type: none"><li data-bbox="500 317 919 470">1. User login to his/her home page and click on course name.</li><li data-bbox="500 478 919 632">2. Click on mark assignment button without inputting all necessary data.</li></ol>	The mark assignment page should display error; check missing inputs.	OK	Pass

## **8 SUMMARY**

This project involved an easy and mobile way of tracking student involvement in providing complete and required assignment solutions by developing an Android operating system mobile application. The application allows the teacher to register, to add course, add assignment, update and delete assignment, mark assignment for the student with correct assignment solution, check the student's assignment submission rate and check assignment completion rate. The application allows the student to register, to enroll for the course, check the available and submitted assignments and check assignment submission rate.

The targeted group tentatively involved VAMK teachers and students and after the development and completion of this study, the project was tested on android mobile device in order to confirm that the required objectives and goals of this study are achieved.

## **9 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.

### **9.1 Challenges**

The challenges in the development of this project include creating a 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.

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

### **9.2 Improvements**

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

## REFERENCES

- /1/ Android (Operating System). Accessed 11.10.2016  
[https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](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](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](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>