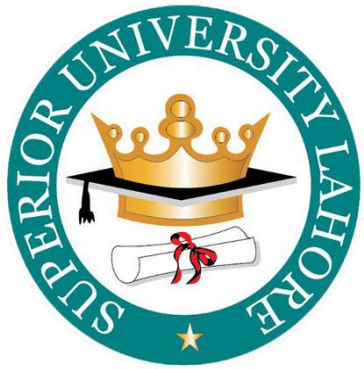
SUPERIOR UNIVERSITY LAHORE



Faculty of Computer Science & IT

Final Year Project

PROJECT REPORT

COLLEGE MANAGEMENT SYSTEM

Project ID: **BITM-F18-FYP-016**

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Abdul Majid	BITM-F15-025	BSIT	0335 – 0142821	majid.qasim1997@gmail.com
Zubda Khanum	BITM-F15-065	BSIT	0303 – 2077604	zubidubi03@gmail.com
Hafiza Shahida Naseem	BITM-F15-043	BSIT	0313 – 8887727	khalidumair573@gmail.com

Mr. Saleem Butt

(Lecturer)

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [] **R**&**D**
Area of specialization	

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BITM-F15-025	Abdul Majid	majid.qasim1997@gmail.com	
(ii)	BITM-F15-065	Zubda Khanum	zubidubi03@gmail.com	
(iii)	BITM-F15-043	Hafiza Shahida Naseem	khalidumair573@gmail.com	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I **Abdul Majid** S/D of **Qasim Ali**, group leader of FYP under registration no

BITM-F18-FYP-016 at Information Technology Department, The Superior College, Lahore. I declare that my FYP proposal is checked by my supervisor and the similarity index is 18% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix D.

Date: 04-09-2019 Name of Group Leader: **Abdul Majid**

Signature: _____

Name of Supervisor: Mr. Saleem Butt

Designation: Lecturer

Signature: _____

HoD:

Signature: _____

Project Report

College Management System

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

We dedicated this project to our parents and our institute who have been a great source of inspiration and support. This project is also dedicated to Mr. Saleem Butt who encourages us to build our motivation.

Acknowledgement

All gestures of recognition are to Almighty Allah, who gave us the quality, learning and bravery to finish this errand. We might want to pay huge amounts of thanks and offer my regards to our Supervisor Mr. Saleem Butt for his valuable direction, advises and support all through the learning procedure that he gave me a helpful course to finish our venture.

Executive Summary

College Management System which can be used by Colleges to manage their daily activities which include the management of Employees, Students, Courses and Parents details, Assignments, Admission Process, Results and Reports, Exams, Events, Attendance, Timetable, and Other Reports. As most of the work is done manually or it is based on paper work such as class attendance, notes, asking for documents etc. These all process takes time. If we include all the work which will be based on online system, then it can save time and reduce work. In case of manual system they need a lot of time, manpower etc. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do within a few minutes.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

College Management System which can be used by Colleges to manage their daily activities which include the management of Employees, Students, Courses and Parents details, Assignments, Admission Process, Results and Reports, Exams, Events, Attendance, Timetable and Other Reports. It provides one-point access to manage these wide ranges of activities both effectively and efficiently. The system will be used by four people, which are Admin, Teacher and Student. Everyone needs to have account on the system and login first to access the system.

1.1. Background

As most of the work is done manually or it is based on paper work such as class attendance, notes, asking for documents etc. These all process takes time. If we include all the work which will be based on online system, then it can save time and reduce work. For example, if any teacher has queried whether all students are present in their class, then he/she have to count manually. So we need a system in which all the details will be produced by a single click on user screen. In this era managing a school, university, college or any educational institution without a perfect software solution in the present times is painful. An appropriate solution is required which can ensure the smooth functioning of the organization as a whole.

1.2. Motivations and Challenges

Time is money for all students and student does not have to face any difficulty if the staff members can track all the problems related to their students, So that they can solve student's problems on time and without wasting time.

1.3. Goals and Objectives

Goals:

To access all activities online within one system instead of all manual work.

- Time Saving
- Efficient Working
- Easily Accessible

Objective:

The objective of College Management System is to handle daily activities on a single system which can access by admin, teacher and student.

1.4. Literature Review/Existing Solutions

In the existing systems, there are many problems like; things are not properly integrated and functioning. There are many management systems running in the school, college and universities. For example: UMS of our university running around but there many problems have to be faced which would be solved in this management system.

- Better layout
- Events Handling
- Proper Time Scheduling

1.5. Gap Analysis

Other systems are available in the market also running in the educational system as well, but there are many things that covered in this system as above mention.

1.6. Proposed Solution

We have decided to develop College Management System, because with College Management System this problem can simply be solved. Teacher, Student are connected with each other on one system. This system is saving the time for student and teacher too.

In this system user can connect any time whenever he/she wants. It is a very efficient solution over a manual college management module and the old fashion college management system. College Management System deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details and other resource related details too. The College management system is an automated version of manual Student Management System. In case of manual system they need a lot of time, manpower etc. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do within a few minutes.

1.7. Project Plan

We have a plan to manage our project in a given time. We have divided complete project work into different phases. First of all we have to gather requirements from the stakeholders, and then we work on those requirements to implement those requirements into the software model. We are working on this model using agile model. After that we will assign the roles to the responsibilities to the group members.

1.7.1. Work Breakdown Structure

1. **Requirement Gathering:**
    - Gather requirements
    - Analyze requirements
    - Separate Functional Requirements and Non-Functional Requirements
2. **Documentation.**
3. **Design.**
4. **Admin Panel:**
    - Login
    - Add student detail
    - Add Teacher detail
    - Add time table
    - Add event detail
    - Admin can see all the courses detail from database

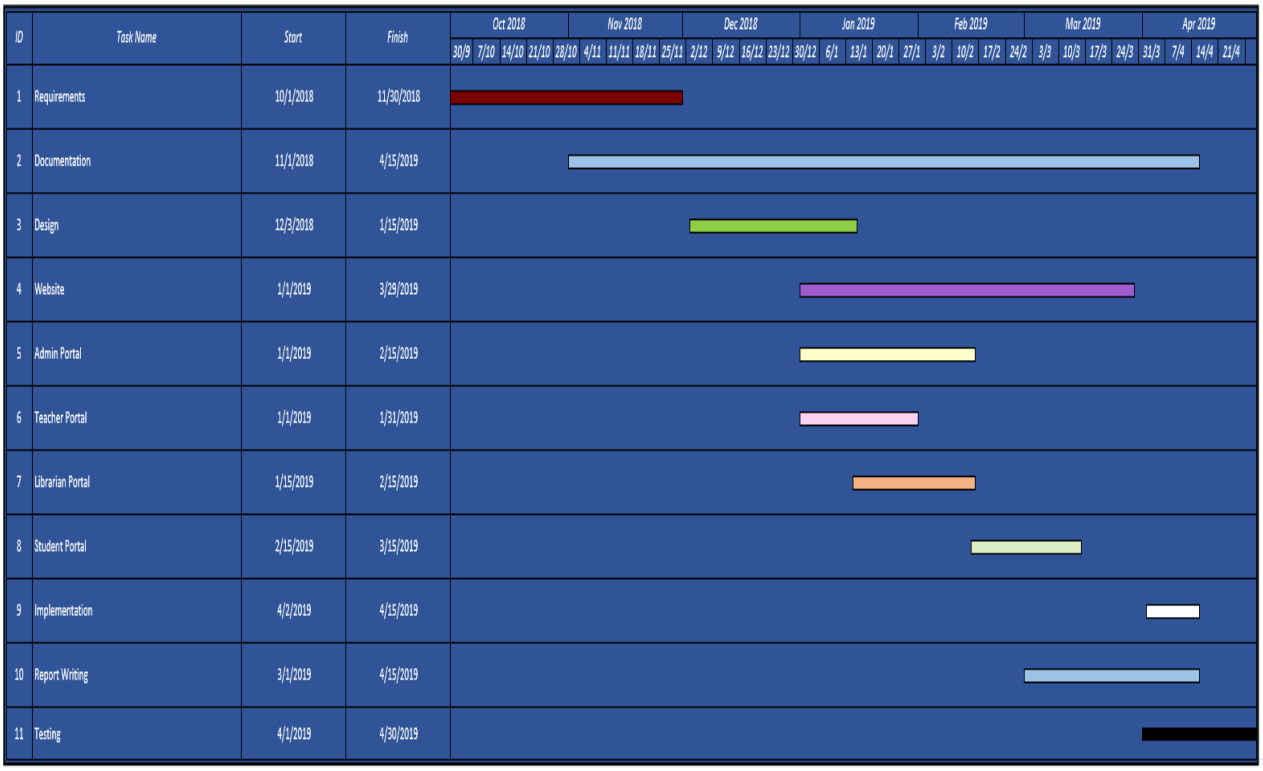
5. **Teacher Panel:**
    - Login
    - Add Assignment
    - Add Attendance
    - Add Result
    - View Event
6. **Student Panel:**
    - Login
    - View Profile
    - View Courses
    - View Time-Table
    - View Assignment
    - View Result
    - View Attendance
    - View Event
7. **Implementation:**
    - Divide whole implementation into modules
    - Divide task to each members
    - Perform meetings for improvement
    - Integrate modules
8. **Testing:**
    - Perform different types of testing's
    - Unit testing
    - Integration testing
    - System testing
    - Performance Testing
    - Usability testing
    - Security testing

1.7.2. Roles & Responsibility Matrix

The purpose of roles & responsibility matrix is to identify who will do what.

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
		1	Requirements	60	Majid/Shahida
		2	Documentation	60	Majid/Zubda
		3	Design	30	Shahida/Zubda
		4	Website	120	Majid/Zubda/Shahida
		5	Admin Portal	30	Majid
		6	Teacher Portal	30	Zubda
		8	Student Portal	30	Majid
		9	Testing	40	Majid/Zubda/Shahida

1.7.3. Gantt Chart



ID	Task Name	Start	Finish
1	Requirements	10/1/2018	11/30/2018
2	Documentation	11/1/2018	4/15/2019
3	Design	12/3/2018	1/15/2019
4	Website	1/1/2019	3/29/2019
5	Admin Portal	1/1/2019	2/15/2019
6	Teacher Portal	1/1/2019	1/31/2019
7	Librarian Portal	1/15/2019	2/15/2019
8	Student Portal	2/15/2019	3/15/2019
9	Implementation	4/2/2019	4/15/2019
10	Report Writing	3/1/2019	4/15/2019
11	Testing	4/1/2019	4/30/2019

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

The purpose of this site is to help client in the management of the college. Client can store all the records of college on this site easily. This will overcome the work burden and also it's a good way to consume less effort and time and perform more work efficiently.

2.1.2. Document Conventions (SEARCH)

Stander rule of documentation has been followed in the order of standardized the work. In order to create efficient CMS Website in the premises of online management system.

2.1.3. Intended Audience and Reading Suggestions

This project is a prototype for the college management system and it is restricted within the college premises. This has been implemented under the guidance of college staff. This project is useful for the college teachers and as well as to the students.

2.1.4. Product Scope

We are working on the college management system which compromises with the four major panels followed by sub modules.

- Admin
- Teacher
- Student

It provides one-point access to manage these wide ranges of activities both effectively and efficiently.

1. **Admin:**
    a. **Login:** Admin can login his personal account.
    b. **Add/View Student:** Admin add student detail into a system.

c. **Add/View Teacher:** Admin add Teacher detail into a system**.**

d. **Add Time-Table:** Admin set and add time table.

e. **Add Event:** Admin add event detail into a system.

f. **View Result**

g. **Add/View Courses:** Admin can see all the courses detail from database.

2. **Teacher:**

a. **Login:** Teacher can login in his personal account.

b. **Add Assignment:** Teacher can add assignment detail for the students.

c. **Add Attendance:** Teacher can add attendance detail into a database.

d. **Add Result:** Teacher can add result detail.

e. **View timetable:** Teacher can view timetable.

f. **View Event:** Teacher can view event detail from database.

3. **Student:**

a. **Login:** Student can login his personal account.

b. **View Profile:** Student can see his profile.

c. **View Courses:** Student can view course added by Admin.

d. **View Time-Table:** Student can view all timetable detail.

e. **View Assignment**: Student can view all assignment detail added by teacher.

f. **View Result:** Student can view result.

g. **View Attendance:** Student can view attendance.

h. **View Event:** Student can view all event detail.

2.1.5. References

1. https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database.

2. https://www.studymode.com/essays/Education-Management-System-Software-Requirements-Specification-1514575.html

2.2. Overall Description

2.2.1. Product Perspective

The CMS is a system that will help in maintaining student and staff information easily and it will maintain student records, important event, attendance, schedules and many other important things related to the college management system.

This website is aimed recording a considerable number of student records and need assistance for managing their data. This website is user-friendly, easy to learn and reliable.

Product Functions

Login: User can login his personal account.

Add/View Student, teacher: Admin add user detail into a system.

Add Time-Table: Admin set and add time table.

Add Event: Admin add event detail into a system.

View Courses: User can see all the courses detail from database.

Add Assignment: Teacher can add assignment detail for the students.

Add Attendance: Teacher can add attendance detail into a database.

Add Result: Teacher can add result detail.

View Event: User can view event detail from database.

Add Courses: Admin can add courses into database for student use.

Register a Course: View detail of all Register courses.

View Time-Table: User can view all timetable detail.

View Assignment: User can view all assignment detail added by teacher.

View Result: User can view result.

View Attendance: User can view attendance.

2.2.2. User Classes and Characteristics

There are three types of users who will be using this website.

Admin:

Admin can add and remove users. He can also add and update important event and schedule. Admin can add and remove courses. He can also update the details of registered course.

Teacher:

Teacher can add attendance of students, assignments and result. He can also view event and schedule.

Student:

Student can view his attendance, schedule, assignments, timetable, result and courses.

2.2.3. Operating Environment

- Window 8 or higher

Platform

- XAMMP Server
- Notepad++
- Bootstrap
- HTML
- CSS
- PHP
- MySQL
- PhpMyAdmin
- JavaScript

For User

- Internet Browser

- Internet Connection

2.2.4. Design and Implementation Constraints

- User detail saves into the database.
- MS SQL Server will be utilized as SQL motor and database.
- The College Management System is running 24 hours per day.
- Users may access from any PC that has Internet perusing abilities and an Internet association.
- Users must have their right username and passwords to go into their online records and do activities.

2.2.5. User Documentation

User must know the details of our product that how it works and how to efficiently use it. User document will be given the user so that it will help user to better understand our product. User need internet to use our product and perform his tasks.

Assumptions and Dependencies

- The system depends on the admin provided rights.
- SQL Server to store the database.
- PHP to build up the System

2.3. External Interface Requirements

2.3.1. User Interfaces

There will be a login page for the user where user will enter his/her username and password. Then system will redirect the user to his/her dashboard according to the user kind (admin, teacher and student).

2.3.2. Hardware Interfaces

- Equipment prerequisites for Insurance on web will be same for both the gatherings which are takes after:
- Processor

- Ram: 128 MB or above.
- HD: 20 GB or above.
- Just the prescribed setup (fundamental necessities of a PC framework) no other particular equipment is required to run the system.

2.3.3. Software Interfaces

All the databases for the college management system will be configured using access 2018 – 2027. These databases include user information. These can be modified by the end user. The CMS database will include the user information. The user or activities information database will contain all the detail of the user or detail of all activities such as is, username and password.

2.3.4. Communications Interfaces

The two gatherings ought to be associated through either by LAN or WAN for the correspondence. Sender Communication channels Receiver.

2.4. System Features

- **Register:** Admin can add users.
- **Log in:** User will login to use the system.
- **Logout:** User can logout.
- **View event:** User can view event.
- **View courses:** User can view added courses.
- **View profile:** User can view his/her profile.
- **View attendance:** Student can view his/her attendance.
- **View timetable:** User can view timetable.
- **View assignment:** Student can see assignment.

- **View result:** Student can view result.
- **Add attendance:** Teacher can mark attendance.
- **Add result:** Teacher can add result.
- **Add course:** Admin can add course.
- **Update registered course details:** Admin can Register course and update details.

System Feature 1:

Function requirement 1

Register

" User can register into the system **"**

Identifiers	FR-001
Title	Allows register access online
Requirement	Email, User name and password
Rational	Register to the system and provide security to the data.
Restriction and risk	Correct data must be entered during registration, otherwise he will not get registered.
Dependency	Pc, connection with server
Priority	Time, security

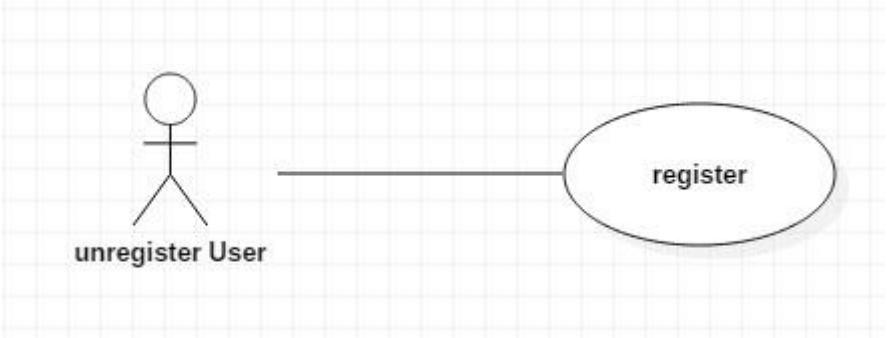
Table 2a- Function requirement Register

Use case 1

Register
Actor
• UnReg-User
Preconditions
• User must open the website.
• User must open the registration page.
Basic flow
• User want to create his account for this website.
• User enter correct data and get registered.
Alternate flow
• User cannot use this website as an unregistered user.
Post conditions
• User must be successfully registered to this website.

Table 2a- Use Case Register

Diagram 1



System features 2:

Function requirement 2

Login Function

" User can login into the system **"**

Identifiers	FR-002
Title	Allows user access online
Requirement	User name, password
Rational	Login to the system and provide security to the data.
Restriction and risk	User name and password must be correct for the login.
Dependency	Pc, connection with server.
Priority	Time, security

Table 2b- Function requirement Login Function

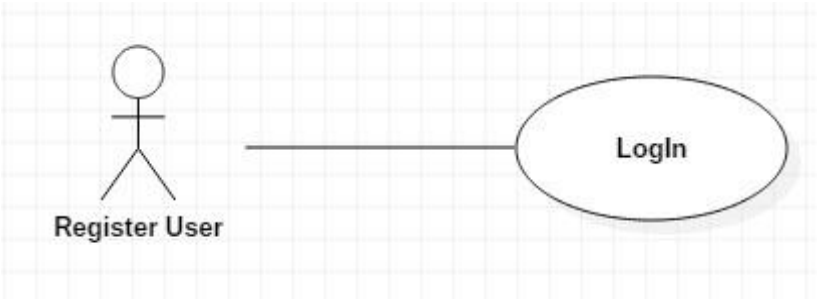
Use case 2

Login
Actor
• Reg-Users
Preconditions
• User must open the website.
• User must have an account for this website.
Basic flow

• User enter the correct required data and gets log in.
Alternate flow
• User cannot use website without a registered account.
Post conditions
• User must be successfully login.

Table 2b- Use Case Login

Diagram 2



System feature 3:

Function requirement 3

Log out Function

‘’ User can log out into the system **‘’**

Identifiers	FR-003
Title	Allows log out access online.
Requirement	User login then logout.
Rational	Log out to the system and provide security to the data.
Restriction and risk	User must login
Dependency	Pc, connection to the server.
Priority	Time, security

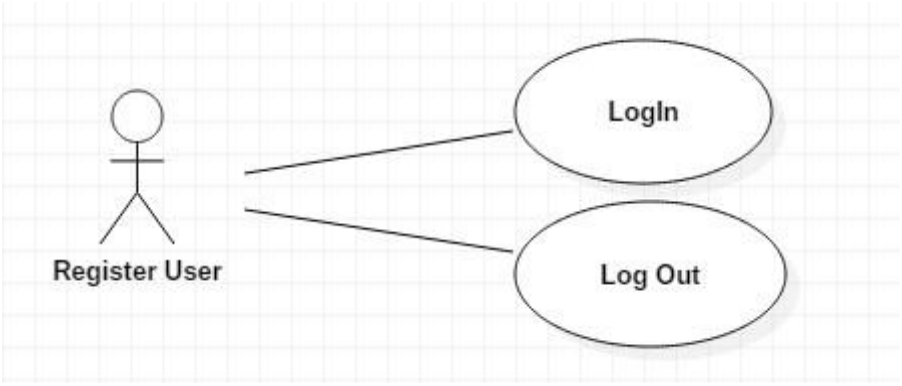
Table 2c- Function requirement log out function

Use case 3

Logout
Actor
• Reg-User
Preconditions
• User must be successfully logged in.
Basic flow
• User wants to log out
Alternate flow
• User don’t want to view the results.
Post conditions
• User must successfully log out.

Table 2c- Use Case Register

Diagram 3



System feature 4:

Function requirement 4

View Event

" User can view the event **"**

Identifiers	FR-004
Title	Allow user to view event.
Requirement	User login to the system and view the events
Rational	Login the system and view event.
Restriction and risk	User must login to system, otherwise he cannot view events.
Dependency	Pc, connection to the server, registered account.
Priority	Time, security.

Table 2d- Function requirement View Event

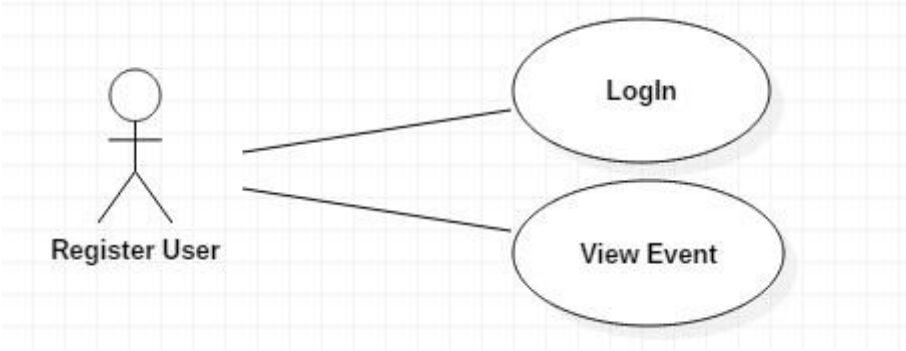
Use case 4

View Event
Actor
• Reg-User
Preconditions
• User must log in to the website.
Basic flow

• User log in to the system and view the events.
Alternate flow
• User don't want to see the events.
Post conditions
• User must successfully view the events.

Table 2d- Use Case View Event

Diagram 4



System feature 5:

Function requirement 5

View Courses

" User can view courses**"**

Identifiers	FR-005
Title	Allows user to view courses
Requirement	User must login to the website.
Rational	Login to the system and view courses.
Restriction and risk	User must login to view courses.
Dependency	Pc, connection to the server, registered account.

Priority	Time, security.

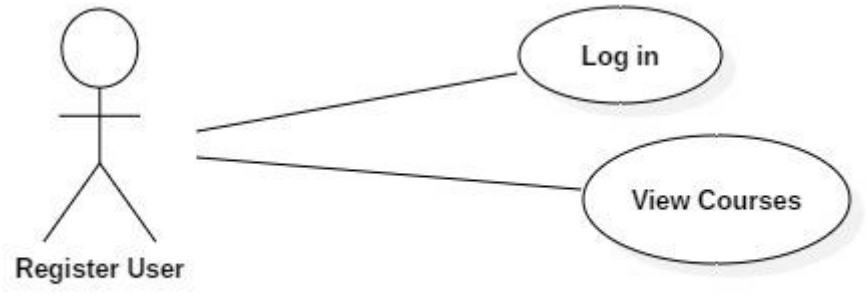
Table 2e- Function requirement View Courses

Use case 5

View Courses
Actor
• Reg-User
Preconditions
• User must login to the website.
Basic flow
• User enter correct data to login and then view courses.
Alternate flow
• User don't want to view courses.
Post conditions
• User must successfully view courses.

Table 2e- Use Case View Courses

Diagram 5



System features 6:

Function requirement 6

View profile

" User can view his profile. **"**

Identifiers	FR-006
Title	Access to his own profile
Requirement	Must login to the website.
Rational	Log in to the website and then view profile.
Restriction and risk	Must log in to view profile, otherwise he cannot access to the profile.
Dependency	Pc, connection to the server, registered account.
Priority	Time, security

Table 2f- Function Requirement View Profile

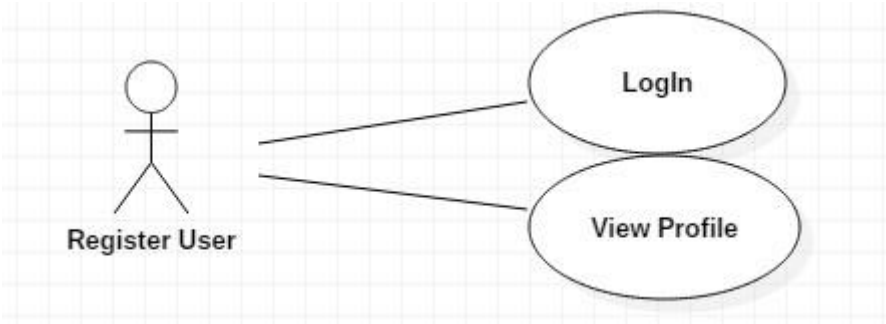
Use case 6

View Profile
Actor
• Reg-User
Preconditions
• User must log in to the website.

Basic flow
• User login to the website and then view his profile.
Alternate flow
• User does not want to view profile.
Post conditions
• User must successfully view profile.

Table 2f- Use Case View Profile

Diagram 6



System features 7:

Function requirement 7

View Attendance

'' User can view his attendance **''**

Identifiers	FR-007
Title	Access to the attendance
Requirement	User must Login
Rational	Login to the website and view attendance.
Restriction and risk	User must login to view attendance, otherwise he cannot view attendance.
Dependency	Pc, registered account, connection to the server.
Priority	None

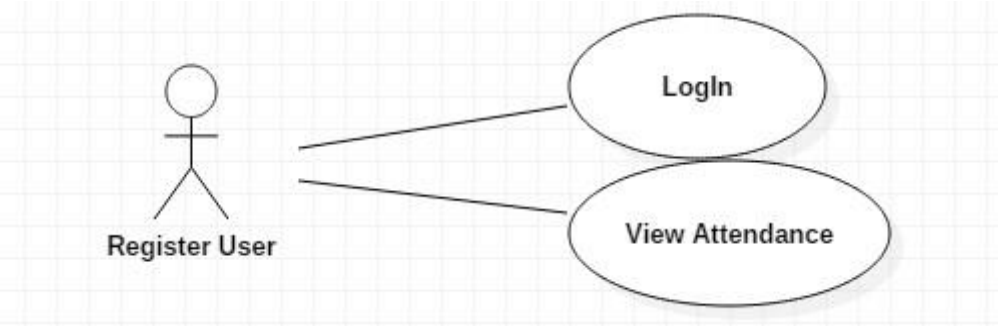
Table 2g- Function Requirement View Attendance

Use case 7

View Attendance
Actor
• Reg-User
Preconditions
• User must login to the website
Basic flow
• User login to the website and then
Alternate flow
• User does not want to view attendance
Post conditions
• User successfully view attendance

Table 2g- Use Case View Attendance

Diagram 7



System feature 8:

Function requirement 8

View timetable

" User can view timetable. **"**

Identifiers	FR-008
Title	Access to view timetable
Requirement	Must login to the website.

Rational	Log in to the website and then view timetable.
Restriction and risk	Must log in to view timetable, otherwise he cannot view timetable.
Dependency	Pc, connection to the server, registered account.
Priority	Time

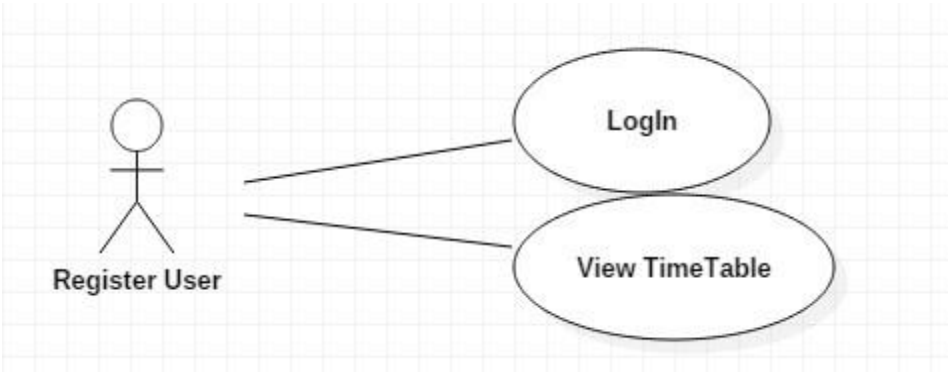
Table 2h- Function Requirement View Timetable

Use case 8

View Timetable
Actor
• Reg-User
Preconditions
• User must log in to the website.
Basic flow
• User login to the website and then view timetable
Alternate flow
• User does not want to view timetable.
Post conditions
• User must successfully view timetable.

Table 2h- Use Case View Timetable

Diagram 8



System feature 9:

Function requirement 9

View result

‘’ students can view result. **‘’**

Identifiers	FR-009
Title	Access to his own profile
Requirement	Must login to the website.
Rational	Log in to the website and then view profile.
Restriction and risk	Must log in to view profile, otherwise he cannot access to the profile.
Dependency	Pc, connection to the server, registered account.
Priority	Time, security

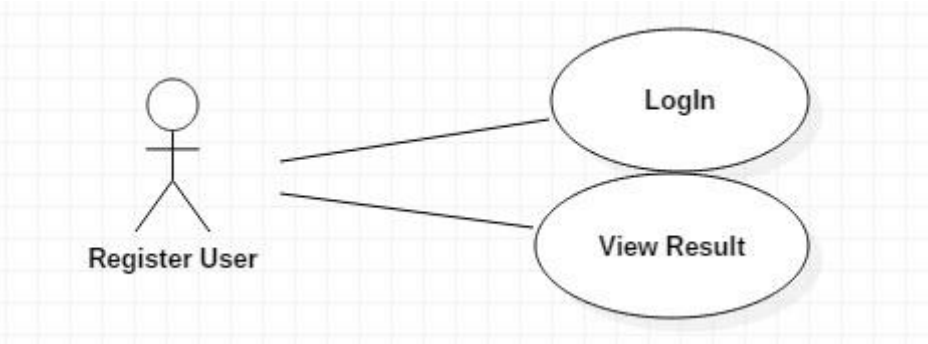
Table 2i- Function Requirement View Result

Use case 9

View result
Actor
• Reg-Student
Preconditions
• User must log in to the website.
Basic flow
• User login to the website and then views his Result.
Alternate flow
• User does not want to view result.
Post conditions
• User must successfully view result.

Table 2i- Use Case View Result

Diagram 9



System feature 10:

Function requirement 10

View Assignment

" students can view assignment. **"**

Identifiers	FR-0010
Title	Access to view assignment
Requirement	Must login to the website.
Rational	Log in to the website and then view assignment.
Restriction and risk	Must log in to view profile, otherwise he cannot view assignment.
Dependency	Pc, connection to the server, registered account.
Priority	Time

Table 2j- Function Requirement View Assignment

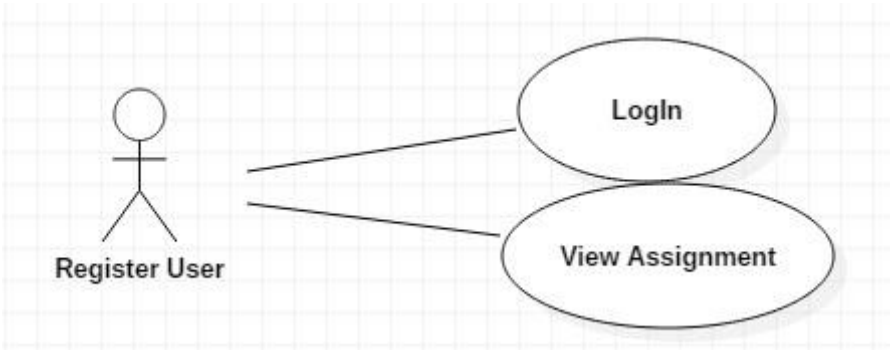
Use case 10

View Assignment
Actor
• Reg-Student
Preconditions

• User must log in to the website.
Basic flow
• User login to the website and then views his assignment.
Alternate flow
• User does not want to view assignment.
Post conditions
• User must successfully view assignment.

Table 2j- Use Case View Assignment

Diagram 10



System feature 11:

Function requirement 11

Add Result

‘’ Teacher can add result of the student. **‘’**

Identifiers	FR-0011
Title	Access to add result.
Requirement	Must login to the website.
Rational	Log in to the website and then add result.
Restriction and risk	Must log in to add result, otherwise he cannot access to add result.
Dependency	Pc, connection to the server, registered account.
Priority	Time, security

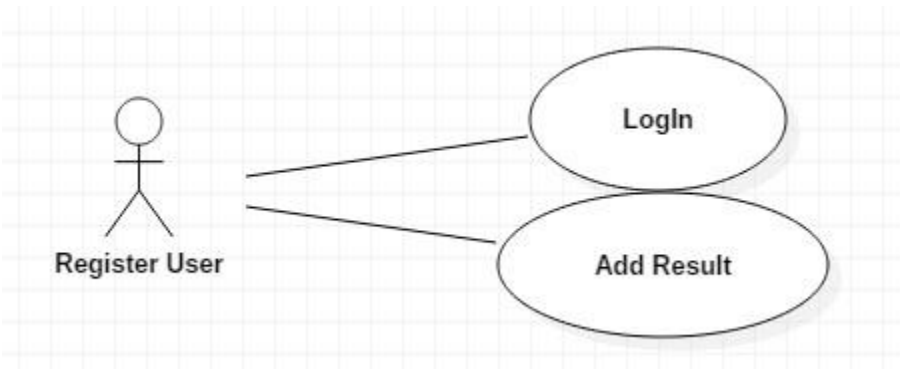
Table 2k- Function Requirement View Result

Use case 11

View Result
Actor
• Reg-Teacher
Preconditions
• User must log in to the website.
Basic flow
• User login to the website and then add result.
Alternate flow
• User does not want to add result.
Post conditions
• User must successfully add result.

Table 2k- Use Case Add Result

Diagram 11



System feature 12:

Function requirement 12

Add Attendance

'' Teacher can add attendance of the student. **''**

Identifiers	FR-0012
Title	Access to add attendance.
Requirement	Must login to the website.

Rational	Log in to the website and then add attendance.
Restriction and risk	Must log in to add attendance, otherwise he cannot access to add attendance.
Dependency	Pc, connection to the server, registered account.
Priority	Attendance record

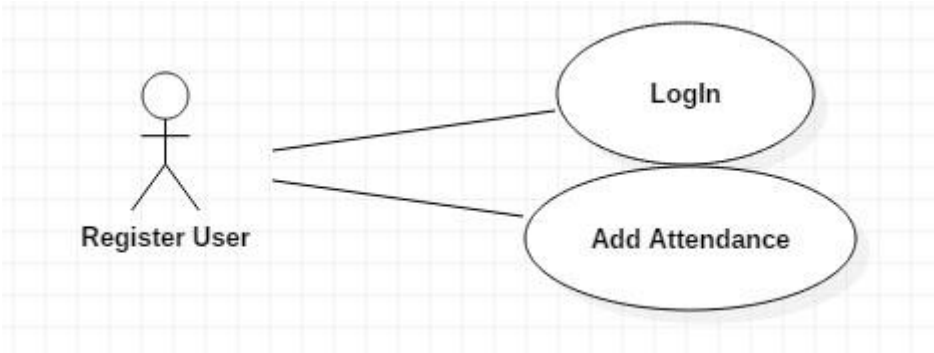
Table 2l- Function Requirement Add Attendance

Use case 12

Add Attendance
Actor
• Reg-Teacher
Preconditions
• User must log in to the website.
Basic flow
• User login to the website and then add attendance.
Alternate flow
• User does not want to add attendance.
Post conditions
• User must successfully add attendance.

Table 2l- Use Case Add Attendance

Diagram 12



System feature 13:

Function requirement 13

Add Course

" Admin can add course. **"**

Identifiers	FR-0013
Title	Access to add course.
Requirement	Must login to the website.
Rational	Log in to the website and then add course.
Restriction and risk	Must log in to add course, otherwise he cannot add course.
Dependency	Pc, connection to the server, registered account.
Priority	Satisfy user need

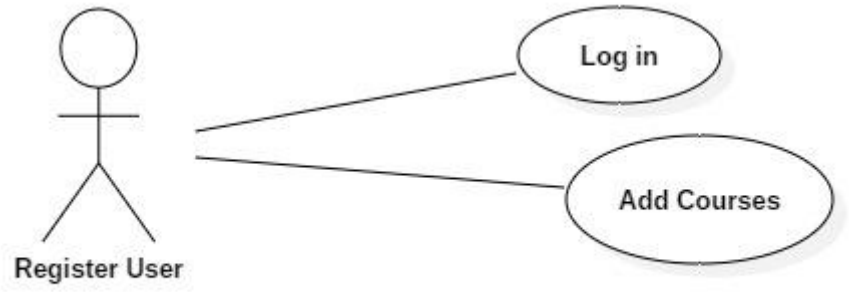
Table 2m- Function Requirement Add Course

Use case 13

Add Course
Actor
• Admin
Preconditions
• User must log in to the website.
Basic flow
• User login to the website and then add course.
Alternate flow
• User does not want to add course.
Post conditions
• User must successfully add course.

Table 2m- Use Case Add Course

Diagram 13



2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

With a specific end goal to keep up a worthy speed at most extreme number of activities permitted from a specific user will be any number of users and can get to the framework whenever. Likewise associations with the servers will be founded on the criteria of properties of the user like his area, and server will work 24X 7 times.

2.5.2. Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure. All tasks, logged data, refreshes, user activities are reinforcement.

2.5.3. Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that must choose their database partner carefully. Any adjustment (insert, update and delete) for the Database might be synchronized and done just by the System Admin.

Software Quality Attributes

- Reliability
- Availability
- Maintainability
- Portability

2.5.4. Business Rules

There are mostly four kinds of users utilizing the system:

- Admin
- Teacher
- Student

Admin has the full authorization of controlling the system.

Teacher has authorization to add attendance, result, and assignments and view events.

Student has authorization to view, attendance, courses, results, assignments and events.

Chapter 3

Use Case Analysis

Chapter 3: System Analysis

3.1. **Use Case Model**

Admin Side:

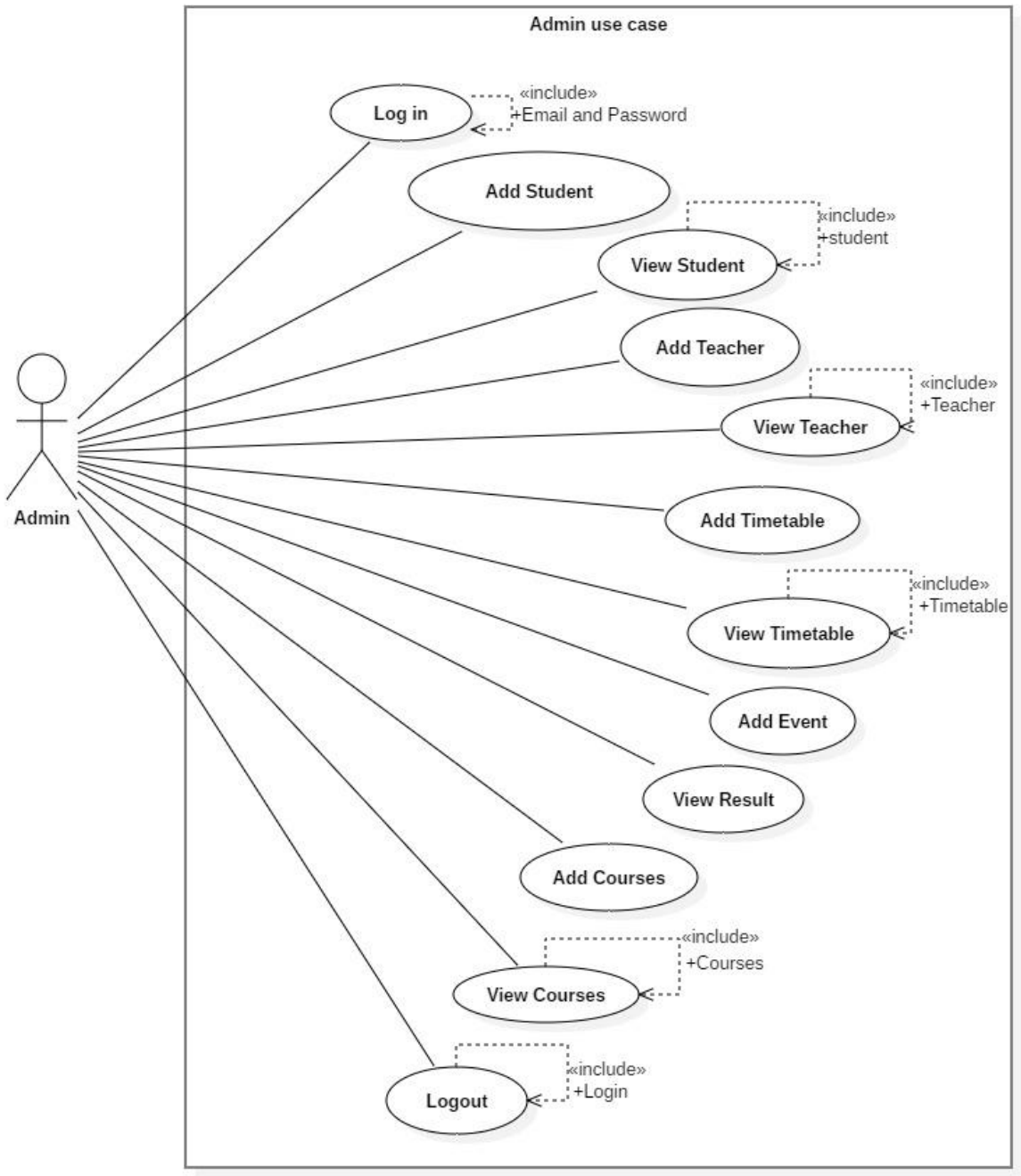


Figure 3.1 USE CASE Diagram

Student side:

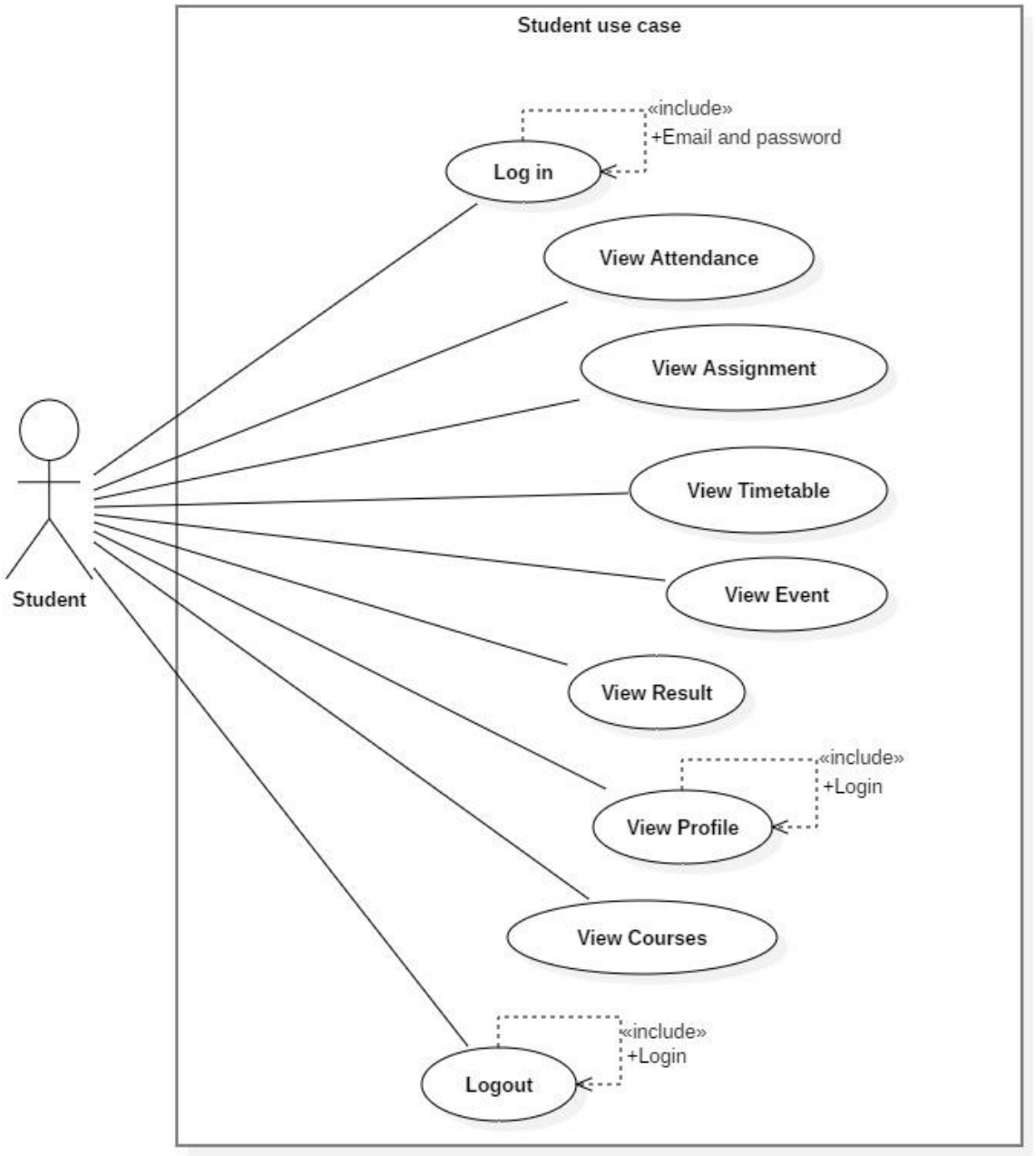


Figure 3.2 USE CASE Diagram

Teacher Side:

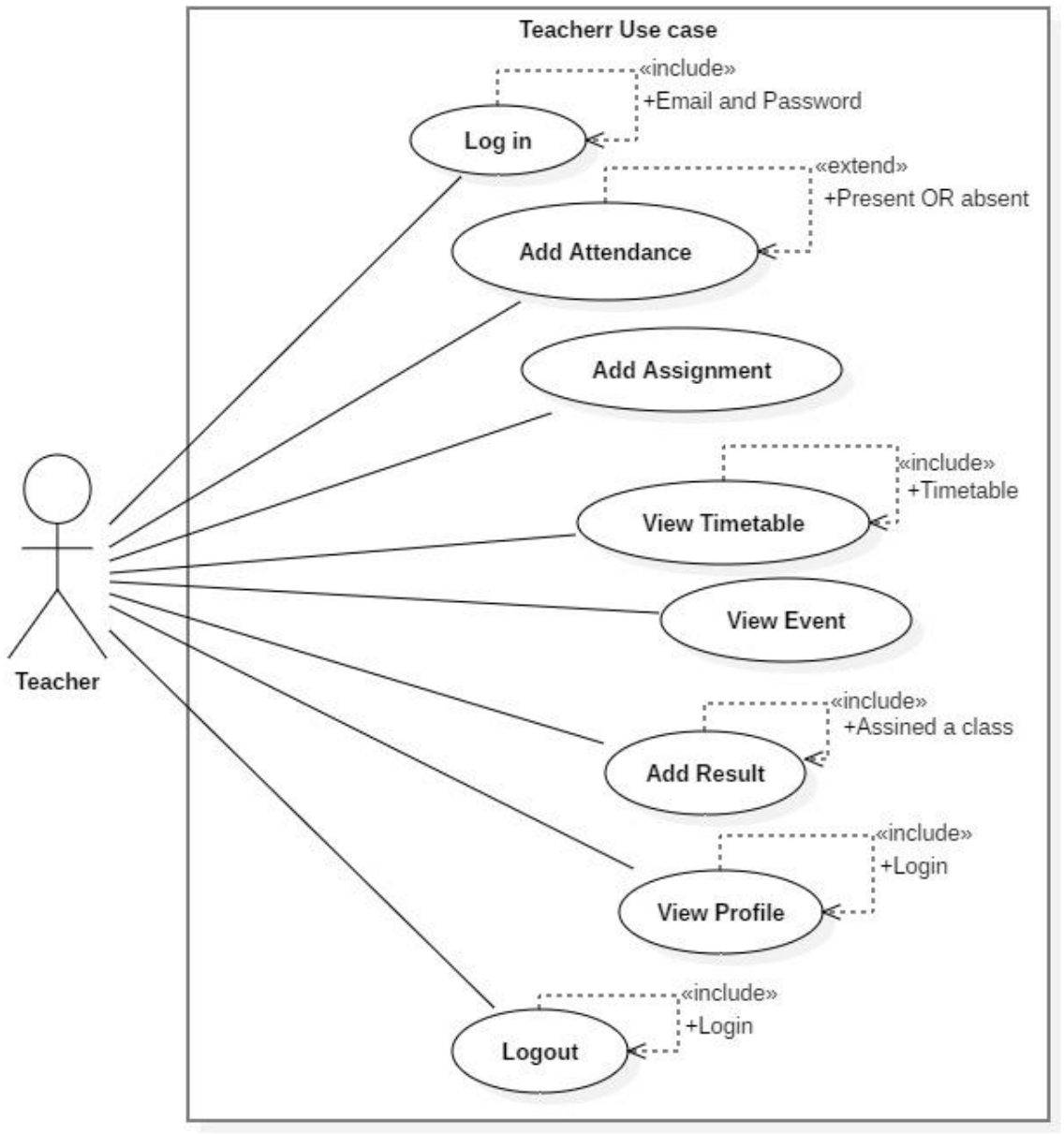
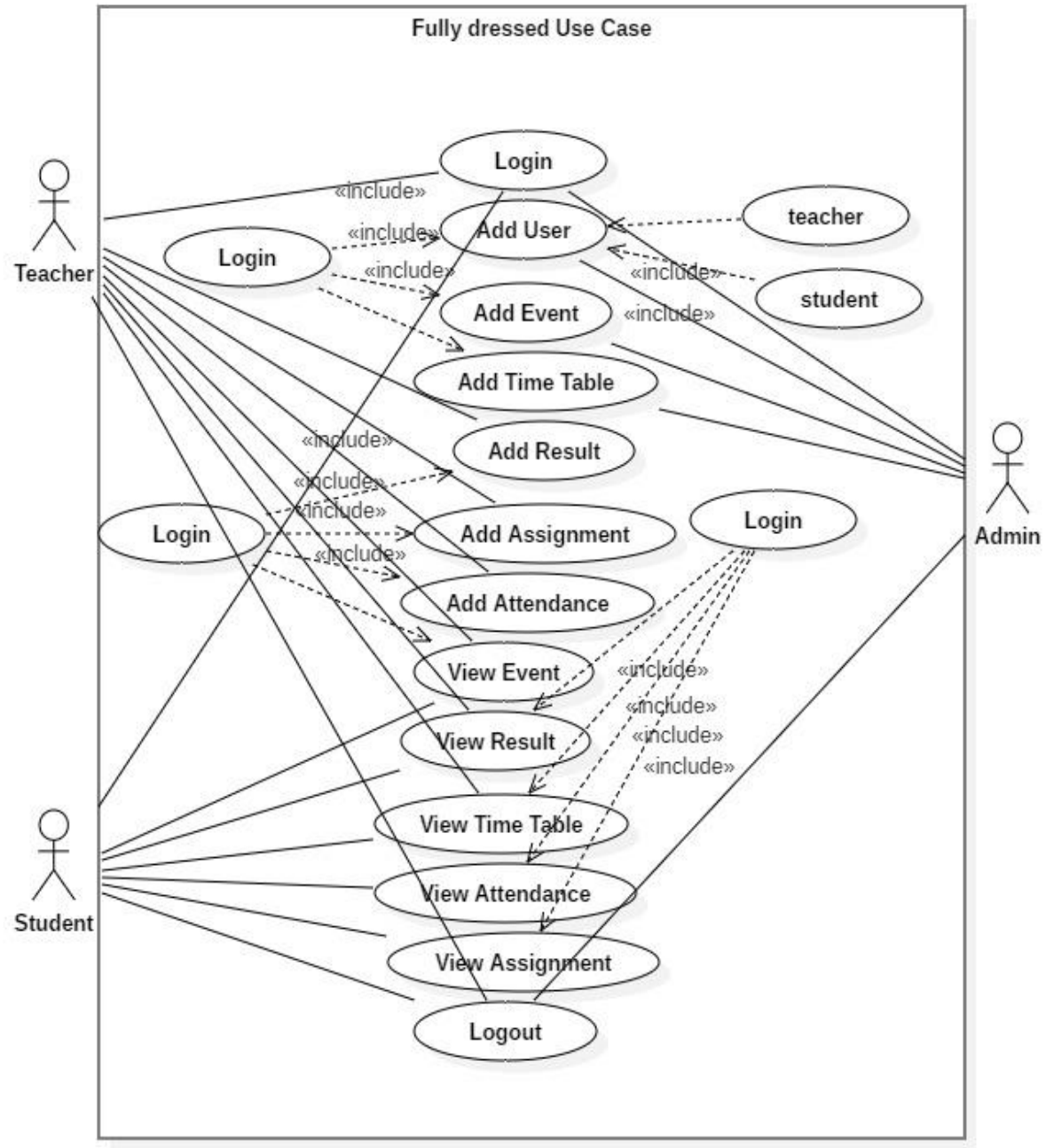


Figure 3.3 USE CASE Diagram

3.2. Fully Dressed Use Cases



Chapter 4

System Design

Chapter 4: System Design

CMS is a web portal where user can will access system through internet to system and data store into the database going from web server. Admin can access database directly as well. System is designed for educational institute to perform and store daily activities.

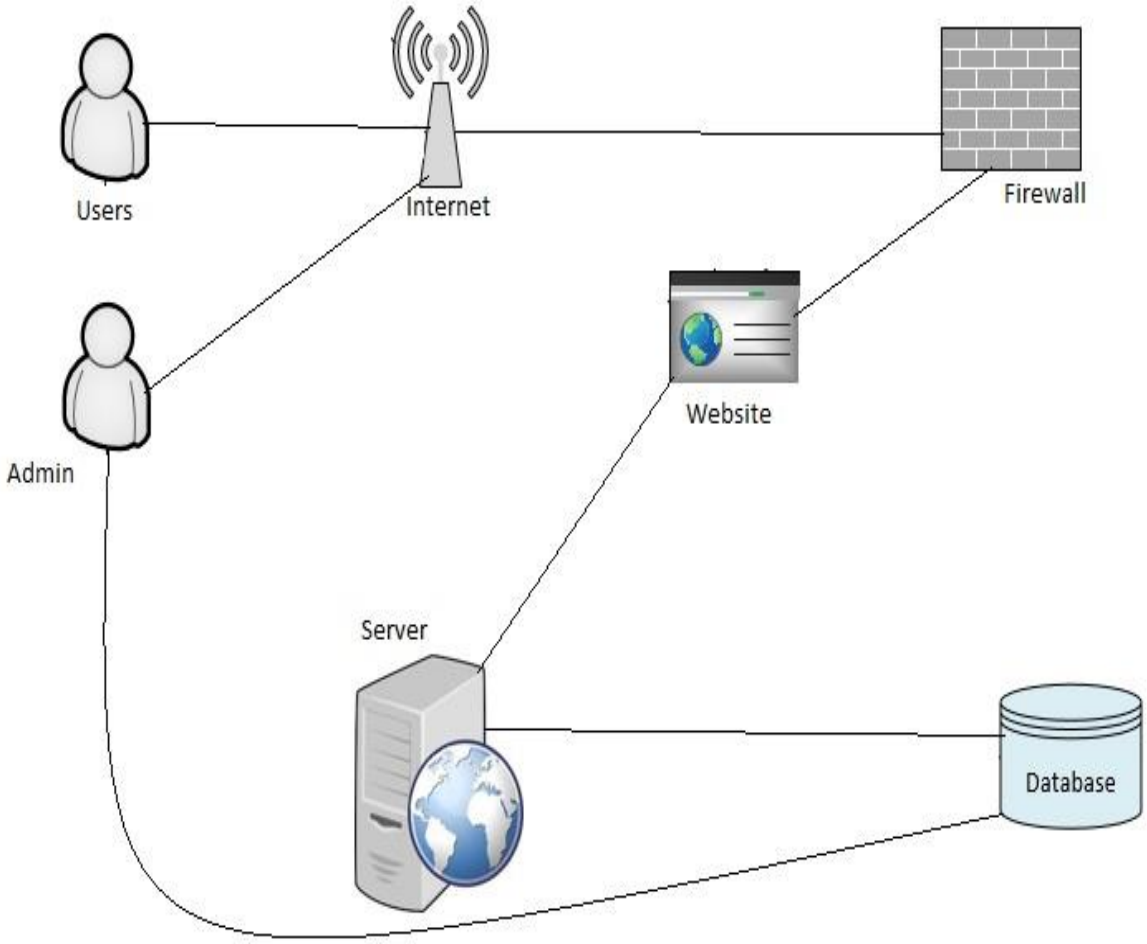


Figure 4.0 System Design

4.1. Architecture Diagram

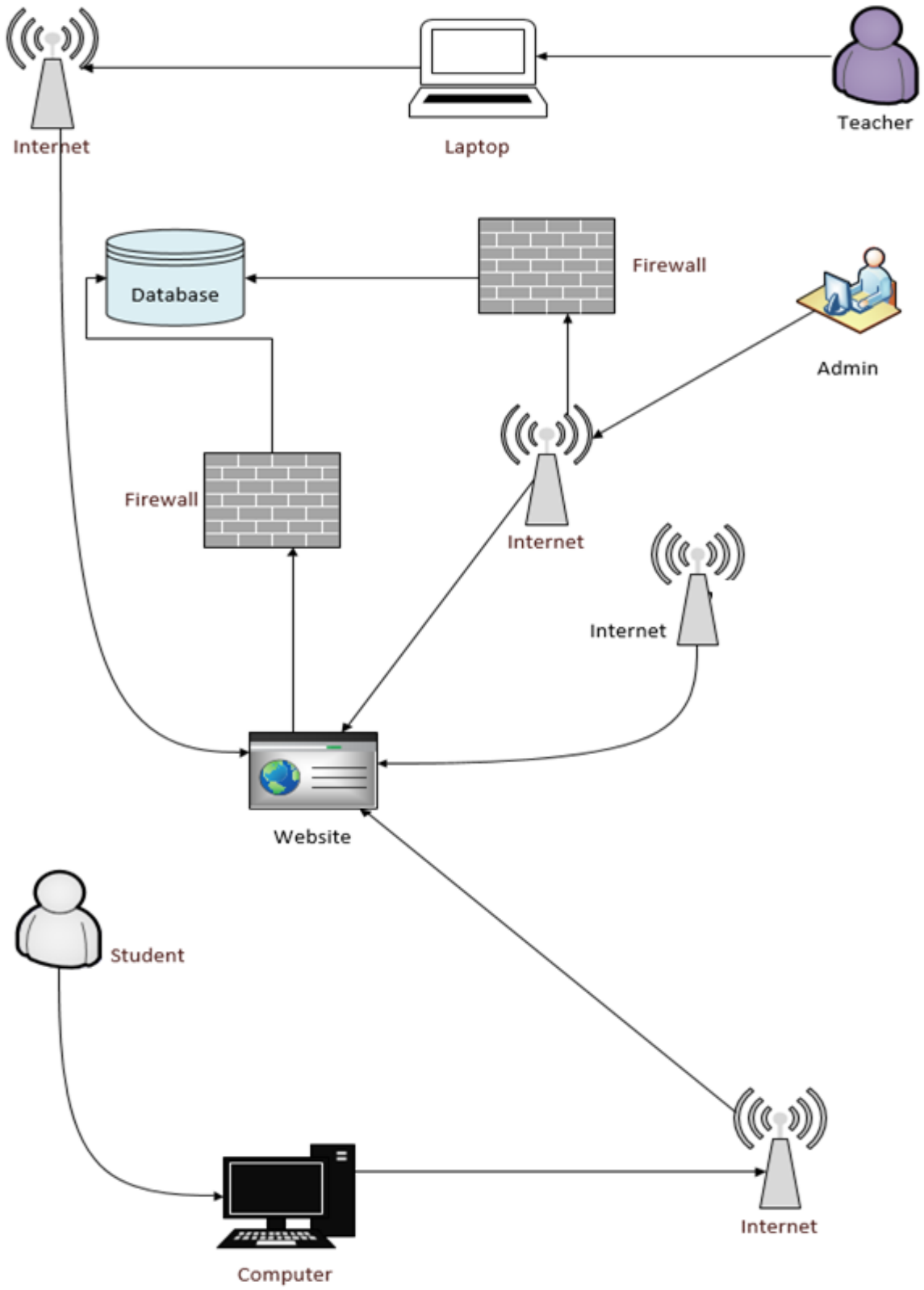


Figure 4.1 Architecture Diagram

4.2. Domain Model

A model that describe important abstract classes and tables. Which helps to understand the programming aspects of the system.

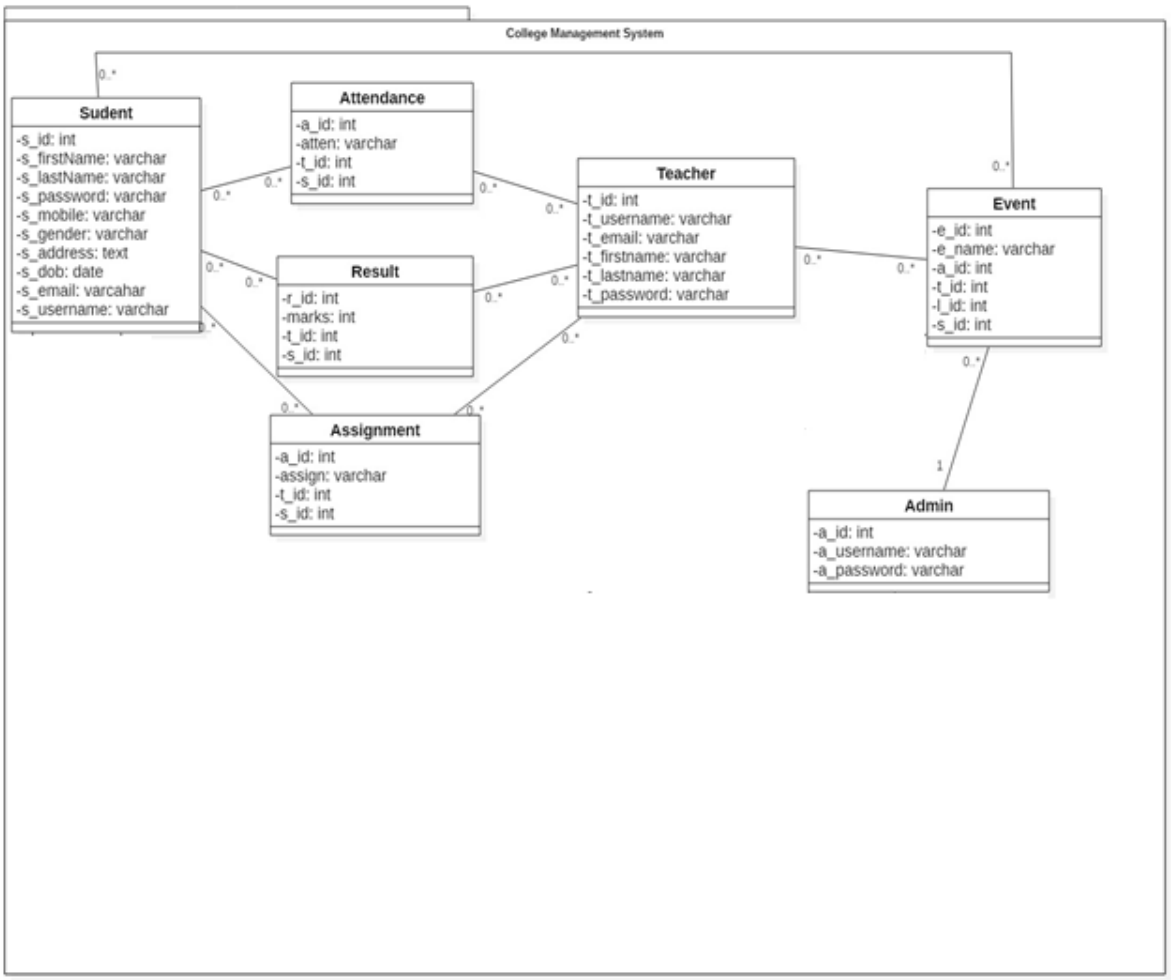


Figure 4.2 Design System

4.3. Entity Relationship Diagram with data dictionary

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

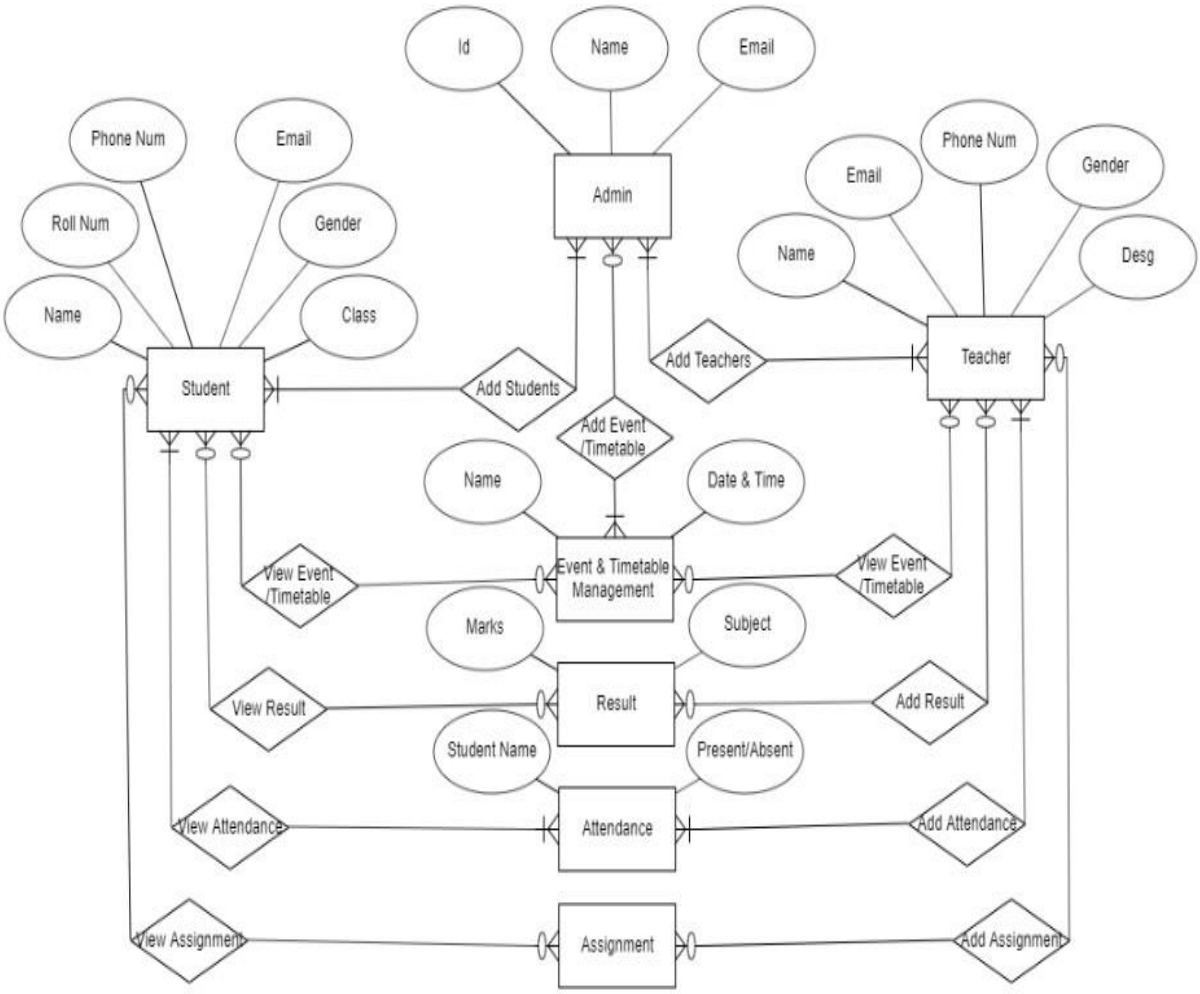


Figure 4.3 Entity Relationship Diagram

4.4. Class Diagram

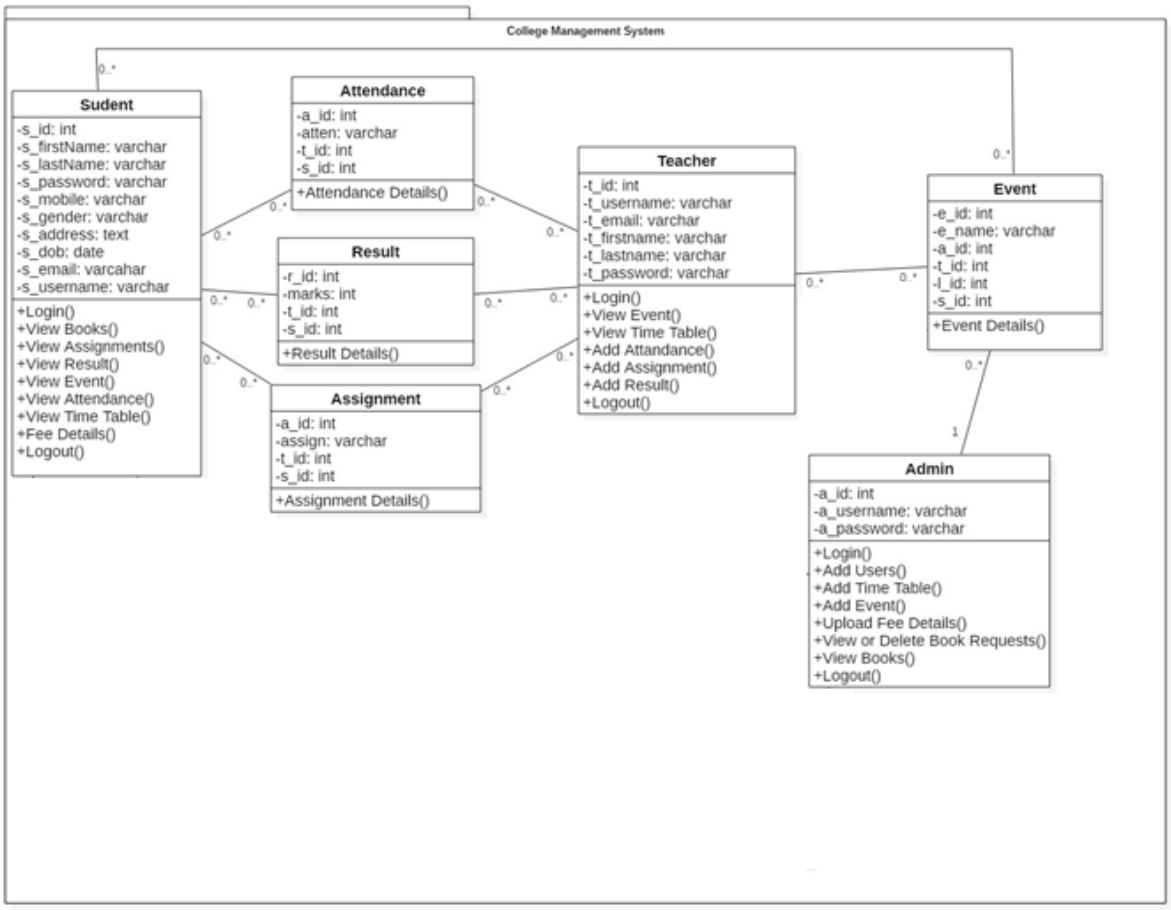


Figure 4.4 Class Diagram

4.5. Sequence / Collaboration Diagram

Admin Side

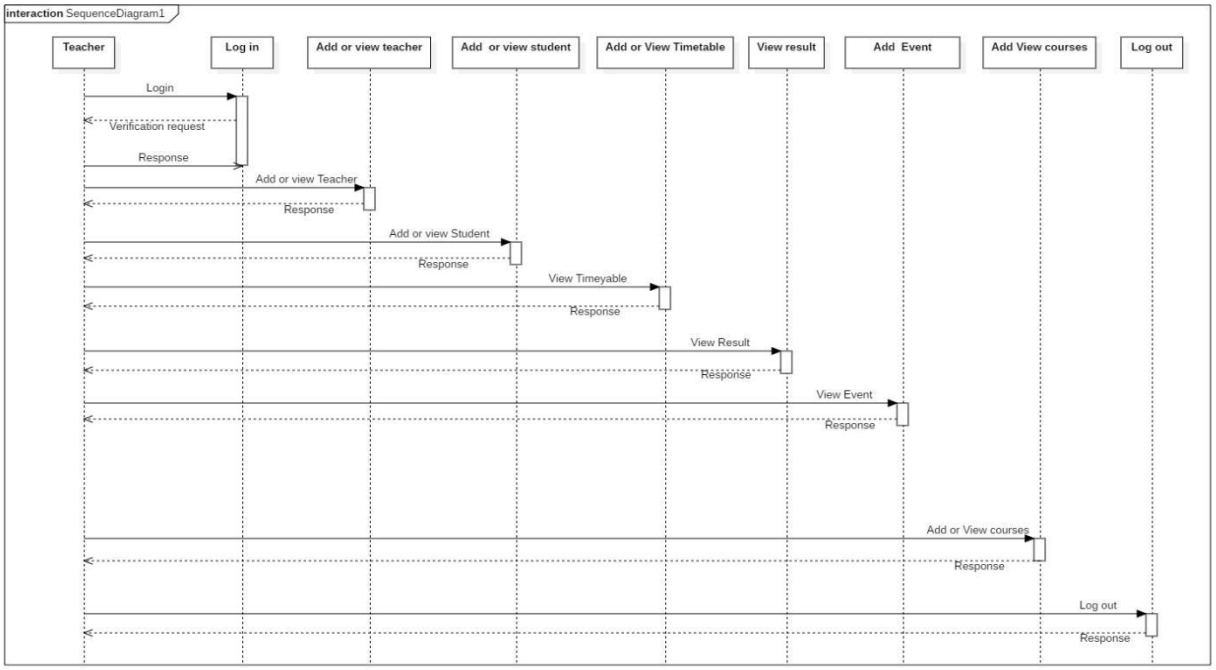


Figure 4.5.1 Sequence Diagram (Admin Side)

Teacher Side

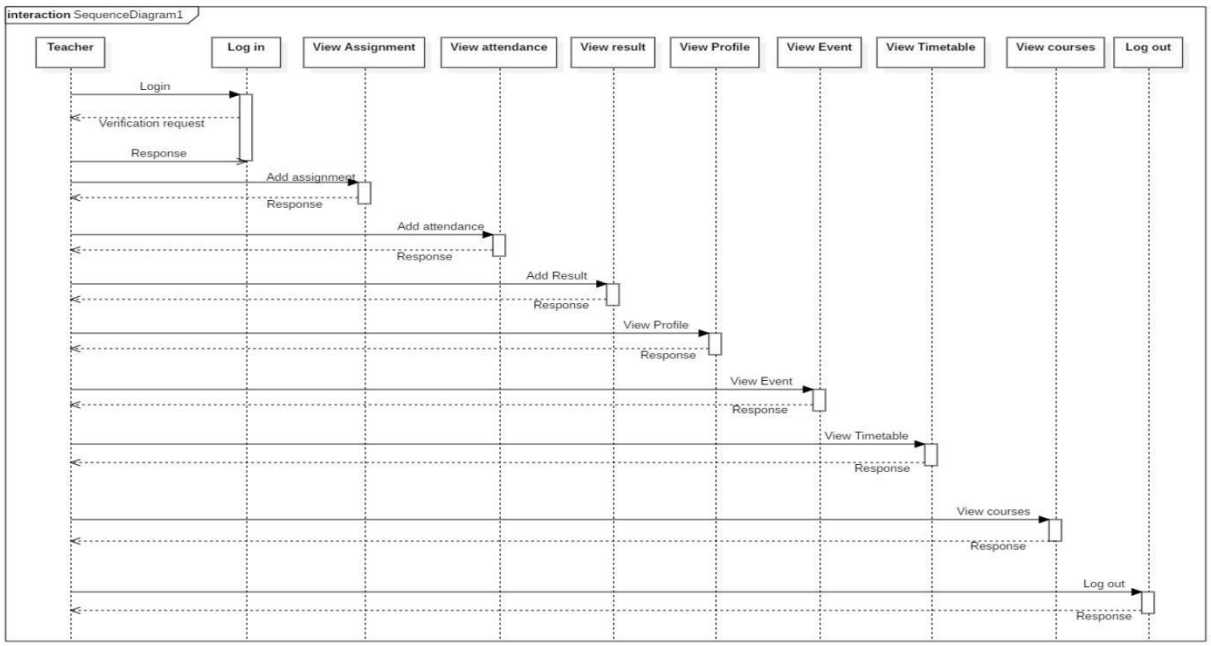


Figure 4.5.1 Sequence Diagram (Teacher Side)

Student Side

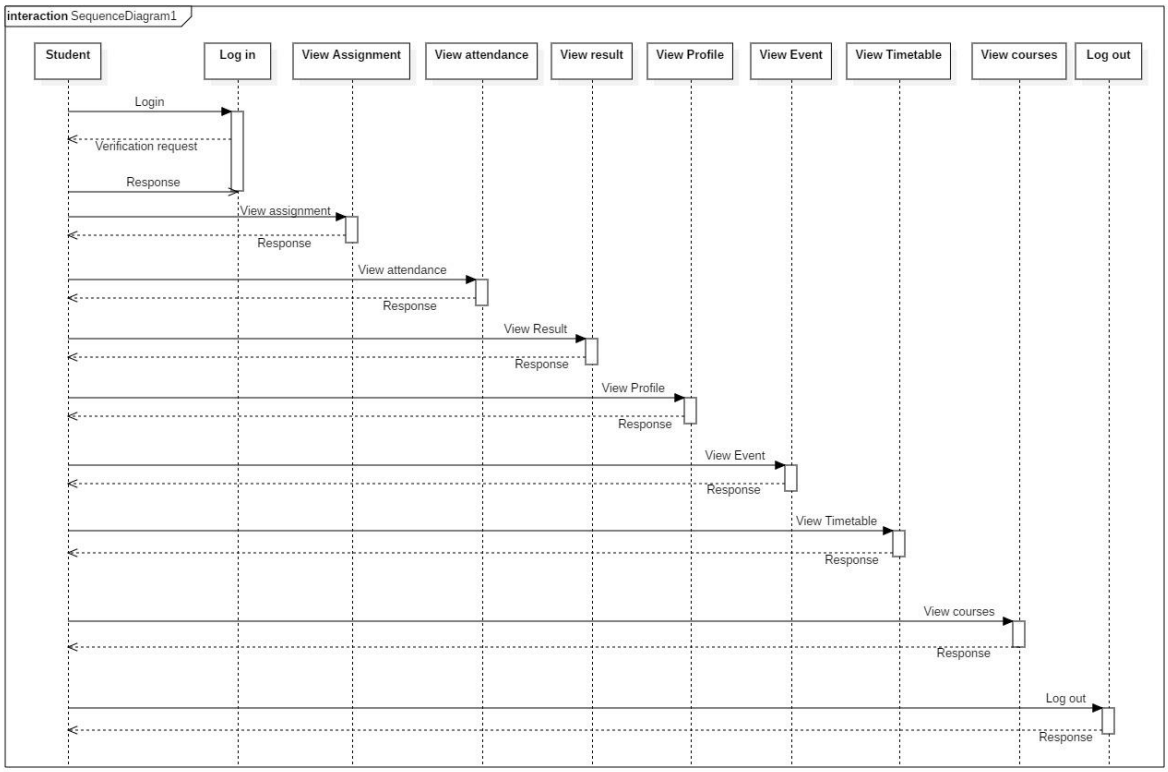


Figure 4.5.1 Sequence Diagram (Teacher Side)

4.6. Operation contracts

4.6.1 Operation Contract#1

- **Name:** Add Users
- **Responsibilities:** To add new users
- **Cross References:** Use Case: Add Users
- **Exceptions:** Invalid user
- **Preconditions:** Admin must be logged in.
- **Post conditions:** Admin redirect to dashboard after click the add user button.

4.6.2 Operation Contract#2

- **Name:** Log in
- **Responsibilities:** Login to system
- **Cross References:** Use Case: Login

- **Exceptions:** Invalid username or password
- **Preconditions:** User must be registered
- **Post conditions:** Successfully login

4.6.3 Operation Contract#3

- **Name:** Logout
- **Responsibilities:** Logout from the site.
- **Cross References:** Use Case: Logout
- **Exceptions:** None
- **Preconditions:** User must be logged in.
- **Post conditions:** Logout successfully.

4.6.4 Operation Contract#4

- **Name:** Event handling
- **Responsibilities:** Add important event details.
- **Cross References:** Use Case: Add Event
- **Exceptions:** Wrong event details
- **Preconditions:** Admin must be logged in.
- **Post conditions:** Event added successfully.

4.6.6 Operation Contract#6

- **Name:** Add/Update Time Table
- **Responsibilities:** Adding and updating schedule of classes.
- **Cross References:** Use Case: Add/update Time Table
- **Exceptions:** Time clash
- **Preconditions:** Admin must be logged in.
- **Post conditions:** Time table updated.

4.6.7 Operation Contract#7

- **Name:** Add Course
- **Responsibilities:** Add Course for the users.
- **Cross References:** Use Case: Add Course

- **Exceptions:** Same course added again
- **Preconditions:** Admin must be logged in.
- **Post conditions:** Courses added successfully.

4.6.9 Operation Contract#9

- **Name:** Register Course
- **Responsibilities:** Add details of course issuing.
- **Cross References:** Use Case: Register Course
- **Exceptions:** Wrong course id
- **Preconditions:** Admin must be logged in.
- **Post conditions:** Course Registered.

4.6.10 Operation Contract#10

- **Name:** Add Attendance
- **Responsibilities:** Mark Attendance of students.
- **Cross References:** Use Case: Add Attendance
- **Exceptions:** Student not registered
- **Preconditions:** Teacher must be logged in.
- **Post conditions:** Attendance uploaded successfully.

4.6.11 Operation Contract#11

- **Name:** Add Assignment
- **Responsibilities:** Upload assignment for students.
- **Cross References:** Use Case: Add Assignment
- **Exceptions:** File size
- **Preconditions:** Teacher must be logged in.
- **Post conditions:** Assignment uploaded.

4.6.12 Operation Contract#12

- **Name:** Add Result
- **Responsibilities:** Upload result of students.
- **Cross References:** Use Case: Add Result

- **Exceptions:** None
- **Preconditions:** Teacher must be logged in.
- **Post conditions:** Result uploaded successfully.

4.6.13 Operation Contract#13

- **Name:** View Attendance
- **Responsibilities:** View attendance on the site.
- **Cross References:** Use Case: View Attendance
- **Exceptions:** Registration Register s
- **Preconditions:** Student must be logged in.
- **Post conditions:** View attendance table.

4.6.14 Operation Contract#14

- **Name:** View Assignment
- **Responsibilities:** View assignment on the site.
- **Cross References:** Use Case: View Assignment
- **Exceptions:** Registration Register
- **Preconditions:** Student must be logged in.
- **Post conditions:** View assignment file or text.

4.6.15 Operation Contract#15

- **Name:** View Result
- **Responsibilities:** View result on the site.
- **Cross References:** Use Case: View Result
- **Exceptions:** Registration Register
- **Preconditions:** User must be logged in.
- **Post conditions:** View result table.

4.6.17 Operation Contract#17

- **Name:** View Course
- **Responsibilities:** View Course Details on the site.
- **Cross References:** Use Case: View Course
- **Exceptions:** Registration Register

- **Preconditions:** Student must be logged in.
- **Post conditions:** View courses in records.

4.6.20 Operation Contract#20

- **Name:** View Event
- **Responsibilities:** View Event details on the site.
- **Cross References:** Use Case: View Event
- **Exceptions:** Registration Register
- **Preconditions:** User must be logged in.
- **Post conditions:** View event details

4.7. Activity Diagram

Admin Side

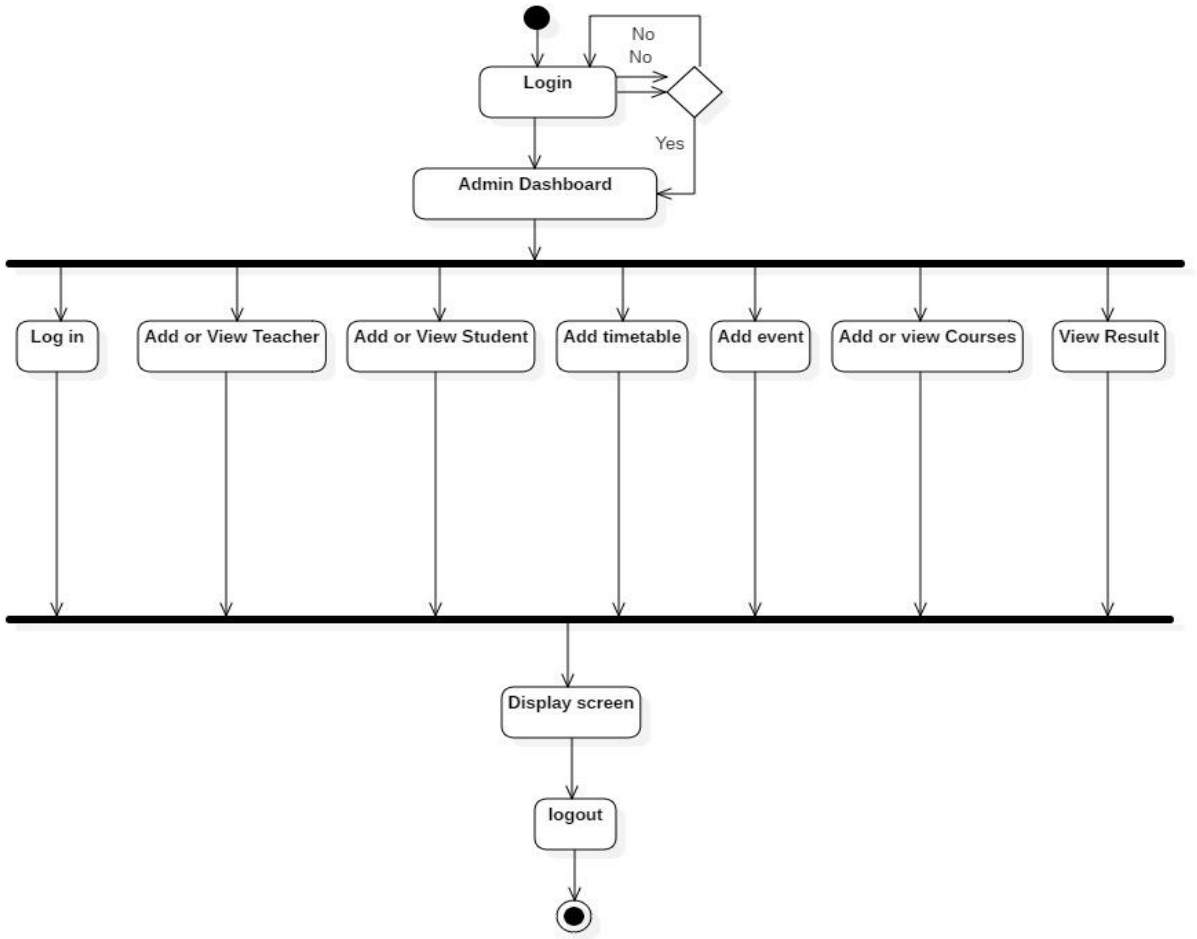


Figure 4.7.1 Activity Diagram (Admin Side)

Teacher Side

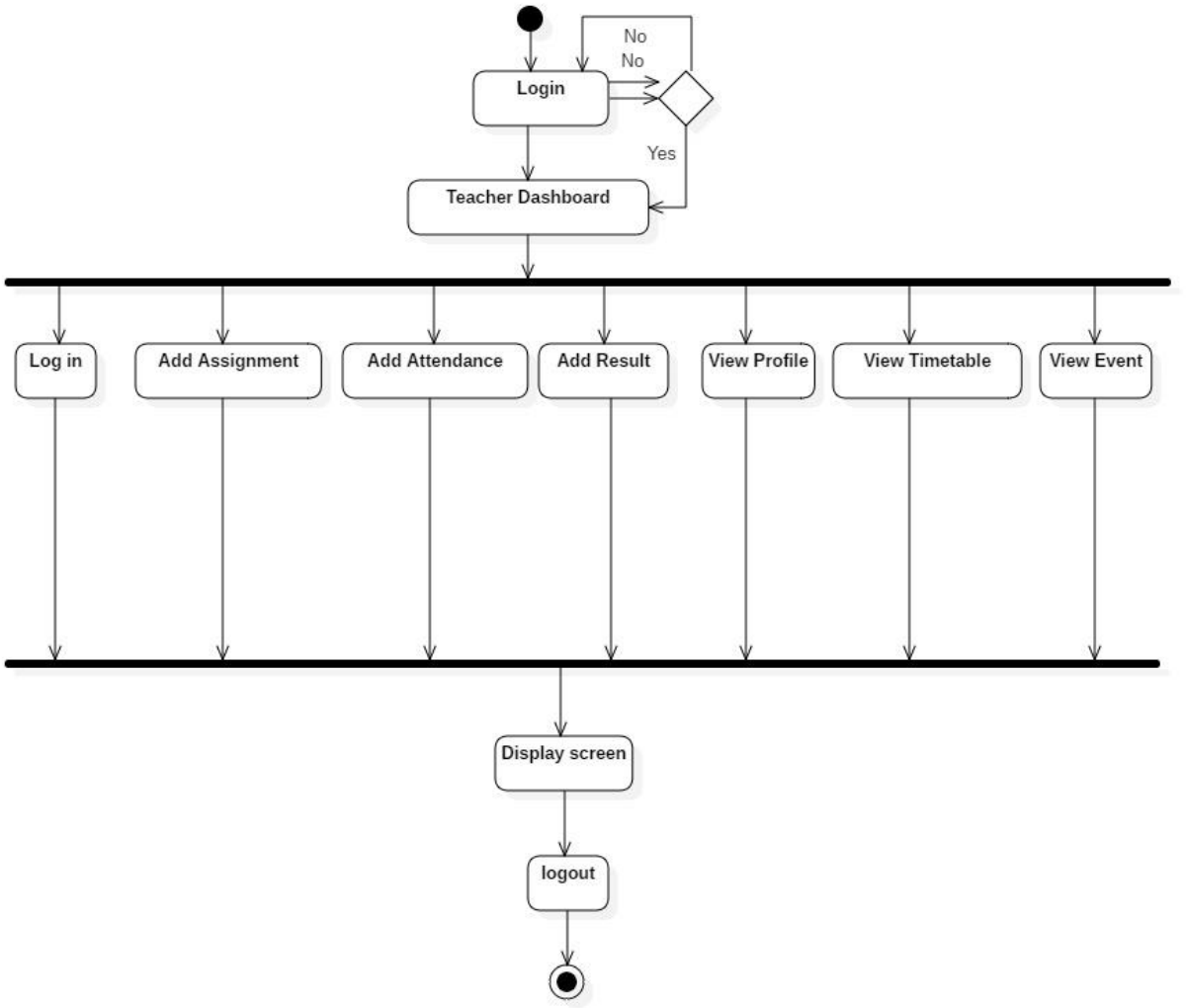


Figure 4.7.2 Activity Diagram (Teacher Side)

Student Side

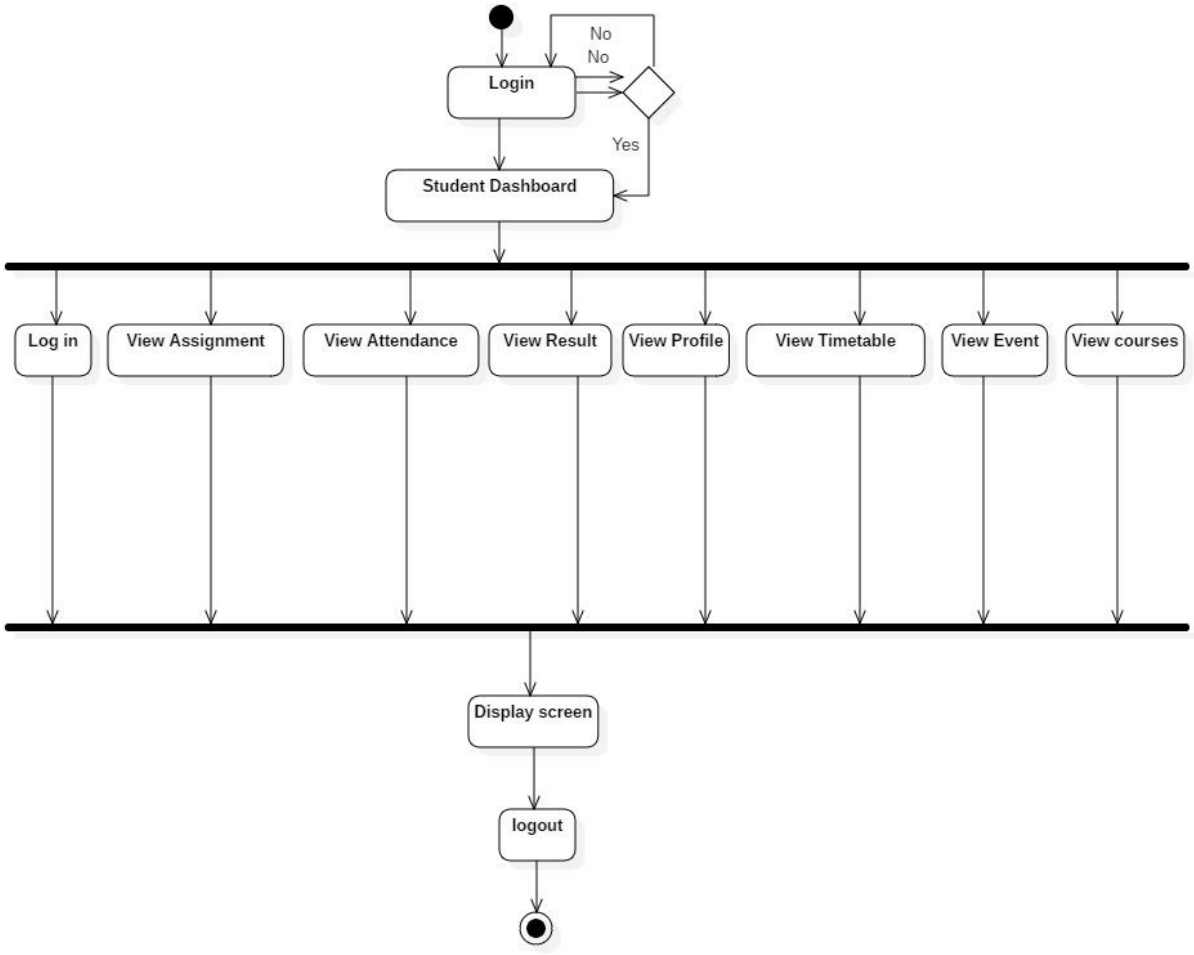


Figure 4.7.2 Activity Diagram (Student Side)

4.8. State Transition Diagram

Admin Side

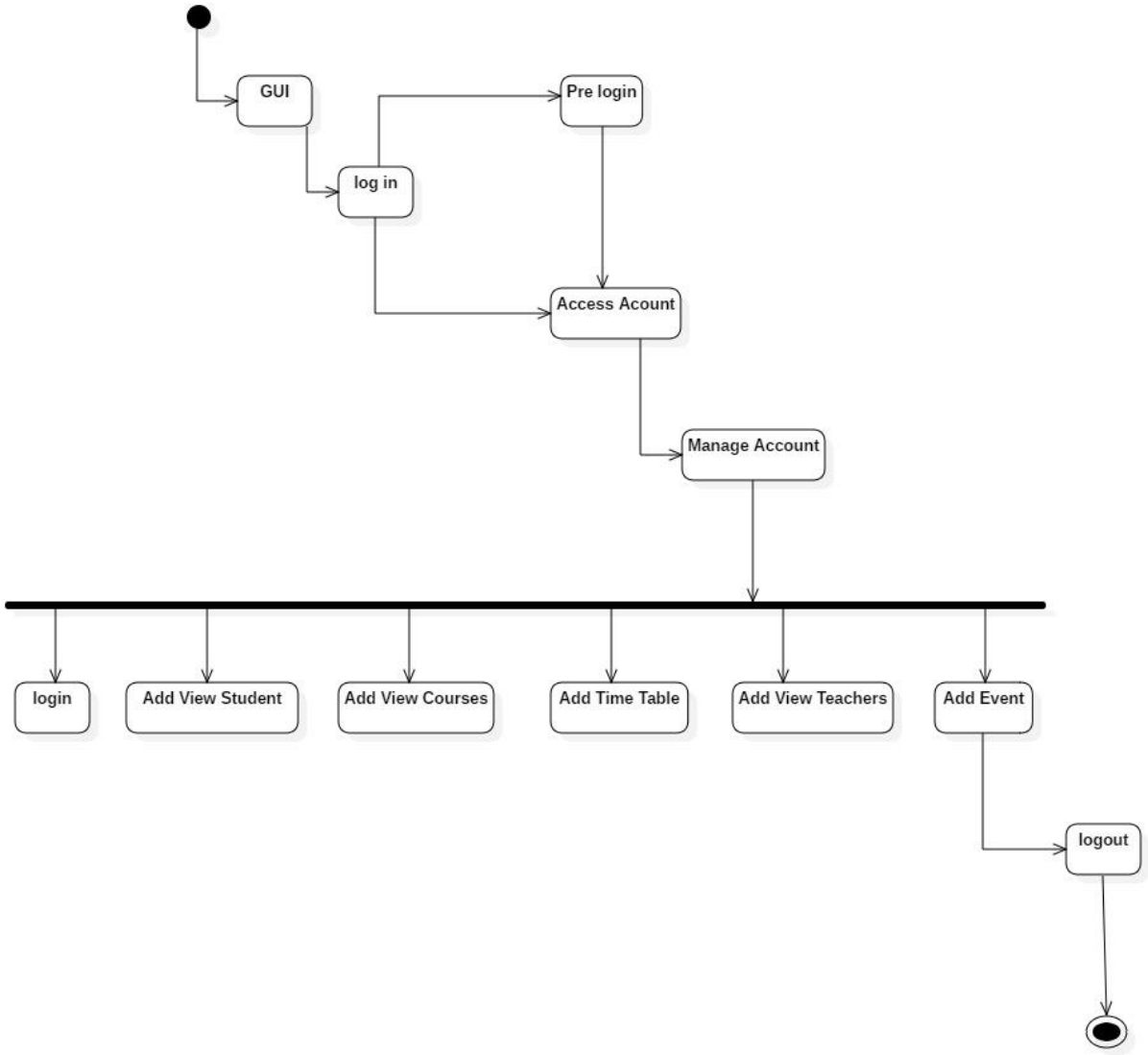


Figure 4.8.1 State Transition Diagram (Admin Side)

Teacher Side

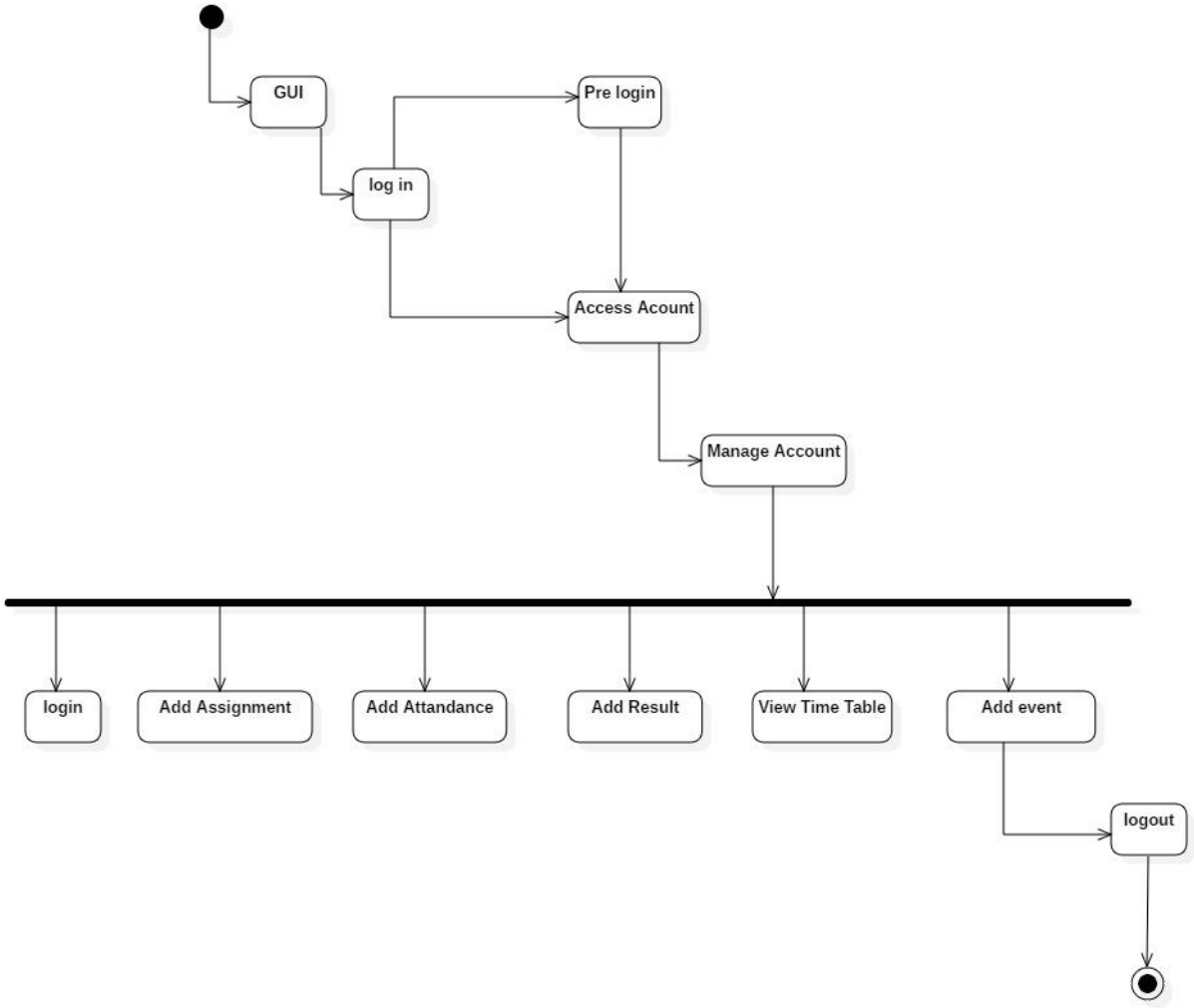


Figure 4.8.3 State Transition Diagram (Teacher Side)

Student Side

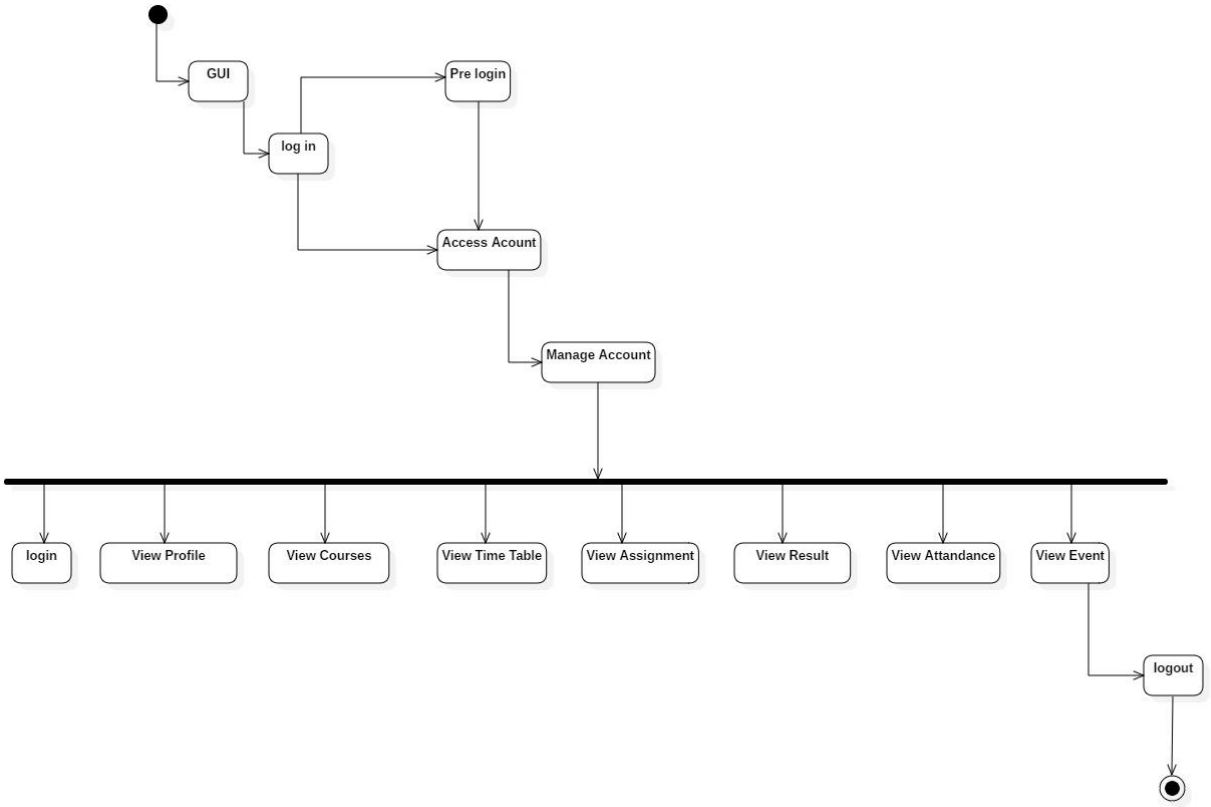


Figure 4.8.4 State Transition Diagram (Student Side)

4.9. Component Diagram

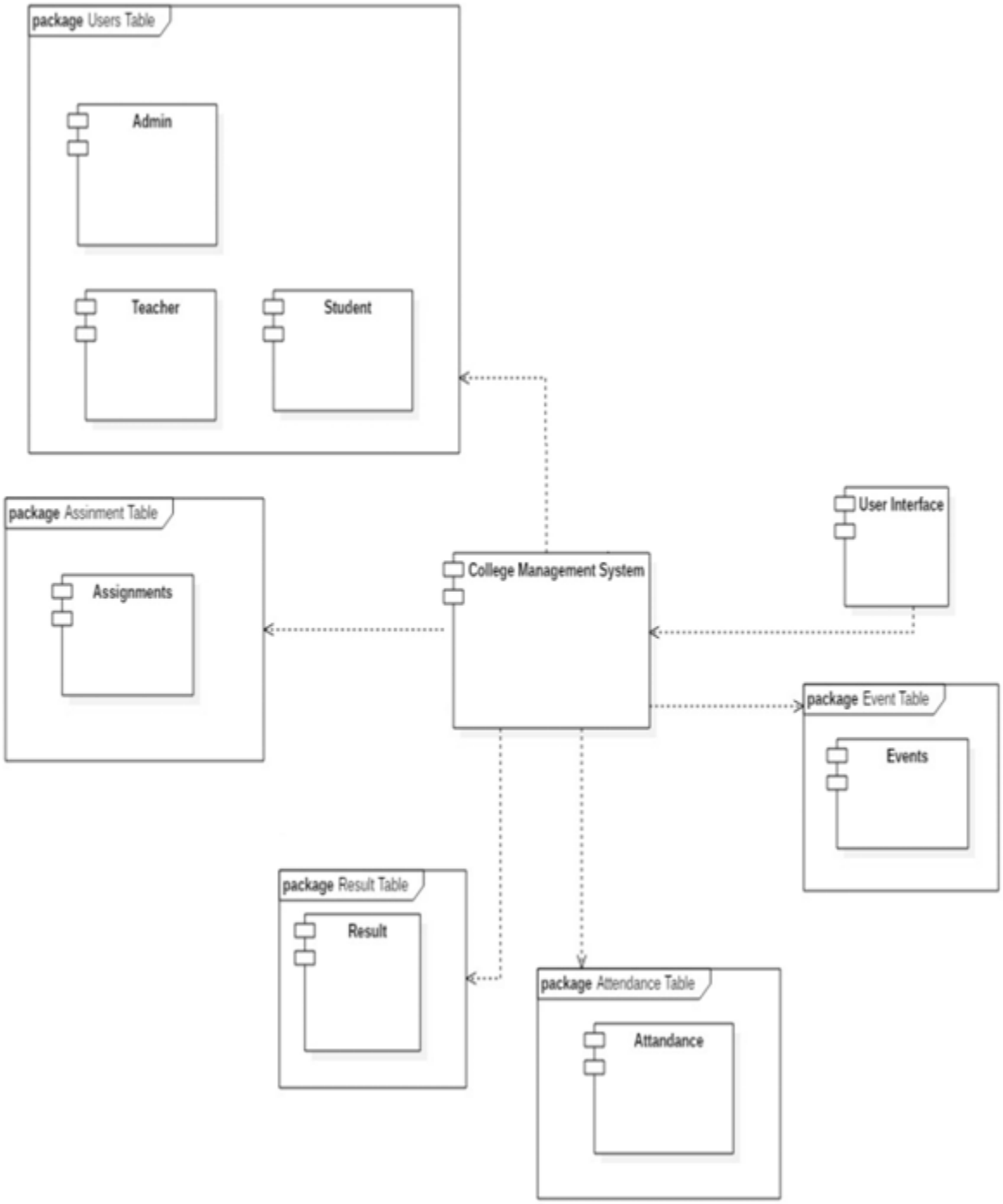


Figure 4.9 Component Diagram

4.10. Deployment Diagram

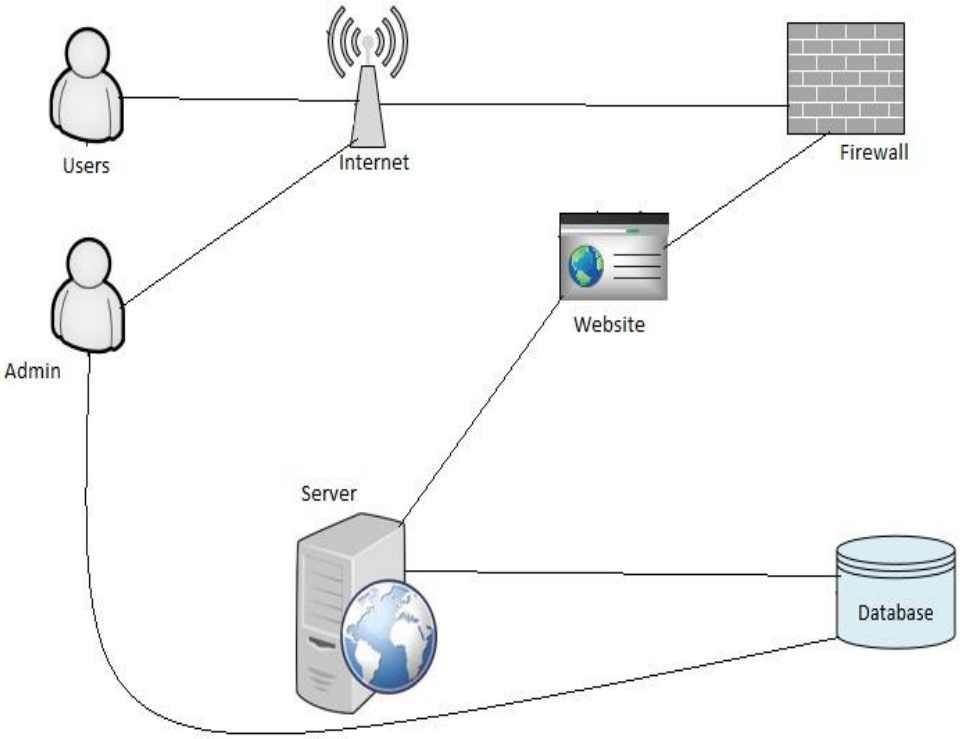


Figure 4.10 Deployment Diagram

4.11. Data Flow diagram

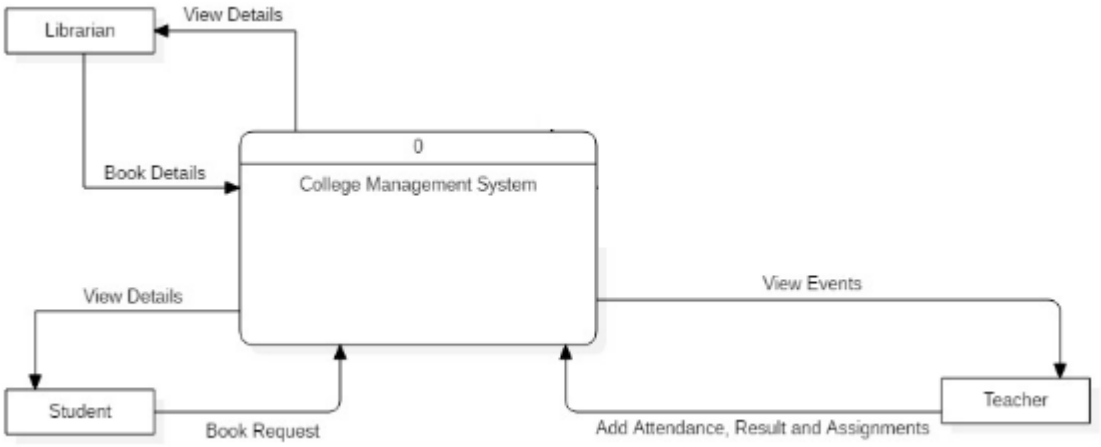


Figure 4.11 Data Flow Diagram (Level 0)

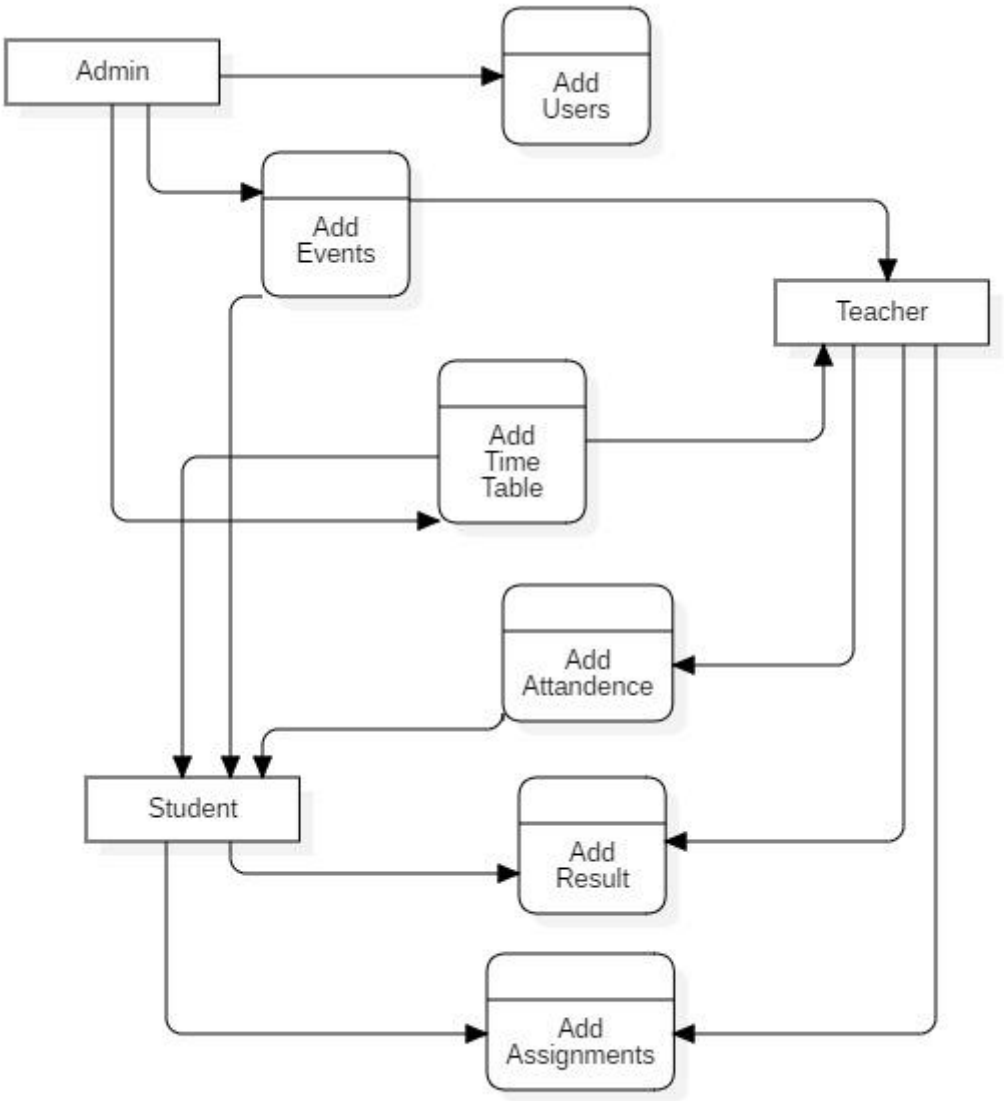


Figure 4.11.1 Data Flow Diagram (Level 1)

Chapter 5

Implementation

Chapter 5: Implementation

To make up an essential web portal where providing a complete college management system, in which all daily activities performing under point of access. User can access web portal from anywhere with internet. This report will talk about every one of the fundamental advances to make an actualize web portal for educational institute.

5.1. Important Flow Control/Pseudo codes

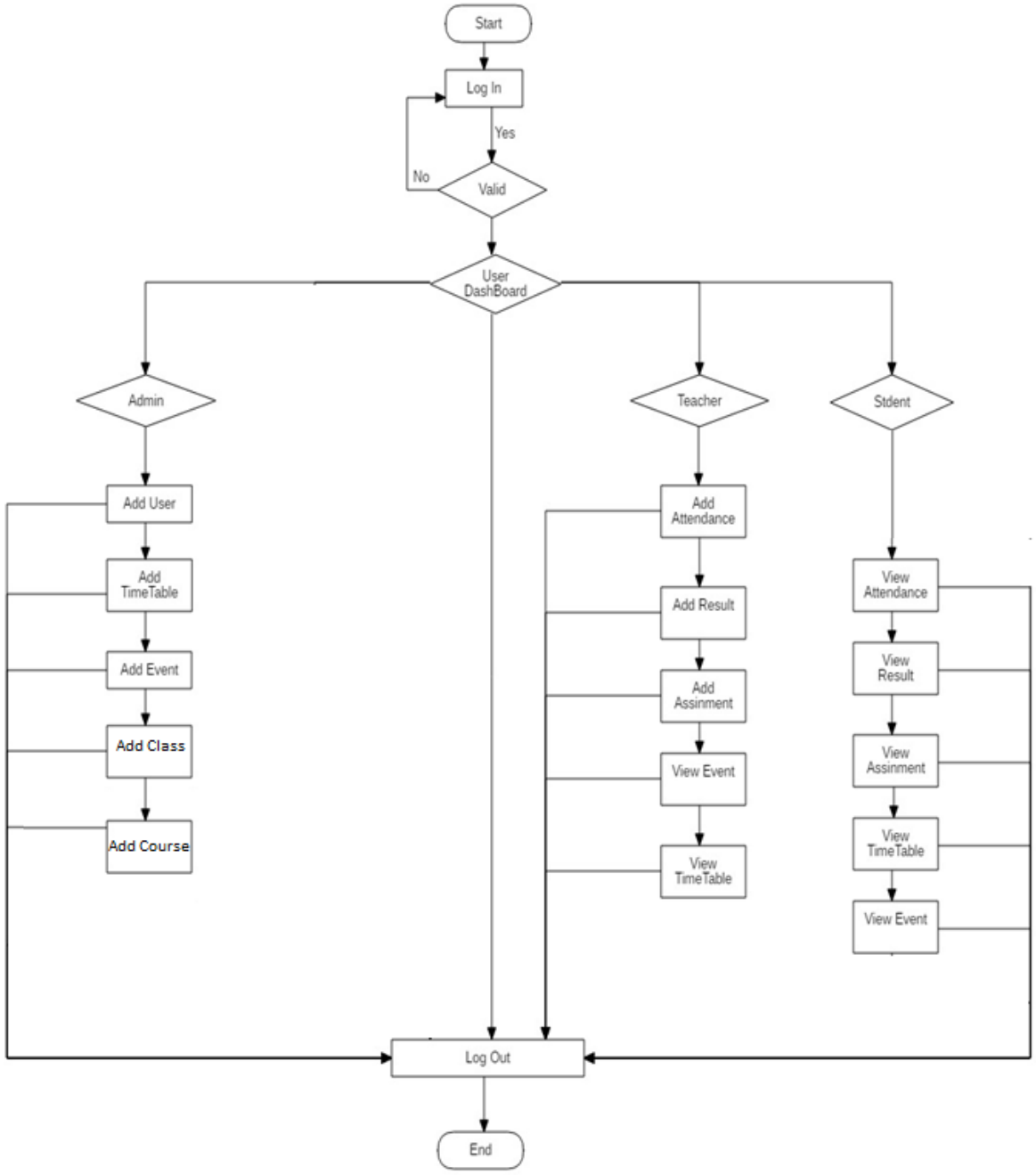


Figure 5.1 Flow Control Diagram

5.2. Components, Web Services and stubs

Properly Structured Website

CMS is easy to use. If any user wants to perform any activity on web portal he/she can easily access. Our website has a simple structure, with the most important pages accessible from the dashboard.

Web Services

- Users straightforwardly perform activities
- 24 hour access

5.3. Deployment Environment

We need a domain on which the college management system will exists and through the domain several people will access the portal. They can easily access the service. It required one administrator who will be in charge of the approval and other supervision duty.

5.4. Tools and Techniques

- **PHP** is a server-side scripting language designed for Web development.
- **HTML** (Hypertext Markup Language) is a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.
- **CSS** (Cascading Style Sheets) is used to provide layout and visual formatting to the HTML pages.
- **JavaScript** is an object-oriented computer programming language commonly used to create interactive effects within web browsers.
- **Bootstrap** is a free and open-source front-end framework for designing responsive websites and web applications.
- **MySQL** is an open-source relational database management system.
- **PhpMyAdmin** is a free and open source administration tool for MySQL.
- **Notepad++** is a text editor and source code editor for use with Microsoft Windows.
- **XAMPP** is a free and open-source cross-platform web server solution stack package developed by Apache Friends.

Project Modeling:

We are applying agile model for the development of the project.

Server:

Server is also required to store the whole data.

5.5. Best Practices / Coding Standards

- No coding repetition.
- Function and classes are always commented showing expected input and output.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Evaluation and testing is very important for a system that is newly generated. In evaluation and testing you are test and evaluate your system step by step. Through these steps you would be sure about your system that is works correctly or not. In CMS testing and evaluation is very important for the reliability of the system. Quick response is very important for the maintenance and scalability of your system. Testing strategy is decided for testing the system so that all modules are tested to fulfill all user requirements. A test strategy is methodology that describes the various steps that need to be performed during testing and the time and effort required for performing them. The following strategies are used for testing.

- Unit Testing
- Integrated Testing
- System Testing
- Recovery Testing
- Security Testing
- Stress Testing
- Performance Testing

6.1. Use Case Testing

In use case testing a system is used by an actor or a user. It tests the entire activity of a system. In our CMS a first activity that is performed by the user is login into the system. In use case actor is represented by their names. System will create a message enter password and user will enter the password and then system generate a response message like valid password or not. This validity of message shows the testing of system and user activity.

Test case 1

Add Users

Id	1	
Name	Add Users	
Summary	Add users into system.	
Priority	5	
Preconditions	User should be teacher and student.	
Post conditions	Users Successfully added.	
Primary actor	Admin, teacher and student.	
Secondary actor	Database	
Trigger	Click on the add user button.	
Main scenario	**Step**	**Actions**
	1	System will display add user form.
	2	Admin will add users into the form.
	3	Admin will click the add user button and user will be added.
Extensions	**Step**	**Actions**
	1	User would not be valid.
	2	Wrong user id.
Open Register s	**Step**	**Actions**
	1	Forget user id.

Table Test Case 6.1

Test case 2

Login

Id	2	
Name	Login	
Summary	User Login into his/her account	
Priority	5	
Preconditions	User should be registered.	
Post conditions	Successfully log in	
Primary actor	Register Users	
Secondary actor	Database	
Trigger	Click on the Login button	
Main scenario	**Step**	**Actions**
	1	System will display login page.
	2	User can perform any activity to his/her accessible rights.
Extensions	**Step**	**Actions**
	1	Not registered.
	2	Wrong username or password.
Open Register s	**Step**	**Action**
	1	Forget username or password.

Table Test Case 6.2

Test case 3

Event Handling

Id	3
Name	Event handling

Summary	Add event	
Priority	2	
Preconditions	Admin must be logged in	
Post conditions	Event Added	
Primary actor	Admin	
Secondary actor	Database	
Trigger	Click on the add event button	
Main scenario	**Step**	**Actions**
	1	System will display event input text box.
	2	Admin will add all the event details.
	3	Admin will click on the add event button.

Table Test Case 6.3

Test case 4

Add/Update Time Table

Id	4	
Name	Add/Update Time Table	
Summary	Add/Update Time Table into system.	
Priority	4	
Preconditions	Admin must be logged in	
Post conditions	Time table updated	
Primary actor	Admin	
Secondary actor	Database	
Trigger	Click on the add time table button.	
Main scenario	**Step**	**Actions**
	1	System will display schedule form.
	2	Admin will add/update all schedule date.
	3	Admin will click on the add/update button.

Extensions	Step	Actions
	1	Time clash.
	2	Room and lab unavailability.
Open Register s	Step	Actions
	1	Manage time according to available slots.

Table Test Case 6.4

Test case 5

Add Course

Id	5	
Name	Add Courses	
Summary	Add course into records	
Priority	4	
Preconditions	Admin must be logged in	
Post conditions	Course added	
Primary actor	Admin	
Secondary actor	Database	
Trigger	Click on add course button	
Main scenario	Step	Actions
	1	System will display courses.
	2	Admin will add course.
	3	Admin will click on the add button.

Table Test Case 6.5

Test case 6

Register Course

Id	6
Name	Register Course
Summary	Register course to student

Priority	3	
Preconditions	Admin must be logged in	
Post conditions	Course Register d	
Primary actor		
Secondary actor	Database	
Trigger	Click on Register course button	
Main scenario	**Step**	**Actions**
	1	System will display courses.
	2	Admin will Register course to student.

Table Test Case 6.6

Test case 7

Add Attendance

Id	7	
Name	Add Attendance	
Summary	Mark student attendance	
Priority	5	
Preconditions	Teacher must be logged in	
Post conditions	Attendance added	
Primary actor	Teacher	
Secondary actor	Database	
Trigger	Click on submit attendance button	
Main scenario	**Step**	**Actions**
	1	System will display attendance table.
	2	Teacher will mark attendance into it.

Table Test Case 6.7

Test case 8

Add Assignment

Id	8	
Name	Add Assignment	
Summary	Upload assignment	
Priority	4	
Preconditions	Teacher must be logged in	
Post conditions	Assignment added	
Primary actor	Teacher	
Secondary actor	Database	
Trigger	Click on upload assignment button	
Main scenario	**Step**	**Actions**
	1	Teacher will upload assignment.
	2	Click on upload assignment button.

Table Test Case 6.8

Test case 9

Add Result

Id	9	
Name	Add Result	
Summary	Upload result	
Priority	4	
Preconditions	Teacher must be logged in	
Post conditions	Result uploaded successfully	
Primary actor	Teacher	
Secondary actor	Database	
Trigger	Click on upload result button	
Main scenario	**Step**	**Actions**
	1	System will display result table.
	2	Teacher will upload result.

Table Test Case 6.9

Test case 10

View Attendance

Id	10	
Name	View Attendance	
Summary	View attendance in system	
Priority	5	
Preconditions	Student must be logged in	
Post conditions	View attendance tables	
Primary actor	Student	
Secondary actor	Database	
Trigger	Click on view attendance button	
Main scenario	**Step**	**Actions**
	1	System will display attendance table.
	2	Student will view attendance.

Table Test Case 6.10

Test case 11

View Assignment

Id	11
Name	View Assignment
Summary	Student view assignment
Priority	5
Preconditions	Student must be logged in
Post conditions	View assignment file or text
Primary actor	Student
Secondary actor	Database
Trigger	Click on view assignment button

Main scenario	Step	Actions
	1	System will display assignment page.
	2	Student will view assignment.

Table Test Case 6.11

Test case 12

View Result

Id	12	
Name	View Result	
Summary	Student and Admin view result	
Priority	4	
Preconditions	Student must be logged in	
Post conditions	View result table	
Primary actor	Admin and Student	
Secondary actor	Database	
Trigger	Click on view result button	
Main scenario	Step	Actions
	1	System will display result table.
	2	Student and admin will view result.

Table Test Case 6.12

Test case 13

View Courses

Id	13
Name	View Courses
Summary	Student view courses
Priority	3
Preconditions	User must be logged in
Post conditions	View courses in records

Primary actor	Admin, Student	
Secondary actor	Database	
Trigger	Click on view courses button	
Main scenario	**Step**	**Actions**
	1	System will display course stable.
	2	User will view courses.

Table Test Case 6.13

Test case 14

View Events

Id	14	
Name	View Events	
Summary	User view events	
Priority	3	
Preconditions	User must be logged in	
Post conditions	View event details	
Primary actor	Teacher and student	
Secondary actor	Database	
Trigger	Click on view event button	
Main scenario	**Step**	**Actions**
	1	System will display event table.
	2	User will view event details.

Table Test Case 6.14

Test case 15: Logout

Id	15
Name	Logout
Summary	User can logout
Priority	4

Preconditions	User must be logged in	
Post conditions	Logout successfully	
Primary actor	Admin, teacher and student	
Secondary actor	Database	
Trigger	Click on logout button	
Main scenario	**Step**	**Actions**
	1	System will display logout option.
	2	Users will logout from system.

Table Test Case 6.15

6.2. Equivalence partitioning

Equivalent partitioning is a black box testing and it can be used for all types of testing like unit testing, integration testing, system testing etc. In our system we divide our system into different parts and in equivalent classes for testing. We can apply this method where a number of inputs are find.

Registration

Input	Invalid Equivalence class partitioning	Valid Equivalence class partitioning
Username	Less than 3 not allowed	Between 3 to 50
Password	Less than 8 not allowed	Between 8 to 20
Email	Admin123 format is not allowed	Admin123@hotmail.com Admin123@gmail.com
Contact	Between the numbers 0123456789	Not in the alphabetic form

Login

Input	Invalid equivalent partitioning	Valid equivalent partitioning
Email	Admin123 is not allow	Admin123@gmail.com
Password	Less than 8 is allowed	Between 8 to 20 is allowed

6.3. Boundary value analysis

Boundary analysis is useful when practically it is hard to test a large amount of data. In our system first we equivalence partition our classes and then apply boundary values on the system individually. This method is very advantageous for the system performance.

Register/Login

Username

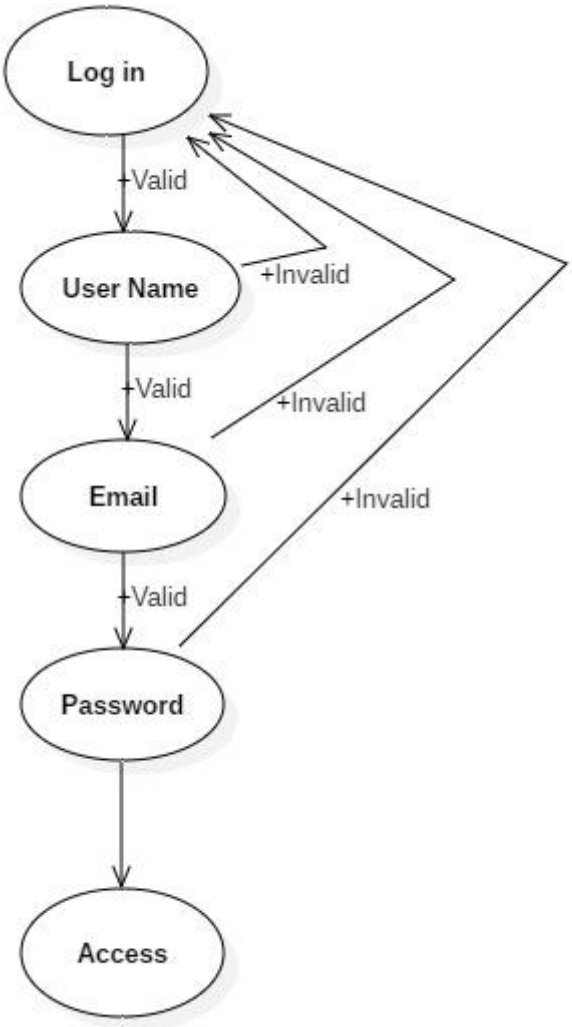
Invalid (min-1)	Valid (min, +max, -min, max)	Invalid (max+1)
3	3 to 50	50

Password

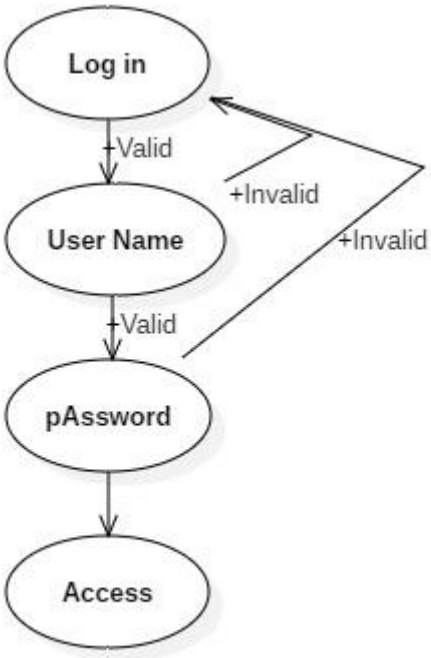
Invalid	Valid	Invalid
Password(min-1)	(min, +max, -min, max)	(max+1)
8	8 to 20	**20**

6.4. Data flow testing

Register:



Login:



6.5. Unit testing

Unit testing is a white-box testing technique. The main consideration in this test is verification of all modules of the software system. Each module is unit tested, as follows:

- Unit testing for the module User Management is tested by valid user or invalid user.
- Unit testing for the module User Account management is provided by testing Login, logout and creating new or deleting updating existing user.
- Unit testing for the module Detail of attendance/assignment/result/courses give detail accurately.
- Unit testing for the module Add/update/delete.

6.6. Integration testing

Integration testing is the technique for testing the interfaces of software components. Each software components in unit tested, and all the components are integrated to perform together. The tests are conducted to ensure that the components are

working properly after interfacing. All modules are integrated by an incremental approach, and integration testing of the system is performed as follows:

- Integrate login module and test that the software is properly connected to the database.
- Integrate all modules and test by inserting updating, and deleting records.

6.6.1SYSTEM TESTING

System testing involves the set of tests that ensures that the entire system performs according to specifications.

6.6.2SECURITY TESTING

System testing is protection testing that test security methods in the system to avoid invalid intrusions. Several security checks have been included in the system. The system requires a logon ID and password to initiate. For allowing only authenticated users to log on.

6.7. Performance testing

Performance testing uses criteria to check that the system functions according to the specifications. The performance of the Inventory and lease management system is tested at all levels of testing. All unit modules have been tested successfully. The integration of these unit modules produces reports in the required format.

6.8. Stress Testing

A system evaluated under normal conditions in all previous testing techniques, and no conditions where system can fail are tested. Stress testing evaluates the system under abnormal conditions. The system is tested for various costs and number to check accuracy and to ensure that the system performs accurately under all input conditions.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

Basically this project helps college management system that how to maintain the students records very well like attendance management, assignment management and others records. In this project we also helps our students that they can view their attendance, Result, assignment, profile, timetable etc. Our project mainly helps to an organization in which teacher and students are involved. We also set authorities of the system that which person has which authority. Like just teacher have the authority to add marks of the students. Similarly an admin have the authority that he can view, update, modify and delete events. The third and main person of our system is Admin he have the authority to view, update, modify event and he can also add and delete other users of the system like student and teacher. All the rules are set and all the configurations are set in the project. Our system is very powerful and It Is very flexible for every environment.

7.2. Achievements and Improvements

Achievements from this project are very demandable for the professionals for getting a good experience. We also get good experience from this project. Now we know our abilities that how much we are capable for a project. How to find out errors from the system and how to overcome these errors from the system. We also improve our system through the proper testing stage like unit testing, data flow testing, use case testing and further testing to improve our system. We observe that how to make a system more capable for an organization. We improve all the errors from the system. We also improve our skills about programming, testing and also documentations skills.

7.3. Critical Review:

Our project is a very good project but we can also add the fee details. Fee submission process is also very hectic process. If the module of fees is added to this project this will facilitate the student and accounts managers. Students face many problems related to fees like some students submit the fees but they are in the fee defaulter list. They have to prove their self by providing the receipt of that fee submission.

All these issues can be resolved by adding the fee module. We will add the fee module. We are looking forward to it.

7.4. Lessons Learnt

The lesson we learnt from this project is that the relation between student and teacher, teacher and admin is very important. In many colleges students have to face many problems due to the misunderstandings. Sometimes student face problems related to the assignment submission, view and memorize timetable, view marks. All these processes can be made easy for the students by this project. We learnt that how to develop a strong and complete system that includes all the activities like handling records maintaining records of students and teachers.

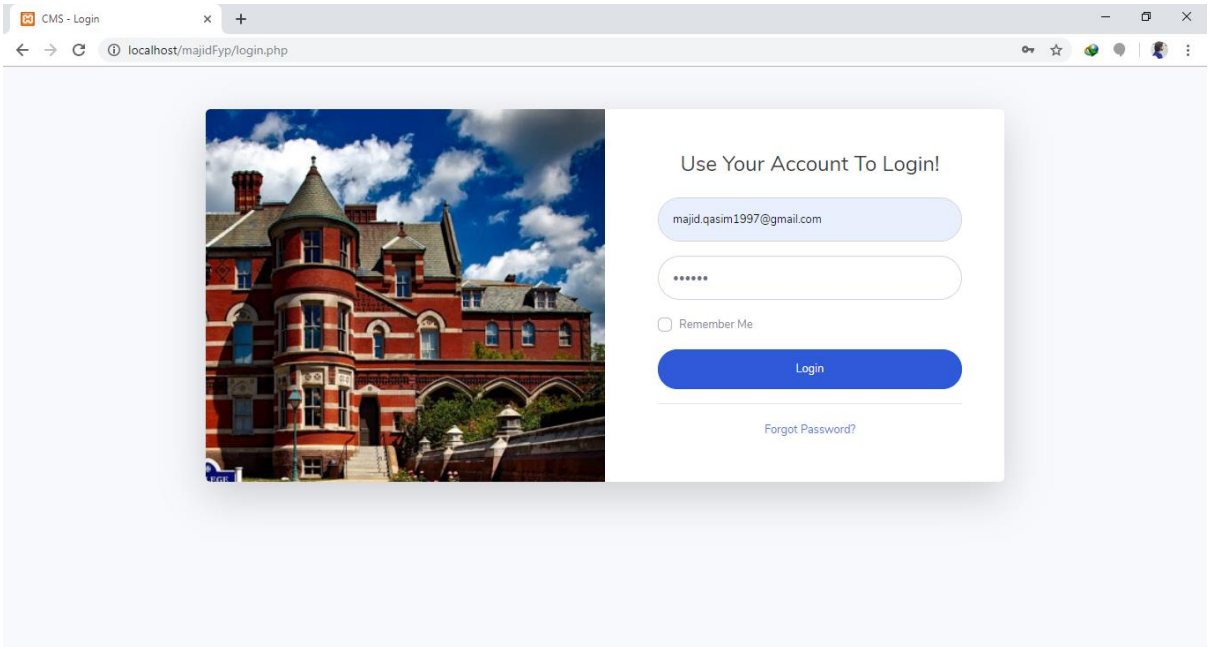
7.5. Future Enhancements/Recommendations

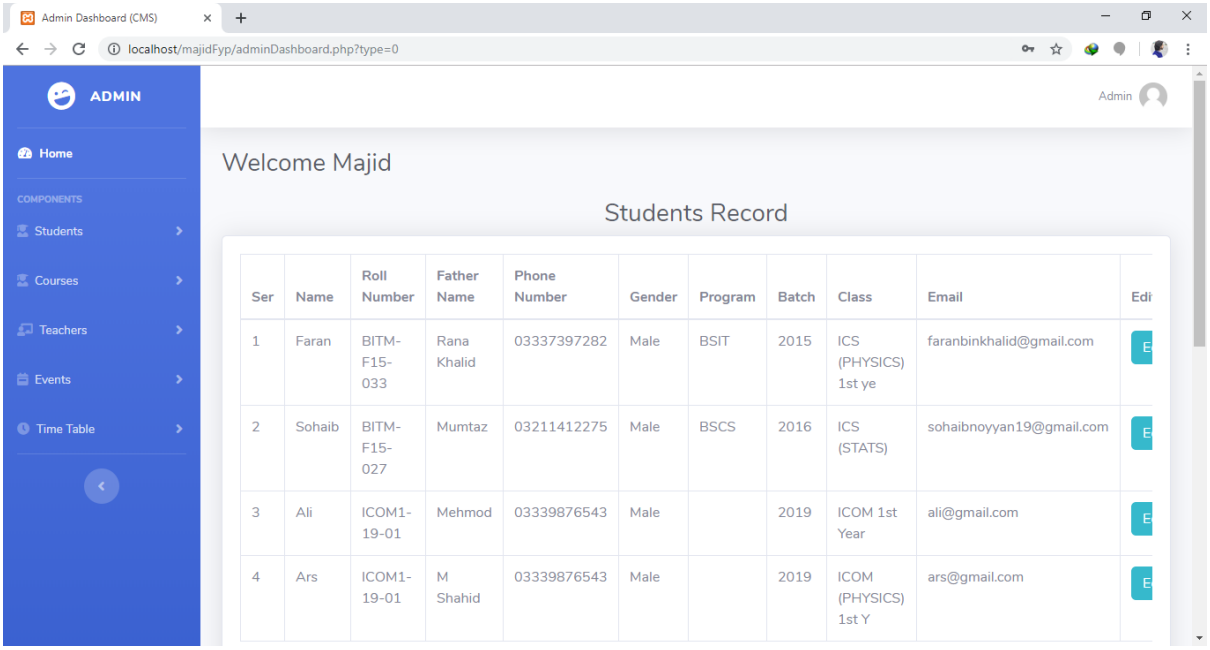
Our project is a very good project but we can also add the fee details. Fee submission process is also very hectic process. If the module of fees is added to this project this will facilitate the student and accounts managers. Students face many problems related to fees like some students submit the fees but they are in the fee defaulter list. They have to prove their self by providing the receipt of that fee submission.

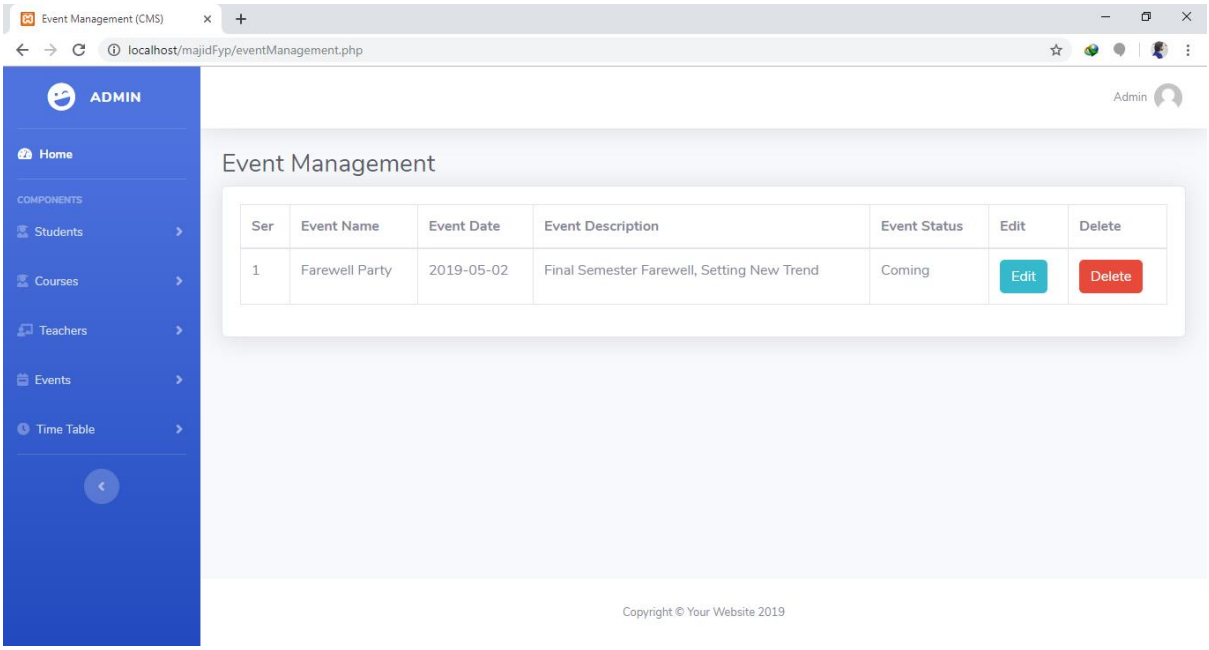
All these issues can be resolved by adding the fee module. We will add the fee module. We are looking forward to it.

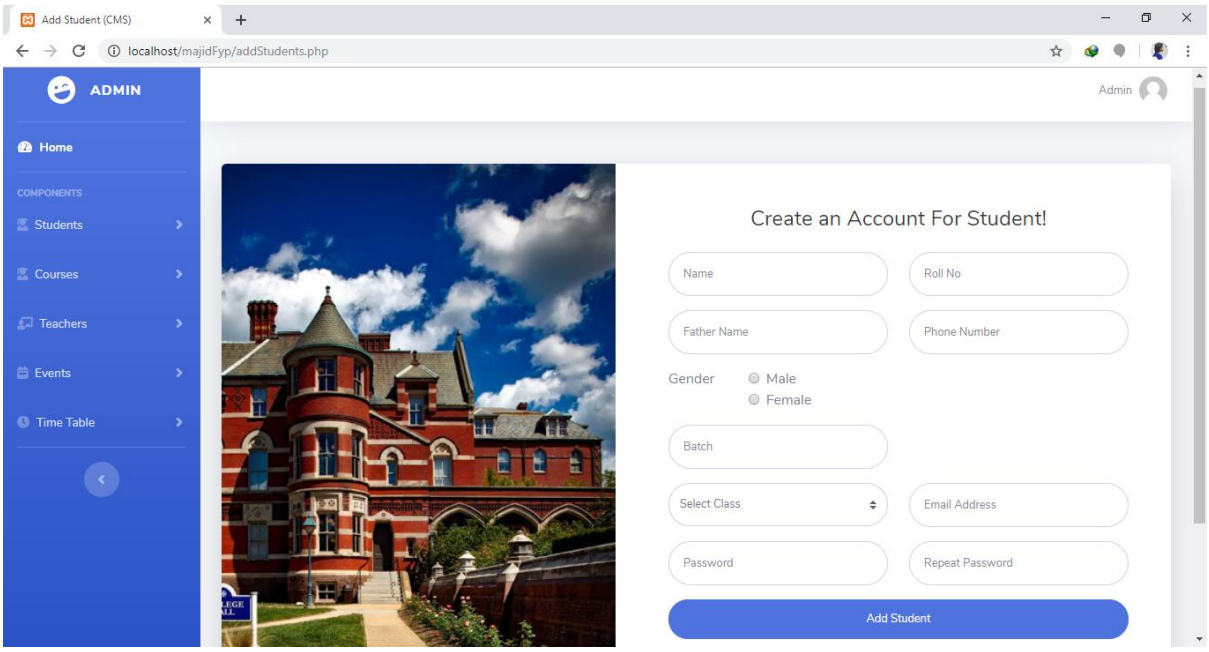
Appendices

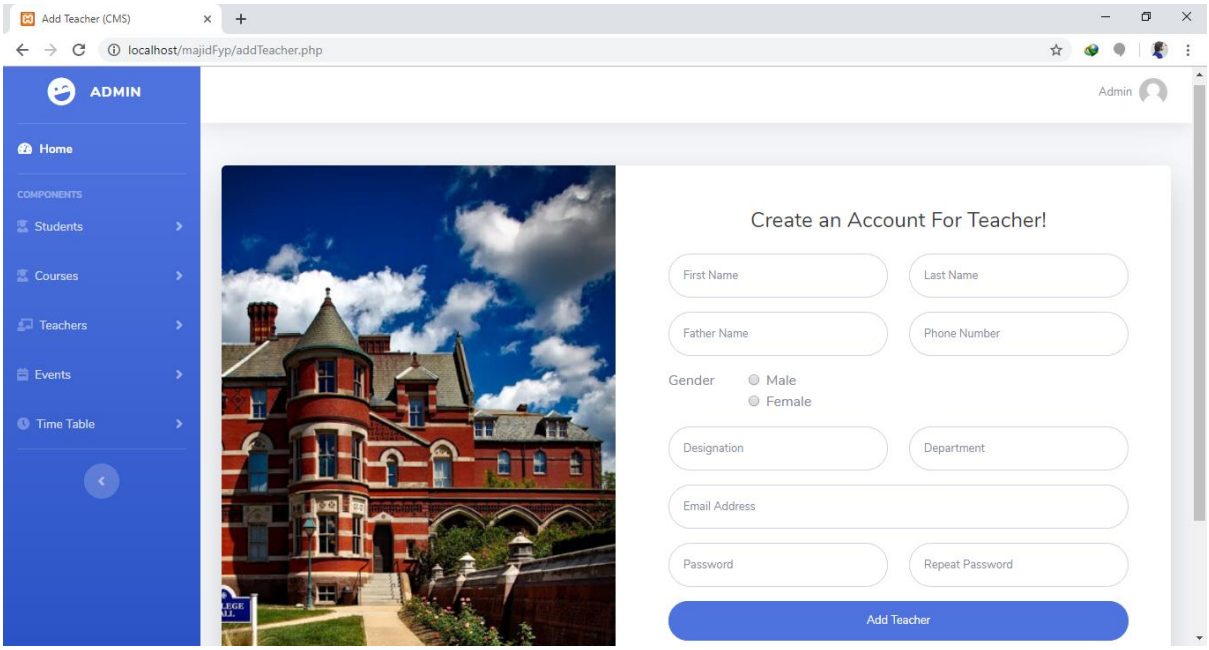
Appendix A: User Manual

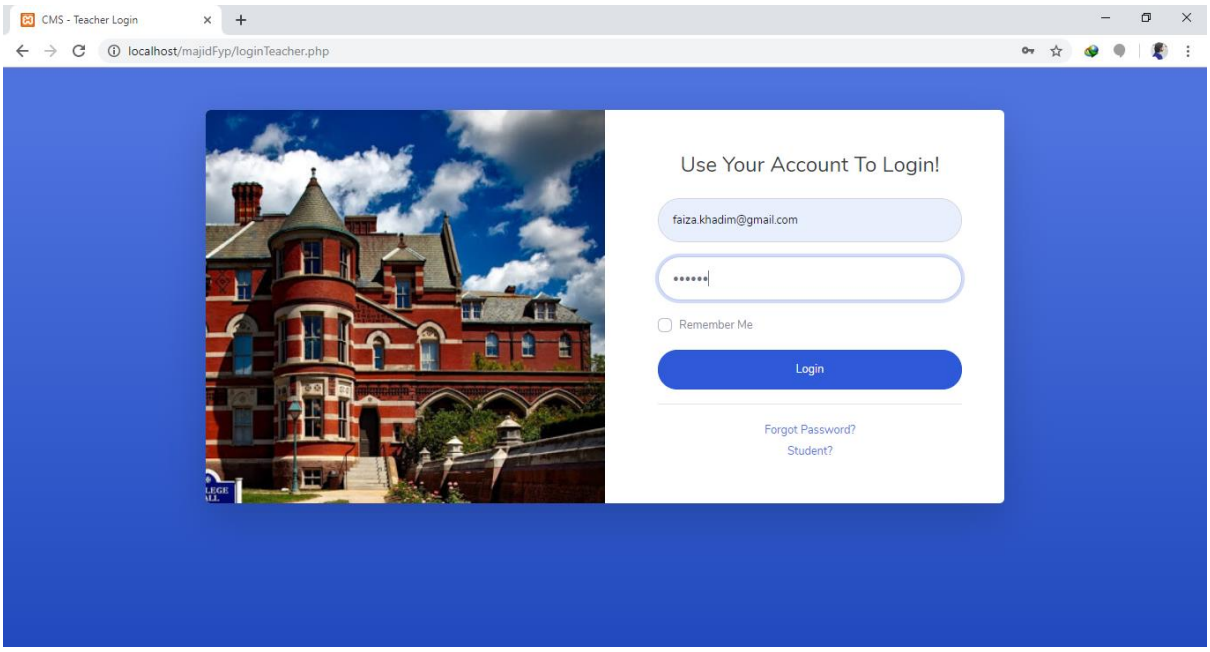


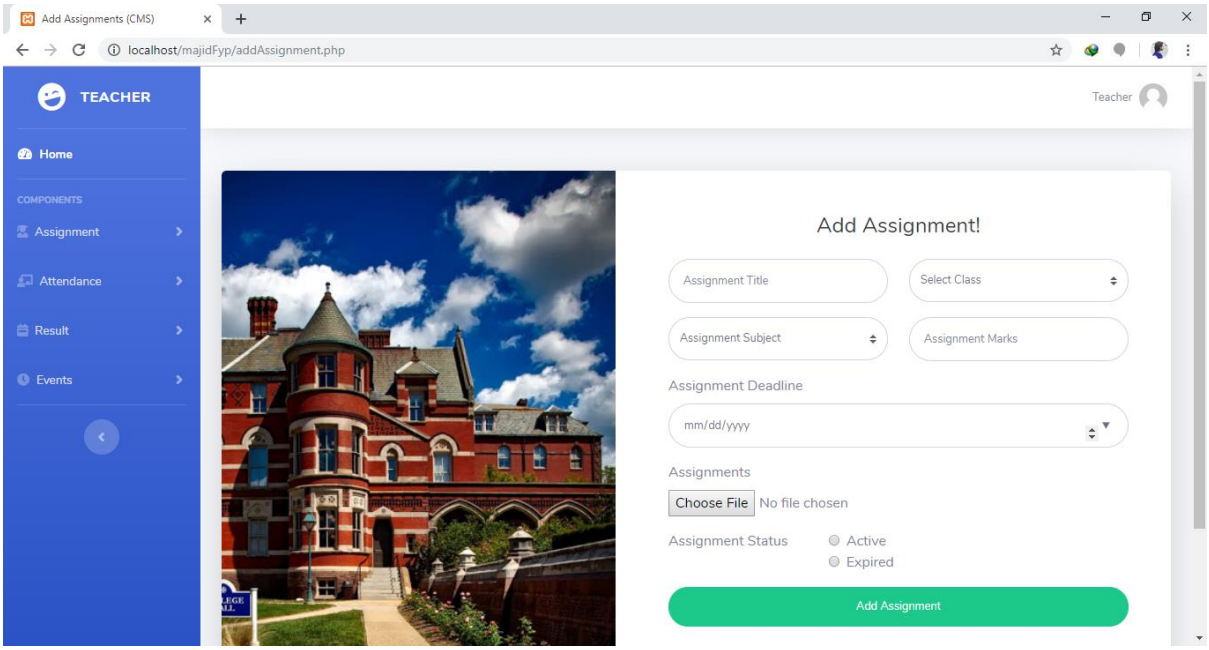


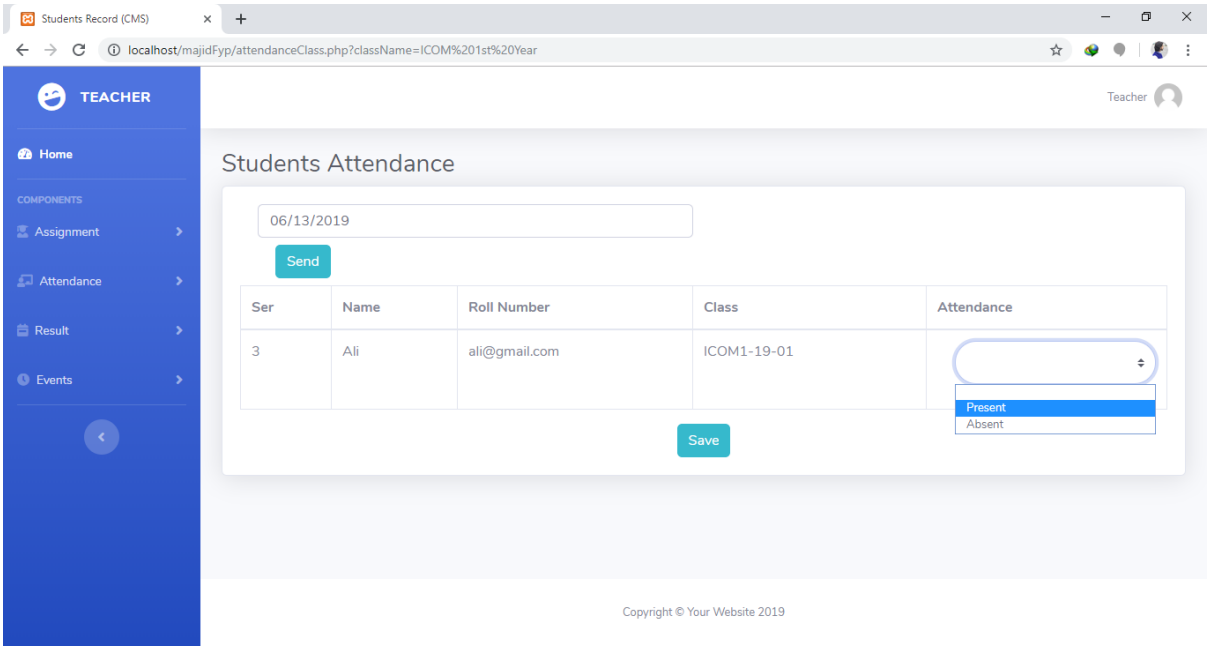

Event Management (CMS)
localhost/majidFyp/eventManagement.php
ADMIN
Home
COMPONENTS
Students
Courses
Teachers
Events
Time Table
Admin
Event Management
Ser
Event Name
Event Date
Event Description
Event Status
Edit
Delete
1
Farewell Party
2019-05-02
Final Semester Farewell, Setting New Trend
Coming
Edit
Delete
Copyright © Your Website 2019

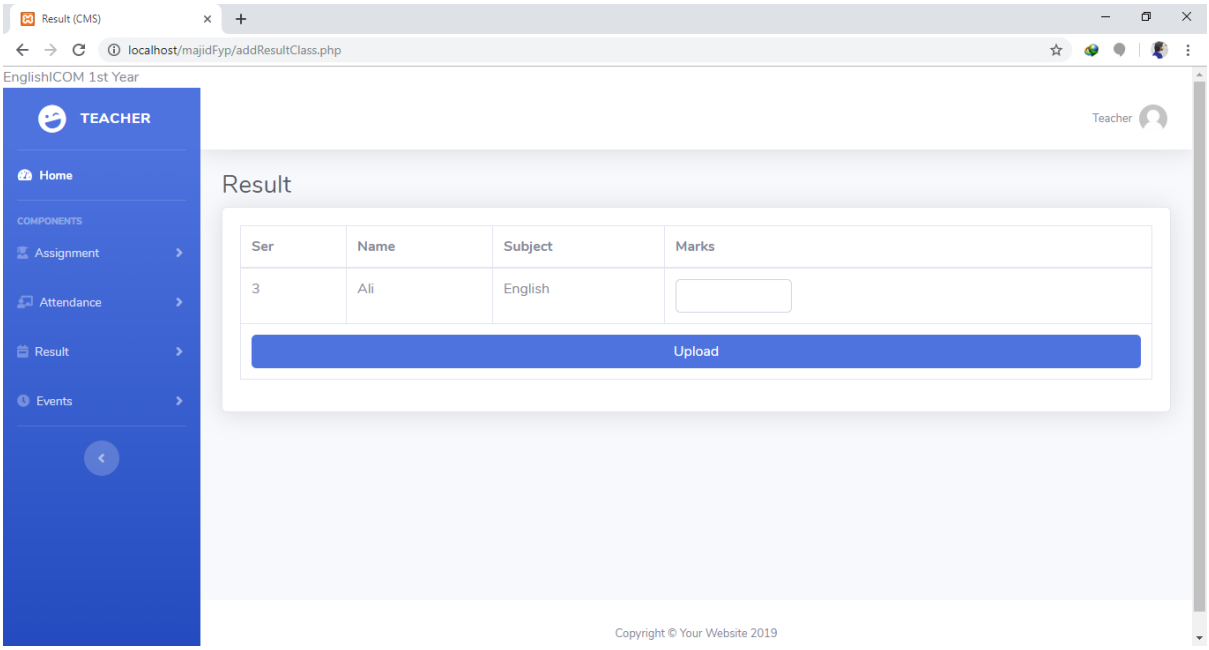

Add Student (CMS)
localhost/majidFyp/addStudents.php
ADMIN
Home
COMPONENTS
Students
Courses
Teachers
Events
Time Table
Admin
Create an Account For Student!
Name
Roll No
Father Name
Phone Number
Gender
Male
Female
Batch
Select Class
Email Address
Password
Repeat Password
Add Student

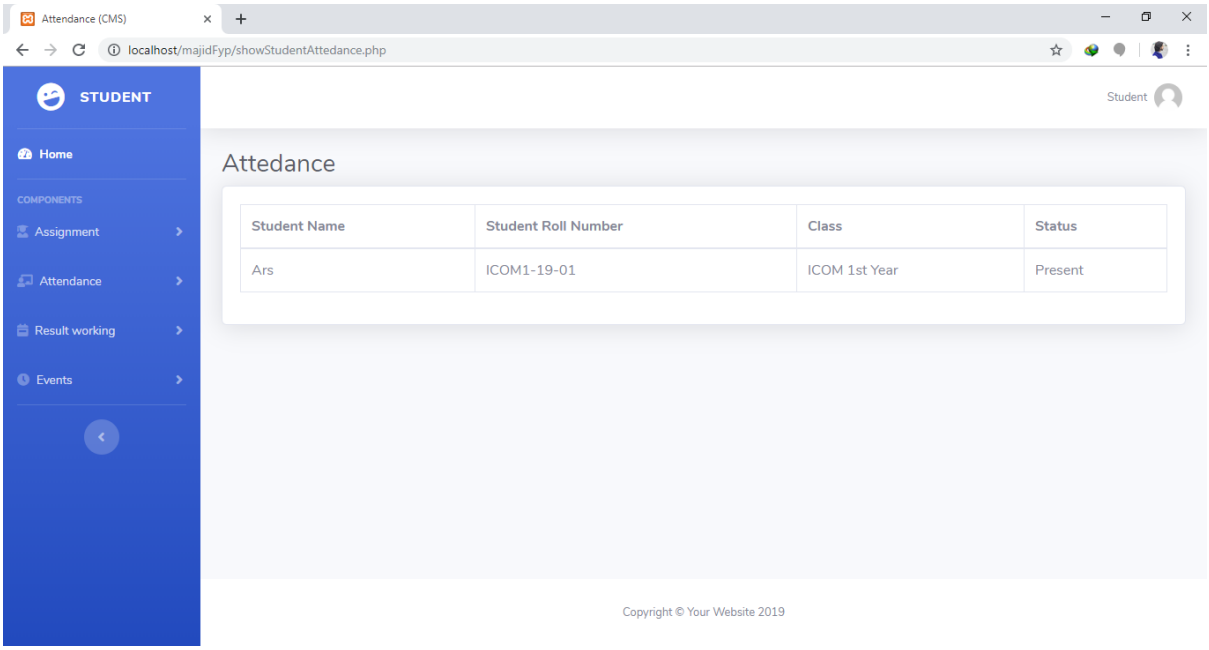

Add Teacher (CMS)
localhost/majidFyp/addTeacher.php
ADMIN
Admin
Home
COMPONENTS
Students
Courses
Teachers
Events
Time Table
Create an Account For Teacher!
First Name
Last Name
Father Name
Phone Number
Gender
Male
Female
Designation
Department
Email Address
Password
Repeat Password
Add Teacher

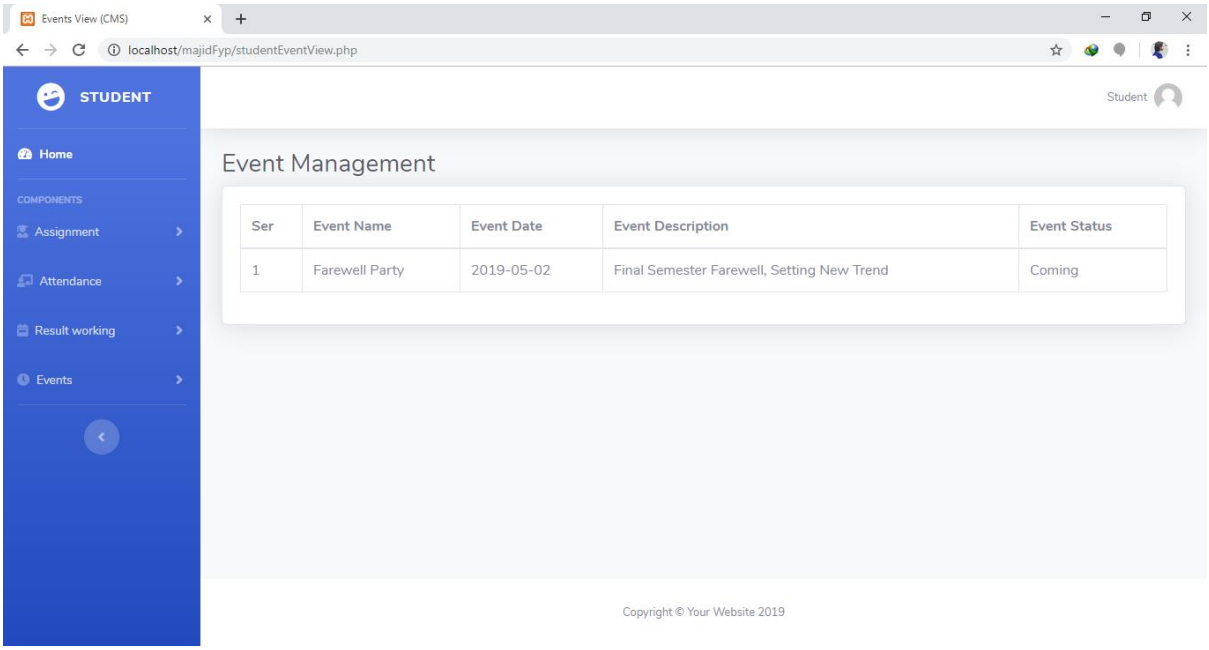

CMS - Teacher Login
localhost/majidFyp/loginTeacher.php
Use Your Account To Login!
faiza.khadim@gmail.com
Remember Me
Login
Forgot Password?
Student?


Add Assignments (CMS)
localhost/majidFyp/addAssignment.php
TEACHER
Home
COMPONENTS
Assignment
Attendance
Result
Events
Teacher
Add Assignment!
Assignment Title
Select Class
Assignment Subject
Assignment Marks
Assignment Deadline
mm/dd/yyyy
Assignments
Choose File
No file chosen
Assignment Status
Active
Expired
Add Assignment


Students Record (CMS)
localhost/majidFyp/attendanceClass.php?className=ICOM%201st%20Year
TEACHER
Home
COMPONENTS
Assignment
Attendance
Result
Events
Teacher
Students Attendance
06/13/2019
Send
Ser
Name
Roll Number
Class
Attendance
3
Ali
ali@gmail.com
ICOM1-19-01
Present
Absent
Save
Copyright © Your Website 2019


Result (CMS)
localhost/majidFyp/addResultClass.php
EnglishICOM 1st Year
TEACHER
Home
COMPONENTS
Assignment
Attendance
Result
Events
Teacher
Result
Ser
Name
Subject
Marks
3
Ali
English
Upload
Copyright © Your Website 2019


Attendance (CMS)
localhost/majidFyp/showStudentAttedance.php
STUDENT
Home
COMPONENTS
Assignment
Attendance
Result working
Events
Student
Attedance
Student Name
Student Roll Number
Class
Status
Ars
ICOM1-19-01
ICOM 1st Year
Present
Copyright © Your Website 2019


Events View (CMS)
localhost/majidFyp/studentEventView.php
STUDENT
Home
COMPONENTS
Assignment
Attendance
Result working
Events
Student
Event Management
Ser
Event Name
Event Date
Event Description
Event Status
1
Farewell Party
2019-05-02
Final Semester Farewell, Setting New Trend
Coming
Copyright © Your Website 2019

Reference and Bibliography

Reference and Bibliography

[1] http://nevonprojects.com/web-based-project-ideas-topics/

[2] SRS https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database.

[3] https://www.studymode.com/essays/Education-Management-System-Software-Requirements-Specification-1514575.html