UNIVERSITY OF DAR ES SALAAM



COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGIES (CoICT)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

CS 499 FINAL PROJECT REPORT

A Project Report in Partial Fulfillment for the Award of Bachelor of Science in Computer Engineering & Information Technology

|  |  |
| --- | --- |
| **Project Title:** | Web and Android Application for Student Course Evaluation |
| **Degree Program:** | B.Sc. Computer Engineering & Information Technology |
| **Student Name:** | Deo, Goodluck |
| **Registration Number:** | 2010-04-00537 |
| **Project Supervisor’s Name:** | Dr. S. Lujara |
| **Submission Date:** | Thursday, June 19, 2014 |
| **Project Supervisor’s Signature:** | ……………………………. |

# Declaration

I, DEO, Goodluck declare to the best of my knowledge that, this project is an origin piece of my work done and has not been reproduced from any piece. The work in this report was carried out in accordance with the Regulations of the University of Dar es Salaam and has not been presented to any other University for examination either in Tanzania or overseas. However, whatever references used such as Books, Standard specifications, offline and online Reports on pursuing this project are clearly shown in references list.

Signature: …………………………. Date: June 19, 2014

# Acknowledgement

First of all I would like to thank our Almighty God for providing me with the good health and the strength within the entire life time. I would like also to thank my parents for promoting me to where I am right now, may God bless them and provide them with more years to live. Also I would like to thank the Project coordinator Eng. Daudi Mnyanghwalo and the examining panel from the Department of Computer Science and Engineering, College of Information and Communication Technologies for accepting this project.

Special thanks to my Project Supervisor Dr. Suzan Lujara for her necessary guidance concerning project progress, may our almighty God bless her for her kindness.

Without forgetting to pass my special thanks to the Quality Assurance Bureau (QAB) Officials under their Associate Director Dr. Mgwatu for showing the super collaboration and cooperation towards the accomplishment of this project.

Likewise, special thanks to my entire fellow student, especially Mr. Mwatima Ramadhani whom I use to study with and share the knowledge we have and which the lecturers provides to us. Appreciate all of them.

# Abstract

The Quality Assurance Bureau (QAB) has the mandate to examine the quality of teaching and learning at the University of Dar es Salaam (UDSM). In this aspect, QAB has been administering paper course evaluations, in this process a member of Academic Staff who is not responsible for the course being evaluated oversees the whole activity, from handling, distributing, collecting and finally compiling the raw data for QAB for further action.

Following the use of Waterfall model as a model for Software Development and the Object-Oriented System Design and Analysis, the Web and Android Application for an Online Course Evaluation System is under development where by it will eventually address the challenges currently facing the traditional evaluation exercise that includes but not limited to difficulties for instance course evaluation being time consuming exercise, difficult to maintain due to course evaluation forms taking a lot of office space and the cost of producing questionnaires.

The purpose for transitioning to an online evaluation process is to improve the assessment, feedback and actions taken to advance the quality of academic programs offered. Online system promise lower costs, timelier feedback for lecturers and more anonymity for students, and flexibility in questionnaire and report design.

# Abbreviations

ADT Android Developer Tool

ARIS Academic Registration Information System

CSE Computer Science and Engineering

CSS3 Cascade Style Sheet 3

CSV Comma Separated Values

CoICT College of Information and Communication Technologies

CVL Center for Virtual Learning

DBMS Database Management System

HTML5 Hypertext Mark-up Language 5

JSON JavaScript Object Notation

LEF Learning Environment and Facilities

OOSDA Object Oriented System Design and Analysis

PDF Portable Document File

PNG Portable Network Graphics

QAB Quality Assurance Bureau

SPSS Statistical Packages for the Social Sciences

UDSM University of Dar es Salaam

UI User Interface

UML Unified Modeling Language

XAMPP Cross Apache MySQL PHP Perl

XML eXtensible Mark-up Language

**Table of Contents**

[Declaration ii](#_Toc390883231)

[Acknowledgement iii](#_Toc390883232)

[Abstract iv](#_Toc390883233)

[Abbreviations v](#_Toc390883234)

[Table of Figures ix](#_Toc390883235)

[List of Tables x](#_Toc390883236)

[CHAPTER ONE: INTRODUCTION 1](#_Toc390883237)

[1.1 Background 1](#_Toc390883238)

[1.2 Problem Statement 1](#_Toc390883239)

[1.3 Remedy to the Problem 2](#_Toc390883240)

[1.4 Objectives 3](#_Toc390883241)

[1.4.1 Main Objectives 3](#_Toc390883242)

[1.4.2 Specific Objectives 3](#_Toc390883243)

[1.5 Scope of the Project 3](#_Toc390883244)

[1.6 Significance of the Project 3](#_Toc390883245)

[CHAPTER TWO: LITERATURE REVIEW 4](#_Toc390883246)

[2.1 Students Course Evaluation 4](#_Toc390883247)

[2.1.1 Summative Evaluation 4](#_Toc390883248)

[2.1.2 Formative Evaluation 4](#_Toc390883249)

[2.2 UDSM Students Course Evaluation 5](#_Toc390883250)

[2.3 Related Works 6](#_Toc390883251)

[2.4 Software Development Models 7](#_Toc390883252)

[2.5 Mobile and Web Technologies 7](#_Toc390883253)

[2.6 Object-Oriented System Design 8](#_Toc390883254)

[2.6.1 Objects 8](#_Toc390883255)

[2.6.2 Class 9](#_Toc390883256)

[2.6.3 Attributes 9](#_Toc390883257)

[2.6.4 Operations 9](#_Toc390883258)

[2.7 Unified Modeling Language (UML) 9](#_Toc390883259)

[2.7.1 UML Diagrams 10](#_Toc390883260)

[2.8 Tools and Technologies 10](#_Toc390883261)

[2.8.1 Bootstrap 3.0.3 10](#_Toc390883262)

[2.8.2 Android Developer Tools (ADT) 10](#_Toc390883263)

[2.8.3 PHP Code Igniter Framework 11](#_Toc390883264)

[2.8.4 Star UML 11](#_Toc390883265)

[2.8.5 XAMPP 11](#_Toc390883266)

[2.8.6 Adobe Photoshop CS6 11](#_Toc390883267)

[2.8.7 Notepad++ & phpDesigner8 11](#_Toc390883268)

[2.8.8 Operating Systems 12](#_Toc390883269)

[2.8.9 MySQL Workbench 12](#_Toc390883270)

[2.9 PHP CodeIgniter Framework 12](#_Toc390883271)

[2.9.1 CodeIgniter as MVC Framework 13](#_Toc390883272)

[2.10 Bootstrap Framework 14](#_Toc390883273)

[2.10.1 Bootstrap file structure 14](#_Toc390883274)

[CHAPTER THREE: METHODOLOGY 16](#_Toc390883275)

[3.1 Introduction 16](#_Toc390883276)

[3.2 The Waterfall Model 16](#_Toc390883277)

[3.2.1 Requirement Capture and Specifications 17](#_Toc390883278)

[3.2.2 System Analysis 17](#_Toc390883279)

[3.2.3 System Design 17](#_Toc390883280)

[3.2.4 Object Design 17](#_Toc390883281)

[3.2.5 Implementation 17](#_Toc390883282)

[3.2.6 Evaluation and Testing 18](#_Toc390883283)

[3.3 System Architecture 18](#_Toc390883284)

[3.4 Data Collection Methods 18](#_Toc390883285)

[CHAPTER FOUR: REQUIREMENTS CAPTURE AND ANALYSIS 20](#_Toc390883286)

[4.1 Requirements Capture 20](#_Toc390883287)

[4.2 Requirement Analysis 20](#_Toc390883288)

[4.2.1 Functional Requirements 20](#_Toc390883289)

[4.2.2 Non-functional Requirements 21](#_Toc390883290)

[4.3 Use Case Modeling 22](#_Toc390883291)

[4.3.1 Basic Use Case Notation 23](#_Toc390883292)

[4.3.2 Use Cases Identification 23](#_Toc390883293)

[4.3.3 Use Case Description 25](#_Toc390883294)

[4.3.4 Conceptual Model Diagram 27](#_Toc390883295)

[CHAPTER FIVE: SYSTEM DESIGN 28](#_Toc390883296)

[5.1 Overview 28](#_Toc390883297)

[5.2 System Sequence Diagram 28](#_Toc390883298)

[5.3 Design Class Diagram 30](#_Toc390883299)

[5.4 Database Design 31](#_Toc390883300)

[5.4.2 Entity Relationship Diagram 32](#_Toc390883301)

[CHAPTER SIX: SYSTEM IMPLEMENTATION 33](#_Toc390883302)

[6.1 Overview 33](#_Toc390883303)

[6.2 PART ONE: iEvaluate Web Based Application 33](#_Toc390883304)

[6.2.1 Creation of iEvaluate Database in a Server 33](#_Toc390883305)

[6.2.2 User Interface Design and Implementation 34](#_Toc390883306)

[6.2.3 iEvaluate Front-end Modules Implementation 35](#_Toc390883307)

[6.2.4 iEvaluate Administrator Modules Implementation 38](#_Toc390883308)

[6.3 PART TWO: iEvaluate Android Application 43](#_Toc390883309)

[6.3.1 Login Module 43](#_Toc390883310)

[6.3.2 Perform Evaluation Module 44](#_Toc390883311)

[CHAPTER SEVEN: SYSTEM TESTING 45](#_Toc390883312)

[7.1 Overview 45](#_Toc390883313)

[7.2 Module Testing 45](#_Toc390883314)

[7.3 System Testing 46](#_Toc390883315)

[CHAPTER EIGHT: SUMMARY, CONCLUSION AND RECOMMENDATIONS 47](#_Toc390883316)

[8.1 Summary 47](#_Toc390883317)

[8.2 Challenges Faced 47](#_Toc390883318)

[8.3 Conclusion 47](#_Toc390883319)

[8.4 Recommendation 48](#_Toc390883320)

[REFERENCES 49](#_Toc390883321)

[APPENDICES 50](#_Toc390883322)

[APPENDIX A: Project Time Schedule and Budget 50](#_Toc390883323)

[APPENDIX B: Use Case Description Template and Sample Use Case Descriptions 51](#_Toc390883324)

[APPENDIX C: iEvaluate Screenshots and Sample Output PDF Reports 53](#_Toc390883325)

# Table of Figures

[Figure 2.1: UDSM Course Evaluation Form, Part B: The Instructor 6](#_Toc390933818)

[Figure 2.2: Class Name, Attributes and Operations 9](#_Toc390933819)

[Figure 2.3: Model-View-Controller process loop 13](#_Toc390933820)

[Figure 2.4: Bootstrap Framework File Structure 15](#_Toc390933821)

[Figure 3.1: Waterfall Model Stages 16](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933822)

[Figure 3.2: The 3-Tier Architecture 18](#_Toc390933823)

[Figure 4.1: Basic Use Case Notation 23](#_Toc390933824)

[Figure 4.2: Use Case Diagram for Student 24](#_Toc390933825)

[Figure 4.3: Use Case Diagram for Admin 25](#_Toc390933826)

[Figure 4.4: System Conceptual Model Diagram 27](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933827)

[Figure 5.1: Sequence Diagram for Perform Evaluation Use Case 28](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933828)

[Figure 5.2: Sequence Diagram for Add Instructor Details Use Case 29](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933829)

[Figure 5.3: Sequence Diagram for Edit Instructor Details Use Case 29](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933830)

[Figure 5.4: System Design Class Diagram 30](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933831)

[Figure 5.5: System Entity Relationship Diagram 32](#_Toc390933832)

[Figure 6.1: iEvaluate Database View from XAMPP phpMyAdmin 34](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933833)

[Figure 6.2: iEvaluate Web Based Application Front-end Homepage 35](#_Toc390933834)

[Figure 6.3: Login Module Illustration with Input Fields Required and if Invalid Login Credentials are submitted 36](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933835)

[Figure 6.4: Timeframe Evaluation Feature. 37](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933836)

[Figure 6.5: Perform Evaluation Module: The Instructor Part 38](#_Toc390933837)

[Figure 6.6: Admin Evaluation Setup Module 39](#_Toc390933838)

[Figure 6.8: Instructor’s Details List 41](#_Toc390933839)

[Figure 6.9: Course Ratings Report Form 42](#_Toc390933840)

[Figure 6.10: Course Ratings Report 42](#_Toc390933841)

[Figure 6.11: iEvaluate Android Application Login Form 43](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933842)

[Figure 6.12: iEvaluate Android Application Perform Evaluation Module with Timeframe Feature 44](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933843)

[Figure 7.1: Add Department Module Testing, Check for Required Fields and Invalid Data 45](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933844)

[Figure 7.2: Add Department Module Testing on Success 46](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933845)

[Figure C1: iEvaluate Android Application Sample Screenshots 53](file:///F:\UNIVERSITY\FINAL%20PROJECT\Reports\GD3-Final%20Report%20CS499.docx#_Toc390933846)

[Figure C2: iEvaluate Sample Instructor Ratings PDF Report 54](#_Toc390933847)

[Figure C3: iEvaluate Sample Venue Table and Chairs Improvement PDF Report 55](#_Toc390933848)

# List of Tables

[Table 4.1: System Functional Requirements 21](#_Toc390691748)

[Table 4.2 : System Non-functional Requirements 22](#_Toc390691749)

[Table 4.3: System Actors and Initiated Events 24](#_Toc390691750)

[Table 4.4: Perform Evaluation Use Case 26](#_Toc390691751)

[Table 4.5: Submit Evaluation Form Use Case 26](#_Toc390691752)

[Table A1: Time Schedule for Semester One 50](#_Toc390691753)

[Table A2: Time Schedule for Semester Two 50](#_Toc390691754)

[Table A3: Budget Proposal 51](#_Toc390691755)

[Table B1: Use Case Description Template 51](#_Toc390691756)

[Table B2: Add Course Use Case 52](#_Toc390691757)

[Table B3: Add Instructor Details Use Case 52](#_Toc390691758)

# CHAPTER ONE: INTRODUCTION

## Background

Most of universities around the world have the tendency of evaluating course lecturers against the teaching service they provide to their students. At the end of each semester in each academic year students are provided with evaluation forms for which they have to rate and give comments on how they have being taught by their lecturers in their respective courses so as to improve student’s academic performance. In most cases, this exercise is accomplished by using course evaluation forms either online or manually. The evaluation form aims at capturing feedback from students regarding the quality of course instructions they have received as well as the learning environment and facilities. The information is confidential and will not be associated with student’s identity at the end of the day, students’ honest and constructive opinions are very useful in improving delivery and quality of the course and the learning environment looking forward in enhancing the academic performance of the students in general.

## Problem Statement

The current way of collecting evaluation information from the students in most of universities particularly the University of Dar es Salaam (UDSM) is not done effectively. This is due to unpredictable number of students’ who can actually fill in the forms. This may happen due to the fact that sometimes not all students are present in the class by the time evaluation forms are being distributed hence missing some students’ responses. Also, the evaluation process becomes somehow difficult and may take a lot of time to some point since students have to evaluate most of their lecturers in different times. Always there is tendency of waiting for the evaluation forms to be ready but with no specific time i.e. by what time they will be ready. The current manual system have become somehow inefficient since it is time consuming, tedious and in some cases not practical due to the bad timing of evaluation which sometimes fall during the examination period and not capturing all students. Time consuming as in the sense that each student depending on the number of taught courses has to evaluate lecturers equal to the number of taught courses this means one evaluation form for each lecturer and always not at the same time.

On other hand how data collected is compiled, processed and the results reported is not that much precise. According to Quality Assurance Bureau (QAB) officials how they conduct this evaluation process, they distribute the hard copies of evaluation forms as usual. After the forms being filled by the students they collect the forms and count them manually and generate an excel file with total number of forms as the response from students with respect to academic units. After having that excel file they calculate the average number of responses for each particular questions and finally comes the analysis stage. In the analysis stage, they use Statistical Packages for the Social Sciences (SPSS) which is a comprehensive and flexible statistical and data management tool. SPSS takes in the excel file and generates tabulated reports and charts. From there it can be seen that the evaluation process have two stages as for the case of QAB, stage one is manual i.e. counting the forms and the other stage is somehow automated i.e. the use of SPSS tool. This still remains as the problem since it is time consuming and the chances of making errors in the analysis are high.

## Remedy to the Problem

The purpose of this project is to design and implement the system which will automate the current way of conducting evaluation in order to reduce the challenges which are brought by that current manual system and improve the processing of the evaluation data submitted by the students. Automating the current way of conducting evaluation will be a crucial factor as in the sense that it will save time in carrying out the process not only the time that will be saved but as well as money as in terms of incurred cost of printing a number of copies of the evaluation forms. The proposed project will also aid in producing reports quickly and accurately whenever required, also providing a back-up for the submitted data in soft form.

Developing a Web based application as well as an Android application for which students using their respective registration number, degree program and the year of study, they will be able to perform or fill in the evaluation forms which are there online for all of their respective lecturers for which a student is being taught by. Students with smartphones running Android Operating System will also have the advantage of using the app which will be available purposely for conducting course evaluation. For those students who do not have smartphones running Android Operating System will have the ability to perform the evaluation process through web interface. For this case will make the process being much faster and precisely since one student can evaluate his/her respective lecturers at once whenever he/she have access to his/her respective online account.

## Objectives

### Main Objectives

The main objective of this project is to design and implement the Web application as well as Android application for student course evaluation; it’s basically automating the current manual system of conducting course evaluation specifically at the University of Dar es Salaam.

### Specific Objectives

The specific objectives are as follows:

1. To identify and analyze the specific user requirements for the system.
2. To design the web interfaces this reflects the physical evaluation form.
3. To design a database this will store the results submitted by the students.
4. To develop a mobile application in Android Platform this speed up the evaluation process

## Scope of the Project

The scope of this project will cover students’ course evaluation of the University of Dar es Salaam through Web based and Android applications.

## Significance of the Project

Transitioning from the current manual evaluation process to online evaluation process will improve the assessment, feedback and actions taken to advance the quality of academic programs offered. Online system promise lower costs, timelier feedback for lecturers and more flexibility in questionnaire and report design.

# CHAPTER TWO: LITERATURE REVIEW

## Students Course Evaluation

A course evaluation is a paper or electronic questionnaire, which requires a written or selected response answer to a series of questions in order to evaluate the instructor of a given course. The term may also refer to the completed survey form or a summary of responses to questionnaires. There are means to produce useful feedback which the lecturers and the university can use to improve their academic quality. The process of gathering information about the impact of learning and of teaching practice on student learning, analyzing and interpreting this information, and responding to and acting on the results, is valuable for several reasons. They are beneficial because lectures can review how others interpret their teaching methods, thereby improving their teaching standards. The information can be also used by administrators, along with other input, to make summative decisions (e.g., decisions about promotion, tenure, salary increases, etc.) and make formative recommendations (e.g., identify areas where a faculty member needs to improve).Course evaluations are implemented in one of two ways, either summative or formative. (Wikipedia, Course Evaluation, 2013)

### Summative Evaluation

Summative evaluation occurs at the end of a semester, usually a week or two before the last day of class. The evaluation is performed by the current students of the class. Students have the option to reflect on the lecturers’ instruction without fear of punishment because course evaluations are completely confidential and anonymous. This can be done in one of two ways; either with a paper form or with online technology.

### Formative Evaluation

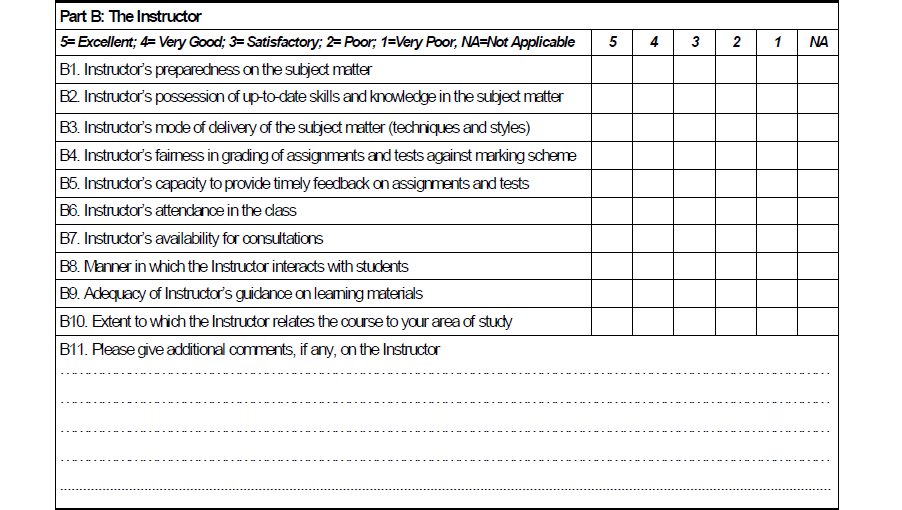
Formative evaluation typically occurs when changes can take place during the current semester, although many institutions consider written comments on how to improve formative as well. Typically this form of evaluation is performed by peer consultation. Other experienced lecturers will review one of their peer’s instructions. The purpose of this evaluation is for the lecturer to receive constructive criticism on teaching. Generally, peer lecturers will sit in on a few lessons given by the lecturer and take notes on their methods. Later on the team of peer lecturers will meet with the said lecturer and provide useful, non-threatening feedback on their lessons.

The peer team will offer suggestions on improvement, which the said lecturer can choose to implement.

## UDSM Students Course Evaluation

During the academic year 2011/2012, the QAB in collaboration with the Center for Virtual Learning (CVL) piloted an online course evaluation exercise using Moodle Learning Management System. The purpose of that pilot study was to examine the viability of the exercise and any challenges that need to be addressed in case of successful online evaluation. That eventually addresses the challenges currently facing the traditional evaluation exercise that includes but no limited to difficulties for instance course evaluation being a time consuming exercise, difficult to maintain due to course evaluation forms taking a lot of office space and the cost of producing questionnaires. There were challenges which they faced by the time they conduct that pilot study which leads up to now days not to have an online course evaluation. (Navwa, 2012)

Not only the University of Dar es Salaam which currently do not have an online evaluation system but also most of the universities in Tanzania there is no any kind of automated system which facilitates lecturer’s evaluation process. In most cases the current technology that is being used is of those printed pages which contain some crucial questions for which will be used to evaluate respective lecturer. The University of Dar es Salaam course evaluation form is divided into four main parts; there is Part A which reflects the particulars such as Course Title and Course Code, Instructor’s name, Department, College/School, student’s Program and Year of study, Part B basically deals with Instructor particulars mainly in relation to the course, Part C is the part for the course itself deeply reflects the course objective and how helpful was the course in relation to the students, and finally Part D which is concerned with the teaching environment and the facilities. So all of these are prepared in the piece of paper ready to be printed and filled by the student. After completions all of these evaluation forms are collected and the analysis stage comes on hand. The forms are counted manually and an excel file is generated with total number of forms as the response from students with respect to academic units. After having that excel file the average number of responses for each particular questions is calculated and finally comes the report. In the analysis stage QAB officials uses SPSS which is a comprehensive and flexible statistical and data management tool. SPSS takes in the excel file and generate tabulated reports and charts. From there we can see that the evaluation process have two stages as for the case of QAB, stage one is manual i.e. counting the forms and the other stage is somehow automated i.e. the use of SPSS tool. Figure 2.1 shows a sample course evaluation Part B for the University of Dar es Salaam for Undergraduate Programs.



##### Figure 2.1: UDSM Course Evaluation Form, Part B: The Instructor

## Related Works

There are number of Universities around the world which employs the online course evaluation system. Get to mention some of them; The University of Oregon in Eugene, Oregon, United States uses an online course evaluation system to collect student feedback about courses and lecturers at the end of each semester. [Access](https://duckweb.uoregon.edu/) to the online course evaluation system (for students completing evaluations, faculty accessing reports, and administrators reviewing results) is available through DuckWeb. [Results](http://courseevals.uoregon.edu/) from instructor and course evaluations, 1997-2007 (collected via paper forms). Results from instructor and course evaluations, spring 2008-present (collected via the online system) are available in [DuckWeb/Course Evaluations menu](https://duckweb.uoregon.edu/). To have access to the results one must have the login credentials so as to view the results. (Affiliates, 2013).

There is another online system known as EvaluationKit which perfoms course evaluation. EvaluationKit is Web Application which is integrated with mobile application in both iOS and Android platforms. EvaluationKIT makes it simple to incorporate these mobile devices into your institution’s evaluation and survey strategies. (EvaluationKit, 2013).

## Software Development Models

The development models are the various processes or methodologies that are being selected for the development of the project depending on the project’s aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use. There are various Software development models or methodologies. They are as follows: Waterfall Model, [V model](http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/), [Incremental model](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/), [RAD model](http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/), [Agile model](http://istqbexamcertification.com/what-is-agile-model-advantages-disadvantages-and-when-to-use-it/), [Iterative model](http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/) and [Spiral model](http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/) (GUIDE, 2013).

Basing on the model which will be employed in the development of this project, start by reviewing the stages of the waterfall model in general. The stages for waterfall model typically resemble the Object Oriented System Design and Analysis stages. On chapter three which is the Methodology chapter will illustrates the basic stages for Waterfall Model.

## Mobile and Web Technologies

It is easy to see how technologies such as mobile phones could be appropriated as learning tools given their essential role of communication. In Australia, a majority of undergraduate students report having mobile phones with cameras, and many have phones that are web-enabled. This suggests that mobile technologies could be utilized for learning activities that not only ask students to access material but also involve capturing and sharing information online. In addition, mobile technologies are said to offer particular advantages for facilitating learning that occurs across contexts, given their affordances as personal and portable technologies that can be carried about and accessed anytime, anywhere. For this reason mobile technologies are well placed to help learners create a link between science learning in the classroom and their experience in the world around them. While digital cameras and camera-enabled mobile phones can be used to capture information, Web tools can be used to publish and share that information, facilitating collaboration and peer knowledge sharing. (Waycott, 2009).

Seeing the contribution of both Mobile and Web Technologies, it’s reasonable to have the online system for student course evaluation process supported by these two technologies so as to make the participation of students become much simple and easier in a right way. Portability of these technologies favors much, with most of students integrated to these two technologies it becomes much easier for them to participate in this evaluation process. Students uses these web apps in their daily school activities in such for reading materials which sounds as a current learning technology, besides web based app the Android Application is well known to many of the students following the invention of smartphones. Having Web based and Android Application will practically suites around student community.

* 1. **Object-Oriented System Design**

Object oriented systems development decomposes the system down into objects i.e. it examines the system in terms of the things in the system and how these things act and interrelate. So the analyst first identifies the objects that comprise the system, then they create an object model which groups the objects into classes, and describes each class in terms of its attributes (or data), methods (or functions), and relationships to other classes. (Kalinga, 2013).

Object Orientation is about viewing and modeling the world/system as a set of interacting and interrelated objects.

* + 1. **Objects**

Objects are the basic run-time entities in an object-oriented system. They may represent a person, a location, an account, a table of data or any item that the program must handle. They may also represent user-defined data such as vectors, time and lists. Program objects should be chosen such that they match closely with the real-world objects.

* + 1. **Class**

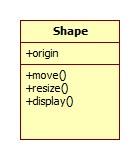
The term Class is simply a template for an object. A class is a description of a set of related objects that share the same attributes, operations. A class describes what attributes and methods will exist for all instances of the class. A class is just a description; it doesn't really exist as such until you declare an instance of the class relationships, and semantics.

* + 1. **Attributes**

An attribute is a named property of a class that describes a range of values that instances of the property may hold. A class may have any number of attributes or no attributes at all. An attribute name may be text. In practice, an attribute name is a short noun or noun phrase that represents some property of its enclosing class.

* + 1. **Operations**

An operation is the implementation of a service that can be requested from any object of the class to affect behavior. An operation is an abstraction/idea of something you can do to an object and that is shared by all objects of that class. A class may have any number of operations or no operations at all. An operation name may be text. In practice, an operation name is a short verb or verb phrase that represents some behavior of its enclosing class. Figure 2.2 illustrates the Class name, Attributes and Operations.



Operations

Attribute

Class Name

##### Figure 2.2: Class Name, Attributes and Operations

## Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, documenting and communicating the artifacts of a software-intensive system. UML is a graphical language for capturing the artifacts of software development that use object-oriented approach. (Kalinga, 2013).

### UML Diagrams

A diagram is the graphical representation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). We draw diagrams to visualize a system from different perspectives, so a diagram is a projection into a system. UML has a lot of different diagrams (models). The reason for this is that it is possible to look at a system from different viewpoints.

UML being a graphical language includes nine such diagrams (models): Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram and Deployment diagram. (Kalinga, 2013)

For this case the system analysis involved Use case diagram, Conceptual diagram, Sequence diagram as well as Design Class diagram.

## Tools and Technologies

Developing software involves the use of different kinds of technological tools which will facilitate the accomplishment of that software. There are so many tools as well as technologies available out there but the selection for particular tools depends on ones requirements and necessity of that tool. On coverage for this project the following tools were used in the development of this project.

### Bootstrap 3.0.3

Bootstrap is the sleek, intuitive and powerful front-end framework for faster and easier web development. Bootstrap will be used for the designing of the web interfaces i.e. web pages in combination with web technologies which are HTML5, CSS3, JavaScript and the jQuery.

### Android Developer Tools (ADT)

The Android Developer Tools (ADT) plugin for Eclipse provides a professional-grade development environment for building Android apps. It's a full Java IDE with advanced features to help you build, test, debug, and package your Android apps. Native Java programming language will be employed in here so as to have the Student Course Evaluation Android app.

### PHP Code Igniter Framework

Code Igniter is an open source PHP Framework, cool toolkit for building web, simple and elegant framework to create web applications. Code Igniter will be used in the development of the back-end application as well as allow interaction between the back-end i.e. database and the front-end side i.e. User Interface (UI) of the student course evaluation system.

### Star UML

Star UML is an open source software modeling tool used to develop fast, flexible, extensible, feature full and freely available UML platform running on windows platform. All UML diagrams for both analysis stage and the design stage will be drawn with star UML.

### XAMPP

XAMPP is the free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database and interpreters for scripts written in the PHP and Perl programming languages. For this particular project Apache will serves as the local host server for testing our system locally, MySQL database will serve as the back-end for storing and retrieving data. PHP will facilitate connection of the front-end and the back-end applications.

### Adobe Photoshop CS6

Adobe Photoshop is a graphics editing program developed and published by Adobe System. This tool will be used for the designing of all PNG images as well as the web interface banners and the XML background images for the Android Application.

### Notepad++ & phpDesigner8

A PHP designer is a powerful visual editing environment for PHP coder’s .With PHP Designer it’s easy to navigate to specific PHP Class definition, PHP function or PHP variable. With Notepad++ these are the source code editors which will be used in writing and editing the scripting languages.

### Operating Systems

Both windows and Linux environments will be used. Windows 8 Pro will be the one to work with for the case of windows environment while Ubuntu 13.10 (Saucy-Salamander) will be the one to work with under the Linux environments.

### MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. It enables the creation of models for database schema graphically.

## PHP CodeIgniter Framework

Code Igniter is an open-source PHP web application framework, which can be used for rapid development without the overhead of having the consoling reusable components, providing logical structure, reusable interface for libraries to perform common tasks. The key features of CodeIgniter are Small footprint (lightweight), Fast Performance and have very little configuration. (Peck, 2014).

The key factors which make CodeIgniter chosen over the others frameworks are as follows:

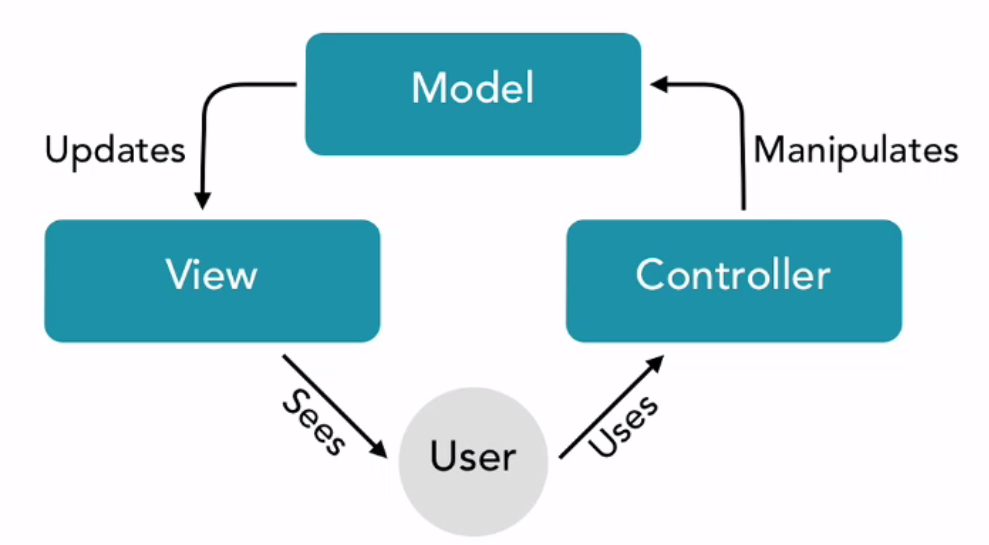
* CodeIgniter is free i.e. Open-source software.
* CodeIgniter is light weighted.
* CodeIgniter have fast performance.
* CodeIgniter allows reusability and modularity i.e. avoiding copy and paste; components can be reused on multiple projects.
* CodeIgniter uses (loosely) Model-View-Controller (MVC) for organizing code and logic.
* CodeIgniter is Extensible.
* CodeIgniter is user-friendly.
* CodeIgniter generates clean URLs.

### CodeIgniter as MVC Framework

CodeIgniter MVC development pattern is a software design pattern that separates the representation of information from user interaction. This explicitly separation of components allows the creation of code reusability and logical separation program responsibilities. MVC keeps the code organized. (Peck, 2014). Figure 2.3 illustrates the MVC process loop.

MVC has three components:

* Model: is the representation of data structure, business roles and functions. This is where the Create, Read, Update & Delete functionality is defined i.e. CRUD
* View: is the representation of information that presented to the user. This is where all HTML rendering takes place.
* Controller: serves as a kind of intermediary between a model and a view. It sends a command to either a model or a view.



##### Figure .3: Model-View-Controller process loop

## Bootstrap Framework

Bootstrap a front-end framework which is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Bootstrap supports responsive design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone). It is compatible with the latest versions of all major browsers. It gracefully degrades when used on older browsers such as Internet Explorer 8. (Kramer, 2014).

There are key factors which makes bootstrap to be selected over the entire front-end frameworks. Mentioning those key factors;

* Bootstrap is a framework that provides an easy-to-use 12-column grid system for a 940 pixel wide container as well as a fluid layout grid that adjusts to the size of a browser.
* Included in the framework are customizable, well documented features—such as dropdown menus, buttons, tabs, breadcrumbs, modals, tooltips, and carousels—that can be implemented without any handwritten JavaScript.

### Bootstrap file structure

Bootstrap framework is just a folder, inside that folder; there is a CSS, FONTS, and JS folder. (Kramer, 2014). Figure 2.4 shows a Bootstrap framework file structure.

* CSS folder: contains the style sheet files. There are two versions of these style sheet files, a .min version and a regular version. A regular version is just human readable file while a .min version is machine readable and is used for small devices design.
* FONT’s folder: contains the fonts that will be used in designing of the pages.
* JS folder: contains the JavaScript files, also in here there are two versions as in CSS folder the regular version and a .min version.



##### Figure 2.4: Bootstrap Framework File Structure

# CHAPTER THREE: METHODOLOGY

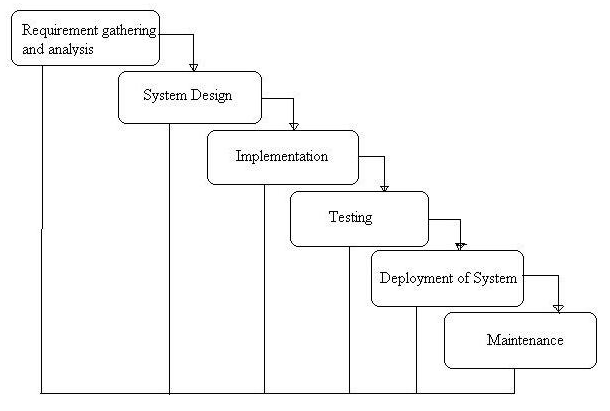
## Introduction

In any software development, there are must be different approaches or methodologies which will be used in the designing and implementation of that software. As being already stated earlier in Chapter Two, on reviewing the various software development models, Waterfall Model has been employed for this project for the analysis and the designing of the system. The Waterfall Model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Requirement Capture and Specifications, System Analysis, System Design, Implementation, System Testing and Maintenance.

Besides the use of waterfall model, various technologies and tools which were employed for the implementation of the whole project have been looked at.

## The Waterfall Model

Figure 3.1 illustrates diagrammatically the stages of the waterfall software development model.



##### Figure 3.1: Waterfall Model Stages

* + 1. **Requirement Capture and Specifications**

Requirement capture is one of the key factors which lead to project implementation and accomplishment. With the help of the requirement from the case study area (which is Quality Assurance Bureau for this project) will make clearly the important things that the project must be composed of. Considering the end user requirements specifications and put forward the scenarios on how the system will be applied, which user features and functions are to be to developed, which platform to be applied and which devices specifically are going to be supported.

* + 1. **System Analysis**

As in any system development approach, in this phase the user requirements will be analyzed and put them as a function system. Based on what the system is required to do, a model for the desired application system will be prepared with the help of UML (Unified Modeling Language).

* + 1. **System Design**

This is the next development approach which decides the overall architecture of the desired system. The system will be divided into smaller sub-systems (modules) interacting together. In an Object Oriented language the system will decompose down to several interacting objects.

* + 1. **Object Design**

In this phase, the details from system analysis and system design will be examined into real world scenarios where the objects identified in the system design phase will be now being designed. This phase will go further up to specifically designing of the classes, objects and partial implementation to some of the modules while exploring the Object Oriented Concepts like abstraction, inheritance, and polymorphism.

* + 1. **Implementation**

This is the phase where the classes and class objects are translated to the actual code depending on the preferred programming language.

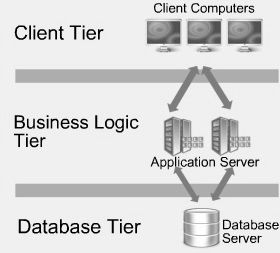
* + 1. **Evaluation and Testing**

There will be two types of testing, the functionality testing and performance testing based on the evaluation from user requirements. Unless the functionality and performance requirements have been full filled, testing and re testing will not cease.

* 1. **System Architecture**

System architecture refers to the way in which desired functionality is met by hardware and software components as well as how these components relate to each other and the intended users of the system. (Golden, 2013).There are several types of system architecture each having its advantages and disadvantages. For the case of this project 3-Tier architecture has been employed.

3-Tier Architecture consists of three tiers (layers) as shown in Figure 2.3. The first tier is min-client tier which consists of end-user interface to the system. On web-based system this is the web browser. Another tier is the business logic and data processing logic runs the application (Application Server). And the last tier is the Database Management System (DBMS) tier, is the one that runs the DBMS and holds the database (Database Server). (Bagile, 2013).



##### Figure 3.2: The 3-Tier Architecture

## Data Collection Methods

Within each general research approach, one or many data collection techniques may be used. Typically, a researcher will decide for one (or multiple) data collection techniques while considering its overall appropriateness to the research, along with other practical factors, such as: expected quality of the collected data, estimated costs, predicted nonresponse rates, expected level of measure errors, and length of the data collection period. It is of course possible that a given research question may not be satisfactorily studied because specific data collection techniques do not exist to collect the data needed to answer such a question. The most popular data collection techniques include: surveys, secondary data sources or archival data, objective measures or tests, and interviews. (Data Collection Techniques, 2013)

Surveys which may be regarded as the primary data collection technique involves basically the face to face interview which one is going to ask the direct questions and obtains the answers at that instant. Regarding to this particular project, Interview was selected as the data collection technique and the primary one. Since Interview is a good approach to gather in-depth attitudes, beliefs, and anecdotal data from individual patrons. Personal contact with participants might elicit richer and more detailed responses. Interview also provides an excellent opportunity to probe and explore questions and finally participants do not need to be able to read and write to respond.

Secondary data collection techniques were also used i.e. collecting some reports as secondary data from the QAB official these are included in the Appendix Chapter. Also searches made from different journals, conferences and websites as well, different articles concerning the course evaluation process in general and the related works as well.

# CHAPTER FOUR: REQUIREMENTS CAPTURE AND ANALYSIS

## 4.1 Requirements Capture

Requirements are a description or statement of a function, feature or condition that a user seeks to have implemented in a desired system. (Kalinga, 2013). In accordance to this project we have analyzed the requirements based on the data collected from the case study. The kind of the data collected led us to analyze the so called functional requirements of the system. Data were also collected in terms of face to face interview which involved asking series of questions. Partly some of the questions which were being asked to the case study (client) were:

1. Quick view on how the evaluation process is being conducted in the University of Dar es Salaam.
2. How QAB officials compile, process and report the evaluation results?
3. Are there any technical (computerized) tools that are used to facilitate the evaluation process?
4. Does evaluation process performed in terms of the academic units?
5. What are the challenges and limitations that are faced during the process?

Having performing this face-to-face interview made us to confidently analyze the system requirements in general.

## Requirement Analysis

In analyzing the system we considered the requirements for which a user seek to be implemented as the system functions. These requirements were later analyzed as the functions that the system will provide. Two broad categories of requirements are being explained as follows.

### Functional Requirements

They relate directly to the functioning of the system. These are the aspects of the system the client is most likely to recognize. They describe the interaction between a system and its environment. Describe how a system should behave under certain stimuli. The system’s function should be categorized in order to priorities them or to avoid from missing them.

Categories included:

—Evident functions should be performed, and user should be aware that is performed.

—Hidden functions should be performed, but not visible to users. Hidden functions are often incorrectly missed during the requirements gathering process.

### Non-functional Requirements

These are constraints/restrictions that imposed on the system. They define how a system is supposed to behave and they are often called qualities of the system. (Kalinga, 2013). Tables 4.1 Shows the System Functional Requirements.

Table 4.1: System Functional Requirements

|  |  |  |
| --- | --- | --- |
| Ref #: | System Functions | Category |
| R1.1 | System should allow student to perform registration by using his/her registration number, degree program and the year of study | Evident |
| R1.2 | System should allow student to login with registration number and password in order to use the system | Evident |
| R1.3 | System should allow admin to login with username and password in order to use the system | Evident |
| R1.4 | System should validate a student and administrator accounts | Hidden |
| R1.5 | System should allow student to view and select the courses that he/she studies with respect to degree program and year of study | Evident |
| R1.6 | System should allow student to fill in the online evaluation form for his/her respective course instructor | Evident |
| R1.7 | System should allow student to submit the already filled evaluation forms to the server | Hidden |
| R1.8 | System should allow student to logout | Evident |
| R1.9 | System should allow admin to perform setup (add/edit Departments, add/edit courses, add/edit colleges/institutes, add/edit courses) | Evident |
| R2.0 | System should allow admin to add course Instructor details | Evident |
| R2.1 | System should allow admin to edit course Instructor details | Evident |
| R2.2 | System should allow admin to delete course Instructor details | Evident |
| R2.3 | System should allow the administrator to logout | Evident |
| R2.4 | System should generate statistical and periodical report | Hidden |

The System Constraint or Non-functional requirements are described in Table 4.2

Table 4.2 : System Non-functional Requirements

|  |  |
| --- | --- |
| **Attribute** | **Constraints** |
| Usability | The interface of the system must be simple and user-friendly. The user would not face any difficulties in viewing and fillings the forms |
| Scalability | The system should be scalable without any need of rewriting the entire system. |
| Compatibility | System is compatible with Chrome browser, Firefox browser and internet explorer 8. The mobile application will run under android platform only. |
| Data Integrity | Any input to the form must be validated to ensure the correctness of data entered. |
| Security | There is no guarantee for the security of the physical device. Security of the data is guaranteed. |
| Performance | System will work efficiently on a standard computer with at least 1GB RAM Memory and a normal Pentium processor.  The mobile application will work effectively on android 4.0 version up to 4.3 although the application can operate on android 2.3 versions but not efficiently. |

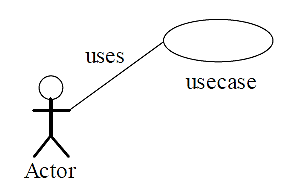
## Use Case Modeling

A use case is a high-level description of a major user requirement. It represents the functionality of the system, is a description of the system’s behavior from a user’s viewpoint. Constitute a complete interaction with the system initiated by a user or another system. (Kalinga, 2013).

A Use Case diagram shows a set of use cases and actors (a special kind of class) and their relationships. This diagram is a valuable aid during analysis. Developing Use Cases helps to understand requirements.

### Basic Use Case Notation

* The Actor represents a user of the system, or any external system that interacts with the system.
* The Use case represents a piece of functionality that is important to the user. Figure 4.1 represents a basic use case notation.



##### Figure 4.1: Basic Use Case Notation

* + 1. **Use Cases Identification**

There mainly two ways of identifying the use cases relating to the system. Use cases can be identified on bases of actor-based and on event-based. (Kalinga, 2013).

* Actor-based: Gives identification of all the actors related to the system i.e. Finding and specifying all the actors by looking at which users will use the system and which other systems must interact with it. For each actor, identify the processes they initiate or participate in by looking at how the actor communicate/interact with (or use) the system to do his work.
* Event-based: In this case, identification of the external events that a system must respond to that is basically relating the events to actors and the use cases is performed.

Table 4.3 shows the Identified Actors and the Events that they initiate for this particular system.

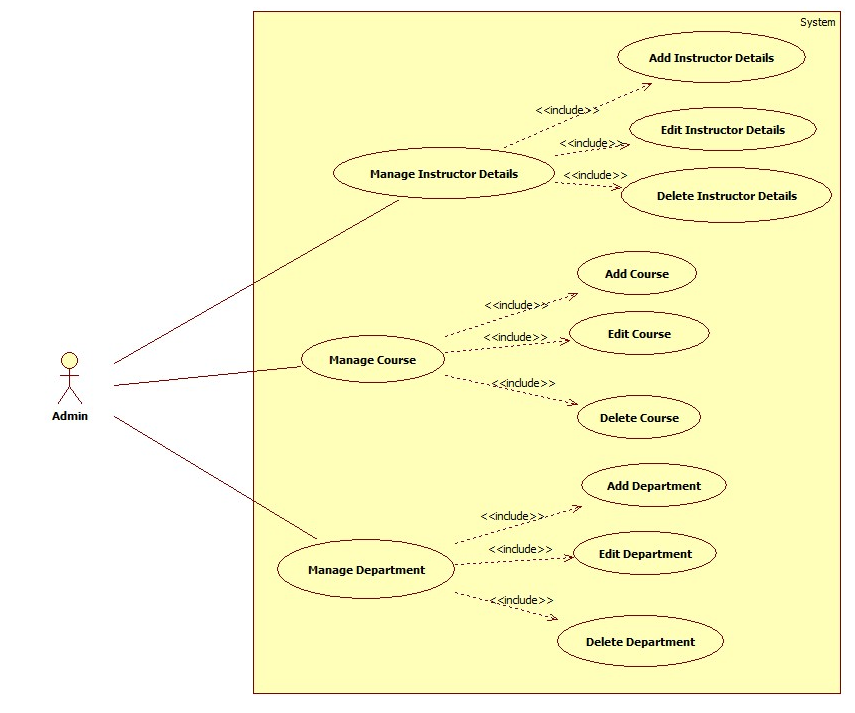
Table 4.3: System Actors and Initiated Events

|  |  |
| --- | --- |
| **Actor** | **Event/Process to initiate** |
| Student | Login, Perform Registration, View and Select course, Perform Evaluation, Submit Evaluation form |
| Admin | Create Account, Add/Edit/Delete Course, Add/Edit/Delete Departments, Add/Edit/Delete Instructors details |
| System | Validate Account, Generate Report |

After identifying the use cases and sees different actors for this system and processes or events that they initiate; now what follow are the use case diagrams for each of the actors of the system. Figure 4.2 and 4.3 shows the use case diagrams for all actors of the system and the process that they initiate.



##### Figure 4.2: Use Case Diagram for Student



##### Figure 4.3: Use Case Diagram for Admin

### Use Case Description

After creating a use case, then the creation of a high-level use case to obtain some understanding of the overall process is the one which follow, and then expand it by adding to it with more details. Each Use Case contains a full set of textual details about the interactions and scenarios contained within it.

Use case descriptions are written following the template which is being illustrated in the Appendix B. The template was later used to write the use case descriptions for the system. Table 4.4 shows two samples of the use cases descriptions for this particular system.

Table 4.4: Perform Evaluation Use Case

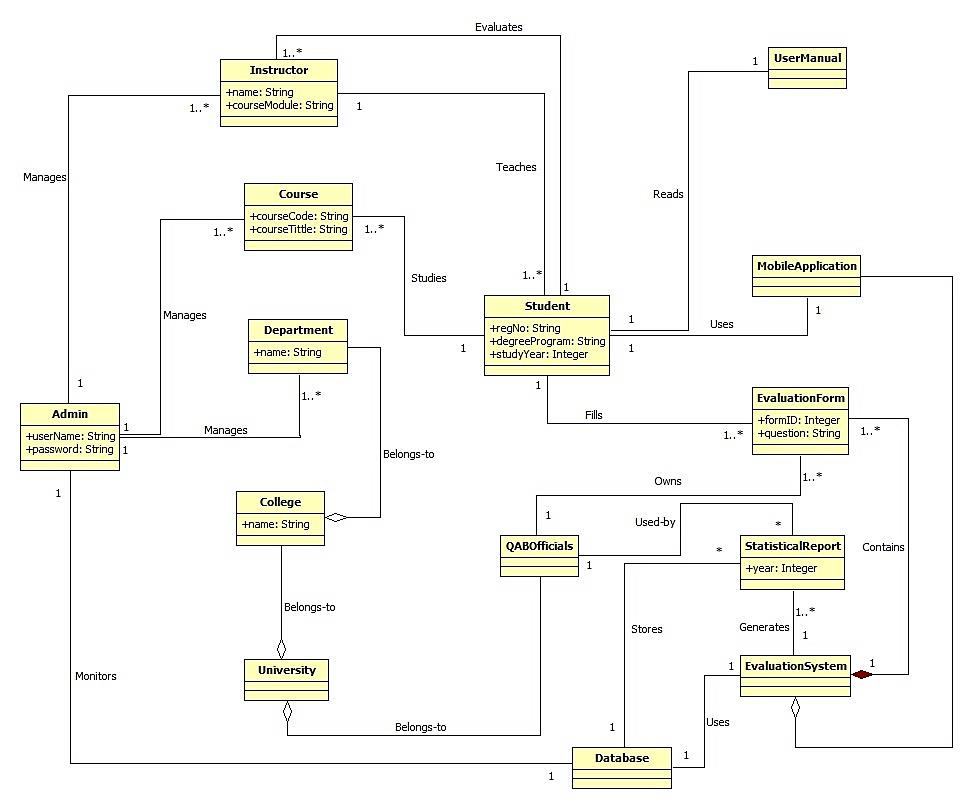
|  |  |
| --- | --- |
| **Use case:** | Perform Evaluation |
| **Actors:** | Student |
| **Short Description:** | Allow the student to perform evaluation through mobile app or website.  Allow student to evaluate his/her course Instructor. |
| **Cross Reference:** | R1.6 |
| **Pre-Conditions:** | The system must be on and student must login |
| **Post-Conditions:** | Evaluation form will be filled and course Instructor will be evaluated |
| **Main Flow:** | 1. Student select one of the course from the list 2. Display the course Instructor name and the evaluation form 3. Student fill in the required field 4. Student initiates the save action |
| **Alternate Flow(s):** | - |
| **Exception Flow(s):** | Line 4: If required fields are empty, indicates error |

Table 4.5: Submit Evaluation Form Use Case

|  |  |
| --- | --- |
| **Use case:** | Submit Evaluation Form |
| **Actors:** | Student |
| **Short Description:** | Allow the student to submit the already filled in evaluation forms |
| **Cross Reference:** | R1.7 |
| **Pre-Conditions:** | The system must be on and student must login |
| **Post-Conditions:** | The evaluation forms will be saved in the database |
| **Main Flow:** | 1. Student click the submit button 2. Successful message display |
| **Alternate Flow(s):** | - |
| **Exception Flow(s):** | Line 1: If required fields are empty, indicates error |

* + 1. **Conceptual Model Diagram**

Conceptual diagram as shown in Figure 4.4 shows all system concepts with attributes and all sorts of relationship between them.



##### Figure 4.4: System Conceptual Model Diagram

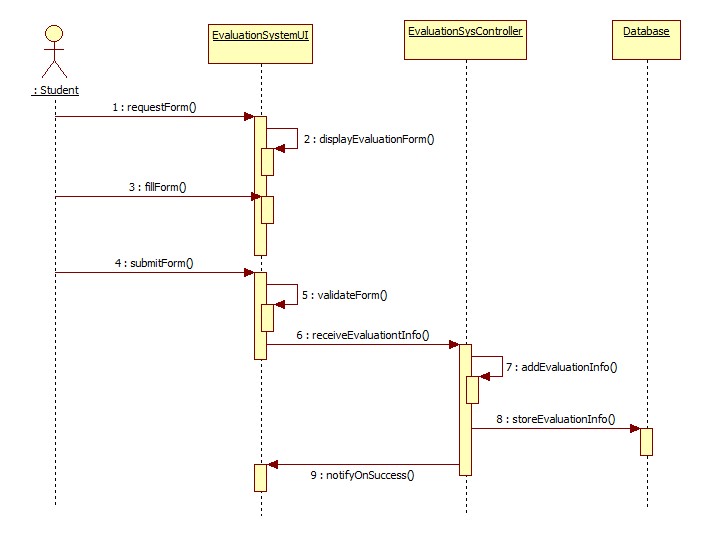
# CHAPTER FIVE: SYSTEM DESIGN

## Overview

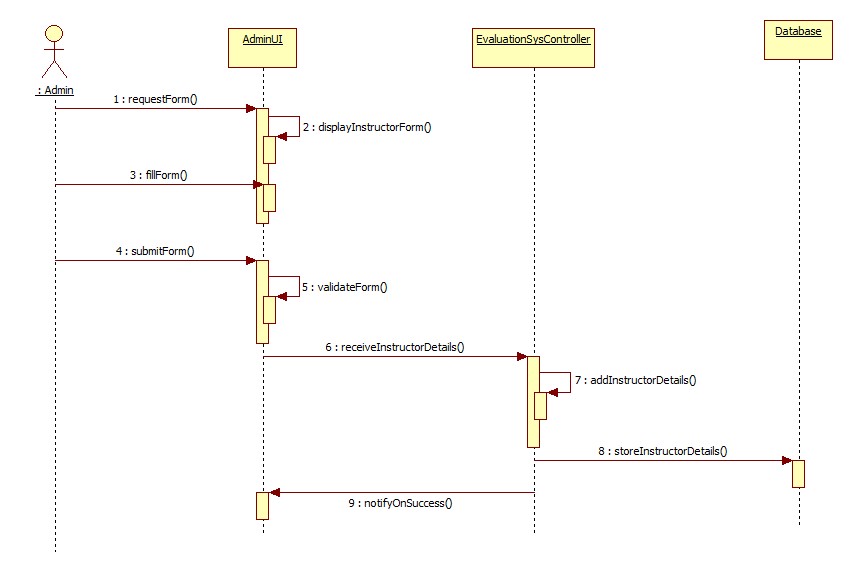
System designing involves defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. Object oriented designing using UML was used in designing the system and the design is categorized as Architectural and Detailed Designs.

## System Sequence Diagram

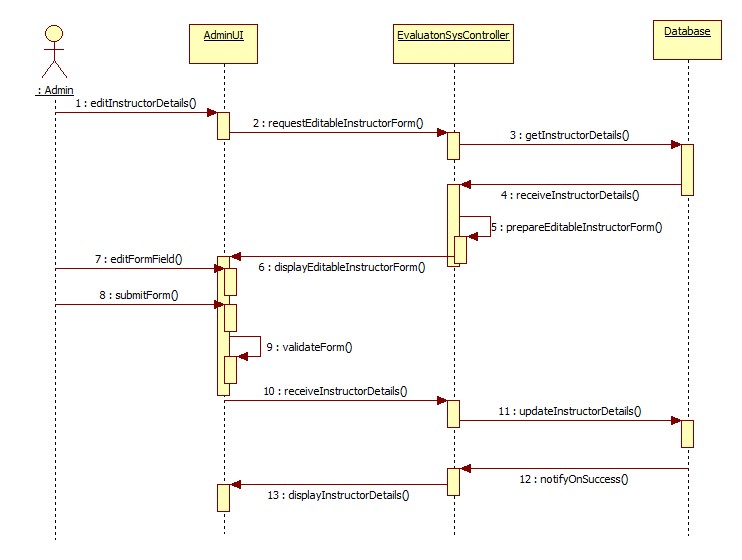
A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages i.e. focus on the order in which the messages are sent and provide a sequential map of message passing between objects over time. (Kalinga, 2013). System sequence diagrams depict the interaction between the system and any user involved with the system. They show different events initiated by user and their corresponding system response. Three of the system sequence diagrams are shown in Figures 5.1, 5.2 and 5.3



##### Figure 5.1: Sequence Diagram for Perform Evaluation Use Case



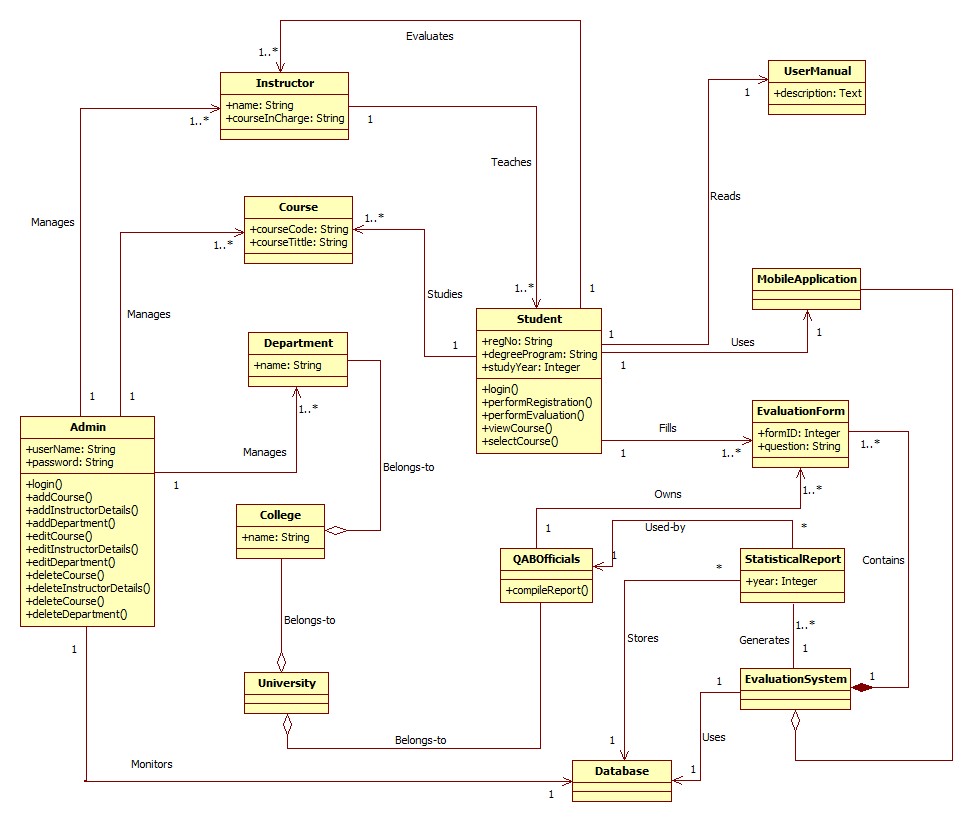
##### Figure 5.2: Sequence Diagram for Add Instructor Details Use Case



##### Figure 5.3: Sequence Diagram for Edit Instructor Details Use Case

## Design Class Diagram

A design class diagram is the one which showing all classes, associations between classes, class attributes, methods, attribute type information, navigability as well as dependencies. The design class diagram for the system is shown in Figure 5.4



##### Figure 5.4: System Design Class Diagram

* 1. **Database Design**

In the designing of the database, we mainly focus on defining all the entities i.e. tables that will be contained in the database and holds certain data. Showing the relationship that exists between these entities as well as interaction between one table and another, defining the data type that each table column holds, describe each table column. To put into consideration all these we will design the database schema, database dictionary and the entity relationship model diagram.

* + 1. **Database Schema**

A database schema of a database system is its structure described in a formal language supported by the DBMS and refers to the organization of data to create a blueprint of how a database will be constructed. The database schema for iEvaluate\_db shows the table definitions with all of the columns. Tables for the iEvaluate\_db are as follows.

Student (**RegNo**, ProgramID, StudyYear)

Program (**ProgramID**, ProgramName, StudyYears, DepartmentID)

Instructor (**InstructorID**, **CourseCode**, Title, FirstName, SurName, OtherNames, Gender,DepartmentID**,** Status)

Course (**CourseCode**, CourseTitle, Units, ClassSize, Semester, ProgramID, StudyYear, VenueID)

Department (**DepartmentID**, DepartmentName, CollegeID)

College (**CollegeID**, CollegeName)

Evaluation (**EvaluationID**, **CourseCode**, **InstructorID**, ProgramID, Semester, DepartmentID, CollegeID, StudyYear, RegNo)

InstructorPart (**InstructorPartID**, **EvaluationID**, B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11)

CoursePart (**CoursePartID**, **EvaluationID**, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10)

LEFPart (**LEFPartID**, **EvaluationID,** D1, D2, D3, D4, D5, D6, D7, D8)

Users (**UserID**, Username, Password)

Venue (**VenueID**, CollegeID)

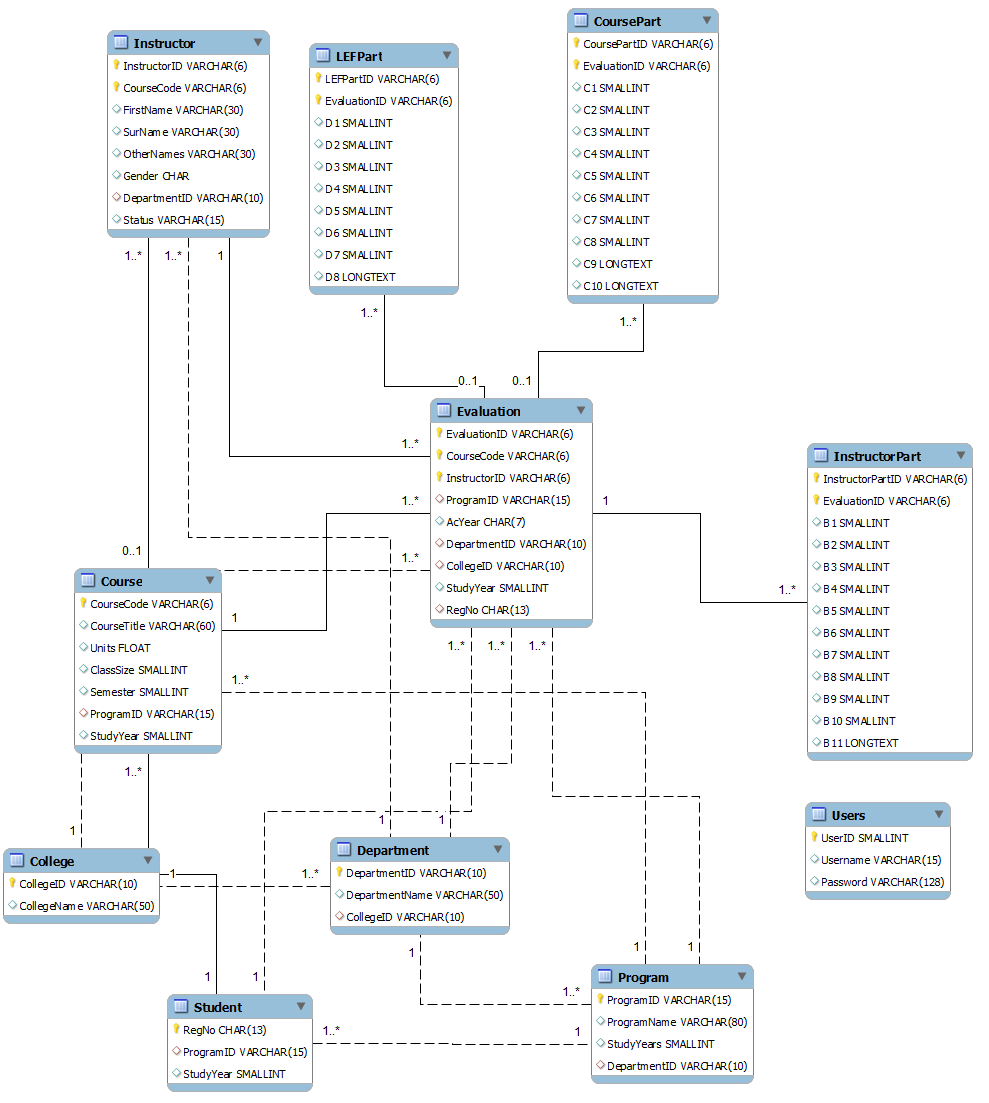
Evaluation\_Setup (**SetupID**, StartDate, EndDate, Semester, AcademicYear)

**NOTE:** Word(s) with bolded font represents the **PRIMARY KEY** for that table i.e. Unique Identification.

Table columns named B1, B2…, C1, C2…, D1, and D2…., are the initials in the sample evaluation form for each part. For example B1 represents “Instructor’s preparedness on the subject matter” which is in “Part B: The Instructor” of the evaluation form. For more details check the main Evaluation form on the Appendix chapter.

## Entity Relationship Diagram

Entity Relationship Diagram (ERD) is a specialized graphic diagram that shows relationship between entities in a database. Figure 5.5 shows the ERD for iEvaluate System.



##### Figure 5.5: System Entity Relationship Diagram

# CHAPTER SIX: SYSTEM IMPLEMENTATION

## 6.1 Overview

Mainly the project covers Student Course Evaluation through Web based and Android Application, the system has been called simply iEvaluate which basically meaning and reflects that “I as a student, I evaluate my courses as well as my Instructors”. So iEvaluate Course Evaluation System have two main parts which are discussed in this chapter.

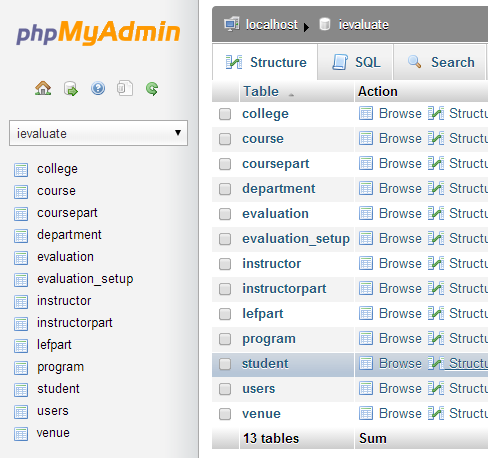
The first part is iEvaluate Web based which has been implemented using Code Igniter framework and basically uses PHP language as a server-side scripting language, Bootstrap framework for Web Interfaces layout designing which contains HTML5 as a markup language for describing web pages, JavaScript as a scripting language and CSS for styling the web pages. MySQL DBMS was used as a database system. The choice of all these tools and languages was supported by their free-use and simplicity they offer throughout the development process.

The second part is iEvaluate Android Application based which has been implemented using Native Java as the native approach of developing android applications, Java programming language was used for implementing all the logics while XML was used for layout designs and JSON was used for creating API to interface with server-side.

## 6.2 PART ONE: iEvaluate Web Based Application

### 6.2.1 Creation of iEvaluate Database in a Server

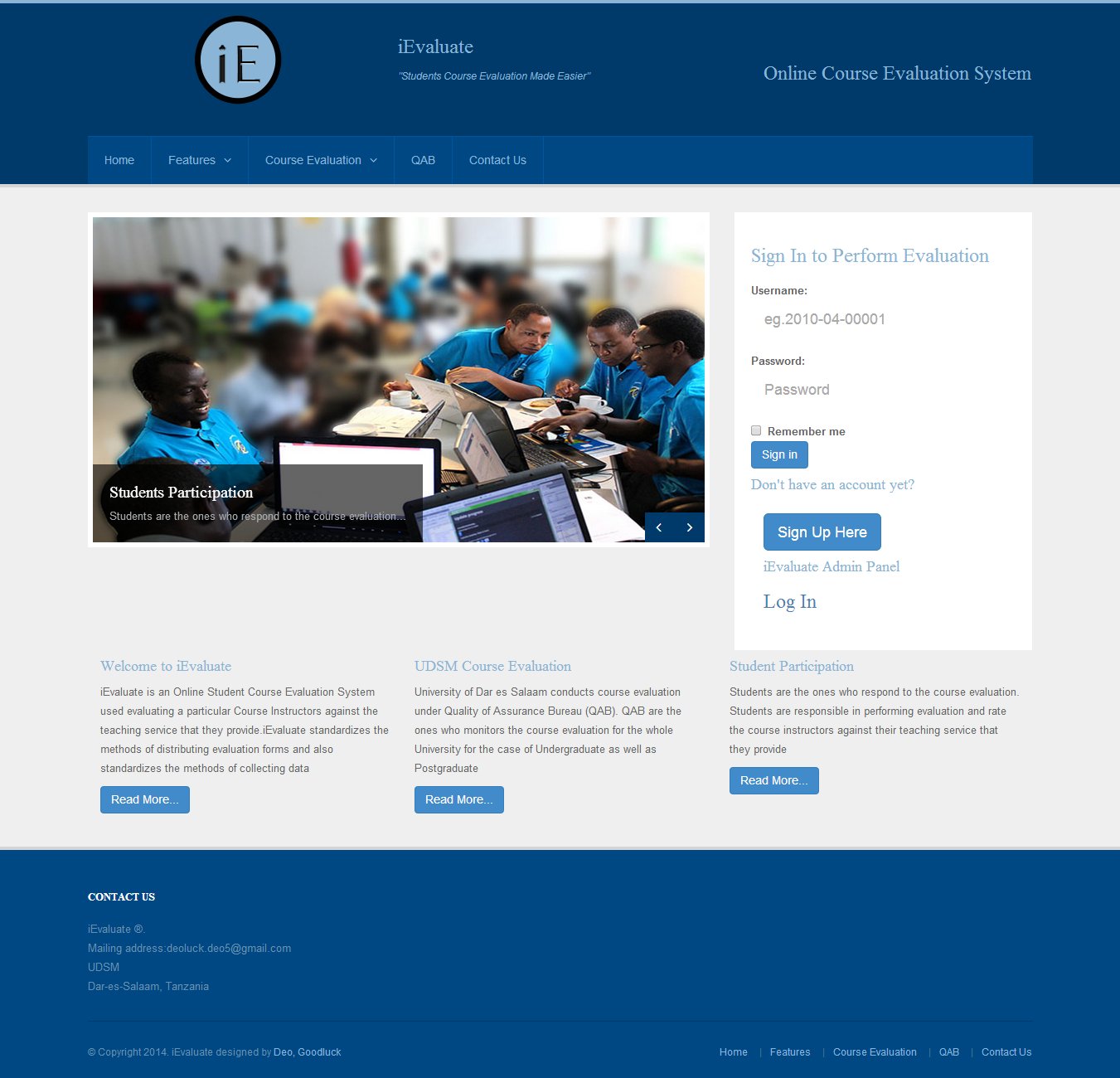
Using the ERD given in Figure 5.5 and the database schema as already being discussed in section 5.4.1, the iEvaluate database was created locally in Personal Computer (PC) after installing XAMPP software package that contains an Apache server and MySQL DBMS. Using database management functionalities provided in phpMyAdmin by XAMPP iEvaluate database was created; security features on the database and server were also added. Figure 6.1 shows the created database called ‘ievaluate together with all the thirteen tables it contains i.e. college, course, coursepart, department, evaluation, instructor, instructorpart, lefpart, program, student, users, venue and evaluation\_setup.



##### Figure 6.1: iEvaluate Database View from XAMPP phpMyAdmin

### 6.2.2 User Interface Design and Implementation

The user interfaces of the iEvaluate Web Based Application were worked out using Bootstrap Framework which is the responsive design for the front-end system as well as for Admin Panel (back-end). All layouts and styling were done using different libraries found within Bootstrap Framework. Different front-end pages were implemented, Figure 6.2 shows the front-end Homepage of iEvaluate Web Based Application. Mainly the kind of users who are accessing the front-end pages are students so the design of the pages were so impressive in order to make them look fancy to the user so as user will keep in touch with the system all the time.



##### Figure 6.2: iEvaluate Web Based Application Front-end Homepage

### 6.2.3 iEvaluate Front-end Modules Implementation

iEvaluate Web based consists of several modules; these modules provide different functionalities as stated in the system requirements in table 4.1. The modules also are being divided in such a way that there are those which are performed on the front-end side and others on the administrator side. Get to mention these modules, there is Login module which covers for both front-end and the admin panel side i.e. both users of the system must login so as to carry any functionality within the system. Front-end modules feature are going to be explained first then followed by the administrator part modules features.

#### 6.2.3.1 Login Module

With Code Igniter Framework it enables to write the system back-end functionalities, login module was successfully implemented. Real scenario is, when a user wants to login into the system he/she is being provided with the login form which the required information are to be provided by writing them through the text inputs provided. The information will pass through validation process which the function written in Code Igniter will handle that process, when the information is valid, a user will be allowed to login otherwise a user will be asked to provide the valid information. Figure 6.3 illustrates the login module in general.

##### Figure 6.3: Login Module Illustration with Input Fields Required and if Invalid Login Credentials are submitted

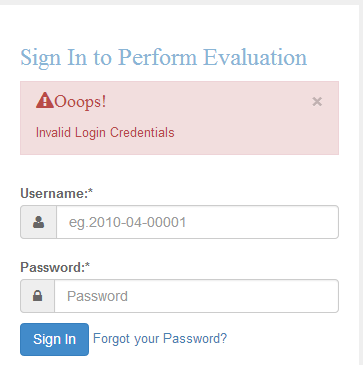


Figure 6.3 (b): Login Module, Input Fields with Invalid Login Credentials

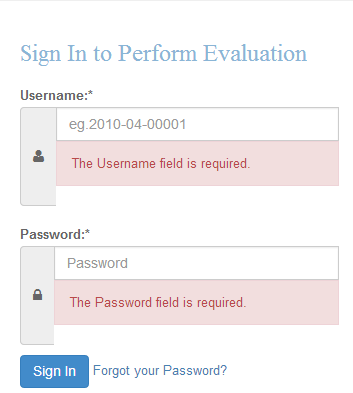
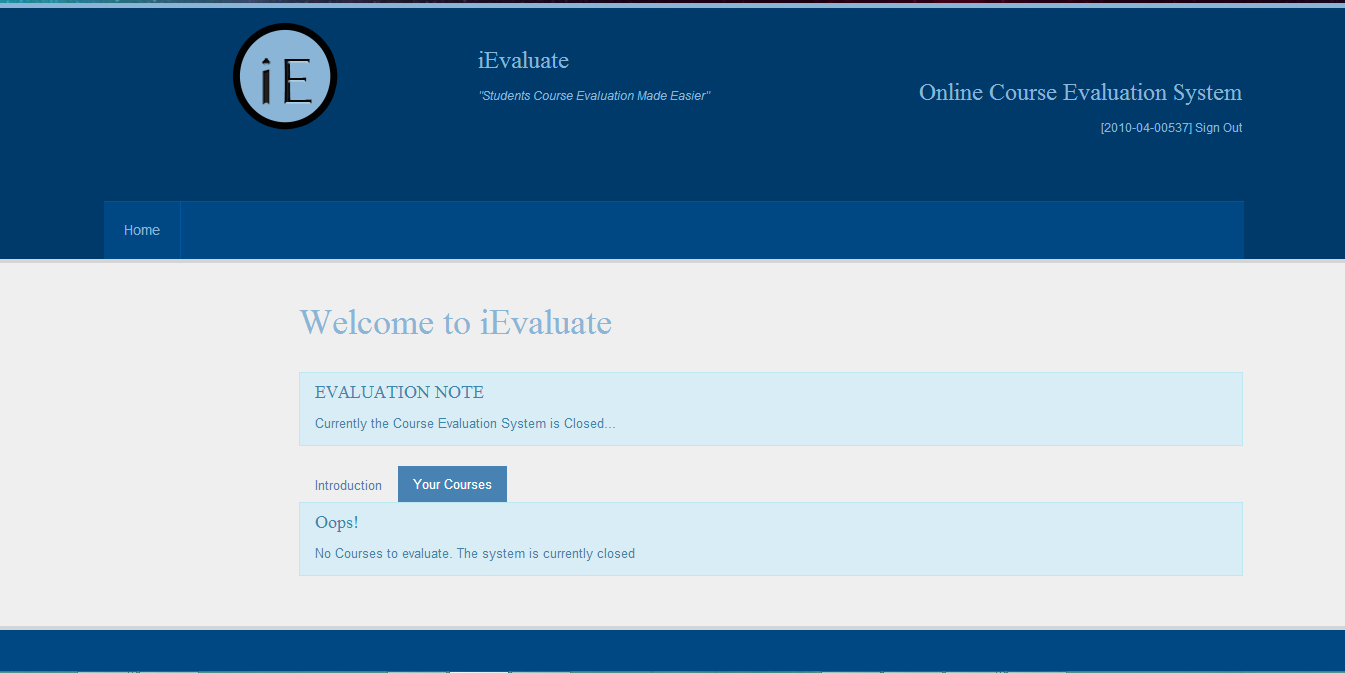


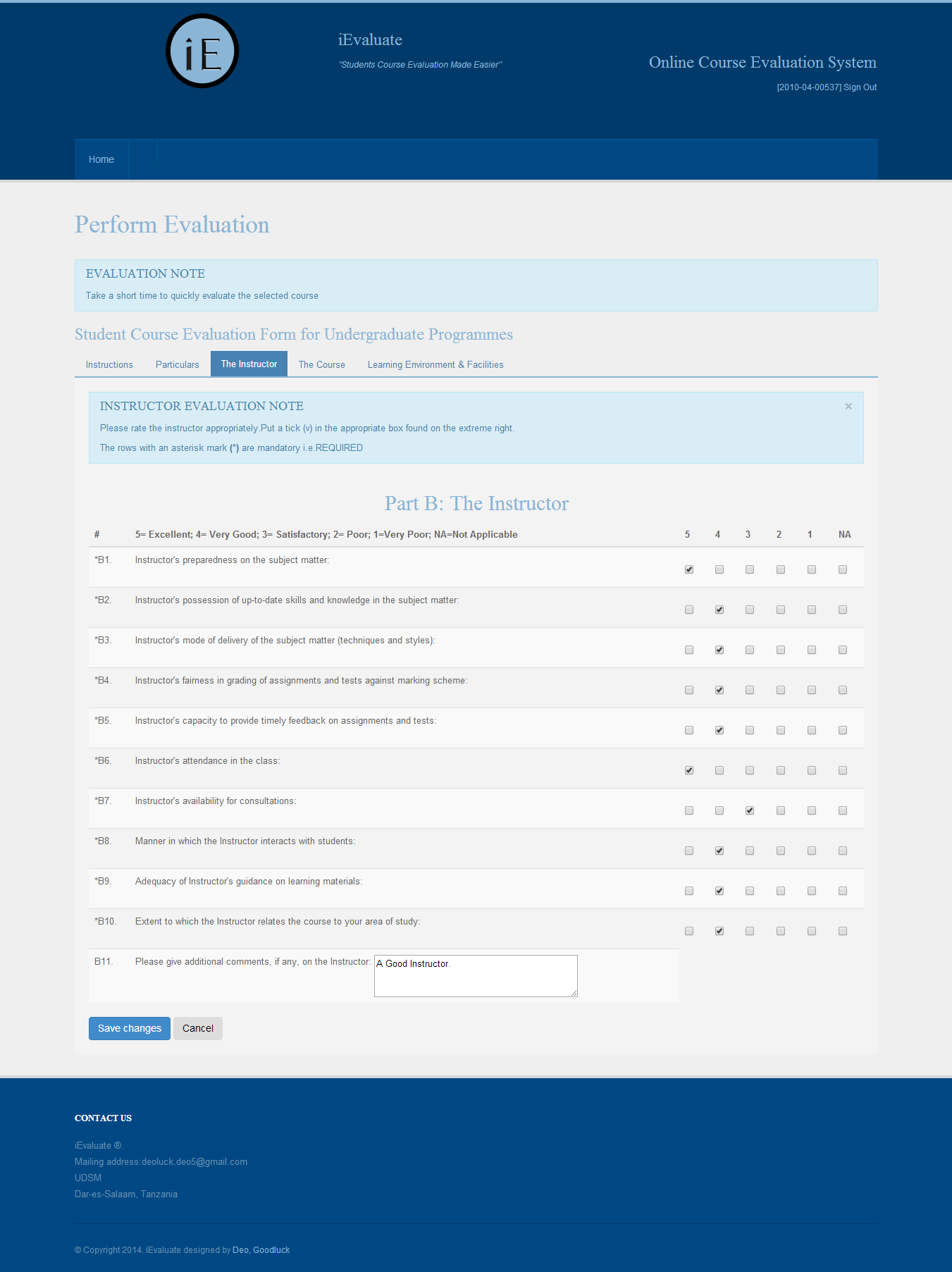
Figure 6.3 (a): Login Module, Check if Input Fields are Empty

#### 6.2.3.2 Perform Evaluation Module

This is the crucial part of the project and it is the point where all the specific objectives come from which make up the main objective to come into existence. Student Course Evaluation as already explained in the earlier chapters that it is conducted once per semester, but now there is an online system which makes it to be available at a predefined set time. Does it mean that students will be able to evaluate their courses all the time even if they are in holidays? Logically does not sound. So with the Perform Evaluation Module a student will be able to evaluate his/her respective course as well as the course instructor in charge. With timeframe evaluation feature which has been in cooperated in the system limits a student not to perform evaluation all the time, which means that a particular student will be able to perform evaluation by the time the System is open for evaluation. Figure 6.4 illustrates the window for Timeframe evaluation feature of the system. The evaluation form is the one which a student has to fill it and submit the result online; Figure 6.5 shows the evaluation form. With these features it makes the process of evaluation easier and convenient.



##### Figure 6.4: Timeframe Evaluation Feature.



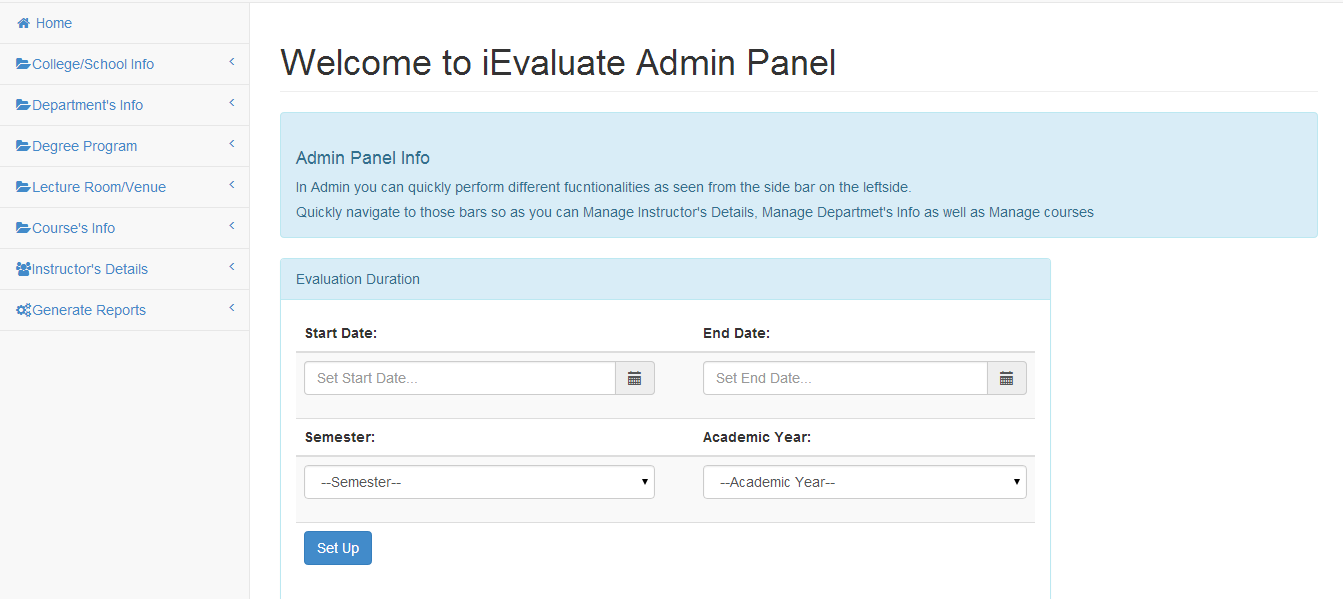
##### Figure 6.5: Perform Evaluation Module: The Instructor Part

### 6.2.4 iEvaluate Administrator Modules Implementation

In most of dynamic system there is a back-end part for interacting with the server so as to perform some management functionalities. For the case of iEvaluate there is Admin Panel which is there purposely for management of various functionalities which the system provides. These include Evaluation Setup, Managing Instructor’s Details i.e. Add, Edit and Delete Instructor’s Details, Managing Course i.e. Add, Edit and Delete Course, Managing Degree Programs, Managing Departments, Managing College/School, Managing Lecture Venues and Report Generation. All of these modules have been implemented and they are all available after one having system administrator credentials and login into the system and perform what is supposed to be performed. The login module is the same as the one illustrated earlier in the front-end part. For Illustration purposes Evaluation Setup, Managing Instructor’s Details and Report Generation are the one to be used here.

#### 6.2.4.1 Admin Evaluation Setup Module

Following the basics of conducting course evaluation, this module was implemented purposely aiming at monitoring each and every course evaluation that will be carried out. On the admin evaluation setup module, the system administrator is responsible for providing the evaluation window i.e. from which date the course evaluation will start and which date will end, not only that but also the semester and the academic year which the particular course evaluation will be conducted. Figure 6.6 shows admin evaluation setup form which the system administrator has to fill it so as to setup particular evaluation.



##### Figure 6.6: Admin Evaluation Setup Module

#### 6.2.4.2 Admin Add Instructor’s Details

This functionality enables the Administrator to add a new instructor’s details. The Administrator is provided with the Add Instructor’s Details Form, which he/she must fill and submit for successful addition of instructor’s details. Data Validation on the Add Instructor’s Details Form is also implemented. If an administrator wants to add multiple instructor at a time there is feature of importing a Comma Separated Values (CSV) file which has to be already created and contains more than one instructor details i.e. bulky data. Figure 6.7 shows Add Instructor’s Details Form with Import Multiple Instructors feature.

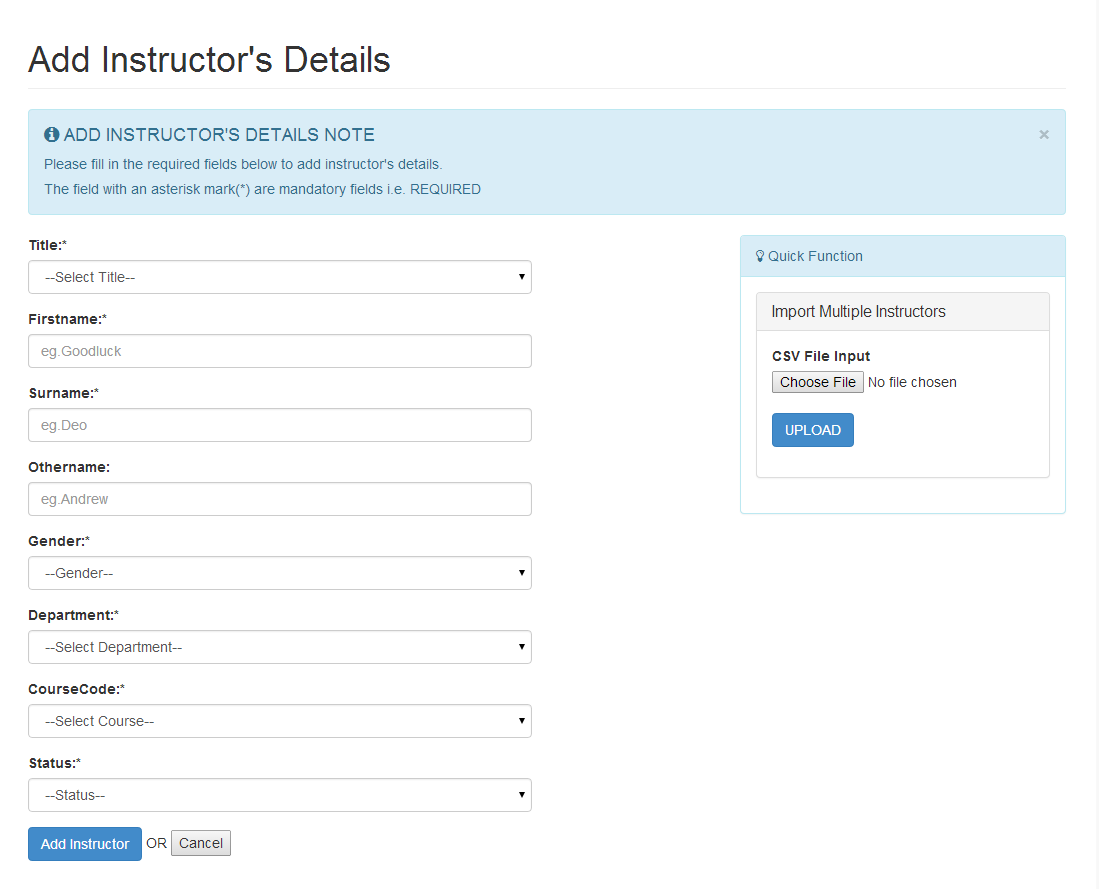
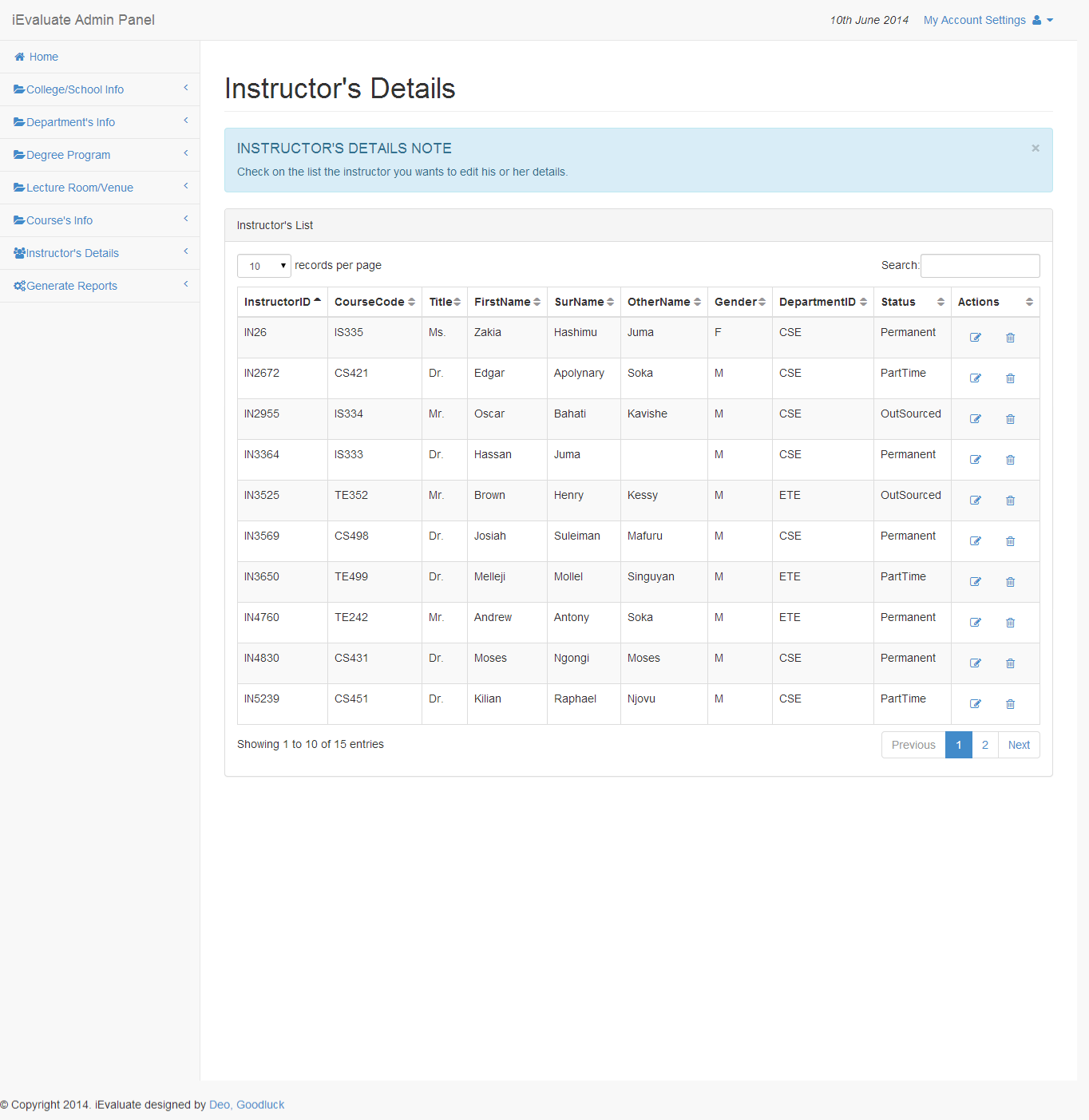


Figure 6.7: Add Instructor’s Details Form with Import Multiple Instructors Feature.

#### 6.2.4.3 Instructor’s Details

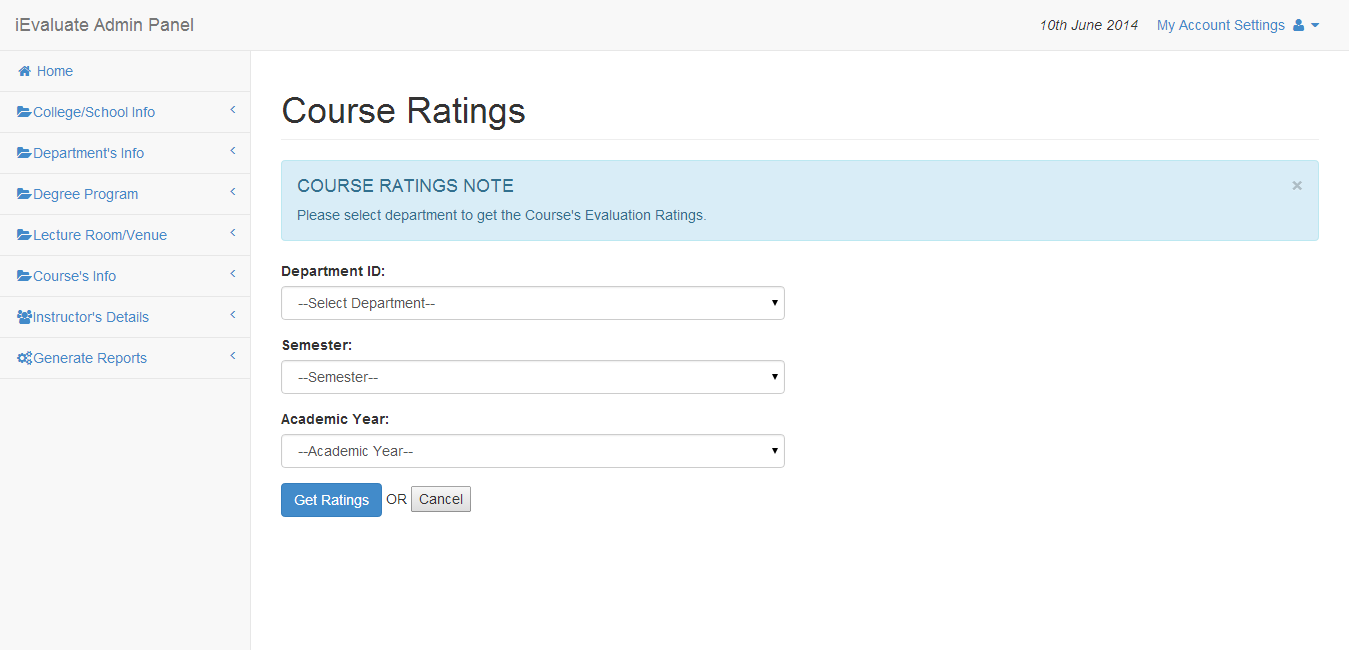
This module enables Administrator to view the list of all registered instructors within the system; Instructor’s details are being displayed showing information such as Instructor ID, Couse Code, First name, Surname, Other names if any, gender, department ID and the status. From there Admin can perform the actions such as Edit Instructor’s Details or Delete Instructor’s Details as seen from the view. Besides those actions, Admin can also search Instructor’s details using different key words such as First name, Surname, Course Code, Department ID and Status. Figure 6.8 shows how the Instructor’s Details list as viewed from the system.



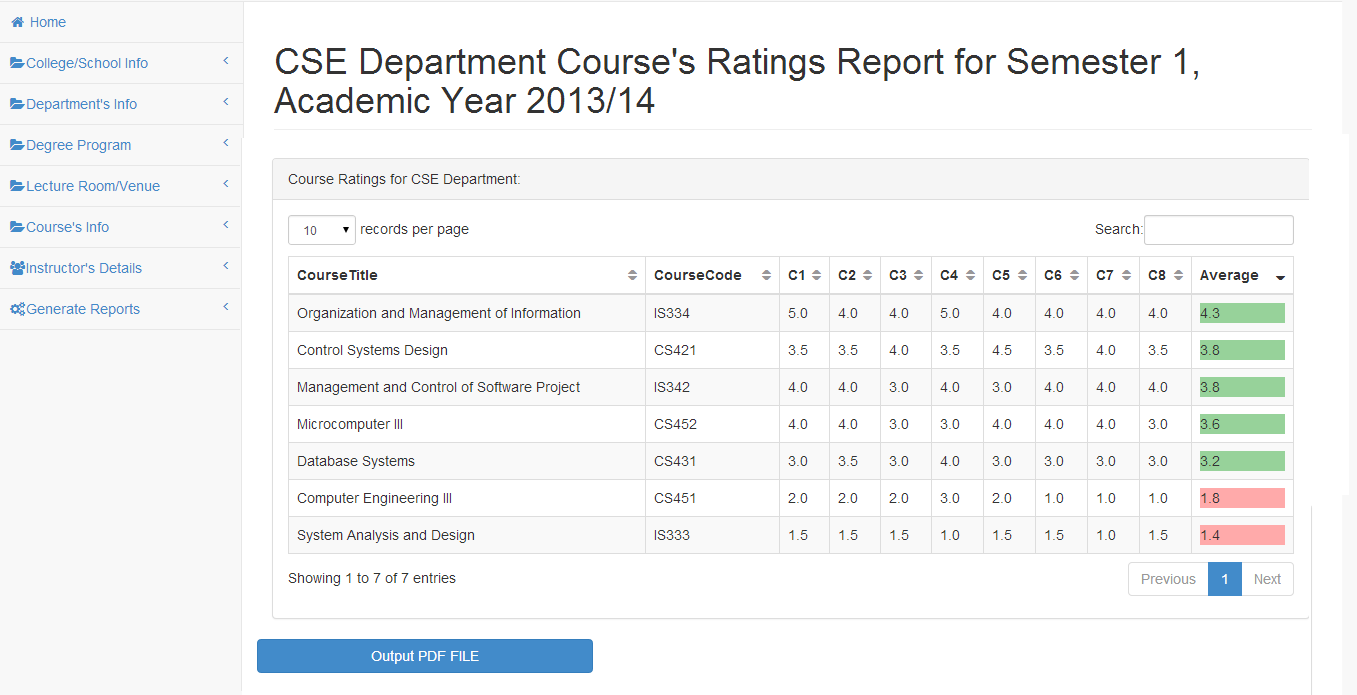
##### Figure 6.8: Instructor’s Details List

#### 6.2.4.4 Report Generation Module

Reports are the essential output of the project. They show what was performed and what was being worked on. Reports can be used to rate the weight of the work that was performed under certain circumstances. In iEvaluate, an administrator can generate different kinds of reports depends on what is required. Different kinds of reports can be obtained from the iEvaluate system; some of these reports require first an administrator to supply input which will be used for comparison with what is stored in the database. There is General Department Report which just shows the evaluation summary for a certain department (depends on admin selection), it shows the course for that department, the ratings for each particular course as well as the ratings for respective instructors. Another kind of report is the Instructor Ratings Report which is obtained after admin has selected department, semester and the academic year and the instructor ratings for that particular department will be displayed. Instructor Ratings Report can be obtained in Portable Document Format (PDF) format by downloading it from the system. Other reports are Department Course Ratings Report, Learning Environment, Facilities Reports and Course Comments Reports. These reports are also available in PDF format for printing purposes. Samples of the printed reports are shown in Appendix C. Figure 6.9 shows the course ratings report form which the administrator have to provide the input so as to get the required report and Figure 6.10 shows sample Course Ratings Report.



##### Figure 6.9: Course Ratings Report Form



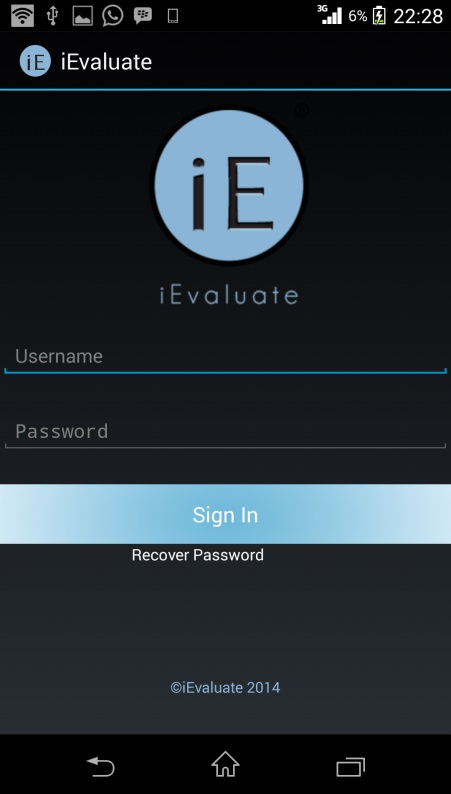
##### Figure 6.10: Course Ratings Report

## 6.3 PART TWO: iEvaluate Android Application

iEvaluate Android Application was created intentionally to speed up the process of collecting data during evaluation time and also give any-time, any-where feature increasing accessibility to the evaluation form using smart phones. The android application is for front-end users i.e. students who are responsible to evaluate/rate their courses. So all functionalities which were explained in iEvaluate Web Based Front-end Application are also implemented in the iEvaluate Android Application, the Application uses the same database that was already created so as to avoid data inconsistency.

### 6.3.1 Login Module

The login module for the android application uses the same approach as the login module in the web based application. User has to supply his/her login credentials which will be validated so as to use the application. Credentials will be validated with those in the main database i.e. main server. Figure 6.11 show the login module illustration for iEvaluate Android Application.



##### Figure 6.11: iEvaluate Android Application Login Form

### 6.3.2 Perform Evaluation Module

Likewise for the web based application, the evaluation process in the android application is the same. After a user has logged in will be provided with Welcome page activity and Course list activity which will enable the user to see all the courses he/she studies at that particular time. By clicking on the course tab/button will make the user to start evaluating that particular course. Processes for perform evaluation module are illustrated in Figure 6.12. Timeframe evaluation feature is also included in the iEvaluate Android Application as being seen in Figure 6.12. More screenshots for iEvaluate Android Application are attached in the Appendix C.

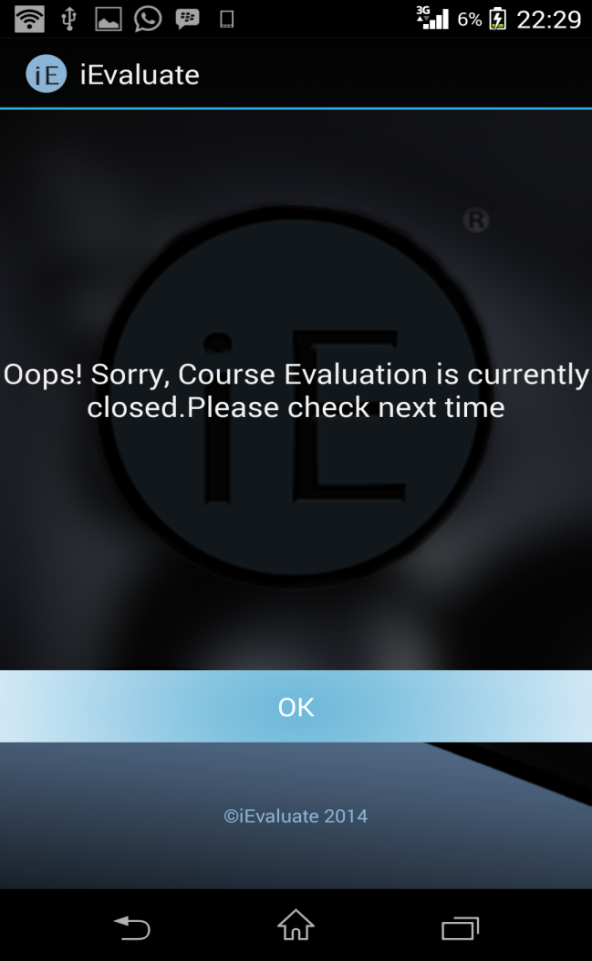


Figure 6.12(b): iEvaluate Android Application with Timeframe Feature

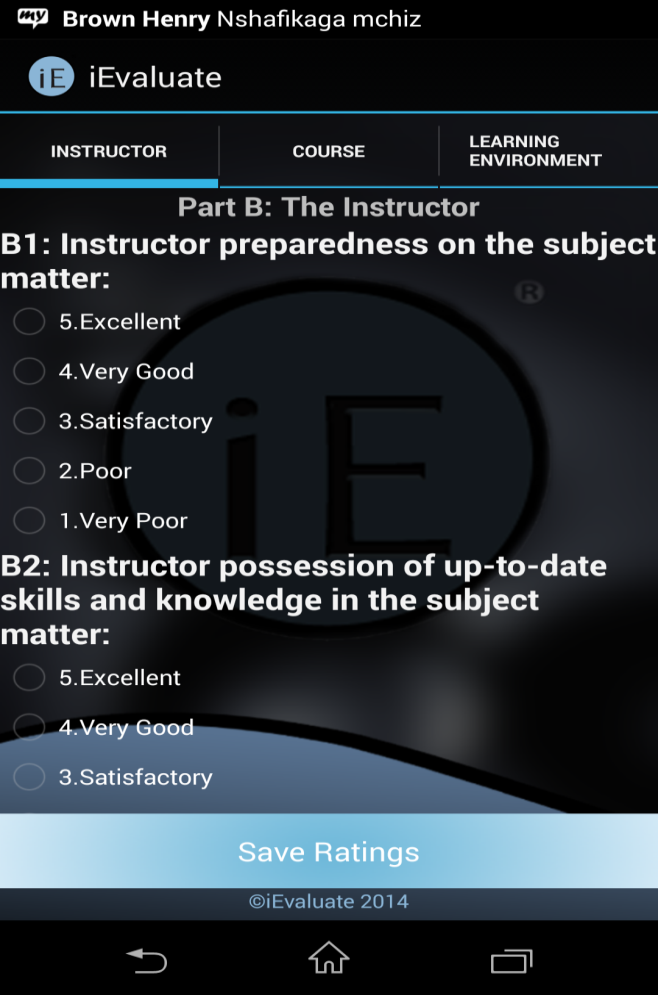


Figure 6.12(a): iEvaluate Android Application Perform Evaluation Module

##### Figure 6.12: iEvaluate Android Application Perform Evaluation Module with Timeframe Feature

# CHAPTER SEVEN: SYSTEM TESTING

## 7.1 Overview

System testing involves checking how the system behaves in various different environments. It assures both data integrity and system reliability. After completion of system implementation, system has to be tested so as to check if it meets the specific objectives as defined earlier. Under system testing each module has to be tested separately before integrate it with other modules. So modules are tested separately before integrated with each other so as to test the whole system. Since there are so many modules in this system, one module will be picked up and used to illustrate how it has being tested. Manage department module will be used for illustration purposes.

## 7.2 Module Testing

Manage department module involves adding new department as well as delete department and edit department. These entire functionalities system administrator has to perform. On adding new department, admin will be provided with form which he/she has to supply valid data for successful new department addition. Any attempt to submit invalid data or an empty form will be denied by displaying errors messages. Figure 7.1 illustrates validation of adding Department’s info feature.

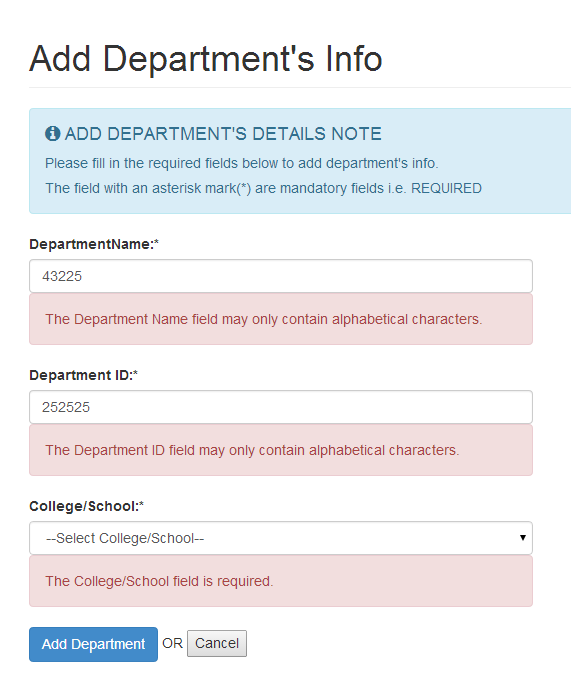


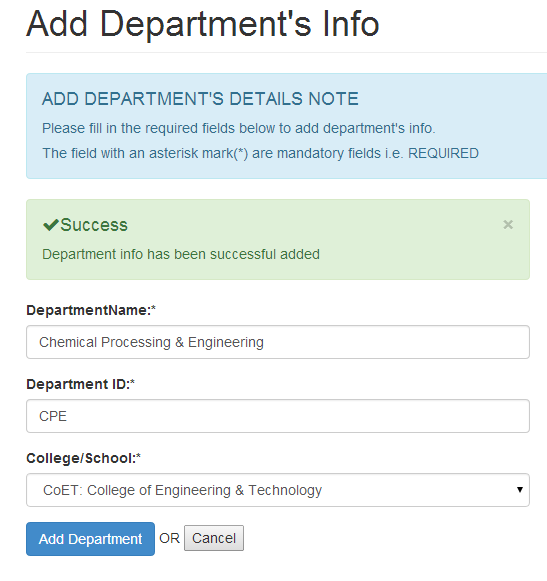
Figure 7.1(b): Check if Invalid Data are submitted



Figure7.1 (a): Check if Input Fields are Empty

##### Figure 7.1: Add Department Module Testing, Check for Required Fields and Invalid Data

If an administrator supplies valid data types, data entered will be saved in the database and a message for successful operation will be shown after the operation have been completed.



##### Figure 7.2: Add Department Module Testing on Success

## 7.3 System Testing

System Testing involves the act of testing the whole system with all of its modules. Each module passes testing stage successfully and all modules were integrated to form one system which was tested again as whole. In integrating the modules there were errors which occurred such as some modules failed to link with each other but the errors were solved accordingly. Successful completion of this stage justifies that the system efficiently does what it is supposed to do as per specified requirements.

# CHAPTER EIGHT: SUMMARY, CONCLUSION AND RECOMMENDATIONS

## 8.1 Summary

iEvaluate was developed following the waterfall model as the software development model including stages such as Requirement Capture and Analysis, System and Software Design, Implementation and Unit Testing, final Integration and System Testing. This system is expected to simplify the whole process of course evaluation by simplifying the process of data collection as well as data manipulation. With the use of iEvaluate the process of conduction course evaluation will be much easier compared to what was being used earlier i.e. traditional pen paper based evaluation.

## 8.2 Challenges Faced

In any Engineering work or project it involves the aim of solving a particular existing problem. The problems in one way or another cannot be solved freely; there are challenges which are encountered when taking particular engineering work. During development of iEvaluate there were challenges which were encountered some were technical while others were just logical. Most of the challenges were solved in the course of doing this project but till now there is a major one which is too logical and somehow technical. Generally a student is required to evaluate his/her core courses that are offered from his/her home department as well as those core courses from other department. Up to now a student can only evaluate the core courses from his/her home department only.

**8.3 Conclusion**

The process of course evaluation at the University of Dar es Salaam is monitored by the Quality Assurance Bureau (QAB). In this project, a system has been designed and developed to overcome the problems brought by the current manual system which is faced by QAB staff in the compilation, computation and reporting. The successful completion of this project will provide a tool which can be used by QAB staff in the whole process of course evaluation for the UDSM.

## 8.4 Recommendation

Online course evaluation systems are not quite new to many of the universities communities now days. However may find slow penetration/adoption to use owing to students fear that their identities will be revealed or in some cases change of mindset to use online systems posed by users both the administration and the students. Hence necessary mechanisms must be applied to make sure that the system is being used effectively for the benefit of the college, like users must be introduced to the system and be informed of the benefits of the system i.e. create awareness of the existence and importance of the system to the users and to the UDSM management. To assure the anonymity of the students, the list of responded must be automatically deleted at the end of the evaluation period.

For the one to perform evaluation he/she must register to the system. Currently at the University of Dar es salaam students are registered in Academic Registration Information System (ARIS) so they already have their accounts there and able to see all the courses they study at particular academic year. Making students registering again to iEvaluate indicates double registration. To avoid double registration for students, it is recommended that this system to be integrated to ARIS.

Evaluating core course from other department can be solved if iEvaluate becomes integrated with ARIS. With ARIS a particular student is able to view the core courses from home department as well as from other department that he/she is supposed to study them within a semester under certain academic year. Same approach used to display the core courses from home and from other department can be used also in iEvaluate.

# REFERENCES

Affiliates, E. C. (2013). DuckWeb. Retrieved November 30, 2013, from DuckWeb Information System: https://duckweb.uoregon.edu/pls/prod/twbkwbis.P\_WWWLogin

Bagile, D. (2013, November 22). CS 431, Database Systems. Dar es Salaam, Tanzania.

Data Collection Techniques. (2013). Retrieved December 2, 2013, from Quantitative, Positivist Research Methods in Information Systems: http://dstraub.cis.gsu.edu/quant/4datacoll.asp

EvaluationKit. (2013). EvaluationKit Mobile. Retrieved December 1, 2013, from evaluationkit.com: http://www.evaluationkit.com/?features=mobile

Golden, B. (2013). What is Systems Architecture ? Retrieved December 2, 2013, from Welcome, I'm Boris Golden: http://www.lix.polytechnique.fr/~golden/systems\_architecture.html

GUIDE, I. (2013). What are the Software Development Models? Retrieved December 1, 2031, from ISTQB Exam Certification: http://istqbexamcertification.com/what-are-the-software-development-models/

Kalinga, D. ( 2013, March). CS332 OBJECT-ORIENTED PROGRAM DESIGN AND ANALYSIS. Dar es Salaam, Tanzania: CSE, CoICT, UDSM,.

Kramer, J. (2014). Up and Running with Bootstrap 2. Retrieved April 17, 2014, from lynda.com: http://www.lynda.com/Bootstrap-tutorials/Up-Running-Bootstrap/110885-2.html

Navwa, Q. A. (2012). Pilot Study*.* Dar es Salaam: Quality Assurance Bureau.

Peck, J. (2014). Up and Running with PHP CodeIgniter. Retrieved April 17, 2014, from lynda.com: http://www.lynda.com/CodeIgniter-tutorials/Up-Running-PHP-CodeIgniter/126122-2.html

University, B. (2013). Course Evaluations. Retrieved November 21, 2013, from BRANDEIS.EDU : http://www.brandeis.edu/provost/faculty-info/courseevaluations /index.html

Waycott, J. &. (2009). Mobile and Web 2.0 technologies in undergraduate*.* Retrieved December 1, 2013, from Situating learning in everyday experience: http://www.ascilite.org.au/conferences/auckland09/procs/waycott.pdf

Wikipedia. (2013, March 25). Course Evaluation. Retrieved November 21, 2013, from Wikipedia Web site: http://en.wikipedia.org/wiki/Course\_evaluation

# APPENDICES

## APPENDIX A: Project Time Schedule and Budget

Table A1 shows the time schedule for semester one

Table A1: Time Schedule for Semester One

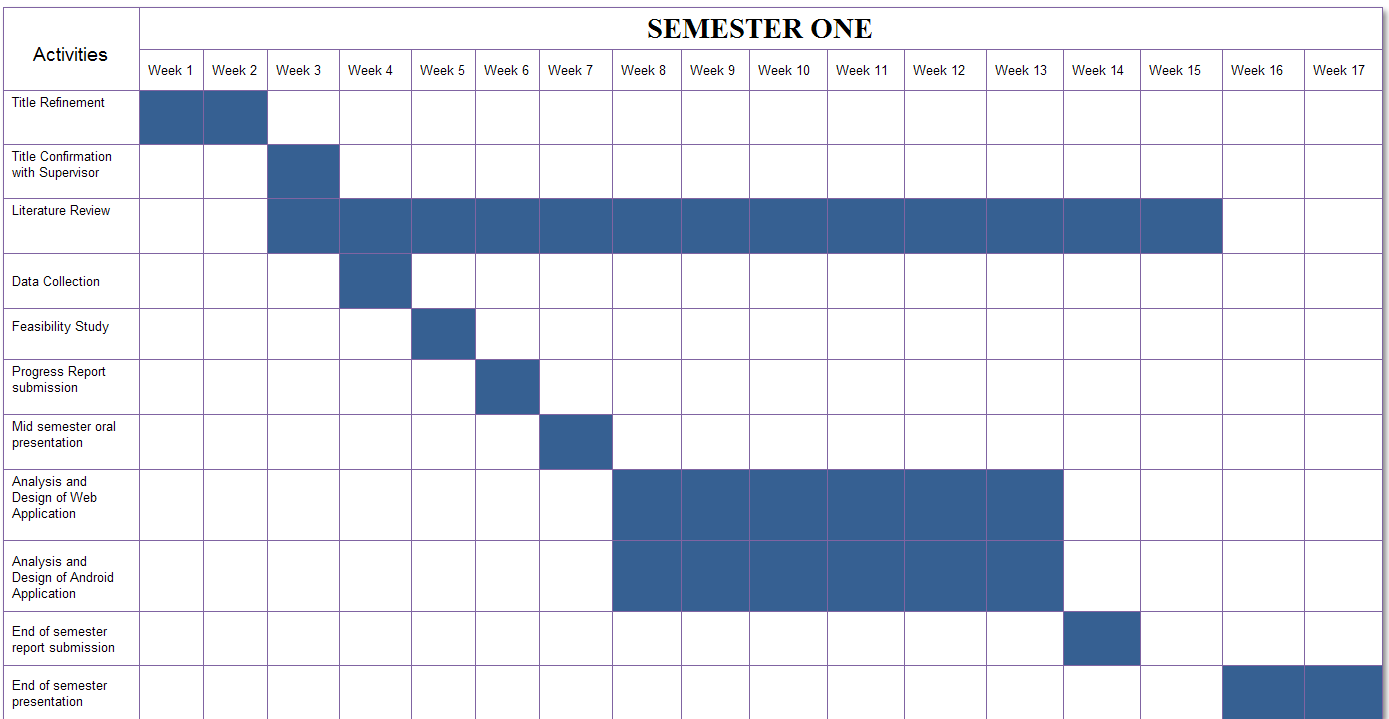
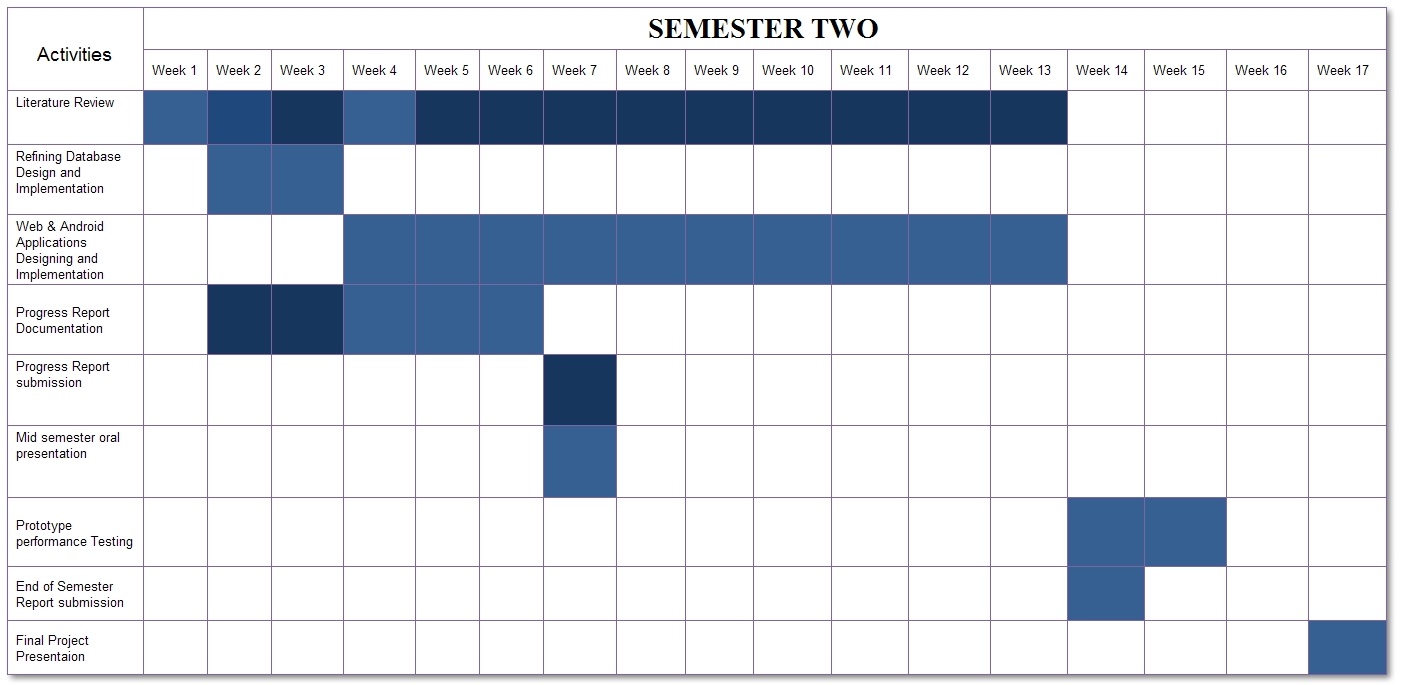


Table A2 shows the time schedule for semester two

Table A2: Time Schedule for Semester Two



To accomplish the process of developing this system hence making this project a success, Table A3 gives the project estimated costs.

Table A3: Budget Proposal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SN | Description | Quantity | Unity Price (Tshs.) | Total Price (Tshs.) |
| 1 | Printing | 250 pages | 100 | 25,000 |
| 2 | Report Binding | 4 reports | 1,000 | 4,000 |
| 3 | Photocopying | 400 pages | 70 | 28,000 |
| 4 | Internet packages | 60 hours | 1,000 | 60,000 |
| 5 | Android Device | 1 | 150,000 | 150,000 |
| 5 | Transport & Communication | - | - | 30,000 |
| 6 | Optical – CD | 4 | 700 | 2800 |
| 7 | Miscellaneous | - | - | 20,000 |
|  | | | **Total (Tshs.)** | **319,800** |

**APPENDIX B: Use Case Description Template and Sample Use Case Descriptions**

Table B1 shows the template for use case description and Table B2, B3 shows the sample use case description of the system.

Table B1: Use Case Description Template

|  |  |
| --- | --- |
| **Use case:** | Use Case name |
| **Actors:** | Role names of people or external entities the use case |
| **Short Description:** | A brief description of the Use Case |
| **Cross Reference:** | Related use cases and system functions. |
| **Pre-Conditions:** | A description of the conditions that must be satisfied before the use case is invoked |
| **Post-Conditions:** | A description of what has happened at the end of the use case. |
| **Main Flow:** | A list of the system interactions that take place under the most common scenario. |
| **Alternate Flow(s):** | A description of possible alternative interactions |
| **Exception Flow(s):** | A description of possible scenarios where unexpected events have taken place |

Table B2: Add Course Use Case

|  |  |
| --- | --- |
| **Use case:** | Add Course |
| **Actors:** | Admin |
| **Short Description:** | Allow admin to register the courses |
| **Cross Reference:** | R1.9 |
| **Pre-Conditions:** | The system must be on and admin must login |
| **Post-Conditions:** | New course will be added to the system |
| **Main Flow:** | 1. Admin click add course link 2. Add course form appear 3. Admin fills all required fields as well as optional fields 4. Admin click save button 5. Successfully message display |
| **Alternate Flow(s):** | - |
| **Exception Flow(s):** | Line 4: If required fields are empty or adding invalid data indicate error |

Table B3: Add Instructor Details Use Case

|  |  |
| --- | --- |
| **Use case:** | Add Instructor Details |
| **Actors:** | Admin |
| **Short Description:** | Allow admin to register the new instructor details |
| **Cross Reference:** | R2.0 |
| **Pre-Conditions:** | The system must be on and admin must login |
| **Post-Conditions:** | New instructor details will be added to the system |
| **Main Flow:** | 1. Admin click add instructor details button 2. Add instructor details form appear 3. Admin fills all required fields as well as optional fields 4. Admin click save button 5. Successfully message display |
| **Alternate Flow(s):** | - |
| **Exception Flow(s):** | Line 4: If required fields are empty or adding invalid data indicate error |

**APPENDIX C: iEvaluate Screenshots and Sample Output PDF Reports**

##### Figure C1: iEvaluate Android Application Sample Screenshots

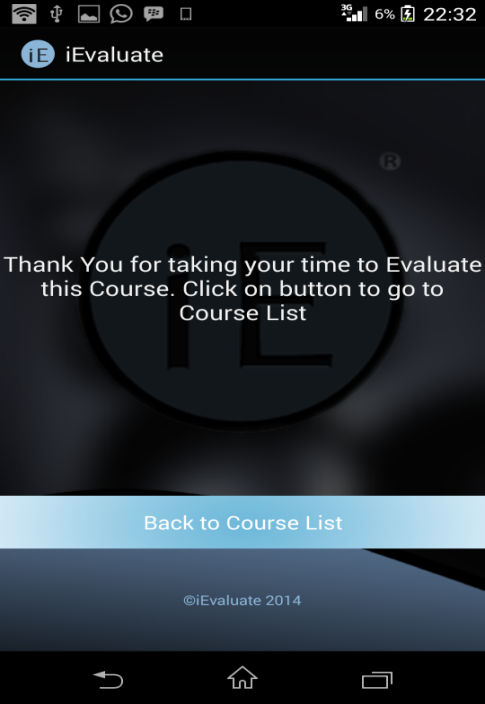


Figure C1 (d): iEvaluate Android Application Final Message after Evaluation

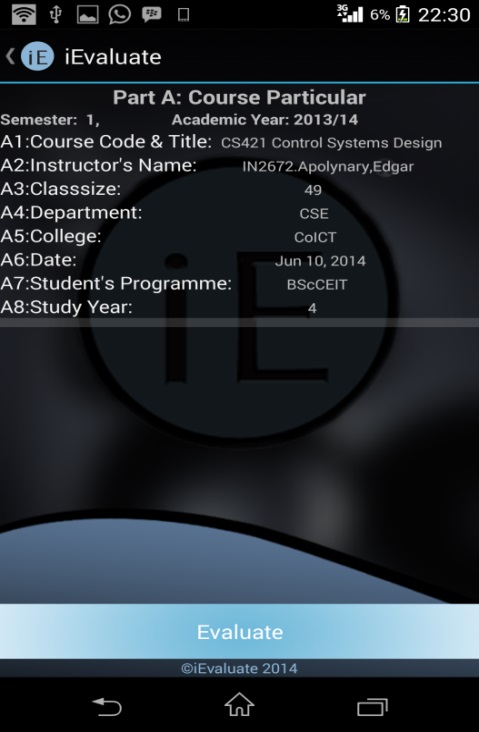


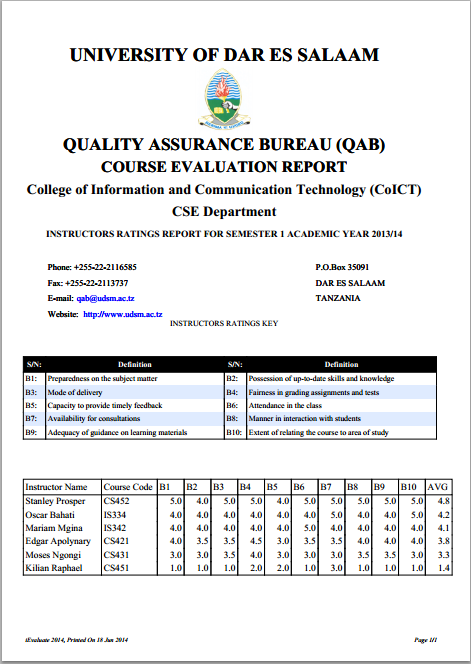
Figure C1 (c): iEvaluate Android Application Course Particular Page



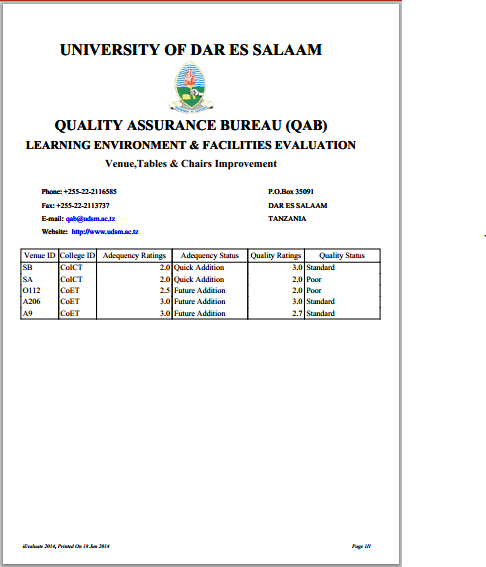
Figure C (b): iEvaluate Android Application, Course List Page



Figure C (a): iEvaluate Android Application Welcome Page



##### Figure C2: iEvaluate Sample Instructor Ratings PDF Report



##### Figure C3: iEvaluate Sample Venue Table and Chairs Improvement PDF Report