



Al Imam Mohammad ibn Saud Islamic University
College of Computer and Information Sciences
Information Systems Department

MCQ 4U



By:

Rawan Naif Alanazi	434024062	IS	271
Nourah Salem Albarqi	434025377	IS	271
Nowair Nasser Alotaibi	434026077	IS	271

Supervised by:

Arwa Alsafi

Project Submitted in Fulfillment for the IS498 Course requirements

Fall-2017

تعهد

أتعهد بعدم المشاركة في الفعاليات أو المبادرات أو المسابقات ذات العلاقة دون أخذ موافقة خطية مسبقة من الكلية، وأقر بمعرفتي أنني إذا خالفت هذا التعهد ستتم محاسبي وفق اللوائح والأنظمة.

اسم الطالبة	الرقم الجامعي	التواقيع
-------------	---------------	----------

Rouan

٤٣٤٠٢٤٠٦٢ ١. روان نايف العنزي

Nourah

٤٣٤٠٢٥٣٧٧ ٢. نوره سالم البارقي

Nouair

٤٣٤٠٢٦٠٧٧ ٣. نوير ناصر العتيبي



Al Imam Mohammad ibn Saud Islamic University
College of Computer and Information Sciences

Information Systems Department

MCQ 4U

By:

Rawan Naif Alanazi	434024062	IS	271
Nourah Salem Albarqi	434025377	IS	271
Nowair Nasser Alotaibi	434026077	IS	271

Supervisor Name	T.Arwa Alsafi
Date	10/4/1439 – 28/12/2017
Signature	

Declaration

We Rawan Naif Alanazi, Nourah Salem Albarqi, and Nowair Nasser Alotaibi being members of final year project group number one, declare that this report contains only work completed by members of our group except for information obtained in a legitimate way from literature, company or university sources. All information from these other sources has been duly referenced and acknowledged in accordance with the University Policy on Plagiarism.

Furthermore, we declare that in completing the project, the individual group members had the following responsibilities and contributed in the following proportions to the final outcomes of the project:

Student ID	Responsibilities ¹	Contribution ² %	Signature
434024062	Team leader, Analysis, Design, Implementation	100%	
434025377	Analysis, Design, Implementation	100%	
434026077	Analysis, Design, Implementation	100%	

¹ Write down your responsibilities in the project

² Must add to 100%

Dedication

We dedicate this work to our beloved family for always supporting us, without their love none of our success would be possible.

Also, we will not forget our beloved supervisor T. Arwa Alsafi, and coordinator Dr. Lujain Aldahash who's their advice and support contributed in presenting this work in a complete picture.

We dedicate "MCQ 4U" application to our beloved family for always supporting us, because they are driving force in our life and career.

Without their love, none of this world matter. Throughout our life, our family has actively supported us in our determination to find and realize our potential and to make this contribution to our CCIS.

Acknowledgment

First and foremost, we would like to present our deepest gratitude to Almighty ALLAH for the bounties and blessings for giving us the ability to finish this project.

It has been a great opportunity to gain lots of experience in real time projects, followed by the knowledge of how to actually design and analyze real projects.

We want to express our deepest thanks to our graduation project supervisors T. Arwa Alsafi and T. Alanoud Alotaibi for their patience and guidance along the year.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing the project within the limited time frame.

In addition, we would like to express our genuine appreciations to our graduation project instructor Dr. Lujain Aldahash for her understanding and direction support during the course.

Finally, thanks to Imam University's students in Information System department for encouragement during the course.

Abstract

This project is an educational application for students in College of Computer and Information Sciences at Imam University, which facilitates understanding a lecture before the next lecture, and allows knowledge exchange between students.

By using the application, the Faculty Member after each lecture can select a group of students that are able to generate Multiple-Choice-Questions and publish it to other classmates for solving. In addition, the application displays statistics forms based on students' solutions, so the Faculty Member can know the students' understanding level in a dynamic way.

Abstract (in Arabic)

هذا المشروع هو تطبيق تعليمي يسمح بتبادل المعرفة بين طلابات كلية علوم الحاسوب والمعلومات في جامعة الامام، حيث يسهل دراسة مواضيع المقررات الدراسية في وقت أقل وفهم المحاضرات قبل المحاضرة القادمة. كما يساعد عضو هيئة التدريس لاتخاذ قرار حول مستوى فهم الطالبات باستخدام نماذج إحصائية.

تتلخص آلية عمل التطبيق في أنه بعد كل محاضرة يقوم عضو هيئة التدريس باختيار مجموعة من الطالبات لإنشاء نموذج يحتوي على أسئلة اختيار من متعدد ونشرها لبقية الطالبات للإجابة على الأسئلة ومن خلال الإجابات سيكون لدى عضو هيئة التدريس لمحة عامة عن مستوى تفكير الطالبات عن طريق عرض إحصائيات إجاباتهن لكل محاضرة بطريقة ديناميكية.

Keywords

Education
Android application
MCQ
Student learns
Exam
Examination System
Lecture Review

List of Abbreviations

CCIS: The College of Computer and Information Sciences at Imam University.

MCQs: Multiple Choice Question that will be generated by students.

UML: Unified Modeling Language

IT: Information Technology

MCQs: Multiple Choice Questions

CCIS: College of Computer and Information Sciences

OS: Operating System

SQL: Structured Query Language

RAM: Random Access Memory

WBS: Work Breakdown Structure

CPU: Central Processing Unit

ISP: Internet Service Provider

Table of Contents

DECLARATION.....	IV
DEDICATION.....	V
ACKNOWLEDGMENT.....	VI
ABSTRACT	VII
ABSTRACT (IN ARABIC)	VIII
KEYWORDS.....	IX
LIST OF ABBREVIATIONS.....	X
TABLE OF CONTENTS.....	XI
LIST OF FIGURES	XIV
LIST OF TABLES	XVI
CHAPTER ONE: INTRODUCTION	1
INTRODUCTION.....	2
1.1: PROJECT OVERVIEW	2
1.2: PROBLEM STATEMENT	2
1.3: PROJECT IMPACT	3
1.4: PROJECT SCOPE	3
1.5: OBJECTIVES.....	3
1.6: BUSINESS REQUIREMENTS.....	4
1.7: ALTERNATIVE SOLUTIONS	5
SUMMARY	6
CHAPTER TWO: BACKGROUND.....	7
INTRODUCTION.....	8
2.1: OVERVIEW OF EXISTING SYSTEMS	8
2.1.1: Blackboard	8
2.1.1.1: Blackboard Advantages	9
2.1.1.2: Blackboard Disadvantages:.....	9
2.1.2: Custom Exam	10
2.1.2.1: Custom Exam Advantages	10
2.1.2.2: Custom Exam Disadvantages.....	10
2.1.3: MCQ Computer Science	10
2.1.3.1: MCQ Computer Science Advantages	10

2.1.3.2: MCQ Computer Science Disadvantages.....	11
2.1.4: Quiz Works	11
2.1.4.1: Quiz Works Advantages	11
2.1.4.2: Quiz Works Disadvantages	12
2.2: EXISTING BUSINESS PROCESSES.....	13
2.3: METHOD / APPROACH	14
2.4: PROJECT PLANNING.....	14
SUMMARY	18

CHAPTER THREE: REQUIREMENTS ANALYSIS.....19

INTRODUCTION.....	20
3.1: REQUIREMENT GATHERING TECHNIQUES.....	20
3.1.1: Questionnaire	20
3.1.2: Observation	21
3.1.3: Group sessions	21
3.2: PROPOSED BUSINESS PROCESS	22
3.3: FUNCTIONAL REQUIREMENTS	24
3.3.1: Admin.....	24
3.3.2: Faculty Member	25
3.3.3: Student	25
3.3.3.4: Coordinator Student	26
3.3.3.5: Asker Student.....	26
3.3.3.6: Respondent Student.....	27
3.3.4: MCQ 4U Use case Daigram.....	27
3.3.5: MCQ 4U Use case Desception	27
3.4: NON-FUNCTIONAL REQUIREMENTS	36
3.4.1: Related to performance or usability	37
3.4.1.1: Usability	37
3.4.1.2: Reliability.....	37
3.4.1.3: Performance	37
3.4.1.4: Security	37
3.4.2: Related to non-technical issues	38
3.4.2.1: Response time	38
3.5: USER INTERFACES	38
SUMMARY	42

CHAPTER FOUR: SYSTEM DESIGN.....43

INTRODUCTION.....	44
4.1: SYSTEM MODELLING.....	44
4.1.1: Activity diagrams	44
4.1.2: Class Diagram	55
4.1.3: Sequence Diagrams.....	56

4.2: DATA MODELLING	64
4.2.1: ER Diagram.....	65
4.2.2: Data Dictionary	65
4.3: DETAILED INTERFACE DESIGN.....	69
4.3.1: Input design.....	69
4.3.2: Output interface.....	72
4.4: COMPONENT DIAGRAM	74
4.5: ALTERNATIVE DESIGNS.....	75
SUMMARY	75
CHAPTER FIVE: SYSTEM IMPLEMENTATION.....	76
INTRODUCTION.....	77
5.1: SYSTEM SPECIFICATION	77
5.1.1: Hardware Specifications	77
5.1.2: Software Specifications.....	78
5.2: SYSTEM TESTING	80
5.2.1: Features to be tested.....	80
5.2.2: Test Cases	80
5.3: SYSTEM DEPLOYMENT	87
5.3.1: Deployment Diagram.....	87
5.3.2: Expected Users/Load	87
5.3.3: Installation Process	88
SUMMERY	88
CHAPTER SIX: CONCLUSION AND FUTURE WORK.....	89
INTRODUCTION.....	90
6.1: CONCLUSION	90
6.2: FUTURE WORK.....	91
SUMMERY	91
REFERENCES	92
APPENDIX	94
A. CODE SNIPPETS	95
B. PRESENTATION SLIDES.....	114
C. MISCELLANEOUS	115
Appendix A: Student questionnaire	115
Appendix B: Faculty member questionnaire.....	118

List of Figures

Figure 1: Blackboard login form.....	8
Figure 2: SU Blackboard.....	8
Figure 3: Custom Exam Home Page	10
Figure 4: MCQ Computer Science logo	10
Figure 5: shows the subjects covered in the MCQ Computer application.....	11
Figure 6: Quiz Works interface.....	11
Figure 7: AS-IS system for making MCQs in the class	13
Figure 8: Waterfall Model SDLC [10].....	14
Figure 9: WBS of MCQ 4U application.	15
Figure 10: Gantt chart of MCQ4u Project.	15
Figure 11: Business Process of MCQ 4U (System To-be)	23
Figure 12: Continue Business Process of MCQ 4U (System To-be).....	24
Figure 13: MCQ 4U Admin Use case diagram.	27
Figure 14: use case diagram of the MCQ 4U.....	27
Figure 15: MCQ 4U Registration form.....	39
Figure 16: MCQ 4U Log in form.....	39
Figure 17: MCQ 4U First interface.....	39
Figure 18: Create course	39
Figure 19: Link students to course	40
Figure 20: link Faculty member to course	40
Figure 21: Faculty member selecting task	40
Figure 22: Select Students Form.....	40
Figure 23: Select statistics Form	41
Figure 24: Answer Statistics Form.....	41
Figure 25: Answer Statistics Form.....	41
Figure 26: Lecture Questions form	41
Figure 27: MCQ Form	42
Figure 28: Activity diagram for all MCQ use cases	44
Figure 29: Build MCQ survey Activity diagram.	45
Figure 30: Publish MCQ survey Activity diagram.	46
Figure 31: View Solution Statistics Activity diagram.	47
Figure 32: Submit Answer Activity diagram.	48
Figure 33: Create MCQ Questions Activity diagram	49
Figure 34: Send message to Asker Students Activity diagram.....	50
Figure 35: Finalize MCQ Survey Activity diagram.....	51
Figure 36: Create course and Link Faculty Member Activity diagram	52
Figure 37: Delete course Activity diagram	53
Figure 38: Link student Activity diagram.....	54
Figure 39: Delete Faculty Member and student Activity diagram.....	55
Figure 40: Class diagram for MCQ 4U.....	56
Figure 41: Build MCQ survey sequence diagram.....	57
Figure 42: Publish MCQ survey sequence diagram.....	57
Figure 43: Submit Answer sequence diagram	58
Figure 44: view solution statistics sequence diagram.....	58
Figure 45: Submit Answer sequence diagram	59
Figure 46: Send Message to Asker Students sequence diagram.....	59
Figure 47: Finalize MCQ Survey sequence diagram	60

Figure 48: Create Course and Link Faculty Member sequence diagram.....	61
Figure 49: Delete Course sequence diagram.....	62
Figure 50: Link Student sequence diagram.....	63
Figure 51: Delete Faculty Member and Student sequence diagram	64
Figure 52: ER diagram.....	65
Figure 53: Component diagram	75
Figure 54: Deployment Diagram	87
Figure 55: Question 1 Graph.....	115
Figure 56: Question 2 Graph.....	115
Figure 57: Question 3 Graph.....	115
Figure 58: Question 4 Graph.....	116
Figure 59: Question 5 Graph.....	116
Figure 60: Question 6 Graph.....	116
Figure 61: Question 7 Graph.....	116
Figure 62: Question 8 Graph.....	117
Figure 63: Question 9 Graph.....	117
Figure 64: Students Suggestions	117
Figure 65: Faculty Members suggestions	118

List of Tables

Table 1: Comparison between existing systems	13
Table 2: Team Member Responsibilities	17
Table 3: Build MCQs Survey UseCase description.....	28
Table 4: Publish MCQ Survey UseCase Description	29
Table 5: View Solution Statistics UseCase Description	30
Table 6: Submit answer UseCase Description	31
Table 7: Create MCQ Question UseCase Description	32
Table 8: Send Message to Asker Students UseCase Description	32
Table 9: Finalize MCQ Form UseCase Description	33
Table 10: Create Course UseCase Description	34
Table 11: Delete Course UseCase Description	35
Table 12: Link Faculty Member and Student UseCase Description.....	35
Table 13: Delete Faculty Member and Student UseCase Description.....	36
Table 14: Non-functional requirements	37
Table 15: User interfaces	42
Table 16: Entities Table	66
Table 17: data description (relationship)	66
Table 18: data description (attributes)	69
Table 19: Input interface	72
Table 20: Output interface	74
Table 21: Mobile Device.....	77
Table 22: computer Device	78
Table 23: Software Specifications	79
Table 24: Register test case.....	80
Table 25: Login test case	81
Table 26: Build a survey test case.....	81
Table 27: Send message to Asker Student test case.....	82
Table 28: Create MCQ question test case	82
Table 29: Delete MCQ question test case	83
Table 30: Review a survey test case	83
Table 31: Publish a survey test case.....	84
Table 32: Answer a survey test case	84
Table 33: Delete user test case	85
Table 34: Link Student to course test case.....	85
Table 35: Delete course test case	86
Table 36: Create course test case	86
Table 37: Questionnaire	117

Chapter One: Introduction



Introduction

This chapter is an introduction to MCQ 4U application's environment that supports Students and Faculty Member as stakeholders. It highlights an overview of the project, problem statement, project impact, project scope, project objectives, business requirements, and alternative solutions.

1.1: Project Overview

CCIS Students suffering from lack of sufficient time to review previous lectures, so the lessons will accumulate on Students and lead to less understanding as a result.

This project aims to provide a mobile application that ensures the students understand and review the previous lectures before the next lecture begin.

There is a lot of software that generates online MCQs for students, but only accessed and used by the Faculty Member, which does not support the "MCQ 4U" idea, that allow students themselves to generate the questions.

MCQ 4U is an educational application which helps students to study in less time by reviewing and studying the MCQs presented by the application.

The application serves Faculty Member, by producing statistics about the understanding level of students after each lecture depending on their answers on MCQ survey, in order to take decisions to repeat the non-understandable parts from the previous lecture.

1.2: Problem statement

Students are suffering in finding enough time in reviewing the received lectures, due to their preoccupation with projects. And they have problems in managing their time to study the materials they received. Therefore, lectures will accumulate and result in less understanding of the course [1].

In order to improve and increase the level of students' understanding of the received lectures, the "MCQ 4U" application allows Students to review the received lectures by answering the MCQ survey produced by the application and generated by the students.

The Faculty Member can use the application to assess the students' understanding of the material, which will help before writing the exam questions.

1.3: Project Impact

“MCQ 4U” serves CCIS in Imam University by increasing Faculty Member's eagerness to teach courses in a clear and efficient way, and improving the university reputation through their graduates who are thoroughly versed and skilful in their specialization

The Faculty Member will be aware of the level of students' understanding of the course before writing the exam and gain opportunity to know the less understandable parts and explain it again to students.

1.4: Project Scope

The “MCQ 4U” is created to help Students in their daily study by generating and solving MCQ survey. Also, the application helps Faculty Members in measuring the extent of students' understanding of the course by reading the graphs produced in the application that include students' answers which helps Faculty Member in deciding which parts of the previous lesson to repeat for students. In addition, helps the Faculty Member for selecting questions when generating exams.

“MCQ 4U” is an Android mobile application which serves CCIS Imam University, the female student section, and Faculty Members. "MCQ 4U" application supports only English language, Android devices.

In the first version, the application does not support other languages such as Arabic, and other colleges in Imam University.

1.5: Objectives

This project has the following objectives:

1. Enhance students' self-learning skills.
2. Allow Faculty Members in perceiving the level of understanding of the course's material by students.

3. Improve thinking and writing skills for students when they are generating MCQs.
4. Provide for the Faculty Members a database for MCQs that generated by Asker Students, which considered as a reference for Faculty Member to make revision before exams. Provide an easy-to-use application, that facilitates the assessment function for the Faculty Members by producing graphs automatically after the students answering the MCQ survey. In addition, students will write the MCQs in an electronic form and there is no need to use papers.

1.6: Business Requirements

The application has three primary stakeholders; Students, Faculty Member and Admin. Business requirements are important in determining their needs. The system should provide three different interfaces to give different authorizations for a Faculty Member, Student and Admin.

For the Admin's interface:

- The system allows Admin to register by filling the ID, first name, middle name, last name and a password, then allows the admin to Log in by entering ID and password.
- The system allows for Admin to create a course by filling course name, course ID, section number, number of lectures and select Faculty Member, then click create course button
- The system allows for Admin to link Faculty Member and Student to the course. The Faculty Member should be linking during the create course process, then allows the Admin to link Student by select a course and student and click on link button.
- The system allows for Admin to delete Faculty Member, Student and course By select Faculty Member or Student or course then click on delete button.

For the Faculty Members interface:

- The system allows the Faculty Member to register by filling the ID, first name, middle name, last name and a password, then allows the Faculty Member to Log in by entering ID and password.

- After that, the system allows the Faculty Member to build MCQ survey and select group contains five students, four Asker Student are responsible for writing MCQ question, and one student coordinates the group work.
- The system allows Faculty Member to approve and publish MCQ survey.
- After the students answered the MCQ survey, the system generates charts of students' answers, which helps the Faculty Member in viewing their answers and assessing them.
- Finally, the system allows the Faculty Member to log out when clicks on the logout button.

For the students' interface:

- The system allows Student to register by filling the ID, first name, middle name, last name and a password, then allows the student to Log in by entering ID and password.
- After the Faculty Member select group of five Students, the Coordinator divide the parts of the lecture between Asker Students who were selected for writing MCQ survey question, then each Asker Student will send the question to the coordinator. Then, the system should allow the coordinator to review MCQ survey.
- After that, the system should allow the coordinator to send the MCQ survey to Faculty Member.
- Then, the system should allow the Respondent Student to answer MCQ survey by click on survey, then solve questions and submit answers.
- Finally, the system allows Student to log out by clicking on the logout button.

1.7: Alternative Solutions

The Students always seek for sources and references that motivate their daily study and help them to practice for exams questions idea from different viewpoints. Some different ways help students to visualize the exams such as Blackboard, websites for students to create and share exam, application contains fixed number of multiple-choice questions that support specialization in a single field.

MCQ 4U as a mobile application that will be supported by Imam University as an educational institution, like Blackboard, but "MCQ 4U" mobile application offer for students to generate MCQs by themselves, and can approved by the Faculty Member.

The project team decide to develop MCQ 4U as an android mobile application for Faculty Members and Students to easily connect to Admin website for managing Students and Faculty Member courses in a short time.

It is a dynamic interactive application that ensures the student collaborate with each other to gain the benefits from the application, and helps to reduce the cost estimation for the project team to developing it in a freely or less expensive than ios, and the Android device is less expensive in ability to buy by students.

Summary

In conclusion, this chapter defines the stakeholders and propose a new solution that will help to improve students' self-learning skills. The "MCQ 4U", is an Android application that serves CCIS Imam University, the female student section, and faculty members, and supports only English language.

Chapter Two: Background

Introduction

This chapter includes an overview of the existing systems that can deal with the MCQ 4U idea in the recent and team member makes literal reviews of existing software can help Faculty Member and student to reach the objective of MCQ 4U idea such as Blackboard, Custom Exam website, MCQ computer science, and Quiz Works. Therefore, team member had discussed the advantages and disadvantages of theses software.

Moreover, the methods and approach that chosen for this research explained. In addition, it covers the WBS, Gantt chart with the responsibility for each member in "MCQ 4U" application.

2.1: Overview of existing systems

This section provides an overview of the existing systems, their advantages and disadvantages and comparing their features to MCQ 4U application.

2.1.1: Blackboard

The Blackboard as in Figure 1 is a virtual learning environment and course management system developed and programmed by Blackboard Corporation.



Figure 1: Blackboard login form

It can be integrated with the students' information systems and the authentication protocols. It can be installed on local servers or hosted by Blackboard ASP Solutions. Its main purposes are to add online elements to courses and to develop completely online courses with little or no face-to-face meetings [6].



Figure 2: SU Blackboard

The admin adds all the semester courses so when the Faculty Members click on the Courses tab (Figure 2), the Faculty Member can select the courses they teach and they can add their students to be an Instructor, a Teaching Assistant, or a Course Builder.

2.1.1.1: Blackboard Advantages

Blackboard Learn provides Faculty Members and students with a platform for communication and sharing content. Also, it provides the following features:

- **Assessments:** This feature allows Faculty Members to post quizzes and exams by writing in MCQ form and allows students to access them via the internet.
- **Notifications:** This feature provides students with notifications about a new quiz to solve, or another course appendix.
- **Mail:** Blackboard mail allows students and Faculty Members to exchange emails. This feature supports a great emailing feature to students in a course.
- **Calendar:** This feature allows Faculty Members to post their dates for assignments, projects and exams.
- **Assignments:** This feature allows assignments to be posted and students to submit assignments and exams online.
- **Grade Book:** Faculty Members may post grades on Blackboard for students to view.

2.1.1.2: Blackboard Disadvantages:

- It does not have statistics for evaluating the students after each lecture.
- It does not facilitate information sharing between students.

2.1.2: Custom Exam

Custom Exam website as shown in Figure 3, allows students to create and share their own exams with friends. They can manage exams easily by create, edit exams information and questions [7].

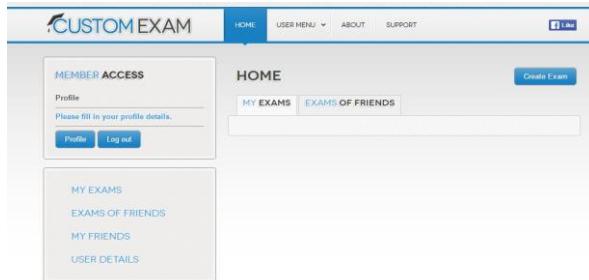


Figure 3: Custom Exam Home Page

2.1.2.1: Custom Exam Advantages

- Enable student to write an exam that includes many questions.
- Enable student submit exam form to specific or all friends.

2.1.2.2: Custom Exam Disadvantages

- Does not provide statistics feature for the Faculty Member to evaluate students understanding level.
- Does not provide Android or iOS version.



Figure 4: MCQ Computer Science logo

2.1.3: MCQ Computer Science

Computer Science MCQ Android application contains multiple-choice questions (MCQs) and answers from MCQ bank, which helps students to review the basic concepts and principles for the Computer Science department. [8]. Figure 4 shows the application logo.

2.1.3.1: MCQ Computer Science Advantages

- Each question has an answer and supports explanations.

- The application is useful for preparing competitive exams like GATE, IBPS, ISRO and other entrance exams.

2.1.3.2: MCQ Computer Science Disadvantages

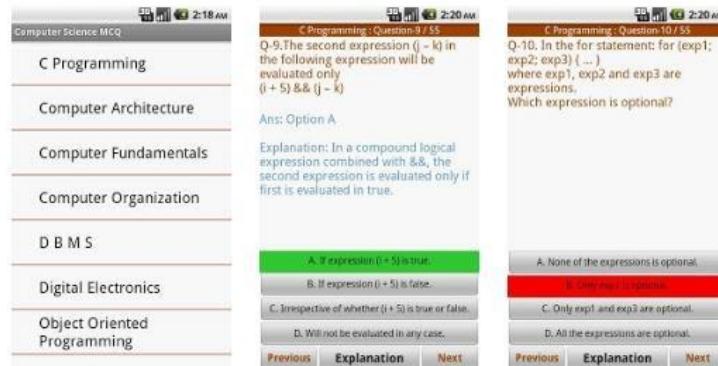


Figure 5: shows the subjects covered in the MCQ Computer application

- Non-modified, fixed database subjects, with a limit number of MCQs.
- Using it as a self-learning source that is not related to an official educational institution.

2.1.4: Quiz Works

Quiz Works is an online learning management system that used to build online e-learning courses, create exams to test the users at the end of each course, and make assessment to evaluate students [9].



Figure 6: Quiz Works interface

2.1.4.1: Quiz Works Advantages

- Easy to build lessons, courses and questions.
- Allow the instructor to create interactive lessons that combining text, practice questions and real tests and exams.

- The exam can be produced as a standalone exam or embedded it within the Faculty Member's course.
- A wide range of question types available: multiple choice, fill in the blanks, free text, etc.
- Students can see the correct answers so they can learn.
- Allow the users to solve the exam more than once, so they can learn from their mistakes.
- The Faculty Member can set the scoring rules and determine what a student has to score to pass the exam. Keep track Faculty Member's students, and download the results and stats to evaluate the course.

2.1.4.2: Quiz Works Disadvantages

- Expensive to download.
- The version does not support Arabic language.
- Disturbance the Faculty Member to write a huge number of questions in case of giving his\her student more than one chance to resolve questions.

2.1.5: Comparison between existing systems

This section provides a comparison between existing systems. Table 1 shows the advantages and disadvantages of each system according to certain criteria. This will help when developing the “MCQ 4U” application, by determining the features that should be improved and provided.

	Blackboard	Custom Exam	MCQ Computer Science	Quiz Works
Supporting the educational institution	Yes	No	No	No
Software type	Website and Android /iOS application	Website	Android application	Website/Application
Generate MCQs	Yes	Yes	No	Yes

Is it free?	Yes	Yes	Yes	No
Allow students to repeat quiz	No	Yes	Yes	Yes
Ability to explain the MCQ?	No	No	Yes	Yes
Display the results immediately	No	Yes.	Yes	Yes
Allow Faculty Member to evaluate the students' answers	Yes	Yes	No	Yes
Allow students solve MCQs in a given time?	Yes	Yes	No	Yes

Table 1: Comparison between existing systems

2.2: Existing Business Processes

The analytical representation and details of our application business processes using the As-IS are used to map out the current (or “as-is”) processes to create a baseline for process improvements. An “as is” business process contains all of the sections in a typical business process model described. Figure 7 shows the business process of the existing system.

The Faculty Member starts to explain the lecture. Then, at the end of the lecture the

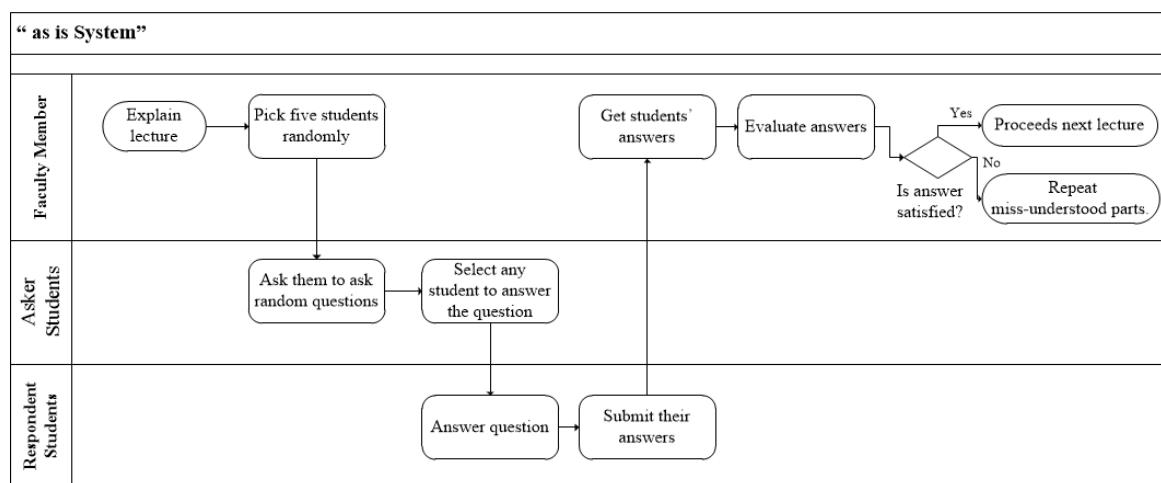


Figure 7: AS-IS system for making MCQs in the class

Faculty Member selects five Asker Students randomly. The selected Asker Students will ask questions about the received lecture and select some Respondent Students to answer them. The other Students in the class should write down the answers and give

them to the Faculty Member. The Faculty Member evaluates the understanding level of students according to their answers and takes a decision to repeat the incomprehensible parts of the lecture.

2.3: Method / Approach

The Waterfall Model (linear-sequential model) selected as the approach of this project. Figure 10 shows the phases of this model. The waterfall Model is very simple to understand and use. In a waterfall model, each phase must be completed fully before the beginning of the next phase. This type of software development model is basically used for small projects where requirements are very well understood and easy to manage. MCQ 4U requirements are very well known; clear and fixed, so the waterfall is a very appropriate model for this project. The waterfall phases are processed and completed without overlapping.

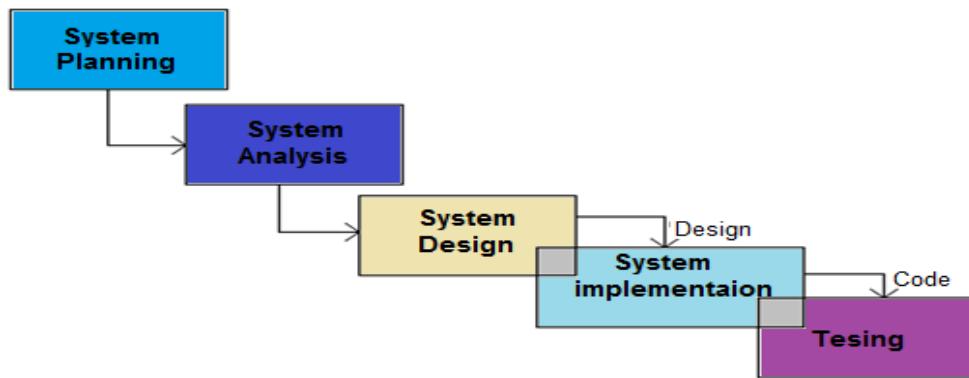


Figure 8: Waterfall Model SDLC [10]

At the end of each phase, a review takes place to determine if the project is on the right path and whether not continue or discard the project [11].

2.4: Project Planning

The project planning section describes the plan and the method used to implement “MCQ 4U” project, as shown in the WBS (Figure 9) that divided the “MCQ 4U” project into fixed phases. Gantt chart describes the duration of each phase and the which tasks assigned to every member of the project team.

2.4.1 Work break down structure (WBS):

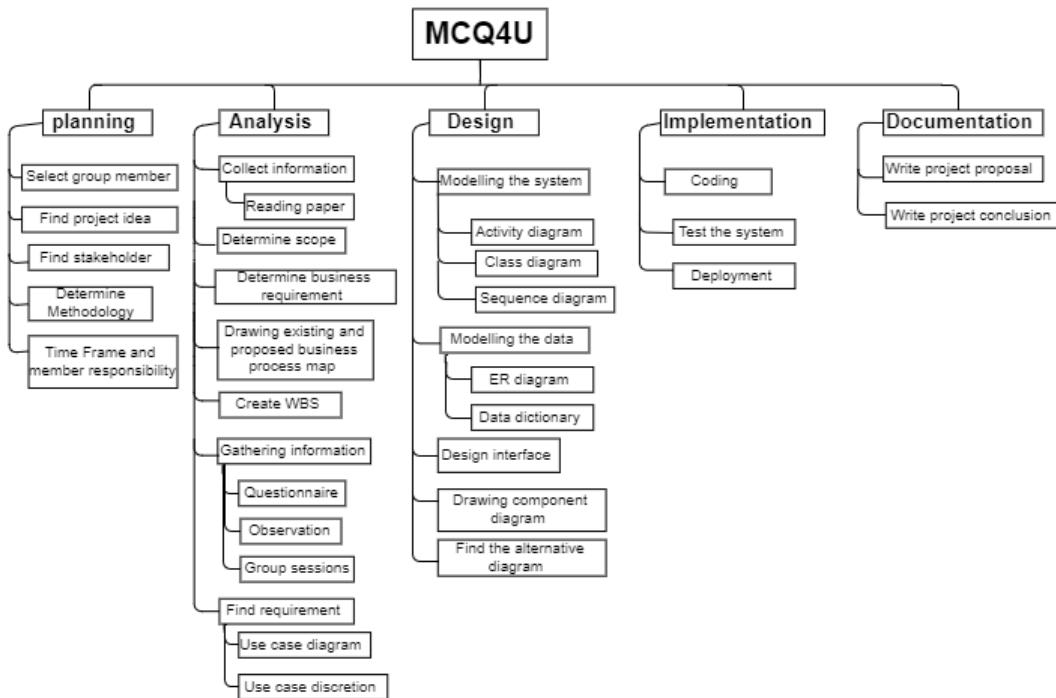


Figure 9: WBS of MCQ 4U application.

2.4.2 Time Frame (Gantt chart):

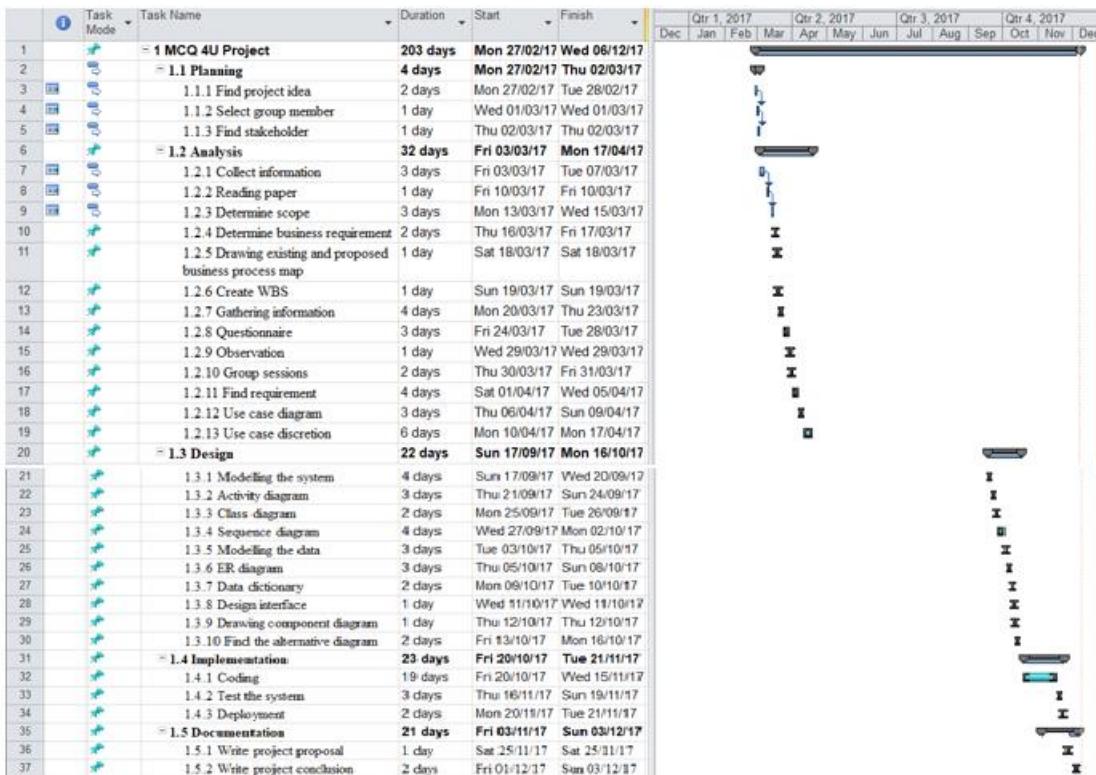


Figure 10: Gantt chart of MCQ4u Project.

Figure 10 shows a graphic chart (Gantt chart), which is a bar chart that illustrates a project schedule. It displays the phases of the project (planning, analysis, design, implementation, testing and documentation) and the duration time for each phase.

2.4.3 Team member Responsibilities:

Every task in the "MCQ 4U" application is divided into sub-tasks, and each sub-task will be assigned to a member or more. Table 2 shows the responsibility of each team member, the start date, and the end date of each sub-task.

Task number	Task name	Team member responsible
1	Planning	
1.1	Select group member	Rawan Naif Alanazi
1.2	Find project idea	Rawan Naif Alanazi
1.3	Find stakeholder	Nourah Salem Albarqi & Nowair Nasser Alotaibi & Rawan Naif Alanazi
2	Analysis	
2.1	Collect information	Rawan Naif Alanazi
2.1.1	Reading paper	Nourah Salem Albarqi & Nowair Nasser Alotaibi & Rawan Naif Alanazi
2.2	Determine scope	Nourah Salem Albarqi & Nowair Nasser Alotaibi & Rawan Naif Alanazi
2.3	Determine business requirement	Nourah Salem Albarqi & Nowair Nasser Alotaibi & Rawan Naif Alanazi
2.4	Drawing existing and proposed business process map	Nourah Salem Albarqi & Rawan Naif Alanazi & Nowair Nasser Alotaibi
2.5	Create WBS	Nowair Nasser Alotaibi
2.6	Gathering information	Nourah Salem Albarqi & Rawan Naif Alanazi & Nowair Nasser Alotaibi
2.6.1	Questionnaire	Rawan Naif Alanazi
2.6.2	Observation	Nourah Salem Albarqi
2.6.3	Group sessions	Nowair Nasser Alotaibi
2.7	Find requirement	Nourah Salem Albarqi & Rawan Naif

		Alanazi &Nowair Nasser Alotaibi
2.7.1	Use case diagram	Nourah Albarqi & Rawan Alanazi & Nowair Alotaibi.
2.7.2	Use case description	Nourah Albarqi
3	Design	
3.1	Modelling the system	Nourah Albarqi & Rawan Alanazi & Nowair Alotaibi.
3.1.1	Activity diagram	Nourah Salem Albarqi
3.1.2	Class diagram	Nourah Salem Albarqi & Nowair Nasser Alotaibi & Rawan Naif Alanazi
3.1.3	Sequence diagram	Rawan Naif Alanazi
3.2	Modelling the data	Nourah Salem Albarqi & Rawan Naif Alanazi & Nowair Nasser Alotaibi
3.2.1	ER diagram	Nowair Nasser Alotaibi
3.2.2	Data dictionary	Rawan Alanazi
3.3	Design interface	Nourah Albarqi
3.4	Drawing component diagram	Nowair Nasser Alotaibi
3.5	Find the alternative diagram	Nourah Albarqi & Nowair Alotaibi & Rawan Alanazi
4	Implementation	
4.1	Coding	Nourah Salem Albarqi & Nowair Nasser Alotaibi & Rawan Naif Alanazi
4.2	Test the system	Nourah Albarqi & Nowair Alotaibi & Rawan Alanazi
4.3	Deployment	Nourah Albarqi & Nowair Alotaibi & Rawan Alanazi
5	Documentation	
5.1	Write project proposal	Nourah Albarqi & Nowair Alotaibi & Rawan Alanazi
5.2	Write project conclusion	Nourah Albarqi & Nowair Alotaibi & Rawan Alanazi

Table 2: Team Member Responsibilities

Summary

In conclusion, the chapter preview the of existing systems which are very important to determine the features that should be considered in the "MCQ 4U" application. Also, the chapter discussed the method of gathering requirements for the project and the approach applied. Additionally, a project plan was presented that shows the primary tasks of the project with time duration for each task, and the responsibilities assigned to each member of the project team.

Chapter Three: Requirements Analysis

Introduction

This chapter explains the requirements gathering techniques used in the project. It discusses all requirements (functional and non-functional) that must be held in the system. To analyse the requirements for the "MCQ 4U" application the object-oriented approach is followed, so use-case diagram and use-case descriptions are used to describe the functional requirements of the system. Additionally, the user interfaces mock-up design for the application is represented at the end of this chapter.

3.1: Requirement Gathering Techniques

A system analyst is a key person in the analysis phase, who is the responsible for gathering the proposed project's requirements. The techniques used in this project to collect requirements are; a questionnaire, observation, and group sessions

Every technique has its advantages and disadvantages and may differ from one project to another, so multiple techniques are used to get the complete requirements set.

3.1.1: Questionnaire

A questionnaire was distributed among students and Faculty Members in the CCIS Imam University to get their requirements before developing the "MCQ 4U" application.

The team member analyzes from the questionnaire problems that's face the students such as students have difficulties in gathering information and do not have enough time to review the received lectures periodically. Faculty Members find difficulty to review the previous lecture and exam subjects before it is conducted.

Team member analyze that faculty members students prefer an education application in smartphones and tablets that help them in studying with some requirements.

From the questionnaire, there are some suggested features that students agree on it such as generating MCQs Answering the MCQs before attend each course's lecture. Add instant public communication feature to connect all the students in the same course. For faculty member, provide the control of choosing a number of students randomly to generate MCQs at the end of each lecture, provides Faculty Member with statistics about

students' understanding level after each course's lecture. See Miscellaneous in Appendix A and B.

3.1.2: Observation

Observation technique in gathering requirement depends on watching the users, the processes flow and the pain points [10].

There are two types of observation:

- Active observation: good for getting an understanding over an existing business process.
- Passive observation: usually provide better feedback [10].

A Faculty Members in Business Administration College in "Princess Nourah bint Abdulrahman University" [12] adopts the idea of "MCQ 4U" application, but they implement it manually without using a software application.

The Faculty Member selects five students in the class to ask random questions. Then, the selected students choose another student in the class to answer the questions. And after answering the questions, Faculty Member gets the answers and corrects the wrong answers.

If the Faculty Members implement the process that discussed in chapter two, they should have to correct all the students' answers to check their understanding level in the received lecture, which increases the overload of the Faculty Member. In addition, it is inefficient due to the short time for the lecture.

3.1.3: Group sessions

Is a process of gathering all information from users who are part of the "MCQ 4U" application? Those users are students, Faculty Members, and project team in CCIS Imam University. This method generates many random ideas that can be later evaluated and extract the Faculty Member and students' requirements that fit the project [10].

In this project two sessions were conducted, and students and Faculty Members suggestions were collected.

In the first session, the student suggestions are:

- Recording lectures in each course.
- Making groups in the application.
- Providing calendar for displaying the courses time
- Before the beginning of the new lesson, the Faculty Member should review the previous lesson and introduce the new one.

However, the Faculty Members suggested to link the students' email addresses with the application, so they can send notices to a particular student who was getting a low grade.

The second session included that the project team needs to evaluate each suggested idea with the agreed criteria such as effectiveness, feasibility, and cost, in order to prioritize the ideas according to these criteria [10].

As a result, from these sessions, the project team agreed on the following requirements:

- Before the beginning of the new lesson, the Faculty Member should review the previous lesson and introduce the new one.
- It is preferable if the students' email addresses were linked with the application, so the Faculty Member can send notices to a particular student who was getting a low grade.

3.2: Proposed Business Process

The project will depict the detailed steps that enhance the learning process and evaluate the Students understanding. The process initiated when the Faculty Member starts the lesson, then at the end; he/she will select five students randomly (one as a Coordinator Student and the Asker Students to generate MCQs).

The Faculty Member Build the MCQ Survey with a lecture name to select the Coordinator Student, Asker Students and determine the time deadline for them to generate MCQ.

The Coordinator Student will receive the MCQ Survey from the Faculty Member to divide the lecture into parts and informs the divided parts to Asker Students by sending a message via MCQ 4U Survey application.

The Asker Students will receive Coordinator Student message to write a single MCQ

for each Asker Student about the named lecture and send the MCQ to the Coordinator Student account. Then, the Coordinator Student reviews the questions to ensure that the Asker Students wrote MCQs depending on his dividing to the named lecture.

After that, the Coordinator Student will send the MCQ Survey that full with four MCQs to the Faculty Member to approve the MCQ Survey and publish it to Respondent Students.

Finally, the Respondent Students enter the MCQ Survey and answer the questions. Then, a statistical form generated according to Respondent Students answers, which then the Faculty Member can use to evaluate the students understanding and repeat the incomprehensible parts.

Figures 11 and 12 show the business process for the "MCQ 4U" application.

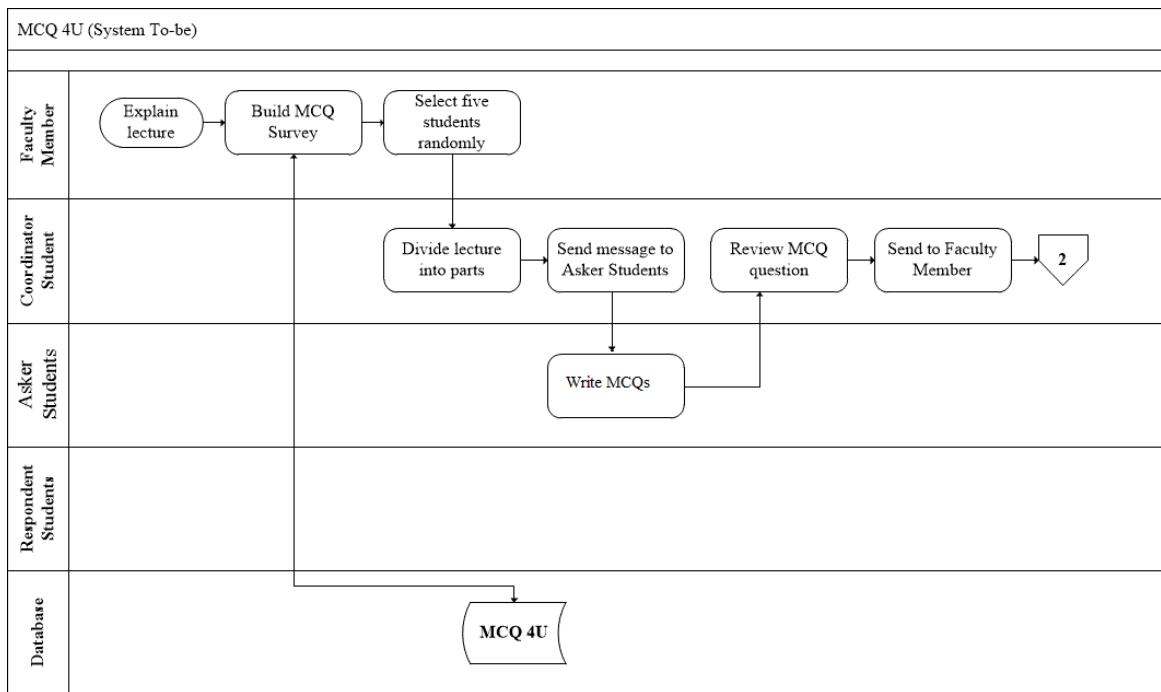


Figure 11: Business Process of MCQ 4U (System To-be)

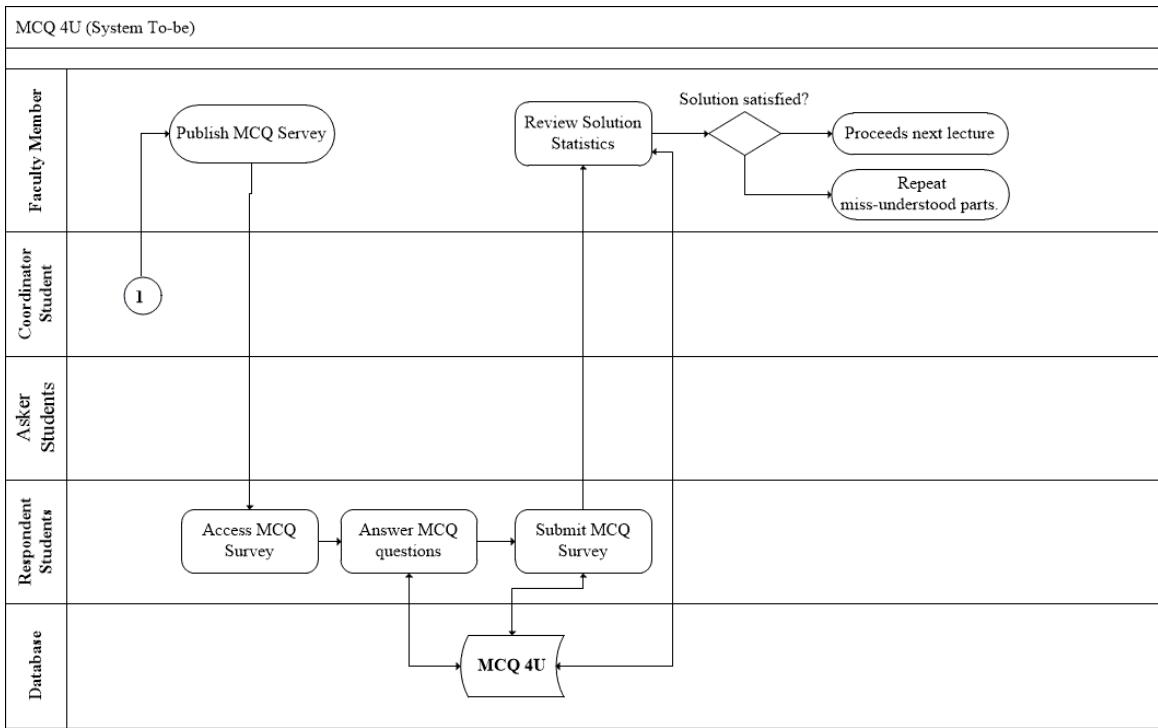


Figure 12: Continue Business Process of MCQ 4U (System To-be)

3.3: Functional Requirements

Use case diagram is an important diagram in UML model, usually called as a driver of all diagrams. Use case diagram determines the system functions or operations; it consists of important stakeholders who will interact with the application [13].

3.3.1: Admin

- The Admin should be able to create a course: the system shall be enabling the Admin to create a course by click on the manage course button, then click on add course button then filling course name, course ID, section number, number of lectures and select Faculty Member, then click create course button.
- The Admin should be able to link Faculty Member and Students to the course: the system shall be enabling the Admin to link Faculty Member while create course process by select faculty member name from a list and Students after course is created by go to link Students to courses page then determine course and Students name then click link button.
- The Admin should be able to delete Faculty Member and Student: the system shall be enabling the Admin to delete Faculty Member and Student by

opening the delete users page, then select student or faculty member and click on delete button.

- The Admin should be able to delete a course: the system shall be enabling the Admin to delete a course by open the manage course page, then, select course and click on delete button.

3.3.2: Faculty Member

- The Faculty Member should be able to register in the application: the system shall be enabling the Faculty Member to register by filling the first name, middle name, last name, id, and the password.
- The Faculty Member should be able to Log in: the system shall be enabling the Faculty Member to Log in by enter id, and password.
- The Faculty Member should be able to build MCQ Survey: the system shall be enabling the Faculty Member to build MCQ form by open the build MCQ page then filling lecture name, lecture number, determining deadline date, selecting four Asker Students and one student as a coordinator by selecting their name from the student list.
- The Faculty Member should be able to publish MCQ form: the system shall be enabling the Faculty Member to publish MCQ form by click on the approve MCQ button then views the MCQ form then click on publish button.
- The Faculty Member should be able to view solution statistics: the system shall be enabling the Faculty Member to view solution statistics by open the statistical MCQs page and select survey MCQs form.
- The Faculty Member should be able to logout: the system shall be enabling the Faculty Member to logout by click on the logout button.

3.3.3: Student

- The Student should be able to register: the system shall be enabling the Student to register by filling the first name, middle name, last name, id, and the password.
- The Student should be able to Log in: the system shall be enabling the Student to login by enter id, and password.

- The Student should be able to logout: the system shall be enabling the Student to logout by click on the logout button.

3.3.3.4: Coordinator Student

- The Coordinator Student should be able to send message to Asker Students: the system shall be enabling the Coordinator Student to send message to Asker Students by click on send message to Asker Student button then filling message that include specific lecture part for each Asker Student and click on send button.
- The Coordinator Student should be able to receive MCQ questions; the system shall be enable the Coordinator Student to receive MCQ questions and review survey MCQ form after the Asker Students wrote.
- The Coordinator Student should be able to finalize MCQ form: the system shall be enabling the Coordinator Student to finalize MCQ form by click on review MCQ button then review four questions with their answers by click on each question then click on send to Faculty Member button.
- The Coordinator Student should be able to delete questions: the system shall be enable the Coordinator Student to delete questions by click on delete button that inside question page.
- The Coordinator Student should be able to send the MCQ form to the Faculty Member: the system shall be enabling the Coordinator Student to send the MCQ form to the Faculty Member by click send to Faculty Member button.

3.3.3.5: Asker Student

- The Asker Student should be able to create MCQ question: the system shall be enabling the Asker Student to create MCQ question by open Asker Student page and click on create MCQ button then filling question, four answers and determine a right answer then click on send to Coordinator Student button.
- The Asker Student should be able to rewrite MCQ question: the system shall be enabling the Asker Student to rewrite MCQ question by send notification to Asker Student who question has been deleted by Coordinator Student and active create MCQ button.

3.3.3.6: Respondent Student

- The Respondent Student should be able to submit answer: the system shall be enabling the Respondent Student to submit answer by open the respondent page, click on the survey, and solve the survey then submit answers.

3.3.4: MCQ 4U Use case Daigram

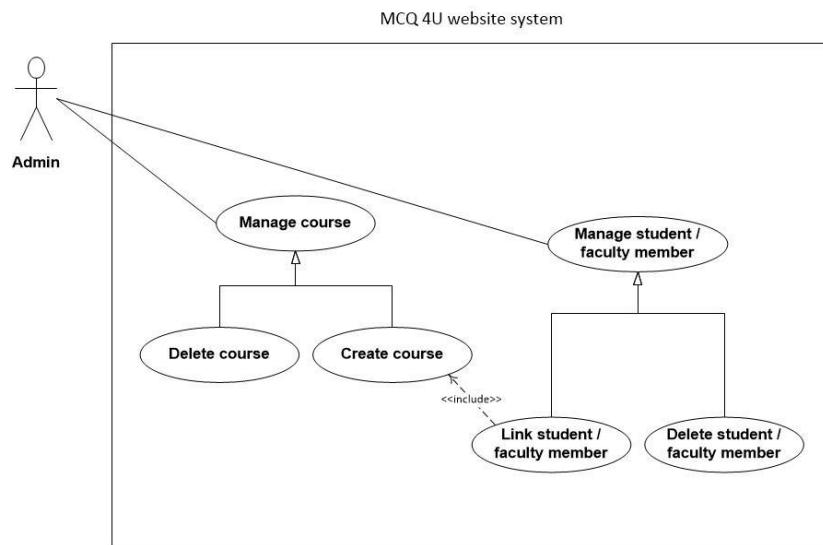


Figure 13: MCQ 4U Admin Use case diagram.

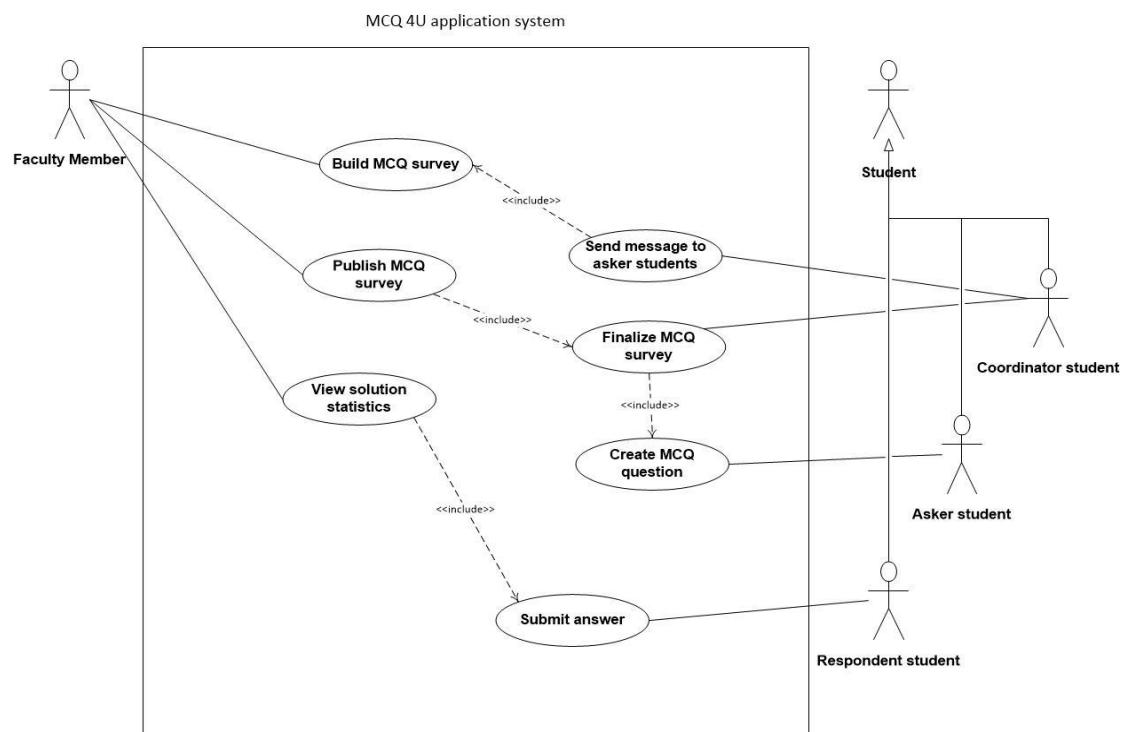


Figure 14: use case diagram of the MCQ 4U

3.3.5: MCQ 4U Use case Desceipton

3.3.5.1: Build MCQ Survey:

Use Case Name: Build MCQ Survey	ID: 1	Importance Level: high		
Primary actor: Faculty Member	Use Case Type: Detail, Essential.			
Stakeholders & interest:				
Faculty Member – wants to build The MCQ Survey.				
Brief Description:				
This use case describes how the Faculty Member Builds the MCQ Survey.				
Trigger: Faculty Member finished the explanation of the lecture.				
Type : Temporal.				
Relationship:				
Association: Faculty Member				
Include: Send message to Asker Students				
Extend:				
Generalization:				
Normal Flow of Event:				
1- The Faculty Member fills lecture name, number in the MCQ survey. 2- The Faculty Member determines deadline date for MCQ survey. 3- The Faculty Member selects four Asker Students and one Coordinator Student randomly. 4- The Faculty Member sends MCQ survey to Coordinator Student.				
Alternative /Exception Flow:				

Table 3: Build MCQs Survey UseCase description

3.3.5.2: Publish MCQ Survey

Use Case Name: Publish MCQ Survey	ID: 2	Importance Level: high
--	--------------	-------------------------------

Primary actor: Faculty Member	Use Case Type: Detail, Essential.
Stakeholders & interest: Faculty Member – wants to publish the MCQ survey.	
Brief Description: this use case describes how the Faculty Member Publish the MCQ survey.	
Trigger: Faculty Member receives completed MCQ survey from Coordinator Student.	
Type : External.	
Relationship:	
Association : Faculty Member	
Include :	
Extend :	
Generalization :	
Normal Flow of Event:	
1- The Faculty Member receives the MCQ survey from Coordinator Student. 2- The Faculty Member approves then publish the MCQ survey.	
Alternative /Exception Flow:	
2a: if the Coordinator Student did not send MCQ survey, the Faculty Member cannot execute publish process.	

Table 4: Publish MCQ Survey UseCase Description

3.3.5.3: View Solution Statistics

Use Case Name: View Solution Statistics	ID: 3	Importance Level: high		
Primary actor: Faculty Member	Use Case Type: Detail, Essential.			
Stakeholders & interest: Faculty Member – wants to view a statistical according to Respondent Student solution on the MCQ survey.				
Brief Description: This use case describes how Faculty Member views the statistic of Respondent Student answers.				
Trigger: The System views statistics during deadline.				
Type : External.				

Relationship:

Association : Faculty Member.

Include :

Extend :

Generalization :

Normal Flow of Event:

- 1- The Faculty Member can view statistics of MCQ survey after Respondent Student solves the MCQ survey and submits the answers.

Alternative /Exception Flow:

- 1a: if the Respondent Student did not solve the MCQ survey and submits the answers, the Faculty Member cannot view solution statistics.

Table 5: View Solution Statistics UseCase Description

3.3.5.4: Submit Answer

Use Case Name: Submit Answer	ID: 4	Importance Level: high		
Primary actor: Respondent Student	Use Case Type: Detail, Essential.			
Stakeholders & interest: Respondent Student – wants to answer all the questions on the MCQ survey.				
Brief Description: This use case describes how the Respondent Student will answer the MCQ survey.				
Trigger: Respondent Student receives published MCQ survey.				
Type : External.				
Relationship:				
Association : Respondent Student				
Include : View Solution Statistics				
Extend :				
Generalization :				

Normal Flow of Event:

- 1- The Respondent Student gets MCQ survey after the Faculty Member execute publish process.
- 2- The Respondent Student solves the MCQ survey and submits the answers.

Alternative /Exception Flow.

1a: if the Faculty Member did not publish the MCQ survey, the Respondent Student cannot solve the MCQ survey and submits the answers.

Table 6: Submit answer UseCase Description

3.3.5.5: Create MCQ Question

Use Case Name: Create MCQ Question	ID: 5	Importance Level: high		
Primary actor: Asker Student	Use Case Type: Detail, Essential.			
Stakeholders & interest: Asker Student – wants to write the question of the MCQ survey.				
Brief Description: This use case describes how the Asker Student write the question of the MCQ survey.				
Trigger: Asker Students receives lecture parts from Coordinator Student.				
Type : External.				
Relationship: Association : Asker Student Include : Finalize MCQ survey Extend : Generalization :				
Normal Flow of Event: 1- The Asker Students receives the lecture parts that assigned for each one from Coordinator Student. 2- Each Asker Student fills a question and four answers depending on specific part of lecture. 3- The Asker Student determines the right answer then sends to Coordinator Student.				

Alternative /Exception Flow.

1a: if the Coordinator Student did not send the lecture parts, the Asker Students cannot write the MCQ survey questions.

Table 7: Create MCQ Question UseCase Description

3.3.5.6: Send Message to Asker Students

Use Case Name: Send Message to Asker Students	ID: 6	Importance Level: high		
Primary actor: Coordinator Student	Use Case Type: Detail, Essential.			
Stakeholders & interest: Coordinator Student – wants to send message to Asker Students.				
Brief Description: This use case describes how the Coordinator Student send message to Asker Students.				
Trigger: Coordinator Student receives MCQ survey from Faculty Member.				
Type : External.				
Relationship: Association : Coordinator Student. Include : Extend : Generalization :				
Normal Flow of Event: 1- The Coordinator Student writes message include lecture parts that divided on each Asker Student. 2- The Coordinator Student sends message to Asker Students.				
Alternative /Exception Flow: 1a: if the Faculty Member did not build the MCQ survey, the Coordinator Student cannot send message to Asker Students.				

Table 8: Send Message to Asker Students UseCase Description

3.3.5.7 Finalize MCQ Survey:

Use Case Name: Finalize MCQ Survey	ID: 7	Importance Level: high
---	--------------	-------------------------------

Primary actor: Coordinator Student	Use Case Type: Detail, Essential.
Stakeholders & interest: Coordinator Student – wants to finalize MCQ survey.	
Brief Description: This use case describes how the Coordinator Student finalize MCQ survey.	
Trigger: Coordinator Student receives MCQ survey questions from Asker Students.	
Type : External.	
Relationship:	
Association : Coordinator Student.	
Include : Publish MCQ survey	
Extend :	
Generalization :	
Normal Flow of Event:	
<ol style="list-style-type: none"> 1- The Coordinator Student receives questions from Asker Students. 2- The Coordinator Student reviews questions. 3- The Coordinator Student sends completed MCQ survey to Faculty Member. 	
Alternative /Exception Flow:	
<p>1a: if the Asker Students did not send the MCQ survey questions, the Coordinator Student cannot finalize the MCQ survey.</p> <p>1b: The Coordinator Student deletes questions if that are written outside the assigned lecture parts for each Asker Student.</p> <p>2b: The system allows Asker Students to rewrite questions if that are deleted and send to coordinator again.</p>	

Table 9: Finalize MCQ Form UseCase Description

3.3.5.8: Create Course

Use Case Name: Create Course	ID: 8	Importance Level: high		
Primary actor: Admin	Use Case Type: Detail, Essential.			
Stakeholders & interest: Admin – wants to create courses for Faculty Member.				
Brief Description: This use case describes how the Admin create course.				

Trigger: Admin needs to create course.

Type : Temporal.

Relationship:

Association :

Include : Link Faculty Member and student

Extend :

Generalization : Manage Course

Normal Flow of Event:

- 1- The Admin fills course name, id, section number, number of lectures.
- 2- The Admin selects a Faculty Member then create course.

Alternative /Exception Flow:

1a: if the Faculty Member did not register, the Admin cannot create course.

Table 10: Create Course UseCase Description

3.3.5.9: Delete Course

Use Case Name: Delete Course	ID: 9	Importance Level: high		
Primary actor: Admin	Use Case Type: Detail, Essential.			
Stakeholders & interest: Admin – wants to delete course.				
Brief Description: This use case describes how the Admin delete course.				
Trigger: Admin needs to delete course.				
Type : Temporal.				
Relationship:				
Association :				
Include :				
Extend :				
Generalization : Manage Course				

Normal Flow of Event:

- 1- The Admin deletes course by click on delete button.

Alternative /Exception Flow:

Table 11: Delete Course UseCase Description

3.3.5.10: Link Faculty Member and student

Use Case Name: Link Faculty Member and student	ID: 10	Importance Level: high		
Primary actor: Admin	Use Case Type: Detail, Essential.			
Stakeholders & interest: Admin – wants to Link the Faculty Member and student.				
Brief Description: This use case describes how the Admin Link the Faculty Member and student.				
Trigger: Admin needs to Link Faculty Member and Student.				
Type : Temporal.				
Relationship: Association : Include : Extend : Generalization : Manage Faculty Member and student				
Normal Flow of Event: 1- The Admin links the Faculty Member during create course process and student after course is created.				
Alternative /Exception Flow: 1a: if the Admin did not create the course, the link process of Faculty Member and student cannot be executed.				

Table 12: Link Faculty Member and Student UseCase Description

3.3.5.11: Delete Faculty Member and Student

Use Case Name: Delete Faculty Member and Student	ID: 11	Importance Level: high		
Primary actor: Admin	Use Case Type: Detail, Essential.			
Stakeholders & interest: Admin – wants to delete the Faculty Member and student.				
Brief Description: This use case describes how the Admin delete the Faculty Member and student.				
Trigger: Admin needs to delete Faculty Member and Student.				
Type : Temporal.				
Relationship: Association : Include : Extend : Generalization : Manage Faculty Member and student				
Normal Flow of Event: 1- The Admin determines Faculty Member or student then delete from system.				
Alternative /Exception Flow: 1a: if the Faculty Member and student are not registered, the Admin cannot execute delete process.				

Table 13: Delete Faculty Member and Student UseCase Description

3.4: Non-functional Requirements

This section is describe variety criteria regarding the performance, security, usability and reliability of the system.

3.4.1: Related to performance or usability

3.4.1.1: Usability	<ul style="list-style-type: none"> • The system must support an easy interaction interface with the users. • The system should be use a clear terminology and simple words. • The system should provide user with clear instructions in case of the user makes an error.
3.4.1.2: Reliability	<ul style="list-style-type: none"> • The system must be available to all users (24/7). • The system should provide accurate statistics. • The system should perform all functions correctly. • The system should update any change that occur on the data.
3.4.1.3: Performance	<ul style="list-style-type: none"> • The system should connect the user to server in one second. • The system should response the user in one second. • The system should update the data in one second. • The system should connect the user to server in one second. • The system should provide all the user's functions when user login. • The system should be able to manage all the information incoming from the database
3.4.1.4: Security	<ul style="list-style-type: none"> • The system must block user account five minutes when the id or password enter wrong three times. • The system must force the user to make a complex password. • The system must provide a backup and restore strategy.

Table 14: Non-functional requirements

3.4.2: Related to non-technical issues

3.4.2.1: Response time

Response time is considered an important issue during using the MCQ 4U application because it represents the amount of time needed to respond to a user request. The response time may be considered as the combination of the service time and the wait time. The service time is a time that the user takes to do a request while the wait time is considered as the time in which the request had to wait in a queue. So, project team focused on response time to be fast in a few seconds.

3.5: User Interfaces

3.5.1: Registration screen

As demonstrated in Figure 15, the Faculty Member and Student can create an account to use the MCQ 4U.

The registration form consists of several input fields: First name, Middle name, Last name, ID, and Password. Below these are two radio buttons for User Type: Student and Faculty Member. At the bottom is a purple "Sign Up" button.

Figure 15: MCQ 4U Registration form

3.5.2: Log in Screen

As demonstrated in Figure 16, the Faculty Member and Student will enter the id and password, to access MCQ 4U.

The log in form has two input fields: ID and Password. Below them is a purple "Login" button.

Figure 16: MCQ 4U Log in form

3.5.5: Home Interface

As demonstrated in Figure 17, the Faculty Member and Student will choose the course to make specific task.

The interface features a sidebar on the left containing course codes: IS200, IS397, IS396, and IS441. The main area is currently empty.

Figure 17: MCQ 4U First interface

3.5.4: Admin

3.5.4.1: Create Courses

As demonstrated in Figure 18, Admin will create all courses that are used in the application.

The create course form includes fields for Course ID (IS397), Course Name (Data Mining), and Faculty Member (T Manal). A purple "Create Course" button is at the bottom.

Figure 18: Create course

3.5.4.2: Link Faculty member and Students to Course

As demonstrated in Figure 19 and 20, the Admin will link the Faculty Member and Student to the course by course and Faculty Member or Student ID.

Course ID :
Email Address
student ID :
Email Address
Link

Figure 19: Link students to course

Course ID :
Email Address
Faculty member ID :
Email Address
Link

Figure 20: link Faculty member to course

3.5.5: Faculty Member

3.5.5.1: Selecting the task

As demonstrated in Figure 21, Faculty Member will choose the task choose students or show statistics.

MCQ 4U
IS39026987
IS200
Choose students
Show statistics

Figure 21: Faculty member selecting task

3.5.5.2: Selecting Five Students

As demonstrated in Figure 22, where Faculty Member select the "Choose Students" option, the Figure 33 show how Faculty member will select Five students (Coordinator and Asker Student).

MCQ 4U
IS397 T. Manal :
▼ Students ▼ Four student ▼ coordinator

Students	Four student	coordinator
Nourah	Nawal	<input checked="" type="checkbox"/> Rawan <input checked="" type="checkbox"/>
Rawan	Sarah	<input checked="" type="checkbox"/>
Nowair	Mona	<input checked="" type="checkbox"/>
sarah	Hala	<input checked="" type="checkbox"/>

Save

Figure 22: Select Students Form

3.5.5.3: Select Statistics

As demonstrated in Figure 23, Faculty Member will choose a specific survey to display statistic.

Figure 23: Select statistics Form

3.5.5.4: Answer Statistics Form

As demonstrated in Figure 24, Faculty Member will display the statistic of specific survey.

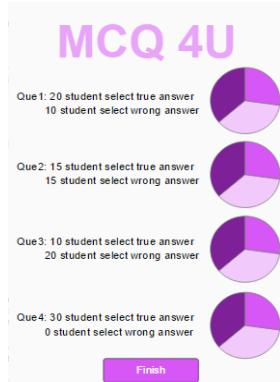


Figure 24: Answer Statistics Form

3.5.6: Students:

3.5.6.2: Selecting task

Figure 25 shows the interface of Student tasks.

Figure 25: Answer Statistics Form

3.5.6.3: Create Questions

Figure 26 shows how the Asker enters the questions.

Figure 26: Lecture Questions form

3.5.6.3: MCQ Survey

Figure 27 shows the final MCQ survey that displays the four questions that written by Asker Students.

MCQ 4U

MCQ title :
Que1 : question description ?
 Choice1 Choice2
Que2 : question description ?
 Choice1 Choice2
Que3 : question description ?
 Choice1 Choice2
Que4 : question description ?
 Choice1 Choice2

Submit

Figure 27: MCQ Form

Table 15: User interfaces

Summary

In conclusion, gathering requirements from students and faculty member is very important to elicit the requirements that they feel are most appropriate to provide in "MCQ 4U" application.

Analyze the "MCQ 4U" application requirements is a critical phase for the design and implementation phases to reduce errors that can will be costly when building the application.

The next chapter is supposed to provide "MCQ 4U" application design that includes the system modeling, data modeling, "MCQ 4U" application component diagram and alternatives design.

Chapter Four: System Design

Introduction

This chapter reflects the modeling of system relying on the UML, which have a key role in the programming the application process in the implementation phase.

4.1: System Modelling

This section views the database structure that used in the application, through database entity relationship diagram first and then the data dictionary table.

4.1.1: Activity diagrams

Activity diagram shows the dynamic aspects of the system, represented as a flowchart viewing the flow between activities which represent the operations of the system.

The flow of control is drawn from one operation to another. the flow types are sequential, branched, or concurrent. the diagram uses a different element such as fork and join to deal with all type of flow controls [14].

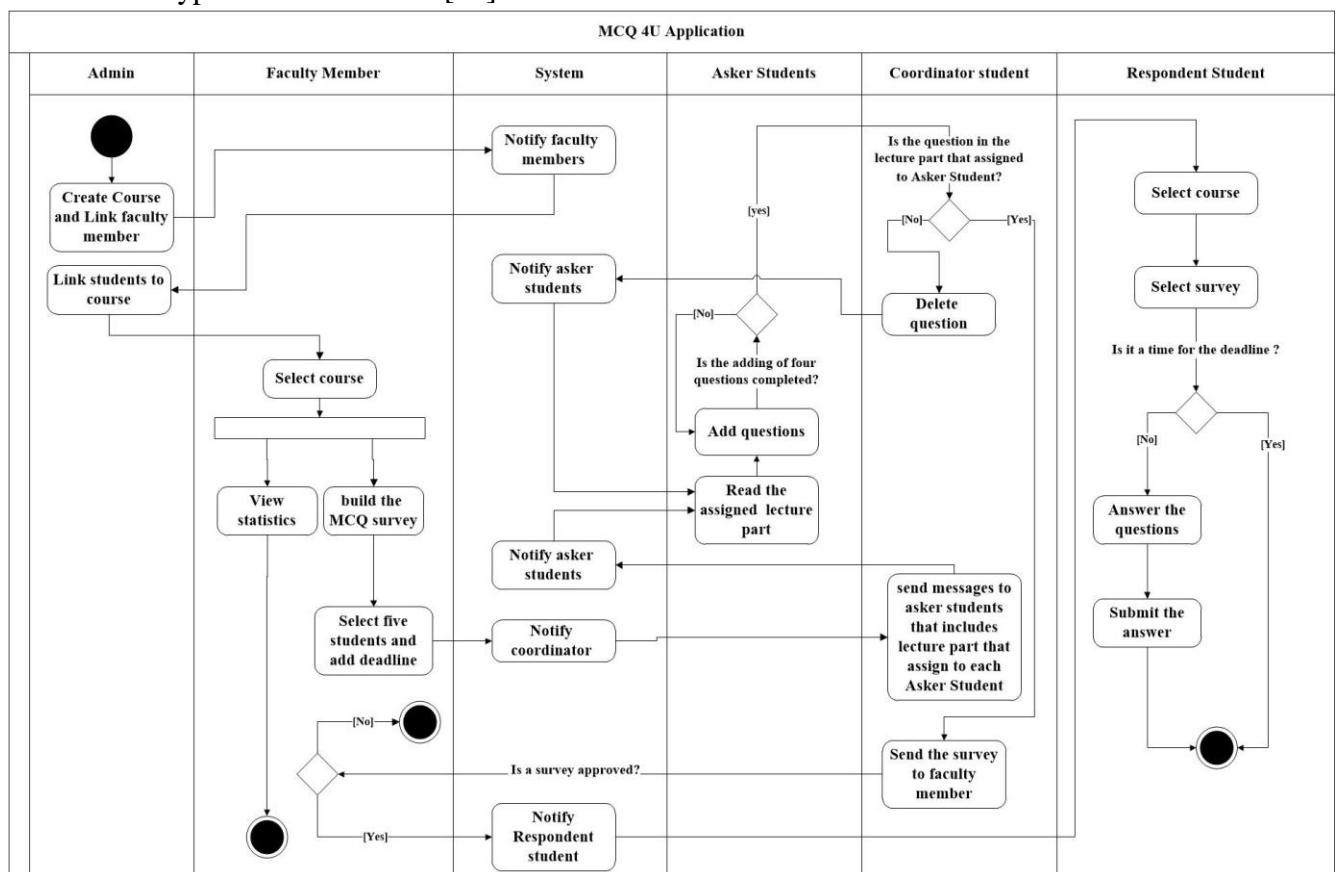


Figure 28: Activity diagram for all MCQ use cases.

4.1.1.1: Build MCQ Survey

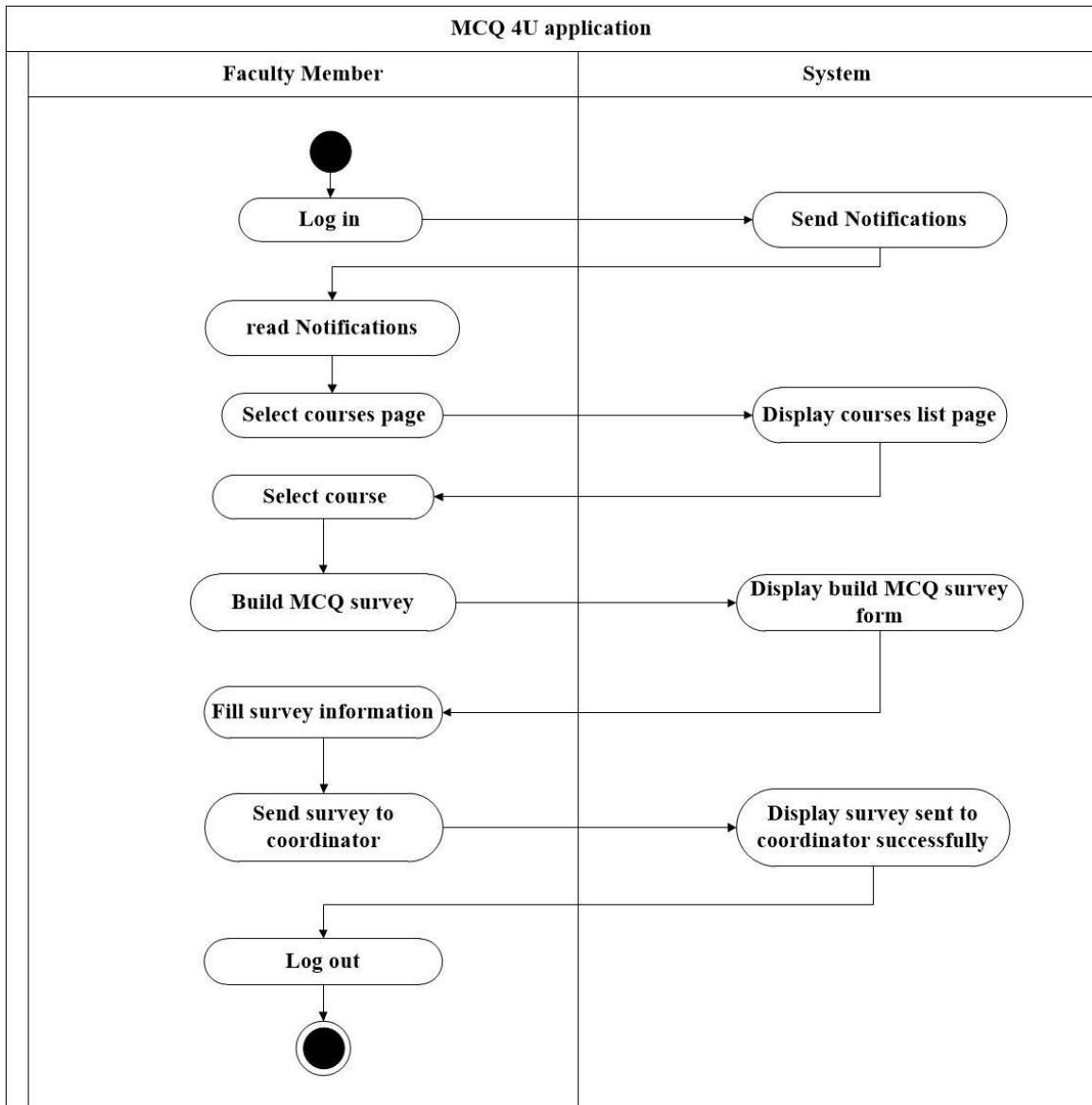


Figure 29: Build MCQ survey Activity diagram.

4.1.1.2 Publish MCQ Survey

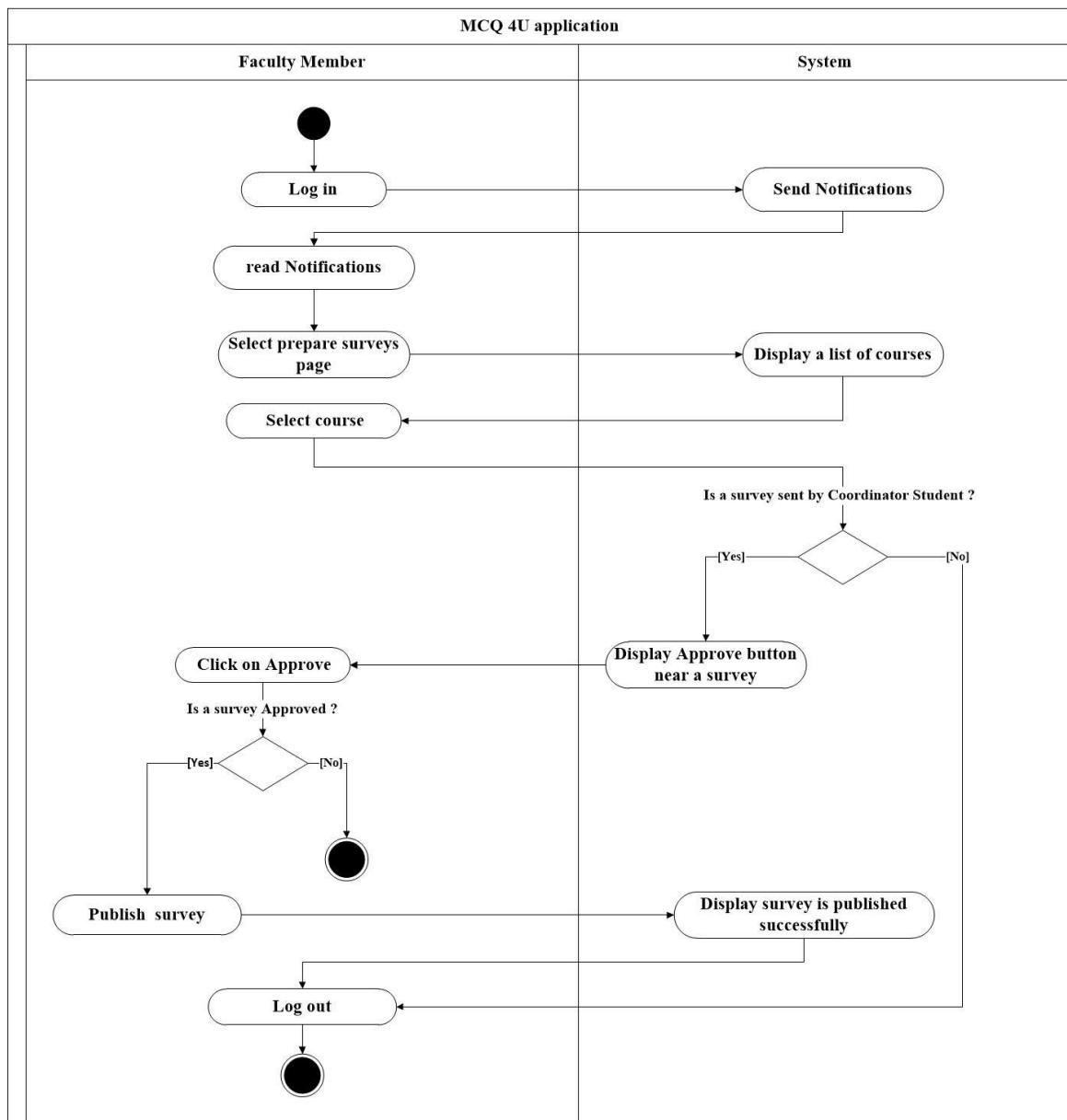


Figure 30: Publish MCQ survey Activity diagram.

4.1.1.3 View Solution Statistics

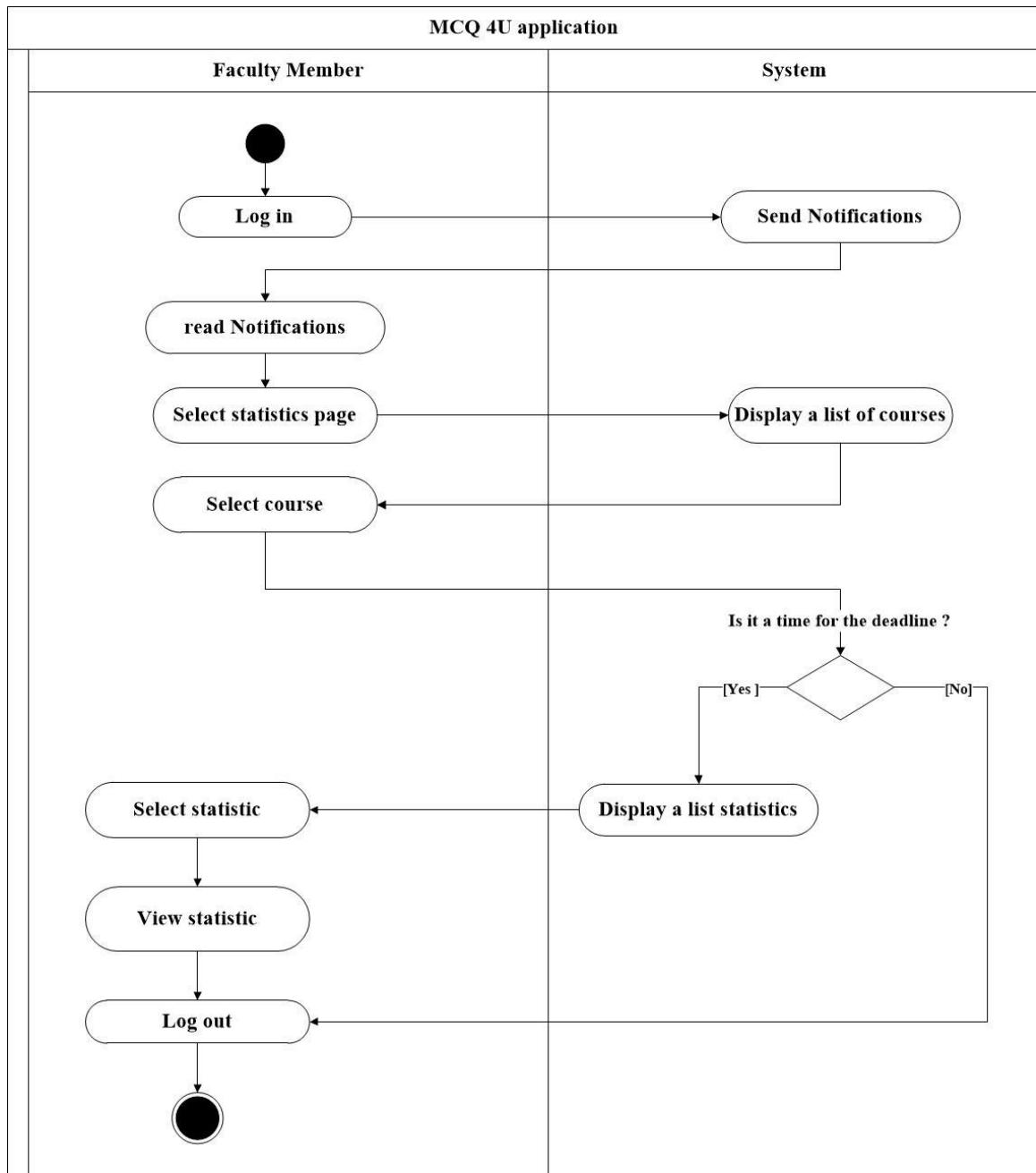


Figure 31: View Solution Statistics Activity diagram.

4.1.1.4 Submit Answer

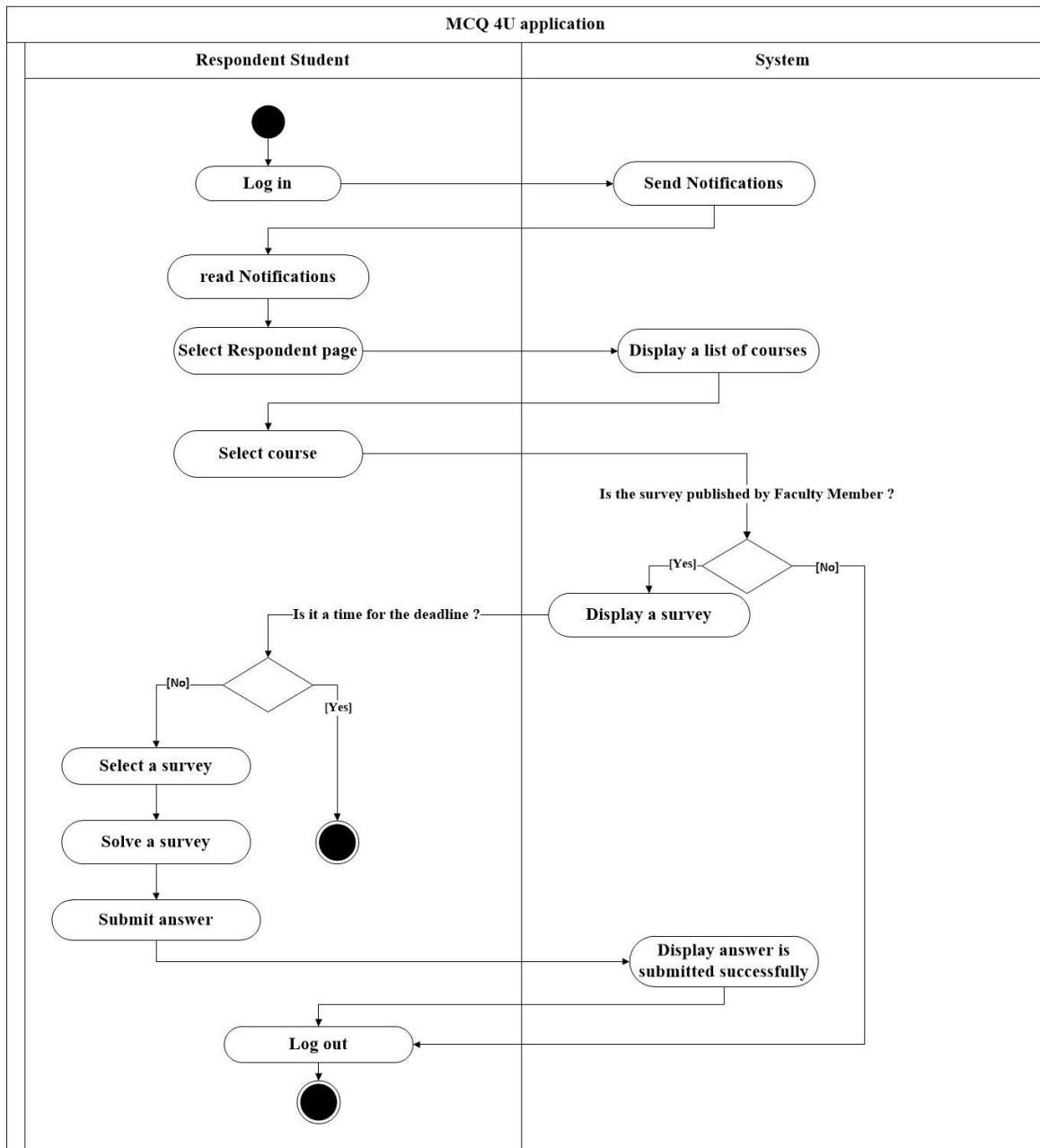


Figure 32: Submit Answer Activity diagram.

4.1.1.5: Create MCQ Questions

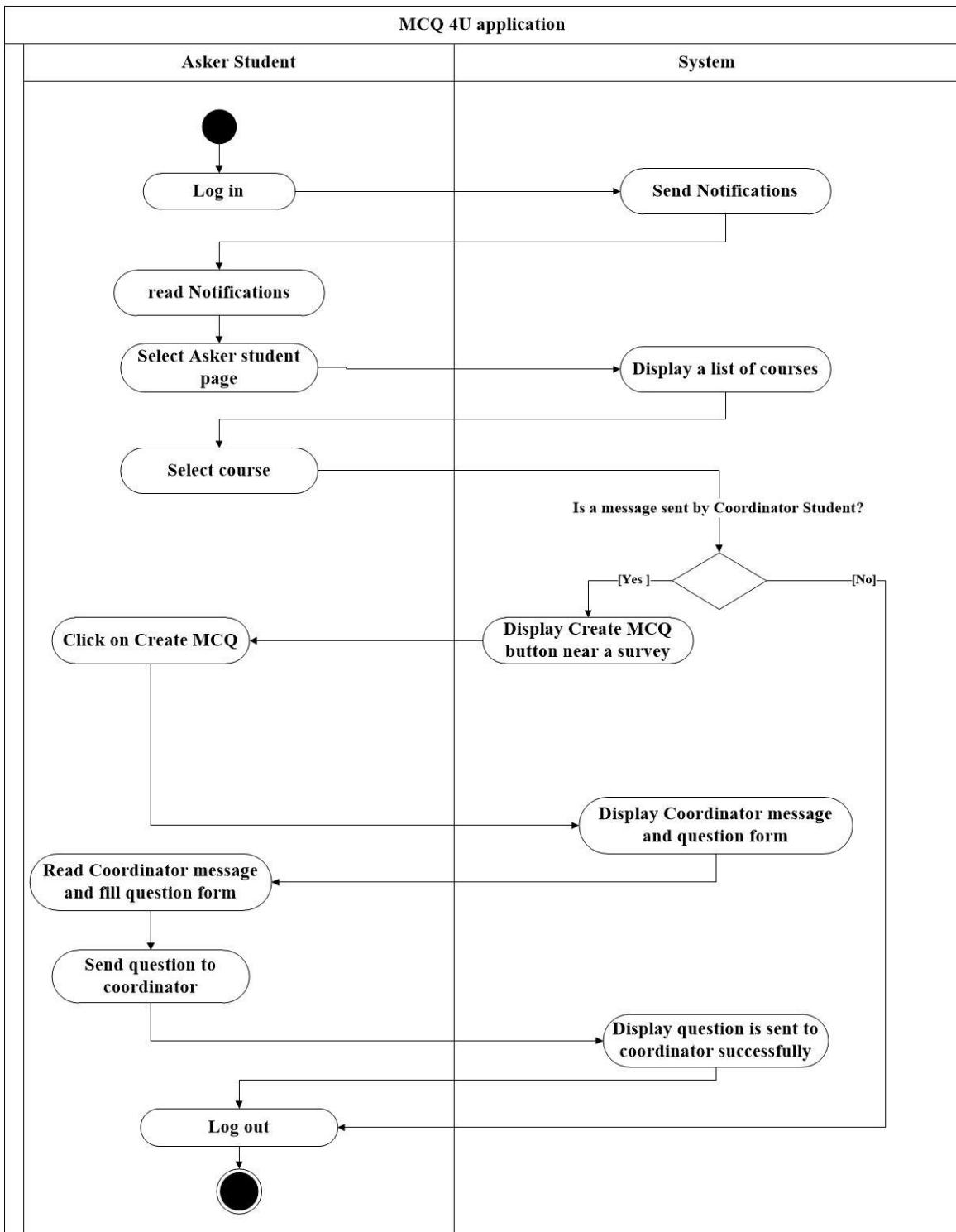


Figure 33: Create MCQ Questions Activity diagram

4.1.1.6: Send Message to Asker Students

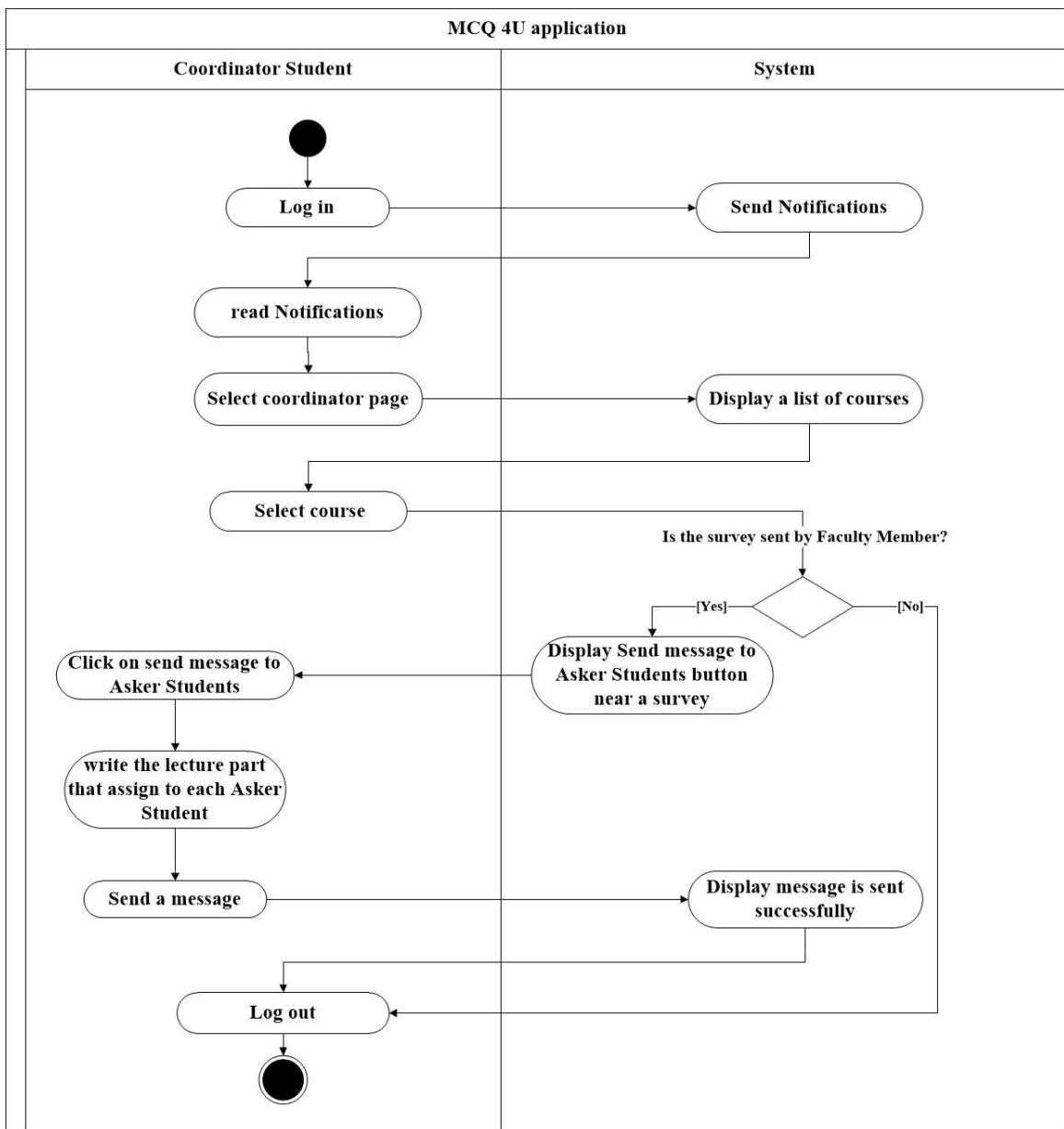


Figure 34: Send message to Asker Students Activity diagram

4.1.1.7 Finalize MCQ Survey

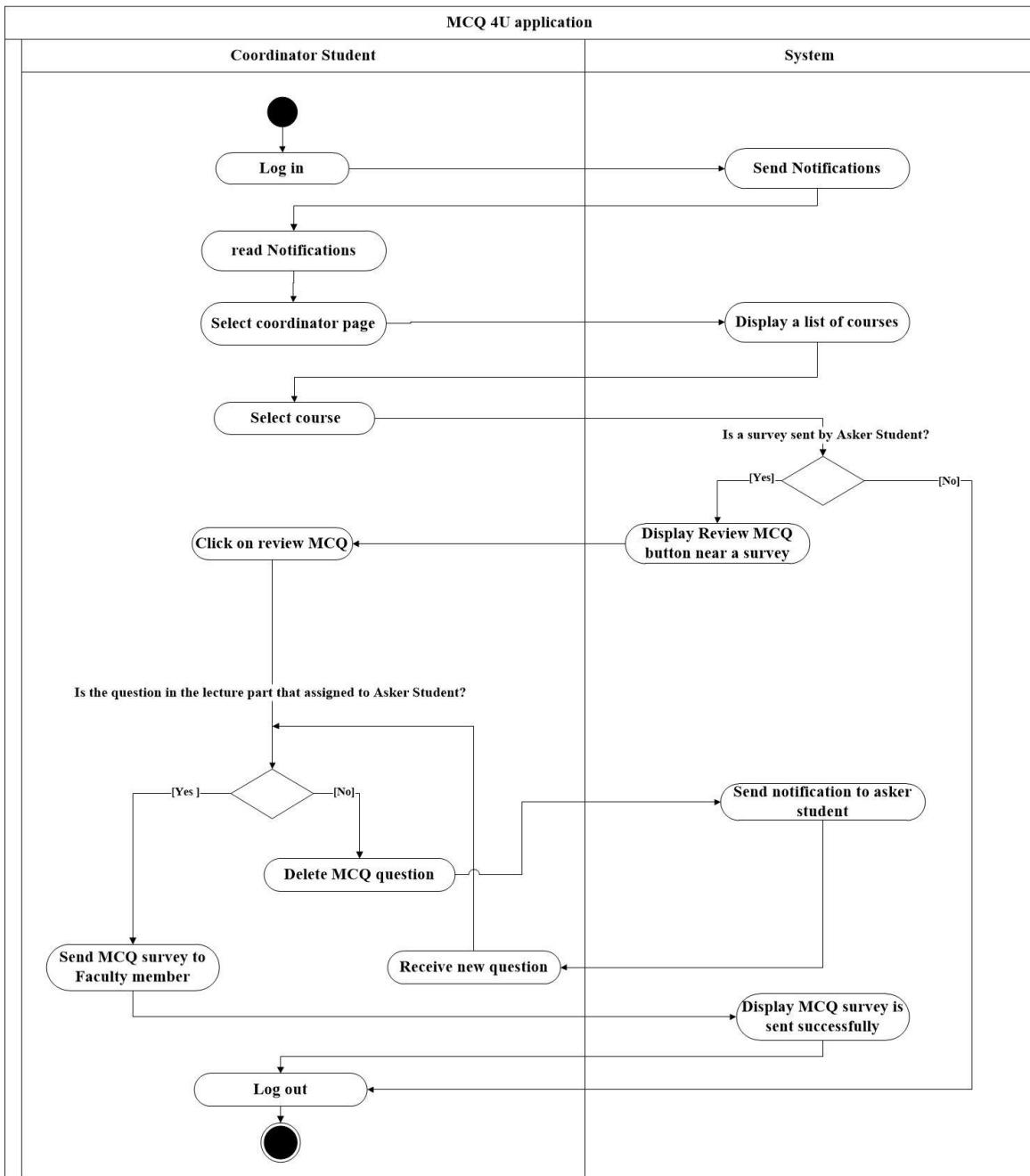


Figure 35: Finalize MCQ Survey Activity diagram

4.1.1.8: Create Course and Link Faculty Member

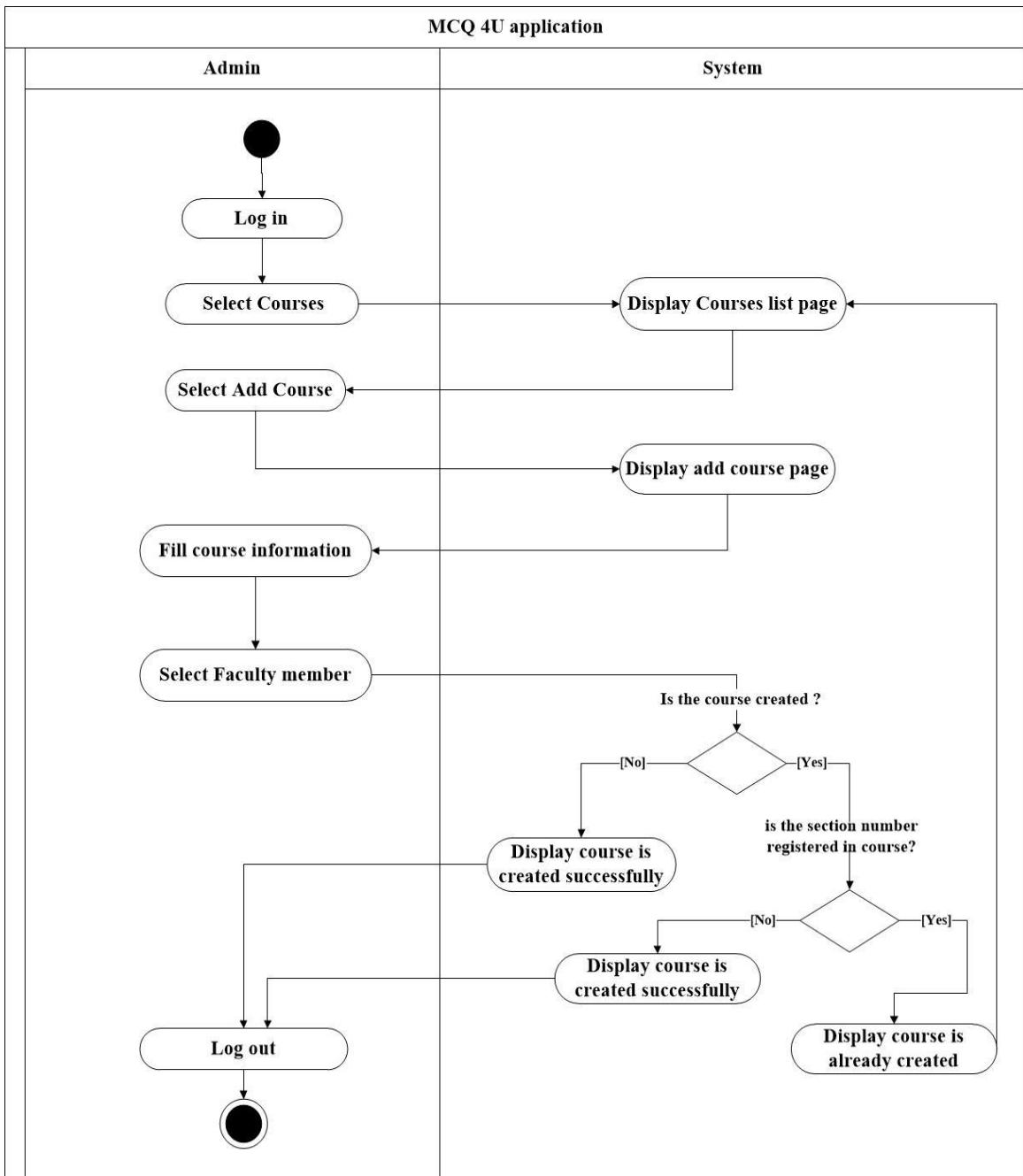


Figure 36: Create course and Link Faculty Member Activity diagram

4.1.1.9: Delete course

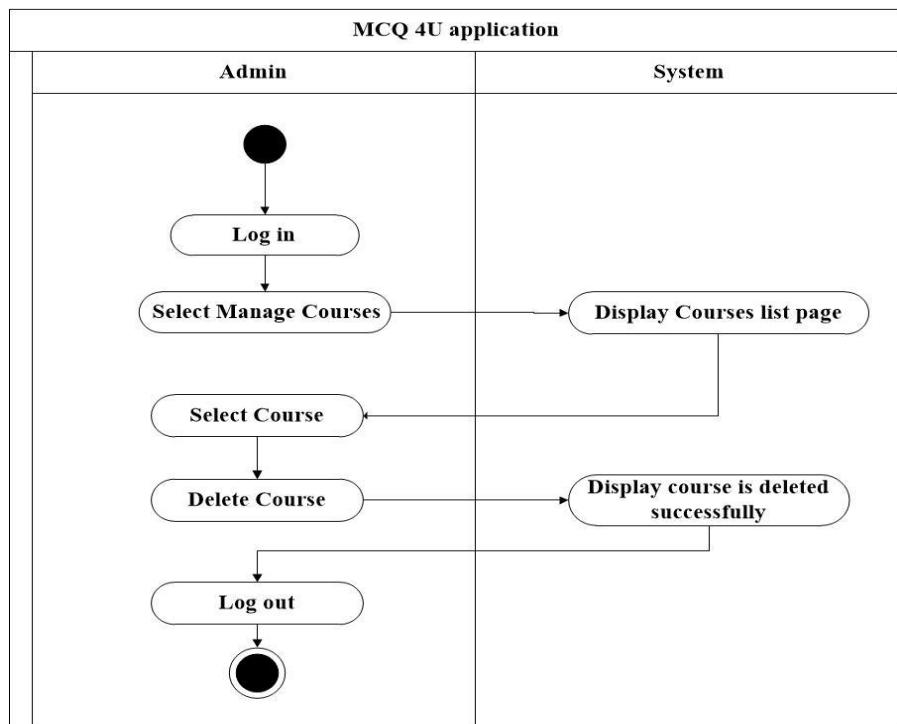


Figure 37: Delete course Activity diagram

4.1.1.10: Link Student

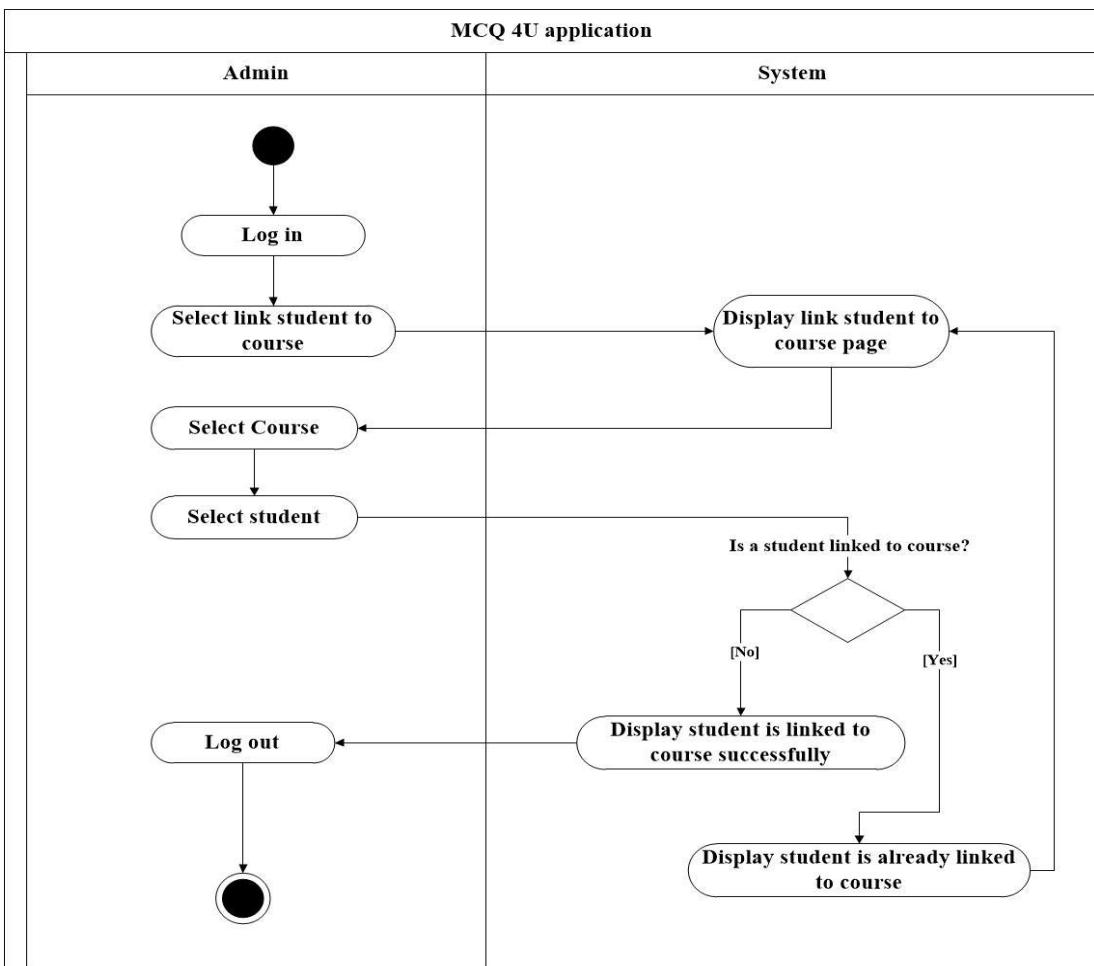


Figure 38: Link student Activity diagram

4.1.1.11: Delete Faculty Member and Student

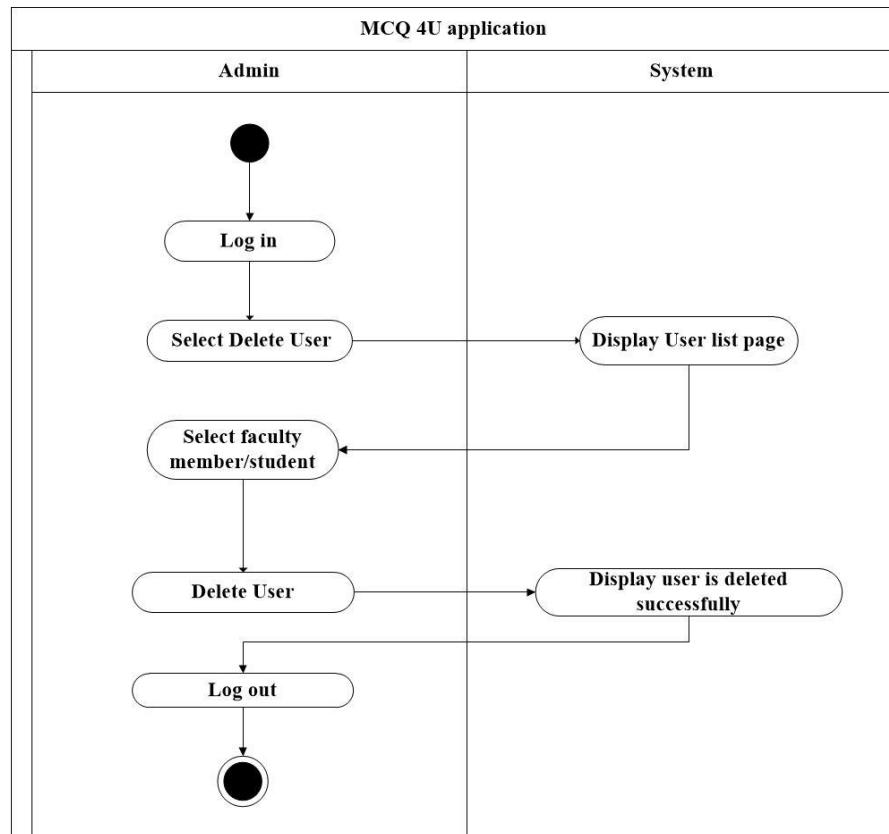


Figure 39: Delete Faculty Member and student Activity diagram

4.1.2: Class Diagram

The class diagram reflects how entities on its different types like people, things, or data related to each other, it also shows the properties and functions of each class, which reflects the structures of the system [15].

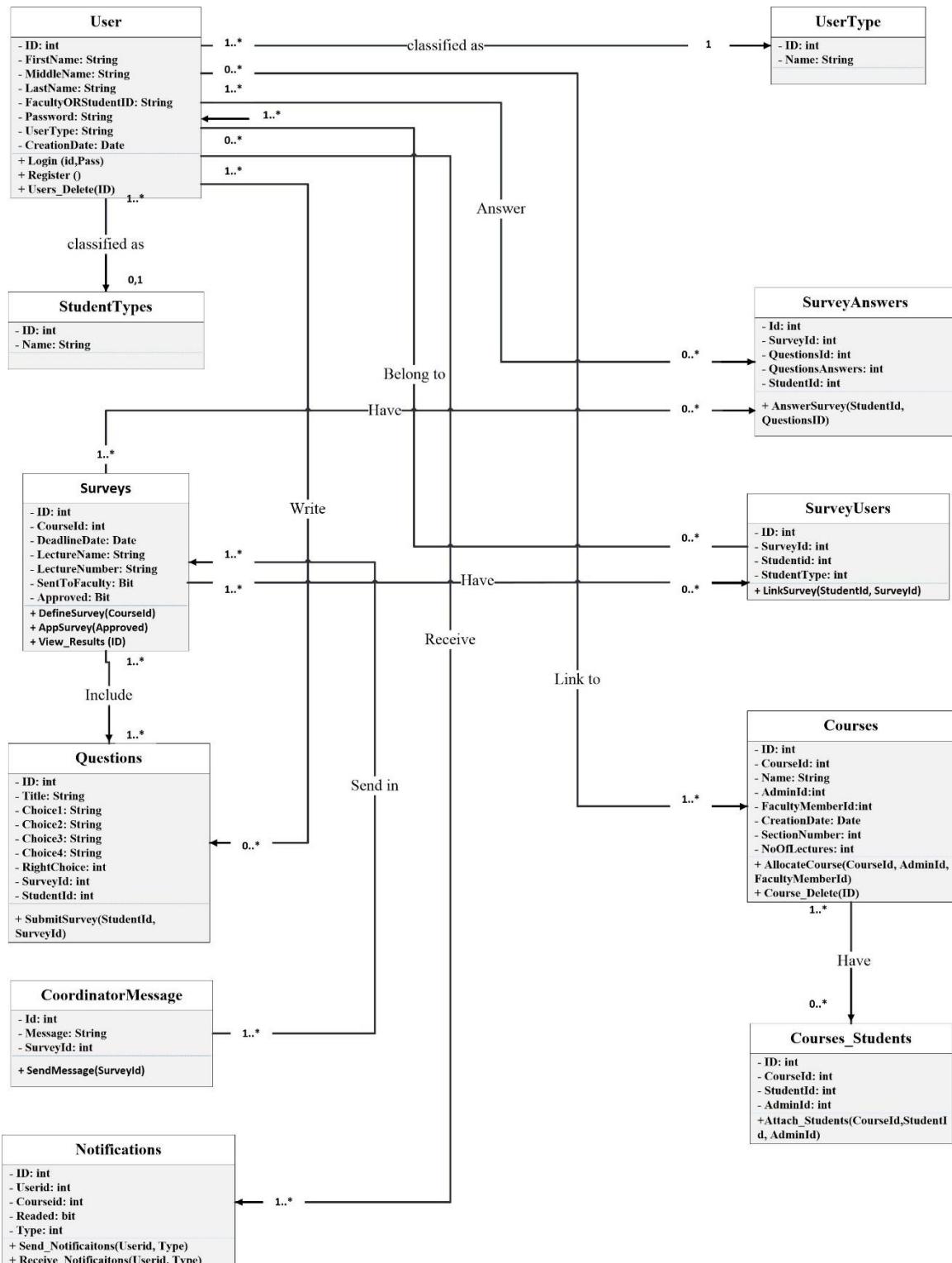


Figure 40: Class diagram for MCQ 4U

4.1.3: Sequence Diagrams

Sequence diagram reflects the sequence of action steps for each main procedure in the system, according to the timeline, the messages and the response from each side of participation in the system [16].

4.1.3.1: Build MCQ survey

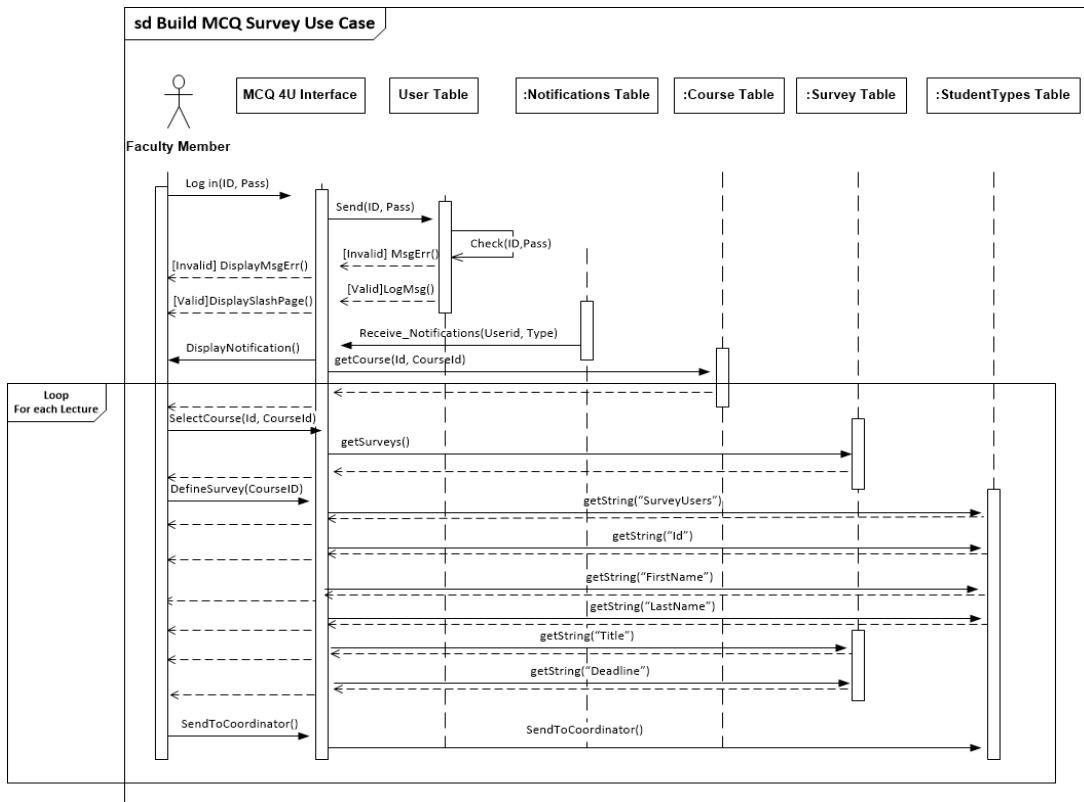


Figure 41: Build MCQ survey sequence diagram

4.1.3.2: Publish MCQ survey

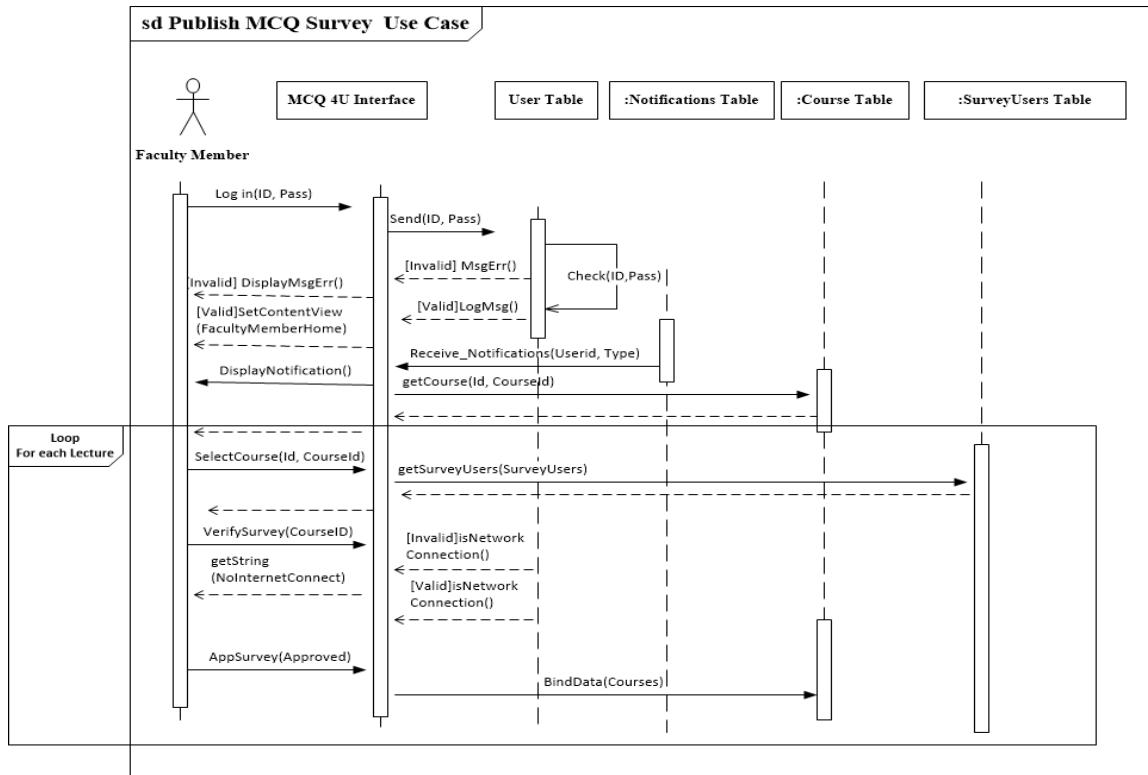


Figure 42: Publish MCQ survey sequence diagram

4.1.3.3: Submit Answer

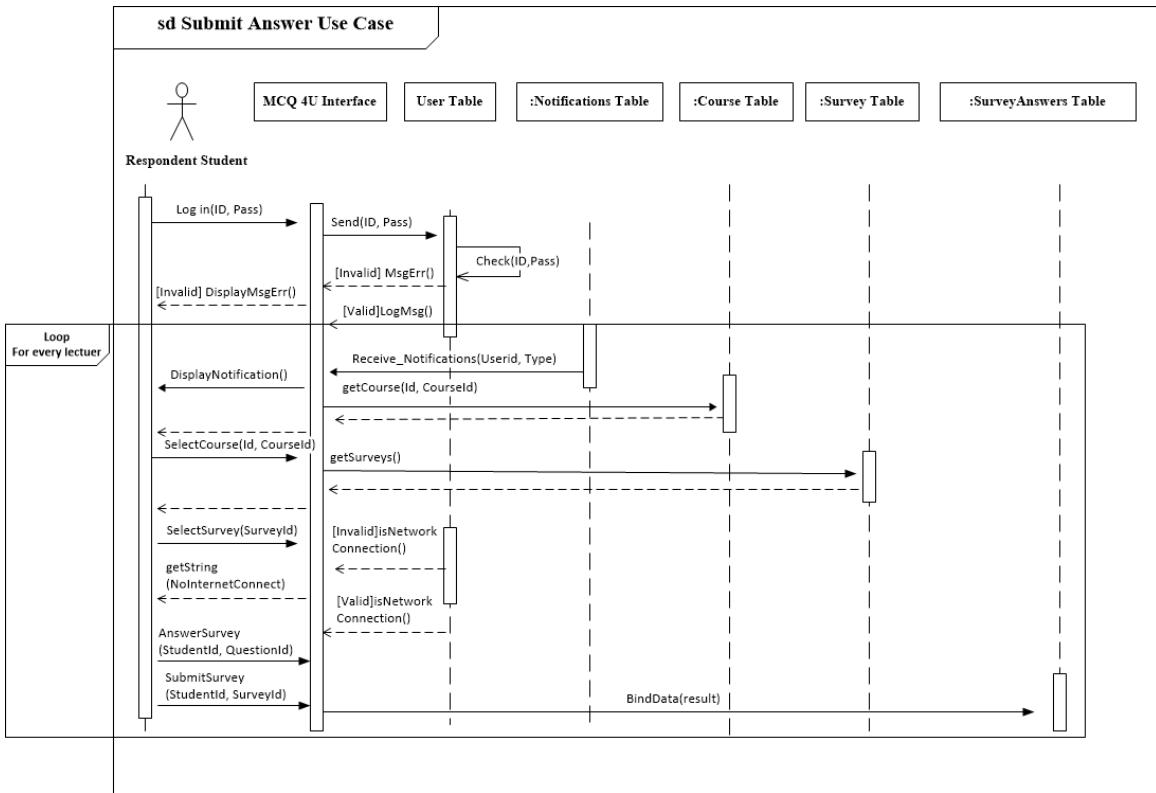


Figure 43: Submit Answer sequence diagram

4.1.3.4: View solution statistics

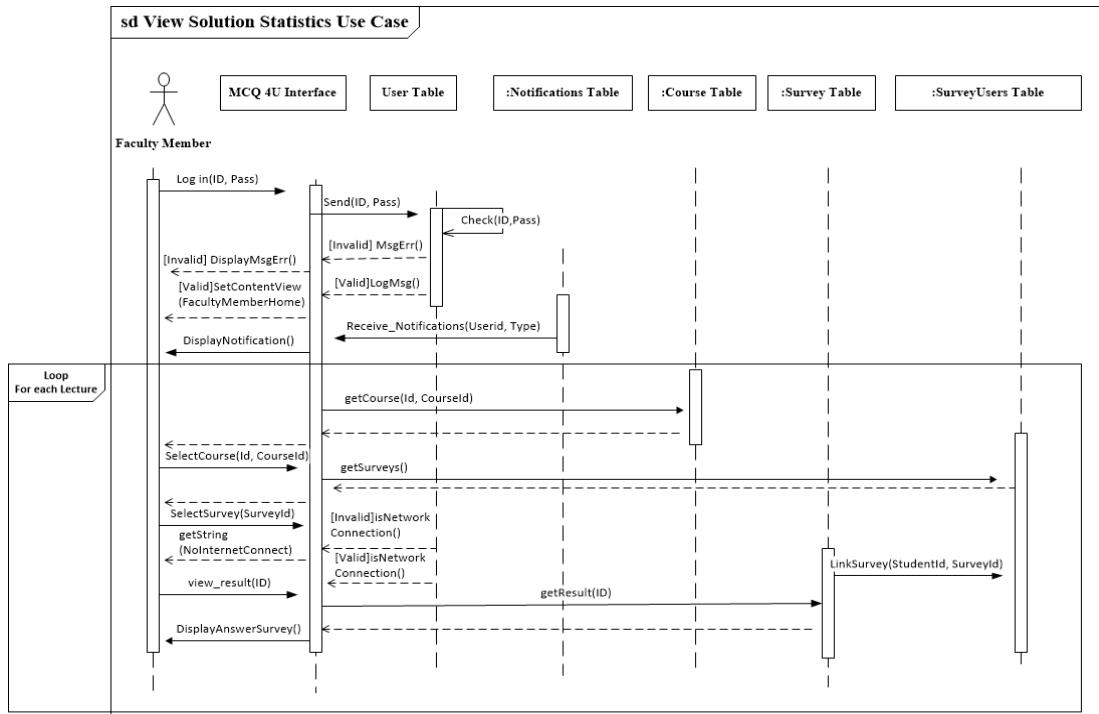


Figure 44: view solution statistics sequence diagram

4.1.3.5: Submit Answer

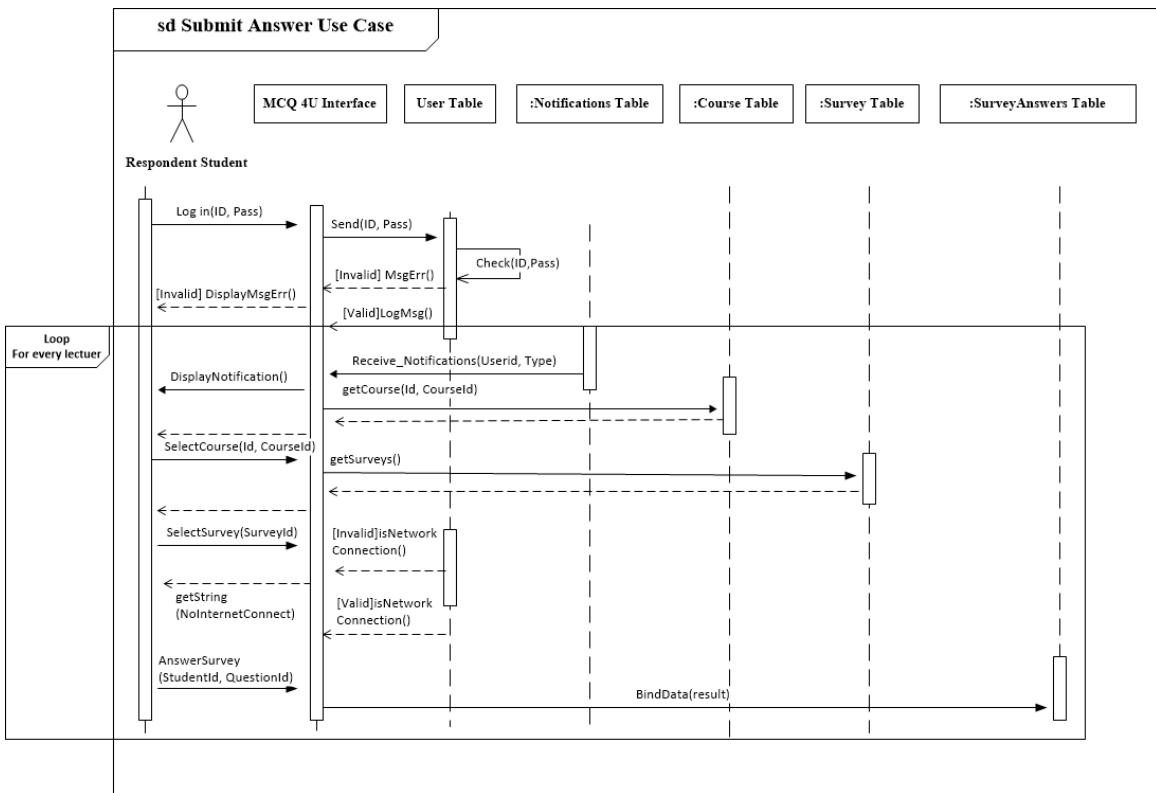


Figure 45: Submit Answer sequence diagram

4.1.3.6 Send Message to Asker Students:

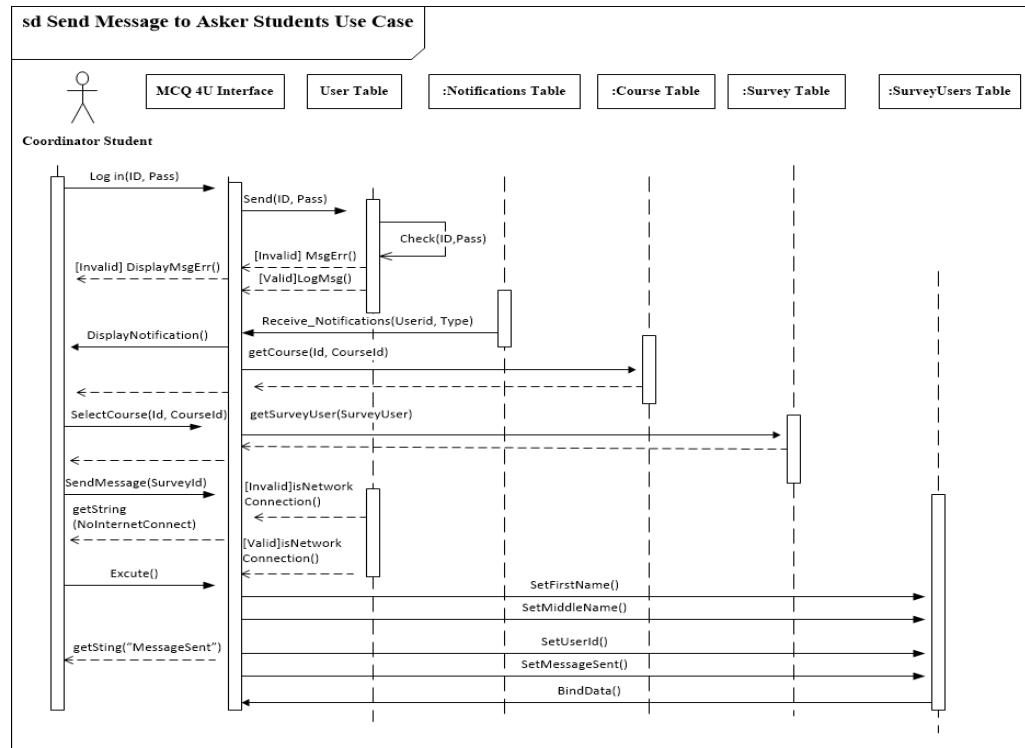


Figure 46: Send Message to Asker Students sequence diagram

4.1.3.7: Finalize MCQ Survey

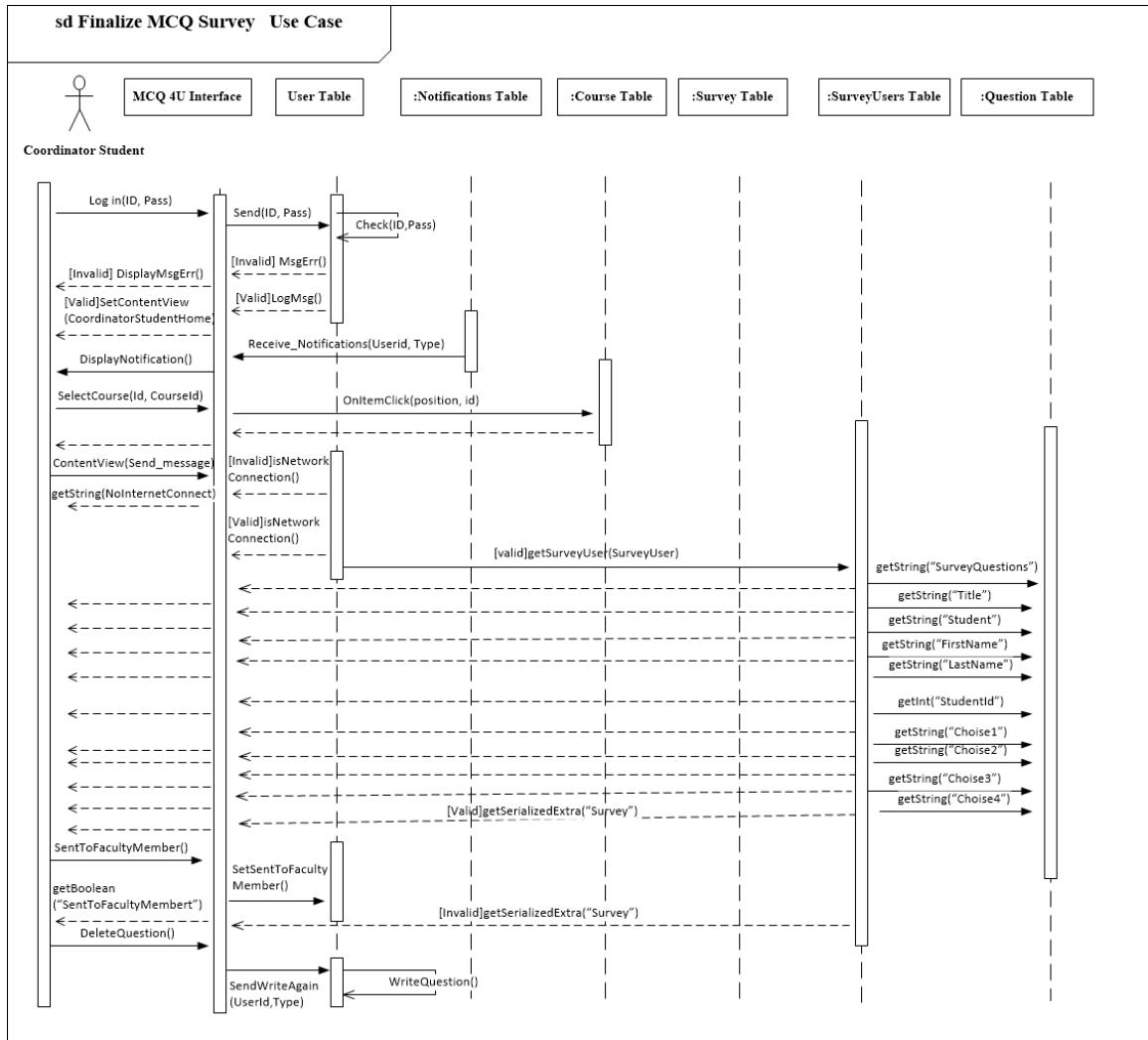


Figure 47: Finalize MCQ Survey sequence diagram

4.1.3.8: Create Course and Link Faculty Member

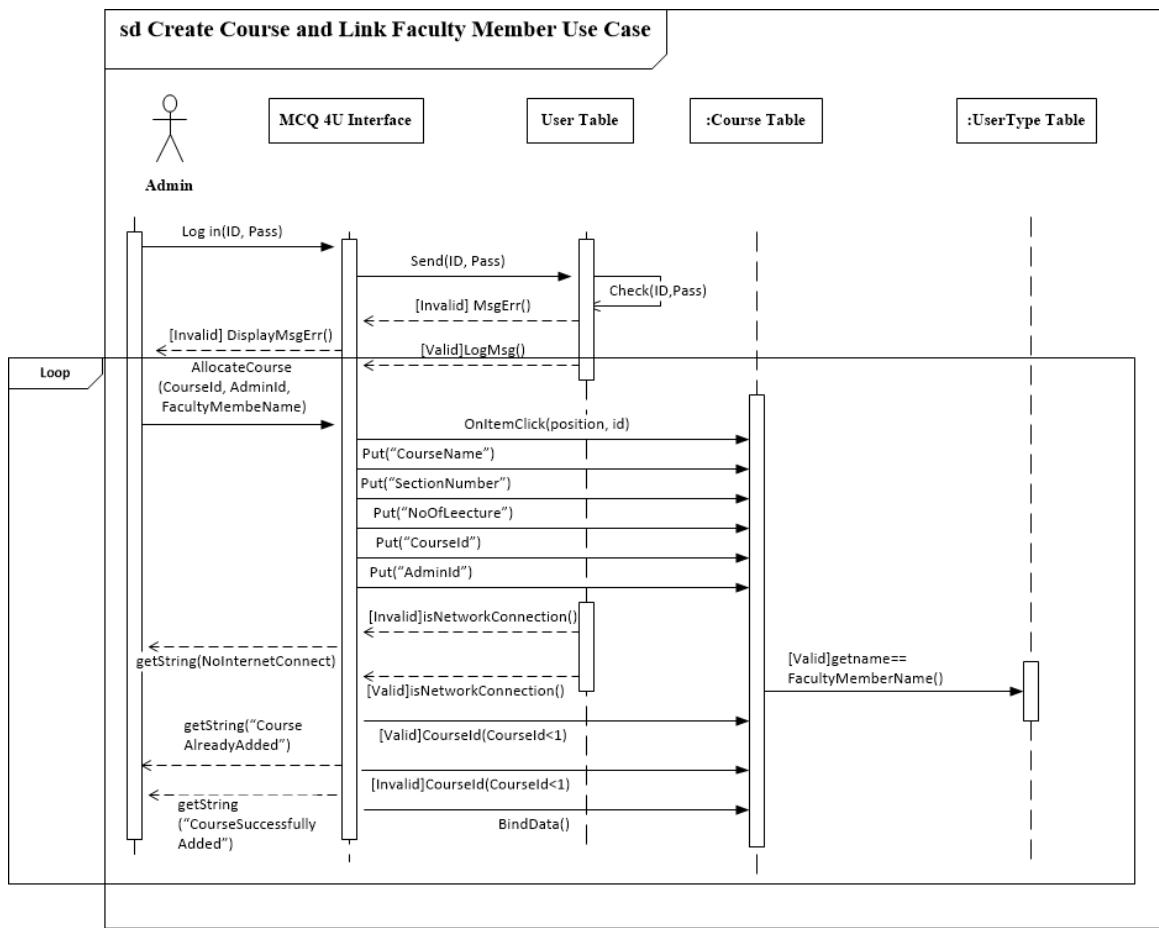


Figure 48: Create Course and Link Faculty Member sequence diagram

4.1.3.9: Delete course

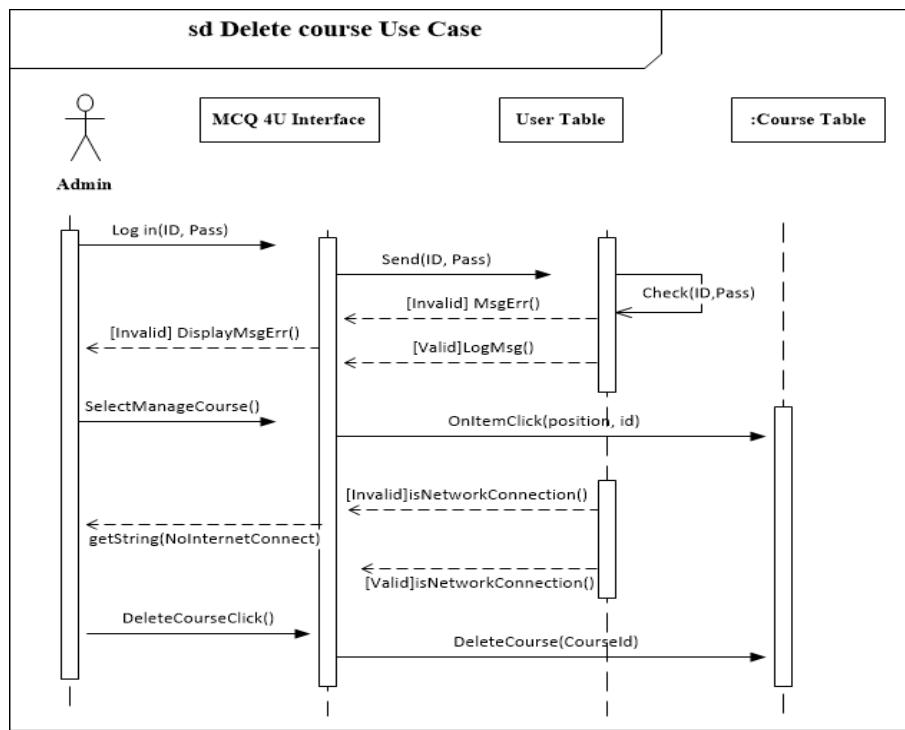


Figure 49: Delete Course sequence diagram

4.1.3.10: Link Student

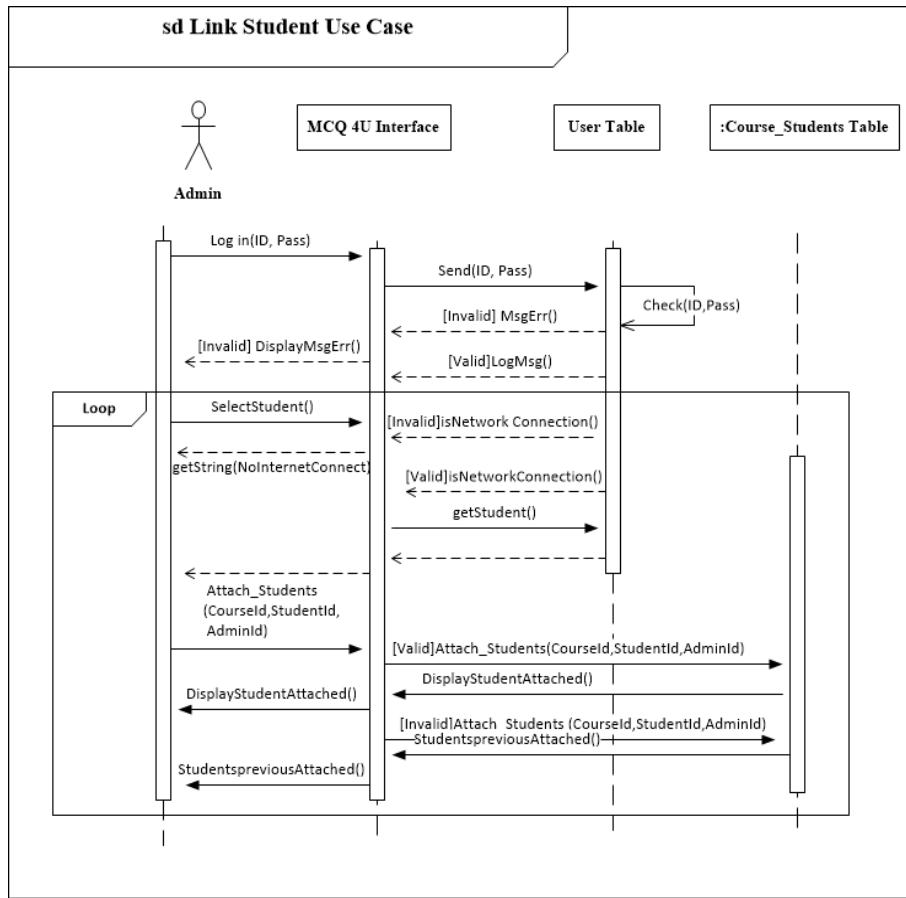


Figure 50: Link Student sequence diagram

4.1.3.11 Delete Faculty Member and Student

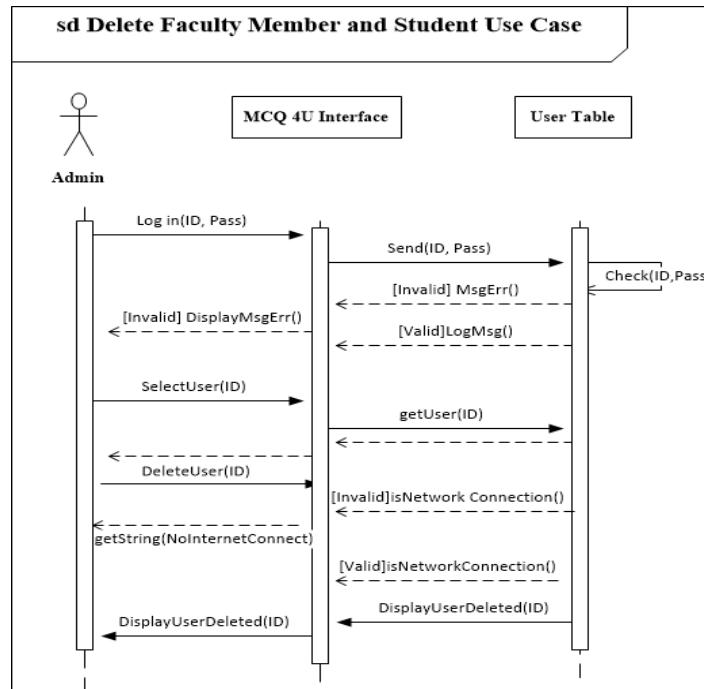


Figure 51: Delete Faculty Member and Student sequence diagram

4.2: Data Modelling

This section views the database structure of system, through database entity relationship diagram first and then the data dictionary table.

4.2.1: ER Diagram

An entity relationship diagram (ERD) illustrate the relationships of entity sets in a database. The entity in this context compose of many attributes, so the ER diagram reflects the logical structure of databases [17].

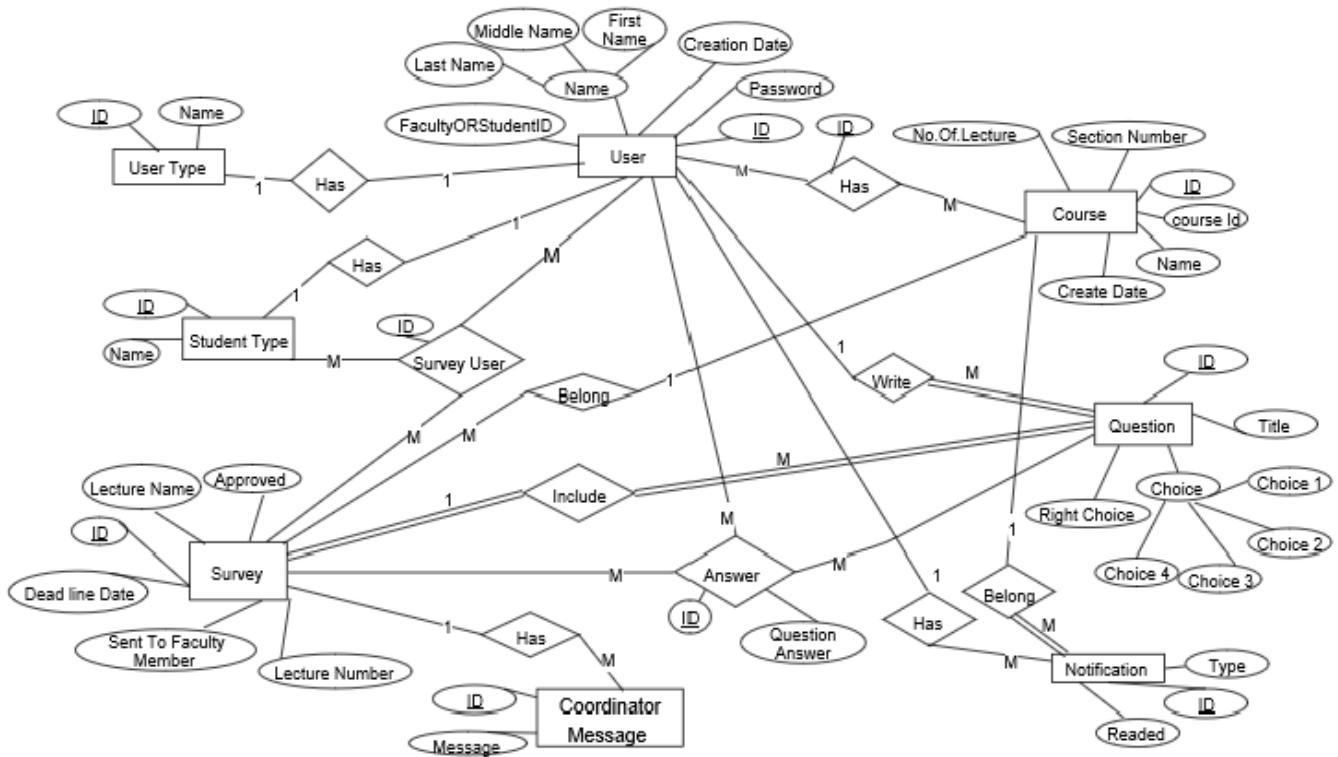


Figure 52: ER diagram

4.2.2: Data Dictionary

A data dictionary contains a list of all tables and its fields in database, and provides detailed information about the attributes.

4.2.2.1: Entities

User	A general term describing user the system.
Surveys	A general term describing survey in the system.
Courses	A general term describing courses in the system.
Questions	A general term describing all questions in the system.
Student Types	A general term describing student type in the system.
User Type	A general term describing user type in the system.
Coordinator Message	A general term describing coordinator message in the system.

Notification	A general term describing notification in the system.
---------------------	---

Table 16: Entities Table

4.2.2.2: Relationships

Entity Name	Multiplicity	Relationship	Entity Name
User Type	M . M	Answers	Question, Survey
Student Types	1 . 1	Has	User
Surveys	1 . M	Include	Question
Surveys	1 . M	Has	Coordinator Message
Notification	M . 1	Belong	Course
Question	M . 1	Write	User Type
User Type	1 . 1	Has	User
Surveys	M . M	Survey User	Student Type, User Type
Courses	1 . M	Belong	Survey
User Type	M . M	Has	Course
Notification	M . 1	Has	User

Table 17: data description (relationship)

4.2.2.3: Attributes

	Attribute	Description	Data Type	Length	Null	PK	FK
User							
	Id	Uniquely identify the user	Number	10	No	Yes	
	First name	Identify the first name of user	Varchar	20	No		
	Middle name	Identify the middle name of user	Varchar	20	No		
	Last Name	Identify the last name of user	Varchar	20	No		
	Password	Identify the user password	Number	10	No		

	User Type	Identify the user type Admin or student or faculty member	Number	2	No		Yes
	Faculty OR Student ID	Identify the university number for Faculty member or student	Varchar	9	No		
	Create Date	Identify create date of the user	Date Time	8	No		
Course							
	Id	Uniquely identify the course	Number	10	No	Yes	
	Course Id	Identify the id course	Number	10	No	-	
	Name	Identify the name of Course	Varchar	30	No		
	Section number	Identify the Section number of Course	Number	10	No		
	Faculty Member ID	Identify the Faculty Member id	Number	10	No		Yes
	Admin ID	Identify the admin id	Number	10	No		Yes
	Create Date	Identify create date of the course	Date time	8	No		
	No.Of.Lectures	Identify the lecture number in course	Number	10	No		
Question							
	Id	Uniquely identify the id question.	Number	10	No	Yes	
	Student id	Identify the id student.	Number	10	No		Yes
	Survey id	Identify the id of survey	Varchar	10	No		Yes
	Title	Identify the title of question	Varchar	100	No		

	Right Choice	Identify the right of question.	Number	4	No		
	Choice 1	Identify the choice 1	Number	100	No		
	Choice 2	Identify the choice 2	Number	100	No		
	Choice 3	Identify the choice 3	Number	100	No		
	Choice 4	Identify the choice 4	Number	100	No		
Surveys							
	Id	Identify the id of surveys	Number	10	No	Yes	
	Dead line Date	Identify the dead line date for surveys	Varchar	20	No		
	Lecture Name	Identify the lecture name	Varchar	30	No		
	Lecture Number	Identify the lecture number	Number	30	No		
	Sent to faculty member	Identify the sent surveys to faculty member	Bit	1	No		
	Approve	Identify approve surveys from faculty member	Bit	1	No		
	Course Id	Identify the id course	Number	10	No		Yes
Notification							
	Id	Identify the id of notification	Number	10	No	Yes	
	Readed	Identify the readed of notification	Bit	1	No		
	Type	Identify the type of notification	Number	1	No		
	User Id	Identify the user id for send notification	Number	10	No		Yes
	Course Id	Identify the id of course	Number	10	No		Yes

Coordinator Message							
	Id	Uniquely identify the coordinator message	Number	10	No	Yes	
	Message	Identify the message content.	Varchar	30	No		
	Survey Id	Identify the id survey	Number	10	No		Yes
User Type							
	ID	Identify the id of user type student or faculty member or admin.	Number	10	No	Yes	
	Name	Identify the name of user type.	Number	20	No		
Student Type							
	ID	Identify the id of student type (asker or coordinator message or respondent)	Number	10	No	Yes	
	Name	Identify the name of student.	Number	20	No		

Table 18: data description (attributes)

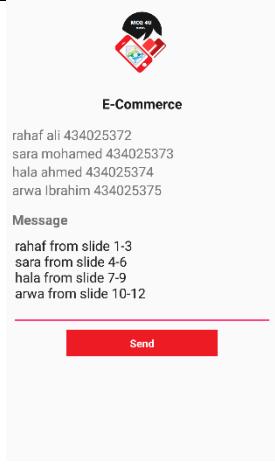
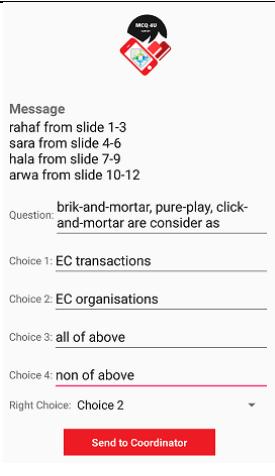
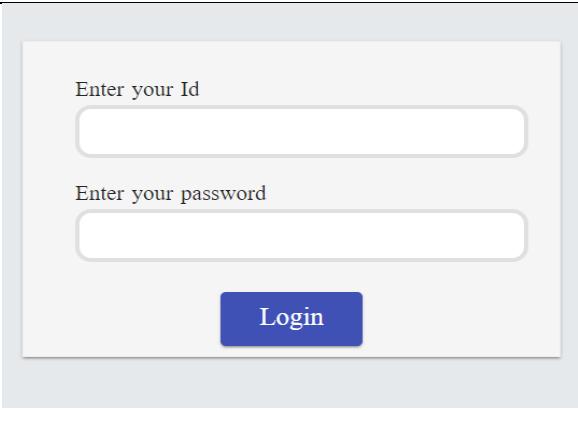
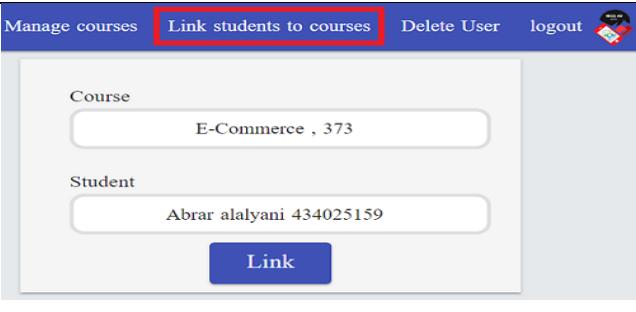
Interface application is way through which a Faculty Member and student to interact with system. The application provide two interfaces different for a Faculty Member, student. Show in the table below.

4.3: Detailed Interface design

4.3.1: Input design

Input interface is primarily used to allow the user to enter data into the system. usually, the system gives a result to the user. The MCQ 4U Application input interface are show in the table 20 below.

Description	Interface																					
Create account interface. This interface allows the user to register in MCQ 4U.	 <p>First name Boshra</p> <p>Middle name ali</p> <p>Last name alali</p> <p>Password *****</p> <p>Confirm password *****</p> <p>Student or faculty member ID 966123584</p> <p>User Type <input type="radio"/> Student <input checked="" type="radio"/> Faculty Member</p>																					
Log in interface. This interface allows the user to Log into MCQ 4U.	 <p>ID: 434025159</p> <p>Password: *****</p> <p>Login</p> <p>have not an account Register</p>																					
Build survey interfaces. This interface allows the Faculty Member to create a survey.	 <p>E-Commerce</p> <p>Deadline date 2017-12-8</p> <p>Lecture Name chapter 1: introduction to EC</p> <p>Number Of Lectures 1</p> <table border="1"> <thead> <tr> <th>Students</th> <th>Four Student</th> <th>Coordinator</th> </tr> </thead> <tbody> <tr> <td>Nourah salem</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>rahal ali</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>sara mohamed</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>hala ahmed</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>arwa ibrahim</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> <tr> <td>jana mohamed</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </tbody> </table> <p>Send to Coordinator</p>	Students	Four Student	Coordinator	Nourah salem	<input type="checkbox"/>	<input checked="" type="checkbox"/>	rahal ali	<input checked="" type="checkbox"/>	<input type="checkbox"/>	sara mohamed	<input checked="" type="checkbox"/>	<input type="checkbox"/>	hala ahmed	<input checked="" type="checkbox"/>	<input type="checkbox"/>	arwa ibrahim	<input checked="" type="checkbox"/>	<input type="checkbox"/>	jana mohamed	<input type="checkbox"/>	<input type="checkbox"/>
Students	Four Student	Coordinator																				
Nourah salem	<input type="checkbox"/>	<input checked="" type="checkbox"/>																				
rahal ali	<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
sara mohamed	<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
hala ahmed	<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
arwa ibrahim	<input checked="" type="checkbox"/>	<input type="checkbox"/>																				
jana mohamed	<input type="checkbox"/>	<input type="checkbox"/>																				

<p>Message interface. This interface allows the Coordinator to send lecture part that assigned for each asker student.</p>	 <p>E-Commerce</p> <p>rahal ali 434025372 sara mohamed 434025373 hala ahmed 434025374 arwa ibrahim 434025375</p> <p>Message</p> <p>rahal from slide 1-3 sara from slide 4-6 hala from slide 7-9 arwa from slide 10-12</p> <p>Send</p>	
<p>Create question interface. This interface allows the Asker Student to read Coordinator message and write question.</p>	 <p>MCQ 40</p> <p>Message</p> <p>rahal from slide 1-3 sara from slide 4-6 hala from slide 7-9 arwa from slide 10-12</p> <p>Question: <u>brik-and-mortar, pure-play, click-and-mortar are consider as</u></p> <p>Choice 1: <u>EC transactions</u></p> <p>Choice 2: <u>EC organisations</u></p> <p>Choice 3: <u>all of above</u></p> <p>Choice 4: <u>non of above</u></p> <p>Right Choice: Choice 2</p> <p>Send to Coordinator</p>	
<p>Login interface. This interface allows the Admin to Log into the system.</p>	 <p>Enter your Id</p> <input type="text"/> <p>Enter your password</p> <input type="password"/> <p>Login</p>	
<p>Link students to courses interface. This interface allows the Admin to link students to courses.</p>	 <p>Manage courses Link students to courses Delete User logout </p> <p>Course</p> <p>E-Commerce , 373</p> <p>Student</p> <p>Abrar alalyani 434025159</p> <p>Link</p>	

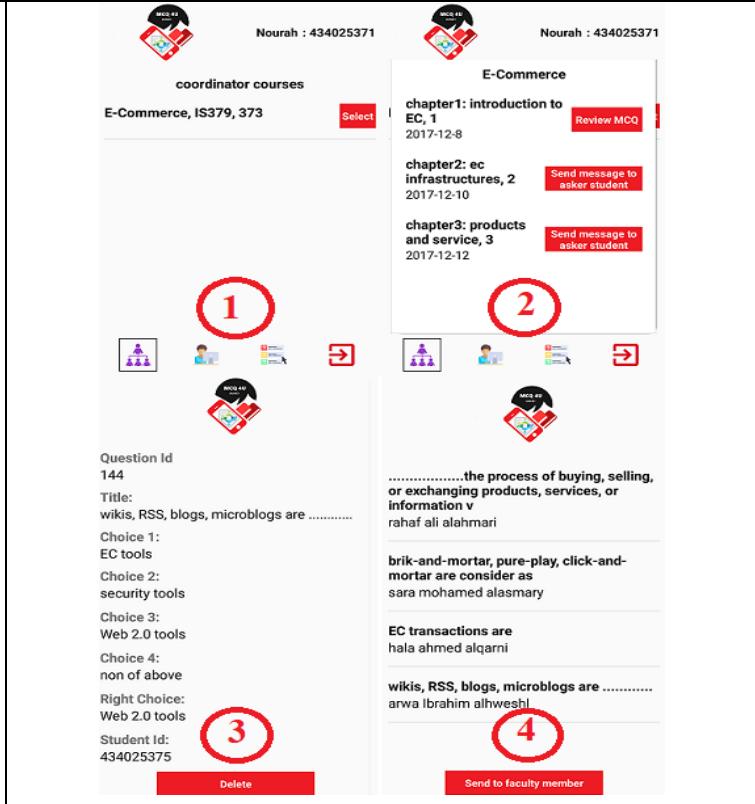
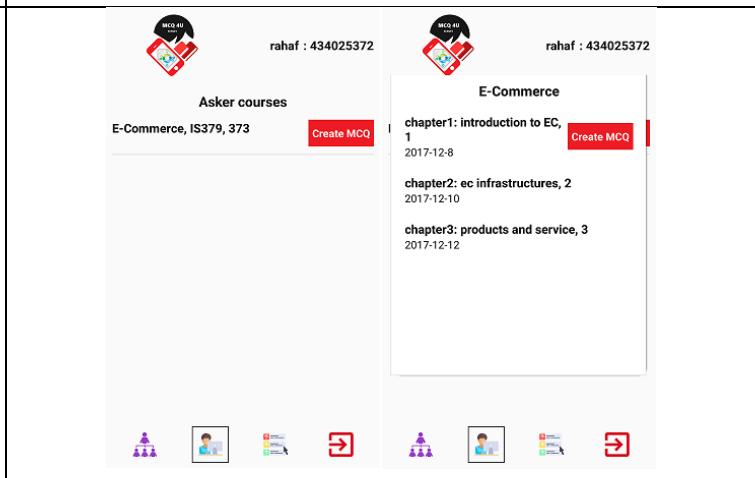
<p>Create course interface. This interface allows the Admin to create course and link it to Faculty Member.</p>	
---	--

Table 19: Input interface

4.3.2: Output interface

Output interface is used to presentation data for the user. The MCQ Application output interface are shown in the table 21 below.

Description	Interface
<p>Home interface. This interface allows the Faculty Member to view her courses.</p>	
<p>Prepared surveys interface. This interface allows the Faculty Member to approve and publish prepared surveys for each course.</p>	

<p>Surveys statistics interface.</p> <p>This interface allows the Faculty Member to view statistics for each survey.</p>	 <p>surveys statistics E-Commerce, IS379, 373</p> <p>E-Commerce</p> <ul style="list-style-type: none"> chapter1: introduction to EC, 1 2017-12-8 chapter2: ec infrastructures, 2 2017-12-10 <p>surveys statisticsthe process of buying, selling, or exchanging products, services, or information v</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3 (60%)</td> <td>3</td> <td>60%</td> </tr> <tr> <td>2 (40%)</td> <td>2</td> <td>40%</td> </tr> </tbody> </table> <p>brik-and-mortar, pure-play, click-and-mortar are consider as</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>2 (40%)</td> <td>2</td> <td>40%</td> </tr> <tr> <td>3 (60%)</td> <td>3</td> <td>60%</td> </tr> </tbody> </table> <p>EC transactions are</p> <table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3 (60%)</td> <td>3</td> <td>60%</td> </tr> <tr> <td>2 (40%)</td> <td>2</td> <td>40%</td> </tr> </tbody> </table>	Response	Count	Percentage	3 (60%)	3	60%	2 (40%)	2	40%	Response	Count	Percentage	2 (40%)	2	40%	3 (60%)	3	60%	Response	Count	Percentage	3 (60%)	3	60%	2 (40%)	2	40%
Response	Count	Percentage																										
3 (60%)	3	60%																										
2 (40%)	2	40%																										
Response	Count	Percentage																										
2 (40%)	2	40%																										
3 (60%)	3	60%																										
Response	Count	Percentage																										
3 (60%)	3	60%																										
2 (40%)	2	40%																										
<p>Coordinator interfaces.</p> <p>These interfaces allow the Coordinator Student to:</p> <ol style="list-style-type: none"> view courses. choose review survey or send message to Asker Student. view the details of Asker Students questions. view Asker Students questions. 	 <p>Nourah : 434025371</p> <p>coordinator courses E-Commerce, IS379, 373</p> <p>E-Commerce</p> <ul style="list-style-type: none"> chapter1: introduction to EC, 1 2017-12-8 chapter2: ec infrastructures, 2 2017-12-10 chapter3: products and service, 3 2017-12-12 <p>1   </p> <p>2   </p> <p>3   </p> <p>4   </p> <p>Question Id: 144 Title: wikis, RSS, blogs, microblogs are rahal ali alahmari Choice 1: EC tools Choice 2: security tools Choice 3: Web 2.0 tools Choice 4: non of above Right Choice: Web 2.0 tools Student Id: 434025375 Delete</p> <p>.....the process of buying, selling, or exchanging products, services, or information v rahal ali alahmari brik-and-mortar, pure-play, click-and-mortar are consider as sara mohamed alasmary EC transactions are hala ahmed alqarni wikis, RSS, blogs, microblogs are arwa ibrahim alhweshi Send to faculty member</p>																											
<p>Asker interface. This interface allows the Asker Student to view courses and select a survey to create question.</p>	 <p>rahal : 434025372</p> <p>Asker courses E-Commerce, IS379, 373</p> <p>E-Commerce</p> <ul style="list-style-type: none"> chapter1: introduction to EC, 1 2017-12-8 chapter2: ec infrastructures, 2 2017-12-10 chapter3: products and service, 3 2017-12-12 <p>Create MCQ</p> <p>rahal : 434025372</p> <p>E-Commerce</p> <ul style="list-style-type: none"> chapter1: introduction to EC, 1 2017-12-8 chapter2: ec infrastructures, 2 2017-12-10 chapter3: products and service, 3 2017-12-12 <p>Create MCQ</p>																											

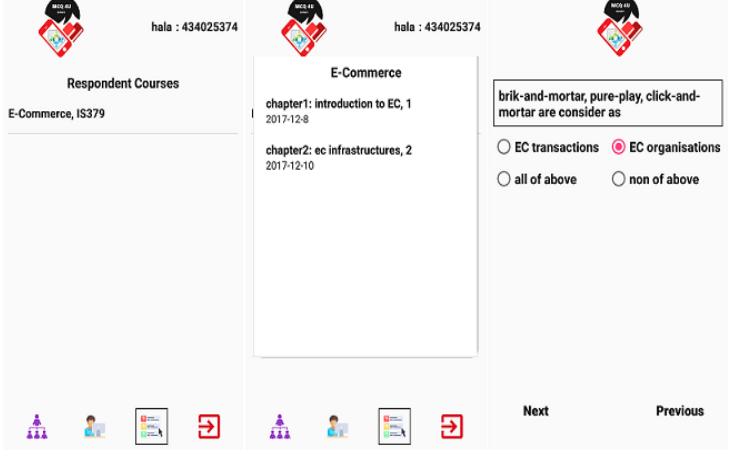
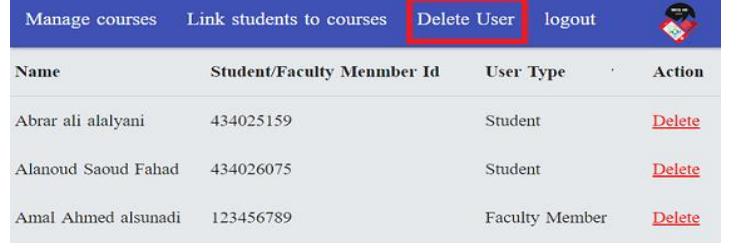
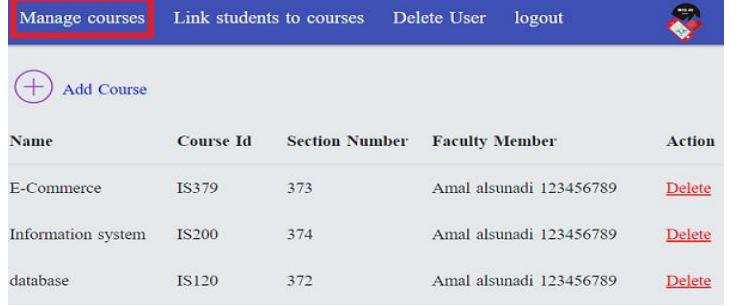
<p>Respondent interface. This interface allows the Respondent Student to solve the surveys for each course.</p>	 <p>The screenshot shows a survey interface for the E-Commerce course (IS379). The user ID is hala : 434025374. The survey consists of two chapters: chapter1: introduction to EC, 1 (2017-12-8) and chapter2: ec infrastructures, 2 (2017-12-10). The survey questions include: brik-and-mortar, pure-play, click-and-mortar are consider as <input type="radio"/> EC transactions <input checked="" type="radio"/> EC organisations <input type="radio"/> all of above <input type="radio"/> non of above</p>																				
<p>Home interface. This interface allows the Admin to move between pages easily.</p>	 <p>Navigation buttons: Delete User Manage courses Link Users To Course</p>																				
<p>Delete user interfaces. This interface allows Admin to delete user either Student or Faculty Member.</p>	 <table border="1"> <thead> <tr> <th>Name</th> <th>Student/Faculty Member Id</th> <th>User Type</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>Abrar ali alalyani</td> <td>434025159</td> <td>Student</td> <td>Delete</td> </tr> <tr> <td>Alanoud Saoud Fahad</td> <td>434026075</td> <td>Student</td> <td>Delete</td> </tr> <tr> <td>Amal Ahmed alsunadi</td> <td>123456789</td> <td>Faculty Member</td> <td>Delete</td> </tr> </tbody> </table>	Name	Student/Faculty Member Id	User Type	Action	Abrar ali alalyani	434025159	Student	Delete	Alanoud Saoud Fahad	434026075	Student	Delete	Amal Ahmed alsunadi	123456789	Faculty Member	Delete				
Name	Student/Faculty Member Id	User Type	Action																		
Abrar ali alalyani	434025159	Student	Delete																		
Alanoud Saoud Fahad	434026075	Student	Delete																		
Amal Ahmed alsunadi	123456789	Faculty Member	Delete																		
<p>Mange courses interface. This interface allows the Admin to delete course.</p>	 <table border="1"> <thead> <tr> <th>Name</th> <th>Course Id</th> <th>Section Number</th> <th>Faculty Member</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>E-Commerce</td> <td>IS379</td> <td>373</td> <td>Amal alsunadi 123456789</td> <td>Delete</td> </tr> <tr> <td>Information system</td> <td>IS200</td> <td>374</td> <td>Amal alsunadi 123456789</td> <td>Delete</td> </tr> <tr> <td>database</td> <td>IS120</td> <td>372</td> <td>Amal alsunadi 123456789</td> <td>Delete</td> </tr> </tbody> </table>	Name	Course Id	Section Number	Faculty Member	Action	E-Commerce	IS379	373	Amal alsunadi 123456789	Delete	Information system	IS200	374	Amal alsunadi 123456789	Delete	database	IS120	372	Amal alsunadi 123456789	Delete
Name	Course Id	Section Number	Faculty Member	Action																	
E-Commerce	IS379	373	Amal alsunadi 123456789	Delete																	
Information system	IS200	374	Amal alsunadi 123456789	Delete																	
database	IS120	372	Amal alsunadi 123456789	Delete																	

Table 20: Output interface

4.4: Component Diagram

4.4.1: Component Diagram

Component diagram present the functional parts in the system with the link between them, by reflecting the structural relationship between each component [18].

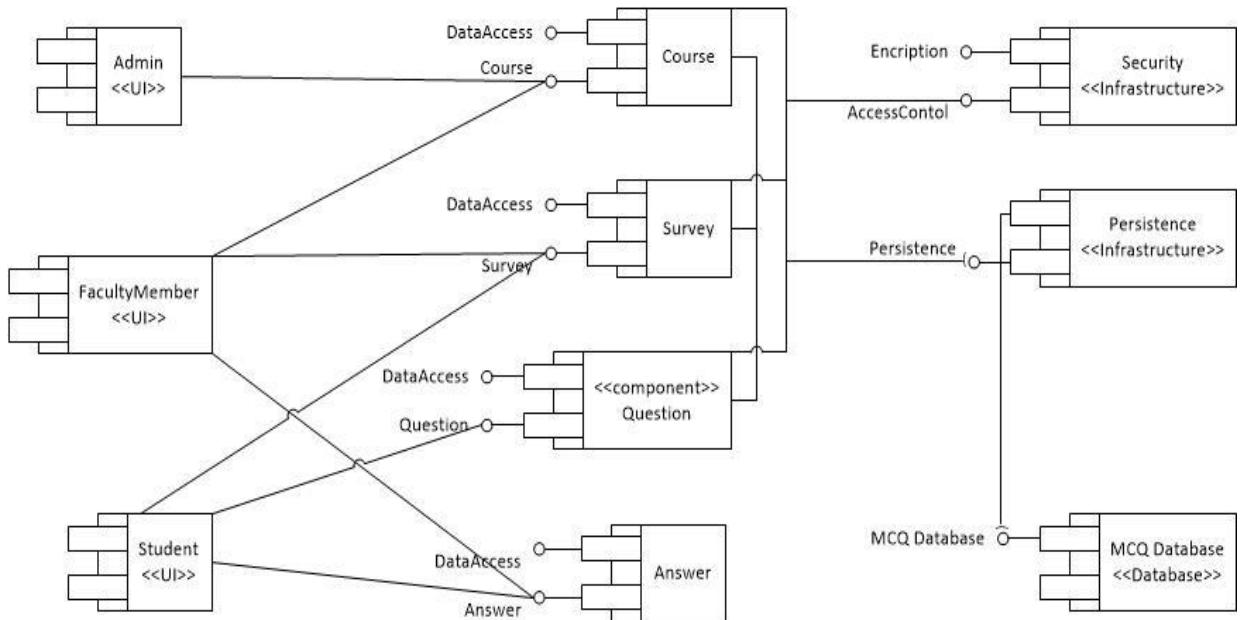


Figure 53: Component diagram

4.5: Alternative Designs

Since the idea of the project is new and the application is specific to Imam University, the application was built from scratch by developing application modules part by part. The properties provided by some libraries in the android language were also used.

The implementation of the project was chosen in the form of a mobile application because it is easy of use among the stakeholders. The application was adopted to work on the mobile devices that carry the android system for the most widespread, the ease of language used in the development of android applications compared to the IOS, in addition to the easy procedures to upload the application to the google play store.

Summary

This chapter gave the analytic structure of the system which consider the critical thing in next phase of the project which is system specification.

Chapter Five: System Implementation

Introduction

The deliverable at the end of this chapter is the system requirements specification (hardware and software) that used to get the services that the application expected to perform. System testing for complete software especially testing after integrated application pages and web pages to evaluate the system's compliance with the specified requirements.

In addition, providing the deployment diagram to describe the physical architecture in MCQ 4U application.

5.1: System Specification

5.1.1: Hardware Specifications

Categorizing the physical components that used to accomplish the application needs.

Mobile Device		
Name	Description	Picture
Galaxy prime.	Android OS version higher than 5.0.	
Qualcomm Quad processor chips or higher.	It is a mobile processor supports smartphone features such as Internet connections, and full HD displays.	
RAM memory card.	RAM helps the processor to work more quickly by providing more space, used between 1.00GB up to max GB.	
Modem	Network hardware device to provide internet access from ISP.	

Table 21: Mobile Device

Computer Device		
Name	Description	Picture
Laptops.	OS type: windows 7 and higher.	
Processor CPU	Used i5 core with speed of 2.3 GHz or higher to execute commands from the hardware and software that included in computer.	
RAM	Used 3 GB ram or higher by the operating system for reaching quickly application programs and data that in current use by the CPU.	
Hard disk	Used to provide large data storage capacity by 500 GB or higher.	

Table 22: computer Device

5.1.2: Software Specifications

Categorizing the software resources that used to meet application's objectives.

Name	Description	Picture
Android Studio	Used for building the Android application by typing the code using code editor and developing tools.	

Name	Description	Picture
Microsoft office	Using its package for writing and edit project document with Microsoft word and representing it using Microsoft representation, creating WBS and Gantt chart using Microsoft Project.	 
Visual Studio	Visual Studio .NET is Microsoft's visual programming environment for creating Web services using the ASP.NET with C#, for database connections.	
Google Drive	Used to share files and storing them by team members in the cloud storage.	
Microsoft Azure	Windows Azure is Microsoft's public cloud computing platform used in MCQ 4U application for hosting the web services and connecting databases.	
Mockups	Used to design mockup interfaces.	
Microsoft Visio	Used to represent system analysis and design phases' diagrams.	
Photoshop Adobe	Used to design MCQ 4U logo.	

Table 23: Software Specifications

5.2: System Testing

5.2.1: Features to be tested

The objectives of this section are to ensure that all requirements are satisfied, the users are satisfied and the system works properly. Features to be tested are login, register, build a survey, send message to Asker Student, create and delete MCQ question, review and publish and answer a survey, link student to course, create and delete course.

5.2.2: Test Cases

5.2.2.1: Register

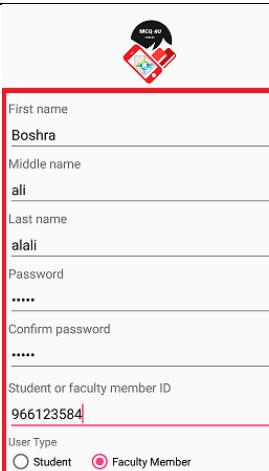
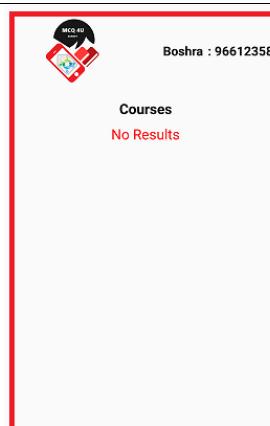
Purpose	Registration
Input	First, middle and last name, password, Faculty Member or student Id, determine user type either Faculty Member or student
Expected output	Open user account
Pass / Fail	Pass
Test result	 

Table 24: Register test case

5.2.2.2: Login

Purpose	Login
Input	Faculty Member or Student id, password
Expected output	Open user account
Pass / Fail	Pass

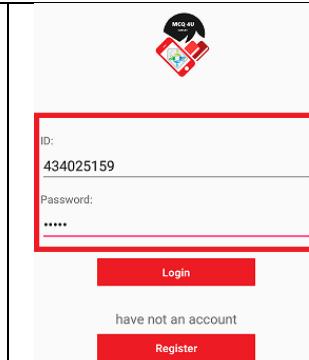
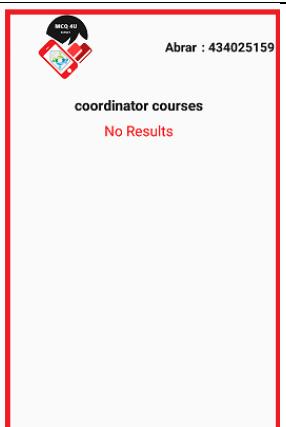
Test result		
-------------	--	---

Table 25: Login test case

5.2.2.3: Build MCQ survey

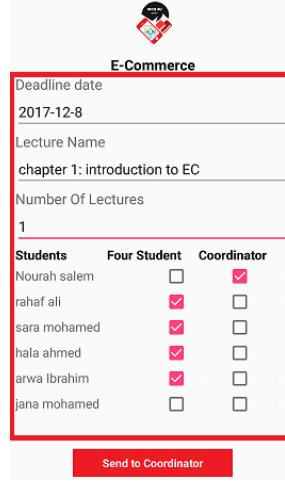
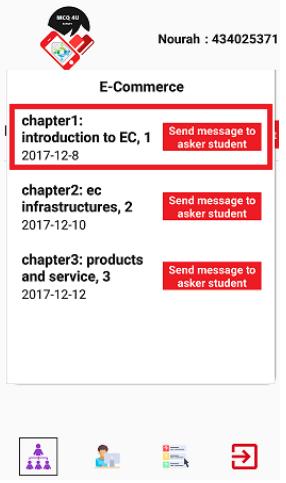
Purpose	Build a survey	
Input	Deadline date, lecture name, number of lecture, select five students four Asker and one Coordinator	
Expected output	Coordinator received MCQ survey	
Pass / Fail	Pass	
Test result		

Table 26: Build a survey test case

5.2.2.4: Send message to Asker Student

Purpose	Send lecture parts to Asker Student
Input	Message that includes lecture parts that assigned to each Asker Student
Expected output	Asker Student received message
Pass / Fail	Pass

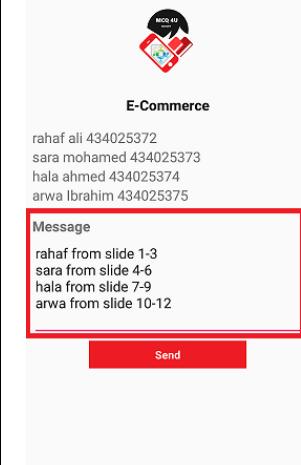
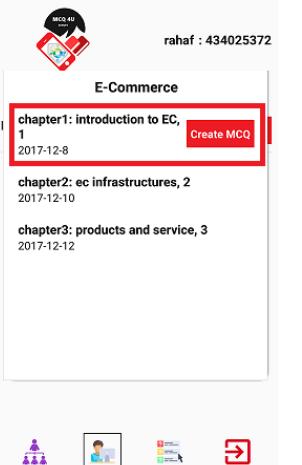
Test result		
-------------	--	---

Table 27: Send message to Asker Student test case

5.2.2.5: Create MCQ question

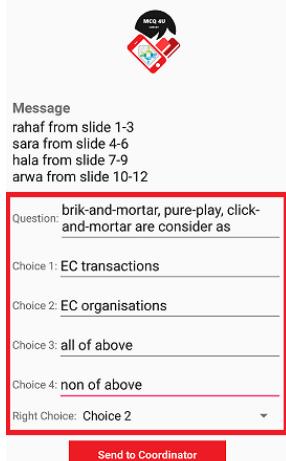
Purpose	Create question
Input	Question, four choices, determine right choice
Expected output	Coordinator received four questions
Pass / Fail	Pass
Test result	

Table 28: Create MCQ question test case

5.2.2.6: Delete MCQ question

Purpose	Delete question by coordinator
Input	Asker Student question
Expected output	Asker Student rewrite question
Pass / Fail	Pass

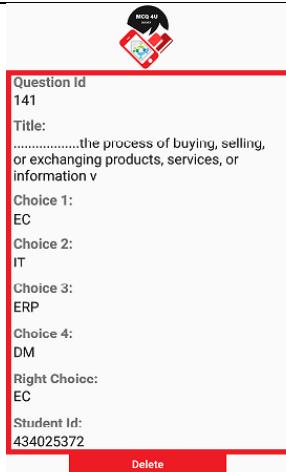
Test result		
-------------	--	---

Table 29: Delete MCQ question test case

5.2.2.7: Finalize MCQ survey

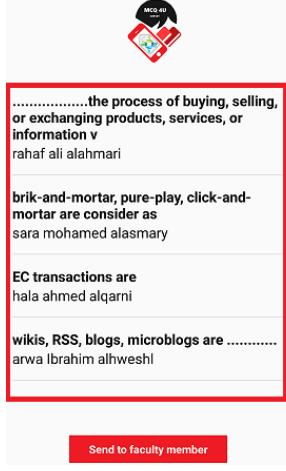
Purpose	Review a survey to make sure each Asker Student wrote question in their responsible part
Input	Four questions that written by Asker Students
Expected output	Faculty Member received completed survey
Pass / Fail	Pass
Test result	

Table 30: Review a survey test case

5.2.2.8: Publish MCQ survey

Purpose	Publish a survey
Input	Four questions that written by Asker students
Expected output	Students received MCQ survey
Pass / Fail	Pass

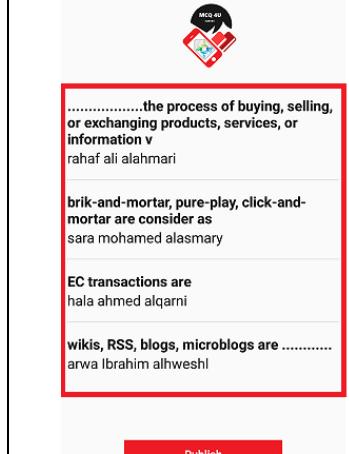
Test result		
-------------	--	---

Table 31: Publish a survey test case

5.2.2.9: Submit Answer

Purpose	Answer a survey
Input	Submit answer of survey
Expected output	Statistics
Pass / Fail	pass
Test result	

Table 32: Answer a survey test case

5.2.2.10: Delete user

Purpose	Delete User from system
Input	user
Expected output	Remove user from users list
Pass / Fail	pass

Test result	Manage courses Link students to courses Delete User logout 			
	Name	Student/Faculty Member Id	User Type	Action
	Abrar ali alalyani	434025159	Student	Delete
	Alanoud Saoud Fahad	434026075	Student	Delete
	Amal Ahmed alsunadi	123456789	Faculty Member	Delete
	Manage courses Link students to courses Delete User logout			
	Name	Student/Faculty Member Id	User Type	Action
	Alanoud Saoud Fahad	434026075	Student	Delete
	Amal Ahmed alsunadi	123456789	Faculty Member	Delete

Table 33: Delete user test case

5.2.2.11: Link Student to course

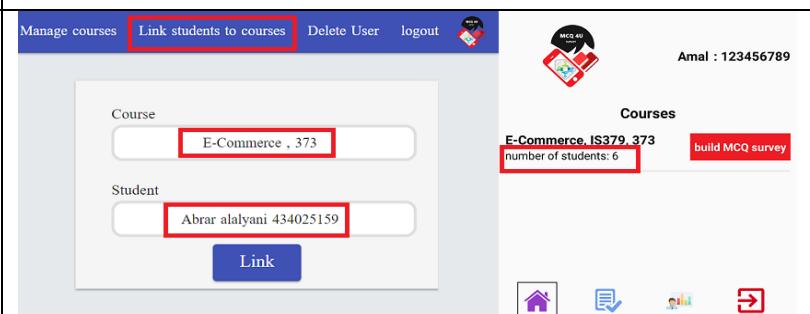
Purpose	Link student to course			
Input	Course, Student			
Expected output	The number of Student registered to course will increase			
Pass / Fail	pass			
Test result				

Table 34: Link Student to course test case

5.2.2.12: Delete course

Purpose	Delete course	
Input	Course	
Expected output	Remove course from courses list	
Pass / Fail	pass	

Test result	 <table border="1" data-bbox="774 316 1430 377"> <thead> <tr> <th>Name</th><th>Course Id</th><th>Section Number</th><th>Faculty Member</th><th>Action</th></tr> </thead> <tbody> <tr> <td>E-Commerce</td><td>IS379</td><td>373</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> <tr> <td>Information system</td><td>IS200</td><td>374</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> <tr> <td>database</td><td>IS120</td><td>372</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> </tbody> </table>  <table border="1" data-bbox="774 608 1430 669"> <thead> <tr> <th>Name</th><th>Course Id</th><th>Section Number</th><th>Faculty Member</th><th>Action</th></tr> </thead> <tbody> <tr> <td>Information system</td><td>IS200</td><td>374</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> <tr> <td>database</td><td>IS120</td><td>372</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> </tbody> </table>	Name	Course Id	Section Number	Faculty Member	Action	E-Commerce	IS379	373	Amal alsunadi 123456789	Delete	Information system	IS200	374	Amal alsunadi 123456789	Delete	database	IS120	372	Amal alsunadi 123456789	Delete	Name	Course Id	Section Number	Faculty Member	Action	Information system	IS200	374	Amal alsunadi 123456789	Delete	database	IS120	372	Amal alsunadi 123456789	Delete
Name	Course Id	Section Number	Faculty Member	Action																																
E-Commerce	IS379	373	Amal alsunadi 123456789	Delete																																
Information system	IS200	374	Amal alsunadi 123456789	Delete																																
database	IS120	372	Amal alsunadi 123456789	Delete																																
Name	Course Id	Section Number	Faculty Member	Action																																
Information system	IS200	374	Amal alsunadi 123456789	Delete																																
database	IS120	372	Amal alsunadi 123456789	Delete																																

Table 35: Delete course test case

5.2.2.13: Create course

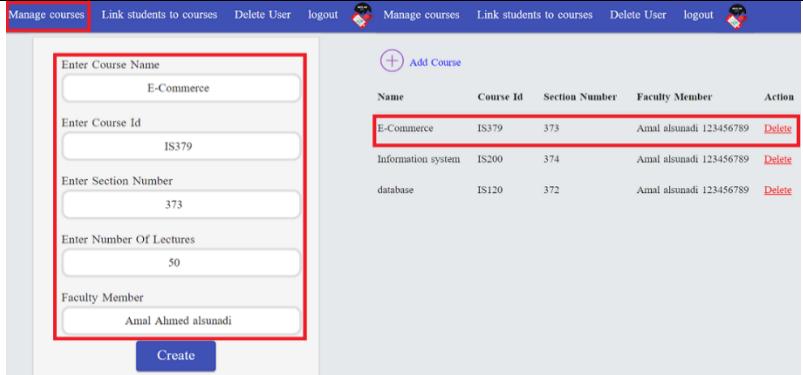
Purpose	Add new course																				
Input	Name and id of course, section number, number of lectures, faculty member																				
Expected output	Add course to courses list																				
Pass / Fail	Pass																				
Test result	 <table border="1" data-bbox="643 1125 1446 1455"> <thead> <tr> <th>Name</th><th>Course Id</th><th>Section Number</th><th>Faculty Member</th><th>Action</th></tr> </thead> <tbody> <tr> <td>E-Commerce</td><td>IS379</td><td>373</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> <tr> <td>Information system</td><td>IS200</td><td>374</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> <tr> <td>database</td><td>IS120</td><td>372</td><td>Amal alsunadi 123456789</td><td>Delete</td></tr> </tbody> </table>	Name	Course Id	Section Number	Faculty Member	Action	E-Commerce	IS379	373	Amal alsunadi 123456789	Delete	Information system	IS200	374	Amal alsunadi 123456789	Delete	database	IS120	372	Amal alsunadi 123456789	Delete
Name	Course Id	Section Number	Faculty Member	Action																	
E-Commerce	IS379	373	Amal alsunadi 123456789	Delete																	
Information system	IS200	374	Amal alsunadi 123456789	Delete																	
database	IS120	372	Amal alsunadi 123456789	Delete																	

Table 36: Create course test case

5.3: System Deployment

5.3.1: Deployment Diagram

It is a structure diagram that shows the architecture of the system as a relationship between software artifacts to deploy hardware components in the physical world such as databases, devices that connected via communication paths.

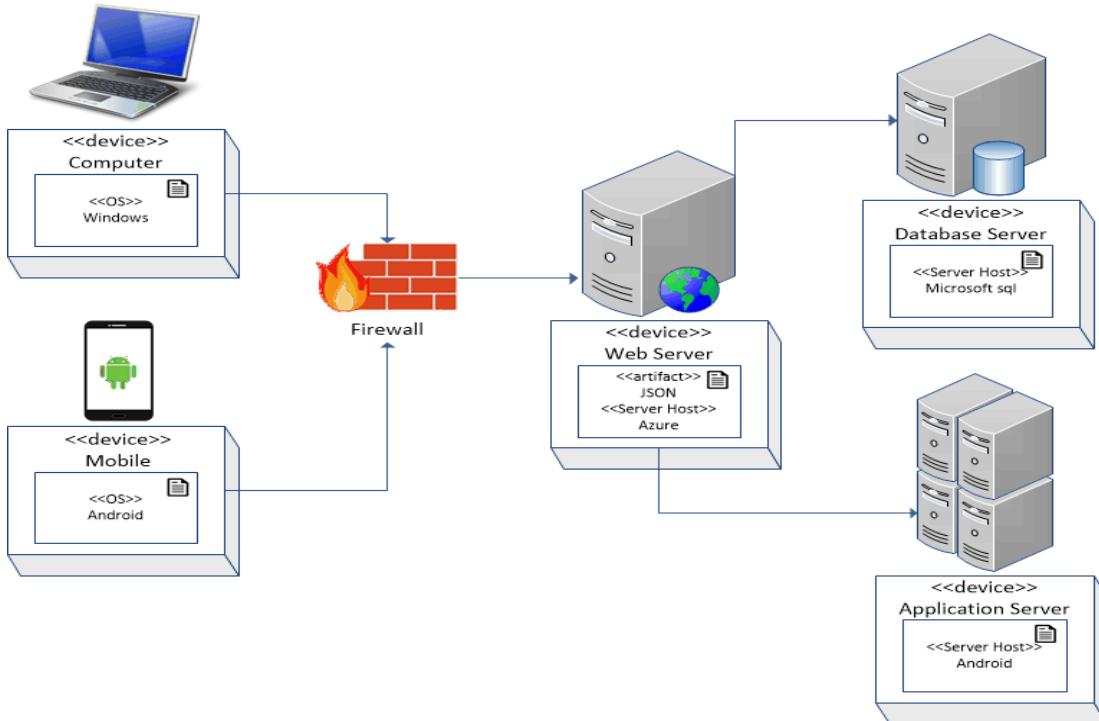


Figure 54: Deployment Diagram

The deployment diagram in figure (51) shows the relationship between hardware components and their software:

Mobile device with android operating system, web server with Azure sever hosting, database server with Microsoft SQL server hosting, and application with Android sever hosting.

5.3.2: Expected Users/Load

MCQ 4U served Students and Faculty Member in College of Computer and Information Sciences in Imam University, the expected load for the MCQ 4U application is:

- Provide the service for users from 1000 to 10000 in synchronization.
- The Database space is provided in 2GB.

5.3.3: Installation Process

- 1- The Students and Faculty Member must own smartphone with Android operating system.
- 2- The Students and Faculty Member must connect to internet.
- 3- The smartphone for Students and Faculty Member must have enough memory space.
- 4- The Students and Faculty Member must login to google play and download MCQ 4U application after searching application name.
- 5- The Students and Faculty Member run the MCQ 4U application.

Summary

In this chapter, team members test MCQ 4U application that achieve project requirements successfully, provides the system specifications that used to accomplish the project tasks, and describes deployment diagram to present how the application run physically, expected user load and process installation.

Next chapter will discuss the team member obstacles, future work and the final conclusion of the project.

Chapter Six: Conclusion and Future Work

Introduction

After system testing in chapter five, team members summarizing the deliverables of MCQ 4U Application that accomplished during the project phases.

This chapter provides the conclusion contain obstacles and suggest project's future work to keep up with evolution needs in education.

6.1: Conclusion

MCQ 4U is an Android OS mobile application that was advanced to run for Computer College Information Sciences in Imam University to improve its students' understanding-level about courses' subjects depending on their selves and display the results of students' knowledge to CCIS faculty members, the services that MCQ 4U provides:

- The students will use the application to generate their own questions that are the key to give them confidence in participating to facilitate the learning process to their faculty members.
- Enhance students' self-learning and collaboration skills, by dividing students into roles to coordinate and avoid duplicate questions.
- Provides faculty member statistics charts about students' thinking level when she/he writes the exams questions.
- Easy for the students to generate MCQs surveys depend on their roles every single lecture.
- Helps the Respondent Students to navigate easily in solving questions.

6.1.1 Obstacles:

MCQ 4U has obstacles and difficulties that team members overcome when developing and document the project that consider:

- Connecting Administrator's website pages to the Android environments.
- Programming the dynamic way that Students can change roles between Coordinator Student, Asker Student, or Respondent Student each lecture depending on the Faculty Member choice.
- Programming different notifications to the different Students change synchronization roles.

- Developing the application and documenting changes in parallel to handle the short time problem.
- The team members have different responsibilities and commitments to various courses that affect the project's progress.
- Learning to develop for Android OS environment using different sources to accomplish analysis requirements.

6.2: Future Work

There are many educational technologies that team member recognized in background chapter and requirement gathering techniques may provide in MCQ 4U application regards to prepare effectiveness and efficiency students for the future using and managing appropriate technological processes and resources.

The future work represented by developing MCQ 4U application on various mobile platforms such as IOS version, and Windows. Linking MCQ 4U application with CCIS database to make the administrator processes more formal such as linking faculty members and students in courses. Providing a public chat for students in each course to collaborate and help the absent student to cover the misunderstood parts in lectures.

Moreover, there are many ideas related to the application notifications which notify the student about recorded scores that calculated by the application of her performance in generating and solving MCQs, notifying the faculty member and students about top active five students in the course arranged in ranks measured by techniques that could be added to the application.

Summary

In this chapter, team members observed the opportunities to develop MCQ 4U application in future especially after achieved these project requirements successfully to encourage and overcome their obstacles to improve Information Technology skills according to the digital development that enhances the education technology.

References

- [1] Xin Chen, Mihaela Vorvoreanu, Krishna P. C. Madhavan, "Mining Social Media Data for Understanding Students' Learning Experiences", *IEEE Transactions on Learning Technologies*, vol. 7, no., pp. 246-259, July-Sept. 2014, doi:10.1109/TLT.2013.2296520
- [2] Cooke, S. (2015), "Social teaching: student perspectives on the inclusion of social media in higher education", *Education and Information Technologies*, Vol. 21 No. 82, pp. 1-15, doi: 10.1007/s10639-015-9444-y.
- [3] Custom Exam, [online], Available at <http://www.customexam.com> [19/4/2017]
- [4] "Computer science MCQ" Educational App Store [website] Available at <http://www.educationalappstore.com/app/computer-science-mcq> [19/4/2017]
- [5] "About Blackboard" Available at <http://www.blackboard.com/about-us/index.aspx> [7/4/2017]
- [6] "About Blackboard" Available at <http://www.blackboard.com/about-us/index.aspx> [7/4/2017]
- [7] Custom Exam, [online], Available at <http://www.customexam.com> [19/4/2017]
- [8] "Computer science MCQ" Educational App Store [website] Available at <http://www.educationalappstore.com/app/computer-science-mcq> [19/4/2017]
- [9] Webdesign, "Online LMS software (Learning Management System) | Easy-LMS.com", *Easy-lms.com*, 2017. [Online]. Available: <https://www.easy-lms.com/>. [20/4/2017].
- [10] Steve Gladstone, "Waterfall mode", Available at <https://www.slideshare.net/SteveGladstone1/overview-of-sdlc-waterfall> [11/4/2017]
- [11] A. Mukhopadhyay and F. Ameri, "An ontological approach to engineering requirement representation and analysis," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 30, no. 4, pp. 337–352, 2016.
- [12] " Princess Nora bint Abdul Rahman University"
<http://www.pnu.edu.sa/arr/Pages/default.aspx>
- [13] T. Mochal, "10 techniques for gathering requirements - TechRepublic", TechRepublic, 2008. [Online]. Available: <http://www.techrepublic.com/blog/10-things/10-techniques-for-gathering-requirements/>. [Accessed: 05- May- 2017]. Used to guide the project team for how to gather requirements techniques.
- [14] tutorialspoint, "UML - Activity Diagrams," 25 10 2017. [Online]. Available: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm. [Accessed 25 10 2017].

- [15] "UML 2 Class Diagrams: An Agile Introduction," Agile Modeling, [Online]. Available: <https://www.lucidchart.com/pages/uml/class-diagram>. [Accessed 2017 11 18].
- [16] "UML 2 Sequence Diagrams: An Agile Introduction," Agile Modeling, [Online]. Available: <http://www.agilemodeling.com/artifacts/sequenceDiagram.htm>. [Accessed 18 11 2017].
- [17] M. Rouse, "entity relationship diagram," techtarget, [Online]. Available: <http://searchcrm.techtarget.com/definition/entity-relationship-diagram>. [Accessed 18 11 2017].
- [18] D. Bell, "The component diagram," ibm, [Online]. Available: <https://www.ibm.com/developerworks/rational/library/dec04/bell/index.html>. [Accessed 18 11 2017].

Appendix

A. Code Snippets

- **AnswerSurvey**

```
_course = (CoursesClass) getIntent().getSerializableExtra("course");
_survey = (SurveyModel) getIntent().getSerializableExtra("Survey");
SurveyQuestions = _survey.getSurveyQuestions();

if (isNetworkConnected() == false) {
    Toast.makeText(AnswerSurveyActivity.this, getString(R.string.NoInternetConnection),
    Toast.LENGTH_LONG).show();
} else {
    onStartDataLoading();
    IsSurveyAnsweredDataLoader Loader = new IsSurveyAnsweredDataLoader(this, _survey.getId(),
    Settings.getLoginUserSerial());
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
        Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    else
        Loader.execute();
}
}

public void BindQuestions(List<QuestionsClass> questions) {
    this.SurveyQuestions = questions;
    if (questions.size() > 0) {
        currentQuestion = 0;
        QuestionsClass question = questions.get(0);
        txtQuestion.setText(question.getName());
        radioChoice1.setText(question.getChoice1());
        radioChoice2.setText(question.getChoice2());
        radioChoice3.setText(question.getChoice3());
        radioChoice4.setText(question.getChoice4());
    } else {
        finish();
    }
}

public void Next(View view) throws JSONException {
    if (checkValue == -1) {
        Toast.makeText(AnswerSurveyActivity.this, getString(R.string.AnswerQuestion),
        Toast.LENGTH_LONG).show();
    } else {
        this.SurveyQuestions.get(currentQuestion).setUserAnswer(checkValue);

        if (SurveyQuestions.size() > currentQuestion + 1) {
            radioGroupAnswers2.clearCheck();
            radioGroupAnswers.clearCheck();
            checkValue = -1;
            currentQuestion++;
            QuestionsClass question = SurveyQuestions.get(currentQuestion);

            int selected = question.getUserAnswer();
            switch (selected) {
                case 1:
                    checkValue = 1;
                    radioGroupAnswers2.clearCheck();
                    radioGroupAnswers.check(R.id.radioChoice1);
                    break;
            }
            case 2:
                checkValue = 2;
                radioGroupAnswers2.clearCheck();
                radioGroupAnswers.check(R.id.radioChoice2);
            }
        }
    }
}
```

```

        break;
    }
    case 3: {
        checkValue = 3;
        radioGroupAnswers.clearCheck();
        radioGroupAnswers2.check(R.id.radioChoice3);
        break;
    }
    case 4: {
        checkValue = 4;
        radioGroupAnswers.clearCheck();
        radioGroupAnswers2.check(R.id.radioChoice4);
        break;
    }
}

txtQuestion.setText(question.getName());
radioChoice1.setText(question.getChoice1());
radioChoice2.setText(question.getChoice2());
radioChoice3.setText(question.getChoice3());
radioChoice4.setText(question.getChoice4());
} else {
    JSONArray arr = new JSONArray();
    int RightAnswers = 0;
    for (int i = 0; i < this.SurveyQuestions.size(); i++) {
        JSONObject obj = new JSONObject();
        obj.put("SurveyId", this._survey.getId());
        obj.put("QuestionId", this.SurveyQuestions.get(i).getId());
        obj.put("StudentId", Settings.getLoginUserSerial());

        if (this.SurveyQuestions.get(i).getRightChoice() == this.SurveyQuestions.get(i).getUserAnswer())
{
            obj.put("QuestionAnswer", 1);
        } else {
            obj.put("QuestionAnswer", -1);
        }

        arr.put(obj);
    }

    if (isNetworkConnected() == false) {
        Toast.makeText(AnswerSurveyActivity.this, getString(R.string.NoInternetConnection),
Toast.LENGTH_LONG).show();
    } else {
        onStartDataLoading();
        AnswerSurveyDataLoader Loader = new AnswerSurveyDataLoader(this, arr.toString());
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
            Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
        else
            Loader.execute();
    }
}
}

public void Previous(View view) {
    this.SurveyQuestions.get(currentQuestion).setUserAnswer(checkValue);

    if (currentQuestion > 0) {
        checkValue = -1;
        currentQuestion--;
        QuestionsClass question = SurveyQuestions.get(currentQuestion);

        int selected = question.getUserAnswer();
        switch (selected) {
            case 1: {
                checkValue = 1;

```

```

        radioGroupAnswers2.clearCheck();
        radioGroupAnswers.check(R.id.radioChoice1);
        break;
    }
    case 2: {
        checkValue = 2;
        radioGroupAnswers2.clearCheck();
        radioGroupAnswers.check(R.id.radioChoice2);
        break;
    }
    case 3: {
        checkValue = 3;
        radioGroupAnswers.clearCheck();
        radioGroupAnswers2.check(R.id.radioChoice3);
        break;
    }
    case 4: {
        checkValue = 4;
        radioGroupAnswers.clearCheck();
        radioGroupAnswers2.check(R.id.radioChoice4);
        break;
    }
}
}

txtQuestion.setText(question.getName());
radioChoice1.setText(question.getChoice1());
radioChoice2.setText(question.getChoice2());
radioChoice3.setText(question.getChoice3());
radioChoice4.setText(question.getChoice4());
}
}

int checkValue = -1;

public void GetChecked(View view) {
    boolean checked = ((RadioButton) view).isChecked();
    switch (view.getId()) {
        case R.id.radioChoice1: {
            checkValue = 1;
            radioGroupAnswers2.clearCheck();
            radioGroupAnswers.check(view.getId());
            break;
        }
        case R.id.radioChoice2: {
            checkValue = 2;
            radioGroupAnswers2.clearCheck();
            radioGroupAnswers.check(view.getId());
            break;
        }
        case R.id.radioChoice3: {
            checkValue = 3;
            radioGroupAnswers.clearCheck();
            radioGroupAnswers2.check(view.getId());
            break;
        }
        case R.id.radioChoice4: {
            checkValue = 4;
            radioGroupAnswers.clearCheck();
            radioGroupAnswers2.check(view.getId());
            break;
        }
    }
}

public void BindData(CoursesClass courseObj) {
    int courseId = courseObj.getId();
    if (courseId < 1) {

```

```

        Toast.makeText(AnswerSurveyActivity.this,
        getString(R.string.ErrorOccured),
        Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(AnswerSurveyActivity.this,
        getString(R.string.DoneSuccessfully),
        Toast.LENGTH_LONG).show();
        Intent intent = new Intent(AnswerSurveyActivity.this, StudentsSurveysCoursesActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        AnswerSurveyActivity.this.finish();
    }
}

public void BindIfAnswered(Users user) {
    if (user.getId() == 1) {
        Toast.makeText(AnswerSurveyActivity.this,
        getString(R.string.alreadyanswersurvey),
        Toast.LENGTH_LONG).show();
        AnswerSurveyActivity.this.finish();
    } else {
        BindQuestions(SurveyQuestions);
    }
}

```

- **LinkSurvey**

```

CoursesList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        if (courses.get(position).getSurveys().size() == 0) {
        } else if (courses.get(position).getSurveys().size() == 1) {
            Intent intent = new Intent(StudentsSurveysCoursesActivity.this, AnswerSurveyActivity.class);
            CoursesClass course = courses.get(position);
            intent.putExtra("course", course);
            intent.putExtra("Survey", course.getSurveys().get(0));
            startActivity(intent);
        } else if (courses.get(position).getSurveys().size() > 1) {
            Intent intent = new Intent(StudentsSurveysCoursesActivity.this, SurveyAnswerActivity.class);
            intent.putExtra("Course", courses.get(position));
            startActivity(intent);
        }
    }
});

if (isNetworkConnected() == false) {
    Toast.makeText(StudentsSurveysCoursesActivity.this,
    getString(R.string.NoInternetConnection),
    Toast.LENGTH_LONG).show();
} else {
    onStartDataLoading();
    GetAllStudentsSurveysCoursesDataLoader Loader = new
    GetAllStudentsSurveysCoursesDataLoader(this, Settings.getLoginUserSerial());
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
        Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    else
        Loader.execute();
}
}

public void BindData(List<CoursesClass> courses) {
    this.courses = courses;
    if (courses.size() > 0) {
        txtNoResult.setVisibility(View.GONE);
        CoursesList.setVisibility(View.VISIBLE);
        StudentsSurveyCoursesListViewAdapter adapter_student = new
        StudentsSurveyCoursesListViewAdapter(StudentsSurveysCoursesActivity.this,
        R.layout.studentssurveyscourses_list_item, courses);
        CoursesList.setAdapter(adapter_student);
    } else {
        CoursesList.setVisibility(View.GONE);
    }
}

```

```
        txtNoResult.setVisibility(View.VISIBLE);
    }
}
}

DefineSurvey

List<ChoicesClass> choices = new ArrayList<ChoicesClass>();
choices.add(new ChoicesClass(1, "Choice One", false));
choices.add(new ChoicesClass(2, "Choice Two", false));

ArrayList<QuestionsClass> questions = new ArrayList<QuestionsClass>();
questions.add(new QuestionsClass(1, "Questio Description 1?", choices));
questions.add(new QuestionsClass(2, "Questio Description 2?", choices));
questions.add(new QuestionsClass(3, "Questio Description 3?", choices));
questions.add(new QuestionsClass(4, "Questio Description 4?", choices));
questions.add(new QuestionsClass(5, "Questio Description 5?", choices));
questions.add(new QuestionsClass(6, "Questio Description 6?", choices));

MCQFormCustomListViewAdapter adapter_student = new
MCQFormCustomListViewAdapter(MCQFormActivity.this, R.layout.mcqform_list_item, questions);
questionList.setAdapter(adapter_student);
}

public void Submit(View view) {
    Intent intent = new Intent(MCQFormActivity.this, AnswerStatisticsActivity.class);
    startActivity(intent);
}
```

- **AppSurvey**

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    if (courses.get(position).getSurveys().size() == 0) {
        Intent intent = new Intent(FacultyMemberApproveSurveyActivity.this,
CourseDetailsActivity.class);
        intent.putExtra("From", "FacultyMember");
        intent.putExtra("course", courses.get(position));
        startActivity(intent);
    } else if (courses.get(position).getSurveys().size() == 1) {
        Intent intent = new Intent(FacultyMemberApproveSurveyActivity.this,
CourseDetailsActivity.class);
        intent.putExtra("From", "FacultyMember");
        intent.putExtra("course", courses.get(position));
        startActivity(intent);
    } else if (courses.get(position).getSurveys().size() > 1) {
        Intent intent = new Intent(FacultyMemberApproveSurveyActivity.this,
SurveyFacultyMemberApproveMain.class);
        intent.putExtra("Course", courses.get(position));
        startActivity(intent);
    }
}
});

if (isNetworkConnected() == false) {
    Toast.makeText(FacultyMemberApproveSurveyActivity.this,
getString(R.string.NoInternetConnection), Toast.LENGTH_LONG).show();
} else {
    onStartDataLoading();
    GetFacultyMemberCoursesToApproveDataLoader Loader = new
GetFacultyMemberCoursesToApproveDataLoader(this);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
        Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    else
        Loader.execute();
}
}
```

```

public void BindData(List<CoursesClass> courses) {
    this.courses = courses;
    if (courses.size() > 0) {
        txtNoResult.setVisibility(View.GONE);
        CoursesList.setVisibility(View.VISIBLE);
        FacultyMembersApproveSurveyListAdapter adapter_student = new
FacultyMembersApproveSurveyListAdapter(FacultyMemberApproveSurveyActivity.this,
R.layout.facultymembersapprovesurvey_list_item, courses);
        CoursesList.setAdapter(adapter_student);
    } else {
        CoursesList.setVisibility(View.GONE);
        txtNoResult.setVisibility(View.VISIBLE);
    }
}
}

• View_Results
SurveyId = Integer.parseInt(getIntent().getStringExtra("SurveyId"));

if (isNetworkConnected() == false) {
    Toast.makeText(StatisticsDetailsActivity.this, getString(R.string.NoInternetConnection),
Toast.LENGTH_LONG).show();
} else {
    onStartDataLoading();
    GetAllStatisticsForOneSurveyDataLoader Loader = new
GetAllStatisticsForOneSurveyDataLoader(this, SurveyId);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
        Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    else
        Loader.execute();
}
}

public void BindData(List<StatisticsClass> statistics) {
    if (statistics.size() > 0) {
        txtNoResult.setVisibility(View.GONE);
        CoursesList.setVisibility(View.VISIBLE);
        StatisticsDetailsListAdapter adapter_student = new
StatisticsDetailsListAdapter(StatisticsDetailsActivity.this, R.layout.statisticdetails_list_item, statistics);
        CoursesList.setAdapter(adapter_student);
    } else {
        CoursesList.setVisibility(View.GONE);
        txtNoResult.setVisibility(View.VISIBLE);
    }
}
}

public StatisticsDetailsListAdapter(Context context, int resourceId, List<StatisticsClass> statistics) {
    super(context, resourceId, statistics);
    this.context = context;
}

private class ViewHolder {
    TextView txtName, txtTrue, txtFalse, txtTrueValue, txtFalseValue;
}

public View getView(int position, View convertView, ViewGroup parent) {
    final StatisticsClass statistic = (StatisticsClass) getItem(position);
    LayoutInflator mInflater = (LayoutInflator)
context.getSystemService(Activity.LAYOUT_INFLATER_SERVICE);

    if (convertView == null) {
        convertView = mInflater.inflate(R.layout.statisticdetails_list_item, null);
        holder = new ViewHolder();
        holder.txtName = (TextView) convertView.findViewById(R.id.txtName);
    }
}

```

```

holder.txtTrue = (TextView) convertView.findViewById(R.id.txtTrue);
holder.txtFalse = (TextView) convertView.findViewById(R.id.txtFalse);
holder.txtTrueValue = (TextView) convertView.findViewById(R.id.txtTrueValue);
holder.txtFalseValue = (TextView) convertView.findViewById(R.id.txtFalseValue);
convertView.setTag(holder);
} else {
    holder = (ViewHolder) convertView.getTag();
}
holder.txtName.setText(statistic.getQuestion());

double rightPercent = 0;
double wrongPercent = 0;
if (statistic.getNumberofStudent() == 0) {

} else {
    rightPercent           = Double.valueOf(statistic.getRightAnswers()) / Double.valueOf(statistic.getNumberofStudent()) * Double.valueOf(100);
    wrongPercent           = Double.valueOf(statistic.getWrongAnswers()) / Double.valueOf(statistic.getNumberofStudent()) * Double.valueOf(100);
}

holder.txtTrueValue.setText(String.valueOf(statistic.getRightAnswers()) + " (" + String.format("%.0f", rightPercent) + "%)");
holder.txtFalseValue.setText(String.valueOf(statistic.getWrongAnswers()) + " (" + String.format("%.0f", wrongPercent) + "%)");

holder.txtTrue.setText(".");
for (int i = 0; i < statistic.getRightAnswers(); i++) {
    holder.txtTrue.setText(holder.txtTrue.getText() + ".");
}

holder.txtFalse.setText(".");
for (int i = 0; i < statistic.getWrongAnswers(); i++) {
    holder.txtFalse.setText(holder.txtFalse.getText() + ".");
}

return convertView;
}
}

• SubmitSurvey
_course = (CoursesClass) getIntent().getSerializableExtra("course");
_survey = (SurveyModel) getIntent().getSerializableExtra("Survey");
SurveyQuestions = _survey.getSurveyQuestions();

if (isNetworkConnected() == false) {
    Toast.makeText(AnswerSurveyActivity.this, getString(R.string.NoInternetConnection),
    Toast.LENGTH_LONG).show();
} else {
    onStartDataLoading();
    IsSurveyAnsweredDataLoader Loader = new IsSurveyAnsweredDataLoader(this, _survey.getId(),
    Settings.getLoginUserSerial());
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
        Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    else
        Loader.execute();
}
}

public void BindQuestions(List<QuestionsClass> questions) {
    this.SurveyQuestions = questions;
    if (questions.size() > 0) {
        currentQuestion = 0;
        QuestionsClass question = questions.get(0);
        txtQuestion.setText(question.getName());
    }
}

```

```

        radioChoice1.setText(question.getChoice1());
        radioChoice2.setText(question.getChoice2());
        radioChoice3.setText(question.getChoice3());
        radioChoice4.setText(question.getChoice4());
    } else {
        finish();
    }
}

public void Next(View view) throws JSONException {
    if (checkValue == -1) {
        Toast.makeText(AnswerSurveyActivity.this,
                      getString(R.string.AnswerQuestion),
                      Toast.LENGTH_LONG).show();
    } else {
        this.SurveyQuestions.get(currentQuestion).setUserAnswer(checkValue);

        if (SurveyQuestions.size() > currentQuestion + 1) {
            radioGroupAnswers2.clearCheck();
            radioGroupAnswers.clearCheck();
            checkValue = -1;
            currentQuestion++;
            QuestionsClass question = SurveyQuestions.get(currentQuestion);

            int selected = question.getUserAnswer();
            switch (selected) {
                case 1: {
                    checkValue = 1;
                    radioGroupAnswers2.clearCheck();
                    radioGroupAnswers.check(R.id.radioChoice1);
                    break;
                }
                case 2: {
                    checkValue = 2;
                    radioGroupAnswers2.clearCheck();
                    radioGroupAnswers.check(R.id.radioChoice2);
                    break;
                }
                case 3: {
                    checkValue = 3;
                    radioGroupAnswers.clearCheck();
                    radioGroupAnswers2.check(R.id.radioChoice3);
                    break;
                }
                case 4: {
                    checkValue = 4;
                    radioGroupAnswers.clearCheck();
                    radioGroupAnswers2.check(R.id.radioChoice4);
                    break;
                }
            }
        }

        txtQuestion.setText(question.getName());
        radioChoice1.setText(question.getChoice1());
        radioChoice2.setText(question.getChoice2());
        radioChoice3.setText(question.getChoice3());
        radioChoice4.setText(question.getChoice4());
    } else {
        JSONArray arr = new JSONArray();
        int RightAnswers = 0;
        for (int i = 0; i < this.SurveyQuestions.size(); i++) {
            JSONObject obj = new JSONObject();
            obj.put("SurveyId", this._survey.getId());
            obj.put("QuestionId", this.SurveyQuestions.get(i).getId());
            obj.put("StudentId", Settings.getLoginUserSerial());

            if (this.SurveyQuestions.get(i).getRightChoice() == this.SurveyQuestions.get(i).getUserAnswer())
    {

```

```

        obj.put("QuestionAnswer", 1);
    } else {
        obj.put("QuestionAnswer", -1);
    }

    arr.put(obj);
}

if (isNetworkConnected() == false) {
    Toast.makeText(AnswerSurveyActivity.this, getString(R.string.NoInternetConnection),
    Toast.LENGTH_LONG).show();
} else {
    onStartDataLoading();
    AnswerSurveyDataLoader Loader = new AnswerSurveyDataLoader(this, arr.toString());
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
        Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
    else
        Loader.execute();
}
}

public void Previous(View view) {
    this.SurveyQuestions.get(currentQuestion).setUserAnswer(checkValue);

    if (currentQuestion > 0) {
        checkValue = -1;
        currentQuestion--;
        QuestionsClass question = SurveyQuestions.get(currentQuestion);

        int selected = question.getUserAnswer();
        switch (selected) {
            case 1: {
                checkValue = 1;
                radioGroupAnswers2.clearCheck();
                radioGroupAnswers.check(R.id.radioChoice1);
                break;
            }
            case 2: {
                checkValue = 2;
                radioGroupAnswers2.clearCheck();
                radioGroupAnswers.check(R.id.radioChoice2);
                break;
            }
            case 3: {
                checkValue = 3;
                radioGroupAnswers.clearCheck();
                radioGroupAnswers2.check(R.id.radioChoice3);
                break;
            }
            case 4: {
                checkValue = 4;
                radioGroupAnswers.clearCheck();
                radioGroupAnswers2.check(R.id.radioChoice4);
                break;
            }
        }
    }

    txtQuestion.setText(question.getName());
    radioChoice1.setText(question.getChoice1());
    radioChoice2.setText(question.getChoice2());
    radioChoice3.setText(question.getChoice3());
    radioChoice4.setText(question.getChoice4());
}
}

```

```

int checkValue = -1;

public void GetChecked(View view) {
    boolean checked = ((RadioButton) view).isChecked();
    switch (view.getId()) {
        case R.id.radioChoice1: {
            checkValue = 1;
            radioGroupAnswers2.clearCheck();
            radioGroupAnswers.check(view.getId());
            break;
        }
        case R.id.radioChoice2: {
            checkValue = 2;
            radioGroupAnswers2.clearCheck();
            radioGroupAnswers.check(view.getId());
            break;
        }
        case R.id.radioChoice3: {
            checkValue = 3;
            radioGroupAnswers.clearCheck();
            radioGroupAnswers2.check(view.getId());
            break;
        }
        case R.id.radioChoice4: {
            checkValue = 4;
            radioGroupAnswers.clearCheck();
            radioGroupAnswers2.check(view.getId());
            break;
        }
    }
}

public void BindData(CoursesClass courseObj) {
    int courseId = courseObj.getId();
    if (courseId < 1) {
        Toast.makeText(AnswerSurveyActivity.this,
                      getString(R.string.ErrorOccured),
                      Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(AnswerSurveyActivity.this,
                      getString(R.string.DoneSuccessfully),
                      Toast.LENGTH_LONG).show();
        Intent intent = new Intent(AnswerSurveyActivity.this, StudentsSurveysCoursesActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        AnswerSurveyActivity.this.finish();
    }
}

public void BindIfAnswered(Users user) {
    if (user.getId() == 1) {
        Toast.makeText(AnswerSurveyActivity.this,
                      getString(R.string.alreadyansw ersurvey),
                      Toast.LENGTH_LONG).show();
        AnswerSurveyActivity.this.finish();
    } else {
        BindQuestions(SurveyQuestions);
    }
}

```

- **AllocateCourse**

```

public void CreateCourse(View view) {
    String CourseName = txtCourseName.getText().toString().trim();
    String SectionNumber = txtSectionNumber.getText().toString().trim();
    String NumberOfLectures = txtNumberOfLectures.getText().toString().trim();
    String FacultyMemberName = txtFacultyMember.getText().toString().trim();
    String CourseId = txtCourseId.getText().toString().trim();

```

```

        if (TextUtils.isEmpty(CourseName)) {
            Toast.makeText(CreateCourseActivity.this,      getString(R.string.CourseName_Error_field_required),
Toast.LENGTH_LONG).show();
            txtCourseName.requestFocus();
            return;
        }

        if (TextUtils.isEmpty(CourseId)) {
            Toast.makeText(CreateCourseActivity.this,      getString(R.string.CourseId_Error_field_required),
Toast.LENGTH_LONG.show();
            txtCourseId.requestFocus();
            return;
        }

        if (TextUtils.isEmpty(SectionNumber)) {
            Toast.makeText(CreateCourseActivity.this,  getString(R.string.SectionNumber_Error_field_required),
Toast.LENGTH_LONG.show();
            txtSectionNumber.requestFocus();
            return;
        }

        if (TextUtils.isEmpty(NumberOfLectures)) {
            Toast.makeText(CreateCourseActivity.this,
getString(R.string.NumberOfLectures_Error_field_required), Toast.LENGTH_LONG).show();
            txtNumberOfLectures.requestFocus();
            return;
        }

        boolean FacultyMemberExist = false;
        for (IdName d : this.FacultyMembers) {
            if (d.getName() != null && d.getName().equals(FacultyMemberName)) {
                FacultyMemberExist = true;
            }
        }

        if (FacultyMemberExist == false || FacultyMembersId == 0) {
            Toast.makeText(CreateCourseActivity.this,  getString(R.string.FacultyMember_Error_field_required),
Toast.LENGTH_LONG).show();
            txtFacultyMember.requestFocus();
            return;
        }

        try {
            obj.put("FacultyMemberId", FacultyMembersId);
            obj.put("Name", CourseName);
            obj.put("SectionNumber", SectionNumber);
            obj.put("NoOfLectures", NumberOfLectures);
            obj.put("CourseId", CourseId);
            obj.put("AdminId", Settings.getLoginUserSerial());
        } catch (JSONException e1) {
            e1.printStackTrace();
        }

        // Log.i("obj", obj.toString());

        /* Intent email = new Intent(Intent.ACTION_SEND);
        email.putExtra(Intent.EXTRA_EMAIL, new String[]{ "aliabdelhady25@yahoo.com" });
        email.putExtra(Intent.EXTRA_SUBJECT, "sub");
        email.putExtra(Intent.EXTRA_TEXT, obj.toString());
        email.setType("message/rfc822");
        startActivity(Intent.createChooser(email, "Select Email Client"));*/
    }

    if (isNetworkConnected() == false) {
        Toast.makeText(CreateCourseActivity.this,      getString(R.string.NoInternetConnection),
Toast.LENGTH_LONG).show();
    } else {

```

```

onStartDataLoading();
AddCourseDataLoader Loader = new AddCourseDataLoader(this, obj.toString());
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
    Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
else
    Loader.execute();
}

public void BindData(CoursesClass courseObj) {
    int courseId = courseObj.getId();
    if (courseId < 1) {
        Toast.makeText(CreateCourseActivity.this,
                      getString(R.string.coursealreadyadded),
                      Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(CreateCourseActivity.this,
                      getString(R.string.SuccessfullyCourseAdded),
                      Toast.LENGTH_LONG).show();
        Intent intent = new Intent(CreateCourseActivity.this, AdminHomeActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        CreateCourseActivity.this.finish();
    }
}

public void BindFacultyMembers(ArrayList<IdName> facultyMembers) {
    this.FacultyMembers = facultyMembers;
    final AutoCompleteTextViewCustomerAdapter adapter = new
    AutoCompleteTextViewCustomerAdapter(this, R.layout.autocompletelistview_item, facultyMembers);

    txtFacultyMember.setThreshold(1);
    txtFacultyMember.setAdapter(adapter);

    txtFacultyMember.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            FacultyMembersId = adapter.getItem(position).getId();
        }
    });
}
}

```

- **Attach_Students**

```

public void Link(View view) {
    try {
        userId = this.users.getSelectedItemPosition().getId();
        courseId = this.courses.getSelectedItemPosition().getId();

        if (courseId == 0) {
            Toast.makeText(LinkCourseWithStudentActivity.this,
                          getString(R.string.CourseName_Error_field_required), Toast.LENGTH_LONG).show();
            txtCourse.requestFocus();
            return;
        }

        if (userId == 0) {
            Toast.makeText(LinkCourseWithStudentActivity.this,
                          getString(R.string.User_Error_field_required), Toast.LENGTH_LONG).show();
            txtUser.requestFocus();
            return;
        }

        JSONObject obj = new JSONObject();
        obj.put("CourseId", courseId);
        obj.put("StudentId", userId);
        obj.put("AdminId", Settings.getLoginUserSerial());
    }
}

```

```

        if (isNetworkConnected() == false) {
            Toast.makeText(LinkCourseWithStudentActivity.this, getString(R.string.NoInternetConnection),
            Toast.LENGTH_LONG).show();
        } else {
            onStartDataLoading();
            LinkUserToCourseDataLoader Loader = new LinkUserToCourseDataLoader(this, obj.toString());
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
                Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
            else
                Loader.execute();
        }
    } catch (Exception ex) {

    }
}

public void BindUsers(final ArrayList<IdName> users) {
    this.users = users;

    String[] usersString = new String[users.size()];
    for (int i = 0; i < users.size(); i++) {
        usersString[i] = users.get(i).getName();
    }

    ArrayAdapter aa = new ArrayAdapter(this, android.R.layout.simple_spinner_item, usersString);
    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    txtUser.setAdapter(aa);

    if (isNetworkConnected() == false) {
        Toast.makeText(LinkCourseWithStudentActivity.this, getString(R.string.NoInternetConnection),
        Toast.LENGTH_LONG).show();
    } else {
        onStartDataLoading();
        GetAllCoursesToLinkDataLoader Loader = new GetAllCoursesToLinkDataLoader(this);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
            Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
        else
            Loader.execute();
    }
}

public void BindCourses(final ArrayList<IdName> courses) {
    this.courses = courses;

    String[] coursesString = new String[courses.size()];
    for (int i = 0; i < courses.size(); i++) {
        coursesString[i] = courses.get(i).getName();
    }

    ArrayAdapter aa = new ArrayAdapter(this, android.R.layout.simple_spinner_item, coursesString);
    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    txtCourse.setAdapter(aa);
}

public void BindDataAfterLink(int id) {
    if (id < 1) {
        Toast.makeText(this, getString(R.string.useralreadyLinked), Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(this, getString(R.string.SuccessfullyUserLinked), Toast.LENGTH_LONG).show();
    }
}

```

- Send/Receive Notifications**

```

public void onStart(Intent intent, int startId) {

```

```

// TODO Auto-generated method stub
super.onStart(intent, startId);
Log.i("Service", "onStart");
Timer timer = new Timer();
//Dept_ID = intent.getExtras().getString("Dept_ID");
TimerTask hourlyTask = new TimerTask() {
    @Override
    public void run() {
        Log.i("Service", "NetWorkConnected");
        if (isNetworkConnected() == true) {
            try {
                int id = 0;
                int ser = Settings.getLoginUserSerial();
                String url = Settings.getDTOServiceUrl() + "/GetAllNotifications/" + String.valueOf(ser);
                Log.i("Url Service", url);

                HttpGet request = null;
                try {
                    StrictMode.ThreadPolicy policy = StrictMode.ThreadPolicy.Builder().permitAll().build();
                    StrictMode.setThreadPolicy(policy);
                    request = new HttpGet(new URI(url));
                } catch (URISyntaxException e1) {
                    e1.printStackTrace();
                }
                @SuppressWarnings({"resource", "deprecation"})
                HttpClient client = new DefaultHttpClient();
                HttpResponse response = client.execute(request);
                JSONArray jsa = null;
                if (response != null) {
                    String jsonResponse = response.getEntity();
                    Log.i("jsonResponse", jsonResponse);
                    jsa = new JSONArray(jsonResponse);
                }
                String line;
                NotificationManager notificationManager =
                        (NotificationManager)
                getSystemService(getApplicationContext().NOTIFICATION_SERVICE);

                for (int i = 0; i < jsa.length(); i++) {
                    JSONObject jo = (JSONObject) jsa.get(i);
                    int courseId = jo.getInt("Id");
                    String Name = jo.getString("Name");

                    int NoOfLectures = 0;
                    JSONObject FacultyMemberOBJ = new JSONObject(jo.getString("FacultyMember"));
                    String FacultyMemberFirstName = FacultyMemberOBJ.getString("FirstName");
                    String FacultyMemberLastName = FacultyMemberOBJ.getString("LastName");
                    String FacultyMemberMiddleName = FacultyMemberOBJ.getString("MiddleName");

                    String LectureName = "";

                    List<SurveyModel> Surveys = new ArrayList<SurveyModel>();
                    JSONArray SurveysArr = new JSONArray(jo.getString("Surveys"));
                    for (int y = 0; y < SurveysArr.length(); y++) {
                        SurveyModel surveyModel = new SurveyModel();
                        JSONObject objSurveys = (JSONObject) SurveysArr.get(y);
                        surveyModel.setId(objSurveys.getInt("Id"));
                        surveyModel.setApproved(objSurveys.getBoolean("Approved"));
                        surveyModel.setCourseId(objSurveys.getInt("CourseId"));
                        surveyModel.setDeadlineDate(objSurveys.getString("DeadlineDate"));

                        if (y == 0)
                            LectureName = objSurveys.getString("LectureName");
                        surveyModel.setLectureName(objSurveys.getString("LectureName"));
                    }
                }
            }
        }
    }
}

```

```

        surveyModel.setLectureNumber(objSurveys.getInt("LectureNumber"));
        NoOfLectures = objSurveys.getInt("LectureNumber");
        surveyModel.setMessage(objSurveys.getString("Message"));
        surveyModel.setMessageSent(objSurveys.getBoolean("MessageSent"));
        surveyModel.setQuestionAdded(objSurveys.getBoolean("QuestionAdded"));

    surveyModel.setSentToFacultyMember(objSurveys.getBoolean("SentToFacultyMember"));

        Surveys.add(surveyModel);
    }

    String CreationDate = jo.getString("CreationDate");
    int AdminId = jo.getInt("AdminId");
    int SectionNumber = jo.getInt("SectionNumber");
    int FacultyMemberId = jo.getInt("FacultyMemberId");

    int NotificatioId = jo.getInt("NotificatioId");
    int NotificatioType = jo.getInt("NotificatioType");

    String CourseId = jo.getString("CourseId");
    CoursesClass course = new CoursesClass(courseId, AdminId, FacultyMemberId, Name,
    FacultyMemberFirstName, CreationDate, FacultyMemberMiddleName, FacultyMemberLastName,
    SectionNumber, NoOfLectures, Surveys, CourseId);
    course.setNotificatioId(NotificatioId);
    course.setNotificatioType(NotificatioType);

    Intent myIntent = new Intent(getApplicationContext(), NotificationDetailsActivity.class);
    Notification myNotification = new Notification(R.drawable.ic_launcher, "Notification!",
    System.currentTimeMillis());

    myIntent.putExtra("course", course);

    myIntent.setAction("actionstring" + System.currentTimeMillis());
    PendingIntent pendingIntent = PendingIntent.getActivity(getApplicationContext(), 0,
    myIntent, Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
    myNotification.defaults |= Notification.DEFAULT_SOUND;
    myNotification.flags |= Notification.FLAG_AUTO_CANCEL;

    RemoteViews contentView = new RemoteViews(getApplicationContext(),
    R.layout.customnotification);
    contentView.setImageResource(R.id.image, R.drawable.ic_launcher);
    if (NotificatioType == 1) { // Admin
        contentView.setTextViewText(R.id.text,
        getString(R.string.youhavebeenenrolledasadminincourse) + " " + Name + " " + SectionNumber);
    } else if (NotificatioType == 2) { // Coordinator
        contentView.setTextViewText(R.id.text,
        getString(R.string.youhavebeenenrolledascoordinatorcourse) + " " + Name + " " + SectionNumber);
    } else if (NotificatioType == 3) { // Asker
        contentView.setTextViewText(R.id.text,
        getString(R.string.youhavebeenassignedasaskerstudents) + " " + Name + " " + SectionNumber);
    } else if (NotificatioType == 4) { // Delete Question
        contentView.setTextViewText(R.id.text,
        getString(R.string.yourQuestionsDeleted).replace("{0}", Name + "" + LectureName + " , " + NoOfLectures
        + ""));
    } else if (NotificatioType == 5) { // received MCQ form
        contentView.setTextViewText(R.id.text,
        getString(R.string.YouhavebeenreceivedMCQformtoreviewing) + " " + Name + " " + SectionNumber);
    } else if (NotificatioType == 6) { // received MCQ form
        contentView.setTextViewText(R.id.text,
        getString(R.string.YouhavebeenreceivedcompletedMCQform) + " " + Name + " " + SectionNumber);
    } else if (NotificatioType == 7) { // received MCQ form
        contentView.setTextViewText(R.id.text, " " + Name + " " + " " + " " + " " +
        getString(R.string.MCQsurveyhavebeenpublishedbyfacultymember));
    }

    myNotification.contentIntent = pendingIntent;
    myNotification.contentView = contentView;
    myIntent.setData(Uri.parse("custom://" + System.currentTimeMillis()));

```

```

        notificationManager.notify(id, myNotification);
        id++;
    }

} catch (JSONException e4) {
    e4.printStackTrace();
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

};

timer.schedule(hourlyTask, 01, 120000);
}

@Override
public IBinder onBind(Intent arg0) {
    return null;
}

protected String getResponse(HttpEntity entity) {
    String response = "";
    try {
        int length = (int) entity.getContentLength();
        StringBuffer sb = new StringBuffer(length);
        InputStreamReader isr = new InputStreamReader(entity.getContent(), "UTF-8");
        char buff[] = new char[length];
        int cnt;
        while ((cnt = isr.read(buff, 0, length - 1)) > 0) {
            sb.append(buff, 0, cnt);
        }
        response = sb.toString();
        isr.close();
    } catch (IOException ioe) {
        ioe.printStackTrace();
    }
    return response;
}
}

```

- **SendMessages**

```

courseName = getIntent().getStringExtra("CourseName");
_survey = (SurveyModel) getIntent().getSerializableExtra("Survey");
txtCourseName.setText(courseName);
for (int i = 0; i < _survey.getSurveyUsers().size(); i++) {
    TextView text = new TextView(this);
    text.setTextSize(18);
    text.setText(_survey.getSurveyUsers().get(i).getFirstName() +
                " " +
                _survey.getSurveyUsers().get(i).getMiddleName() + " " +
                _survey.getSurveyUsers().get(i).getUniversityNumber());
    lin_Users.addView(text);
}
public void Send(View view) {
    try {
        String Message = txtMessage.getText().toString().trim();

```

```

        if (TextUtils.isEmpty(Message)) {
            DisplayToast(getString(R.string.Message_Error_field_required));
            return;
        }

        JSONObject object = new JSONObject();
        object.put("Message", Message);
        object.put("SurveyId", _survey.getId());

        if (isNetworkConnected() == false) {
            Toast.makeText(SendMessageActivity.this, getString(R.string.NoInternetConnection),
                    Toast.LENGTH_LONG).show();
        } else {
            onStartDataLoading();

            AddCoordinatorMessagesDataLoader Loader = new AddCoordinatorMessagesDataLoader(this,
                    object.toString());

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB)
                Loader.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
            else
                Loader.execute();
        }
    } catch (Exception ex) {
}
}

public void BindData(CoursesClass courseObj) {
    int courseId = courseObj.getId();
    if (courseId < 1) {
        Toast.makeText(SendMessageActivity.this, getString(R.string.ErrorOccured),
                Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(SendMessageActivity.this, getString(R.string.SuccessfullyMessageSent),
                Toast.LENGTH_LONG).show();
        Intent intent = new Intent(SendMessageActivity.this, CoordinatorHomeActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
        SendMessageActivity.this.finish();
    }
}
}

```

Queries:

- 1- Query about lecture name and look if approved = 1 not approved =NULL.

The screenshot shows the Object Explorer on the left with various database objects like Keys, Constraints, Triggers, etc. A query window titled 'SQLQuery2.sql' is open with the following code:

```
Object Explorer
Connect ▾
HP\SQLEXPRESS.MCQ4U - dbo.Surveys SQLQuery2.sql - H...4U (HP\HP-PC (52))*
Select LectureName , Approved From Surveys;
```

The results grid shows the following data:

LectureName	Approved
chapter1	1
chapter 2	NULL
introductio...	1
chapter 3	NULL
chapter 3	1
chapt...	1

A message at the bottom says 'Query executed successfully.'

- 2- Query update student Nowair for Id university from 43026077 to 434055088 .

The screenshot shows two windows. The top window has the following code:

```
HP\SQLEXPRESS.MCQ4U - dbo.Surveys SQLQuery2.sql - H...4U (HP\HP-PC (52))*
Update Users Set UniversityNumber=434055088 Where FirstName='Nowair' And LastName='Alotaibi' ;
```

The bottom window has the following code:

```
HP\SQLEXPRESS.MCQ4U - dbo.Surveys SQLQuery2.sql - H...4U (HP\HP-PC (52))*
Select * from Users;
```

Both windows show the same results grid with data rows numbered 5 to 11. Rows 8 and 9 are highlighted with red boxes. Row 8 contains 'Nowair' and row 9 contains 'Alotaibi'. The bottom window also shows a message 'Query executed successfully.'

- 3- Query display section number, name and when creation date where only course= IS 200

HP\SQLEXPRESS.MCQ4U - dbo.Surveys SQLQuery2.sql - H...4U (HP\HP-PC (52))* HP\SQLEXPRESS.MCQ4U - dbo.Users

```
Select SectionNumber, Name , CreationDate from Courses where CourseId='IS200';
```

100 %

Results Messages

	SectionNumber	Name	CreationDate
1	372	information system	2017-12-05 04:48:47.810
2	374	Information system	2017-12-07 19:14:50.280

Query executed successfully.

| HP\SQLEXPRESS (10.0 SP1) | HP\HP-PC (52) | MCQ4U | 00:00:0

```

SELECT [Courses].[Id]
,[Courses].[CourseId]
,[Courses].[Name]
,[Courses].[AdminId]
,[Courses].[FacultyMemberId]
,[Courses].[CreationDate]
,[Courses].[SectionNumber]
,[Courses].[NoOfLectures]
,Users.FirstName AS AdminFirstName
,Users.LastName AS AdminLastName
,Users.MiddleName AS AdminMiddleName
,Users.Email AS AdminEmail
,Users_1.FirstName AS FacultyMemberFirstName
,Users_1.LastName AS FacultyMemberLastName
,Users_1.MiddleName AS FacultyMemberMiddleName
,Users_1.Email AS FacultyMemberEmail
FROM [dbo].[Courses]
INNER JOIN Users ON [Courses].[AdminId] = [Users].[Id]
INNER JOIN Users AS Users_1 ON [Courses].[FacultyMemberId] = Users_1.Id
where [Courses].[Id] in (select Courses.Id from Courses
INNER JOIN Surveys ON [Courses].[Id] = [Surveys].[CourseId]
where Surveys.Approved = 1 and FacultyMemberId = @StudentId
)

```

```

INSERT INTO [MCQ4U].[dbo].[SurveyAnswers]
([SurveyId]
,[QuestionId]
,[QuestionAnswer]
,[StudentId])
VALUES
(@SurveyId
,@QuestionId
,@QuestionAnswer
,@StudentId)

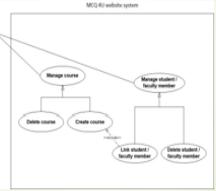
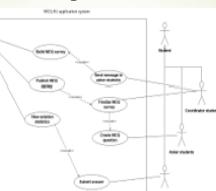
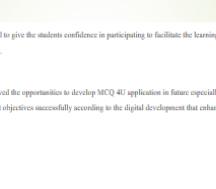
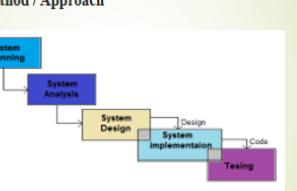
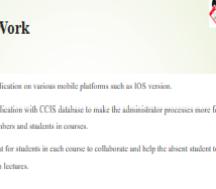
set @Id = SCOPE_IDENTITY()
END

BEGIN
INSERT INTO [dbo].[Courses]
([CourseId]
,[Name]
,[AdminId]
,[FacultyMemberId]
,[CreationDate]
,[SectionNumber]
,[NoOfLectures])
VALUES
(@CourseId
,@Name
,@AdminId
,@FacultyMemberId
,GETDATE()
,@SectionNumber
,@NoOfLectures)

set @Id = SCOPE_IDENTITY()
END

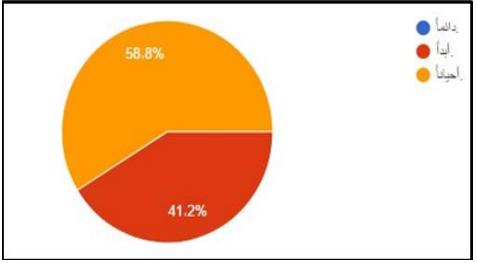
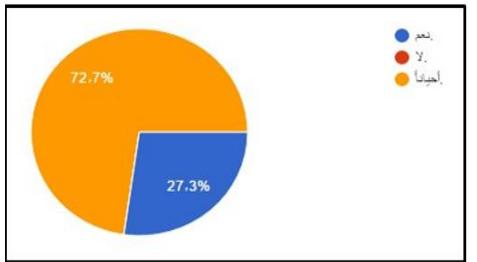
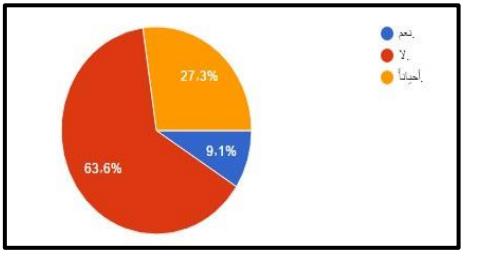
```

B. Presentation Slides

1  <table border="1" data-bbox="247 460 512 527"><tr><th>Student name</th><th>Student ID</th></tr><tr><td>Rawan Naf Alnazi</td><td>414024002</td></tr><tr><td>Nearah Salim Alharqi</td><td>434025377</td></tr><tr><td>Navar Naser Alstabi</td><td>414026077</td></tr></table>	Student name	Student ID	Rawan Naf Alnazi	414024002	Nearah Salim Alharqi	434025377	Navar Naser Alstabi	414026077	2  Project Scope "MCQ 4U" is an Android mobile application which serves CCIS Imam University, the female student section, and Faculty Members. "MCQ 4U" application supports only English language, Android devices. In the first version, the application does not support other languages such as Arabic, and other colleges in Imam University.	3  Objectives There is a lot of software that generates online MCQs for students, but only accessed and used by the Faculty Member, which does not support the "MCQ 4U" idea, that allow students themselves to generate the questions. MCQ 4U is an educational application which helps students to study in less time by solving multiple-choice questions (MCQs) that generated by their classmates using the application. The application serves Faculty Member, by producing statistics about the understanding level of students after each lecture depending on their answers on MCQs questions.	4  Admin use case diagram MCQ 4U website system Admin Manage course Delete course Create course Link student / Faculty member Delete student / Faculty member
Student name	Student ID										
Rawan Naf Alnazi	414024002										
Nearah Salim Alharqi	434025377										
Navar Naser Alstabi	414026077										
5  Users use case diagram MCQ 4U website system Faculty Member Student Coordinator Student Asker Student Respondent Student Administrator Delete course Create course Link student / Faculty member Delete student / Faculty member Generate MCQ Answer MCQ Evaluate MCQ Statistics Coordinator student Asker student Respondent student	6  Conclusion MCQ 4U is developed to give the students confidence in participating to facilitate the learning process to their faculty members. Team members observed the opportunities to develop MCQ 4U application in future especially after achieved these project objectives successfully according to the digital development that enhances the education technology.	7  Problem statement Students are suffering in finding enough time in reviewing the received lectures, due to their preoccupation with projects. And they have problems in managing their time to study the materials they received. Therefore, lectures will accumulate and result in less understanding of the course. In order to improve and increase the level of students' understanding of the received lectures, the "MCQs 4U" application allows students to review the received lectures by answering the MCQ's questions produced by the application and generated by the students. The Faculty Member can use the application to assess the students understanding of the material, which will help before writing the exam questions.	8  Method / Approach System Planning → System Analysis → System Design → System implementation → Code → Testing								
9 Obstacles Connecting Administrator's website pages to the Android environments. Programming the dynamic way that Students can change roles between Coordinator Student, Asker Student, or Respondent Student each lecture depending on the Faculty Member choice. Programming different notifications to the different Students change synchronization roles.	10  Obstacles Developing the application and documenting changes in parallel to handle the short time problem. The team members have different responsibilities and commitments to various courses that affect the project's progress. Learning to develop for Android OS environment using different sources to accomplish analysis requirements.	11  Future Work Provide MCQ 4U application on various mobile platforms such as iOS version. Linking MCQ 4U application with CCIS database to make the administrator processes more formal such as linking faculty members and students in courses. Providing a public that for students in each course to collaborate and help the absent student to cover the misunderstood parts in lectures.	12  Thank you for listening								
13 			7								

C. Miscellaneous

Appendix A: Student questionnaire

<p>Q1: In students' questionnaire, do you have a time to review the previous lecture periodically?</p> <p>41.2% of students do not have enough time to review the received lectures periodically. See Figure 55.</p>	 <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>نعم</td><td>58.8%</td></tr><tr><td>لا</td><td>41.2%</td></tr></tbody></table>	Response	Percentage	نعم	58.8%	لا	41.2%		
Response	Percentage								
نعم	58.8%								
لا	41.2%								
<p>Q2: In Faculty Member' questionnaire, do you have difficulties to review the previous lecture with students?</p> <p>27.3% of Faculty Members find difficulty to review the last lecture and connect it with the current lecture's content, and 72.7% of Faculty Member sometimes suffer from this problem. See Figure 56.</p>	 <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>نعم</td><td>72.7%</td></tr><tr><td>لا</td><td>27.3%</td></tr></tbody></table>	Response	Percentage	نعم	72.7%	لا	27.3%		
Response	Percentage								
نعم	72.7%								
لا	27.3%								
<p>Q3: In Faculty Member' questionnaire, do you have time to review the topics before any exam?</p> <p>There are 63.6% of Faculty Members do not have enough time to review the exam subjects before it is conducted. See Figure 57.</p>	 <table border="1"><thead><tr><th>Response</th><th>Percentage</th></tr></thead><tbody><tr><td>نعم</td><td>63.6%</td></tr><tr><td>لا</td><td>27.3%</td></tr><tr><td>محيانا</td><td>9.1%</td></tr></tbody></table>	Response	Percentage	نعم	63.6%	لا	27.3%	محيانا	9.1%
Response	Percentage								
نعم	63.6%								
لا	27.3%								
محيانا	9.1%								

Q4: In students' questionnaire, do you use any software to help you in studying?

70.6% of students use various software to help in their study before exams, and 29.4% say no. See Figure 58.

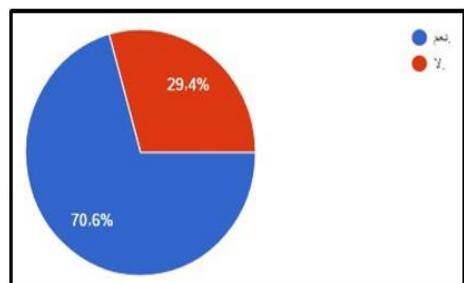


Figure 58: Question 4 Graph

Q5: In students' questionnaire, do you have difficulties in gathering information to understand your lecture?

63.6% of students have difficulties in gathering information in case they miss a lecture, and 30.3% sometimes suffer from this problem. See Figure 59.

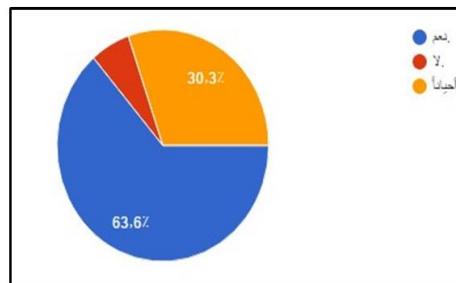


Figure 59: Question 5 Graph

Q6: In students' questionnaire, do you find that software helps them in studying? 41.9% of students find that a software helps them in studying, because of that the project team adopts the idea of allowing students of each course to generate MCQs, so that compels them to review the lectures before the next lecture. See Figure 60.

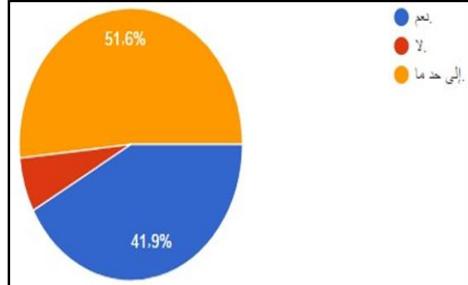


Figure 60: Question 6 Graph

Q7: In students' questionnaire, do you prefer the software to be as an application in smartphones and tablets? 97% of students prefer the software to be as an application in smartphones and tablets, and 3% are neutral. See Figure 61.

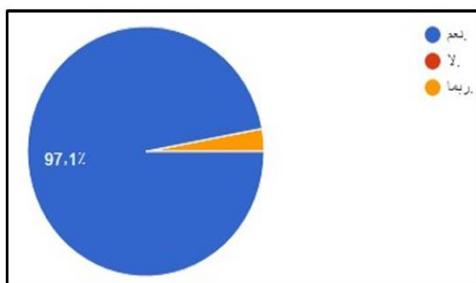


Figure 61: Question 7 Graph

Q8: In students' questionnaire, do you prefer the software send to them notifications when it is necessary?

91.2% of students prefer the software send to them notifications when it is necessary, and 8.8% are neutral. See Figure 63.

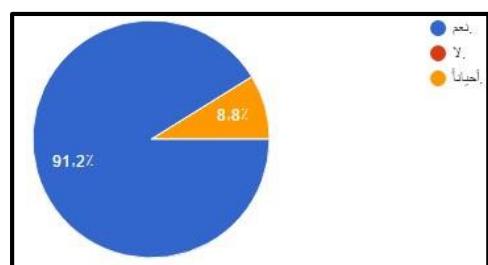


Figure 62: Question 8 Graph

Q9: In Faculty Member' questionnaire, you prefer the software to be as application in smartphones and tablets?

90.9% of Faculty Members prefer the software to be as application in smartphones and tablets, and 9.1% are neutral. See Figure 63.

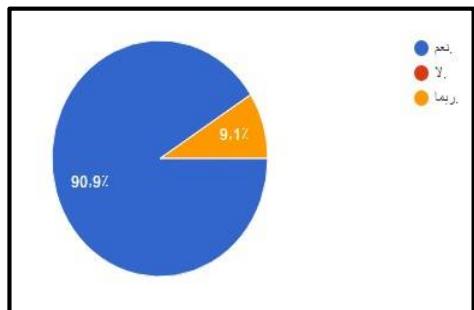


Figure 63: Question 9 Graph

Table 37: Questionnaire

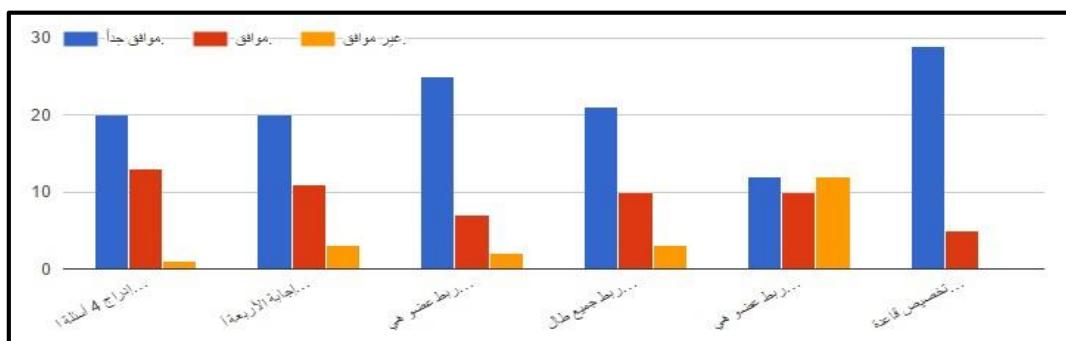


Figure 64: Students Suggestions

Appendix B: Faculty member questionnaire

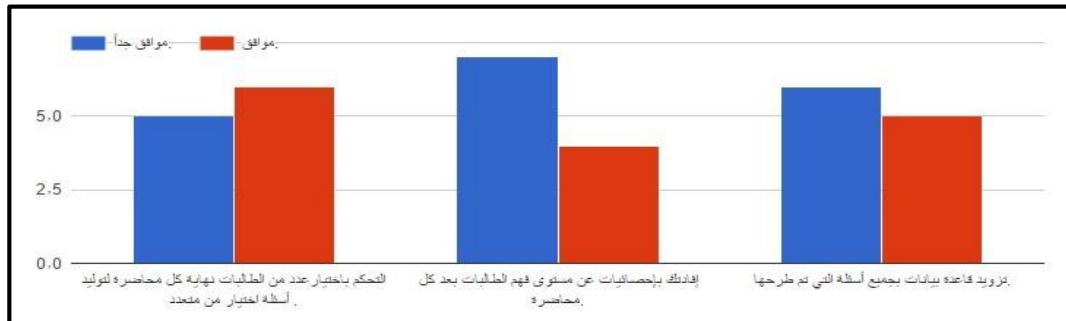


Figure 65: Faculty Members suggestions