STUDENT PROJECT TRACKING

By

Hamza Ahmad

Supervised By

Dr Khalid Saleem

Department of Computer Sciences

Quaid-i-Azam University

Islamabad

2019

Acknowledgement

In the name of ALLAH, Most Beneficent, Most Merciful.

First, I thank Allah at the completion of my project, as I completed this task only by His favor and grace. At this moment, this is due on me, to thank some personalities, because without their cooperation and supervision, I was unable to complete this work.

Secondly, my respected teacher and supervisor, Dr. Ghazanfar Farooq, whose door of kindness always remains open for me. His advices and instructions will guide me throughout my life. Completion of my task would not be possible without his help, encouragement, dynamic supervision and constructive criticism. I am highly indebted expressing my gratitude to him for his entire collaboration. Special thanks to Dr. M. Afzal Bhatti, Dr. Onaiza Maqbool, Dr. Rabeeh Ayaz Abbasi, Dr. Mubashir Mushtaq, Miss Ifrah Farrukh Khan, Madam Memoona Afsheen Malik, Dr. Mudassir Azam Sindhu, Dr. Shuaib Karim, Dr. Khalid Saleem, Dr. Muhammad Usman, Dr. Muhammad Naqi, Dr. Umer Rasheed and Dr. Muhammad Imran for their kind support and cooperation.

It is also necessary to me, to thank my parents whose prayers are treasure of my life. I have no words to pay gratitude to them whose affection, guidance and continuous encouragement did their best to shape my character.

Hamza Ahmad

2014-2018

Abstract

Student project tracking is done by a teacher. This is done by giving the students tasks and to review the outcome of tasks in meetings with the students. Teacher cannot handle student's record manually. When there are several students under his/her supervision. It is difficult to keep in touch with students and their projects manually. Sometimes teacher forgets which assignment he/she has given to student. What deadline is to be given for completing assignment? Manual tracking of projects is not easy. "Student Project Tracking" is an online web-based system. It will keep track of records related to project assignments/tasks of student. The system has three user roles - Student, Teacher and Project Coordinator (Project Coordinator is a teacher). Teacher can easily communicate with student. Teacher assigns students' assignments pertaining to project. Teacher can check the current assignment given to student and can keep the record of previous assignments. Student can view assignment given to him and can submit completed assignment. Project coordinator verifies tasks like student project allocation and add announcements to the system. This system is only for the students and teachers. It is assumed that Students and teachers will get sign in from their CMS (Campus Management System) ID's. We are developing this application in Notepad++ using HTML, JavaScript, PHP language and MySQL.

Table of Contents

List of Figures i

List of Table iii

CHAPTER 1 1

Introduction 1

1.1 Problem Definition 2

1.2 Proposed Solution 2

1.3 Scope 2

1.4 Objectives 3

1.5 Project Organization 3

1.6 Project Management Plan 4

1.7 Report Structure 7

CHAPTER 02 8

Requirement Gathering and Analysis 8

2.1. Product Overview 9

2.2. Major Inputs and outputs 9

2.3. Definitions, Acronyms and Abbreviations 10

2.4. User Characteristics 10

2.5. Constraints 10

2.6. Assumptions and Dependencies 10

2.7. Specific Requirements 10

2.8. Software System Attributes 12

2.9. Product Functions 13

2.10. Use Case Diagram 14

2.11. Use Case Description 15

2.12 Summary 26

CHAPTER 3 27

Software Design Description 27

3.1 Introduction 28

3.2 System Architecture Design 30

3.3 Low-fidelity Interfaces 31
3.4 System Sequence Diagrams (SSD) 37
3.5 Entity Relationship Diagram 42
3.6 Data flow diagrams (DFD) 42
3.7 Activity diagrams 48
3.9 **Database Design** 50
3.9 **Summary** 50
CHAPTER 4 51
Software Implementation Plan 51
4.1. Language Selection 52
4.2. Tools Selection 52
4.3 Summary 53
CHAPTER 5 54
Software Test Document 54
5.1 Introduction 55
5.2 Test Plan 55
5.3 Test Cases 56
CHAPTER 6 61
Conclusions and Future Enhancements 61
6.1. Conclusions 62
6.2. Future enhancement 62
Appendix 63

List of Figures

Figure 1. 1 Time Schedule [2/2] 5
Figure 1. 2 Gantt chart [1/2] 6
Figure 1. 3 Gantt chart [2/2] 7
Figure 2. 1 Use case diagram 14
Figure 3. 1 System Architecture Design 31
Figure 3. 2 Login Teacher (project coordinator)/Student 32
Figure 3. 3 Teacher Panel 32
Figure 3. 4 Teacher (Project Coordinator) 33
Figure 3. 5 Add Assignment 33
Figure 3. 6 View Announcement 34
Figure 3. 7 View Assignment 34
Figure 3. 8 Add Announcement 35
Figure 3. 9 Add Project Coordinator 35
Figure 3. 10 View Announcement (Teacher) 36
Figure 3. 11 View Assignment (Student) 36
Figure 3. 12View Announcement (Student) 37
Figure 3. 13 Login SSD 38
Figure 3. 14 Assigning new project coordinator SSD 38
Figure 3. 15 Make Announcement SSD 39
Figure 3. 16 Add new project SSD 39
Figure 3. 17 View Student List SSD 40
Figure 3. 18 Update deadline SSD 40
Figure 3. 19 Response to announcement SSD 41
Figure 3. 20 ERD 42
Figure 3. 21 Zero level DFD 43
Figure 3. 22 First level DFD (Project Coordinator) 44
Figure 3. 23 First level DFD (Teacher) [1/2] 45
Figure 3. 24 First level DFD (Teacher) [2/2] 46
Figure 3. 25 First level DFD (Student) 47
Figure 3. 26 Activity Diagram (Teacher) 48
Figure 3. 27 Activity Diagram (Student) 49
Figure 3. 28 database design 50

Figure A. 1 Project Coordinator Dashboard 63
Figure A. 2 Teacher Dashboard 64
Figure A. 3 Student Dashboard 64
Figure A. 4 Verify Student Project Allocation 64
Figure A. 5 Assign New Project Coordinator 65

Figure A. 6 Project Coordinator Log 66
Figure A. 7 Add Announcements 66
Figure A. 8 Add Projects 67
Figure A. 9 Allocate Project to Student 67
Figure A. 10 View Student List 68
Figure A. 11 Add Assignment 68
Figure A. 12 View Assignment results 69
Figure A. 13 View Announcements 69
Figure A. 14 View List of not Submitted Assignment 70
Figure A. 15 View Assignment (student) 70
Figure A.16 Extension Request 71
Figure A. 17 Reply Assignment 71
Figure A. 18 View Announcement (student) 72
Figure A. 19 Response to Announcement 72
Figure A. 20 Assignment Deadline Alert 73
Figure A. 21 Assignment Email Alert 73
Figure A. 22 Evaluation Interface 73

List of Table

Table 2. 1 Definitions, Acronyms and Abbreviations 10
Table 2. 2 Login description 15
Table 2. 3 Add Project description 16
Table 2. 4 Verifying Student project allocation description 17
Table 2. 5 View Student list description 17
Table 2. 6 Add assignment description 18
Table 2. 7 View Assignment result description 19
Table 2. 8 Update deadline description 20
Table 2. 9 Assigning New Project Coordinator description 21
Table 2. 10 Add Announcement description 21
Table 2. 11 View Announcement description 22
Table 2. 12 View Assignment deadline request description 22
Table 2. 13 View Assignment description 23
Table 2. 14 Reply Assignment description 24
Table 2. 15 View Announcement description 24
Table 2. 16 Response announcement description 25
Table 2. 17 View list of not submitted assignment description 26
Table 3. 1 Requirement Traceability Matrix 29
Table 5. 1 Login 57
Table 5. 2 Add Assignment 57
Table 5. 3 View Announcement 58
Table 5. 4Add project 59
Table 5. 5 Add Announcement 59
Table 5. 6 list of not Submitted Assignment 60

CHAPTER 1

Introduction

CHAPTER 1 Introduction

This report first introduces student project tracking. It then highlights the problem that has been addressed in this work along with the designed and developed solution. Chapter 1 elaborates project organization and project planning. Finally, the chapter elucidates the scope and objectives of this project.

1.1 Problem Definition

Student project tracking is done by a teacher. This is done by giving the students tasks and to review the outcome of tasks in meetings with the students. Teacher cannot handle student's record manually. When there are several students under his/her supervision. It is difficult to keep in touch with students and their projects manually. Sometimes Teacher forgets which assignment he/she has given to student. What deadline is to be given for completing assignment? Manual tracking of projects is not easy.

1.2 Proposed Solution

"Student Project Tracking" is an online web-based system. It will keep track of records related to project assignments/tasks of student. Teacher can easily communicate with student. Teacher assigns students' assignments preparing to project. Teacher can check the current assignment given to student and can keep the record of previous assignments. Student can view assignment given to him and can submit completed assignment.

1.3 Scope

Scope of the system defines the boundaries of the system that what will the software deliver to users.

1. Project Allocation.
2. Scheduling of Assignment related to project.
3. Evaluation of the Assignments.
4. Adding events like announcements.

1.4 Objectives

The primary objective of the system is to facilitate both teachers and students. This system overcomes the burden of manual works. Student has to come to department for meeting. Sometimes teacher may not be available in department. Keeping records manually is not easy. Records may be misplaced by teacher, so web system is manageable. It would be time-saving and easily manageable.

1.5 Project Organization

Project organization is a structure that facilitates the coordination and implementation of project activities. Its main reason is to create an environment that fosters interactions with a minimum amount of disruptions, overlaps, and conflict. Project organization is basically how the project is organized according to software engineering. Which process model is going to be followed and what are major roles and responsibilities. Finally, which tools and techniques will be used in order to develop this system?

Software Process Model

For this project water fall process model will be followed. The reason for using this process model is that, requirements are well-defined and they are well-understood.

Roles and Responsibilities

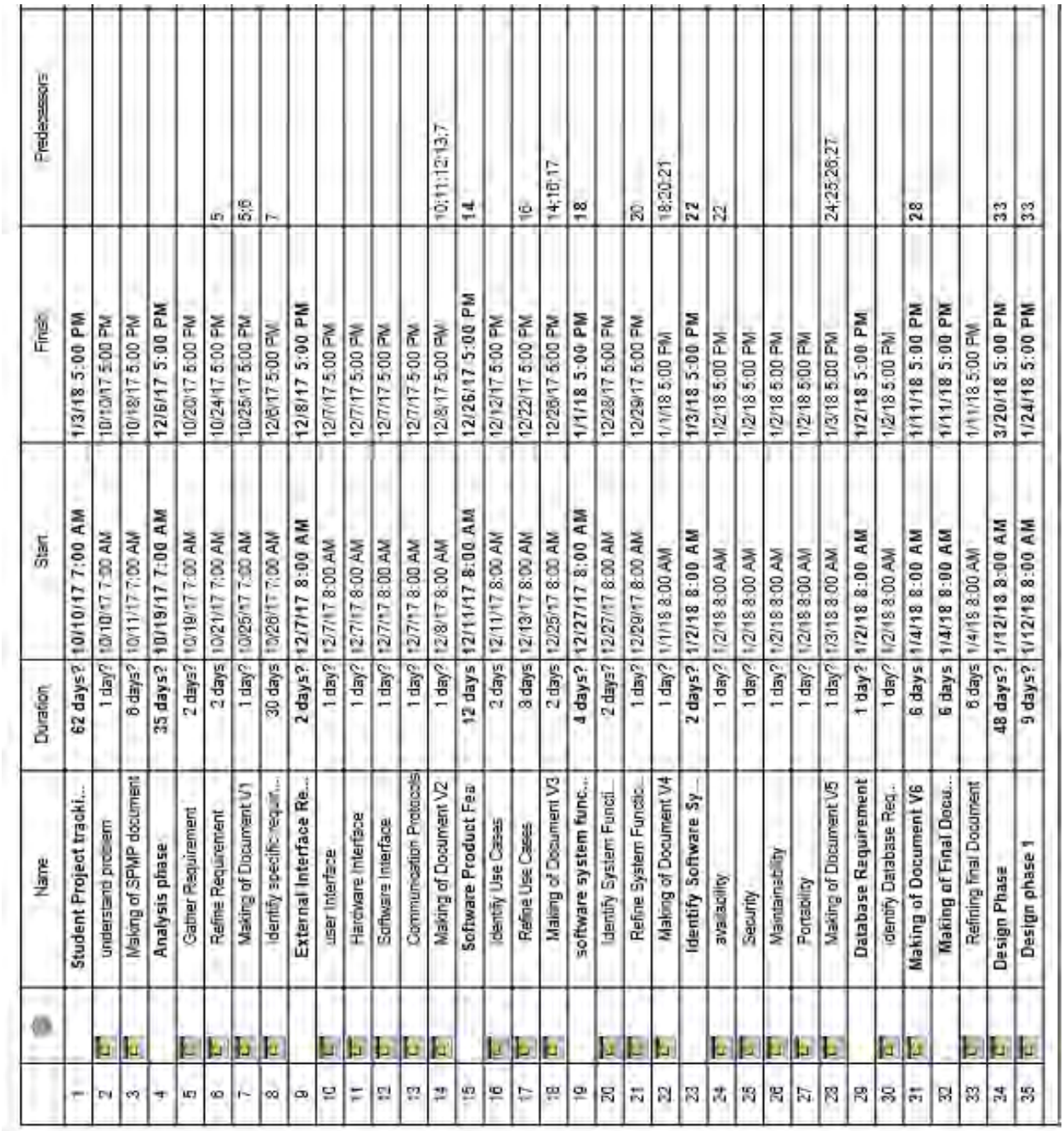
The "student's project tracking" is a individual project, therefore, roles and responsibilities are covered by a individual student.

Tools and Techniques

Tools that are being supported for Analysis, Design and Documentation of this system are notepad++, Microsoft Visio for diagrams such as use case diagram, class diagram, activity diagram, domain model and Entity relationship diagram and for writing documentation Microsoft word is used. For designing project management plan of the system, Project Liber is used.
Student Project Tracking will be developed using PHP, JavaScript, HTML and CSS. HTML will be used to create front end of application. CSS will be used as cascading style sheets to style the contents. PHP will be used for server-side programming. For database MySQL will be used.

1.6 Project Management Plan

Software project planning is assignment which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitate software production.



	Name	Duration	Start	Finish	Predecessors
1	**Student Project tracki...**	**62 days?**	**10/10/17 7:00 AM**	**1/3/18 5:00 PM**	
2	understand problem	1 day?	10/10/17 7:00 AM	10/10/17 5:00 PM	
3	Making of SPMP document	8 days?	10/11/17 7:00 AM	10/18/17 5:00 PM	
4	**Analysis phase**	**35 days?**	**10/19/17 7:00 AM**	**12/6/17 5:00 PM**	
5	Gather Requirement	2 days?	10/19/17 7:00 AM	10/20/17 5:00 PM	
6	Refine Requirement	2 days	10/21/17 7:00 AM	10/24/17 5:00 PM	5
7	Making of Document V1	1 day?	10/25/17 7:00 AM	10/25/17 5:00 PM	5;6
8	Identify specific requir...	30 days	10/26/17 7:00 AM	12/6/17 5:00 PM	7
9	**External Interface Re...**	**2 days?**	**12/7/17 8:00 AM**	**12/8/17 5:00 PM**	
10	user Interface	1 day?	12/7/17 8:00 AM	12/7/17 5:00 PM	
11	Hardware Interface	1 day?	12/7/17 8:00 AM	12/7/17 5:00 PM	
12	Software Interface	1 day?	12/7/17 8:00 AM	12/7/17 5:00 PM	
13	Communication Protocols	1 day?	12/7/17 8:00 AM	12/7/17 5:00 PM	
14	Making of Document V2	1 day?	12/8/17 8:00 AM	12/8/17 5:00 PM	10;11;12;13;7
15	**Software Product Fea**	**12 days**	**12/11/17 8:00 AM**	**12/26/17 5:00 PM**	**14**
16	Identify Use Cases	2 days	12/11/17 8:00 AM	12/12/17 5:00 PM	
17	Refine Use Cases	8 days	12/13/17 8:00 AM	12/22/17 5:00 PM	16
18	Making of Document V3	2 days	12/25/17 8:00 AM	12/26/17 5:00 PM	14;16;17
19	**software system func...**	**4 days?**	**12/27/17 8:00 AM**	**1/1/18 5:00 PM**	**18**
20	Identify System Functi...	2 days?	12/27/17 8:00 AM	12/28/17 5:00 PM	
21	Refine System Functio...	1 day?	12/29/17 8:00 AM	12/29/17 5:00 PM	20
22	Making of Document V4	1 day?	1/1/18 8:00 AM	1/1/18 5:00 PM	18;20;21
23	**Identify Software Sy...**	**2 days?**	**1/2/18 8:00 AM**	**1/3/18 5:00 PM**	**22**
24	availability	1 day?	1/2/18 8:00 AM	1/2/18 5:00 PM	22
25	Security	1 day?	1/2/18 8:00 AM	1/2/18 5:00 PM	
26	Maintainability	1 day?	1/2/18 8:00 AM	1/2/18 5:00 PM	
27	Portability	1 day?	1/2/18 8:00 AM	1/2/18 5:00 PM	
28	Making of Document V5	1 day?	1/3/18 8:00 AM	1/3/18 5:00 PM	24;25;26;27
29	**Database Requirement**	**1 day?**	**1/2/18 8:00 AM**	**1/2/18 5:00 PM**	
30	identify Database Req...	1 day?	1/2/18 8:00 AM	1/2/18 5:00 PM	
31	**Making of Document V6**	**6 days**	**1/4/18 8:00 AM**	**1/11/18 5:00 PM**	**28**
32	**Making of Final Docu...**	**6 days**	**1/4/18 8:00 AM**	**1/11/18 5:00 PM**	
33	Refining final Document	6 days	1/4/18 8:00 AM	1/11/18 5:00 PM	
34	**Design Phase**	**48 days?**	**1/12/18 8:00 AM**	**3/20/18 5:00 PM**	**33**
35	**Design phase 1**	**9 days?**	**1/12/18 8:00 AM**	**1/24/18 5:00 PM**	**33**

Figure 1. 1 Time Schedule [1/2]

		Name	Duration	Start	Finish	Predecessors
36		Devlop Architectural D...	2 days?	1/12/18 8:00 AM	1/15/18 5:00 PM	
37		Refine Architectural de...	2 days	1/16/18 8:00 AM	1/17/18 5:00 PM	36
38		Develop Interface Design	2 days	1/12/18 8:00 AM	1/15/18 5:00 PM	
39		Refine Interface Design	1 day?	1/16/18 8:00 AM	1/16/18 5:00 PM	38
40		Develop Sequence Dia...	2 days	1/12/18 8:00 AM	1/15/18 5:00 PM	
41		Refine Sequence Diagr...	3 days	1/16/18 8:00 AM	1/18/18 5:00 PM	40
42		Develop Domain Model	2 days	1/19/18 8:00 AM	1/22/18 5:00 PM	41
43		Refine Domain Model	2 days	1/23/18 8:00 AM	1/24/18 5:00 PM	42
44		**Design Phase II**	**7 days**	**1/23/18 8:00 AM**	**1/31/18 5:00 PM**	**42**
45		Develop Class diagram	5 days	1/23/18 8:00 AM	1/29/18 5:00 PM	
46		Refine Class diagram	2 days	1/30/18 8:00 AM	1/31/18 5:00 PM	43;45
47		**Develop Algorithm**	**22 days**	**2/1/18 8:00 AM**	**3/2/18 5:00 PM**	**46**
48		Develop Pseudo Code	10 days	2/1/18 8:00 AM	2/14/18 5:00 PM	
49		Review Pseudo code	2 days	2/15/18 8:00 AM	2/16/18 5:00 PM	48
50		Draw Decision Table	7 days	2/19/18 8:00 AM	2/27/18 5:00 PM	49
51		Review Decision	3 days	2/28/18 8:00 AM	3/2/18 5:00 PM	50
52		**Evaluate Design**	**12 days?**	**3/5/18 8:00 AM**	**3/20/18 5:00 PM**	**50**
53		validate design	3 days	3/5/18 8:00 AM	3/7/18 5:00 PM	51
54		verify design	3 days	3/8/18 8:00 AM	3/12/18 5:00 PM	53
55		Review and refine design	5 days	3/13/18 8:00 AM	3/19/18 5:00 PM	54
56		Finalize document	1 day?	3/20/18 8:00 AM	3/20/18 5:00 PM	55

Figure 1. 2 Time Schedule [2/2]

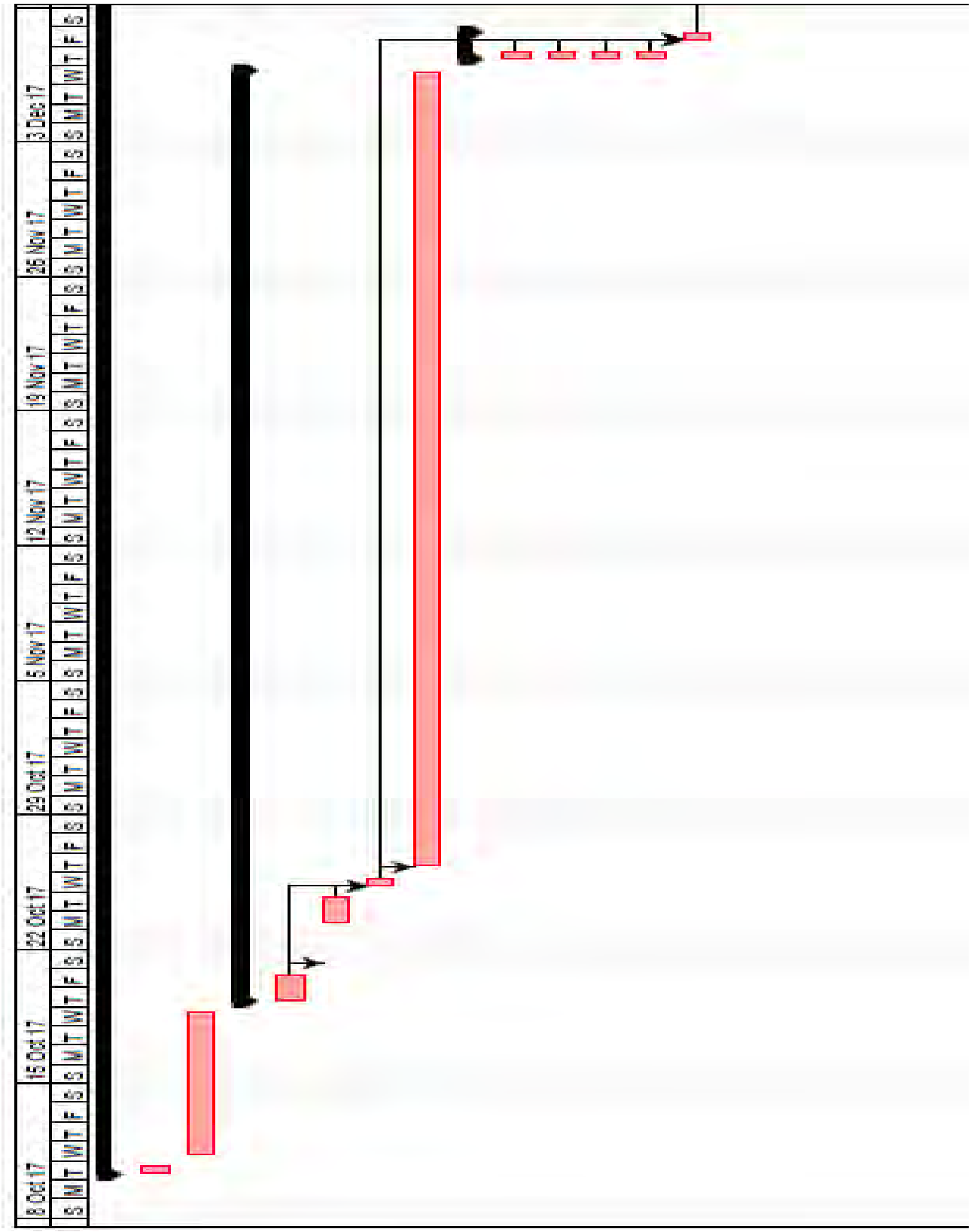


Figure 1. 3 Gantt chart [1/2]

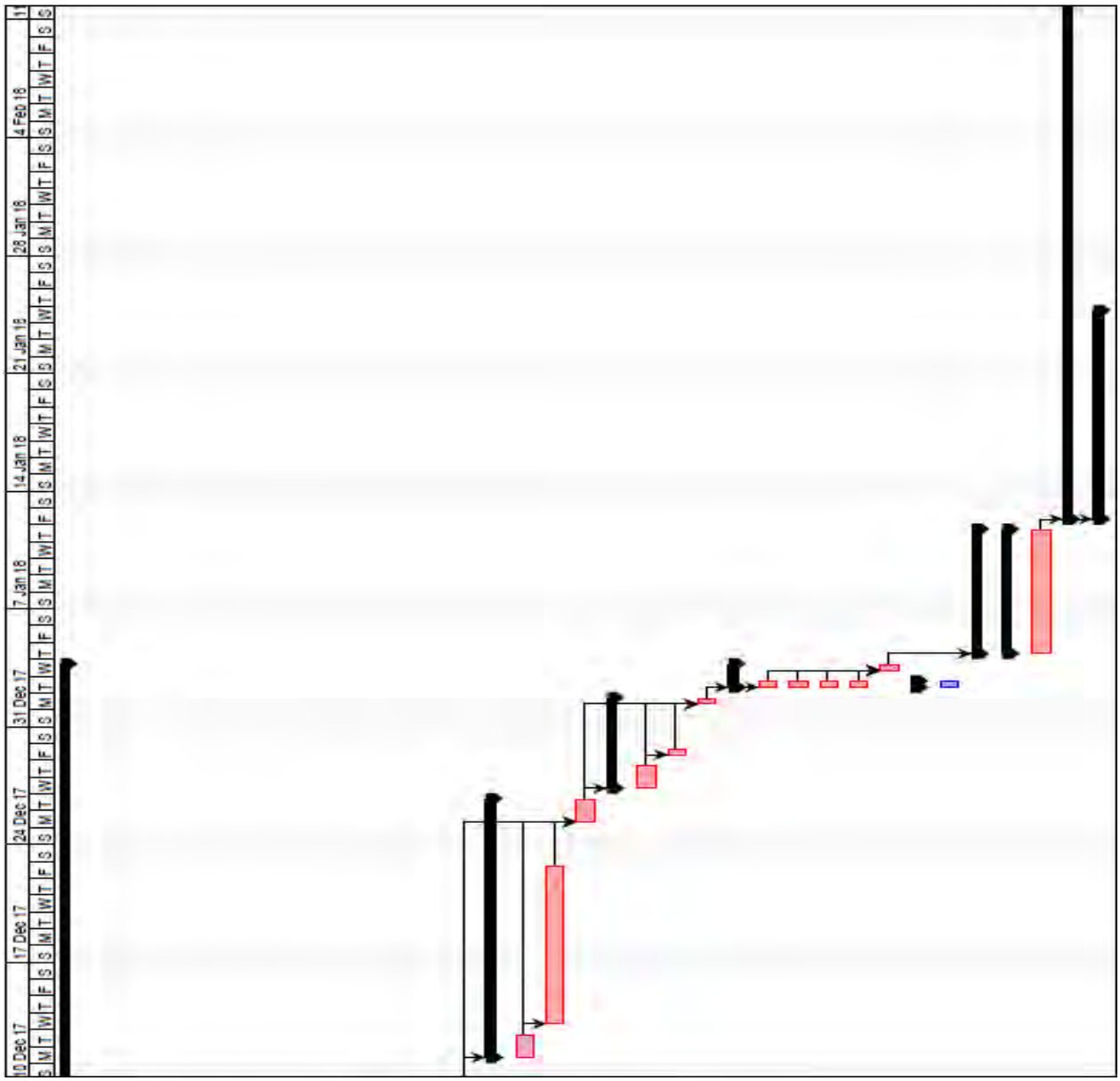


Figure 1. 4 Gantt chart [2/2]

1.7 Report Structure

Chapter 1 briefly describes the introduction of the system, what are actual problem and proposed solution, its scope, objective, describe the organization of project and project management plan. Chapter 2 describes the requirements of the system and to decide what the system should do and what the system should not do. Briefly describes complete details of functional and non-functional requirements specifications for the system.

CHAPTER 2

Requirement Gathering and Analysis

CHAPTER 2 Requirement Gathering and Analysis

The purpose of this requirement gathering and analysis is to clear the requirements of the system and to decide what the system should do and what the system should not do. To clear the requirements like function and non-functional requirements for the system and understand the major inputs for and outputs from the system.

2.1. Product Overview

"Student Project Tracking" is a system to keep track of project assignments. This system is basically web-based. The teacher can use this system for tracking student project. The teacher will be able to check current assignment and can as well assign new assignments. When teacher assigns new assignment, a notification will be sent to student.

Major Functions

Teacher will sign in from the mobile or web. System will keep teachers user name as an id and keep password for signing. Teacher has authority to update deadline i.e. if student has requested for extension in assignment deadline. If student has submitted assignment after deadline then late submission status will be shown to teacher. Teacher can assign new assignments to his/her student with the start of date and end date of assignment.

2.2. Major Inputs and outputs

Major inputs are given in following:

- Teacher enters password.
- Teacher enters assignment description and gives deadline to it.
- Teacher adds documents links related to assignments completed.

Major outputs are given in following:

- Assignments are assigned successfully to student and notification is sent to student.

2.3. Definitions, Acronyms and Abbreviations

Table 2. 1 Definitions, Acronyms and Abbreviations

Teacher	Supervisor
SPT	Student project tracking

2.4. User Characteristics

Users of this system are Project coordinator, project teacher(supervisor) and students. It is assumed that the user has basic knowledge of computer or laptop and knowledge of web application and mobile. Users must have knowledge of how to use web-based applications and is able to perform certain tasks.

2.5. Constraints

User should have computer or laptop and internet connection to access the system.

2.6. Assumptions and Dependencies

This web-based application depends upon the availability of internet. It is assumed that the users have a computer or laptop to access the system.

2.7. Specific Requirements

Functional Requirements

Functional requirements are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. The system is for Department of Computer Science. The system allows teacher to access the system through entering his name and password. The system gives the proper message of any invalid entry. The system allows teacher to manage assignments with there projects. The system enables teacher to make new assignment and system will send email to student of project his new assignment. Main functional requirements are:

- Teacher will add projects with title and description.
- Teacher will assign project to student.
- Project coordinator will approve student with his/her project.
- Project coordinator will add announcements.

- Project coordinator will assign new project coordinator.
- Project coordinator can check previous project coordinators with start and end dates.
- Teacher will assign project assignments to student.
- Teacher will evaluate assignment.
- Students will response against the announcements.
- Student can reply against assignment.
- Student request for deadline.
- Teacher can update/edit extension deadline.
- Email will be sent to student.
- Assignment Deadline Alert will be send to student 1 day before last date.

External Interface Requirement

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

User Interface

Through user interface user will be able to interact with the system. This will be Web-based application for users. User can use this application through internet. User would have unique username and password to access this application.

Software Interfaces

Student project tracking system is web-based system and it will be implemented through PHP; therefore, this system will be run on any operating system. The internet is required to access the system. The system can be accessed through any internet browser that supports html, css, JavaScript or from any mobile.

Communication Protocol

Communication protocols required for this system are; Hypertext transfer protocol [http] for communication over the internet.

2.8. Software System Attributes

Reliability

System should be reliable. There should be no occurrence of the failure. The system should be able to work properly all-time means the extent to which it works as and when needed.

Availability

System should be available to every user at any time. All the teachers and students are able to access the system at any time.

Security

Since this system will be hosted on server, all the user data will be kept on that server. Product should be able to protect privacy of user data. User can only be accessed the system through user own credentials and any other user should not be able to access to the user private data. User can only perform operation under the permission. Some user groups can be configured that they can never have certain permission.

Maintainability

There should be aspect of maintainability for the system. In some cases, maintainability involves a system of continuous improvement learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience. The application should be easy to extend. The code should be written in a way that it favors implementation of new functions. In order for future functions to be implemented easily to the application.

Portability

This is web based system so main purpose of developing web-based system is to improve the portability of system. To improve portability. System should be lightweight therefore that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase.

Performance

Since this system is going to web based and mobile based responsive application, it does require a powerful server machine with high band internet access. Server machine should have a powerful CPU and high-speed internet access therefore that it can handle multiple users at the same time. As the system just deployed in computer science department of Quaid I Azam university. Not much burden will be on system because system is bounded for computer science department.

2.9. Product Functions

Teacher Log in

Teacher will add his/her user name and password to login to the system.

Project coordinator

Project coordinator will add his/her user name and password to login to the system.

Student login

Student will add his/her user name and password to login to the system.

Add Project

Teacher will add projects.

Assign Project

Teacher will assign project to student.

Add Assignment

Teacher can add assignments.

View Assignment Deadline request

Teacher will view assignment deadline requests.

Add Announcement

Project coordinator will add announcement.

Assign new Project coordinator

Project coordinator will assign new project coordinator.

Approve Projects

Project coordinator will approve projects with student.

Reply Assignment

Student will add completed assignment.

View Assignment

Student can view assignment.

2.10. Use Case Diagram

List of functions mentioned in previous diagram are described in detail as use cases, therefore we can look more precisely that how user can interact with system to perform different functions.

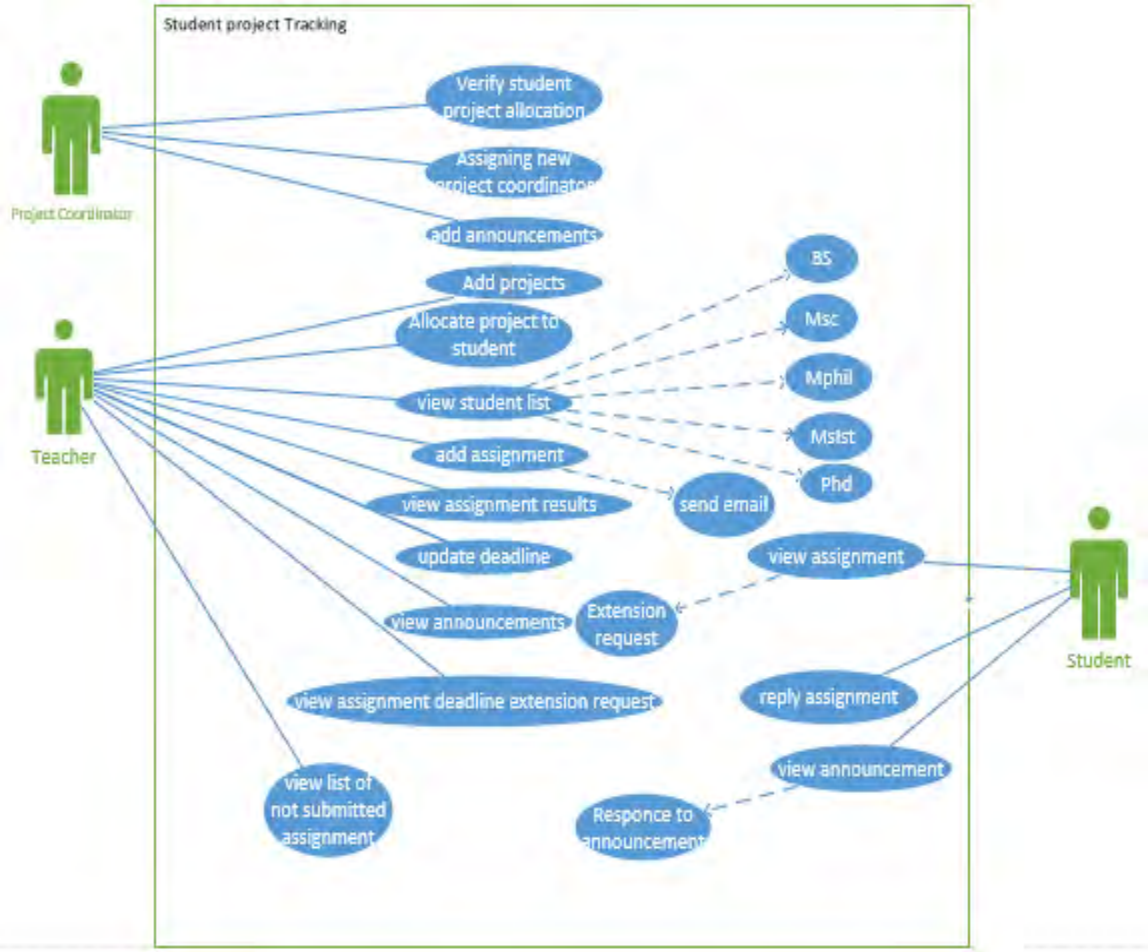


Figure 2. 1 Use case diagram

2.11. Use Case Description

Writing use cases description is an excellent technique to understand and describe requirements. Use cases are a mechanism to help to understand which the system is going to be developed. Informally, they are stories of using a system to meet goals.

Use case 1: login

ID	**UC1**
Name	Login
Primary Actor	Teacher
Pre-Conditions	User must be from faculty of department
Post-Conditions	User is logged in to the system
Main Success Scenario	1. User enter his/her user name. 2. User enters his/her password 3. User selects login.
Alternative flows or Extensions	1. If User selects login without inserting user name or password, a message will be displayed that, this field is required. 2. If User enters wrong user name or password, a message will be displayed that, please check your username or password.
Frequency	Could be nearly continuous

Table 2. 2 Login description

Use case 2: Add Project

ID	**UC2**
Name	Add Project
Primary Actor	Teacher
Pre-Conditions	Teacher is logged in
Post-Conditions	Project is added by the teacher
Main Success Scenario	1. Teacher is logged in. 2. Teacher has added Project.
Alternative flows or Extensions	None.
Frequency	Could be one time.

Table 2. 3 Add Project description

Use case 3: Verifying student project allocation

ID	**UC3**
Name	Verifying student project allocation
Primary Actor	**Project coordinator**
Pre-Conditions	Project coordinator is logged in
Post-Conditions	Project with student is approved.

Main Success Scenario	1. Project coordinator is logged in. 2. Project coordinator has approved project with student.
Alternative flows or Extensions	None
Frequency	Could be more than one time.

Table 2. 4 Verifying Student project allocation description

Use case 4: View Student list

ID	**UC4**
Name	View student list
Primary Actor	**Teacher**
Pre-Conditions	Teacher is logged in
Post-Conditions	Teacher has viewed student list.
Main Success Scenario	1. Teacher is logged in. 2. Teacher has viewed student list.
Alternative flows or Extensions	1. There is no student in a list.
Frequency	Could be more than one time

Table 2. 5 View Student list description

Use case 5: Add Assignment

ID	**UC5**
Name	Add Assignment
Primary Actor	**Teacher**
Pre-Conditions	Teacher has student with project
Post-Conditions	Assignment is added successfully by teacher.
Main Success Scenario	1. Teacher is logged in. 2. Teacher will select student. 3. Teacher will add title. 4. Teacher will add description. 5. Teacher will upload file. 6. Teacher will add assignment successfully.
Alternative flows or Extensions	1. Teacher has no student with project.
Frequency	Occurs only one time.

Table 2. 6 Add assignment description

Use case 6: View Assignment results

ID	**UC6**
Name	View Assignment results
Primary Actor	**Teacher**
Pre-Conditions	Assignment is sent by the student
Post-Conditions	Teacher has view assignment reply.
Main Success Scenario	1. Teacher is logged in. 2. Teacher has viewed assignment reply.
Alternative flows or Extensions	1. There is no reply from student.
Frequency	Could be more than one time.

Table 2. 7 View Assignment result description

Use case 7: Update Deadline

ID	**UC7**
Name	Update deadline
Primary Actor	Teacher
Pre-Conditions	Request of extension of deadline
Post-Conditions	Deadline is extended
Main Success Scenario	1. Teacher will view the extension request. 2. Teacher will extend deadline date.
Alternative flows or Extensions	None.
Frequency	Could be nearly continuous.

Table 2. 8 Update deadline description

Use case 8: Assigning New project coordinator

ID	**UC8**
Name	Assigning new project coordinator
Primary Actor	Project coordinator
Pre-Conditions	Project coordinator is logged in
Post-Conditions	Project coordinator has assigned new project coordinator

Main Success Scenario	1. Project coordinator is logged in. 2. Project coordinator will assign new project coordinator.
Alternative flows or Extensions	None.
Frequency	Could be nearly continuous.

Table 2. 9 Assigning New Project Coordinator description

Use case 9: Add Announcements

ID	**UC9**
Name	Add announcements
Primary Actor	Project coordinator
Pre-Conditions	Project coordinator is logged in
Post-Conditions	Project coordinator has add announcement
Main Success Scenario	1. Project will log in. 2. Project coordinator will add announcement.
Alternative flows or Extensions	None.
Frequency	Could be nearly continuous.

Table 2. 10 Add Announcement description

Use case 10: View Announcement

ID	UC10
Name	View announcement
Primary Actor	Teacher
Pre-Conditions	There is an announcement from project coordinator.
Post-Conditions	Teacher has viewed announcement
Main Success Scenario	1. Teacher is logged in. 2. Announcements has been viewed by teacher.
Alternative flows or Extensions	1. There is no announcement.

Table 2. 11 View Announcement description

Use case 11: View Assignment deadline extension request

ID	UC11
Name	View assignment deadline extension request
Primary Actor	Teacher
Pre-Conditions	Request is done by student against deadline extension.
Post-Conditions	Teacher has viewed assignment deadline extension request
Main Success Scenario	1. Teacher is logged in. 2. Teacher has viewed assignment deadline extension request
Alternative flows or Extensions	None
Frequency	Could be nearly continuous

Table 2. 12 View Assignment deadline request description

Use case 12: View Assignment

ID	**UC12**
Name	view assignment
Primary Actor	**Student**
Pre-Conditions	Student is logged in
Post-Conditions	Student has viewed assignment.
Main Success Scenario	1. Student is logged in. 2. Student will view assignment.
Alternative flows or Extensions	None
Frequency	Could be nearly continuous.

Table 2. 13 View Assignment description

Use case 13: Reply assignment

ID	**UC13**
Name	Reply assignment
Primary Actor	**Student**
Pre-Conditions	Assignment is send by the teacher
Post-Conditions	Student has replied against assignment.

Main Success Scenario	1. Student is logged in. 2. Student will upload the completed assignment. 3. Student will reply the uploaded file against assignment.
Alternative flows or Extensions	1. Student has not completed his assignment.
Frequency	Could be nearly continuous

Table 2. 14 Reply Assignment description

Use case 14: View Announcement

ID	**UC14**
Name	View announcement
Primary Actor	**Student**
Pre-Conditions	There is an announcement from project coordinator.
Post-Conditions	Student has viewed announcement.
Main Success Scenario	1. Student is logged in. 2. Student has viewed announcements.
Alternative flows or Extensions	There is no announcement
Frequency	Could be nearly continuous.

Table 2. 15 View Announcement description

Use case 15: Response to Announcement

ID	**UC15**
Name	Response to announcement
Primary Actor	**Student**
Pre-Conditions	There is an announcement from project coordinator
Post-Conditions	Student has respond to announcement.
Main Success Scenario	1. Student is logged in. 2. Student has viewed announcement. 3. Student has respond against announcement.
Alternative flows or Extensions	None
Frequency	Could be nearly continuous

Table 2. 16 Response announcement description

Use case 16: View list of not submitted assignment

ID	**UC 16**
Name	View list of not submitted assignment
Primary Actor	**Teacher**
Pre-Conditions	Assignments is been sent by teacher

Post-Conditions	Teacher has viewed not submitted list.
Main Success Scenario	1. Teacher is logged in. 2. Teacher has viewed list.
Alternative flows or Extensions	1. Teacher has not assigned any assignment to student. 2. Teacher has no student with project. 3. Students have submitted all assignments.
Frequency	Could be nearly continuous

Table 2. 17 View list of not submitted assignment description

2.12 Summary

Chapter 2 briefly describes the requirements of the system and to decide what the system should do and what the system should not do. Briefly described complete details of functional and non-functional requirements specifications for the system.

Chapter 3 gives the complete description of software design. It then elaborates the architectural design and detailed description of components of system.

CHAPTER 3

Software Design Description

CHAPTER 03 Software Design Description

This chapter first gives the complete description of software design. It then elaborates the architectural design and detailed description of components of system. Finally, this chapter elucidates the user interface design and interaction diagrams mainly system sequence diagram and class diagram.

3.1 Introduction

Software Design Description (SDD) is the representation of a software design to be used for communication design information to its stakeholders. It shows how the software system will be structured to satisfy the requirements. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, that is the detailed design stage, more detailed data structures are defined and algorithms and codes are developed for the defined architecture.

Design Overview

Design is meaningful engineering representation of something that is to be built. The design mainly focus on four major areas of concern mainly data, architecture, interfaces and components [3]. Software design is an iterative process through which requirements are translated into a blueprint for constructing the software. It shows how end user can interact with the system therefore mainly focus on user interface design. Design begins with requirement model and at each stage, software design work product are reviewed for clarity, correctness, completeness and consistency with the requirements and with one another. Software design sits at the technical kernel of software engineering and is applied regardless of the software process model that is used. The requirements translated clearly through designing class diagram, sequence diagram, system sequence diagram and user interface interactions.

Requirement Traceability Matrix

Requirements traceability matrix is a matrix in which we describe that which requirement is mapping with which sequence diagram, test case, and method of class diagram. The purpose of traceability matrix is that when requirements have to be updated then one can update that requirement using traceability matrix instead of going through the whole document. It is often used with high-level requirements and detailed requirements of the product to the matching parts of

high-level design, detailed design, test plan, and test cases. Traceability matrix of this system is shown below.

Requirement Id	Requirement Name	Test Case	SSD	Interface
UC:1	Login	Yes	Yes	Yes
UC:2	Approve project	No		Yes
UC:3	Assign new project coordinator	No	Yes	Yes
UC:4	Add announcement	Yes	Yes	Yes
UC:5	Add project	Yes	Yes	Yes
UC:6	Assign project to student	No	No	Yes
UC:7	View student list	No	Yes	Yes
UC:8	Add assignment	Yes	No	Yes
UC:9	View assignment results	No	No	Yes
UC:10	Update deadline	No	Yes	No
UC:11	View Announcements	Yes	No	Yes
UC:12	View assignment deadline request	No	No	Yes
UC:13	View assignment	No	No	Yes
UC:13.1	Extension request	No	No	Yes
UC:14	Reply assignment	No	No	Yes
UC:15	View announcement	Yes	No	Yes
UC:15.1	Response to announcement	No	Yes	Yes
UC:16	View list of not submitted assignment	Yes	Yes	Yes

Table 3. 1 Requirement Traceability Matrix

3.2 System Architecture Design

Architectural design is defines the relationship between major structural elements of the software. It defines the design patterns that can be used to achieve the requirements that have been defined for the system. Architecture design entails the manner in which these components interact and the structure of data that are used by the components. Components or modules are generalized to represent major system elements and their interactions.

Chosen system architecture

The chosen architecture for this system is Three Tier Architectural pattern Three-tier architecture is a software architecture pattern in which the user interface (presentation),functional process logic (business rules), computer data storage and data access are developed and maintained as independent modules Interacting between components of system is shown in diagram. The singular quality of a three-tier architecture is the separation of the application logic into a distinct logical middle tier of software. The interface tier is relatively free of application processing; windows or web pages forward assignment requests to the middle tier. The middle tier communicates with the back-end storage layer.[5]

A three-tier architecture separates its tiers from each other based on the complexity of the users and how they use the data present in the database. In database tier, the database resides along with its query processing languages. This tier also have the relations that define the data and their constraints at this level. The middle tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database. In user interface tier end-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

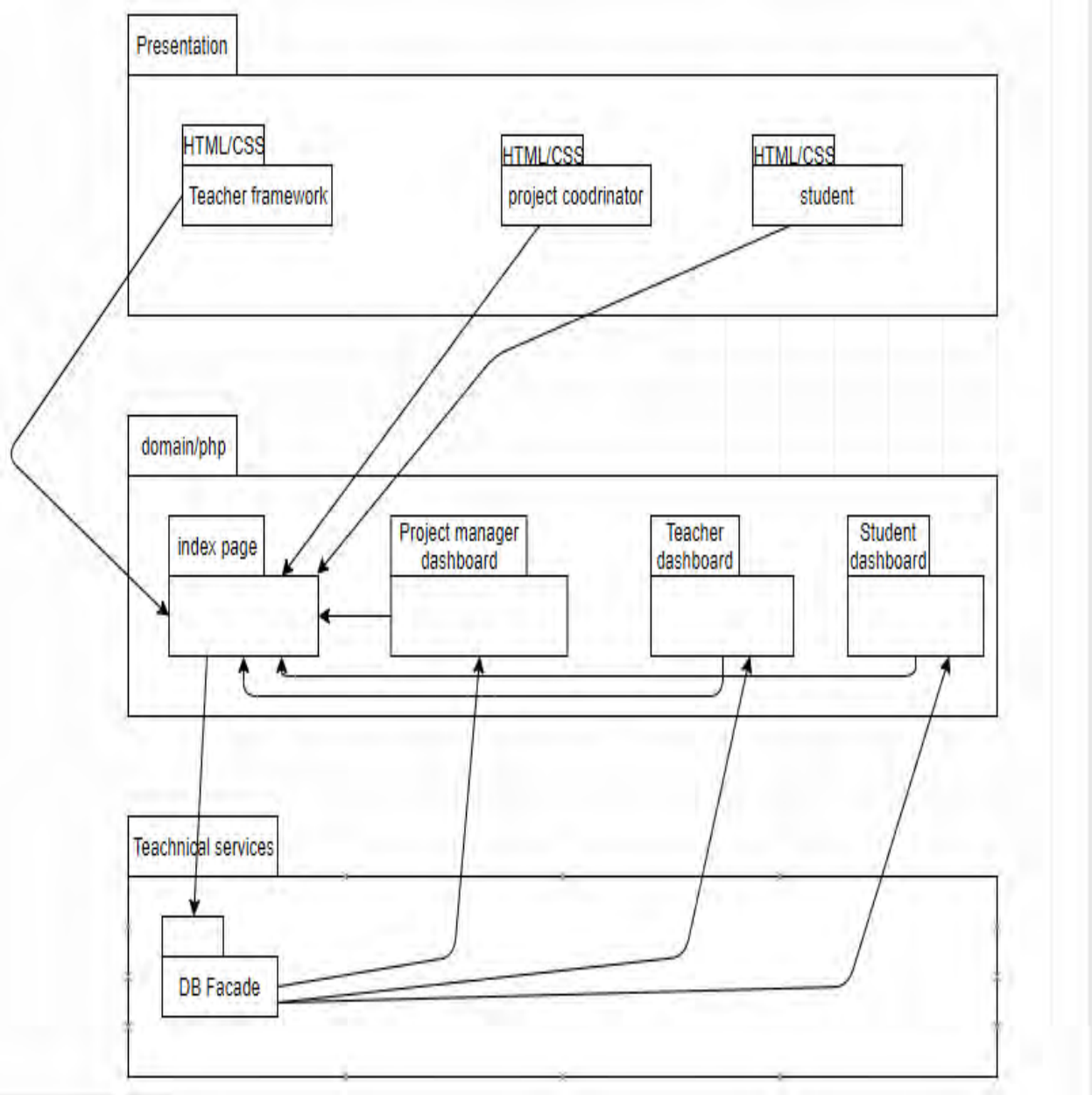


Figure 3. 1 System Architecture Design

3.3 Low-fidelity Interfaces

A **prototype** that is sketchy and incomplete, that has some characteristics of the target product but is otherwise simple, usually in order to quickly produce the **prototype** and test broad concepts.

Login Teacher (project coordinator)/Student

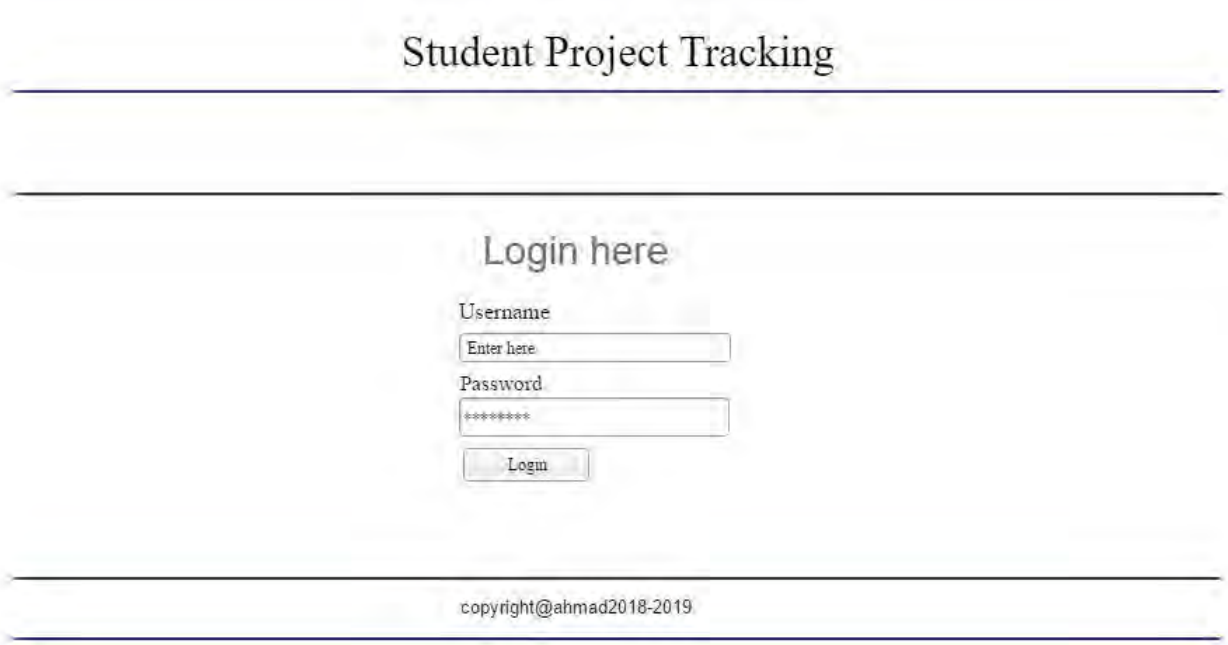


Figure 3. 2 Login Teacher (project coordinator)/Student

Teacher Panel

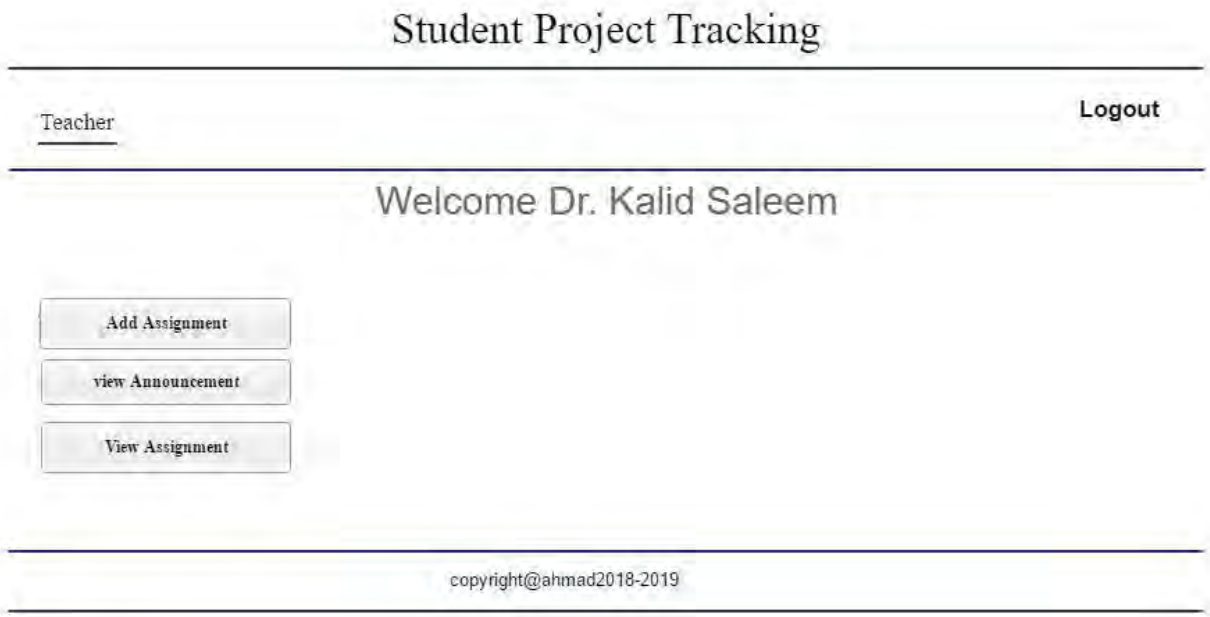


Figure 3. 3 Teacher Panel

Teacher (Project Coordinator)

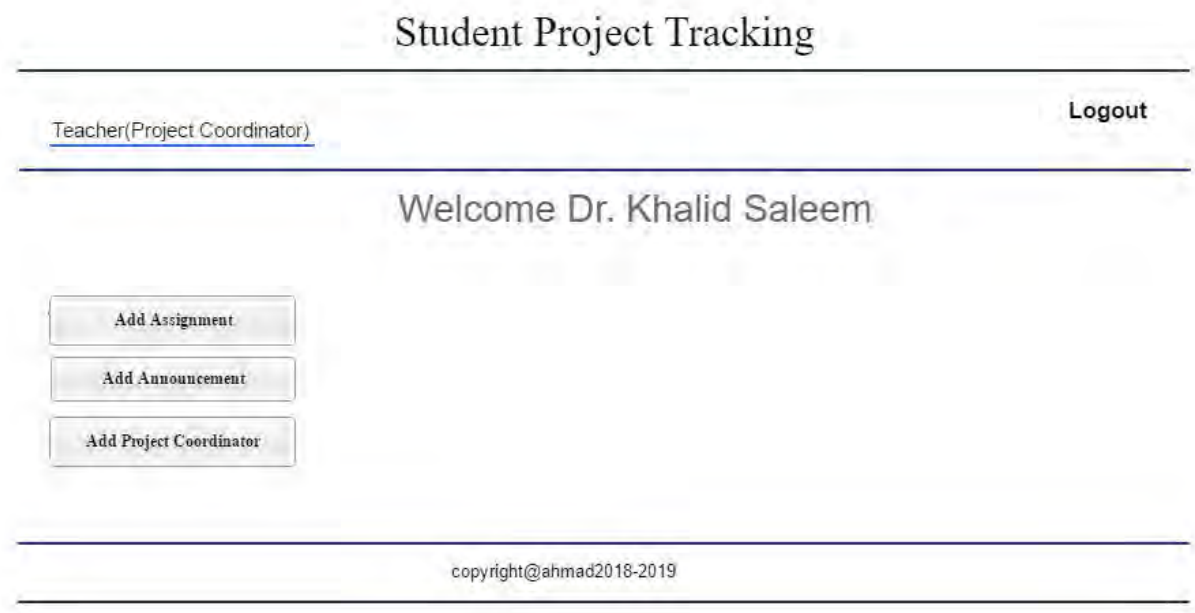


Figure 3. 4 Teacher (Project Coordinator)

Add Assignment

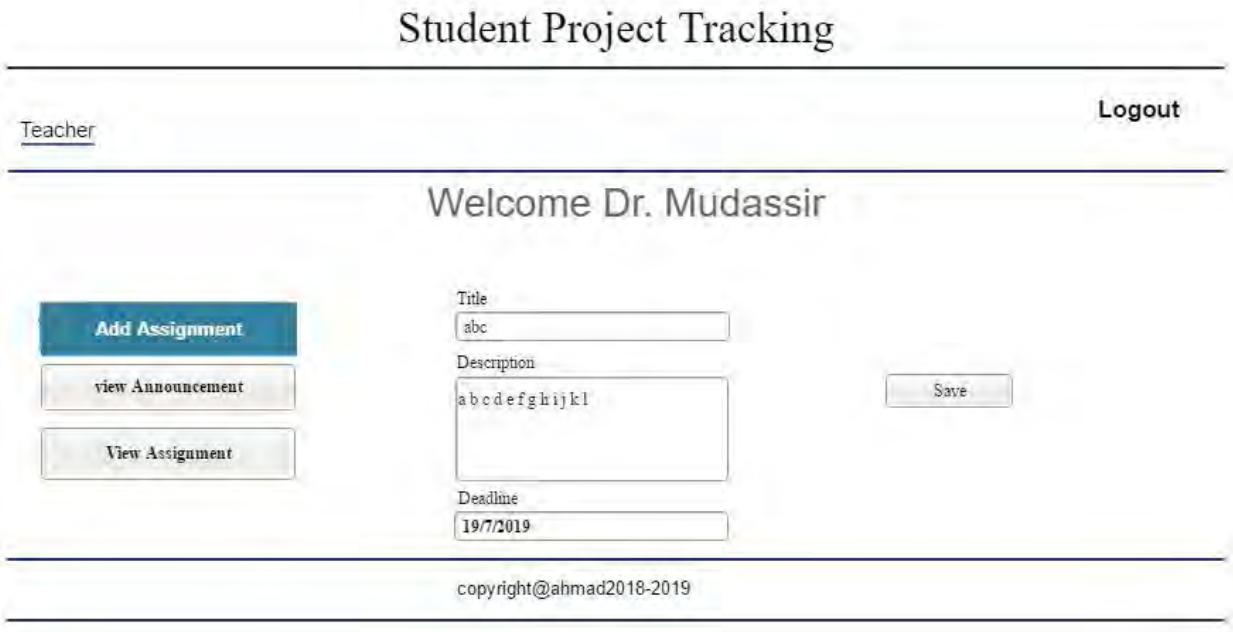


Figure 3. 5 Add Assignment

View Announcement

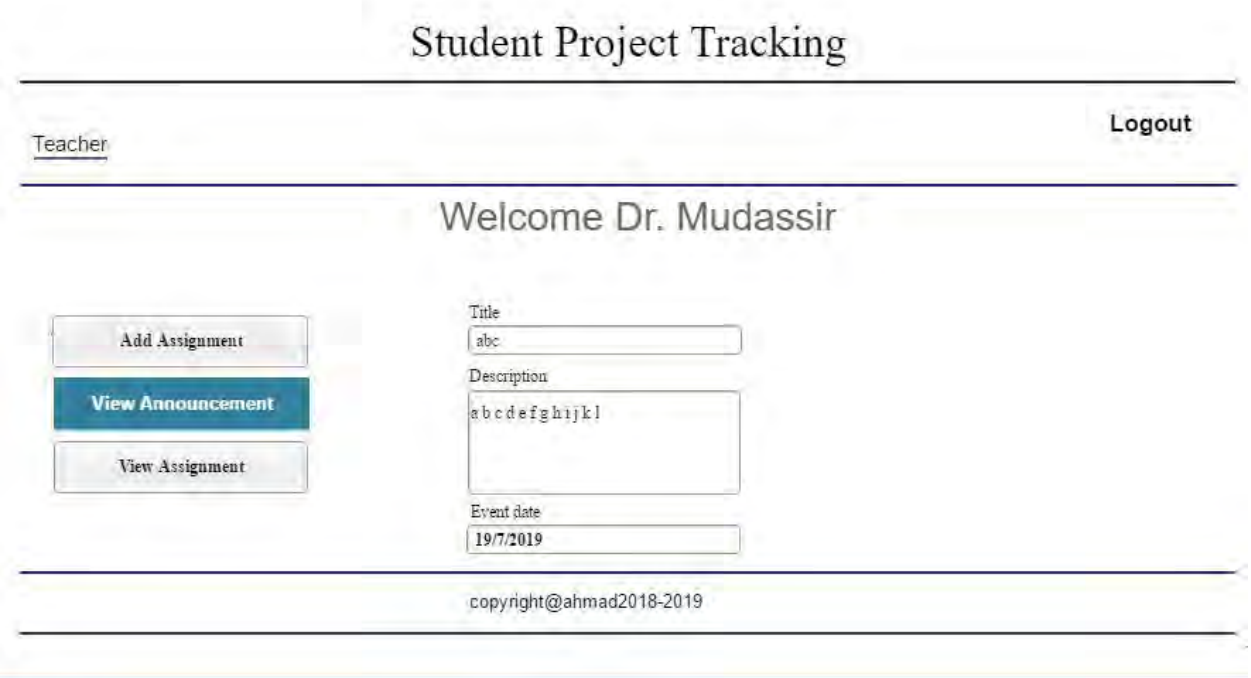


Figure 3. 6 View Announcement

View Assignment

Figure 3. 7 View Assignment

Add Announcement

Figure 3. 8 Add Announcement

Add Project Coordinator

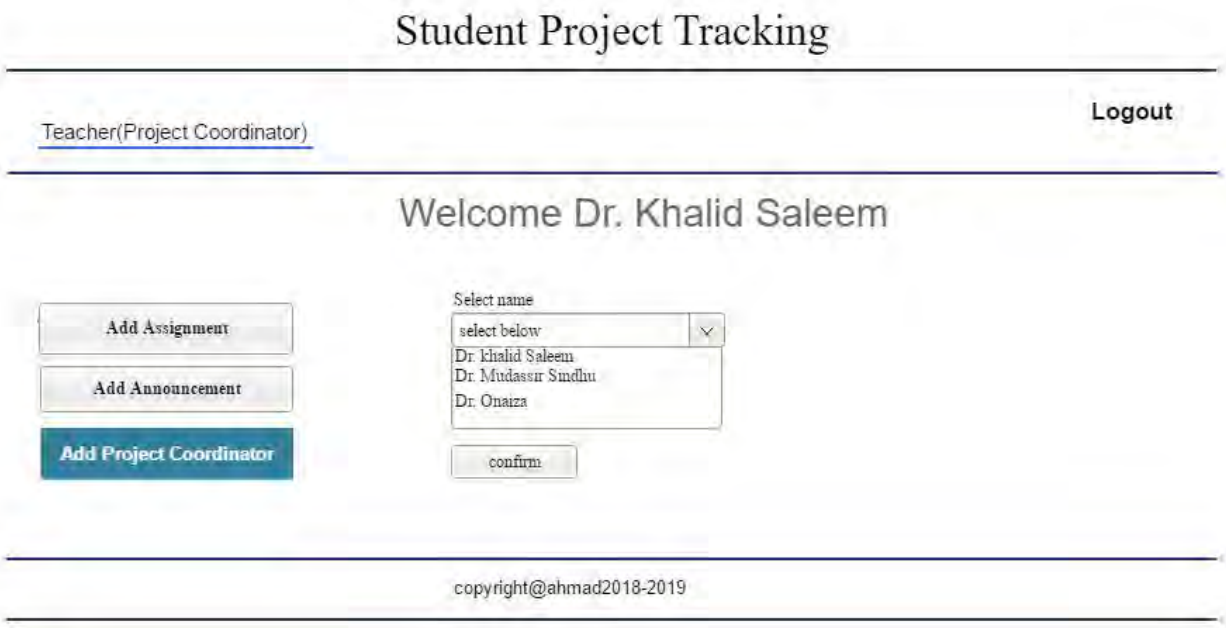


Figure 3. 9 Add Project Coordinator

View Announcement (Teacher)

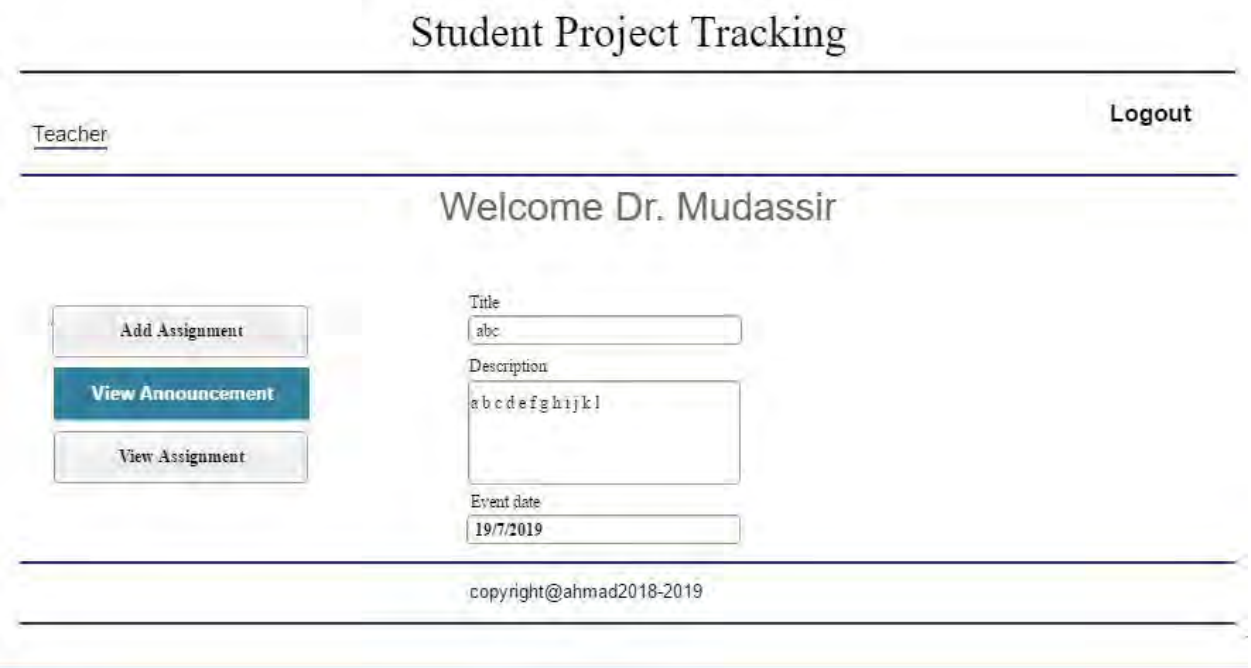


Figure 3. 10 View Announcement (Teacher)

View Assignment (Student)

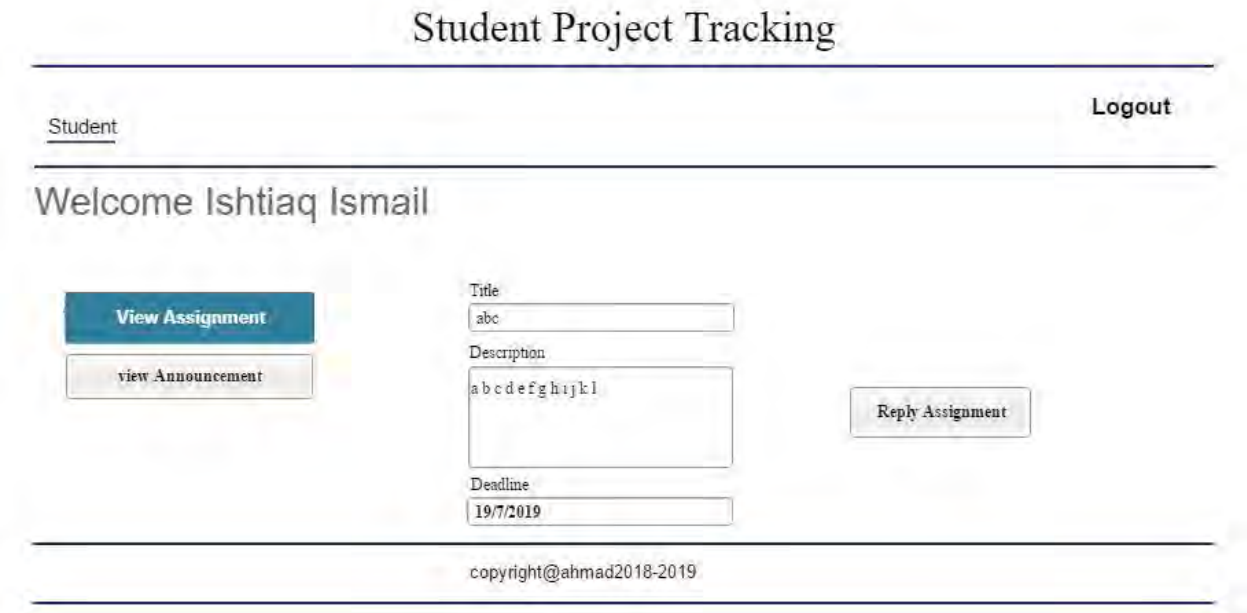


Figure 3. 11 View Assignment (Student)

View Announcement (Student)

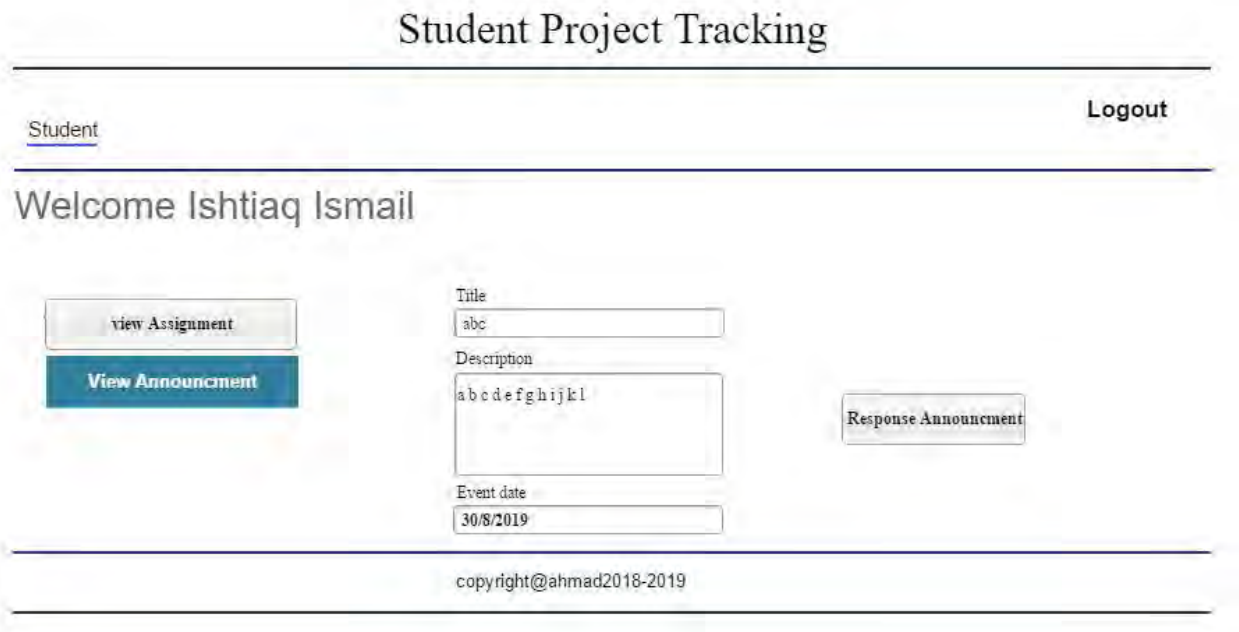


Figure 3. 12View Announcement (Student)

3.4 System Sequence Diagrams (SSD)

A SSD is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate, their order, and possible inter-system events. SSD is visual summaries of the individual use cases.

SSD-1 (login)

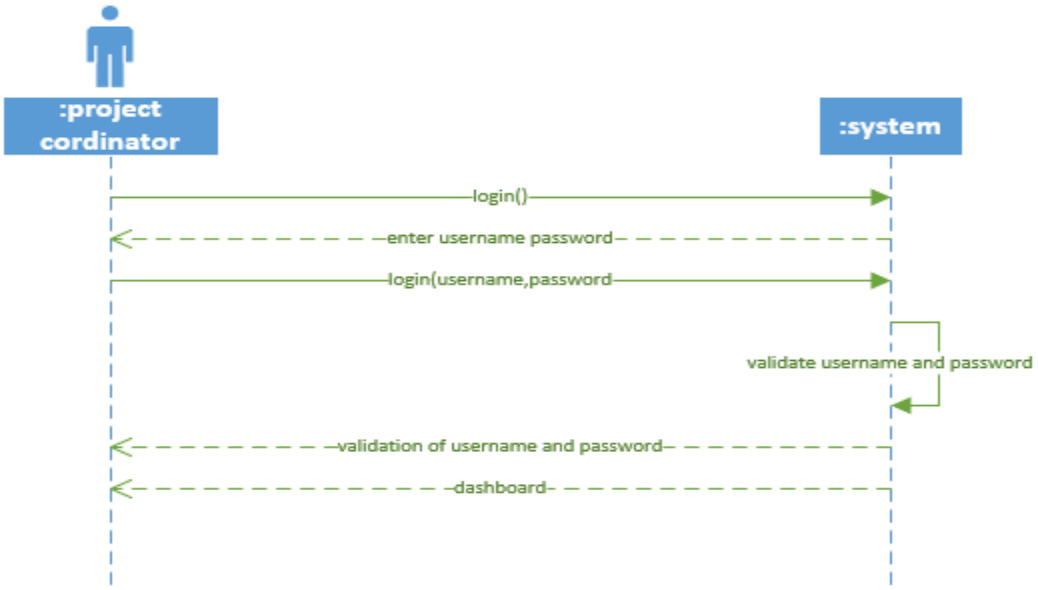


Figure 3. 13 Login SSD

SSD-2 (Assigning new project coordinator)

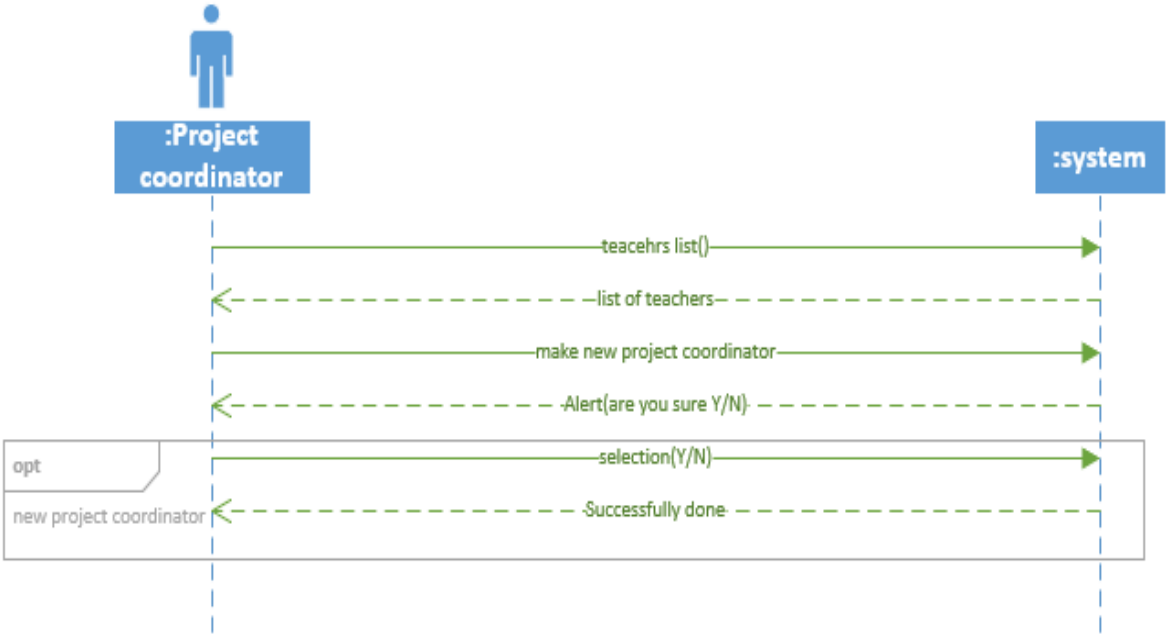


Figure 3. 14 Assigning new project coordinator SSD

SSD-3 (Add announcements)

Figure 3. 15 Make Announcement SSD

SSD-4 (Add project)

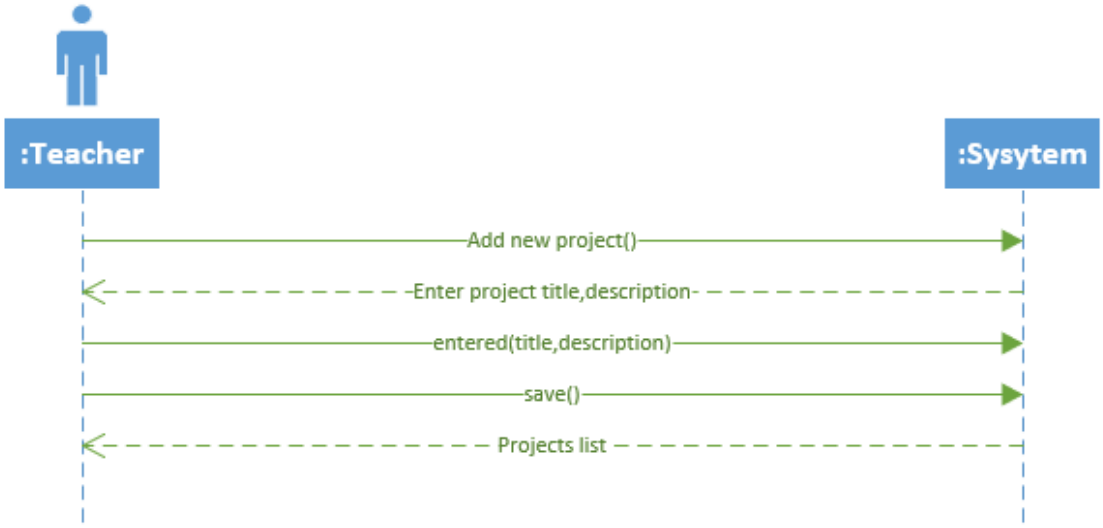


Figure 3. 16 Add new project SSD

SSD-5 (View student list)

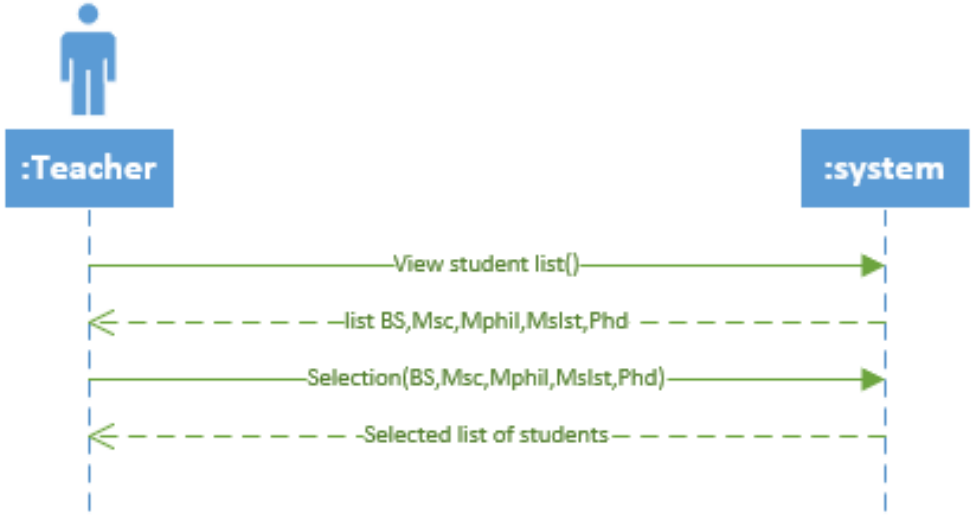


Figure 3. 17 View Student List SSD

SSD-6 (Update deadline)

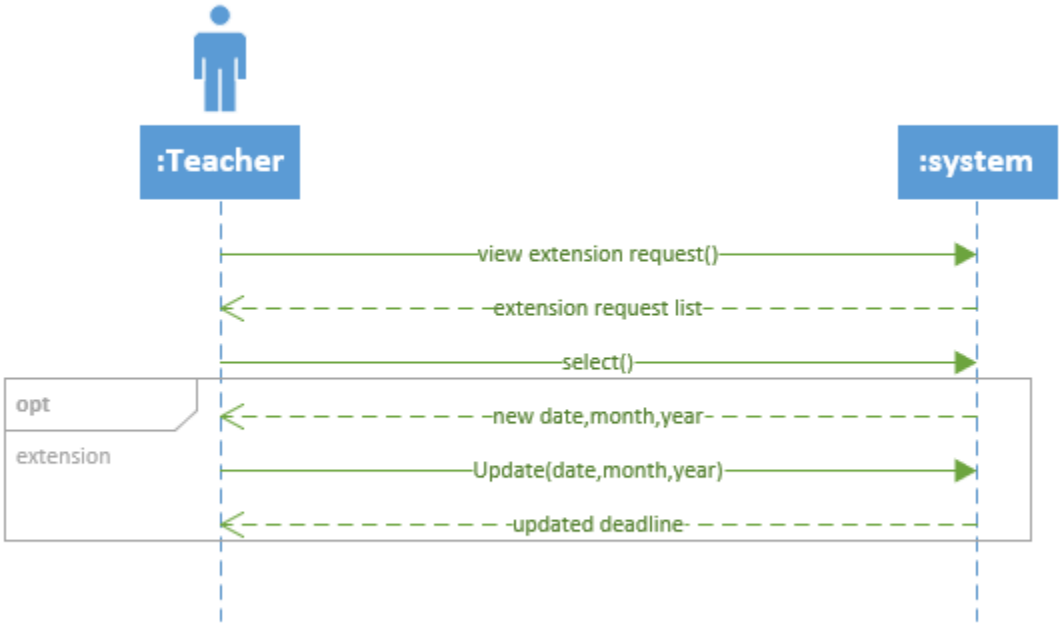


Figure 3. 18 Update deadline SSD

SD-7 (Response to announcement)

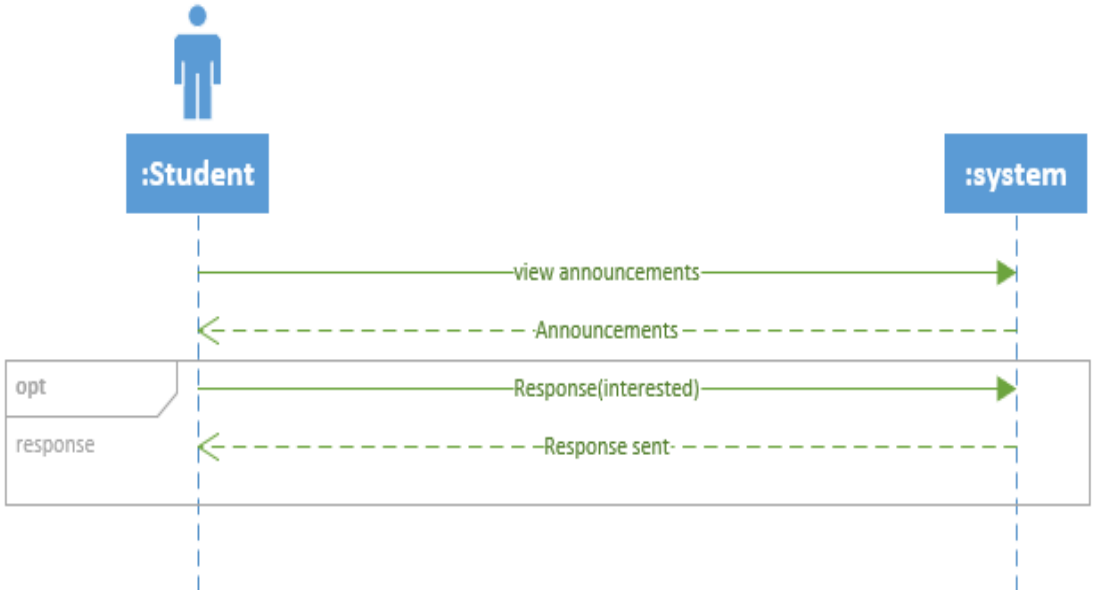


Figure 3. 19 Response to announcement SSD

3.5 Entity Relationship Diagram

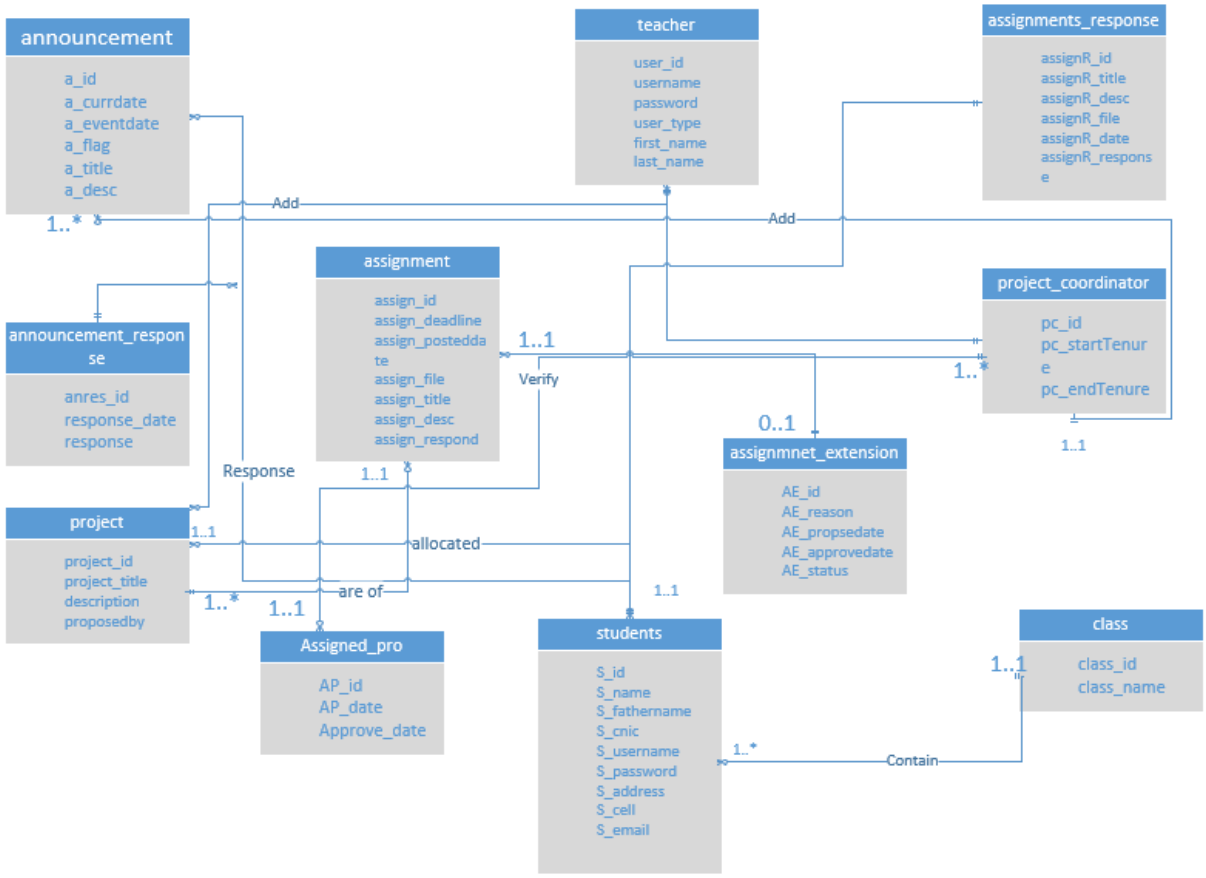


Figure 3. 20 ERD

3.6 Data flow diagrams (DFD)

A data flow diagram represents the way information flows through a process or system. It contains data inputs and outputs, data stores, and the various sub-processes the data moves through. I am following “Yourdon and Coad” notation for making DFD.

Zero level DFD (Context level)

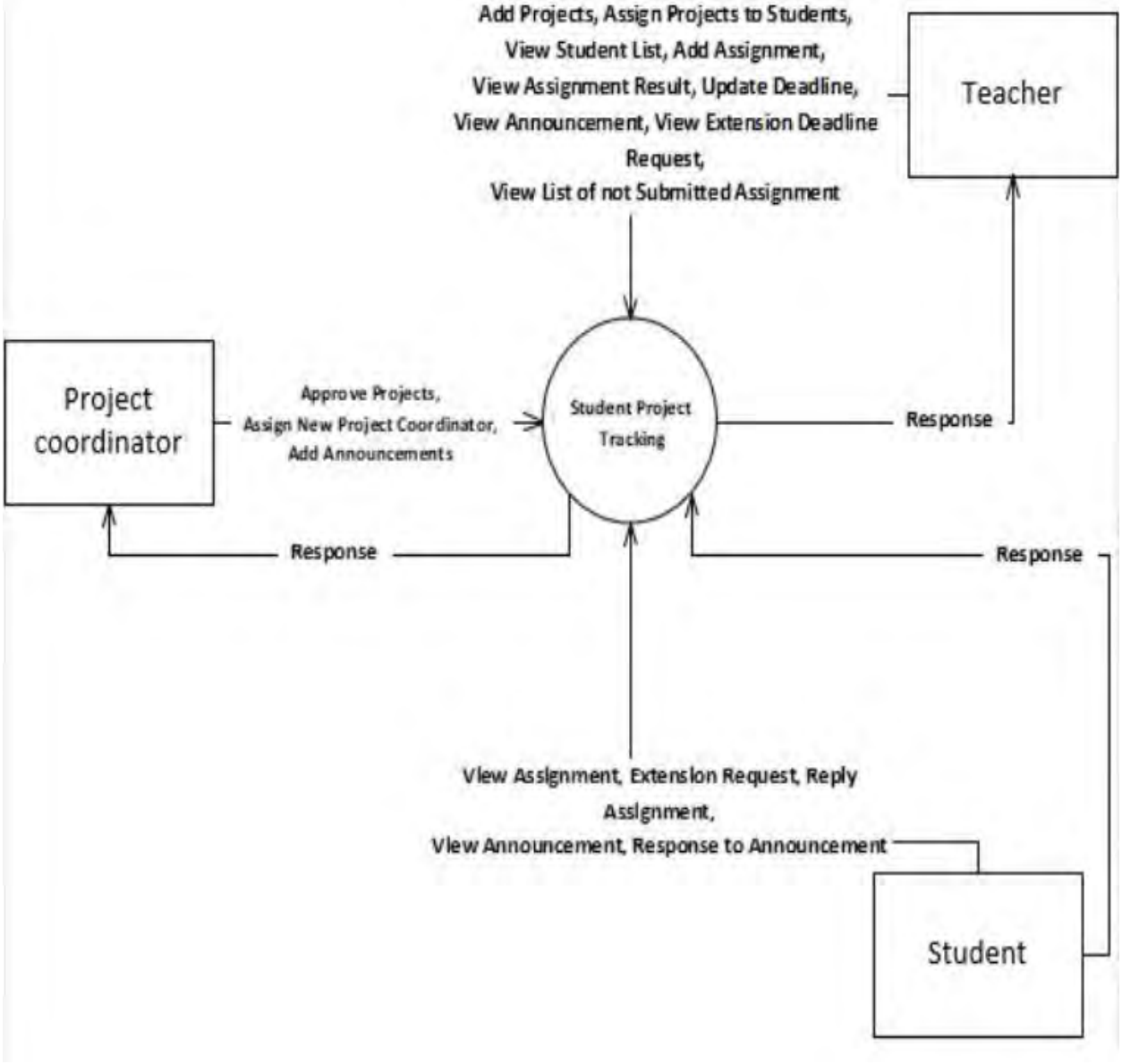


Figure 3. 21 Zero level DFD

First level DFD (Project coordinator)

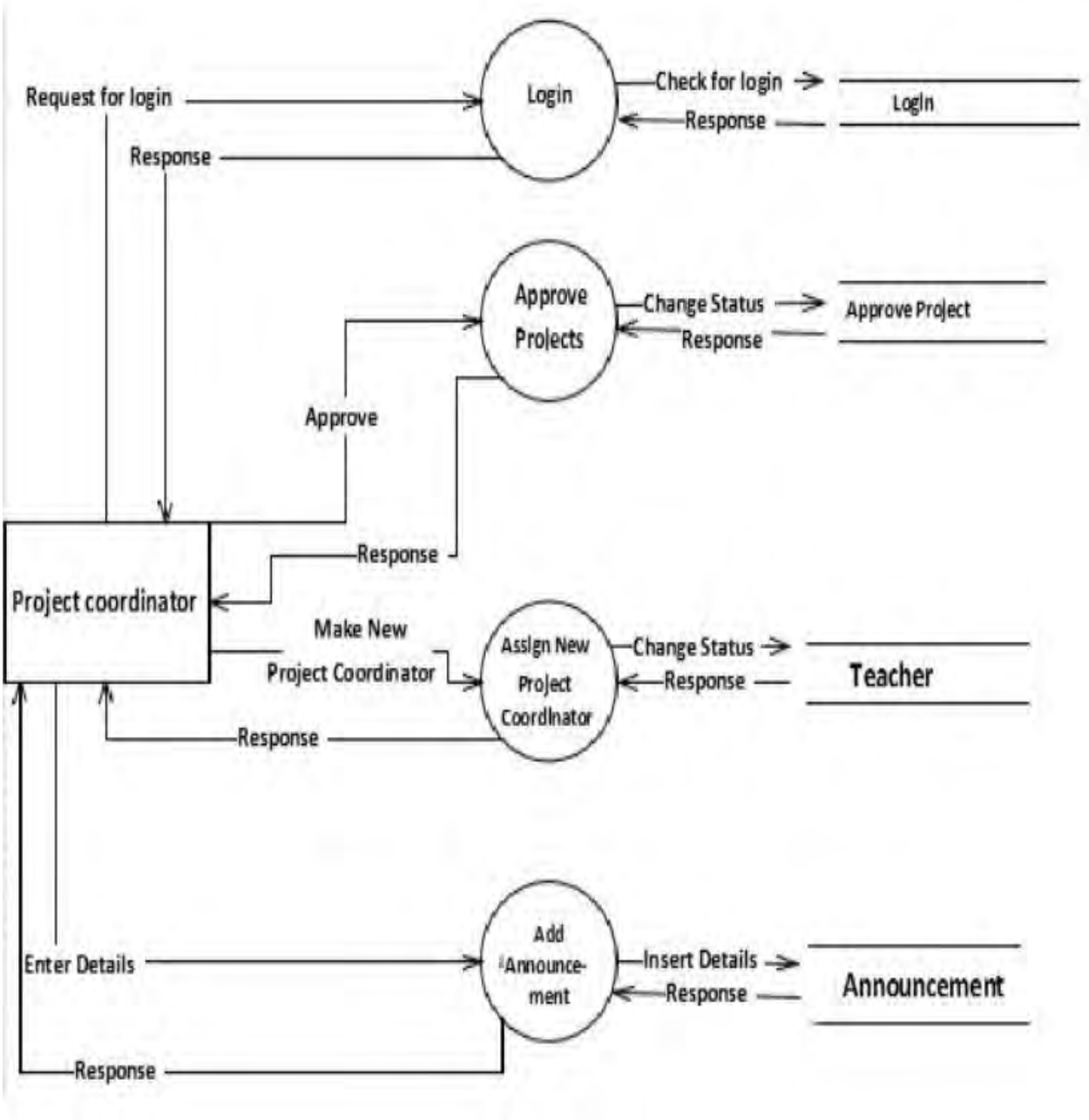


Figure 3. 22 First level DFD (Project Coordinator)

First level DFD (Teacher)

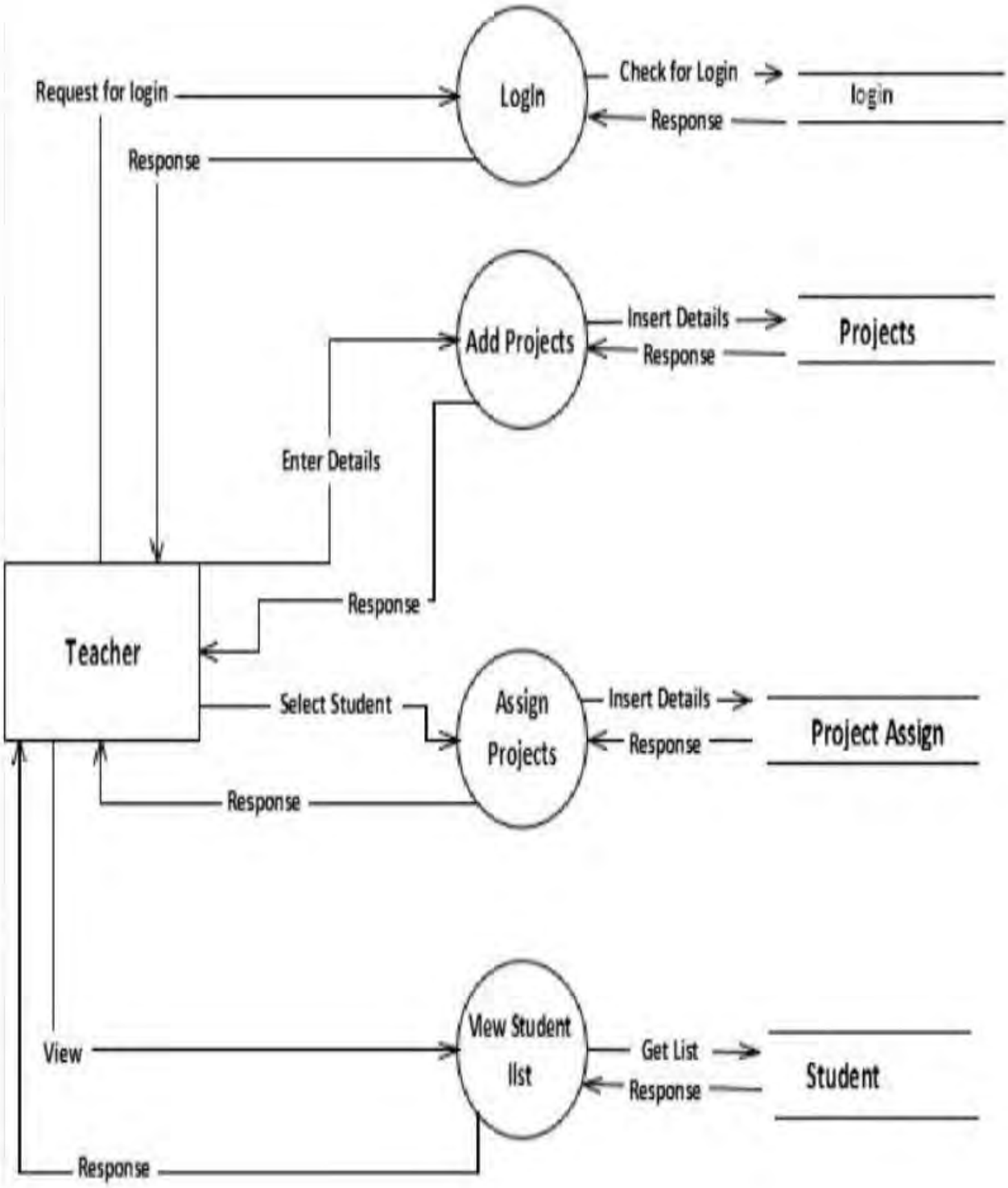


Figure 3. 23 First level DFD (Teacher) [1/2]

First level DFD (Teacher)

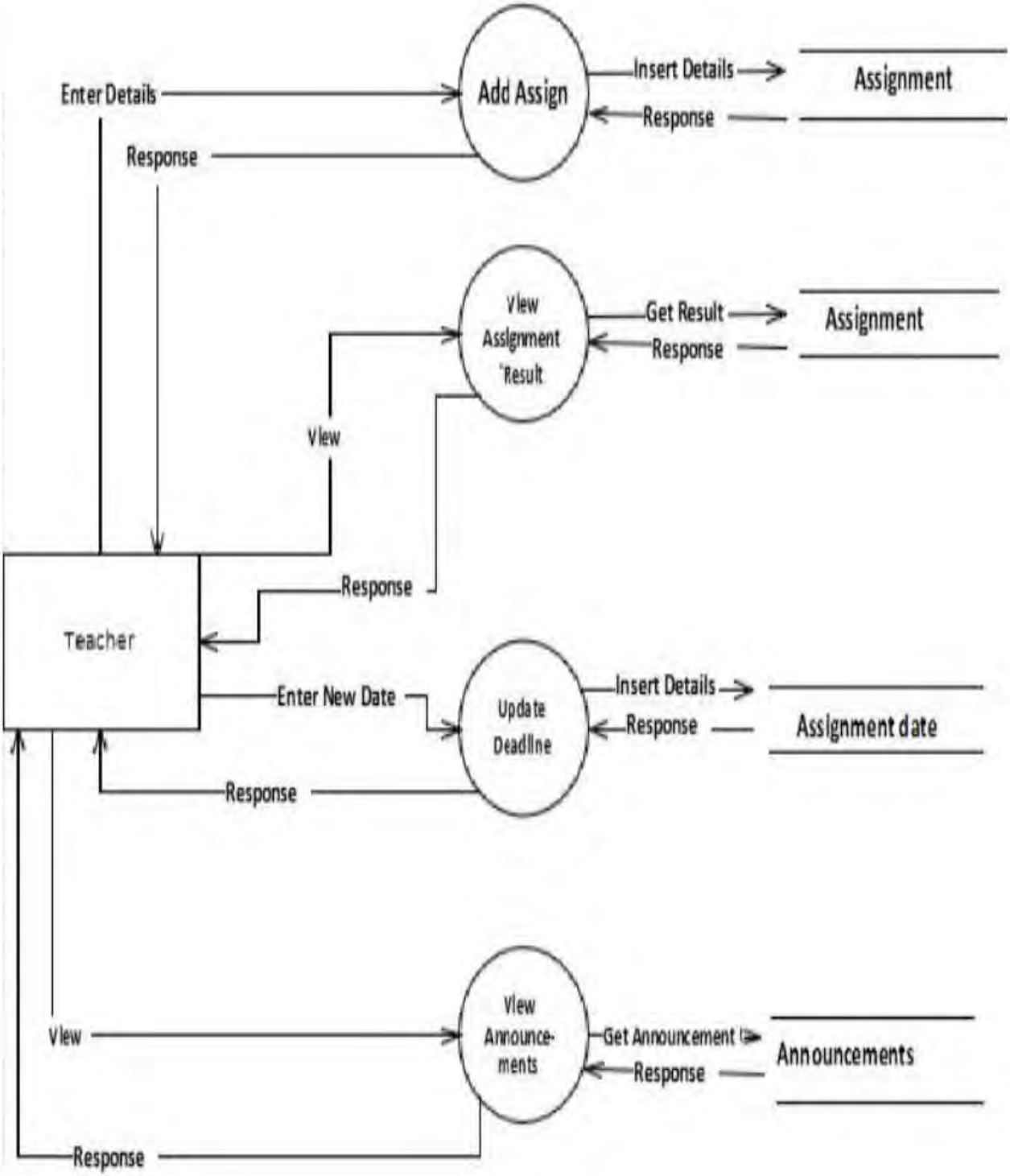


Figure 3. 24 First level DFD (Teacher) [2/2]

First level DFD (Student)

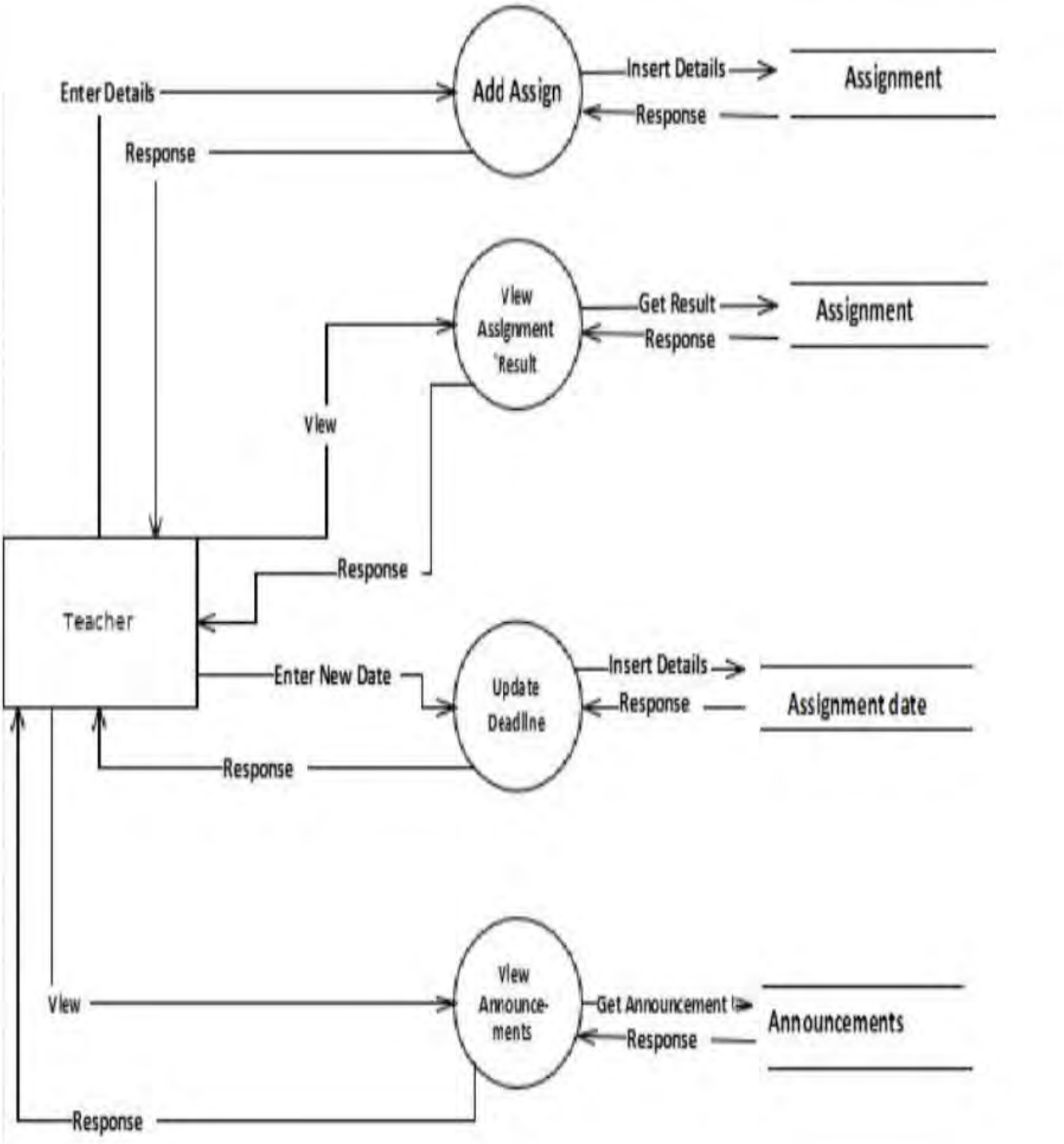


Figure 3. 25 First level DFD (Student)

3.7 Activity diagrams

An activity is a state of doing something. The activity diagram describes the sequencing of activities. Activity diagram depicts the dynamic behavior of a system or part of a system through the flow of control between actions that the system performs. It is similar to a flowchart except that an activity diagram can show concurrent flows. Activity diagrams for this system is given in following:

Teacher (Activity diagram)

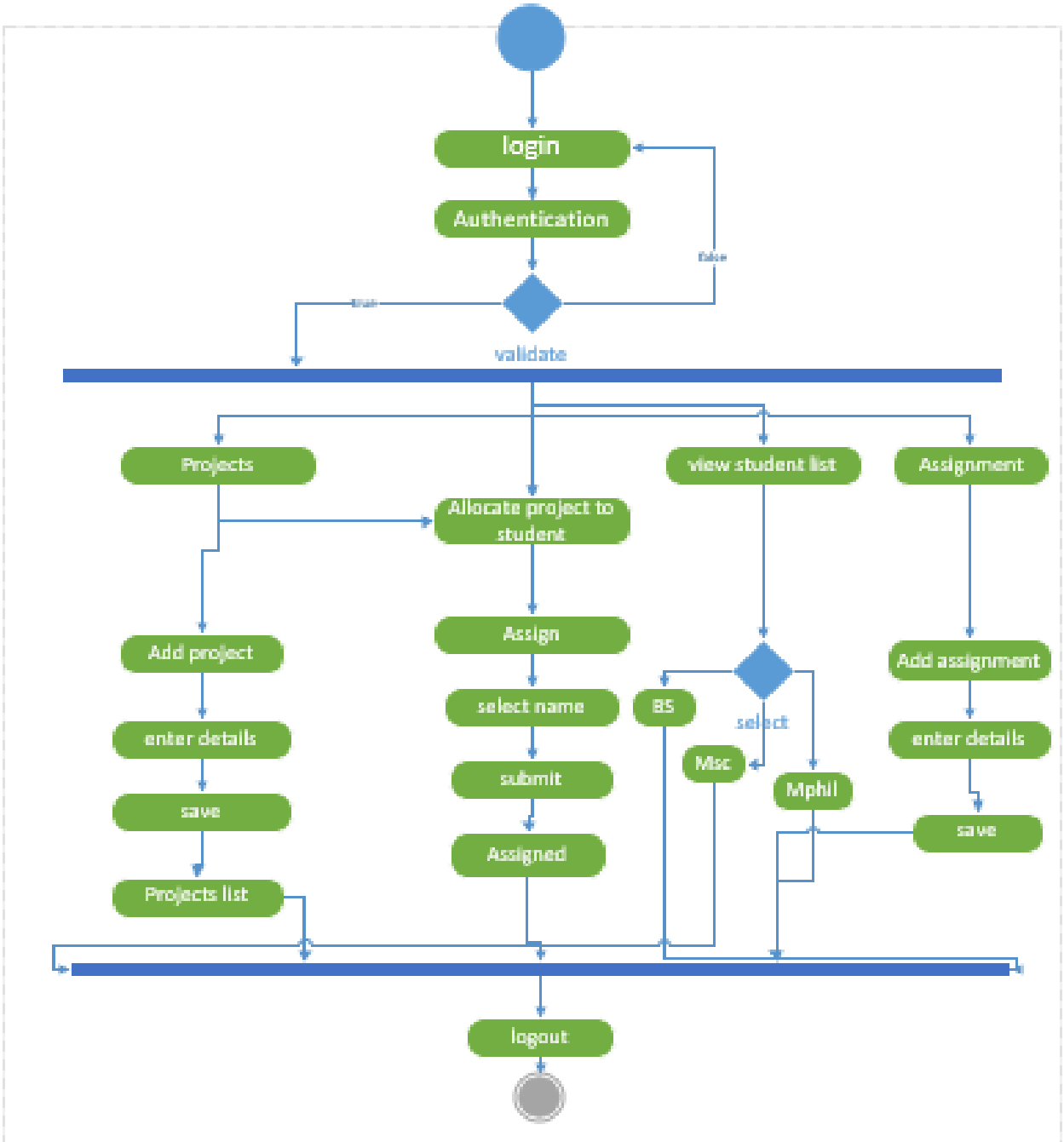


Figure 3. 26 Activity Diagram (Teacher)

Student (Activity diagram)

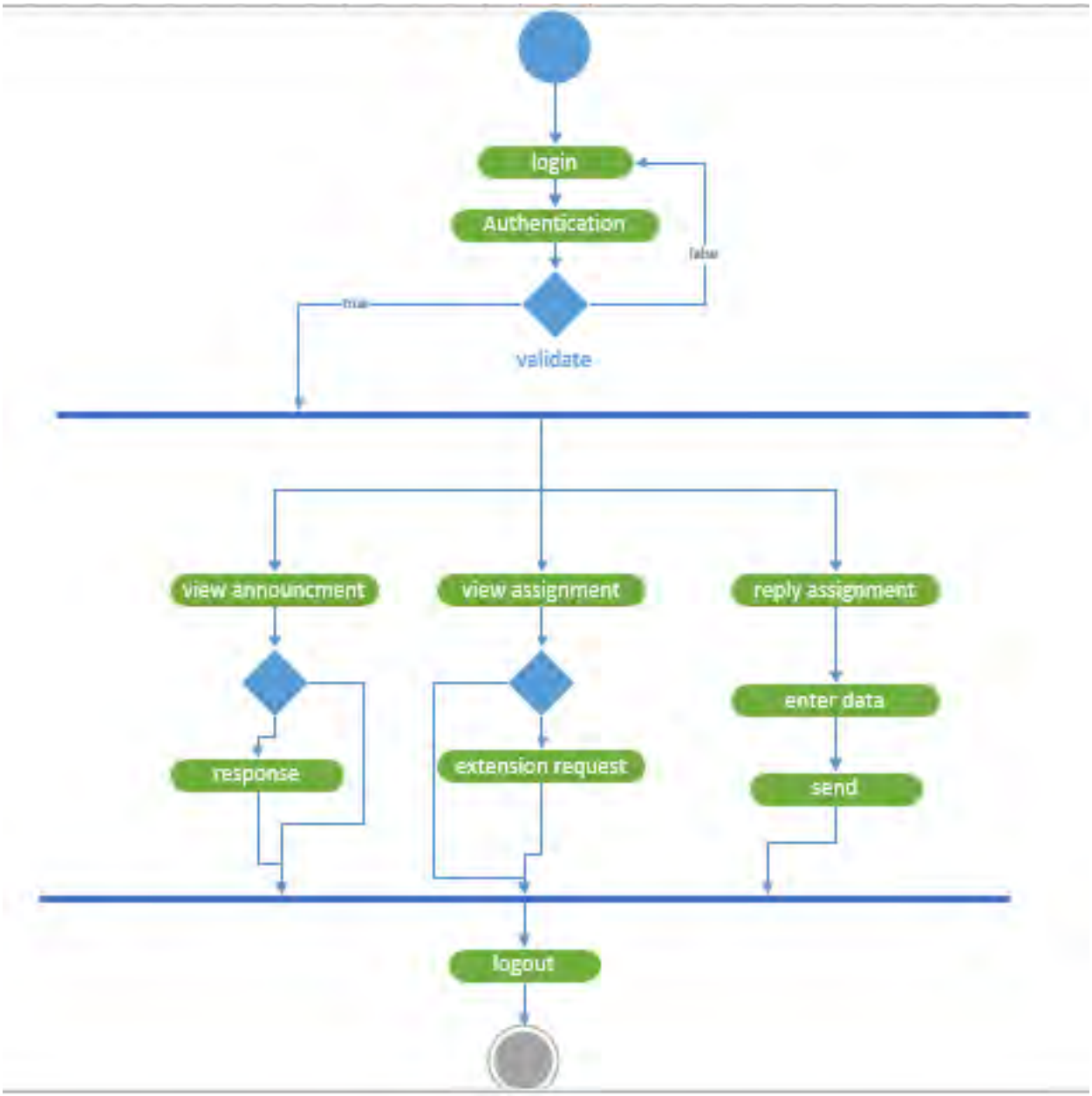


Figure 3. 27 Activity Diagram (Student)

3.9 Database Design

MySQL database is using for storing all information for the running of web-bases e-cosmetic by Fair cosmetics. There are twelve tables in which information will be stored and retrieved for running of system.

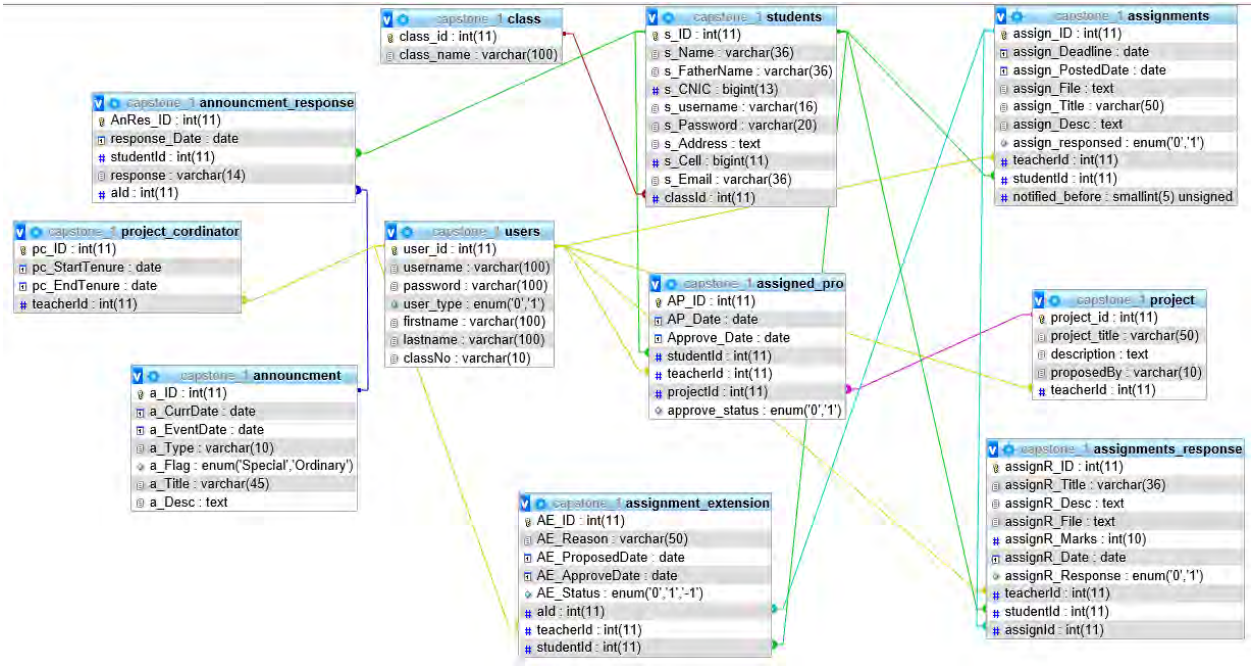


Figure 3. 28 database design

3.9 Summary

Chapter 3 briefly describes the complete description of software design. It then elaborates the architectural design and detailed description of components of system.

Chapter 4 describes the project implementation for “Student Project Tracking” for Department of Computer Science. It has list of tools, language and used in the development of project.

CHAPTER 4

Software Implementation Plan

CHAPTER 4 Software Implementation Plan

This chapter describes the project implementation for "Student Project Tracking" for Department of Computer Science. It has list of tools, language and used in the development of project.

4.1. Language Selection

This project implements in following languages:

- **HTML/CSS/Bootstrap**
  Use for designing front-end web pages.
- **Ajax**
  Use for the client-side browser to communicate with the server without having to perform a page
  refresh.
- **JQuery/JavaScript**
  Use for scripting and validation.
- **PHP**
  Use for server side scripting
- **MySQL**
  Use for database.

4.2. Tools Selection

List of tools used for implementation of the project is given below:

- **Notepad++**

  Use for writing code for front-end and backend languages.

- **Web browser**

  Use for running of OPD web based application.

- **XAMPP Server**

  Use for running of database.

4.3 Summary

Chapter 4 briefly describes the project implementation for “Student Project Tracking” for Department of Computer Science. It has list of tools, language and used in the development of project.

Chapter 5 describes software testing and software testing processes. This chapter further elaborates the acceptance test cases which are used to test the functional and non-functional requirements after coding of software.

CHAPTER 5

Software Test Document

CHAPTER 5 Software Test Document

This chapter describes software testing and software testing processes. This chapter further elaborates the acceptance test cases which are used to test the functional and non-functional requirements after coding of software.

5.1 Introduction

Software test document involves the documentation of artefacts that should be developed before or during the testing of software. Software testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Test Approach

Manual testing includes testing a software manually without using any automated tool or any script. The tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug. There are different stages for manual testing such as unit testing, system testing, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it. Unit testing is the process of testing program components, such as methods or object and classes. Individual method or function are simplest type of the component. System testing during development involves integrating components to create a version of the system and then testing the integrated system. System testing checks that components are compatible, interact correctly and transfer the right data at the right time across their interfaces. Test-first development is an approach to development where tests are written before the code to be tested. Small code changes are made and the code is refactored until all tests execute successfully.

5.2 Test Plan

Test planning is an activity that ensures that there is initially a list of assignments and milestones in a baseline plan to track the progress of the project. Test plan determines the scope and the risk that need to be tested and are not to be tested. Deciding fail and pass criteria.

Testing Tools and Environment

A testing environment is a setup of software and hardware on which testing of the newly built product is performed. This setup consists of the physical setup which includes hardware, and logical setup that includes operating system, client operating system, database server, front end running environment, browser (if web application), IIS (version on server side) or any other software components required to run this software product. This testing setup is to be built on both the ends the server and client.

5.3 Test Cases

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement. Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

Login

ID	T1
Description	Checking either teacher student can login to the system or not.
Tester	User
Setup	1. Register teacher with Username: Khalid Password: Khalid. 2. Register another teacher with Username: mudassir Password: mudassir
Instructions:	1. Enter Username Khalid and Khalid. 2. Press login 3. Enter Username: mudassir and mudassir

	4. Press login
Expected Results	Teacher Khalid should log in successfully Teacher mudassir should log in successfully

Table 5. 1 Login

Add Assignment

ID	T2
Description	Teacher will add assignment for the Student.
Tester	User
Setup	1. Teacher will press the assignment button. 2. Teacher will select add assignment. 3. Teacher will enter title deadline and description of assignment. 4. Assignment will be send to student.
Instructions:	1. Select the Assignment. 2. Select add assignment. 3. Enter details. 4. Press save.
Expected Results	1. Assignment interface will be appear. 2. Add assignment interface will appear. 3. Details have been entered. 4. Assignment is sent to student.

Table 5. 2Add Assignment

View Announcement

ID	T3
Description	Teacher and student both can view announcement.
Tester	User
Setup	1. Teacher will select the program 2. Teacher will view the student.
Instructions:	1. Select the program. 2. Press on announcement button.

Expected Results	1. Program is selected. 2. Announcement interface is appeared. 3. Assignment is viewed.

Table 5. 3 View Announcement

Add Project

ID	T4
Description	Teacher can add project with its title, proposed by and description
Tester	User
Setup	1. Teacher will press the project button. 2. Teacher will select add project. 3. Teacher will enter Project title proposal and description of project. 4. Teacher will press save button to add project.
Instructions:	1. Select the project. 2. Select the add project. 3. Enter details. 4. Press save.
Expected Results	1. Project interface is appeared. 2. Add project interface is appeared. 3. Details have been entered. 4. Project is saved.

Table 5. 4 Add project

Add Announcement

ID	T5
Description	Project coordinator can add announcement with event date title type and description.
Tester	User
Setup	1. Project coordinator will press announcement button. 2. Project coordinator will press add announcement button. 3. Project coordinator will enter event date, title, type and description. 4. Project coordinator will press save button to add announcement.
Instructions:	1. Select the announcement. 2. Select the add announcement. 3. Enter details. 4. Press save.
Expected Results	1. Announcement interface is appeared. 2. Add Announcement interface is appeared. 3. Details have been entered. 4. Announcement is added.

Table 5. 5 Add Announcement

List of not Submitted Assignment

ID	T6
Description	Teacher can view list of students who have not submitted assignments.
Tester	User
Setup	1. Teacher will press the Assignment button. 2. Teacher will press not submitted assignment button. 3. Teacher will view list.
Instructions:	1. Select the assignment. 2. Select the not submitted assignment. 3. View the list.
Expected Results	1. Assignment interface is appeared. 2. Not submitted assignment interface is appeared. 3. List of the student is appeared.

Table 5. 6 list of not Submitted Assignment

CHAPTER 6

Conclusions and Future Enhancements

CHAPTER 6 Conclusions and Future Enhancements

This document describes the project conclusions and future enhancements i.e. what type of new features can be added with time.

6.1. Conclusions

Project coordinator can make new project coordinator. Project coordinator approves project assigned to any student by teacher. Teacher can assign an assignment to any student. Teacher accept for the extension in deadline of assignment. Student views his assignment and uploads solution. Student requests for extension in deadline.

6.2. Future enhancement

This application is not containing any type of SMS alert. So, it can be made that SMS alert will provide to any student when new assignment added for him.

Appendix

Application Screen Shots

Application screen shots helps to understand the system, its design, input fields and output fields. These screenshots help to understand the application views. These are the screenshots of the application Student Project Tracking.

Project Coordinator Dashboard

Figure A. 1 Project Coordinator Dashboard

Teacher Dashboard

Figure A. 2 Teacher Dashboard

Student Dashboard

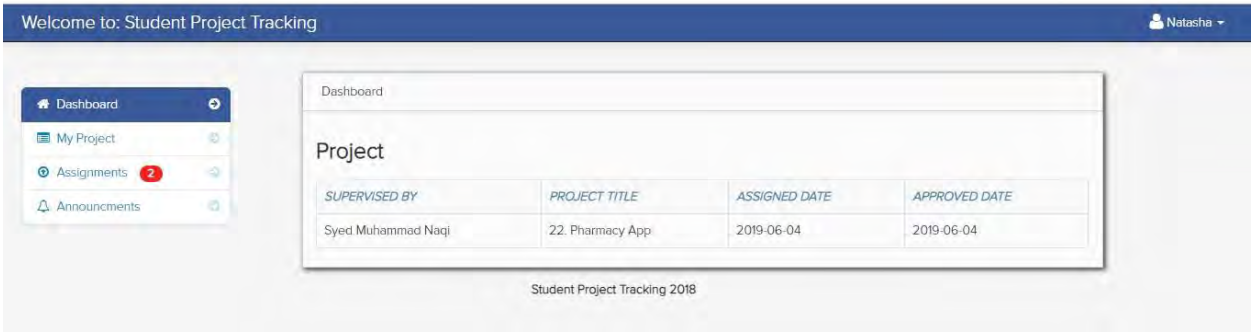


Figure A. 3 Student Dashboard

Verify student Project Allocation

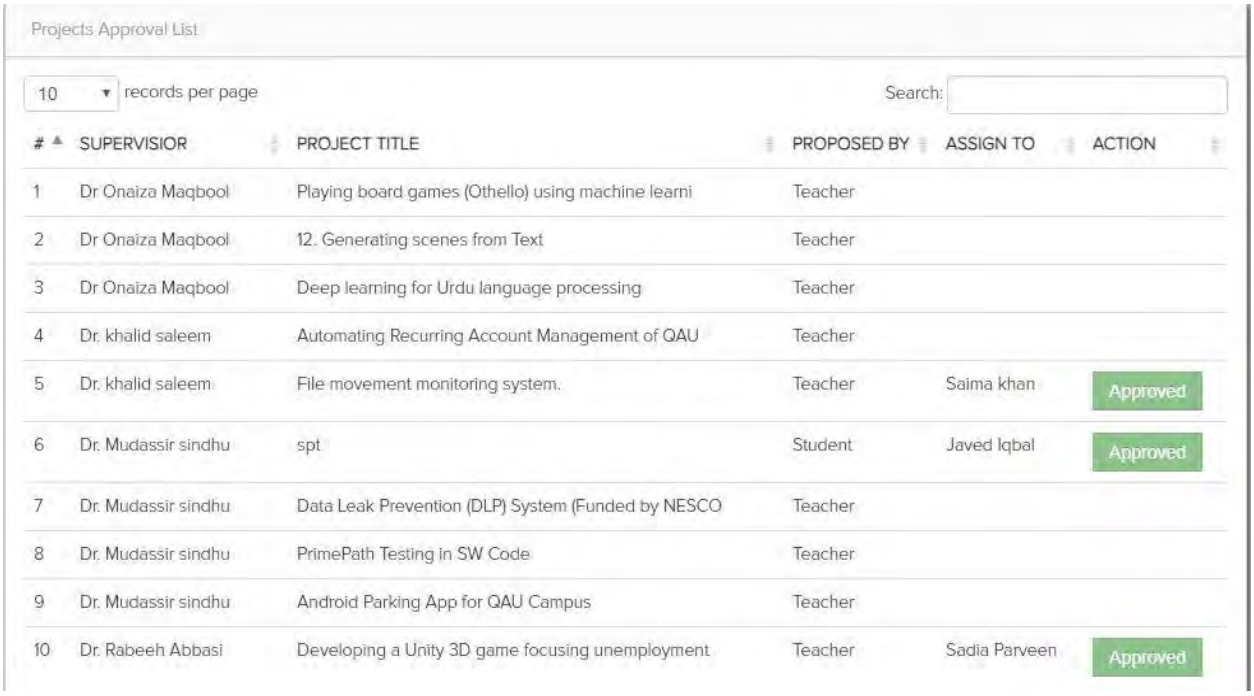


Figure A. 4 Verify Student Project Allocation

Assign New Project Coordinator

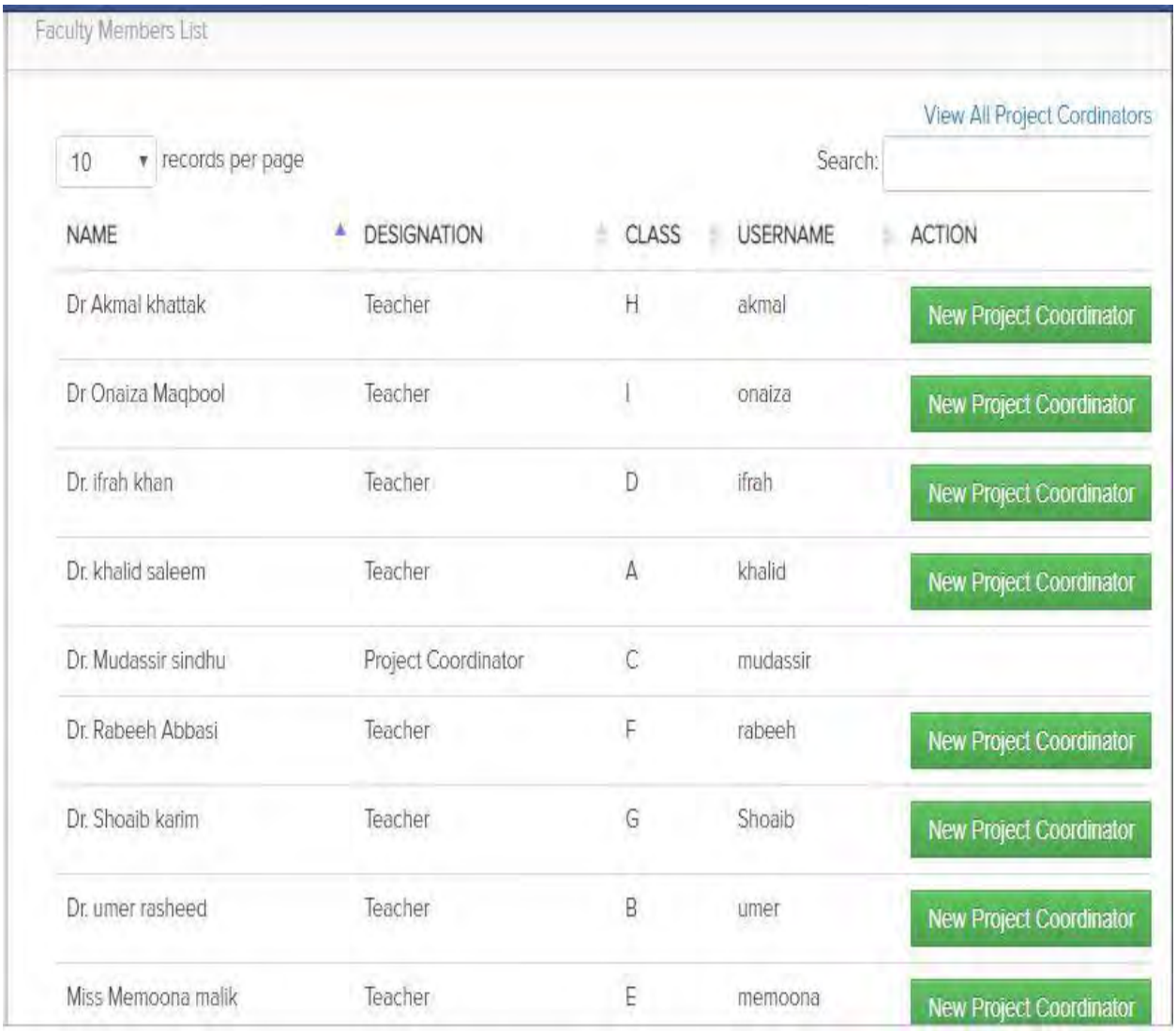


Figure A. 5 Assign New Project Coordinator

Project Coordinator Log

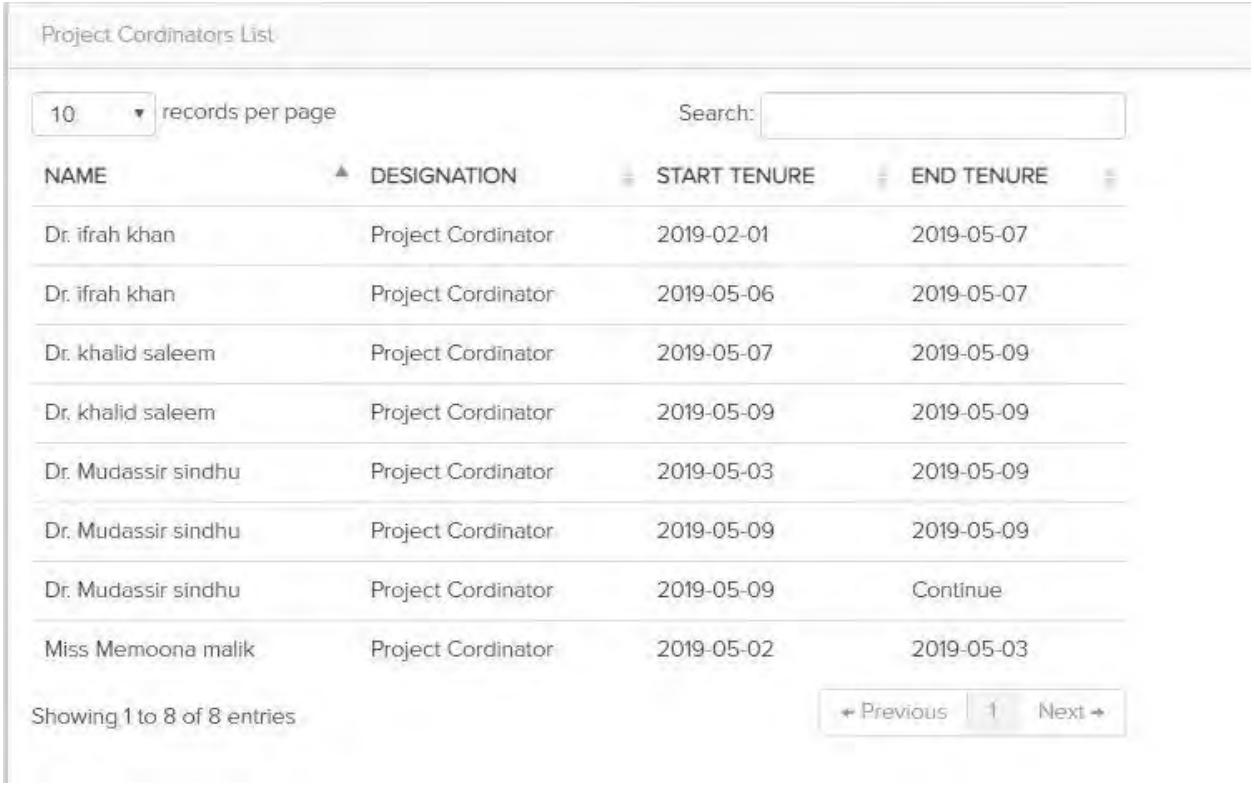


Figure A. 6 Project Coordinator Log

Add Announcements

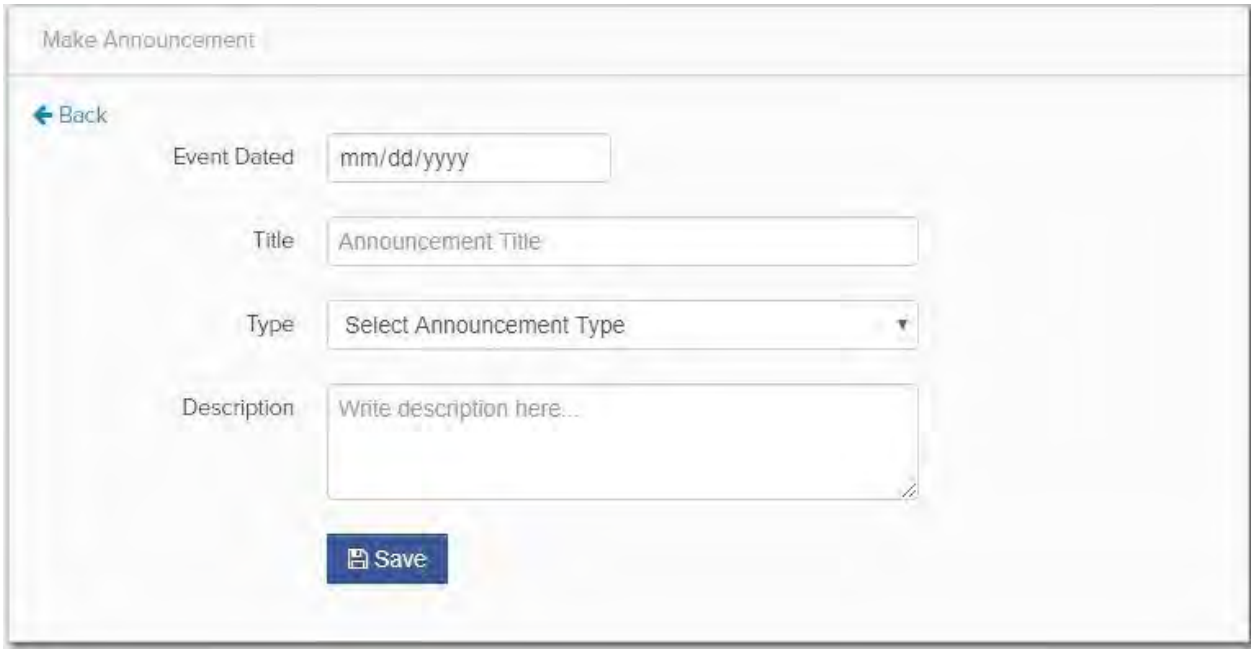


Figure A. 7 Add Announcements

Add Projects

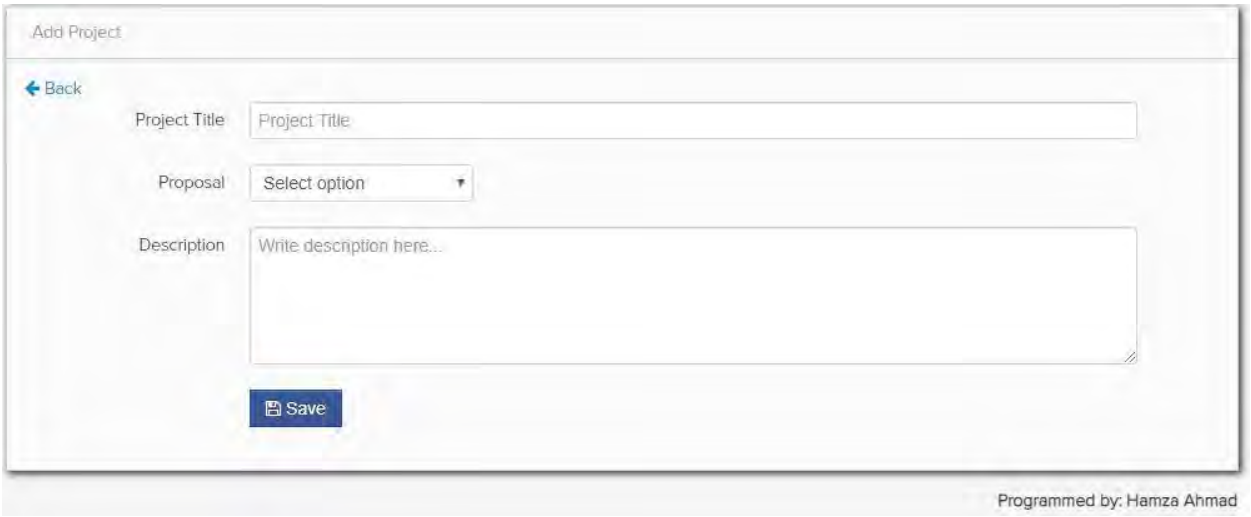


Figure A. 8 Add Projects

Allocate Project to Student

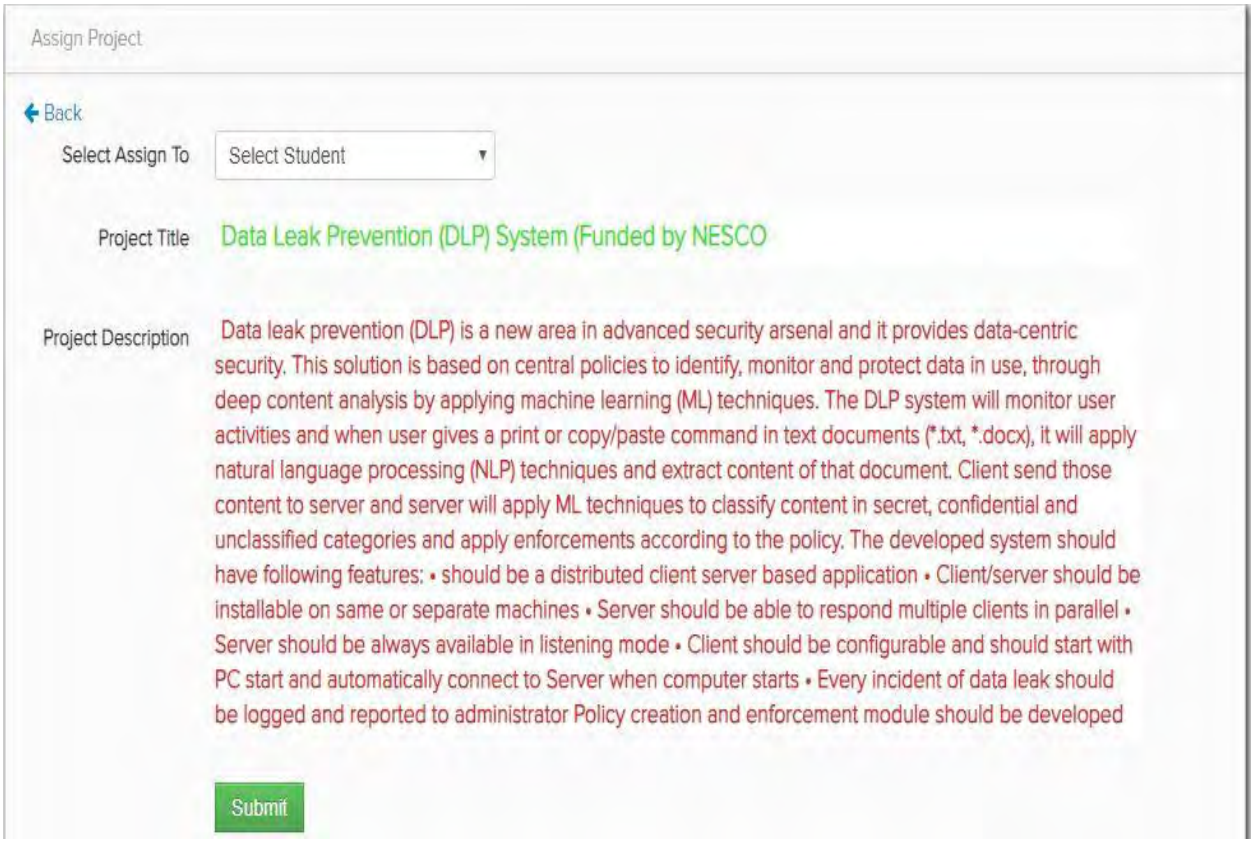


Figure A. 9 Allocate Project to Student

View Student List

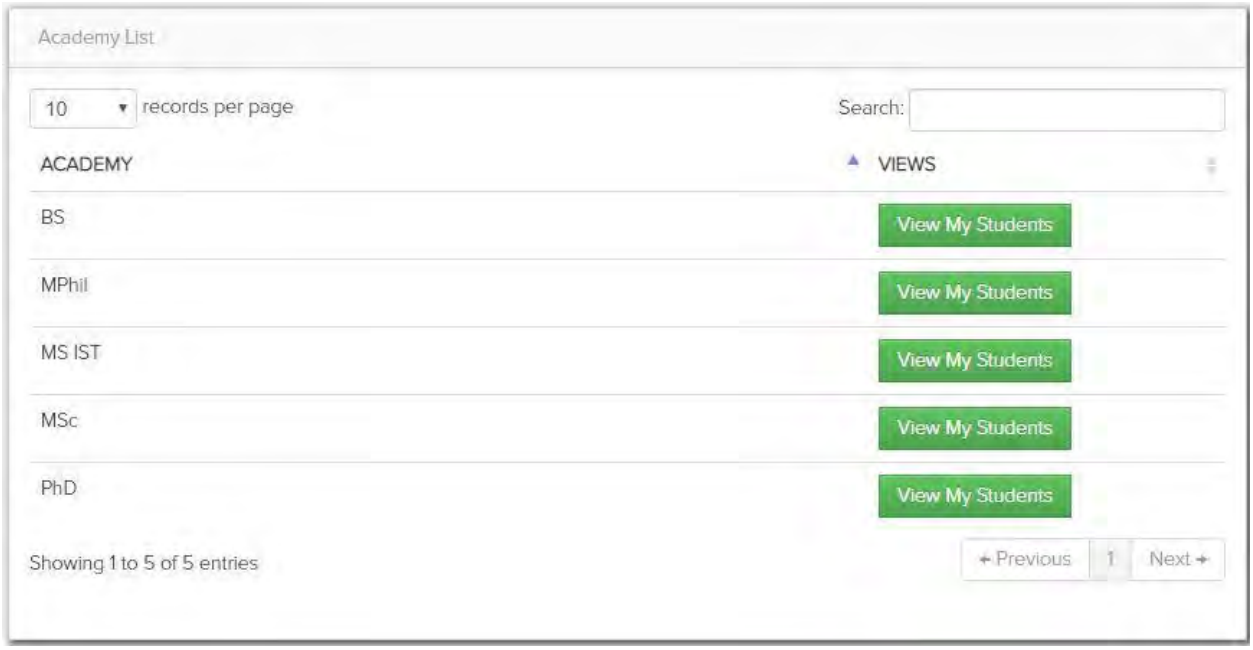


Figure A. 10 View Student List

Add Assignment

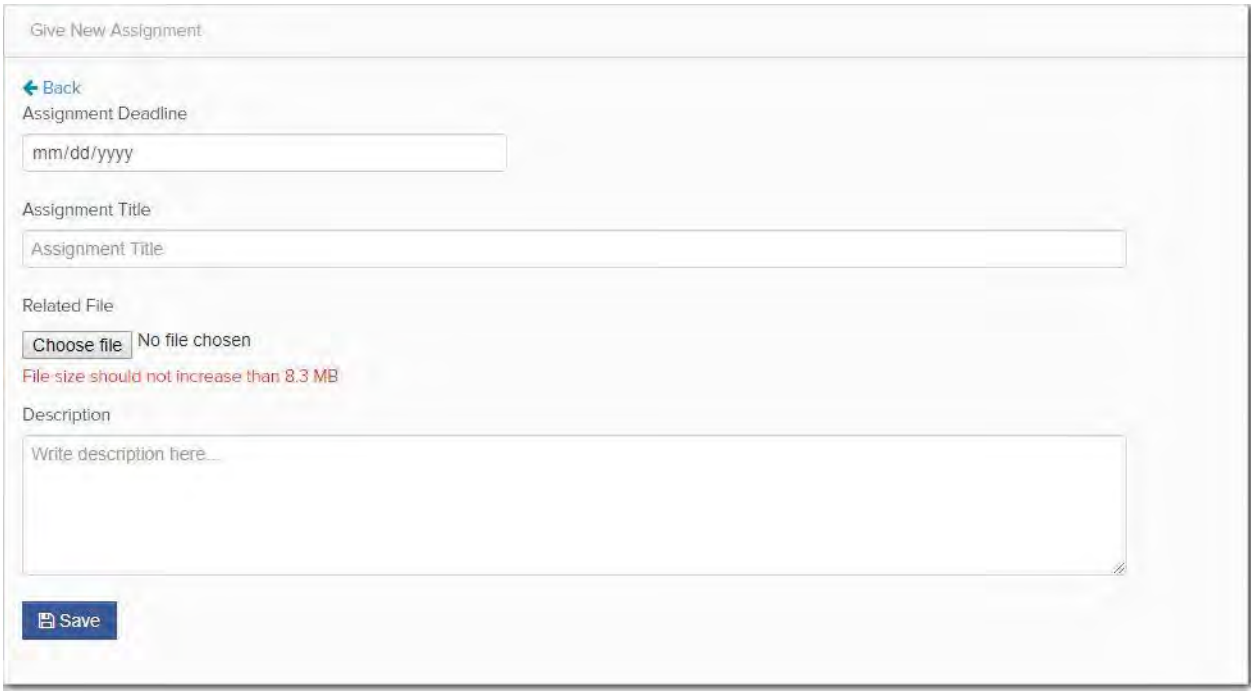


Figure A. 11 Add Assignment

View Assignment results

Figure A. 12 View Assignment results

View Announcements

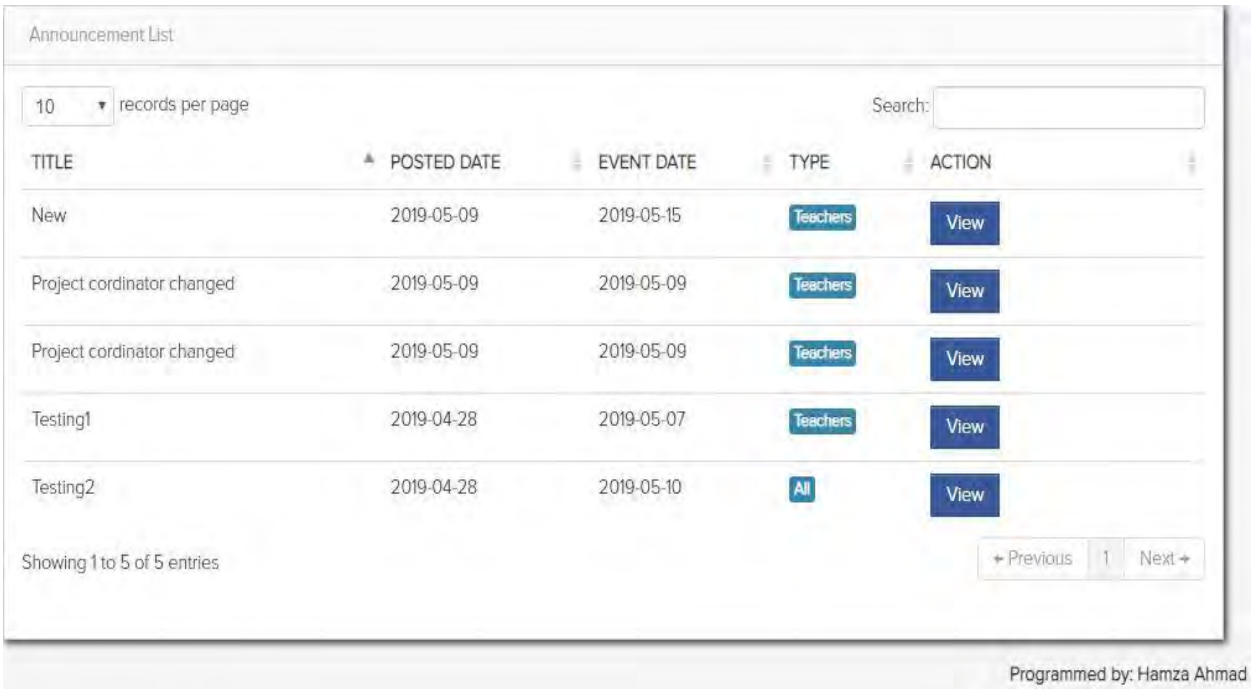


Figure A. 13 View Announcements

View List of Not Submitted Assignment

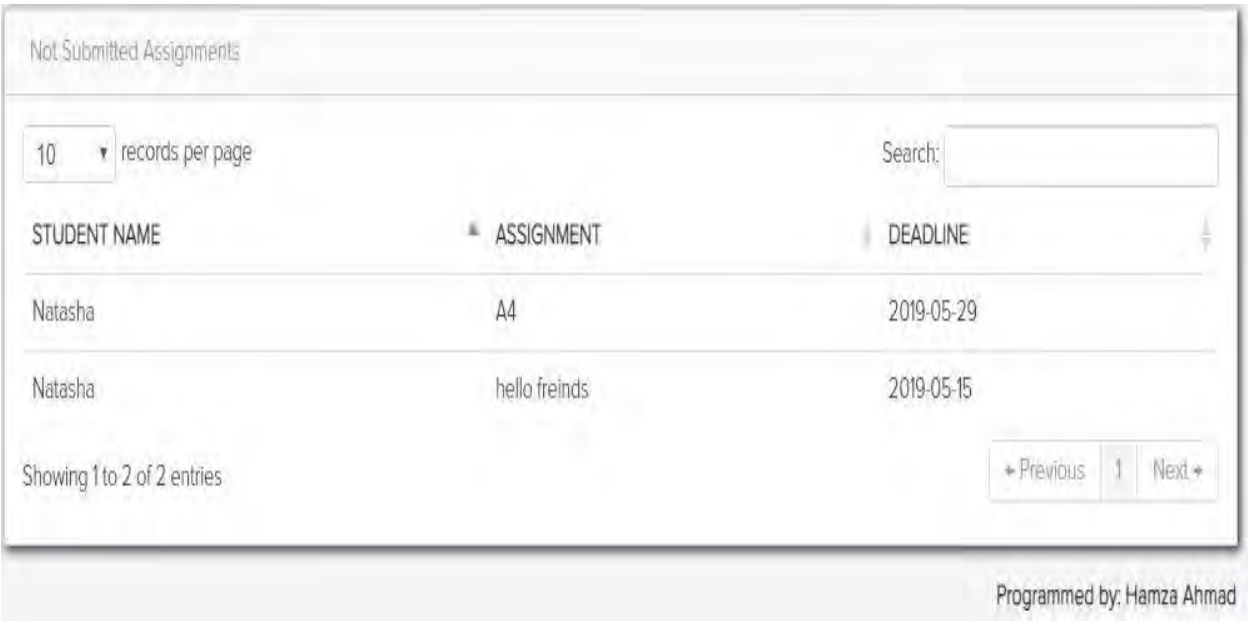


Figure A. 14 View List of not Submitted Assignment

View Assignment (student)

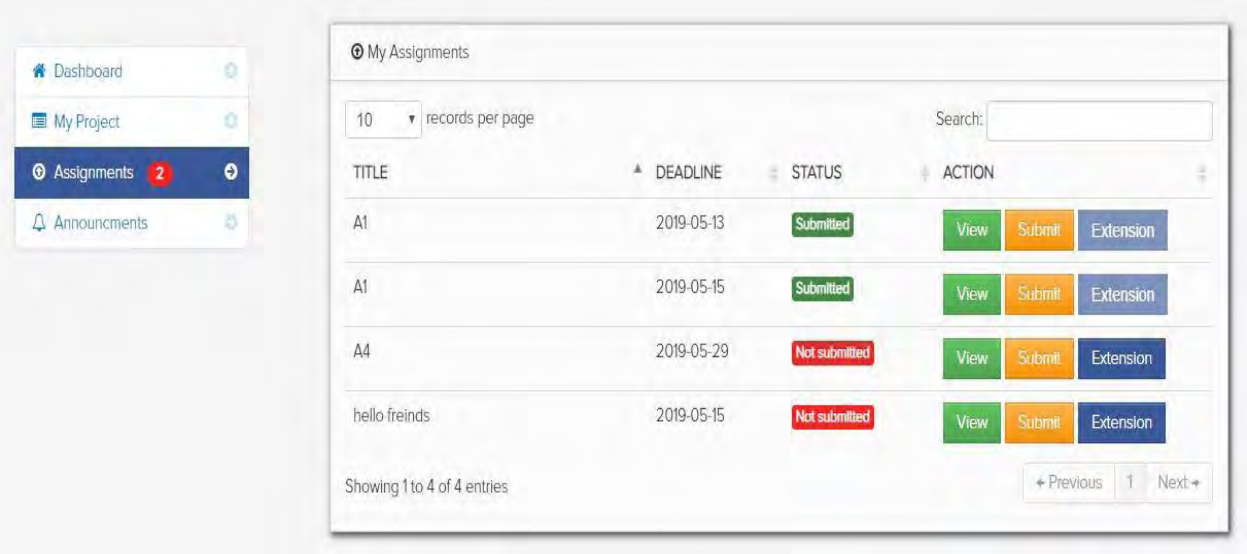


Figure A. 15 View Assignment (student)

Extension Request

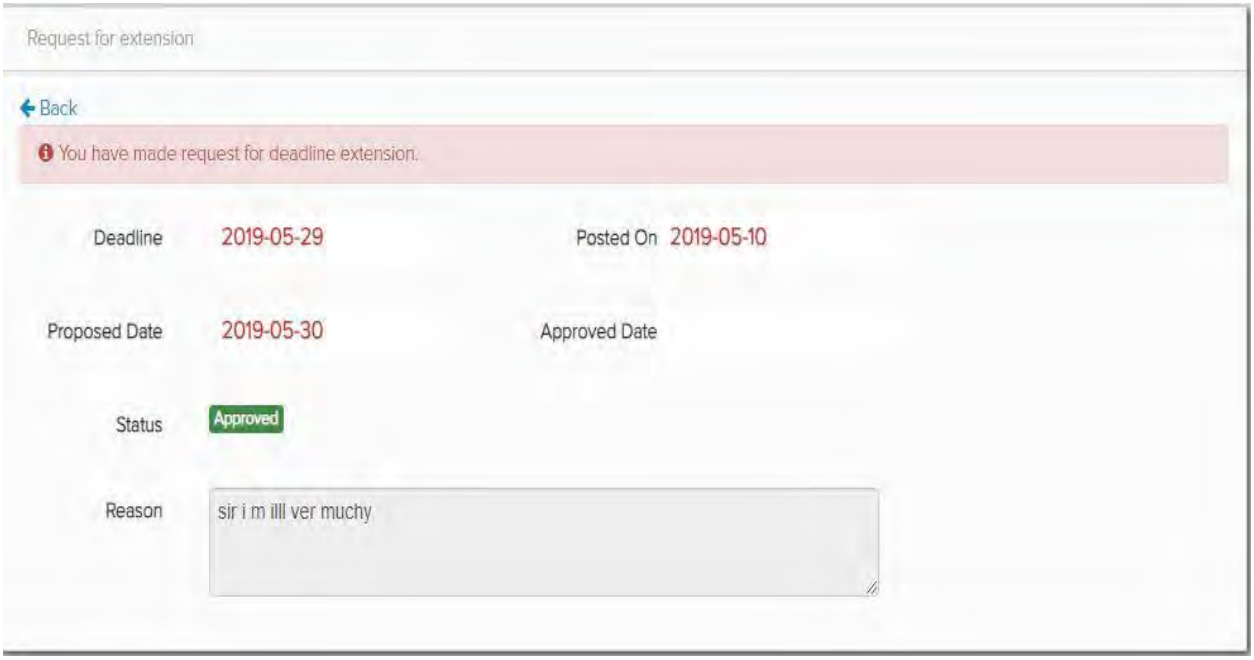


Figure A.16 Extension Request

Reply Assignment

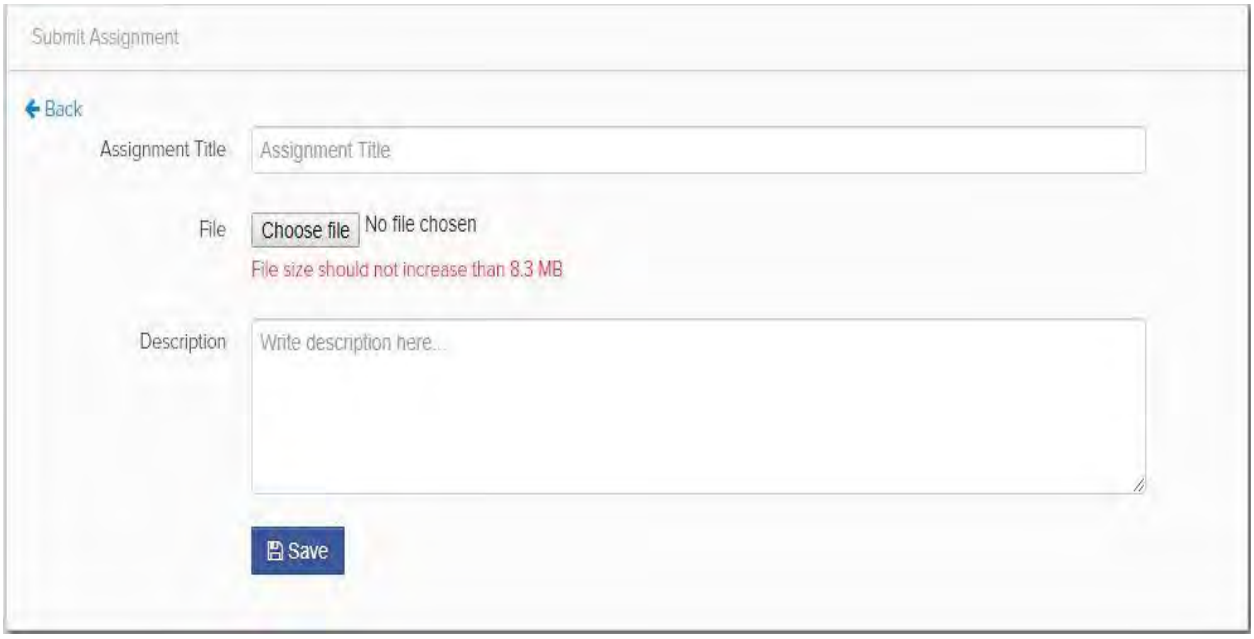


Figure A. 17 Reply Assignment

View Announcement (student)

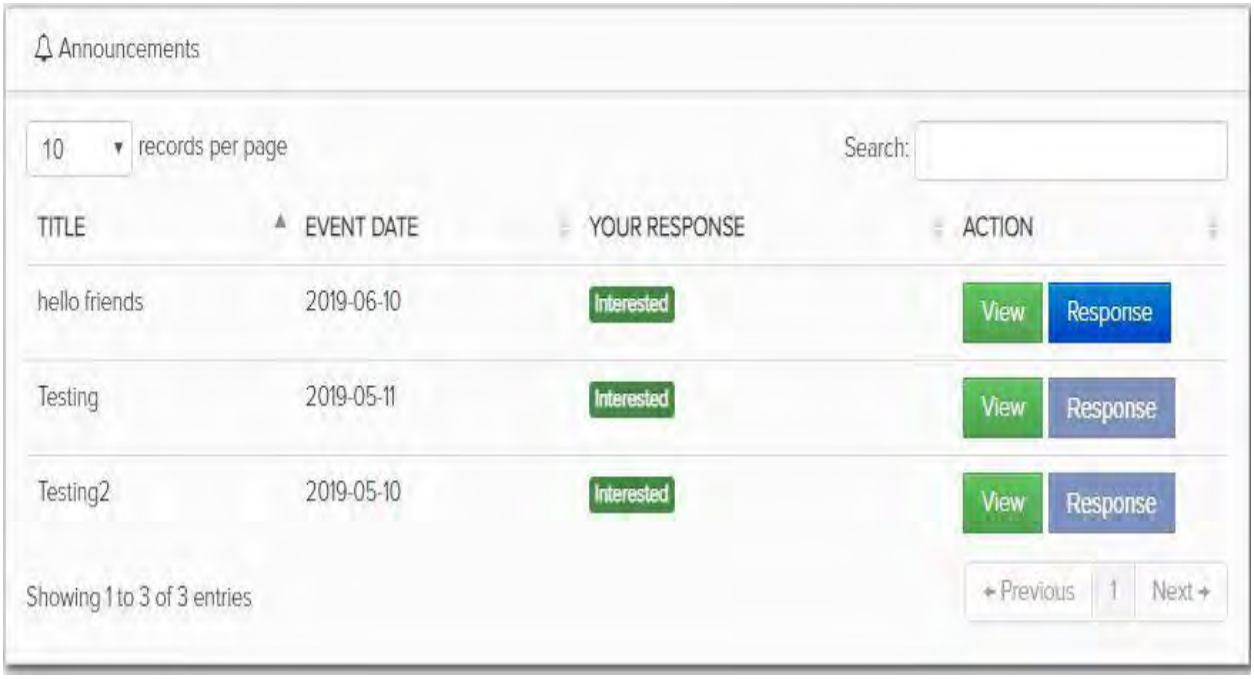


Figure A. 18 View Announcement (student)

Response to Announcement

Figure A. 19 Response to Announcement

Assignment Deadline Alert

Figure A. 20 Assignment Deadline Alert

Assignment email Alert

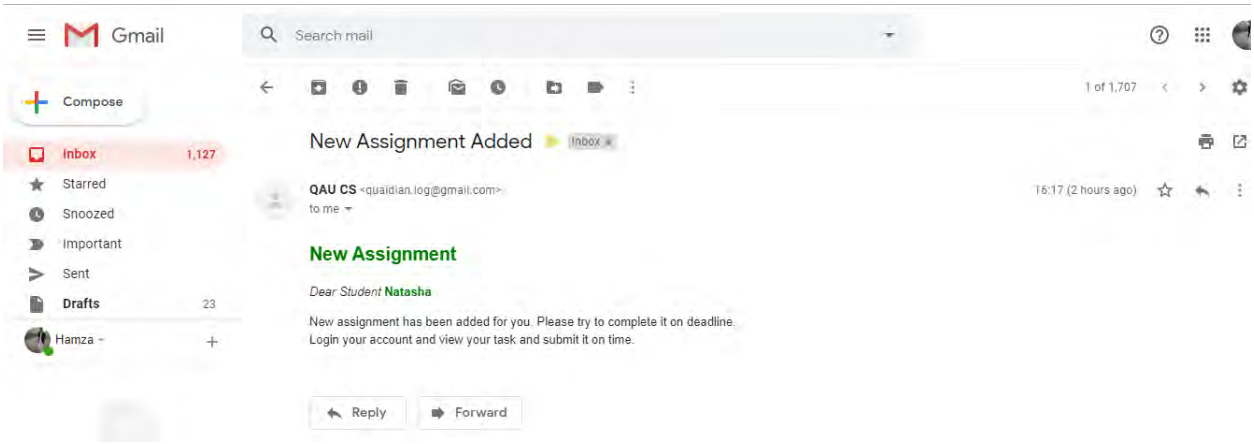


Figure A. 21 Assignment Email Alert

Evaluation Assignment

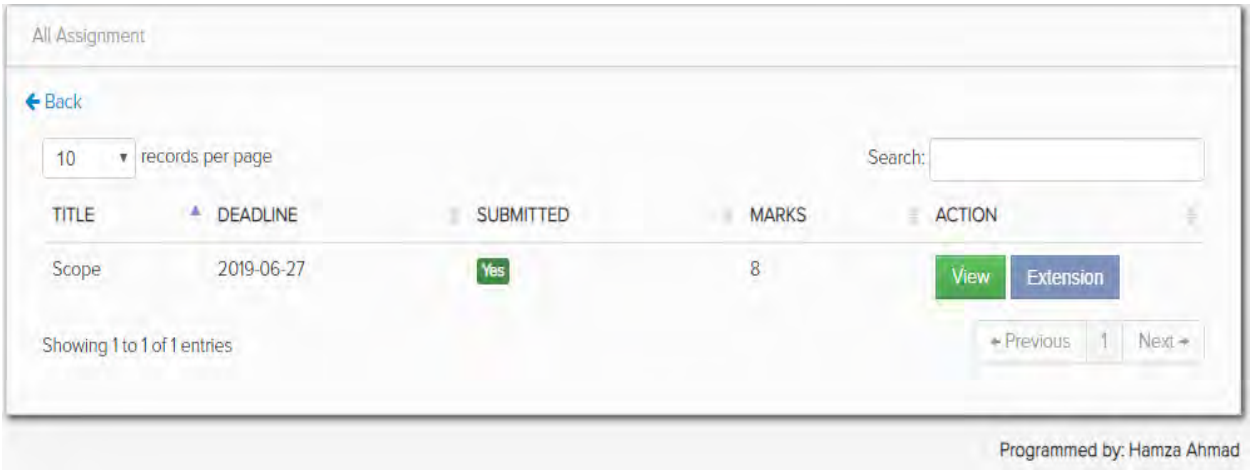


Figure A. 22 Evaluation Interface