**THE SUPERIOR COLLEGE LAHORE**

|  |
| --- |
| chk_captcha |

**Faculty of Computer Science & IT**

**Department of Software Engineering**

**Final Year Project**

**PROJECT REPORT (Part-1)**

**Leave Management System**

Project ID: **[write ID here Issued by FYP Manager]**

**Project Team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Name** | **Student ID** | **Program** | **Contact Number** | **Email Address** |
| Muhammad Saad | Bcsm-f16-529 | BSCS | 03026661234 | ms8739060@gmail.com |
| Muhammad Usama | Bcsm-f16-525 | BSCS | 03119529988 | usamajilani4050@gmail.com |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Sir. Ahmed Bilal**

(lecturer)

# Chapter 2

# Software Requirement Specifications

**Chapter 2:** Software Requirement Specifications



## Introduction

## Purpose

The main objective of the leave management system is to build an application program to reduce the manual work for managing the leave, leave type. A Leave Management System(LMS) automates the leave request process, making it hassle-free for both the management and the employees. This solution provides all the stages involved in leave management process: application, approval/rejection, filing leaves, managing leave balance, and analysis. This is a platform that enables a company to easily and correctly allocate, track and grant leave as well as allow for employees to request and track their own leave. Leave management systems make it easy for an employee to request leave from their own desk or even from home if sick. The employee can also attach a doctor’s certificate to the requested leave application if required.

## 

## Document Conventions

**Main section:**

* Font: Calibri (body)
* Face: Bold
* Size: 14
* Line Spacing: 1.5
* Text align: Justified

**Sub section:**

* Font: Calibri (body)
* Face: Bold
* Size: 12
* Line Spacing: 1.5
* Text align: Justified

## Intended Audience and Reading Suggestions

The document is intended to be a guide for developers, users, testers and students. The goal of this document is to identify the requirements of Leave Management System web application.

**Developers:**

A developer may directly want to know the requirements by reading the section of USER and SYSTEM REQUIREMENTS and skipping other sections if required.

**Users:**

User can use this document to see features and functionalities, and their usage as a guide.

**Students:**

Students can use this document for making projects like Leave Management system.

## Product Scope

In the existing leave record system every department in the organization follow manual procedure in which faculty enters information in a record book. At the end of the month/session ,Administration department calculates leave of every members which is time taking process and there is a chance of losing data or error in the record.

This module is a single leave management system that is critical for organization HR tasks and keeps the record of vital information like leaves and balancing the leaves. It intelligently adopt to organization HR policy of the management and allow employees and their managers to manage leaves and replacement(if required).

## References

There are following references that are utilize in system requirement specification.

* A use case and fully dressed use cases are available in part 3 along with product description and product features.
* System features are describe in section: **2.3**

## Overall Description

## Product Perspective

It is aimed at replacing the tedious paper works that the companies or organization use. The system will collect data and store it for easy access. The system will provide users with complete record of the employee leaves. It will also provide information about the leave balance (availability). The system is thus helpful to reduce the time and complexity of maintaining the records.

## Product Functions

There are some major leave management system functionalities are as follow:

* Sign-up/sign-in
* Maintain employee leave record
* Apply for leave
* Approve or reject leave application
* Send notification
* Edit profile

## User Classes and Characteristics

Table 1.1user classes and characteristics

|  |  |
| --- | --- |
| **Class** | **Characteristics** |
| **Admin** | Admin class maintains and manages all stuff e.g. employee accounts management, employee queries. |
| **User** | User class has access to see the leave availability, notifications, can edit profile. Account sign-up, apply for leave, and access to all functionalities available for employee. |
| **HOD** | Can approve or reject the leave application. |

## Operating Environment

Leave management system is a web based application, so it follow the following technologies

* Support in every web browser like Chrome, Firefox , Edge/Internet Explorer.
* Internet connection.

## Design and Implementation Constraints

**Design Constraints:**

User interference design should be simple and easy to understand. There should be High performance system design. Website interference design should be user friendly and easy to use and learn. There should be very fast response time of the system.

**Implementation Constraints:**

Coding standard will be followed by the programmer. Code must be clean and easy to readable. System should be developed within the given time and budget.

## User Documentation

We will be provide the user manual separately for this Leave management web application in the form of website road map.complete user manual will be provided with this document later and that can be accessible through working system.

## Assumptions and Dependencies

There are following assumption and dependencies that the system should have:

* User must be familiar with computer.
* User must understand the English language.
* User must be familiar with Internet .
* User must know how to connect the internet.
* User must be familiar with the web pages.
* User know the database.

## External Interface Requirements

## 

## User Interfaces

The user interface of this system is attractive ,fully responsive,user friendly, modern and fully compatible with any type of browser that can connect with internet Explorer with version 10,Firefox version 16 and above, Microsoft edge version 35 and above, safari with version 6 and other mobile browsers with IOS 11 and above and with android OS 6 and above.

## 

## Hardware Interfaces

This app shall run on:

* Processor : Intel(R) Core (TM) i5-4210U CPU @ 1070GHz
* Memory: 4 GB RAM and 50 MB hard or more

## Software Interfaces

There are following Software interface for leave management system web app:

* Operating system : Any
* Web browser
* HTML 5
* CSS 3
* PHP

## Communications Interfaces

The communication between different component is very important because components are dependent to each others.This web app supports Google Chrome and Mozilla Firefox.The app involves FAQ forms for the requesting information, queries and problems etc.This require the suitable internet connection.

## System Features

This LMS contain the following features:

* Sign-up/sign-in
* Maintain employee leave record
* Apply for leave
* Approve or reject leave application
* Send notification
* Edit profile
* Logout

## Sign-up

## Description and Priority

This system will provide efficient and secure facility of employee registration. Employee will enter all the details into the sign-up/registration form and submit it . After this admin will approve the employee to sign-in the system. The priority of this feature is high because it is used for entering into the system.

## Stimulus/Response Sequences

Table 1.2 user sign-up

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| User will Enter Required detail in the particular input fields, select options and click on submit button. | System will get data from sign up form. |
| System will validate entered data according to defied set of rules. |
| If entered data is not valid, system will display error massage on the particular field with invalid input. |
| If entered data is valid, System will verify email entered by employee. |
| If entered email already exists, system will display an error message saying “Email already exists”. |
| If entered email does not exist, system will save all detail entered by employee. |
| Then system will display employee’s profile. |

## Functional Requirements

REQ-SF1-1: User click on sign-up button.

REQ-SF1-2: System will display sign-up page.

REQ-SF1-3: User enter the text in input fields.

REQ-SF1-4: User name field only accept alphabets.

REQ-SF1-5:User Email field only accept email format.

REQ-SF1-6:User password field only accept alphanumeric text.

REQ-SF1-7:User password field hide with dots.

REQ-SF1-8:User password field accept only 8 characters.

REQ-SF1-9:System display the option of gender.

REQ-SF1-10:User can select only 1 option.

REQ-SF1-11:Phone number field only accept numeric value.

REQ-SF1-12:System validate all the data entered by user.

REQ-SF1-13:System verify the existence of email entered by user.

REQ-SF1-14:System display the message if email is already exist.

REQ-SF1-15:System will save the details and login to user if email is already not exist.

## Login

## Description and Priority

System will provide the facility to login in the system. User enter the login details like email and password and click on login button. The priority of the system is **High** because user wants to enter into the system and access their account.

## Stimulus/Response Sequences

Table 1.3 Login

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| User will Enter email and password in respective input fields and click the login button. | System will get email from email input field of login form. |
| System will get password from password input field of login form. |
| System will validate entered data according to defied set of rules. |
| If entered data is not valid, system will display error massage on the respective field with invalid input. |
| If entered data is valid, System will verify email and password entered by user. |
| If entered email and password matches with the record, Then system will display user’s profile. |
| If entered email and password does not match, system will display an error message saying “Invalid Email or Password”. |

## Functional Requirements

REQ-SF1-1: User will click on login button.

REQ-SF1-2: System will display the login form.

REQ-SF1-3: User enter the text in input fields.

REQ-SF1-4: User Email field only accept email format.

REQ-SF1-5:User password field only accept alphanumeric text.

REQ-SF1-6:User password field accept only 8 characters.

REQ-SF1-7:User will click on login button.

REQ-SF1-8:System will validate all the text entered by user.

REQ-SF1-9:System verify and match the Email entered by user.

REQ-SF1-10:System verify and match the password entered by user.

REQ-SF1-11:System will load the user dashboard after matching the Email and password with database.

REQ-SF1-12:System will generate the appropriate message if Email or password incorrect entered by user.

## Logout

## Description and Priority

System give the facility to logout from the system.user will click on logout button and system will destroy the session of the user and loads the login page. The priority of the this feature is **High**  because user will use the system according to their need.

## Stimulus/Response Sequences

Table 1.4 Logout

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| User clicks on “logout” button | System will destroy user’s session. |
| System will load login page |

## Functional Requirements

REQ-SF1-1:User will click on logout button.

REQ-SF1-2: system will destroy the session.

REQ-SF1-3: System will redirect to the login page.

## Apply for Leave

## Description and Priority

System provides the facility to apply for leave. User click on apply for leave option system display the page apply for leaves and user will enter the details for applying the leave. The priority of this feature is **HIGH**.

## Stimulus/Response Sequences

Table 1.5 apply Leave

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| User will Enter the details in respective input fields and click the Apply button. | System will get the details from field of apply form. |
| System will get the date from date and leave type from input field of apply form. |
| System will validate the date . |
| If entered data is not valid, system will display error massage on the respective field with invalid input. |
| If entered data is valid, System will save the record and submitted notification occur. |

## Functional Requirements

REQ-SF1-1:User will click on apply leave option.

REQ-SF1-2: System display the apply for leave page.

REQ-SF1-3: User enter the details in respective fields of apply leave form.

REQ-SF1-4:Leave type drop down should be in alphabet.

REQ-SF1-5:Apply leave Date must be in Date format.

REQ-SF1-6:Apply leave description field must be in alphanumeric.

REQ-SF1-7:System save the form record and display the message of confirmation.

## Leave History

## Description and Priority

User will be able to see the Leave History. The priority of this feature is medium.

## Stimulus/Response Sequences

Table 1.6 leave History

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| User clicks on leave history option | Leave History is displayed |

## Functional Requirements

REQ-SF1-1: User will be able to see the history of leave Applications.

## Admin

## Description and Priority

Admin will be able to do different task. The priority of this feature is medium.

## Stimulus/Response Sequences

Table 1.7 admin task

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| Admin add employee. | Employee are displayed |
| Admin delete employee. | Employee are removed |
| Admin edit employee. | Employee are displayed |
| Admin add leave type. | Specific course displayed on specific teacher profile |
| Admin remove leave type. | Leave type removed |
| Admin edit leave type. | Leave type are displayed |
| Admin add department. | Department are displayed |
| Admin remove department. | Department are removed |
| Admin edit department. | Departments are displayed |
| Admin view the leaves. | Leave are displayed |

## Functional Requirements

REQ-SF1-1: Admin will be able to Add/Remove/Edit employee.

REQ-SF1-2: Admin will be able to Add/Remove/Edit leave type.

REQ-SF1-3: Admin will be able to Add/Remove/Edit Department.

REQ-SF1-4:Admin will be able to view the leave record.

## HOD

## Description and Priority

Head of department will be able to view the leave record ,employee details,approve or reject the leave applications.the priority of this feature is **HIGH.**

## Stimulus/Response Sequences

Table 1.8 HOD task

|  |  |
| --- | --- |
| **User Actions** | **System Responses** |
| HOD can view the leave application | System display the record of applications. |
| HOD can approve or reject the applications | System update the application record. |

## Functional Requirements

REQ-SF2-1:HOD will be able to view the record.

REQ-SF2-2:system display the leave record.

REQ-SF2-3:HOD will be able to approve or reject the leave applications.

REQ-SF2-4:system update the record and send the notification to applicant.

## Other Nonfunctional Requirements

## Performance Requirements

* System will not take much time to process on user queries.
* System will take 3 millisecond to process on user queries that submitted by user.
* The system shall be able to display the confirmation massage within 4 millisecond when the user submit the information to system.

## Safety Requirements

There are following safety requirement for the system

* User details must be verified.
* Their is a backup of database.
* User must be authenticated.
* Use strong password make sure that password at least 8 character long and consist of some symbols
* Users details manage properly.
* In case of any problem user can communicate with the system.

## Security Requirements

There are following security requirement for the system.

* During registration user must be enter validate email address.
* Password is hide with dot when the user enter the password.
* After enter the Email and password user get the access of their profile.
* Password shall be stored in database as hash value.
* Information about the user must be secure and safe.
* Transformation of data through HHTPS and SSL, so that data is encrypted during transfer.
* User private details kept private.
* User can be registered after verification.

## Software Quality Attributes

There are following software Quality Attribute of the system.

**Reliability:**

* System should be handle up to 500 concurrent user.
* System should be able to display the correct result of user queries.

**Usability:**

* System should b easy to use.
* User interface should be simple and cleared for better understanding to end user.
* System should be user friendly.
* System design should be easy for new and infrequent user to learn.
* System shall be stable.

**Maintainability:**

* System should be develop in such a way that new code can be added in existing code.
* It should be easy to upgrade new feature and new technologies from time to time.

**Portability:**

* Design the system in such a way that could run on different devices like laptop ,mobile and tablet.
* There should not arise the technical issue when system run on different devices.

**Correctness:**

* System should display the correct and accurate details of the user.
* System should display the accurate and appropriate results of user queries.

**Availability:**

* System should be available 24/7.

**Testability:**

* System should be bugs free.
* There is no changes occur when the system test in different environment.

**Flexibility:**

* System should be flexible when third party component added into the system.
* System should be flexible to enough modification.

**Robustness:**

* System will handle the crashed situation.

## Business Rules

There are following business rules for this system.

* User followed the terms and condition of the system.
* User will accept the terms and condition of the system.
* User will sign-in and login-in in terms to use the system.
* User will read the system policy correctly.

## Other Requirements

* If system unable to process the on request then message or notification displayed.
* Use HTTPS protocols for secure communication.

# Chapter 3

# Use Case Analysis

**Chapter 3:** System Analysis

In this chapter we describe the structure of different use cases, we also describe that how different Actors interact with system.In this chapter we define the functionality of the system to identify, clarify, and categorize system requirements. We will discuss and will describe each case separately. Main functionalities of the system are defined which are divided into different modules. In major use cases functionalities of the system and the Actor activities are defined like View leaves ,Apply for Leaves,Approve or Reject the Leaves, sign-up, Login, Logout etc.

## Use Case Model

This web app is used for managing the leave Application. There are following function of our web based application.

* Sign-up
* Login
* Logout
* View Leaves
* Apply for Leave
* Approve or reject
* Manage employee/user
* Manage Leaves

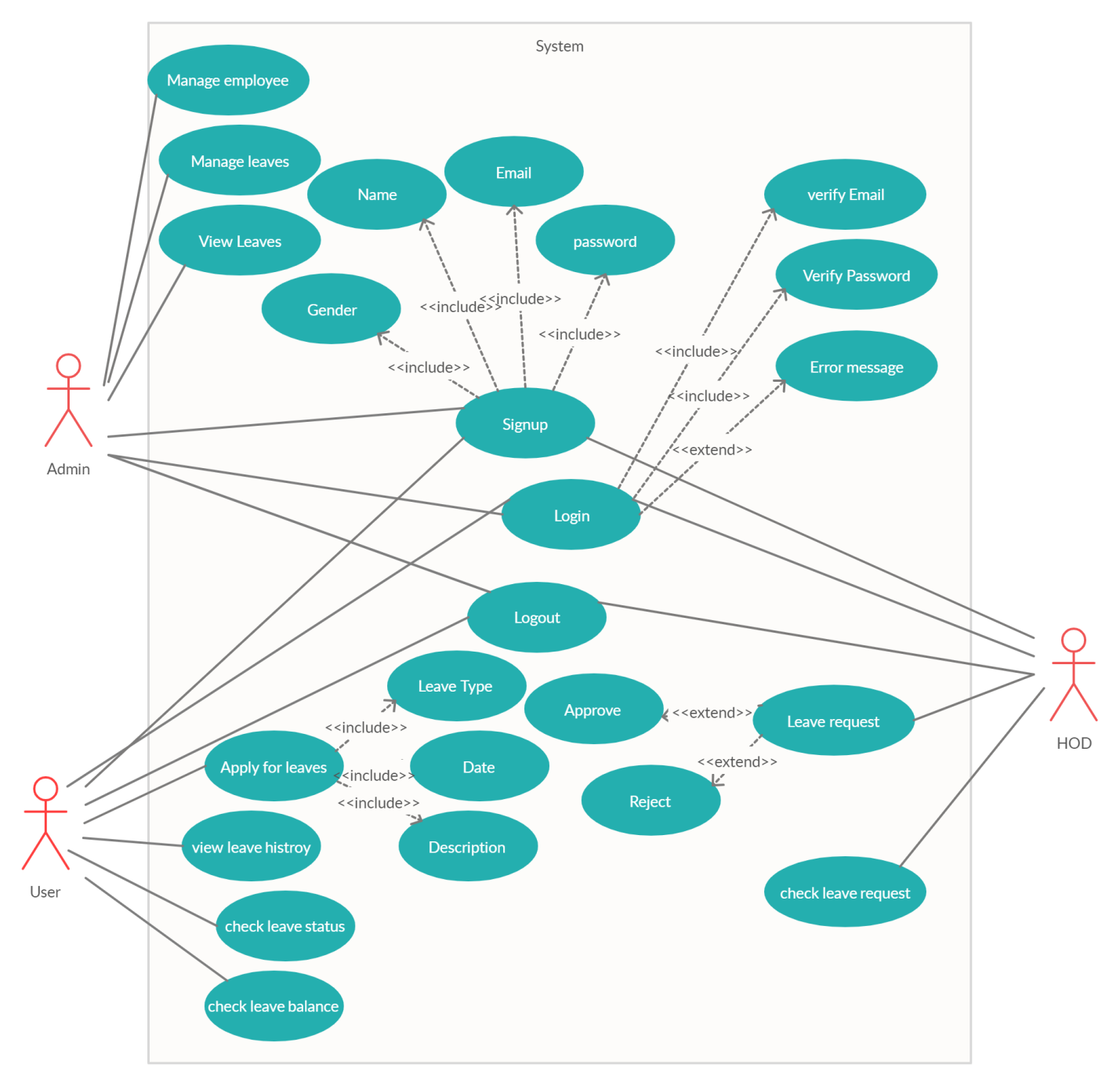


Figure 1.1 use case

## Fully Dressed Use Cases

There are following use cases of Leave Management System app are given below:

Table 2.1 sign-up fully dress use case

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Sign-up** | |
| **Level** | user | |
| **Scope** | LMS (organization website) | |
| **Primary Actor** | Employee, Admin , HOD | |
| **Stakeholders and interests** | **User**: He wants to make account for using the website so that he can access privately to his account and can update easily.  **Admin:** He is managing the website. He is allowing managing the website. He should have a unique identification for every user so that they can identify every user and know about the details of every event. | |
| **Precondition** | User must have a valid email, valid password, and valid name. | |
| Sign-up page is loaded. | |
| **Post condition** | User registered account is created. | |
| User can login his account. | |
| User can view his profile. | |
| **Main success scenario** | **User/Admin** | **System** |
| He requests to load the sign-up page of website. | Sign-up page loaded. |
| He enters name, email, dob, pass, gender. | Check valid or not. |
| Clicks the button” sign-up”. | System saves all the records. |
| **Extension** | **Website(registration page ) could not run successfully:**   * The system will display an error message “Reload the page”. * User either refreshes the page or tries again. * User tries to load the page after some time.   **Registration failed:**   * System displays message “registration is failed”.   **Provide details are not valid:**   * System details that is invalid. * User will correct the invalid details.   User will try to register again. | |
| **Special Requirements** | * User friendly interface. * Response time is less than 4s or less. | |
| **Technology and data variation list** | Use new web development technologies and framework. | |
| **Miscellaneous** | N/A | |

Table 2.2 login fully dress use case

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Login** | |
| **Scope** | LMS (organization website) | |
| **Level** | Employee, Admin , HOD | |
| **Primary Actor** | User | |
| **Stakeholders and interests** | **User:** He wants to login to his account. So that he can view the leave status,leave history and apply for leave.  **Admin:** He is for managing the website. He can view the leaves ,manage the employees. | |
| **Precondition** | User must registered. | |
| User must have a valid email, valid password. | |
| User must have click on the button “login”. | |
| **Post condition** | User is login to his account. | |
| User can view their activity. | |
| **Main success scenario** | **User/Admin** | **System** |
| He request to login. | Login page will open. |
| Enters password and email. Click login. | Validates and open the account. |
| User login successfully | User’s profile is loaded |
| **Extension** | **Website(Login page ) could not run successfully:**   * The system will display an error message “Reload the page”. * Tells the reason why it cannot run. * User either refreshes the page or tries again. * User tries to load the page after some time.   **User could not be login:**   * System displays message “logged in is failed”. * User tries again after some time.   **Provide details are not valid:**   * System highlights the details that are invalid. * User will correct the invalid details into the valid details.   User again will try to logged in. | |
| **Special Requirements** | User friendly interface.  Response time is less than 4s or less. | |
| **Technology and data variation list** | Use new web development technologies and frameworks for the development. | |

Table 2.3 logout fully dress use case

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Logout** | |
| **Scope** | LMS (organization website) | |
| **Level** | User | |
| **Primary Actor** | Employee, Admin , HOD | |
| **Stakeholders and interests** | **User:** He wants to logout to his private account.  **Admin:** He is for managing the website. Only he is allowing managing the website. For managing the website he has to login to his account, to know about the events that are related to the particular user. | |
| **Precondition** | The page for logging the system to logout the account. | |
| User must have click on the button “logout” | |
| **Post condition** | User logout his account. | |
| **Main success scenario** | **User/Admin** | **System** |
| User logout the system. | Logout page will open. |
| **Extension** | **Website(logout page ) could not run successfully:**   * The system will display an error message “Reload the page”. * Tells the reason why it cannot run. * User either refreshes the page or tries again. * User tries to load the page after some time. | |
| **Special Requirements** | User friendly interface.  Response time is less than 2 second or less. | |
| **Technology and data variation list** | Use new web development technologies and frameworks for the development of this functionality | |

Table 2.4 view profile fully dress use case

|  |  |  |
| --- | --- | --- |
| **Use case name** | **View profile** | |
| **Scope** | LMS (organization website) | |
| **Level** | User | |
| **Primary Actor** | Employee, Admin , HOD | |
| **Stakeholders and interests** | **User:** He wants to login to his private account.  **Admin:** he is for managing the website. Only he is allowing managing the website. For managing the website he has to login to his account, to know about the events that are related to the particular user. | |
| **Precondition** | The page for logging in will open successfully so that user can fill the email and password. | |
| User must have a valid email, valid password. | |
| User must have click on the button “login”. | |
| **Post condition** | User is login to his account. | |
| User can view his The page for logging in will open successfully so that user can fill the email and password. | |
| User must have a valid email, valid password. | |
| User must have click on the button “login” profile. | |
| **Main success scenario** | **User/Admin** | **System** |
| He request to login. | Login page will open. |
| Enters password and email. Click login. | Validates and open the account. |
| User login successfully | User’s profile is loaded |
| **Extension** | **Website(Login page ) could not run successfully:**   * The system will display an error message “Reload the page”. * Tells the reason why it cannot run. * User either refreshes the page or tries again. * User tries to load the page after some time.   **User could not be login:**   * System displays message “logged in is failed”. * User tries again after identifying the cause. * User tries again after some time.   **Provide details are not valid:**   * System highlights the details that are invalid. * User will correct the invalid details into the valid details.   User again will try to logged in. | |
| **Special Requirements** | User friendly interface.  Response time is less than 2 second or less. | |
| **Technology and data variation list** | Use new web development technologies and frameworks for the development of this functionality | |
| **Frequency of occurrence** | All the time. | |

Table 2.5 apply for leave fully dress use case

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Apply for Leave** | |
| **Level** | User | |
| **Scope** | LMS (organization website) | |
| **Primary Actor** | Employee ,admin | |
| **Stakeholders and interests** | **User**: Employee can easily apply for leave.  **Admin:** He is managing the website. He is allowing managing the website. He should have a unique identification for every user so that they can identify every user and know about the details or every event. | |
| **Precondition** | Employee must be register and login the system to apply for Leave. | |
| **Post condition** | Employee login the system and apply Leave page is open. | |
| Employee enter the details. | |
| **Main success scenario** | **User/Admin** | **System** |
| He requests to load the apply leave activity page of website. | Apply leave page loaded |
| He entered the information. | Check leave activity. |
| Clicks the apply activity | System saves all the records. |
| **Extension** | **Website(Apply Leave page ) could not run successfully:**   * The system will display an error message “Reload the page”. * User either refreshes the page or tries again. * User tries to load the page after some time.   **Apply leave page failed:**   * System displays message “sorry time is finish to fill the activity”. | |
| **Special Requirements** | * User friendly interface. * Response time is less than 4s or less. | |
| **Technology and data variation list** | Use new web development technologies and framework. | |

Table 2.6 Approve or reject fully dress use case

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Approve or Reject** | |
| **Level** | User | |
| **Scope** | LMS (organization website) | |
| **Primary Actor** | HOD ,admin | |
| **Stakeholders and interests** | **HOD**: can approve or Reject the Leave application.  **Admin:** He is managing the website. He is allowing managing the website. He should have a unique identification for every user so that they can identify every user and know about the details or every event. | |
| **Precondition** | User must be register and login the system to process on leave request. | |
| **Post condition** | User login the system and leave request page is open. | |
| User enter the details. | |
| **Main success scenario** | **User/Admin** | **System** |
| He requests to load the process leave page of website. | Leave page loaded |
| He approve or reject the leaves | System saves all the records and send notification. |
| **Extension** | **Website(process Leave page ) could not run successfully:**   * The system will display an error message “Reload the page”. * User either refreshes the page or tries again. * User tries to load the page after some time.   **Process Leave page failed:**   * System displays message “sorry time is finish to fill the activity”. | |
| **Special Requirements** | * User friendly interface. * Response time is less than 4s or less. | |
| **Technology and data variation list** | Use new web development technologies and framework. | |