



# SCIENT INSTITUTE OF TECHNOLOGY

Ibrahimpattanam. R.R Dist - 501506

(Approved by AICTE & Affiliated to JNTUH, Hyderabad)

## SOFTWARE ENGINEERING LAB MANUAL

### Course Objectives

- To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

### Course Outcomes

- Ability to translate end-user requirements into system and software requirements
- Ability to generate a high-level design of the system from the software requirements
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

### PROJECT #1 : COURSE MANAGEMENT SYSTEM (CMS)

#### 1) Development of problem statement

##### Problem Statement:

The current system is a manual one where in the organization maintains all the information in the form of records. There by collecting necessary information which require a manual search in the record(s). Transfer of information between different sections of the enterprise is in the form of documents. Due to mismanagement the work is delayed to later date than the due date.

##### Purpose:

The Course Management System is an Intranet based application that can be accessed throughout the organization or a specified group/Dept. This system can be used to automate the workflow of courses. The Lab experiment will explain the purpose and features of the system, the interfaces of the system will do the constraints under which it must operate and how the system will react to external stimuli.

##### Solution:

Course Management System is very helpful to the users. The aim of this project is to provide quick, immediate access to the course. It can provide special advantages to the administration, faculties and students down the road. Login module helps the user to login to the site from anywhere. For that he/she must type the username and password correctly.

A Course Management System (CMS) is a collection of software tools providing an online environment for course interactions. A CMS typically includes a variety of online tools and environments, such as:

1. An area for faculty posting of class materials such as course syllabus and handouts
2. An area for student posting of papers and other assignments
3. A grade book where faculty can record grades and each student can view his or her grades
4. An integrated email tool allowing participants to send announcement email messages to the entire class or to a subset of the entire class
5. A chat tool allowing synchronous communication among class participants
6. A threaded discussion board allowing asynchronous communication among participants

In addition, a CMS is typically integrated with other databases in the university so that students enrolled in a particular course are automatically registered in the CMS as participants in that course. The Course Management System (CMS) is a web application for department personnel, Academic Senate, and Registrar staff to view, enter, and manage course information formerly submitted via paper. Departments can use CMS to create new course proposals, submit changes for existing courses, and track the progress of proposals as they move through the stages of online approval.

## **2) Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.**

### **2.1 Software Requirement Specification document:**

This document specifies the requirements for a Course Management System (CMS).

#### **2.1.1 Functional Requirements**

##### **2.1.1.1 Creating Courses**

##### **Integration with Registration System:**

- The System Shall periodically Upload The Latest Registrar's Classes List to determine Courses That Offered In The Current Semester.
- The System Shall Generate Course For Each Class That Registered And Determine The Current Set Of Students That Enrolled In That Class.
- The System Shall Allow Course Instructor To Update Course Content.

## 2.1.1.2 Grade Management

### Allow Grades To Be Entered Online

- The System Shall Allow Instructors To Enter And Modify Grades Online.

### Allow Students To Access Their Grades Online:

- The System Shall Allow Student To Log In Their Account And Check Their Grades at anytime.
- The System Shall Provides statistical Information Such As Averages, Standard Deviation, Median about Students Grades.

### Track And Handle Re-Grade Requests:

- The System Shall Be Able To Track And Handle Requests For Re- Grades, And All Information About Re-Grades Shall Be Available To The Student, And The Course Instructor.

### Homework Submission: Accept Submissions In Multiple Formats:

- The System Shall Accept Submissions In Multiple Formats, Including .Zip, .Cpp , .Txt, .Doc,Etc.

## 2.1.1.3 General requirements shared by stakeholders are:

- The system shall provide static course information
- The system shall be able to store static course information
- The system shall be able to represent static course information
- The system shall provide dynamic course information
- The system shall be able to store dynamic course information
- The system shall be able to represent dynamic course information
- The system shall provide a messaging system

## 2.1.1.4 Requirements by Students

- The system shall enable students to retrieve contact information of students and lecturers of subscribed courses.
- The system shall provide the history of a course (view contents of a course over the years)
- The system shall provide the history of attended courses.
- The system shall enable students to subscribe/unsubscribe to courses.
- The system shall enable students to subscribe/unsubscribe to exams.

- The system shall be able to provide a collaboration environment in a course (so students can share files and notes within a team).
- The system shall be able to let students submit textual content.
- The system shall be able to let students upload files.
- The System shall allow sending messages to individuals, teams or all course participants at once.
- The system shall allow students to create teams.
- Teams are created by students inviting other students in the same course using the messaging system.
- The system shall facilitate searches in all static information of courses.
- The system shall facilitate searches within all dynamic information and files in a course
- The system shall allow students to edit their personal information
- The system shall allow students to change their password
- The system shall provide a password reset function, which resets the password and mails it to the user
- The system shall notify students of events (posted news messages, team invites and scheduled exams)
- The system shall allow students to customize the notification behavior
- The system shall allow students to view course grade statistics per semester

## 2.1.2 Non functional requirements

Nonfunctional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes.

Let's have a close look at typical nonfunctional requirements.

### Privacy

- The system shall protect the user's privacy.
- The system shall prevent students from viewing grades of others
- The system shall provide a user-customizable visibility policy for the personal information

## Availability

- The system shall have high availability.
- The system shall not have unexpected downtime.
- The system shall have downtime at most 4 hours/month.
- The system shall have its expected downtime announced at least 48 hours in advance.
- The system shall have downtime only during low-intensity hours

## User friendliness

- The system will be user friendly.
- The system shall have a maximum of 3 clicks to reach any content.
- The system shall have a single login to access all content.
- The system shall have a consistent UI (in all the views and dialogs, the UI elements behave and are placed in a similar way)
- The system shall have a UI which is intuitive (the behavior of the system is according to the intuition of a standard end user)
- The system shall have a descriptive UI (all UI elements should have a descriptive text)

## Accessibility

- The system shall have high accessibility.
- The system shall be accessible by disabled (blind) users, who should be able to navigate the system and have access to all content and functionality

## Security

- The system shall allow lecturers to view the dynamic course information of courses given by other lecturers.
- The system shall allow lecturers to manage the dynamic content visibility (visible for students and lecturers, visible for lecturers, visible to self only).
- The system shall allow students to view only their own grade.
- The system shall allow lecturers to view all grades of all students in the course

## Interoperability

- The system shall be able to import BOZ roster information into the course roster

## 2.2 Design document

### 2.2.1 Introduction

Detailed design starts after the system phase and system has been certified through the review. The goal of this phase is to develop internal logic of each of the modules identified during system design

In the system design, the focus is on identifying the modules, whereas during the detailed design the focus is on designing the logic for the modules. In other words in system design attention is on what components are needed, while in the detailed design how the component can be implemented in the software is the issue.

The design process for software system has two events. At the first level focus is on deciding which modules are needed for the system, the specification of these modules and how the modules should be interconnected. This is called system design or top level design. In the specification of the module can be satisfied is decided. This design level is often called detailed design or logic design, because the detailed design is extension of system design, system design controls the major structural characteristics of the system. The system design has a major impact testability and modifiability of a system and impacts its efficiency much of the design efforts for the designing software are spent creating the system design.

### 2.2.2 System design

In the flexibility of the uses the interface has been developed keeping a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Data updation along with the extensive data search capabilities. The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

## 2.2.2.1 Modules description

The system after careful analysis has been identified to be presented with the following modules:

### The modules involved are:

- Administration
- Student
- Search
- Report
- Authentication

### Administrator:-

In this module the Administrator has the privileges to add all the students and register them in the organization and check the information of the student and search is done based on the student and course and report is generated for individual student.

### Search:-

This module contains complete search like course search, faculty based on the course they are taking.

### Student-

In this module the student has the privileges to use his username and password for login and he can see the .Use the login id for submitting assignments and taking courses and enter the problems while he is undergoing a course.

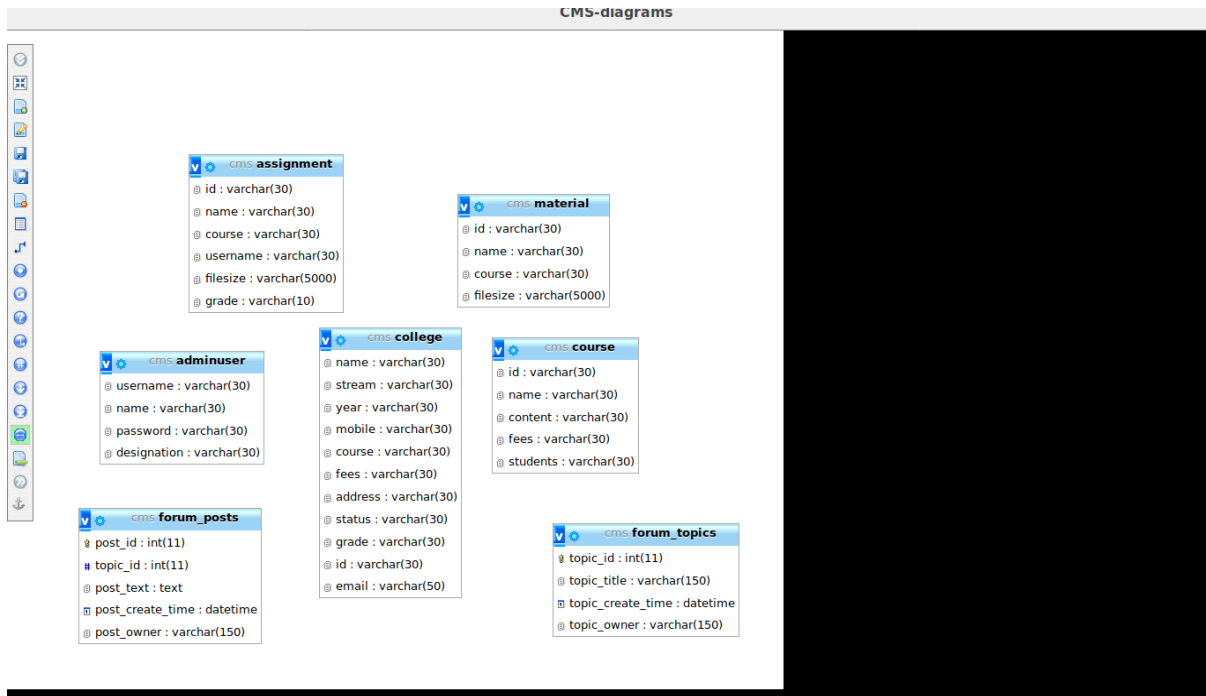
### Reports:-

This module contains all the information about the reports generated for the students based on the Performance by them for a specific course.

### Authentication:-

This module contains all the information about the authenticated user. User without his username and password can't enter into the login if he is only the authenticated user then he can enter to his login.

## 2.2.2.2 Database design



## 2.3 Testing related documents

### 2.3.1 Introduction to System testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 2.3.2 Types of testing

#### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each



unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### **White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a

black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **3) Preparation of Software Configuration Management and Risk Management related documents.**

#### **3.1 Software Configuration Management**

The overall objective of a Software Configuration Management (SCM) Plan is to document and inform project stakeholders about CM with the project, what CM tools will be used, and how they will be applied by the project to promote success. The CM Plan defines the project’s structure and methods for

- Identifying, defining, and base lining configuration items (CI)
- Controlling modifications and releases of CIs
- Reporting and recording status of CIs and any requested modifications
- Ensuring completeness, consistency, and correctness of CIs
- Controlling storage, handling, and delivery of the CIs

##### **3.1.1 Configuration Management Activities**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

##### **3.1.2 Software and Hardware Requirements**

###### **Software Requirements:**

Operating System: Windows XP or later

Database : Microsoft SQL Server-2005

## Hardware Requirements:

Processor: Intel Pentium or More

Ram: 512 MB Ram

Hard Disk: PC with 20GB

## 3.2 Risk Management document

Risk is defined as the possibility of any negative occurrence that may happen due to external or internal factors, and that may be mitigated through preventive actions. All projects are subject to risks. In fact, there is an infinite number of things that might prevent you from achieving your goals when working on a project. Risk management minimizes those threats that could cause project failure, and allows you to stay in control of your project's schedule, budget and quality requirements.

Risk management can be divided into four processes:

- **Identification:** Detect risks that might prevent you from achieving your project's goals.
- **Analysis:** Determine what risks are the most dangerous.
- **Planning:** Plan for the most dangerous risks.
- **Monitoring and control.** Maintain the project's plan and continually identify risks.

Some of the risks encountered in CMS are:

- Form validation required, used JS for validation purpose
- CMS must be more scalable
- We implement various techniques to boost application stability.
- Also we use Java to ensure the quality of our code

In terms of risk management, such an approach allows us to identify and fix errors at the earliest stage, when they do not yet lead to losses, and easily maintain the code in the long term.

These are several common risks in website development, and some common approaches to solving them. Course Management System (CMS) is a collection of software tools providing an online environment for course interactions.during our work to foresee risks and mitigate them. We also reduce the risk of failing to meet deadlines through efficient communication within our teams and through technical checks.

## 4) Study and usage of any Design phase CASE tool

### Study of CASE tool

Computer aided software engineering (CASE) is the implementation of computer facilitated tools and methods in software development. CASE is used to ensure a high-quality and defect-free software. CASE ensures a check-pointed and disciplined approach and helps designers, developers, testers, managers and others to see the project milestones during development.

CASE can also help as a warehouse for documents related to projects, like business plans, requirements and design specifications. One of the major advantages of using CASE is the delivery of the final product, which is more likely to meet real-world requirements as it ensures that customers remain part of the process.

CASE illustrates a wide set of labor-saving tools that are used in software development. It generates a framework for organizing projects and to be helpful in enhancing productivity. There was more interest in the concept of CASE tools years ago, but less so today, as the tools have morphed into different functions, often in reaction to software developer needs. The concept of CASE also received a heavy dose of criticism after its release.

### CASE Tools:

The essential idea of CASE tools is that in-built programs can help to analyze developing systems in order to enhance quality and provide better outcomes. Throughout the 1990, CASE tool became part of the software lexicon, and big companies like IBM were using these kinds of tools to help create software. Various tools are incorporated in CASE and are called CASE tools, which are used to support different stages and milestones in a software development life cycle.

### Types of CASE Tools:

#### Diagramming Tools:

It helps in diagrammatic and graphical representations of the data and system processes. It represents system elements, control flow and data flow among different software components and system structure in a pictorial form.

For example, Flow Chart Maker tool for making state-of-the-art flowcharts.

#### Computer Display and Report Generators:

It helps in understanding the data requirements and the relationships involved.

#### Analysis Tools:

It focuses on inconsistent, incorrect specifications involved in the diagram and data flow. It helps in collecting requirements, automatically check for any irregularity, imprecision in the diagrams, data redundancies or erroneous omissions.

### **Central Repository:**

It provides the single point of storage for data diagrams, reports and documents related to project management.

### **Documentation Generators:**

It helps in generating user and technical documentation as per standards. It creates documents for technical users and end users.

For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

### **Code Generators:**

It aids in the auto generation of code, including definitions, with the help of the designs, documents and diagrams.

### **Advantages of the CASE approach:**

As special emphasis is placed on redesign as well as testing, the servicing cost of a product over its expected lifetime is considerably reduced.

- The overall quality of the product is improved as an organized approach is undertaken during the process of development.
- Chances to meet real-world requirements are more likely and easier with a computer-aided software engineering approach.
- CASE indirectly provides an organization with a competitive advantage by helping ensure the development of high-quality products.

### **Disadvantages of the CASE approach:**

- **Cost:** Using case tool is a very costly. Mostly firms engaged in software development on a small scale do not invest in CASE tools because they think that the benefit of CASE are justifiable only in the development of large systems.
- **Learning Curve:** In most cases, programmers productivity may fall in the initial phase of implementation , because user need time to learn the technology. Many consultants offer training and on-site services that can be important to accelerate the learning curve and to the development and use of the CASE tools.

- **Tool Mix:** It is important to build an appropriate selection tool mix to urge cost advantage  
CASE integration and data integration across all platforms is extremely important.

## Usage of any CASE tool

**StarUML** is the tool we suggest because it makes a clear conceptual distinction between models, views and diagrams. A Model is an element that contains information for a software model. A View is a visual expression of the information contained in a model, and a Diagram is a collection of view elements that represent the user's specific design thoughts.

StarUML is build as a modular and open tool. It provides frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta-model and tool through COM Automation, and it provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

StarUML supports the following diagram types

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- Statechart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Composite Structure Diagram

The user interface is intuitive. On the upper right side, a window allows to rapidly navigate between all the content of a project, adopting either a model or a diagram view. Multiple diagrams can be open at the same time and tabs allow switching rapidly between views. The lower right window allows to document the current diagram, either with plain text or attaching an external document. During diagram editing, "wizards" are located around the object that give you the quick shortcuts to main associated tasks with your current operation, like adding an attribute when you create a class for instance. A right-click on the mouse brings the full set of operations at your disposal.

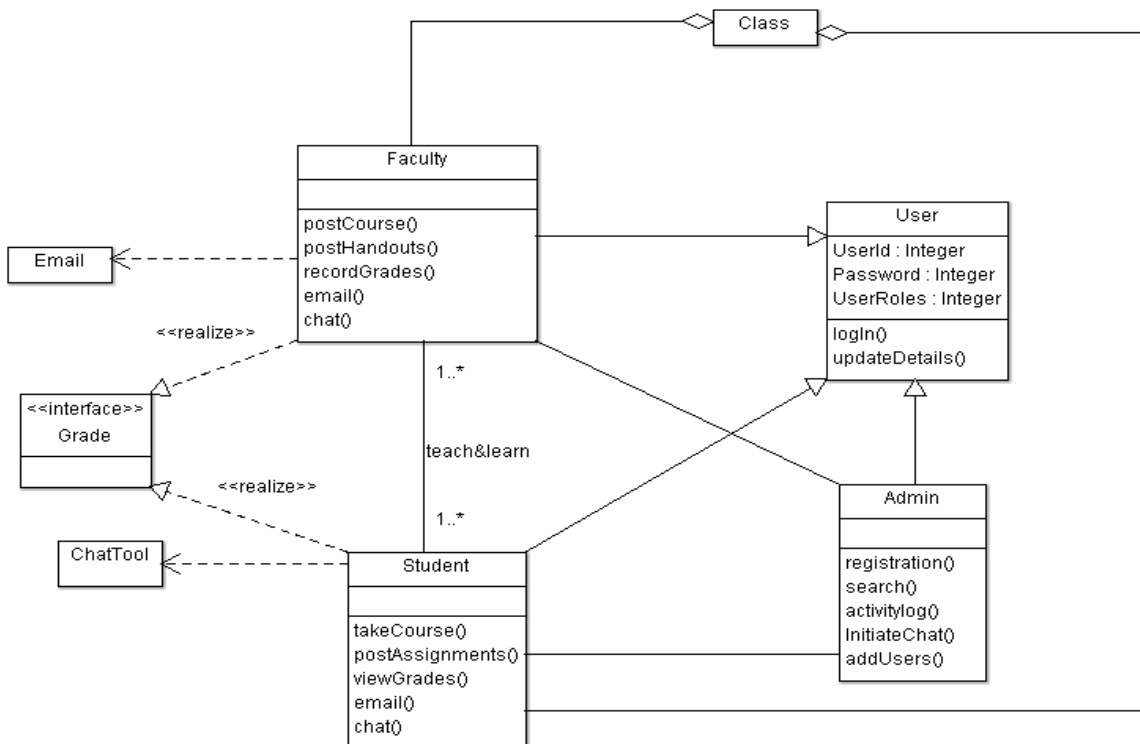
StarUML has also a model verification feature. You can export diagram in different formats (jpg, bmp, wmf). It also supports a patterns approach and import of Rational Rose files. StarUML

Generator is platform module to generate various artifacts (like as Microsoft Word, Excel, PowerPoint, and Text-based artifacts) by templates depending on UML model elements in StarUML. The users can define their own templates and can apply many different kinds of templates to the same UML model, so the users can get various artifacts automatically, easily and fast. The tool supports code generation and reverse engineering for Java, C# and C++. StarUML has many powerful features and is certainly more than a "simple" diagramming tool. With its support of MDA (Model Driven Architecture), it is more aimed at people using UML in an intensive way and with some code generations objectives than for simply drawing diagrams to document requirements. However, using StarUML just as a diagramming tool work fine, especially on Windows as the tool is built with Delphi and might execute faster than the Java-based tools.

## 5) Performing the Design by using any Design phase CASE tools.

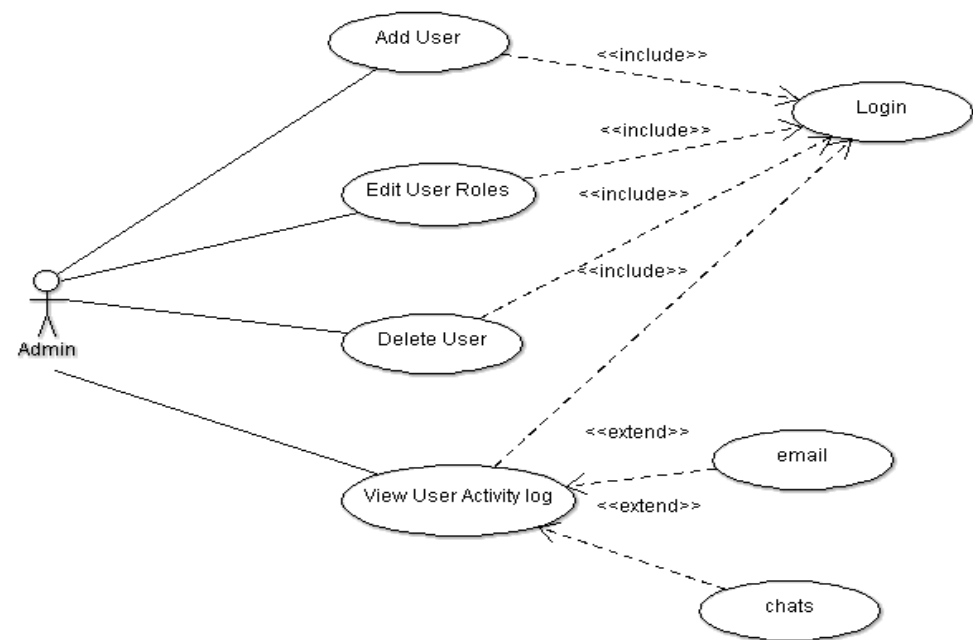
### 5.1 Design phase in CMS

#### 5.1.1 Class Diagram for faculty, student and admin

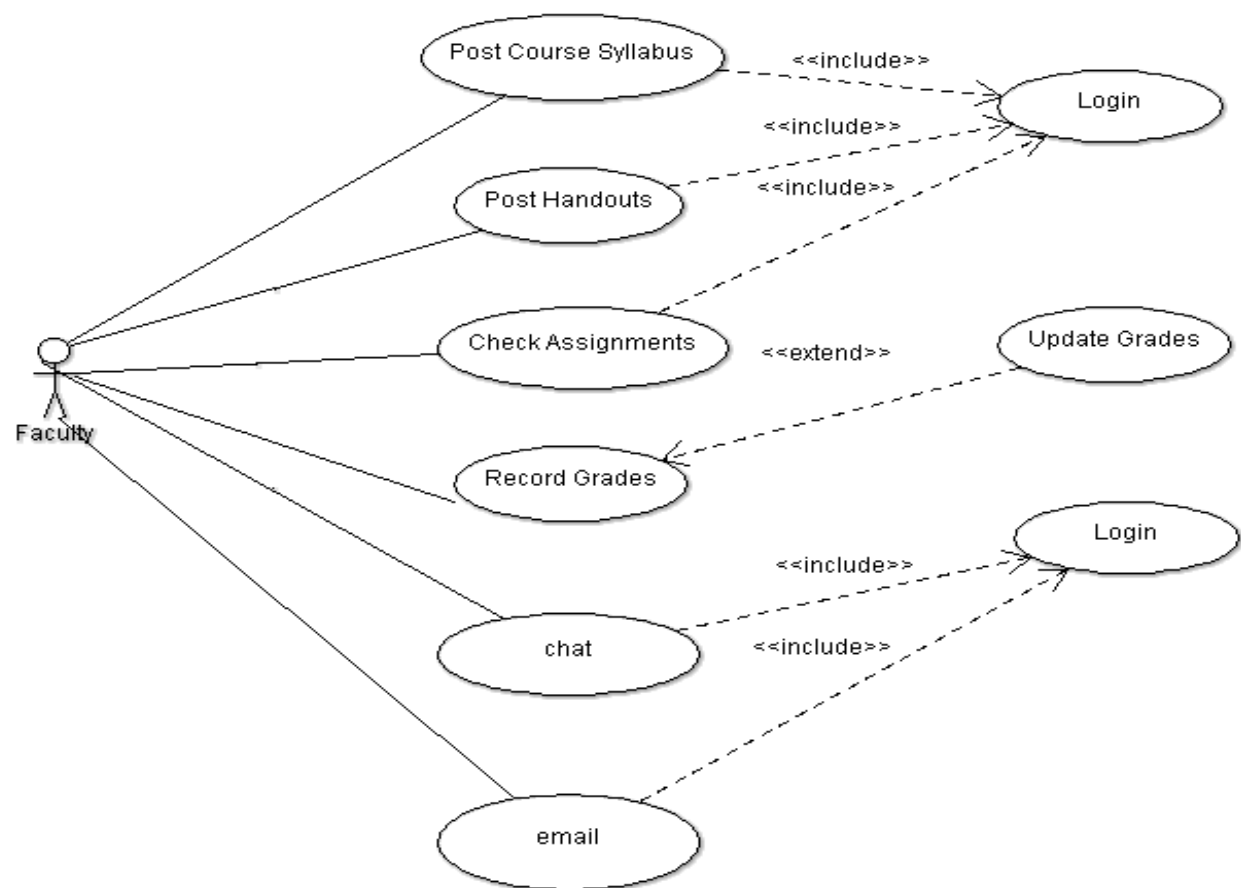




5.1.2 Use case diagram for admin

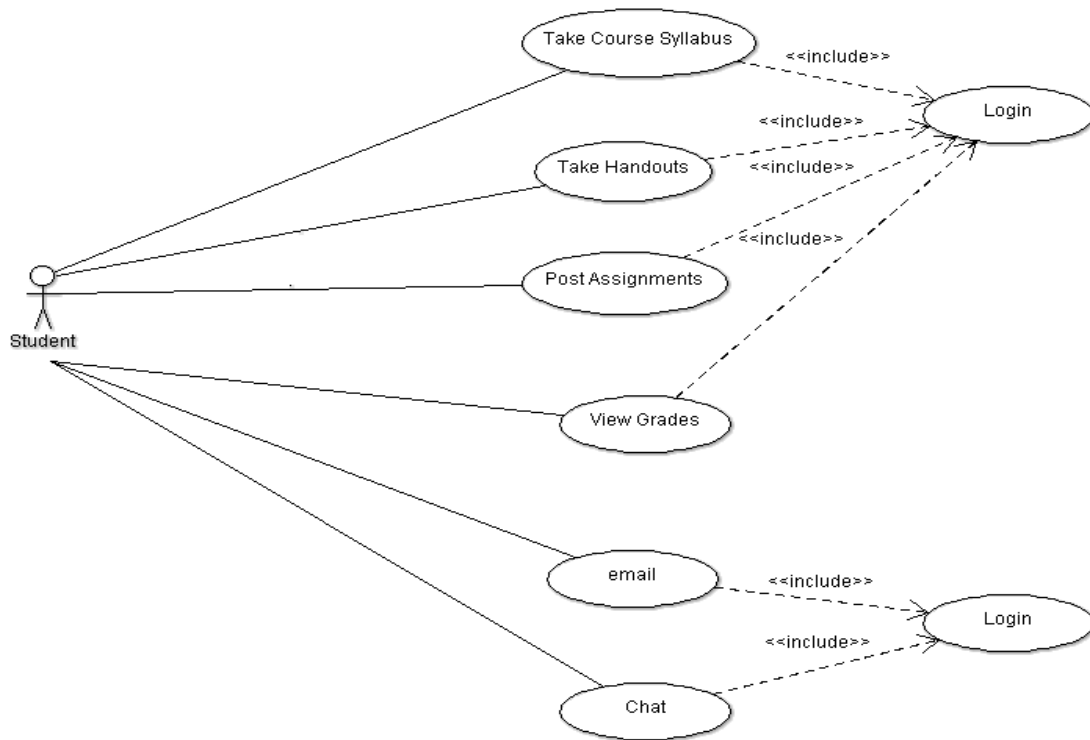


5.1.3 Use case diagram for faculty

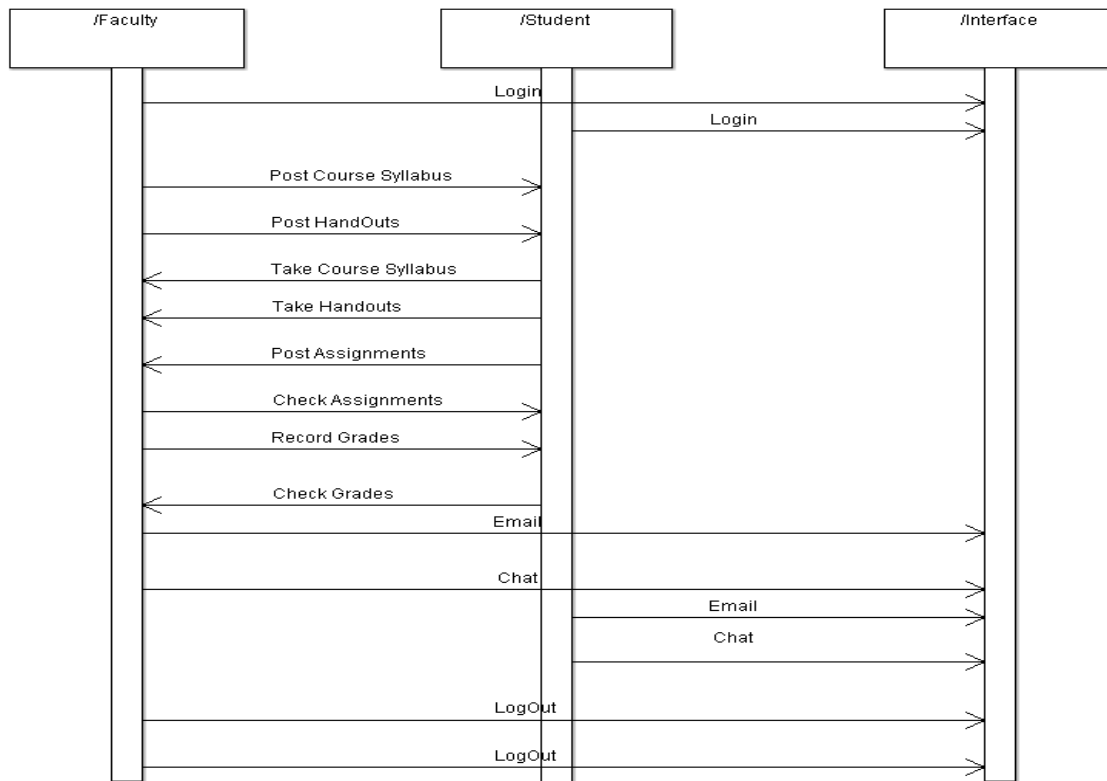




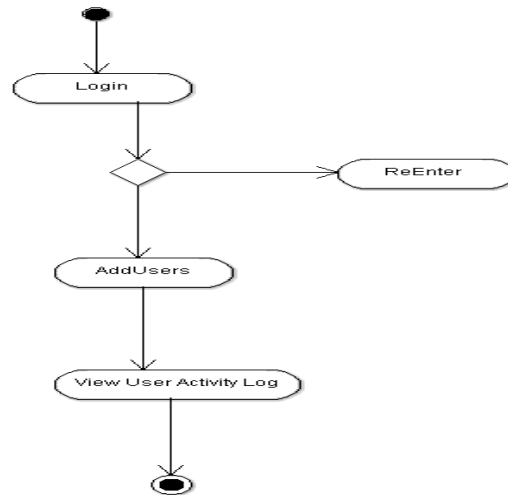
### 5.1.4 Use case diagram for Student



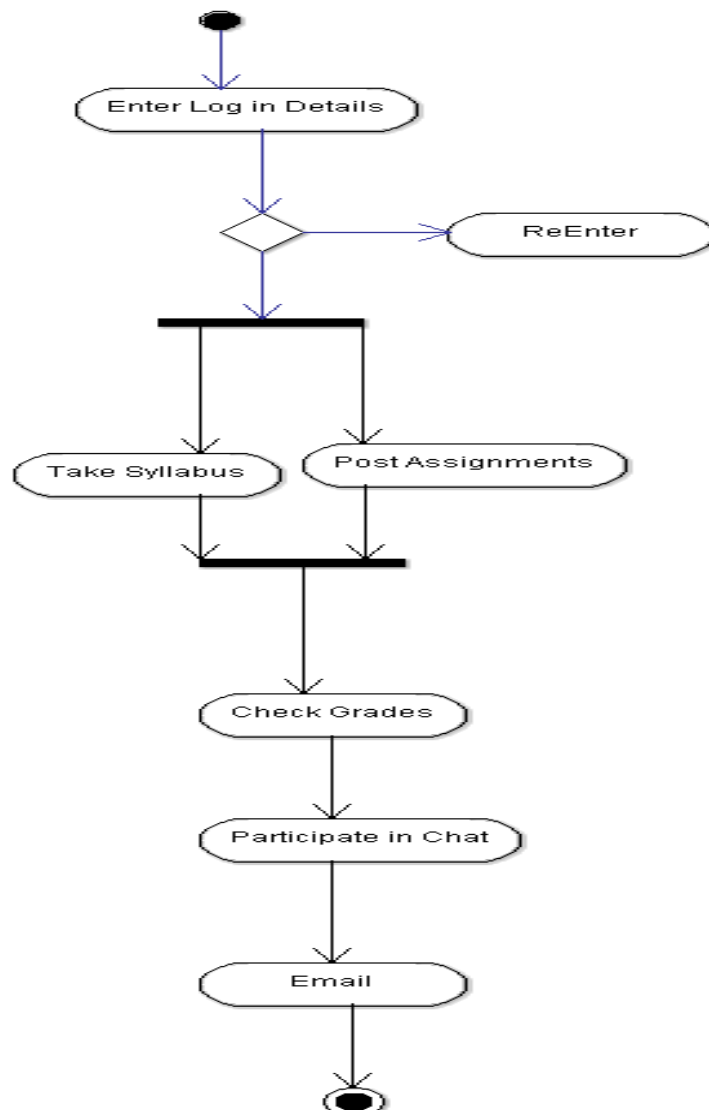
### 5.1.5 Sequence diagram



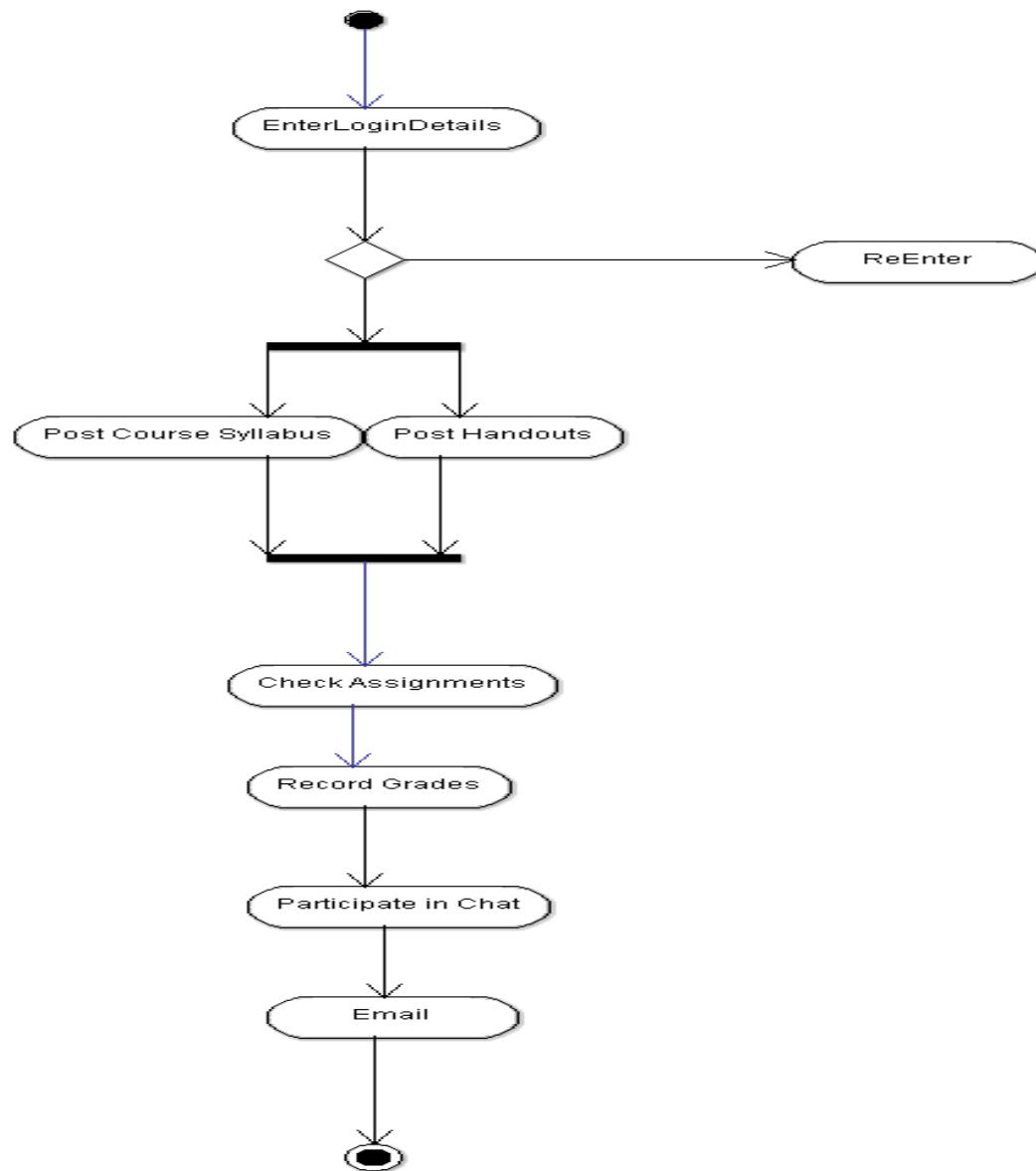
## 5.1.6 Activity diagram for Admin



## 5.1.7 Activity diagram for Student

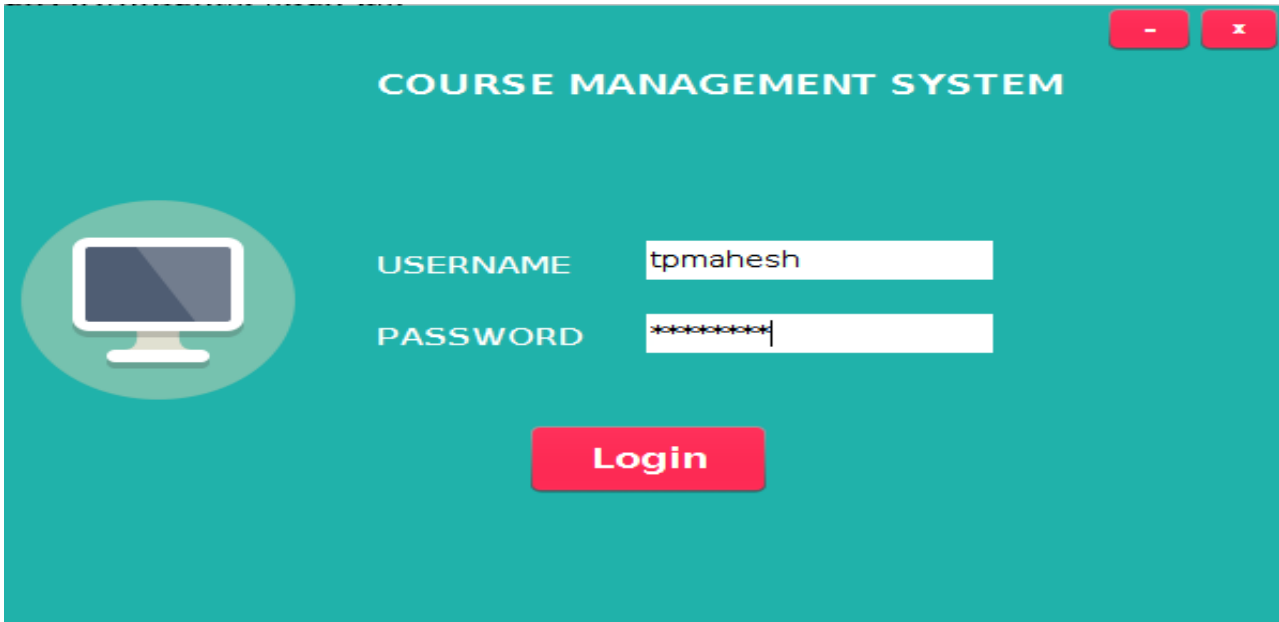


## 5.1.8 Activity diagram for Faculty



## 6) Test cases for unit testing and integration testing

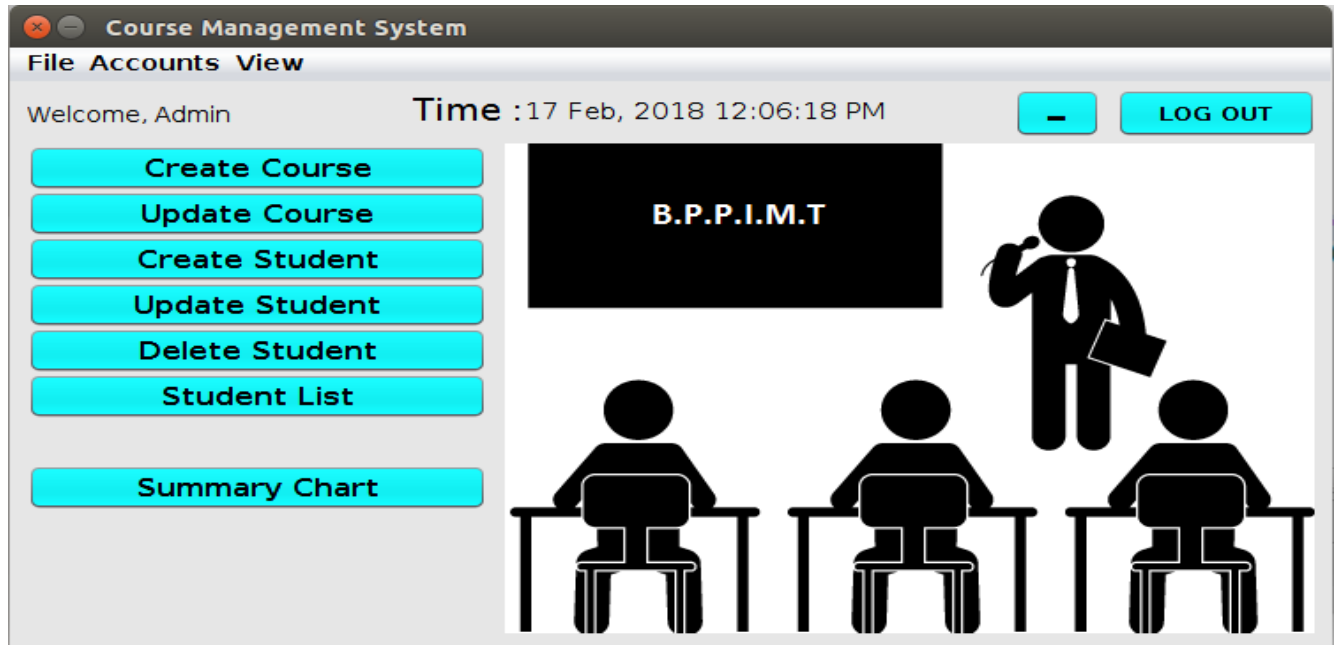
### 6.1 Testing login for CMS



- Check if the username field accepts valid username and password field accepts valid password.
- Check if the wrong username and valid password allows access to any specific account.
- Check if the valid username and wrong password allows access to any specific account.
- Check if the forgot username link leads to username recovery page.
- Check if the forgot password link leads to the password recovery page.
- Check if the invalid username and password triggers any warning.
- Check if the invalid credentials open the random account.
- Check if the user is logged in, allows you to logout by using the link at the bottom of the application.
- Check if the logout link functions as expected.

## 6.2 Testing course and student management in CMS

This screenshot shows to add courses and students into CMS



- Check if the student can be searched using the first name or last name or H.T Number.
- Check if the course details are updated.
- Check if the student details are updated.
- Check if the student and faculty data can be modified if you are admin.
- Check if the new student is possible to add into the system using create student button.
- Check if the student can be removed using delete member feature.

## 6.3 Testing searching and enrolling of courses in CMS

This screenshot provides the details about student enrollment in courses.

Course Enrolment

Choose Course

C

Enrol Now

New Student

Name

Mahesh

STREAM

CSE

YEAR

1

Mobile

9885008881

Course

C

Fees

1000

Address

Hyderabad

Status

Paid

Grade


B

STID

100

Email

mahesh@gmail.com



Register

- Check if the courses can be searched using the search feature.
- Check if admin can add the course using type
- Check if you can modify or edit the course
- Check if you can save the course information.
- Check each field for the limit of the text fields and also valid input for the form.

### 2.3.3 Test cases for black box and white box testing

#### WHITE-BOX TESTING

White box some times called “Glass box testing” is a test case design uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests where made on the system:

- A) All independent paths within a module have been exercised once.
- B) All logical decisions were checked for the truth and falsity of the values.

#### BLACK-BOX TESTING

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program.

- 1) Interface errors
- 2) Performance in data structure
- 3) Performance errors
- 4) Initializing and termination errors

#### TESTCASES

##### Admin login form

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	Enter valid admin username and password	CMS should display home window	Successful
2	Enter invalid username and password	CMS should display incorrect username and password	Successful

### Admin page

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	On click of “ Create Course”	Admin need to fill the details of courses and click on submit then CMS should display course is created successfully	Successful
2	On click of “Update Course”	If admin wants to update the course details , after updating the required fields and click on update then CMS should display course is updated successfully	Successful
3	On click of “Create student”	Admin need to fill the details of the students and click on register then CMS should display student registered successfully	Successful
4	On click of “Update student”	If admin wants to update the students details , after updating the required fields and click on update then CMS should display student details updated successfully	Successful

### Faculty page

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	On click of “Material upload”	Faculty need to fill the form and upload the document then click on upload then CMS should display material uploaded successfully	Successful
2	On click of “Grade student”	If faculty need to grade the student then by allotting a grade	Successful



		to the student CMS should display student grade was allotted successfully	
3	On click on “View Students”	If wants to check the list of enrolled students in the CMS then by selecting the Academic year and department CMS should display the students list	Successful
4	On click on “Send Email”	Faculty wants to sent an Email to the students then after filling the form CMS should display Email sent successfully	Successful

**Student page**

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	On click on “View available courses”	Student wants to check number of courses available after clicking on that CMS should display list of courses which are live will be available	Successful
2	On click on “Enroll a course”	Students wants to enroll a course then after filling all the details CMS should display course is successfully enrolled	Successful
3	On click on “Assignment upload”	If a student wants to upload an assignment after selecting the course and uploading the file then click on submit then CMS should display assignment is successfully uploaded	Successful

\*\*\*

## **PROJECT #2 : EASY LEAVE MANAGEMENT**

### **1) Development of problem statement**

#### **Problem Statement:**

#### **Purpose:**

This project is aimed at developing a web based Leave Management Tool, which is of importance to either an organization or a college. The Easy Leave is an Intranet based application that can be accessed throughout the organization or a specified group/Dept. These systems can be used to automate the workflow of leave applications and their approvals. The periodic crediting of leave is also automated. There are features like notifications, cancellation of leave, automatic approval of leave, report generators etc in this Tool.

#### **Solution:**

Easy leave is the web based management tool where the information of the entire firm will be maintained at a centralized data base any changes made by the other departs are known to the higher or lower departments instantly. Provide Interactive interface through which a user can interact with different areas of application easily. Deploy the application on a single system and make is available on all the systems within the network, thereby reducing the maintenance cost of software.

### **2) Preparation of Software Requirement Specification Document, Design**

#### **Documents and Testing Phase related documents.**

#### **2.1 Software Requirement Specification document:**

This document specifies the requirements for a Easy leave.

##### **2.1.1 Functional Requirements**

Admin of the software will register all the employees into the system. Out of which some are approvers. An approver can also be a requestor. In an organization, the hierarchy could be Engineers/Managers/Business Managers/Managing Director etc. In a college, it could be Lecturer/Professor/Head of the Department/Dean/Principal etc.

Following is a list of functionalities of the system: A person should be able to

- Login to the system through the first page of the application
- Change the password after logging into the system

- See his/her eligibility details (like how many days of leave he/she is eligible for etc)
- Query the leave balance
- See his/her leave history since the time he/she joined the company/college
- Apply for leave, specifying the from and to dates, reason for taking leave, address for communication while on leave and his/her superior's email id
- See his/her current leave applications and the leave applications that are submitted to him/her for approval or cancellation
- Approve/reject the leave applications that are submitted to him/her
- Withdraw his/her leave application (which has not been approved yet)
- Cancel his/her leave (which has been already approved). This will need to be approved by his/her superior.
- Get help about the leave system on how to use the different features of the system
- As soon as a leave application /cancellation request /withdrawal /approval /rejection /password-change is made by the person, an automatic email should be sent to the person and his superior giving details about the action
- The number of days of leave (as per the assumed leave policy) should be automatically credited to everybody and a notification regarding the same be sent to them automatically
- An automatic leave-approval facility for leave applications which are older than 2 weeks should be there. Notification about the automatic leave approval should be sent to the person as well as his superior

## 2.1.2 Non functional requirements

Non-functional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes.

Let's have a close look at typical non-functional requirements.

### Privacy

- The system shall protect the user's privacy.
- The system shall prevent students from viewing grades of others
- The system shall provide a user-customizable visibility policy for the personal information

## Availability

- The system shall have high availability.
- The system shall not have unexpected downtime.
- The system shall have downtime at most 4 hours/month.
- The system shall have its expected downtime announced at least 48 hours in advance.
- The system shall have downtime only during low-intensity hours

## User friendliness

- The system will be user friendly.
- The system shall have a maximum of 3 clicks to reach any content.
- The system shall have a single login to access all content.
- The system shall have a consistent UI (in all the views and dialogs, the UI elements behave and are placed in a similar way)
- The system shall have a UI which is intuitive (the behavior of the system is according to the intuition of a standard end user)
- The system shall have a descriptive UI (all UI elements should have a descriptive text)

## Accessibility

- The system shall have high accessibility.
- The system shall be accessible by disabled (blind) users, who should be able to navigate the system and have access to all content and functionality

## Security

- The system shall allow employees to view the leave approvals by the principal or higher officials
- The system shall allow employees to manage the dynamic content visibility
- The system shall allow employees to view only their leave approval.

## Interoperability

- The system shall be able to import BOZ roster information into the course roster

## 2.2 Design document

### 2.2.1 Introduction

Detailed design starts after the system phase and system has been certified through the review. The goal of this phase is to develop internal logic of each of the modules identified during system design

In the system design, the focus is on identifying the modules, whereas during the detailed design the focus is on designing the logic for the modules. In other words in system design attention is on what components are needed, while in the detailed design how the component can be implemented in the software is the issue.

The design process for software system has two events. At the first level focus is on deciding which modules are needed for the system, the specification of these modules and how the modules should be interconnected. This is called system design or top level design. In the specification of the module can be satisfied is decided. This design level is often called detailed design or logic design, because the detailed design is extension of system design, system design controls the major structural characteristics of the system. The system design has a major impact testability and modifiability of a system and impacts its efficiency much of the design efforts for the designing software are spent creating the system design.

### 2.2.2 System design

In the flexibility of the uses the interface has been developed keeping a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

3. Administrative user interface
4. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Data updation along with the extensive data search capabilities. The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

## 2.2.2.1 Modules description

The system after careful analysis has been identified to be presented with the following modules:

### **The modules involved are:**

- Administration
- Employee
- Search
- Authentication
- Leave management

### **Administrator:-**

Administrator has the privileges to add all the Employees and register them in the organization and check the information of the Employee and check the status of the leave when they have taken and what type of leave they have taken and search is done based on the employee and report is generated based on employee.

### **Search:-**

This module contains complete search like Leave search, Type of Leave, Employee based on the leave and starting and ending day of leave.

### **Employee-**

In this module employee has the privileges to use his username and password for login and he can see the request given by the customer and he can pass the process to the Business Manager and maintain the record of the customers.

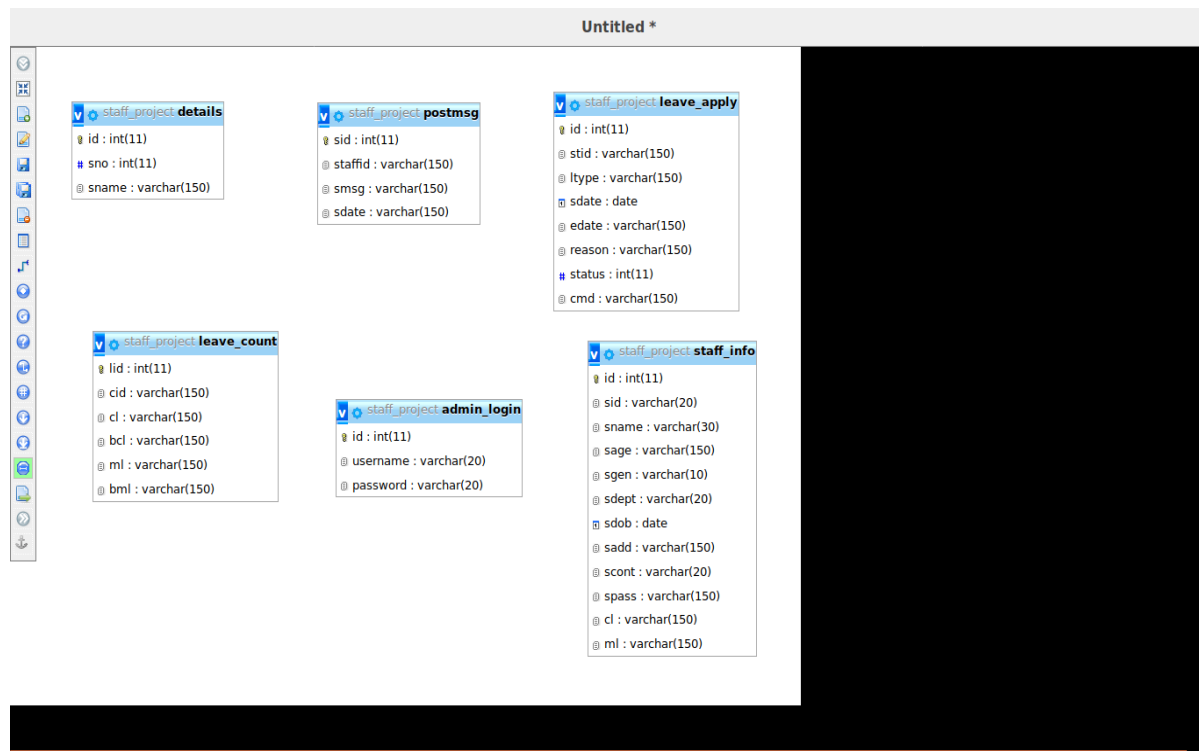
### **Reports:-**

This module contains all the information about the reports generated by the Employees based on the Performance and by the leave status.

### **Authentication:-**

This module contains all the information about the authenticated user. User without his username and password can't enter into the login if he is only the authenticated user then he can enter to his login.

## 2.2.2.2 Database design



## 2.3 Testing related documents

### 2.3.1 Introduction to System testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 2.3.2 Types of testing

#### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and

test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### **Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### **White Box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

### **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as



specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **3) Preparation of Software Configuration Management and Risk Management related documents.**

#### **3.1 Software Configuration Management**

The overall objective of a Software Configuration Management (SCM) Plan is to document and inform project stakeholders about CM with the project, what CM tools will be used, and how they will be applied by the project to promote success. The CM Plan defines the project’s structure and methods for

- Identifying, defining, and base lining configuration items (CI)
- Controlling modifications and releases of CIs
- Reporting and recording status of CIs and any requested modifications
- Ensuring completeness, consistency, and correctness of CIs
- Controlling storage, handling, and delivery of the CIs

##### **3.1.1 Configuration Management Activities**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

## 3.1.2 Software and Hardware Requirements

### Software Requirements:

Operating System: Windows XP or later

Database : Microsoft SQL Server-2005

Front end : JAVA,JSP

### Hardware Requirements:

Processor: Intel Pentium or More

Ram: 512 MB Ram

Hard Disk: PC with 20GB

## 3.2 Risk Management document

Risk is defined as the possibility of any negative occurrence that may happen due to external or internal factors, and that may be mitigated through preventive actions. All projects are subject to risks. In fact, there is an infinite number of things that might prevent you from achieving your goals when working on a project. Risk management minimizes those threats that could cause project failure, and allows you to stay in control of your project's schedule, budget and quality requirements.

Risk management can be divided into four processes:

- **Identification:** Detect risks that might prevent you from achieving your project's goals.
- **Analysis:** Determine what risks are the most dangerous.
- **Planning:** Plan for the most dangerous risks.
- **Monitoring and control.** Maintain the project's plan and continually identify risks.

Some of the risks encountered in Easy leave are:

- Form validation required, used JS for validation purpose
- Easy leave must be more scalable
- We implement various techniques to boost application stability.
- Also we use Java to ensure the quality of our code

In terms of risk management, such an approach allows us to identify and fix errors at the earliest stage, when they do not yet lead to losses, and easily maintain the code in the long term.

These are several common risks in website development, and some common approaches to solving them. Easy Leave is a web management tool for providing an online environment for leave

management. During our work to foresee risks and mitigate them. We also reduce the risk of failing to meet deadlines through efficient communication within our teams and through technical checks.

#### **4) Study and usage of any Design phase CASE tool**

##### **Study of CASE tool**

Computer aided software engineering (CASE) is the implementation of computer facilitated tools and methods in software development. CASE is used to ensure a high-quality and defect-free software. CASE ensures a check-pointed and disciplined approach and helps designers, developers, testers, managers and others to see the project milestones during development.

CASE can also help as a warehouse for documents related to projects, like business plans, requirements and design specifications. One of the major advantages of using CASE is the delivery of the final product, which is more likely to meet real-world requirements as it ensures that customers remain part of the process.

CASE illustrates a wide set of labor-saving tools that are used in software development. It generates a framework for organizing projects and to be helpful in enhancing productivity. There was more interest in the concept of CASE tools years ago, but less so today, as the tools have morphed into different functions, often in reaction to software developer needs. The concept of CASE also received a heavy dose of criticism after its release.

##### **CASE Tools:**

The essential idea of CASE tools is that in-built programs can help to analyze developing systems in order to enhance quality and provide better outcomes. Throughout the 1990, CASE tool became part of the software lexicon, and big companies like IBM were using these kinds of tools to help create software. Various tools are incorporated in CASE and are called CASE tools, which are used to support different stages and milestones in a software development life cycle.

##### **Types of CASE Tools:**

###### **Diagramming Tools:**

It helps in diagrammatic and graphical representations of the data and system processes. It represents system elements, control flow and data flow among different software components and system structure in a pictorial form.

For example, Flow Chart Maker tool for making state-of-the-art flowcharts.

###### **Computer Display and Report Generators:**

It helps in understanding the data requirements and the relationships involved.

## **Analysis Tools:**

It focuses on inconsistent, incorrect specifications involved in the diagram and data flow. It helps in collecting requirements, automatically check for any irregularity, imprecision in the diagrams, data redundancies or erroneous omissions.

## **Central Repository:**

It provides the single point of storage for data diagrams, reports and documents related to project management.

## **Documentation Generators:**

It helps in generating user and technical documentation as per standards. It creates documents for technical users and end users.

For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

## **Code Generators:**

It aids in the auto generation of code, including definitions, with the help of the designs, documents and diagrams.

## **Advantages of the CASE approach:**

As special emphasis is placed on redesign as well as testing, the servicing cost of a product over its expected lifetime is considerably reduced.

- The overall quality of the product is improved as an organized approach is undertaken during the process of development.
- Chances to meet real-world requirements are more likely and easier with a computer-aided software engineering approach.
- CASE indirectly provides an organization with a competitive advantage by helping ensure the development of high-quality products.

## **Disadvantages of the CASE approach:**

- **Cost:** Using case tool is a very costly. Mostly firms engaged in software development on a small scale do not invest in CASE tools because they think that the benefit of CASE are justifiable only in the development of large systems.
- **Learning Curve:** In most cases, programmers productivity may fall in the initial phase of implementation, because user need time to learn the technology. Many consultants offer training and on-site services that can be important to accelerate the learning curve and to the development and use of the CASE tools.

- **Tool Mix:** It is important to build an appropriate selection tool mix to urge cost advantage  
CASE integration and data integration across all platforms is extremely important.

## Usage of any CASE tool

**StarUML** is the tool we suggest because it makes a clear conceptual distinction between models, views and diagrams. A Model is an element that contains information for a software model. A View is a visual expression of the information contained in a model, and a Diagram is a collection of view elements that represent the user's specific design thoughts.

StarUML is build as a modular and open tool. It provides frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta-model and tool through COM Automation, and it provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

StarUML supports the following diagram types

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Collaboration Diagram
- State chart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Composite Structure Diagram

The user interface is intuitive. On the upper right side, a window allows to rapidly navigate between all the content of a project, adopting either a model or a diagram view. Multiple diagrams can be open at the same time and tabs allow switching rapidly between views. The lower right window allows to document the current diagram, either with plain text or attaching an external document. During diagram editing, "wizards" are located around the object that give you the quick shortcuts to main associated tasks with your current operation, like adding an attribute when you create a class for instance. A right-click on the mouse brings the full set of operations at your disposal.

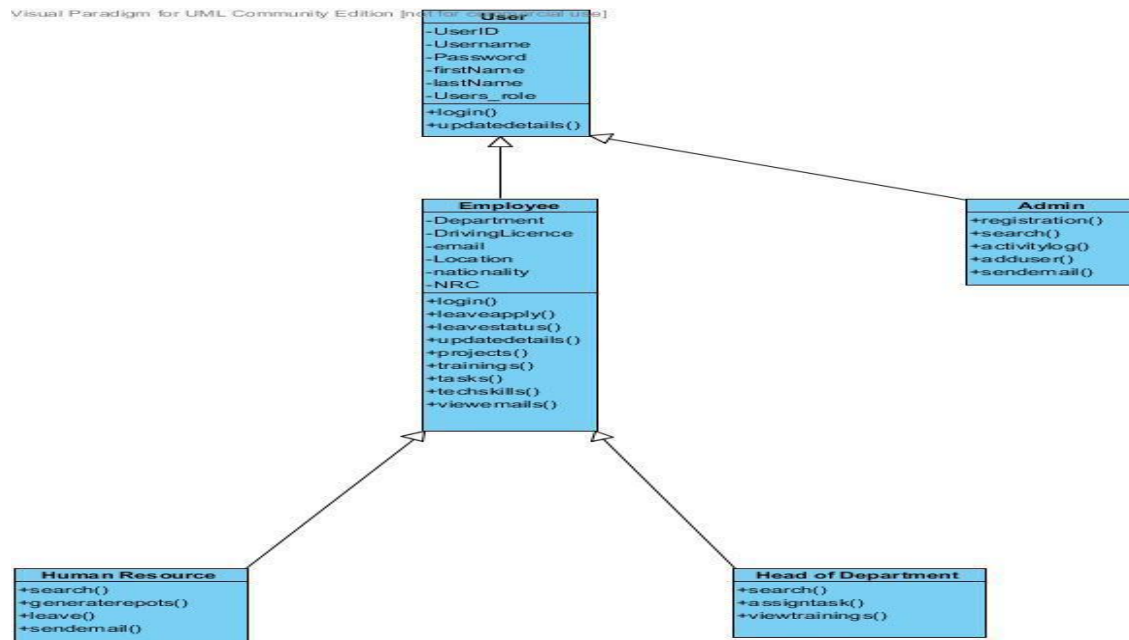
StarUML has also a model verification feature. You can export diagram in different formats (jpg, bmp, wmf). It also supports a patterns approach and import of Rational Rose files. StarUML

Generator is platform module to generate various artifacts (like as Microsoft Word, Excel, PowerPoint, and Text-based artifacts) by templates depending on UML model elements in StarUML. The users can define their own templates and can apply many different kinds of templates to the same UML model, so the users can get various artifacts automatically, easily and fast. The tool supports code generation and reverse engineering for Java, C# and C++. StarUML has many powerful features and is certainly more than a "simple" diagramming tool. With its support of MDA (Model Driven Architecture), it is more aimed at people using UML in an intensive way and with some code generations objectives than for simply drawing diagrams to document requirements. However, using StarUML just as a diagramming tool work fine, especially on Windows as the tool is built with Delphi and might execute faster than the Java-based tools.

## 5) Performing the Design by using any Design phase CASE tools.

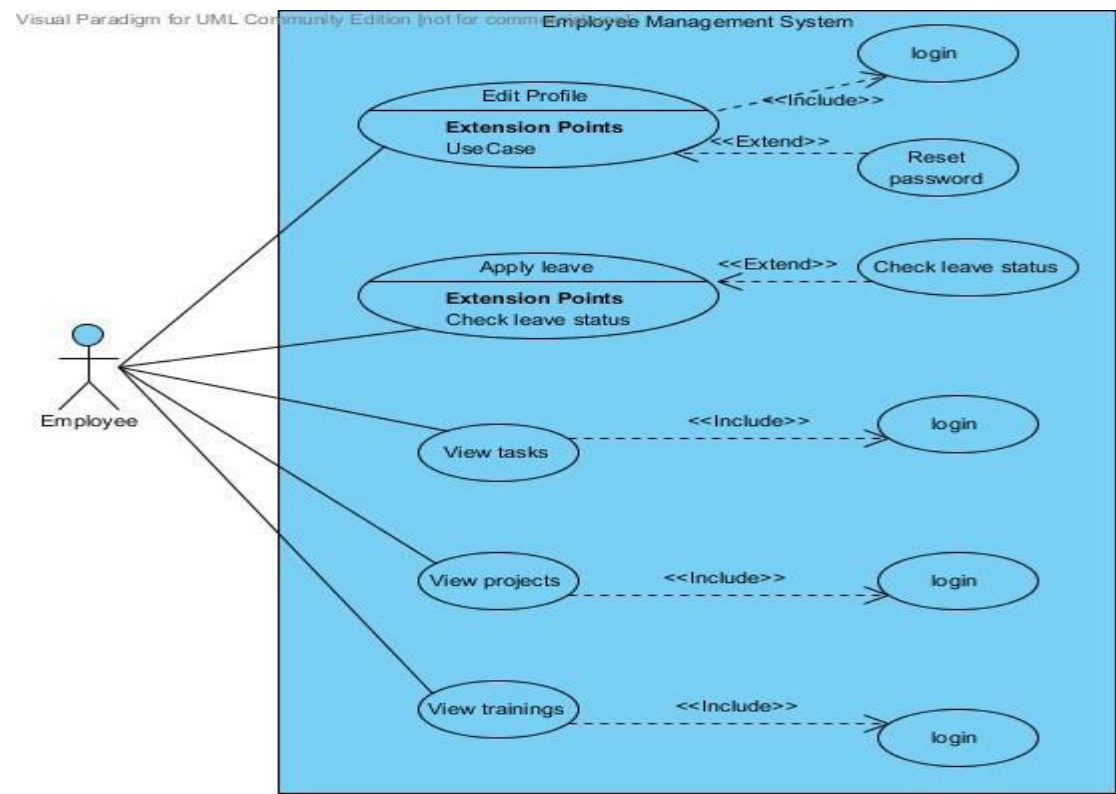
### 5.1 Design phase in Easy leave

#### 5.1.1 Class Diagram for Employee, User ,Admin, HR & HOD

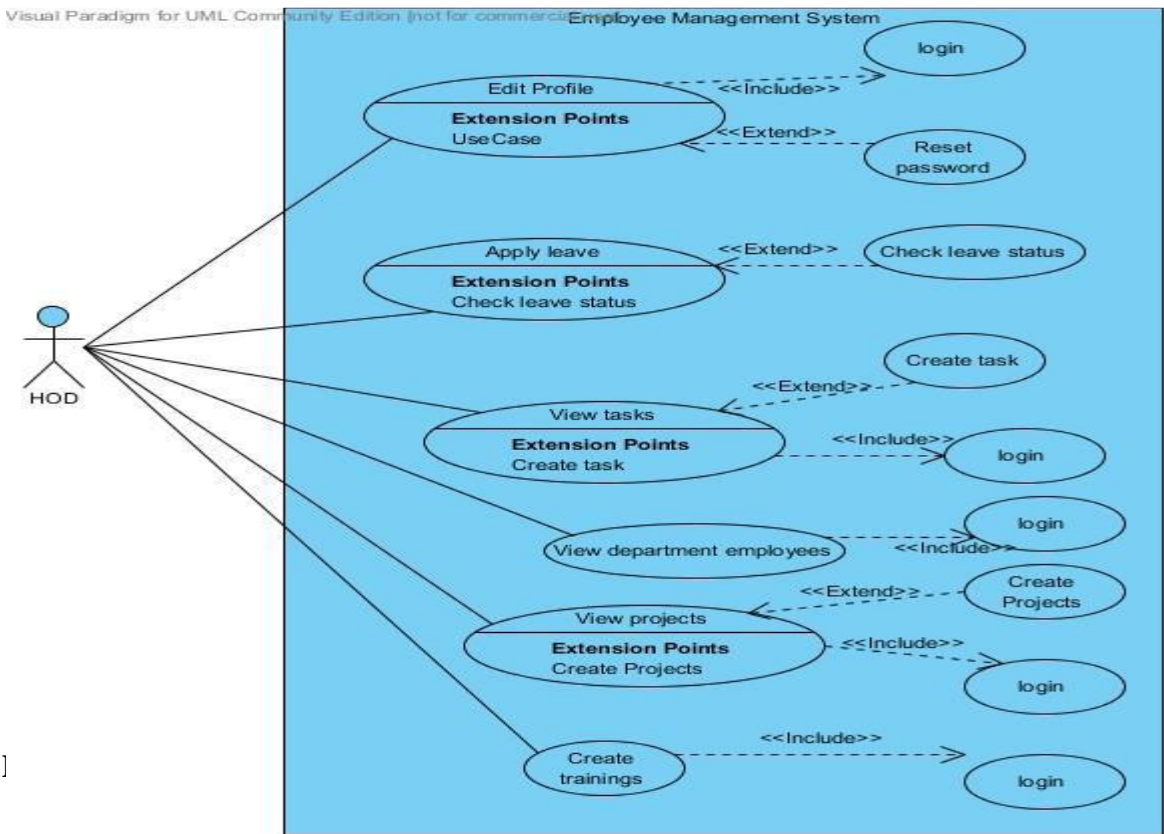




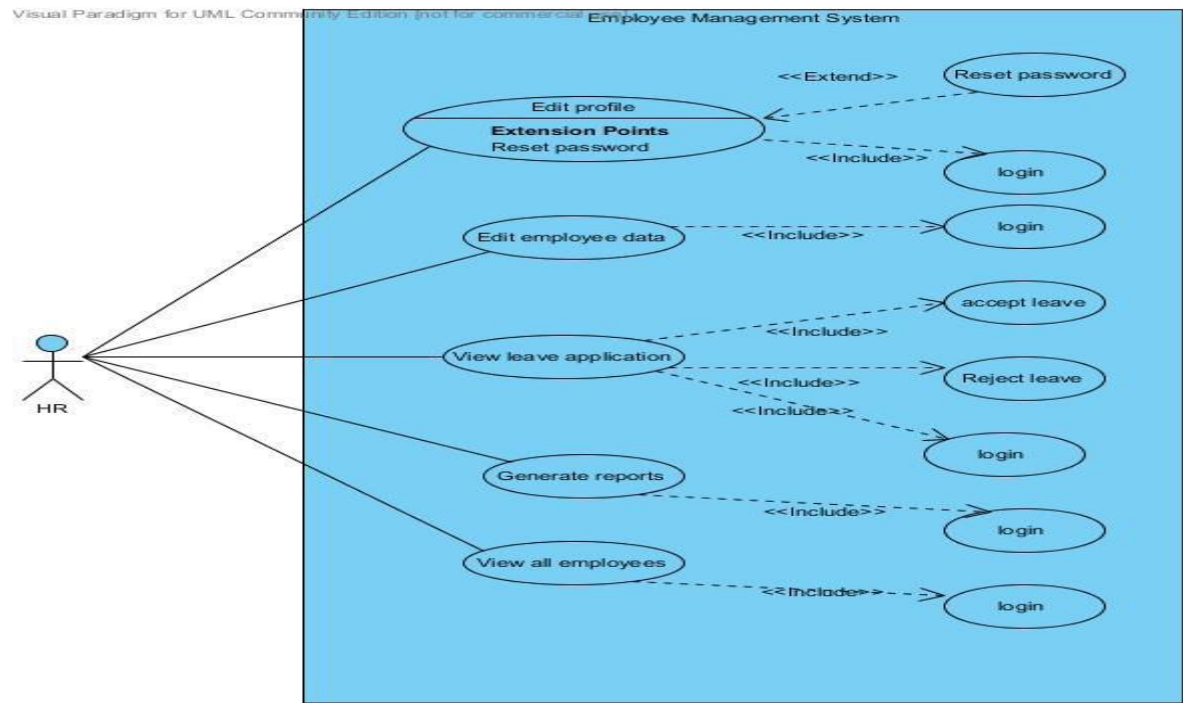
5.1.2 Use case diagram for Employee



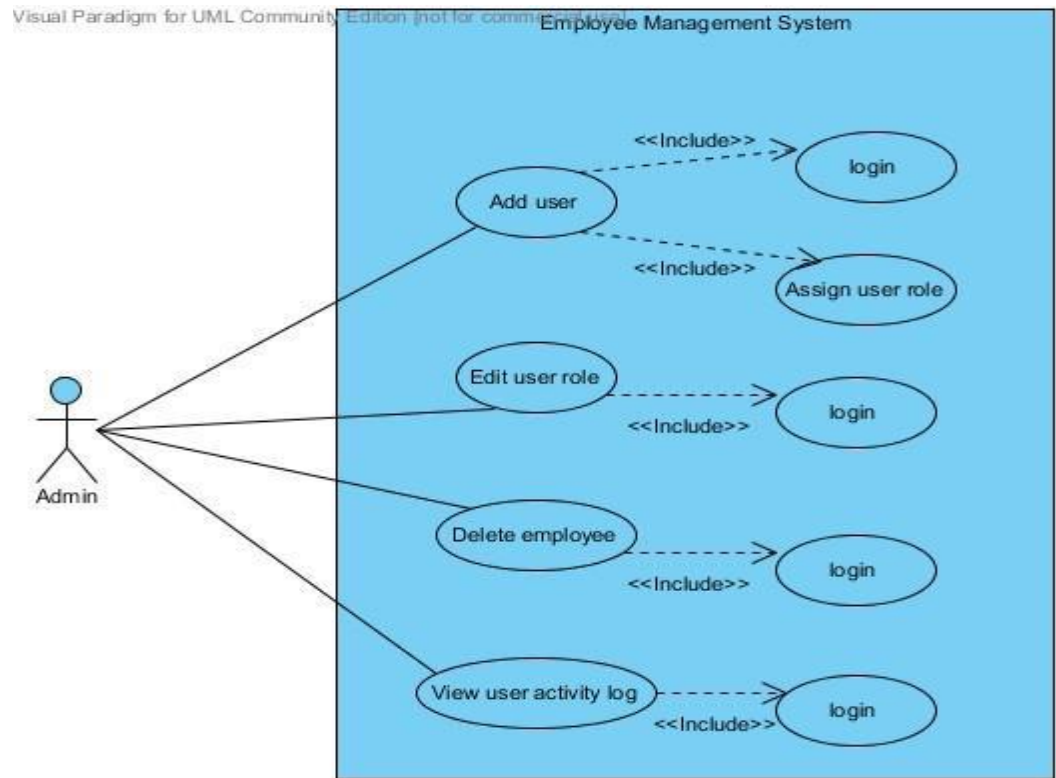
5.1.3 Use case diagram for HOD



5.1.4 Use case diagram for HR

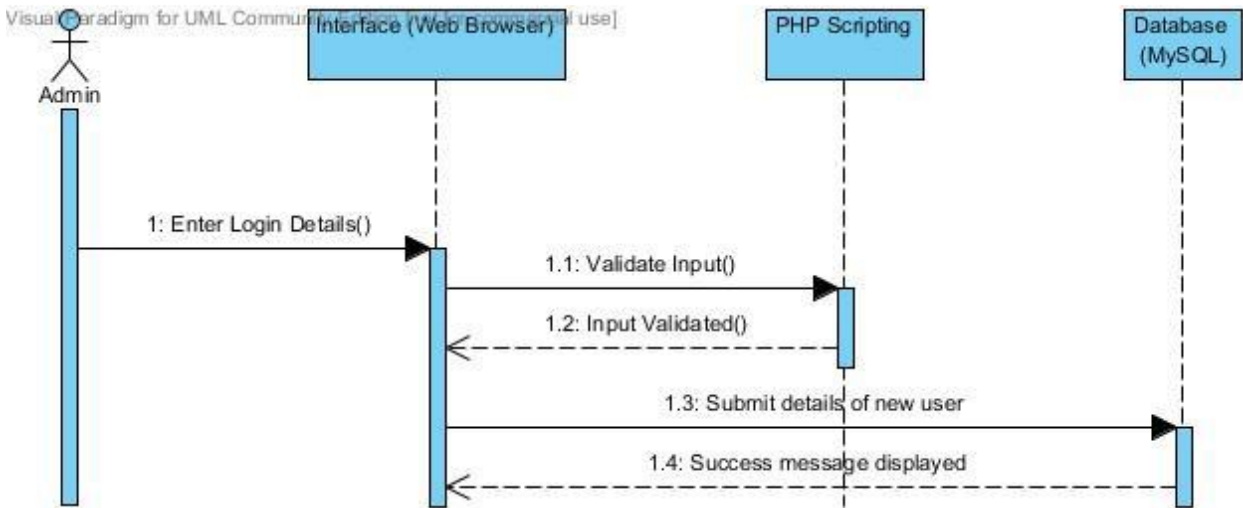


5.1.5 Use case diagram for Admin

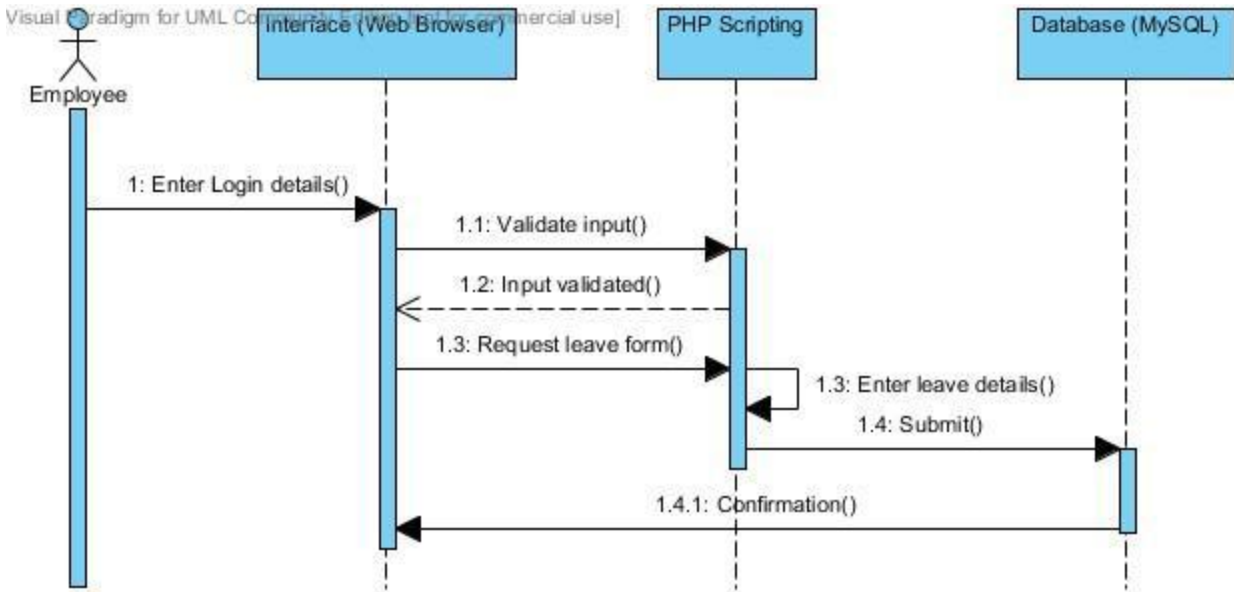




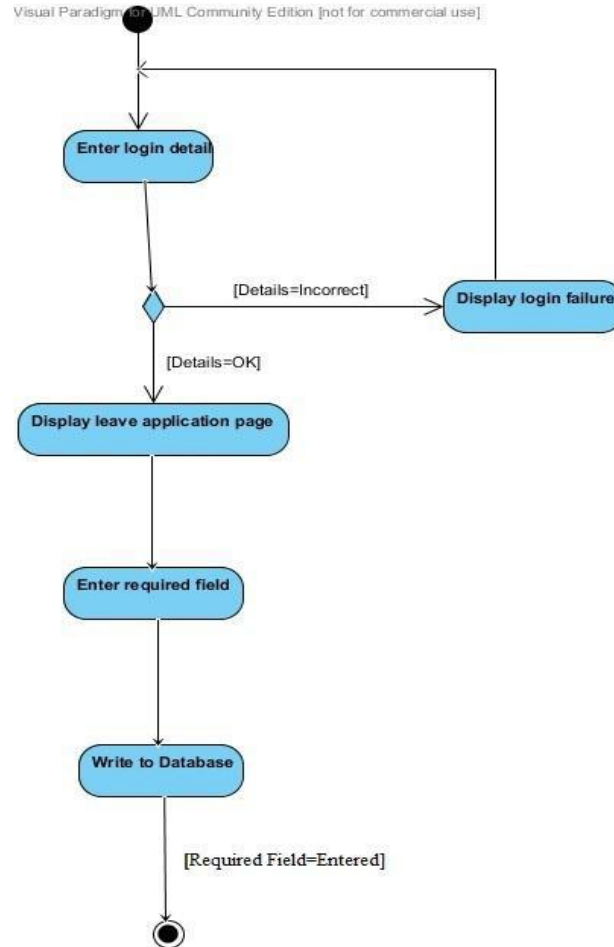
5.1.6 Sequence diagram to add new employee



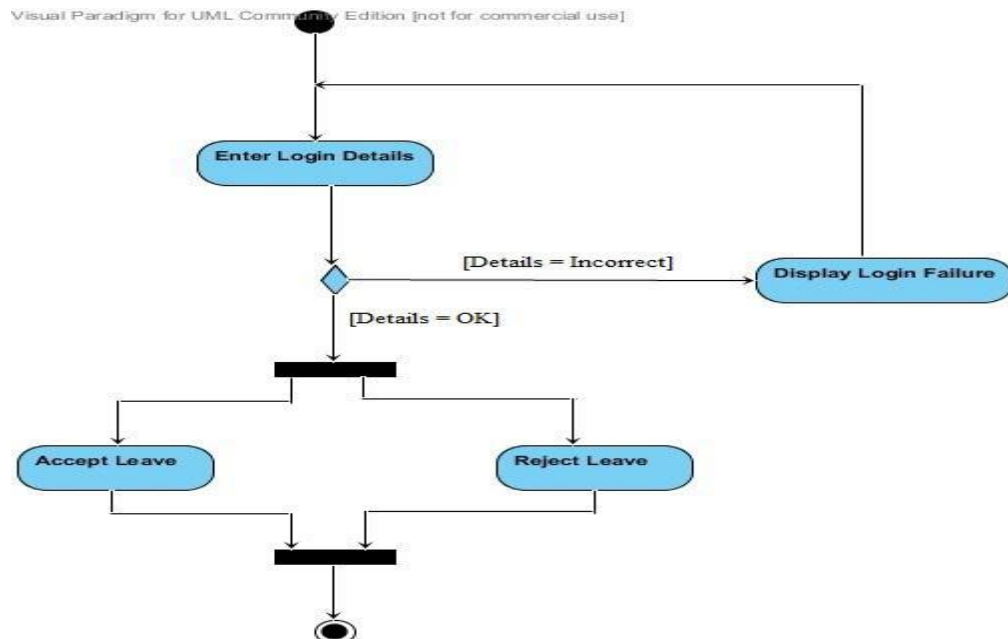
5.1.7 Sequence diagram for leave application



### 5.1.6 Activity diagram for Leave application



### 5.1.7 Activity diagram for leave rejection



## 6) Test cases for unit testing and integration testing

### 6.1 Testing login for Easy leave



The screenshot shows a web browser window titled "Login Page". The page has a light blue background with the text "Easy Leave" in blue. Below this, the word "Login" is written in blue. A central white box contains the login form with the following fields: "User Name" with the value "admin", "Password" with masked characters "\*\*\*\*\*", and "User Type" with a dropdown menu showing "Admin". At the bottom of the form is a button with a user icon and the text "Login".

- Check if the username field accepts valid username and password field accepts valid password.
- Check if the wrong username and valid password allows access to any specific account.
- Check if the valid username and wrong password allows access to any specific account.
- Check if the forgot username link leads to username recovery page.
- Check if the forgot password link leads to the password recovery page.
- Check if the invalid username and password triggers any warning.
- Check if the invalid credentials open the random account.
- Check if the user is logged in, allows you to logout by using the link at the bottom of the application.
- Check if the logout link functions as expected.

## 6.4 Testing employee leave management in Easy leave

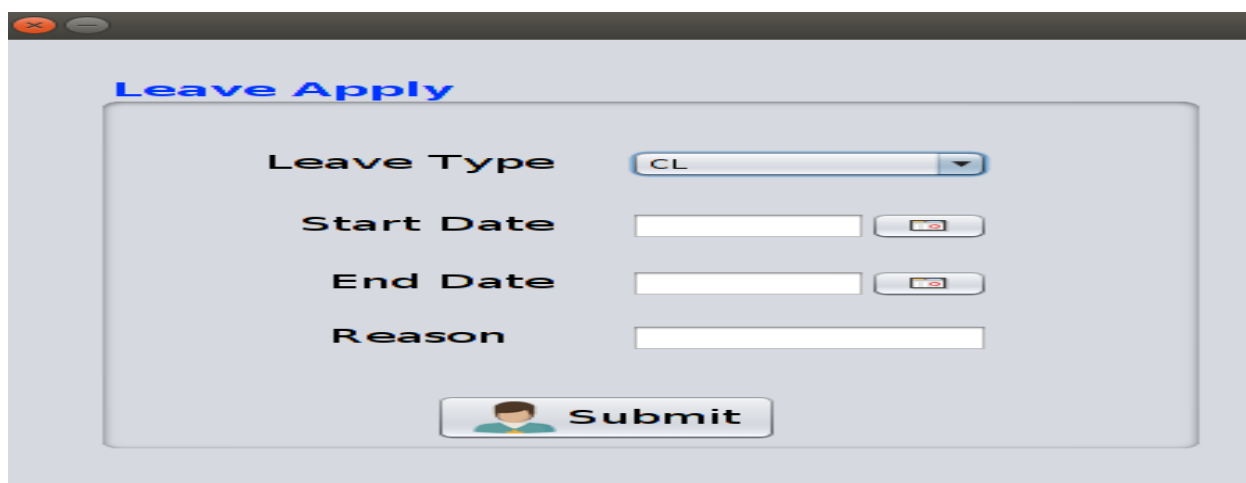
This screenshot shows how admin will add staff and managing their leaves



- Check if the employee applied leave by using his EMP ID.
- Check if the leave details are updated.
- Check if the employee details are updated.
- Check if the faculty details and their leaves can be modified if you are admin.
- Check if the new employee is possible to add into the system using add staff button.
- Check if the staff can be removed using delete employee feature.

## 6.5 Testing Apply leave

This screenshot provides the details about employees leave applications.



- Check if the type of leaves will get displaying in the drop down.
- Check if date is getting stored in date format
- Check if you can submit the data

## 6.6 Testing Sending messages in Staff leave management



The screenshot shows a web application titled "Staff Leave Management". It features a "Select All" button at the top left. Below it is a list of employees with checkboxes for selection:

Employee ID	Employee Name
<input type="checkbox"/> CS1001	Ram
<input type="checkbox"/> CS1002	Sam
<input type="checkbox"/> ME1001	Latha
<input type="checkbox"/> CS1003	Raja
<input type="checkbox"/> EE1001	Bala
<input type="checkbox"/> EE1002	Mahesh
<input type="checkbox"/> EE1003	Nikitha
<input type="checkbox"/> CS1004	mahesh

To the right of the list is a "Message:" label above a text input field. Below the input field are two buttons: "Clear" and "Send".

- Check if the admin selecting the employee and by entering the message can leave the message
- Check if multi select is working is not
- Check if the admin click on send the message then the message is delivered to the staff or not.

## 7 ) Test cases for black box and white box testing

### WHITE-BOX TESTING

White box some times called “Glass box testing” is a test case design uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests where made on the system:

- All independent paths within a module have been exercised once.
- All logical decisions were checked for the truth and falsity of the values.

## BLACK-BOX TESTING

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program.

- 1) Interface errors
- 2) Performance in data structure
- 3) Performance errors
- 4) Initializing and termination errors

## TESTCASES

### Admin login form

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	Enter valid admin username and password	Easy Leave should display home window	Successful
2	Enter invalid username and password	EasyLeave should display incorrect username and password	Successful

### Add Employee page

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	On click of “New”	Admin need to fill the details of employee and click on submit then Easy Leave should display course is created successfully	Successful
2	On click of “Update”	If admin wants to update the employee details , after updating the required fields and click on update then Easy Leave should display employee details is updated successfully	Successful
3	On click of “Delete”	If Admin wants to delete the employee details then after selecting the EMPID then Easy	Successful

		Leave must display employee details deleted successfully	
4	On click of “Send Messages”	If admin wants to send message to faculty then after checking employees and entering the text in the text area clicks on end then Easy leave must send message to employee	Successful

**Faculty page**

SNO	TEST CASE	EXPECTED RESULT	TEST RESULT
1	On click of “Leave request”	Faculty need to fill the form after selecting the leave type then Easy leave should display material uploaded successfully	Successful
2	On click of “Reports”	If faculty need to check this leave approvals then after clicking on Reports Easy Leave should display Leave approvals by higher officials(Admin)	Successful

\*\*\*