

Intranet Based Campus Management system

B.Sc. Software Engineering



Zoia Ismail, 13-SE-34

Kinza Naseer, 13-SE-13

Marium Shakeel, 13-SE-50

Supervisor:

Engr. Afzal Ahmed

Department of Software Engineering

Mirpur University of Science and Technology (MUST)

A project report submitted in partial fulfillment of the requirements for the B.Sc. Software Engineering

Declaration

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed in this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Abstract

This project is going to develop an Intranet College Management System (CMS) that is of significance to either a collage or a school. This system may be used for monitoring attendance for the college. Students as well as staffs logging in may also access or can be search any of the information regarding college. Attendance of the staff and students as well as marks of the students will be updated by staff. This system (CMS) is being developed for Asian college to maintain and facilitate easy access to information. For this the users must be registered with the system after which they can access as well as modify data as per the permissions given to them. CMS is an intranet based application that aims at providing information to all the levels of management within an organization. This system can be used as a knowledge/information management system for the college. For a registered student/staff (technical/Non-technical) can access the system to either upload or download some information from the database.

Acknowledgement

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Table of Contents

Declaration.....	i
Abstract.....	ii
Acknowledgement	iii
Chapter 1 Introduction	1
1.1 Goals	1
1.2 Motivation.....	1
1.3 Method	2
1.4 Overview	2
1.5 Report overview	2
Chapter 2 Background and Problem Statement	3
2.1 Introduction.....	3
2.2 Literature Review.....	3
2.3 Problem Statement	3
Chapter 3 Project Management	5
3.1 Approach.....	5
3.2 Initial Project Plan.....	5
3.3 Problems and Changes to the Plan.....	6
3.4 Final Project Record	6
Chapter 4 Analysis.....	7
4.1 Problem Modelling	7
4.1.1 Administrator Usecase	7
4.1.2 Teacher Usecase.....	12
4.1.3 Student Usecase	16
4.2 Functional Requirements	20
4.2.1 Administrative staff.....	20
4.2.2 Teacher.....	21
4.2.3 Student	22
2.1.5 Applicant should enter his\her tel(res), office and mobile.	23
4.2.4 Parent	23
4.3 Non-Functional Requirements	24

4.3.1 Performance:	24
4.3.1 Availability:	26
4.3.2 Maintainability:	26
4.3.3 Reliability:	27
4.3.4 Security:	27
4.3.5 Portability	27
Chapter 5 Product Design	28
5.1 Introduction	28
5.2 Product Features	28
5.2.1 Administrative staff	28
5.2.2 Student	28
5.2.3 Teacher	28
5.2.4 Parents	28
5.2.5 Reports	29
5.3 User Interface	29
5.4 Non-Functional Requirements	31
5.4.1 Performance Requirements	31
5.4.2 Safety Requirements	32
5.4.3 Security Requirements	32
5.5 Data Storage	33
5.6 Design Verification	35
Chapter 6 Software Design	37
6.1 Introduction	37
6.2 High Level Design	37
6.3 Activity Diagram	38
Chapter 7 Implementation	39
7.1 Introduction	39
7.2 Coding	39
7.3 Verification	39
7.4 Validation	41
Chapter 8 Discussion and Conclusion	42
8.1 Solution Review	42
8.2 Project Review	42

8.3 Key Skills.....	42
8.4 Future Work.....	42
8.5 Conclusion	42
References	44

List of figures

Figure 3.1 Initial Project plan	5
Figure 3.2 Final Project Plan part 1	6
Figure 3.3 Final Project Plan part 2	6
Figure 4.1 Use Case For Admin.....	7
Figure 4.2 Use Cases for Teacher	12
Figure 4.3 Use Cases for Student.....	16
Figure 5.1 Home Screen	29
Figure 5.2 Login Screen.....	30
Figure 5.3 Admin Account	30
Figure 5.4 Teacher Account.....	31
Figure 5.5 Student Account	31
Figure 5.6 Data Storage View.....	34
Figure 5.7 Sequence Diagram.....	35
Figure 5.8 Create Account Sequence Diagram.....	36
<i>Figure 6.1 Class Diagram for the project.....</i>	<i>37</i>
Figure 6.2 Activity Diagram	38

List of Table

Table 4.1 Use case login	8
Table 4.2 Use Case Provide login.....	8
Table 4.3 Use Case Maintain record.....	9
Table 4.4 Use Case generate school leaving certificate.....	9
Table 4.5 Use Case Maintain Timetable.....	10
Table 4.6 Use Case Generate Fee Challan.....	10
Table 4.7 Use Case Maintain Fee Record.....	11
Table 4.8 Use Case for teacher login	13
Table 4.9 Use Case for View Teacher Profile	13
Table 4.10 Use Case View Calendar	13
Table 4.11 Use Case View Timetable.....	14
Table 4.12 Use Case Maintain Lectures	14
Table 4.13 Maintain results.....	15
Table 4.14 Use Case Maintain Attendance	15
Table 4.15 Use Case for Online Admission.....	17
Table 4.16 Use Case for Student Login	17
Table 4.17 Use Case of Viewing Timetable for Student	18
Table 4.18 Use Case for Download Lectures.....	18
Table 4.19 Use Case View Calendar	19
Table 4.20 Use Case Download mark sheet	19
Table 4.21 Initial requirements for Administrator	20
Table 4.22 Initial requirements for teacher.....	21
Table 4.23 Initial Requirements for Student.....	22
Table 4.24 Parent Initial Requirements.....	23
Table 4.25 UI loading time as non-functional requirement	24
Table 4.26 Query Response time as Non-Functional Requirement	24
Table 4.27 Login Verification as Non-Functional Requirement	25
Table 4.28 System dependability as Non-Functional Requirement.....	25
Table 4.29 System Availability as Non-Functional Requirement	26
Table 4.30 Application extendibility as Nonfunctional requirement.....	26
Table 4.31 Reliability as Non-Functional Requirement.....	27
Table 4.32 Comm. Security as Non-Functional Requirement	27
Table 4.33 Portability as Non-Functional Requirement	27
Table 7.1 Test Case Login Screen	39
Table 7.2 Test Case Login Failed	40
Table 7.3 Online Admission	40
Table 7.4 Test Case Registration	40
Table 7.5 Test Case Delete Calendar	40
Table 7.6 Test Case Fee Challan Generation.....	41
Table 7.7 Test Case Logout	41
Table 7.8 Test Case Logout Fail	41

Chapter 1 Introduction

Intranet based College Management System (CMS) gives an environment where information and data about student and staff is controlled effectively. The main goal of the system is to automate the process carried out in the organization with improved performance. The system intranet based CMS can be used to manage the data of all type of educational institutes. The main users of the system are:

1. Administrative staff
2. Teachers
3. Students
4. Parents

1.1 Goals

College Management System is being designed for the students, parents, and teacher in college. The main objectives of the College Management System project are following:

- Easy searching of student and teacher information.
- Decrease overheads.
- Minimization redundancy.
- Make data more secure.
- Improve report generation process.
- To provide fast accurate and consistent response.
- Proposed system is according to the customer demand.

In this project, we automated all the manual work of college i.e. admission, attendance, results and other activities that were done traditionally.

1.2 Motivation

Traditional method of college management system was hectic and time consuming and above all there was a lot of inconsistency in records. Our project solved these problems by eliminating inconsistencies and saving precious time of administration. To find some old record about anything required to go through a lot of files.

- The software facilitates the administrators to know the present status of a student.
- The software contains the information such as student personal data, student fees details, results etc. whereas in old system the information was sparse.

- Generating the reports of student's fee as well as result details....
- Hence we conclude that the present system would help the user by saving time and effort by reducing the processing time and volume of errors.
- The efficiency of the work done would be improved and work satisfaction on the part of the employees would be high.
- The customer satisfaction would be definitely higher when compared to the old manual system

1.3 Method

We started our project by preliminary investigations. We gathered functional requirements from Asian College of Sciences and prepared SRS. Then we moved towards development phase.

1.4 Overview

Our application is web-based and will be accessible to administrative staff, teachers and students through intranet.

1.5 Report overview

This thesis has introduced a system that allows a reader to explore the initial concepts of the system. Report consists of eight chapters.

Chapter 1 reviews about the introduction and basic motives of the system. It is the overview the whole system and about the factors involving in developing the system. Main ideas and abstraction of the system is carried out in this chapter.

Chapter 2 reviews the background of the project. Problem statements which we found to develop this system. This section will review literatures that will provide the understanding and explanation of critical success factors in project management and discussion system. The literature review will include Project success factors, Critical success factors and social interaction among the users.

Chapter 3 reviews the plan through which the project is carried out. Methods and approach used to develop the project are discussed in the section. Gantt charts are made to show the work plan in which the system is developed.

Chapter 4 reviews the requirements of the system. Functional and non-functional requirements are discussed. Use cases for different users are discussed and their functions are discussed.

Chapter 5 and 6 reviews the design and architecture of the system. Features that the product have are discussed and explored in the section. User interfaces are shown in the section. System architectures and designs are designed in this section.

Chapter 7 reviews the coding methods and techniques and testing of the product. Test cases are designed to check the product. Chapter 8 reviews the overall system and different key point observed in developing the system. References are discussed using which we got help.

Chapter 2 Background and Problem Statement

2.1 Introduction

The existing system which is being used in college is traditional process is a complete manual process. Now-a-days, education is playing very significant role in the society. Day-by-day, the percentage of illiterates are decreasing and the percentage of literates is increasing. Education will change the society in all the aspects and everyone wants to study higher professional degrees. Admissions are increasing day by day so there by ratio of establishment new colleges and schools are also increasing. But the actual challenge is starting from now. Most of the schools and colleges are maintain student information in records. When the number of records increased, it is difficult to maintain the information of each student in the old manual system. Maintaining the records manually leads to error prone and required more man power and it consumes more time for processing the records.

2.2 Literature Review

Three attributes have developed for CMS:

(1) A CMS takes after the standards of incorporated application frameworks, which, for example, incorporate a single purpose of information entry, a single database and UI, the ongoing access to data and in addition the help of cross-functional procedures. (2) Compared to academic or college IS made as individual programming, CMS are particularly composed as standard programming, which is modularized and customized. If needed singular necessities can be met by extra programming. (3) From an utilitarian perspective, CMS cover all operational and in addition all business knowledge functionalities in advanced education [1]

The model view controller (MVC) is a central design for the partition between UI and business logic. The model-view controller (MVC) design is a technique for arranging an application into three particular segments; the model, the view, and the controller.

MVC is implemented in two different technique; the ASP.Net structure and Java server pages (JSP) structure. Implementing the ASP.Net structure is less demanding than JSP system to accomplish the MVC significant idea [2]

2.3 Problem Statement

After visiting some schools and colleges it was observed that most of the schools have a manual system. The information about the student, faculty and accounts is maintained in the files. Since records are stored in multiple files so changes are difficult to make. The chances of inconsistencies are high. In account management accountants of colleges/schools are facing a lot of problems in maintaining ledgers. Attendance is stored in registers and teacher are assigned additional tasks to perform weekly/ monthly check on the attendance

and at the end reports are generated. Parents have to visit college for any information they want to have either it is about securing admission or it is about getting a report card of children. To solve all the above mentioned problems there is most definitely need of an automated system to replace the old manual system.

Chapter 3 Project Management

3.1 Approach

We started our project by finding the problems faced by administrative staff of colleges. We selected ASP.NET framework and MVC development environment because of little bit know how. After exploring more, we build our project CMS. The reason for selecting MVC model was because it provides clean separation of concerns. MVC allows to write less amount of code to build the web applications as lots of components are integrated to provide the flexibility in the program.

3.2 Initial Project Plan

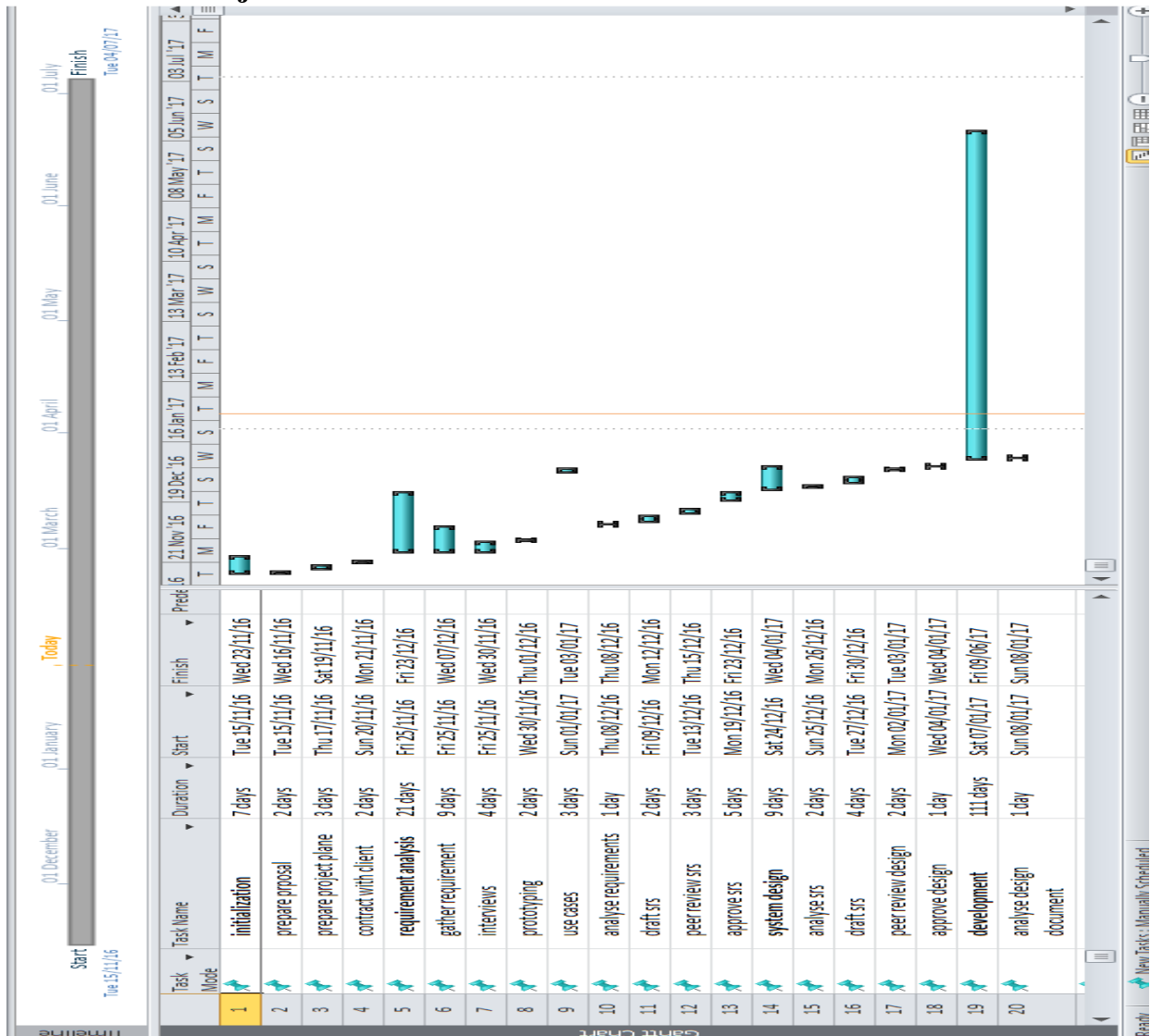


Figure 3.1 Initial Project plan

3.3 Problems and Changes to the Plan

Requirement analysis phase was delayed because of client's unavailability that impacted our initial project plan.

3.4 Final Project Record

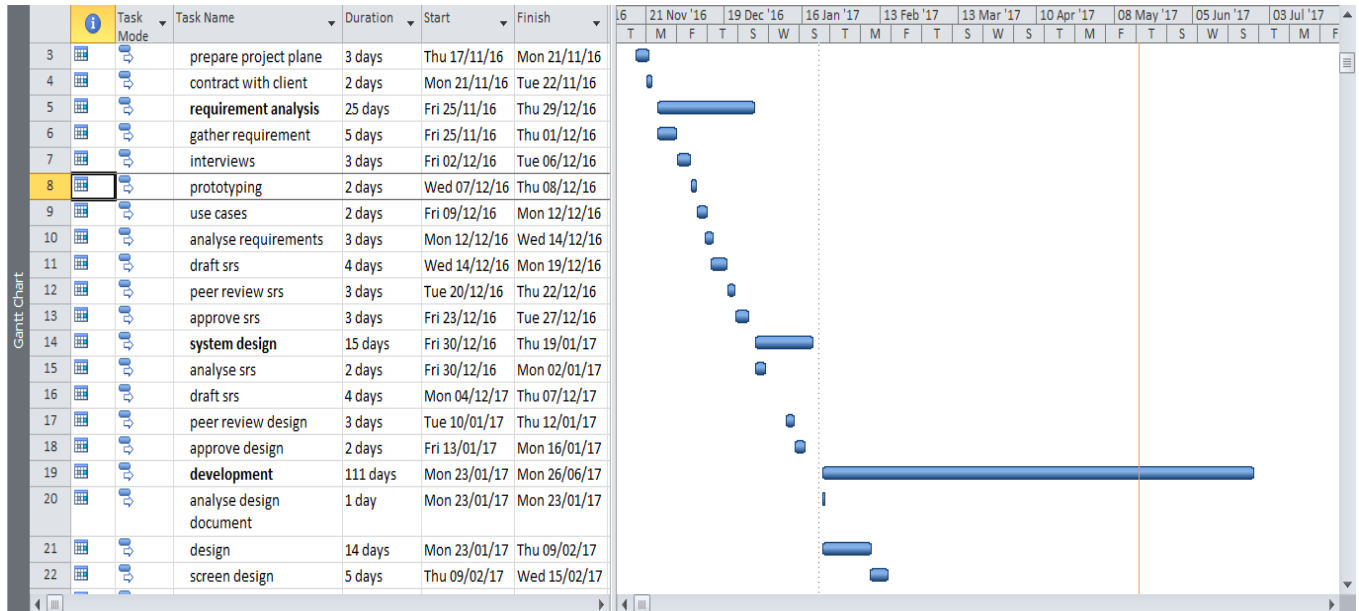


Figure 3.2 Final Project Plan part 1

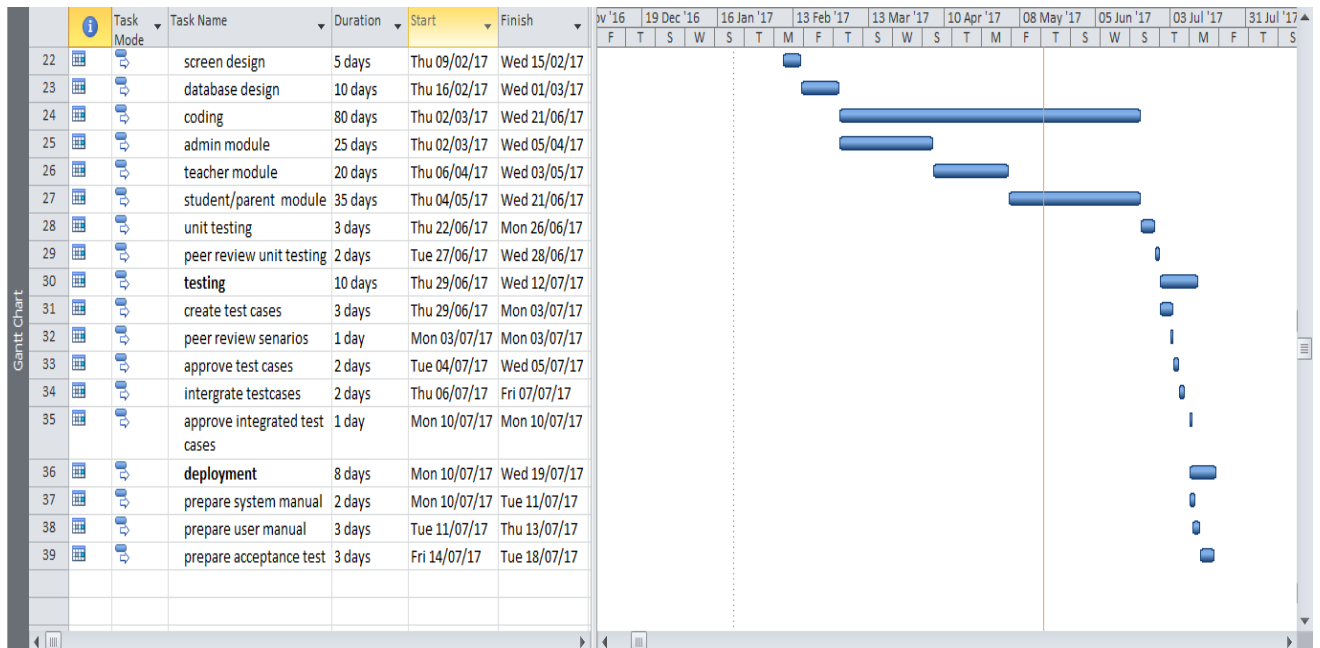


Figure 3.3 Final Project Plan part 2

Chapter 4 Analysis

4.1 Problem Modelling

4.1.1 Administrator Usecase

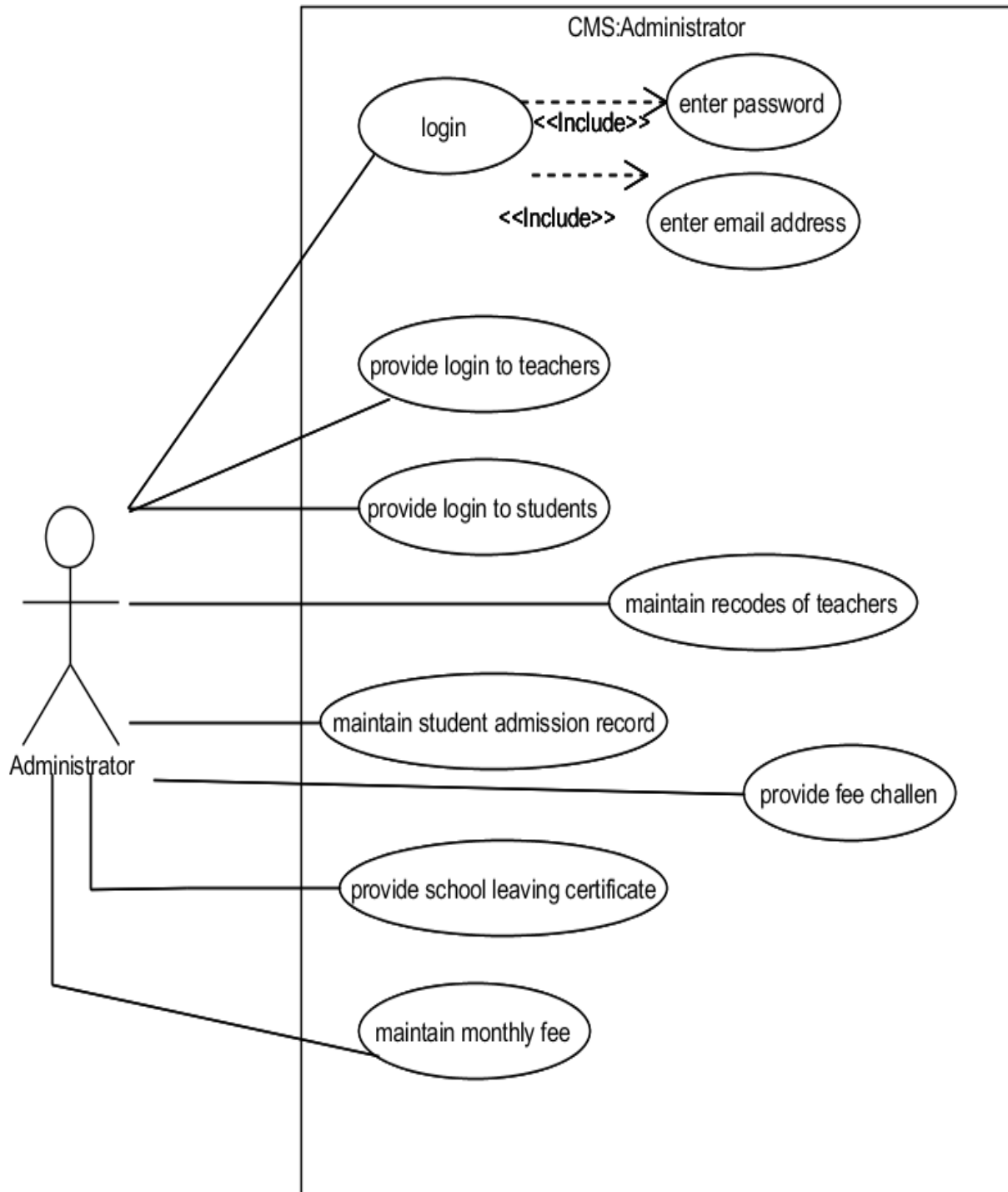


Figure 4.1 Use Case For Admin

Use Cases Description:

Table 4.1 Use case login

Use case ID	UC-01
Use case Name	Login
Actors	Administrative staff
Description	User logs in to the system
Precondition	User should have a valid account
Post condition	If the use case was successful, the user is now logged into the system. If not, the system state is unchanged.
Normal flow	<ol style="list-style-type: none">1. The system requests that the user enter his/her user name and password.2. The user enters his/her user name and password.3. The system validates the entered user name and password and logs the user into the system.

Table 4.2 Use Case Provide login

Use case ID	UC-02
Use case Name	Provide login to teachers/students
Actors	Administrator
Description	Provide user name and password to teachers/students
Precondition	Admin should be logged in
Post condition	Teacher and student credentials saved into database.
Normal flow	<ol style="list-style-type: none">1. Administrator logs in2. Click Teachers/Students>>Add New Teacher/ Add New Student3. Enter username and password4. Click Add button

Table 4.3 Use Case Maintain record

Use case ID	UC-03
Use case Name	Maintain teacher/student record
Actors	Administrator
Description	Administrator maintains complete records of teachers and students
Precondition	<ul style="list-style-type: none"> • Admin should have logged in • Teacher and student record should exist
Post condition	If administrator adds/deletes or updates teacher/student record, then database shall change accordingly
Normal flow	<ol style="list-style-type: none"> 1. Administrator login. 2. Select Teachers/Students button. 3. System shall display complete record of teachers/students. 4. Admin shall maintain (add, update, delete) the record of teacher (Name, CNIC, DOB, Address, Telno, password and email etc) and record of student (Name, Father Name, Admission No, DOB, Address, Telno and Group (S.S.C, H.S.S.C)). 5. Admin shall save record.

Table 4.4 Use Case generate school leaving certificate

Use case ID	UC-04
Use case Name	Generate school leaving certificate
Actors	Administrator
Description	System should generate school leaving certificate
Precondition	<ul style="list-style-type: none"> • Admin should have logged in • System should have enrolled student
Post condition	Status of student should update as Left.
Normal flow	<ol style="list-style-type: none"> 1. Administrator login. 2. Select school leaving certificate 3. Administrator enters admission number of student all the required detail should automatically fill and click on print button

Table 4.5 Use Case Maintain Timetable

Use case ID	UC-05
Use case Name	Maintain Timetable
Actors	Administrator
Description	Administrator shall maintain the timetable.
Precondition	Admin should have to be logged in
Post condition	After adding/editing/deleting timetable, new timetable shall be added and shall be visible in student and teacher account.
Normal Flow	<ol style="list-style-type: none"> 1. Administrator logs in 2. Select timetable button 3. To add new timetable, select new entry button. Enter data and click create 4. To edit or delete select class-subject
Alternative flow	<ol style="list-style-type: none"> 1. Administrator shall select the timetable button. 2. Administrator shall select the delete option. 3. Timetable is deleted 4. Administrator shall select the timetable to be uploaded. 5. System shall prompt user to select a timetable form the directory 6. Administrator shall select the upload option. 7. timetable is updated.

Table 4.6 Use Case Generate Fee Challan

Use case ID	UC-06
Use case Name	Maintain fee challan
Actors	Administrator
Description	Fee challan shall be provided to student
Precondition	System shall have valid fee and student details
Post condition	Fee challan shall be generated
Normal flow	<ol style="list-style-type: none"> 1. Administrator logs in. 2. Select fee challan 3. Administrator enters admission number of student. 4. When admin provide admission no of student Tuition Fee and arrear shall be filled automatically. 5. Select print button 6. Report shall be generated.

Table 4.7 Use Case Maintain Fee Record

Use case ID	UC-07
Use case Name	Maintain monthly fee record
Actors	Administrator
Description	Monthly fee records are maintained
Precondition	Accountant shall be logged in. System should have valid fee and student record.
Post condition	Record properly saved on database
Normal flow	<ol style="list-style-type: none"> 1. login. 2. Select fee management>>All Classes button. 3. Click on open button to view enrolled students 4. Click on Pay section 5. Administrator shall enter date, misc fee 6. Click on save button

Table 4.8 Use Case admission Record

Use case ID	UC-08
Use case Name	Maintain student admission record
Actors	Administrator
Description	Maintain records of those students who applied for new admission
Precondition	Administrator shall be logged in.
Post condition	Students shall be enrolled in new class
Normal flow	<ol style="list-style-type: none"> 1. login. 2. Select admission>>Waiting Admission Requests 3. All newly applied students shall be displayed 4. Click on open file link 5. Enroll student in class

4.1.2 Teacher Usecase

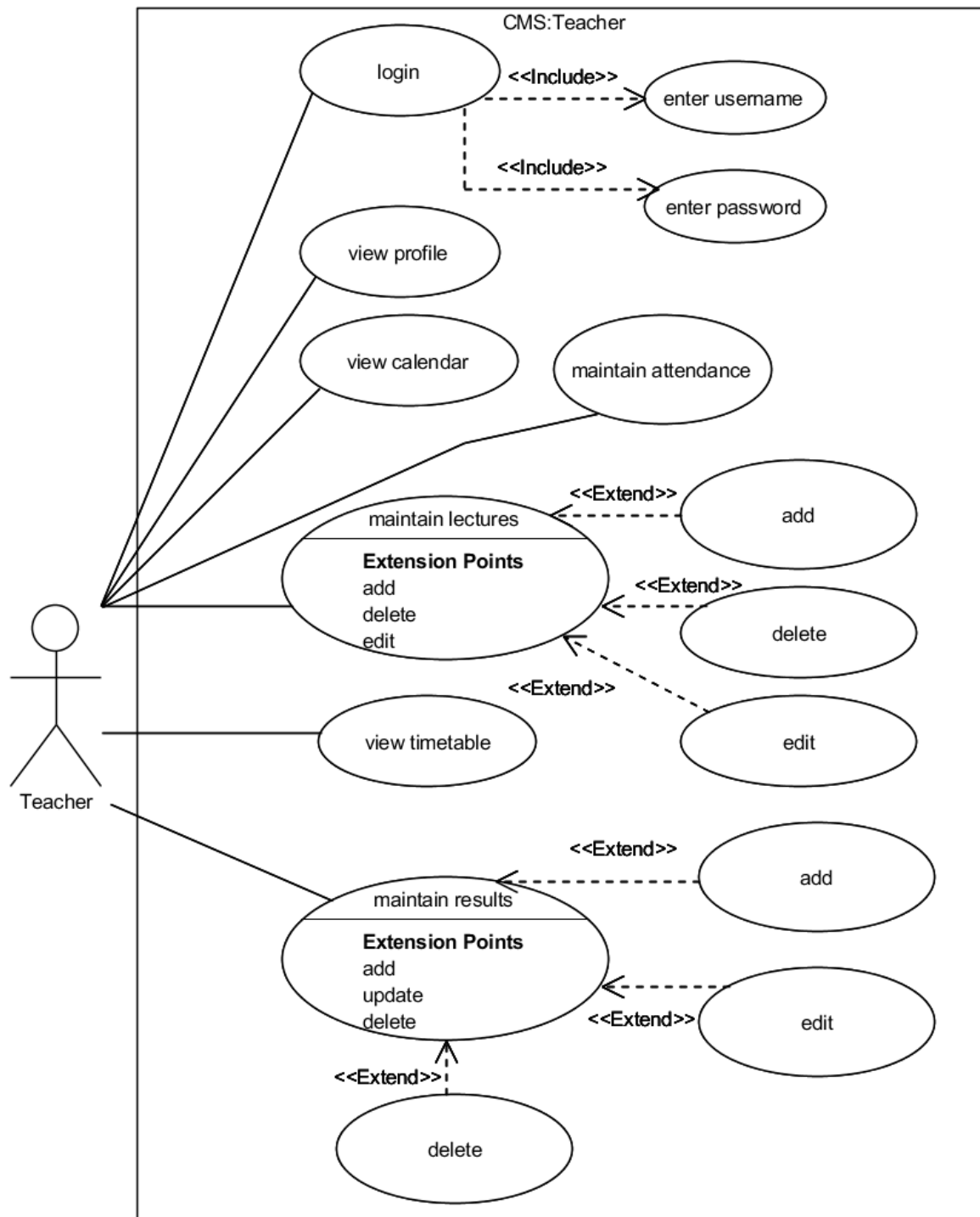


Figure 4.2 Use Cases for Teacher

Table 4.8 Use Case for teacher login

Use case ID	UC-01
Use case Name	Login
Actors	Teacher
Description	Teacher shall be able to login to the system
Precondition	The user should have a valid account.
Post condition	If the use case was successful, the actor is now logged into the system. If not the system state is unchanged.
Normal flow	<ol style="list-style-type: none"> 1. The system requests that the actor enter his/her user email address and password. 2. The actor enters his/her email address and password. 3. The system validates the entered email address and password and logs the actor into the system.

Table 4.9 Use Case for View Teacher Profile

Use case ID	UC-02
Use case Name	View Profile
Actors	Teacher
Description	Teacher shall be able to view and change profile picture or password.
Precondition	should have logged in
Post condition	Profile is successfully updated.
Normal flow	<ol style="list-style-type: none"> 1. Teacher logs in 2. Open profile 3. To change picture clicks, update profile picture button 4. Changes picture and saves. 5. To change password clicks, new password button 6. Click save button.

Table 4.10 Use Case View Calendar

Use case ID	UC-03
Use case Name	View calendar
Actors	Teacher
Description	Teacher shall be able to view all the holidays ahead and exam dates
Precondition	Teacher should be logged in
Post condition	Teacher should be able to view calendar
Normal flow	<ol style="list-style-type: none"> 1. Teacher logs in 2. Clicks calendar button 3. Views calendar

Table 4.11 Use Case View Timetable

Use case ID	UC-04
Use case Name	View timetable
Actors	Teacher
Description	Teacher shall be able to view timetable
Precondition	Teacher should be logged in
Post condition	Teacher should be able to view timetable
Normal flow	<ol style="list-style-type: none"> 1. Teacher logs in 2. Clicks timetable button 3. Selects class 4. Timetable shows up

Table 4.12 Use Case Maintain Lectures

Use case ID	UC-05
Use case Name	Maintain lectures
Actors	Teacher
Description	Teacher shall maintain the lectures.
Precondition	Teacher shall be able to login into the system.
Post condition	After adding, deleting and updating the lectures the lectures information shall be updated in the database as well as in the teacher and student account.
Normal flow	<ol style="list-style-type: none"> 1. Teacher login. 2. Click classes button. 3. Select class and click option button 4. Choose lecture. 5. Choose a subject to see its lectures. 6. Select upload new button. 7. Click add button.
Alternative flows	<ol style="list-style-type: none"> 1. Teacher shall select the lectures. 2. Teacher shall select the delete option. 3. Lecture is deleted 4. Teacher shall select the lecture to be edited. 5. System shall prompt user to select a lectures form the directory 6. Teacher shall select the edit option. 7. Lecture is edited.

Table 4.13 Maintain results

Use case ID	UC-06
Use case Name	Maintain results.
Actors	Teacher
Description	Teacher shall maintain the results.
Precondition	Teacher shall be able to login in to the system.
Post condition	After adding, deleting and updating the results the result information shall be updated in the database as well as in the teacher account and student account.
Normal flow	<ol style="list-style-type: none"> 1. Teacher login. 2. Select result button. 3. Select update result. 4. Choose a subject to update its result. 5. Click save button.
Alternative flow	<ol style="list-style-type: none"> 1. Teacher shall select the result button. 2. Teacher shall select the view option. 3. Click result card. 4. Result is updated.

Table 4.14 Use Case Maintain Attendance

Use case ID	UC-07
Use case Name	Maintain attendance
Actors	Teacher
Description	Teacher shall maintain the attendance of the students.
Precondition	Teacher shall be able to login into the system.
Post condition	After adding, deleting and updating the attendance the attendance information shall be updated in the database as well as in the teacher and student account.
Normal flow	<ol style="list-style-type: none"> 1. Teacher login. 2. Select attendance button. 3. Select class. 4. Choose a subject to see its attendance. 5. Click Take New Attendance. 6. Click save button.
Alternative Flow	<ol style="list-style-type: none"> 1. Teacher shall select the attendance button. 2. Teacher shall select the delete option. 3. Attendance is deleted.

4.1.3 Student Usecase

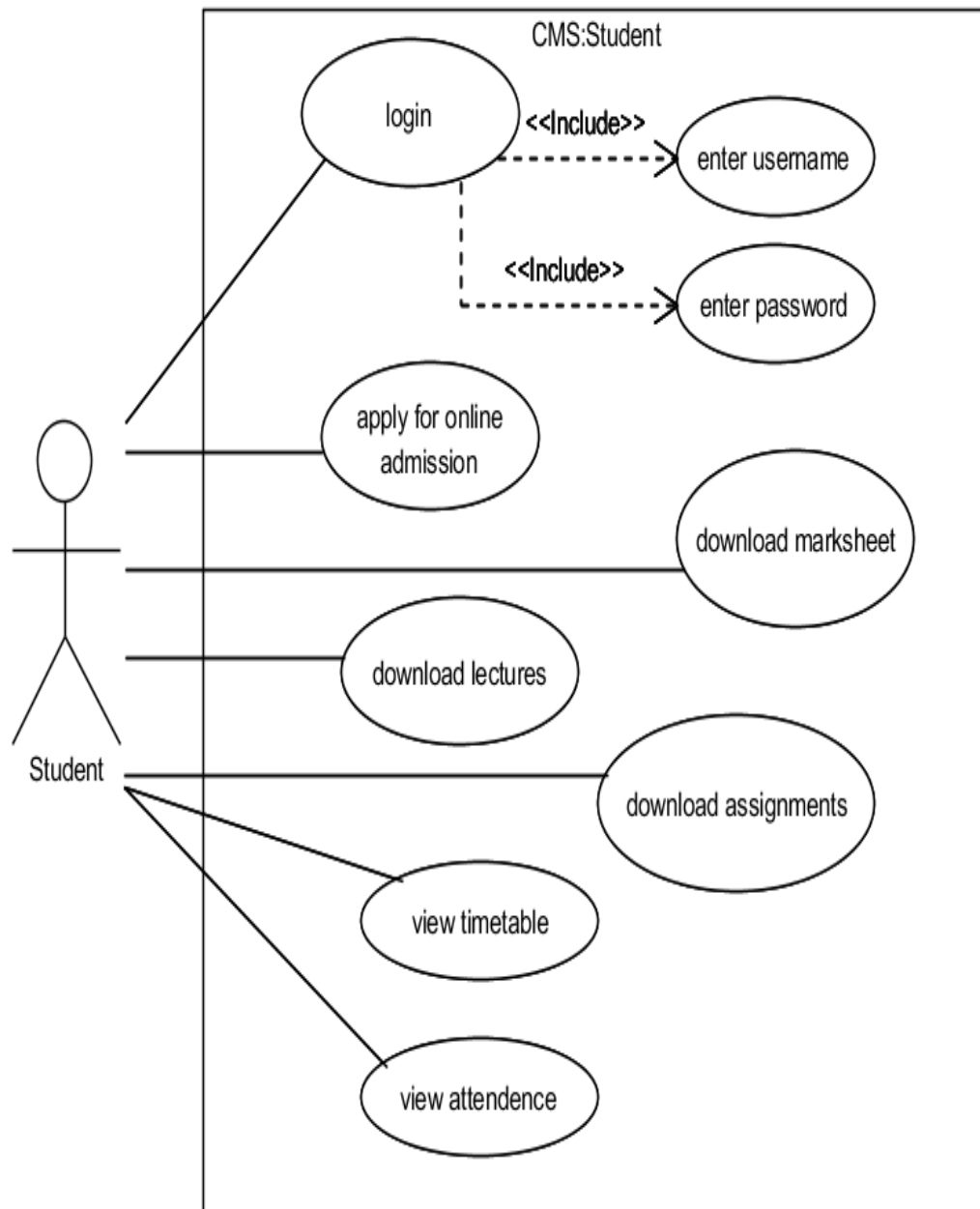


Figure 4.3 Use Cases for Student

Table 4.15 Use Case for Online Admission

Use case ID	UC-01
Use case Name	Apply for online admission
Actors	Student
Description	New Student shall apply for online admission to get admission in the university.
Precondition	Nil
Post condition	Student shall successfully submit the form.
Normal flow	<ol style="list-style-type: none"> 1. Open the university website. 2. Click on apply for online admission 3. Register user 4. Fill the form. 5. Click on submit button.

Table 4.16 Use Case for Student Login

Use case ID	UC-02
Use case Name	Login
Actors	Student
Description	Actor logs in to the system.
Precondition	The user should have a valid account.
Post condition	If the use case was successful, the actor is now logged into the system. If not the system state is unchanged.
Normal flow	<ol style="list-style-type: none"> 1. The system requests that the actor enter his/her email and password. 2. The actor enters his/her email and password. 3. The system validates the entered email and password and logs the actor into the system.

Table 4.17 Use Case of Viewing Timetable for Student

Use case ID	UC-03
Use case Name	View Timetable
Actors	Student
Description	Student shall view the timetable.
Precondition	Teacher should upload the timetable.
Post condition	The student shall be able to view the timetable that is uploaded by the admin.
Normal flow	<ol style="list-style-type: none"> 1. Student login. 2. Click timetable button. 3. System shall display timetable screen according to that class.

Table 4.18 Use Case for Download Lectures

Use case ID	UC-04
Use case Name	Download lectures
Actors	Student
Description	Student shall be able to download lectures
Precondition	Lectures should be logged in
Post condition	Lectures should be downloaded to student's pc
Normal flow	<ol style="list-style-type: none"> 1. Student login 2. Select classes button 3. Select subject 4. Select lectures 5. System shall display screen containing lectures 6. Select lecture 7. Select download button

Table 4.19 Use Case View Calendar

Use case ID	UC-06
Use case Name	View calendar
Actors	Student
Description	Student shall view the calendar
Precondition	<i>Student should have logged in</i>
Post condition	Student shall be able to view the calendar
Normal flow	<ol style="list-style-type: none"> 1. Student login 2. Select calendar button 3. System shall display screen containing calendar

Table 4.20 Use Case Download mark sheet

Use case ID	UC-07
Use case Name	Download mark sheet
Actors	Student
Description	Student shall be able to download mark sheet
Precondition	Results should be uploaded
Post condition	Mark sheet should be downloaded to student's pc
Normal flow	<ol style="list-style-type: none"> 1. Student login 2. Select classes 3. Select result card button 4. System shall display screen containing mark sheet 5. Select download button

4.2 Functional Requirements

4.2.1 Administrative staff

Table 4.21 Initial requirements for Administrator

Table Initial Requirements Para #	Initial Requirements
1.1	Admin shall be able to login.
1.2	Admin should provide login credentials (username and password) to teachers.
1.3	Admin should provide login credentials (username and password) to students.
1.4	Admin should maintain (add, update, delete) the record of teachers/students (Name, CNIC, DOB, Address, Telno, password and email).
1.5	Admin should maintain (add, update, delete) the record of students (Name, Father Name, Admission No, DOB, Address, Telno and Group (S.S.C, H.S.S.C)).
1.6	Admin should maintain (add, update, delete) the calendar of whole year including information (exam dates, government holidays, unscheduled holidays, curricular events, annual function).
1.7	Admin should maintain student monthly record table by entering admission no, month, date, voucher no, class Fee, miscFee, arrear. Total monthly fee should be calculated automatically

1.8	Admin should provide fee challan(report) to students. When admin provide admission no of student TutionFee and arrear should be filled automatically while generating this report.
1.9	Admin should provide leaving certificate (report) to student. When admin enter admission no of student, other fields such as student name, father name, DOB, address, date of admission, class of admission should be filled automatically during generating this report.

4.2.2 Teacher

Table 4.22 Initial requirements for teacher

Table Initial Requirements Para #	Initial Requirements
1.1	Teacher shall be able to login.
1.2	Teacher shall be able to view his\her profile (change password, picture).
1.3	Teacher shall be able to add, update, and delete the attendance. Notifications should be issued for less than 80 percent attendances.
1.4	Teacher shall be able to add, update, and delete the results.
1.5	Teacher shall be able to add, update, delete and view the lectures.
1.6	Teacher shall be able to add, update, delete and view the assignments.
1.7	Teacher shall be able to view timetable.

1.8	Teacher shall be able to view calendar.
2.1	Teacher shall be able to generate reports of results.
2.2	Teacher shall be able to view students' information.
2.1	Teacher shall be able to generate reports of attendances.

4.2.3 Student

Table 4.23 Initial Requirements for Student

Initial Requirements para #.	Initial Requirements
1.1	Students shall be able to login.
1.2	Students shall be able to view his\her profile.
1.3	Students shall be able to view timetable.
1.4	Students shall be able to view lectures.
1.5	Students shall be able to view assignments.
1.6	Students shall be able to view mark sheet.
1.7	Students shall be able to view calendar.
1.8	Students shall be able to view result subject wise.
1.9	Students shall be able to view short attendance.
1.10	Students shall be able to download assignments/lectures.
1.11	Students shall be able to download mark sheet.
2.1	Online Admission 2.1.1 Applicant should enter complete information ((Name, Father name, religion, permanent district, class last attend\passed, name of status of school\college, last attend, class to which admission required. medium).

2.3	<p>2.1.2 Applicant should select group S.S.C\H.S.S.C by checking one of checkbox.</p> <p>For S.S.C options are Sciences Arts and computer science.</p> <p>For H.S.S.C options are pre engr, pre-med, G Sciences (ICS), 1. Com, humanities.</p> <p>2.1.3 Applicant should enter his\her complete permanent address (permanent address\No, sector \village, P.O, Tehsil, District).</p> <p>2.1.4 Applicant should enter his\her complete present address (permanent address\No, sector \village, P.O, Tehsil, District).</p> <p>2.1.5 Applicant should enter his\her tel(res), office and mobile.</p> <p>2.1.6 Applicants name and father name should be filled in capital block letters</p> <p>2.1.7 Applicants date of birth should be written in figures</p> <p>2.1.8 After admission form filled by the student these columns of student table should be automatically filled: s_admission_no,s_name, s_fname,s_address, s_dob, s_classOfAdmissions_dateOfAdmission.</p>
2.4	<p>Student should get fee challan (report).</p> <p>Student should get leaving certificate(report).</p>

4.2.4 Parent

Table 4.24 Parent Initial Requirements

Table Initial Requirements Para #	Initial Requirements
1.1	Parents should login.
1.2	Parents should be able to view attendance of their children.
1.3	Parents should be able to view attendance of their children.

4.3 Non-Functional Requirements

4.3.1 Performance:

Table 4.25 UI loading time as non-functional requirement

User interface loading time	
TAG	Performance. User interface loading time
AMBITION	In every action-response of the system, there are no immediate delays.
SCALE	If the system takes much longer time to display a page or the system gets some strange input, the user should be informed.
METER	Measurements obtained from 20 hours during testing
MUST	No more than 5 seconds 100% of the time
PLAN	No more than 4 seconds 100% of the time
WISH	No more than 3 second 100% of the time

Table 4.26 Query Response time as Non-Functional Requirement

Query Response time	
TAG	Performance. Query Response time
AMBITION	In every action-response of the system, there should no immediate delays.
SCALE	A measure of how well the system utilizes processor capacity, disk space or memory.
METER	Measurements obtained from 100 searches during testing
MUST	No more than 5 seconds 100% of the time
PLAN	No more than 3 seconds 100% of the time
WISH	No more than 2 second 100% of the time

Table 4.27 Login Verification as Non-Functional Requirement

Login verification Time	
TAG	Performance. Login verification time
AMBITION	Checking the fact that the system must perform as what every user expects. So, in every action-response of the system, there are no immediate delays.
SCALE	If the system takes much longer to authenticate the user or user enters wrong username or password, the user should be informed.
METER	Measurements obtained from 20 hours during testing
MUST	No more than 5 seconds 100% of the time
PLAN	No more than 4 seconds 100% of the time
WISH	No more than 3 second 100% of the time

Table 4.28 System dependability as Non-Functional Requirement

System dependability	
TAG	Performance. System Dependability
AMBITION	The fault tolerance of the system.
SCALE	If the system takes much longer time to display a page or the system gets some strange input, the user should be informed.
METER	Measurements obtained from 10 hours of usage during testing
MUST	More than 98% of the time
PLAN	More than 99% of the time
MUST	100% of the time

4.3.1 Availability:

Table 4.29 System Availability as Non-Functional Requirement

System Availability	
TAG	System Availability
AMBITION	The availability of the system when it is used
SCALE	The average system availability (not considering network failing).
METER	Measurements obtained from 10 hours of usage during testing
MUST	More than 98% of the time
PLAN	More than 99% of the time
WISH	100% of the time

4.3.2 Maintainability:

Table 4.30 Application extendibility as Nonfunctional requirement

Application extendibility	
TAG	Maintainability. Application extendibility
AMBITION	In order for future functions to be implemented easily to the application.
SCALE	The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.
METER	None

4.3.3 Reliability:

Table 4.31 Reliability as Non-Functional Requirement

System Reliability	
TAG	System Reliability
AMBITION	The reliability of the system
SCALE	The reliability that the system gives the right result on a search.
METER	Measurements obtained from 100 searches during testing.
MUST	More than 98% of the searches.
PLAN	More than 99% of the searches.
WISH	100% of the searches.

4.3.4 Security:

Table 4.32 Comm. Security as Non-Functional Requirement

Communication Security	
TAG	Communication Security
AMBITION	Security of the communication between the system and applicant
SCALE	The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages.
METER	Attempts to get user-name and password through obtained messages on 1000 log-in session during testing.
MUST	99% of the Communication Messages in the communication of a log-in session should be encrypted.
PLAN	100% of the Communication Messages in the communication of a log-in session should be encrypted.
WISH	100% of the Communication Messages in the communication of a log-in session should be encrypted.

4.3.5 Portability

Table 4.33 Portability as Non-Functional Requirement

Application portability	
TAG	Application portability
AMBITION	The adaptable platform for the application to run on.
SCALE	The application should be portable with window 7 and above
METER	None

Chapter 5 Product Design

5.1 Introduction

We designed our software by first conducting meetings with our client in which we discussed main features of the system. Then we designed SRS based on collected requirements then we moved towards use case diagrams to find out how our end users will interact with CMS. After finding out interactions b/w users and system we made sequence diagram to find out in which sequence users actions will be performed. To illustrate the logical structure of database we developed ERD. It shows the relationships of entity sets stored in a database. Finally, several classes were identified and grouped together in a class diagram that helped us to determine the static relations between them

5.2 Product Features

5.2.1 Administrative staff

- Provides login credentials to students, faculty.
- Maintain student/faculty records (Add, Update, View and delete)
- Time table: admin can be upload a time table.
- Notice Board: In this the administrator, will create the notice board and holiday list and display it in the notice board list menu.
- Course: In this the administrator, will enter all available course details in the college. The course module has options of adding, deleting, editing the data.
- Calendar: In this the administrator, will update a calendar for students. Information
 - about the midterms, practical and final exam dates in this calendar.

5.2.2 Student

- Online admission: Student shall be able to apply for online admission.
- Attendance: In this the Student, will view his/her attendance.
- Results: In this the student, will view his/her results weekly/annually.
- Calendar: In this the student, will view calendar.
- Profile: student could change profile picture.
- Download lectures and assignments uploaded by teacher.

5.2.3 Teacher

- upload Assignments/lectures: Add, Delete assignments.
- Add Attendance: In the attendance module teacher can submit attendance at the end of the month.
- Add results: In the result module teacher can submit monthly result as well as annual
- Calendar: faculty can see the calendar of student's examination date sheet.
- View Student Details: This provides the staff to view the student details.

5.2.4 Parents

Admin give a unique parents id & password for parents. Parents shall be able to view attendance, marks, assignments, exam schedule, and holidays' list. If admin has any

issue of attendance & marks of student, then admin can update notice in student account and parents account. So, parents can see the notice of college regarding to student.

5.2.5 Reports

Above-mentioned data are stored in the back end and can be retrieved as reports with filtering options. The Following are the reports can be taken from this system

1. Student attendance Report
2. Results Report (subject/class wise)
3. Fee challan
4. School leaving certificate

5.3 User Interface



Figure 5.1 Home Screen

Login

Username

Password

Login As:

Student

Let me in.

f

t

G+

yt

i

@

in

Login

All Rights Reserved CMS © 2017 Powered by - Department of Software Engineering - MUST

Figure 5.2 Login Screen

Dashboard

Dashboard

Demo Admin

Admin

Last Access : 7/18/2017 2:00:26 AM

MENU

Profile

Classes

All Subjects

Admissions

Enrollments

Attendance

Results

Parents

Students

Teachers

Admin Account

Name

Demo Admin

Email

demo@admin.com

Update

Figure 5.3 Admin Account

30

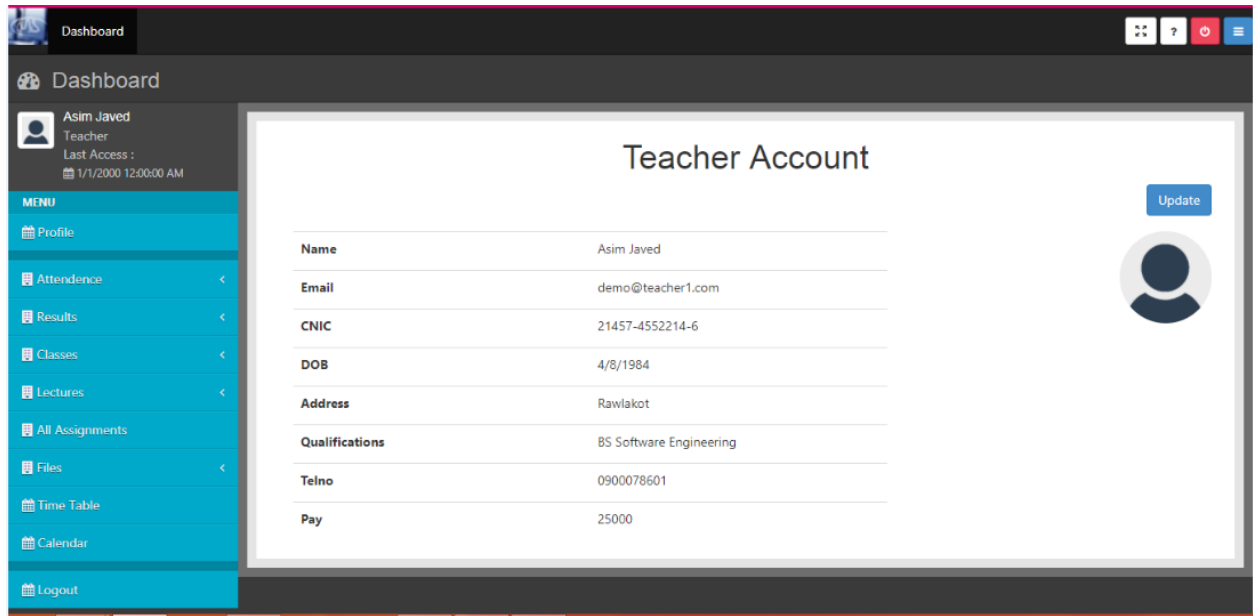


Figure 5.4 Teacher Account

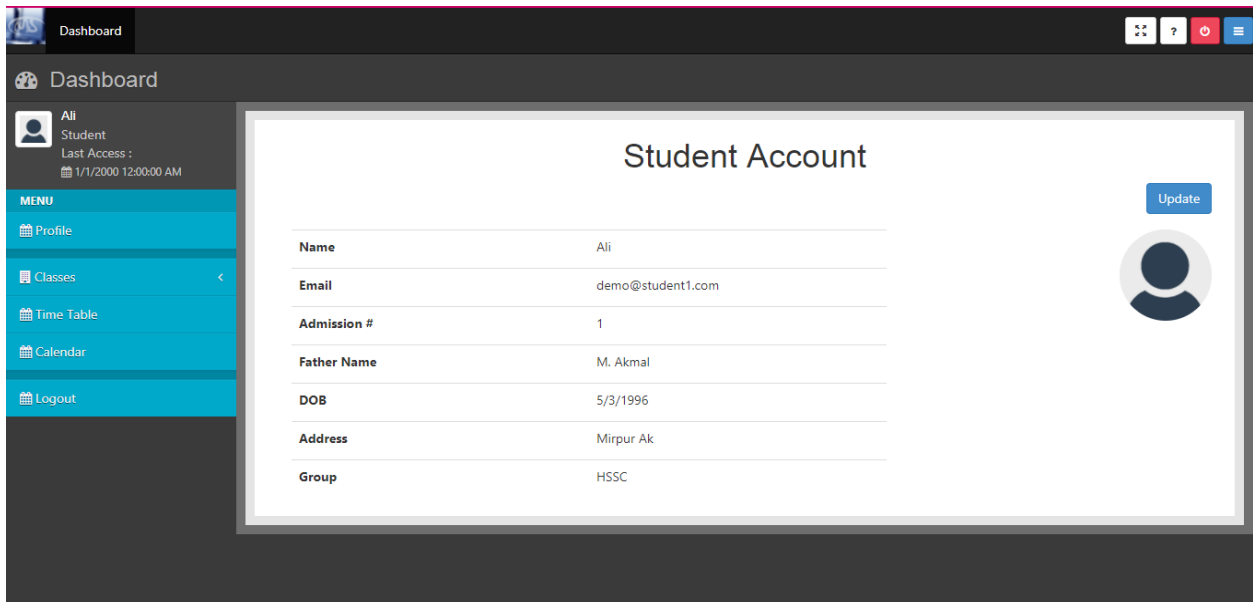


Figure 5.5 Student Account

5.4 Non-Functional Requirements

5.4.1 Performance Requirements

- Workload : The system shall be available for use for as many as 50 users at the same time.
- Response Time: 95% of all response time should be less than 1.0 second.
- Scalability: Software shall be capable of supporting at least 100 users

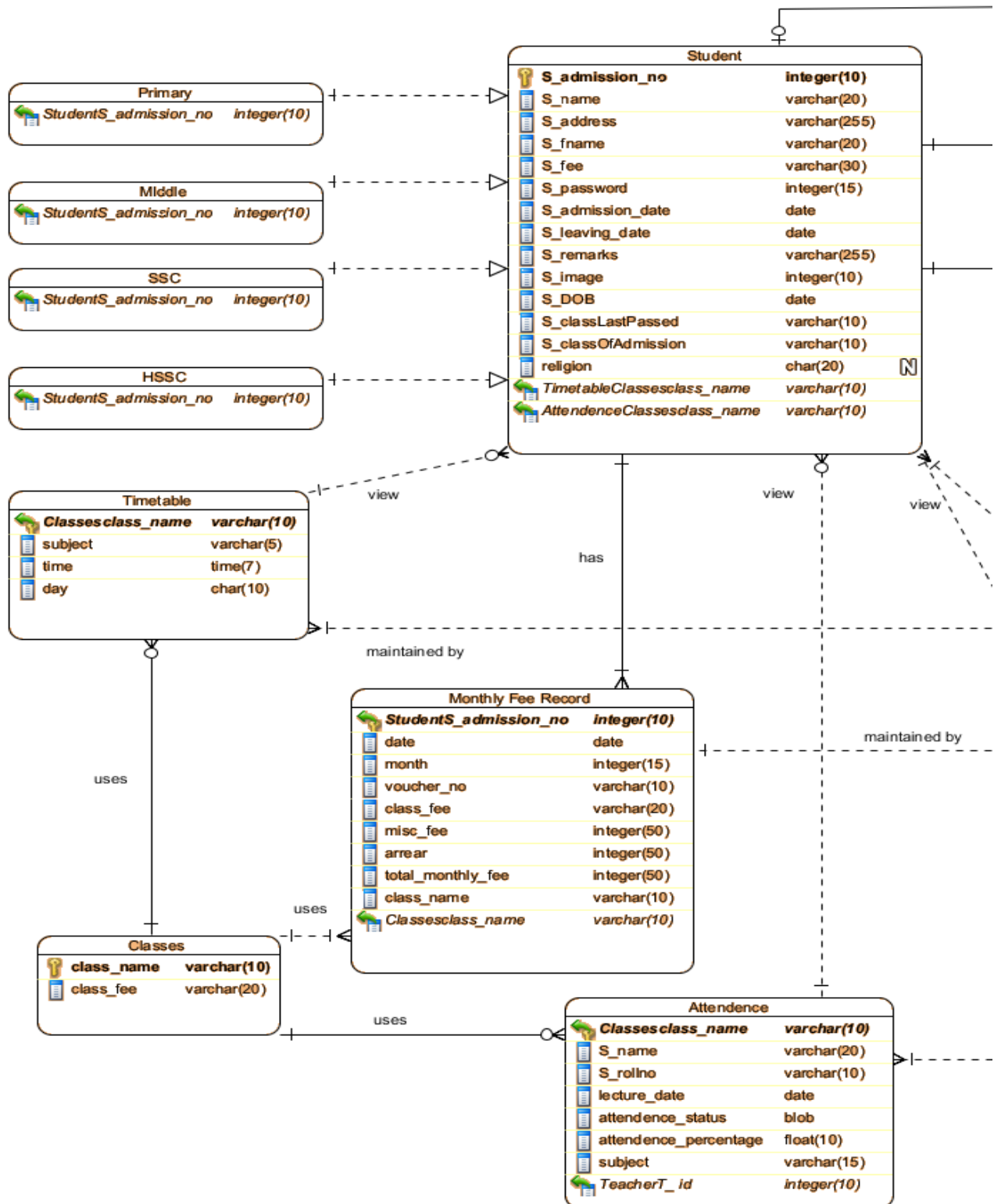
5.4.2 Safety Requirements

- System shall have a validation mechanism for all input requirements while login setup.
- Users must enter their first and last name, email and password while signup.
- Users shall be able to enter only one email address on signup.
- Users shall enter a password having minimum length 7 digits.
- Password must contain alphabets, Uppercase and numeric values & a special character

5.4.3 Security Requirements

- Users will be assigned their own logins and password combination.
- Each user will have their own unique sign-in details.
- No user will be allowed access without proper authentication details.
- No user will be allowed to access modules other than those allowed to their class.

5.5 Data Storage



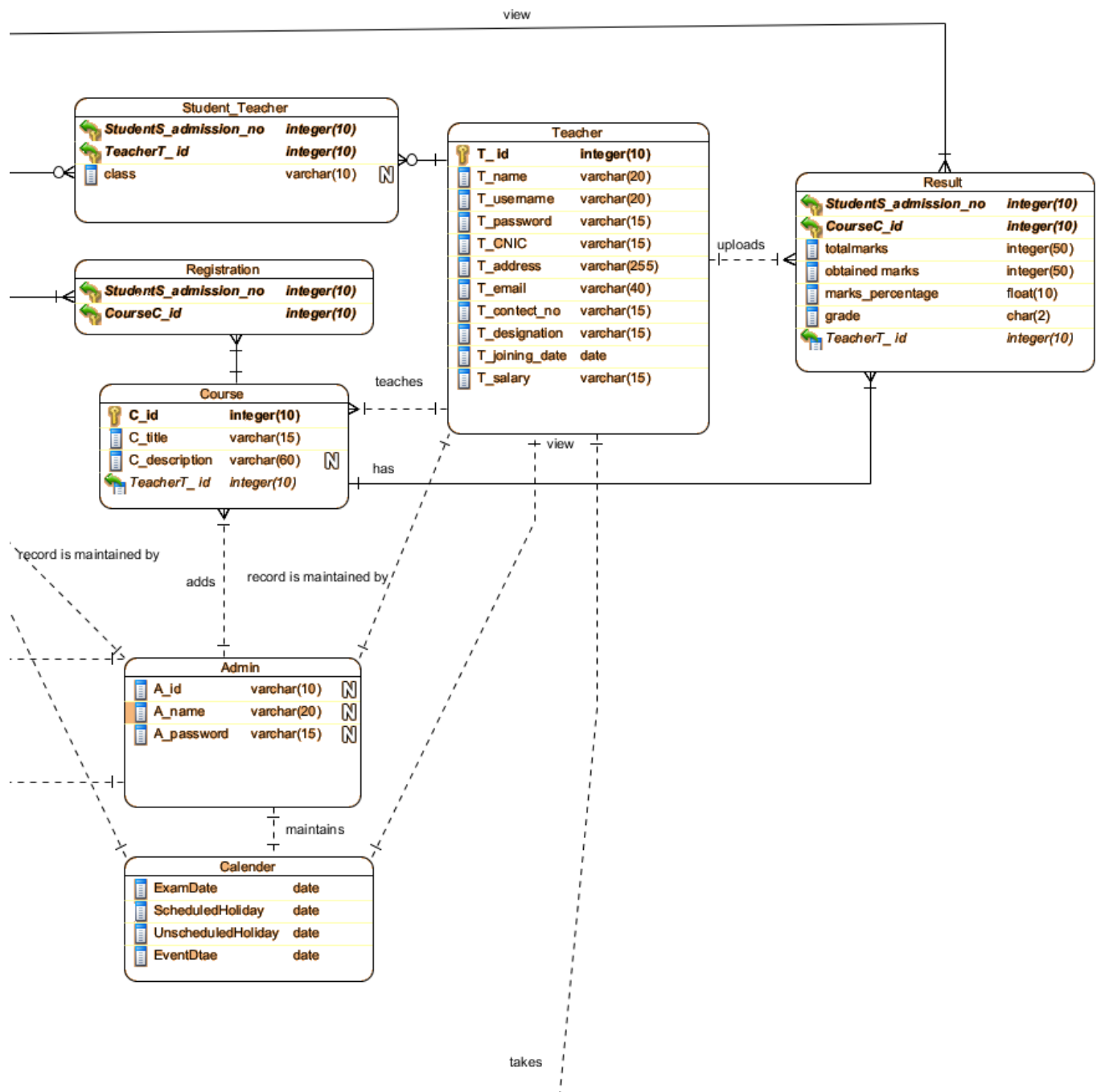


Figure 5.6 Data Storage View

5.6 Design Verification

sd Sequence Diagram1

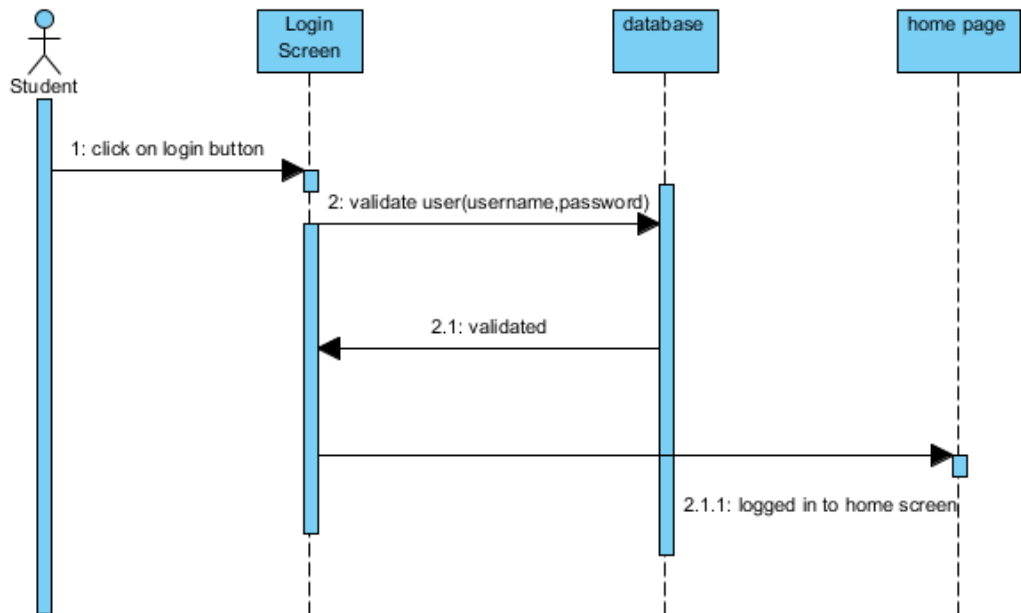


Figure 5.7 Sequence Diagram

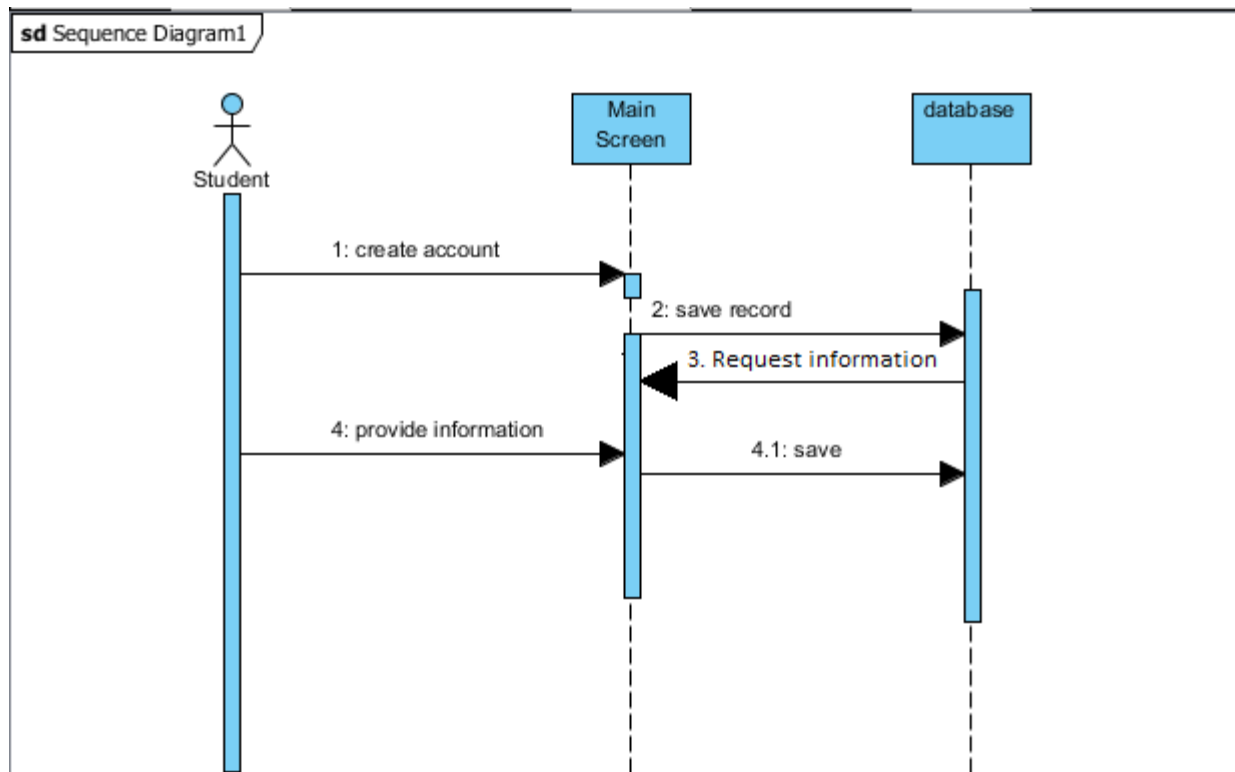


Figure 5.8 Create Account Sequence Diagram

Chapter 6 Software Design

6.1 Introduction

Model-view-controller (MVC) is a software architectural pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user.

6.2 High Level Design

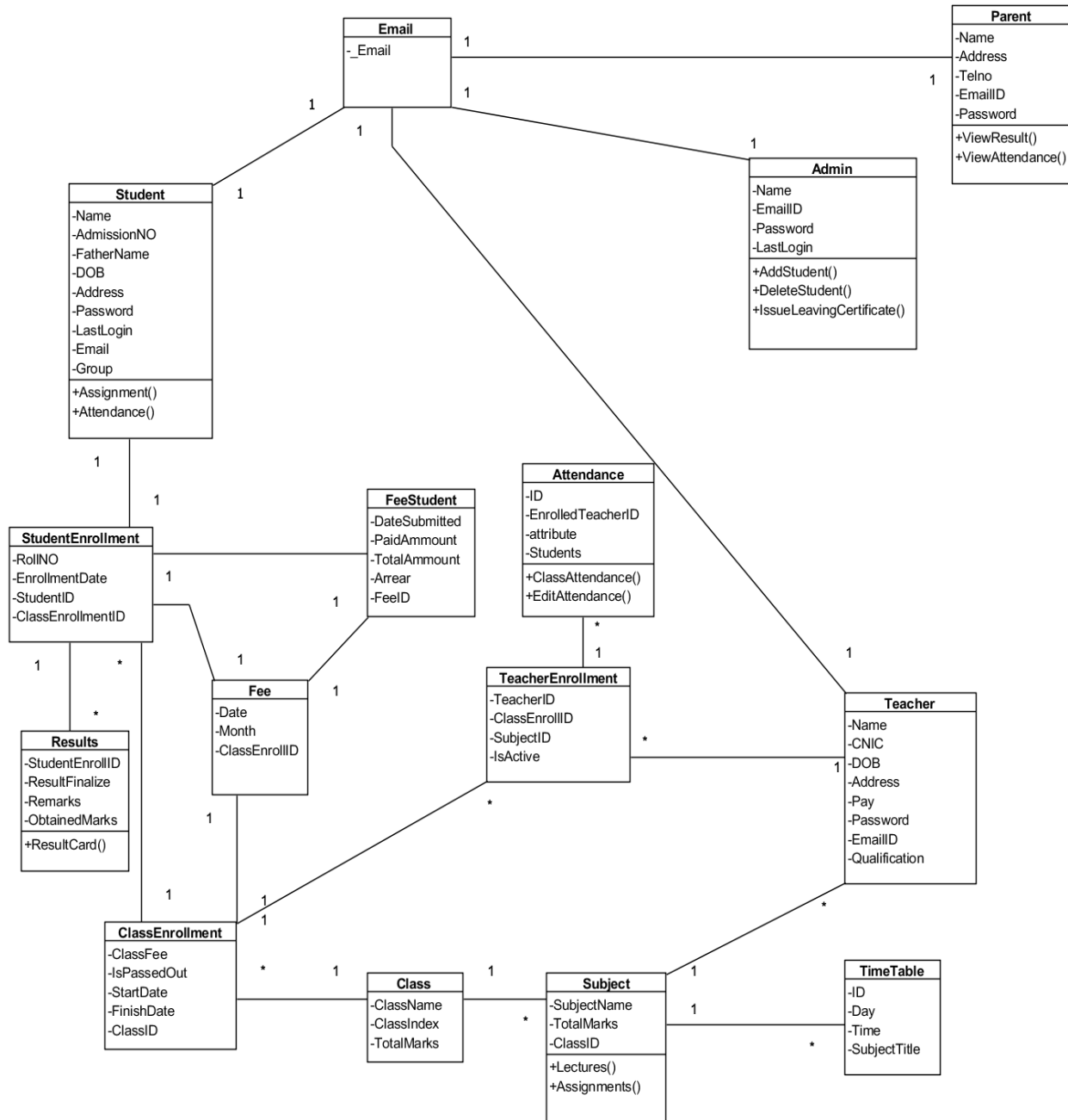


Figure 6.1 Class Diagram for the project

6.3 Activity Diagram

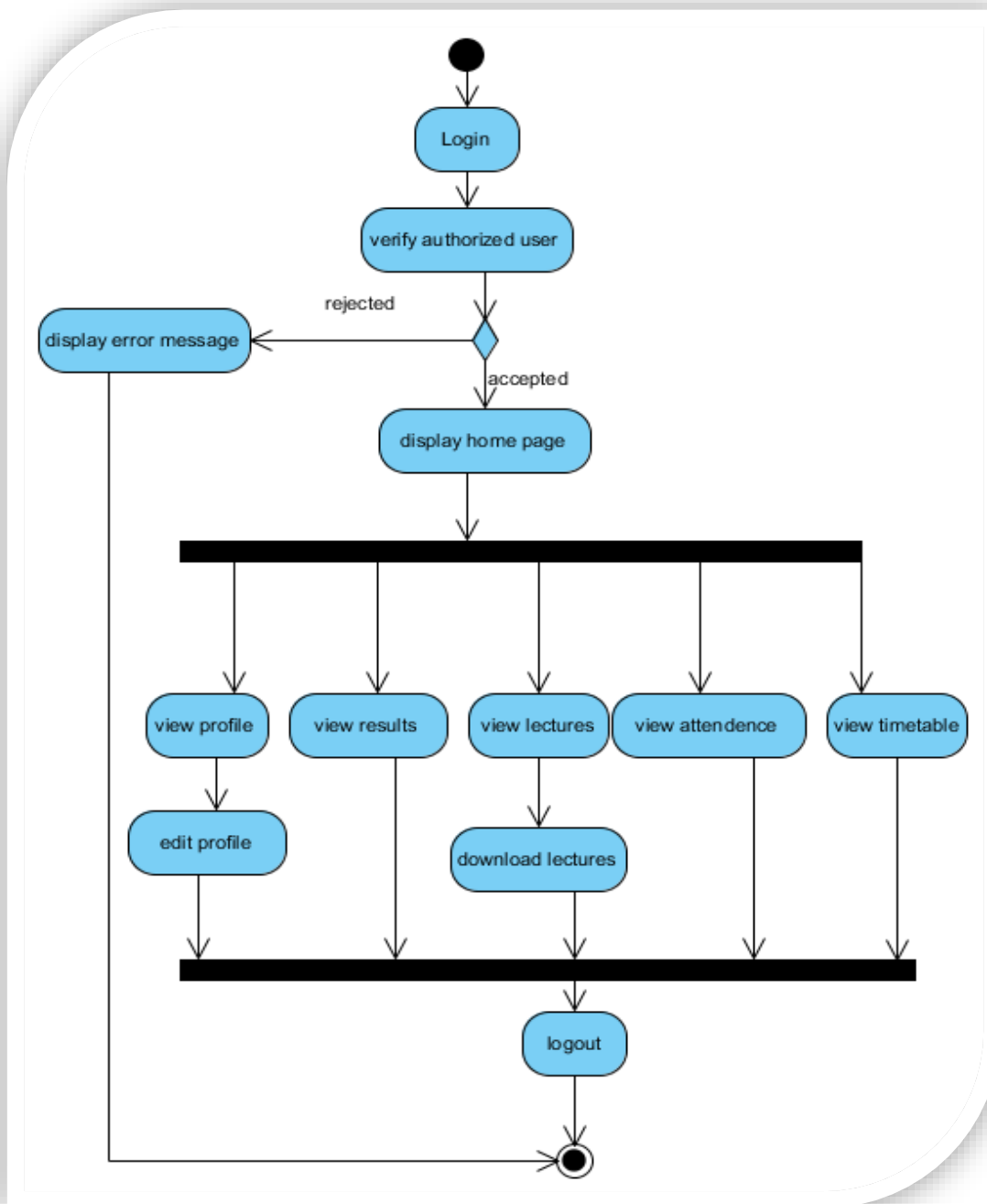


Figure 6.2 Activity Diagram

Chapter 7 Implementation

7.1 Introduction

We used C# .Net framework and MVC development environment to develop the system and SQL server entity framework for database.

We used the following tools:

- Visual Studio
- SQL server
- Visual Paradigm
- MS Word

7.2 Coding

After the completion of the class diagram, it is decided to translate them into c# code. So, for that purpose, every design class is converted into respective c# class. The whole coding is generated per a plan. Since we have models and controllers in the class diagram, so we put model classes into a folder Models. Every model in the design is like a table in the database. Because we are using entity framework, so every class will be converted into respective table in sql database per a database context used in the system. We have also constructed that database context using entity framework rules. The association among the classes is created per rules of entity framework. According to these rules, the attribute, named as ID will become the ID column of a table and the attribute, named as (foreign Table)ID will be considered as foreign key column of a foreign table automatically by the entity framework so according to these rules, every table has primary key and some contains foreign keys. Using this scheme of working, all the model classes are created in c# code.

For the Controllers, every controller class has been put into a folder 'Controller'. Each controller has Action Methods that can be called by a user request. A Method Controller will get the request from user, perform some action per user requirement and then it will return a respective action method view back to user. Every action method has a view that is defined in the Views folder in the respective controller folder. So, in this way all the code is managed in the whole project.

7.3 Verification

Table 7.1 Test Case Login Screen

Test Case No	1
Test Case Action	Login
Input	Enter username and password
Expected Output	login successfully.
Pass/Fail	Pass

Table 7.2 Test Case Login Failed

Test Case No	2
Test Case Action	Login
Input	Enter incorrect username and password
Expected Output	login fail.
Pass/Fail	Fail.

Table 7.3 Online Admission

Test Case No	3
Test Case Action	Online Admission
Input	Enter all required fields.
Expected Output	Data is save in the Database.
Pass/Fail	Pass

Table 7.4 Test Case Registration

Test Case No	4
Test Case Action	Registration
Input	Provide Username and password.
Expected Output	User successfully register.
Pass/Fail	Pass

Table 7.5 Test Case Delete Calendar

Test Case No	5
Test Case Action	Delete calendar.
Input	Click on delete button.
Expected Output	Calendar successfully deleted.
Pass/Fail	Pass

Table 7.6 Test Case Fee Challan Generation

Test Case No	6
Test Case Action	Fee challan Generation.
Input	Enter all required fields.
Expected Output	Total fee is calculated Challan generated successfully.
Pass/Fail	Pass

Table 7.7 Test Case Logout

Test Case No	7
Test Case Action	Logout
Input	Click on logout button.
Expected Output	Logout successfully.
Pass/Fail	Pass

Table 7.8 Test Case Logout Fail

Test Case No	8
Test Case Action	Logout
Input	Click on logout button.
Expected Output	Time span out.
Pass/Fail	Fail

7.4 Validation

Every system has some validity criteria, so this system also has some. For the validity of system's component, every component is taken out for validation testing. The validation components are highlighted in the system and a validity criterion is created according the requirement of that specific component.

Chapter 8 Discussion and Conclusion

8.1 Solution Review

In this project, we automated all the manual work of college like admission attendance, results, account and other activities that were done traditionally. It reduces Paper work, improve work efficiency and save time efficiently.

8.2 Project Review

The Campus Management System is a new self-contained software product which will be produced by the project team to overcome the problems that have occurred due to the current manual system. The newly introduced system will provide an easy access to the system and it will contain user friendly functions with attractive interfaces. The system will give better options for the problem of handling large scale of physical file system, for the errors occurring in calculations and all the other required tasks that has been specified by the client. The final outcome of this project will increase the efficiency of almost all the tasks done at the College in a much convenient manner.

8.3 Key Skills

The key skill that we learnt during this project is that we became familiar with C# .Net framework and MVC development environment to develop the system and SQL server entity framework for database.

8.4 Future Work

This project is of great significance as it will help in automating all the manual work of college. It eliminates duplication of data and provide work efficiency. Projects will be developed in better manners and cooperation.

8.5 Conclusion

The project is going to provide the information requested by students with the use of automated College Management System in an efficient and timely manner. Using CMS, instead of direct contacting with the faculty the student can check the results from the System if the student is registered in CMS. CMS will be helpful for educational institute to do regular activities accurately, fast and reliably. By using CMS student and faculty can find out overall attendance percentages, fee details and result analysis. Hence the CMS is going to serve the colleges in an efficient, effective and reliable way.

CMIS increases quality in work for educational institutes.

- The software facilitates the administrators to know the present status of a student of the college.
- The software gives the information such as student personal data, student fees details, results etc.

- Generating the print reports of student personal, fee as well as result details....
- Hence we conclude that the present system (CMS for Colleges) would definitely help the user by saving time and effort by reducing the processing time and volume of errors.
- The efficiency of the work done would be improved and work satisfaction on the part of the employees after computerization would definitely on high.
- The customer satisfaction would be definitely higher when compared to the old manual system

References

- [1] Clarke, Amelia, and Rosa Kouri. "Choosing an appropriate university or college environmental management system." *Journal of Cleaner Production* 17.11 (2009): 971-984.
- [2] Masound, F. A., and Dana H. Halabi. "ASP. NET and JSP frameworks in model view controller implementation." *Information and Communication Technologies, 2006. ICTTA'06. 2nd. Vol. 2. IEEE, 2006.*

CMSFinal

ORIGINALITY REPORT

20%

SIMILARITY INDEX

18%

INTERNET
SOURCES

2%

PUBLICATIONS

11%

STUDENT PAPERS

PRIMARY SOURCES

africageodownloads.info

Internet Source

4%

1

2

www.docstoc.com

Internet Source

1%

3

ieeexplore.ieee.org

Internet Source

1%

Rainer Alt. "Campus Management System", 1%
Business & Information Systems Engineering,

4

04/27/2010

Publication

5

ir.knust.edu.gh

Internet Source 1%

6

www.mfsa.com.mt

Internet Source 1%

dspace.knust.edu.gh:8080

Internet Source

1%

7

8

bcd.oregon.gov

Internet Source

1%

9

Submitted to University of the West Indies

Student Paper <1%

www.child
lead.com 10
Internet Source

<1 %

siou
xfalls.org
11
Internet Source

<1 %

www.law.u
penn.edu 12
Internet Source

<1 %

etd.a
au.edu.et
13
Internet Source

<1 %

extranet.acsys
web.com 14
Internet Source

<1 %

Submitted to University of London
External 15
System
Student Paper

<1 %

www.jds
upra.com 16
Internet Source

<1 %

legacy.cookcounty
gov.com 17
Internet Source

<1 %

www.cityofsth
elena.org 18
Internet Source

<1 %

z
test.
net
19
Internet Source

<1 %

www.dundee.ac.uk
20 Internet Source

<1 %

www.ac
ecma.org 21
Internet Source

<1 %

thesis.bi
nus.ac.id 22
Internet Source

<1 %

www.masonco
mets.org 23
Internet Source

<1 %

www.cs
e.scu.edu 24
Internet Source

<1 %

www.cos
.ohio.gov 25
Internet Source

<1 %

do
cslide.u
s 26
Internet Source

<1 %

www.authorstr
eam.com 27
Internet Source

<1 %

www.alexandr
java.gov 28
Internet Source

<1 %

www.eqb.sta
te.mn.us 29
Internet Source

<1 %

www
.rspo.org
30
Internet Source

<1 %

www.mnscha.or
g 31
Internet Source

<1 %

www.c
nmipss.org
32

Internet Source

<1 %

www.vtb.us
courts.gov 33

Internet Source

<1 %

works.be
press.com 34

Internet Source

<1 %

www
.unece.or
g 35

Internet Source

<1 %

www.perryville
school.org 36

Internet Source

<1 %

www.oakla
ndnet.com 37

Internet Source

<1 %

warm
oregon.org
38

Internet Source

<1 %

tea
.texas.g
ov 39

Internet Source

<1 %

documents.atlantareg
ional.com 40
Internet Source

<1 %

boardofed
.idaho.gov 41
Internet Source

<1 %

www.acsp.org

42 Internet
Source

<1 %

Waheed, Bushra, Faisal Khan, Brian Veitch,

43 <1
%

and Kelly Hawboldt. "An Integrated Decision-
Making Framework for Sustainability
Assessment: A Case Study of Memorial
University", Higher Education Policy, 2011.

Publication

www.elbertcounty-
co.gov 44
Internet Source

<1
%

pk 45 www.cchd.org <1
Internet Source %

lt 46 old.regula <1
Internet Source %

47 Submitted to University of Dundee <1
Student Paper %

48 Raghuramu, Arun, Parth H. Pathak, Hui Zang, <1
Jinyoung Han, Chang Liu, and Chen-Nee %
Chuah. "Uncovering the footprints of
malicious traffic in wireless/mobile networks",
Computer Communications, 2016.
Publication

in 49 www.ddu.ac <1
Internet Source %

s 50 boyne.k12.mi.u <1
Internet Source %

pbadupws.nrc.gov

Internet

Source 51

<1 %

g 52 oldweb.roundrockisd.or

Internet Source

<1 %

g 53 www.shavanopark.or

Internet Source

<1 %

us 54 www.dec.state.ak.

Internet Source

<1 %

rg 55 www.tegacaysc.o

Internet Source

<1 %

g 56 www.wmsym.or

Internet Source

<1 %

57 Submitted to Coventry University

Student Paper

<1 %

uk **58** www.iacom.org.
Internet Source

<1
%

EXCLUDE QUOTES OFF EXCLUDE MATCHES OFF
EXCLUDE OFF
BIBLIOGRAPHY