Software Requirements Specification

Version 1.0

17 Jan 2016

Leave Management System

Submitted in partial fulfillment

Of the requirements of

CS 223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223). The students who submitted this team project were Abhimanyu Singh Gaur, Ajeet Ujjwal, Ashish Sahu, Daval Pargal.

# Table of Contents

[Table of Contents i](#_Toc440664050)

[List of Figures ii](#_Toc440664051)

[1.0. Introduction 1](#_Toc440664052)

[1.1. Purpose 1](#_Toc440664053)

[1.2. Scope of Project 1](#_Toc440664054)

[1.3 Constraints 1](#_Toc440664055)

[1.4 Assumptions and Dependencies 1](#_Toc440664056)

[1.3. Glossary 1](#_Toc440664057)

[1.4. References 1](#_Toc440664058)

[1.5. Overview of Document 1](#_Toc440664059)

[2.0. Overall Description 2](#_Toc440664060)

[2.1 System Environment 2](#_Toc440664061)

[2.2 Functional Requirements Specification 2](#_Toc440664062)

[2.2.1 Use case 1 2](#_Toc440664063)

[Use case: 2](#_Toc440664064)

[2.3 User Characteristics 2](#_Toc440664065)

[2.4 Non-Functional Requirements 2](#_Toc440664066)

[3.0. Requirements Specification 3](#_Toc440664067)

[3.1 Functional Requirements 3](#_Toc440664068)

[3.1.1 << Name of the first feature>> 3](#_Toc440664069)

[3.1.2 << Name of the second feature>> 3](#_Toc440664070)

[3.3 Detailed Non-Functional Requirements 3](#_Toc440664071)

[***3.4*** ***Logical Structure of the Data*** 3](#_Toc440664072)

[4.0 Supporting information 3](#_Toc440664073)

[4.1 Table of contents and index 3](#_Toc440664074)

[4.2 Appendixes 3](#_Toc440664075)

# List of Figures

**No table of figures entries found.**

# 1.0. Introduction

## 1.1. Purpose

The purpose of this document is to give a detailed description of the requirements for the “Leave Management System” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to course instructor for its approval and a reference for developing the first version of the system for the development team.

## 1.2. Scope of Project

The “Leave Management System” is a database based web application, which is primarily designed to help Faculty, Staff and Students of any Academic Institute in the process of applying for leave by automating the process, which would otherwise have to be done manually by them. It also helps the Institute Administration in granting and managing the leaves. If the leave is granted the users will be notified on the system itself as well as via an e-mail.

In the existing system, leaves are maintained using file records. The students and employees need to submit their leaves manually to their respective authorities. This increases the paperwork & maintaining the records becomes tedious. The main objective of the proposed system is to decrease the paperwork and help in easier record maintenance by having a particular centralized Database System, where Leaves are maintained. The proposed system automates the existing system. It decreases the paperwork and enables easier record maintenance. It also reduces chances of Data loss.

## 1.3 Constraints

* The number of users that can be handled is limited by the capacity of database server.
* This system will fail if the server fails, but the data will be safe in the database.

## 1.4 Assumptions and Dependencies

* User should be familiar with using computers.
* User should have internet facility and should know how to use internet.
* Users are expected to have a valid e-mail id.
* Users are expected to have a compatible web browser.
* There is only one admin whose credentials are validated in the code itself.
* There are some LeaveTypes which require attachments while applying for leave.
* User can’t apply for new leave if there is a pending leave.

## 1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Leave Type | Types of leaves provided by an organization like Casual Leave, Medical Leave. |
| User Type | Types of users in an organization like Faculty, Staff and Students in an Academic Institution |
| User | A User is anyone who works in that organization |
| Admin | An Admin is anyone whom the organization has authorized to monitor the system. |
| Requestor | Any user who applies for Leave |

## 1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.5. Overview of Document

The rest of the document is designed in the following way:

# 2.0. Overall Description

## 2.1 System Environment

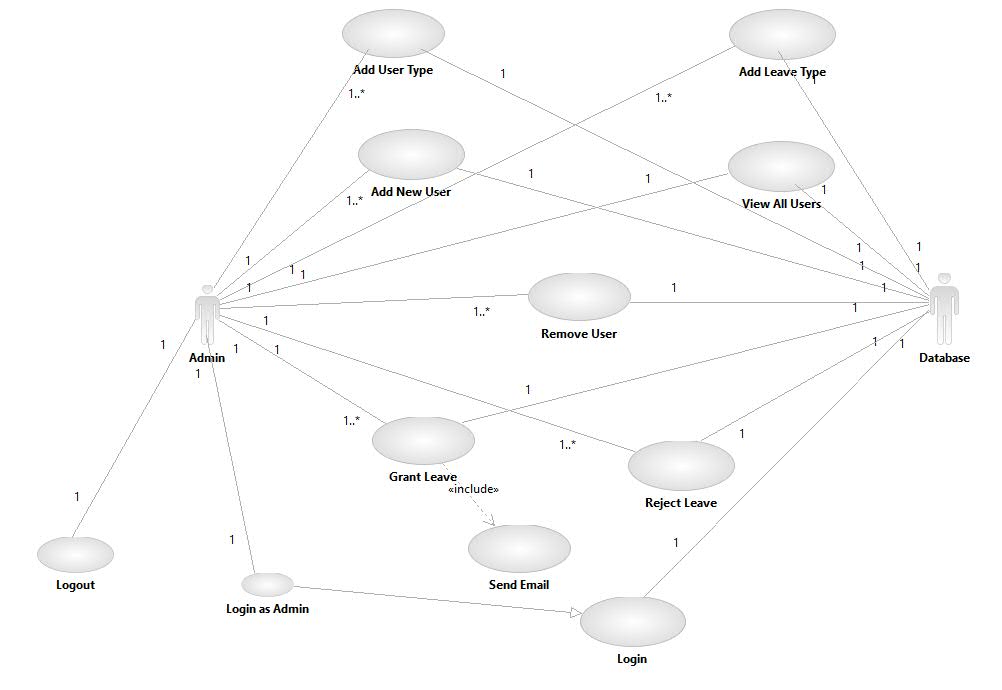
<< Keep blank for the time being >>

## 2.2 Functional Requirements Specification

. << Keep blank for the time being >>

### 2.2.1 Admin Use cases

#### Use case:



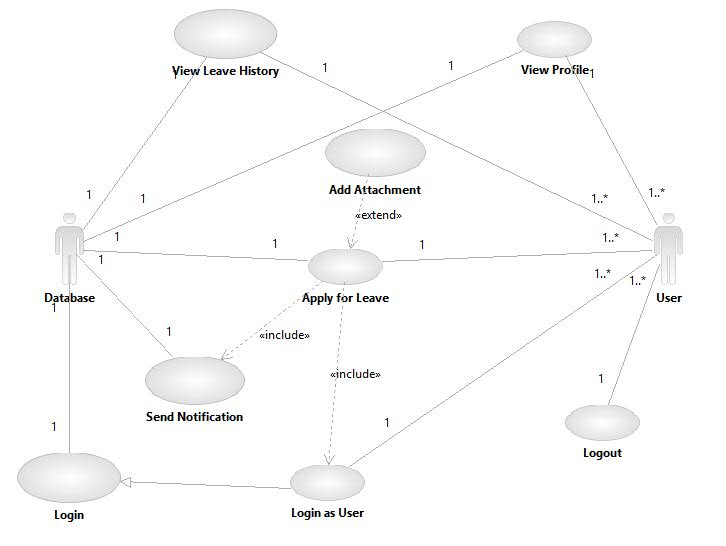
**Diagram:**

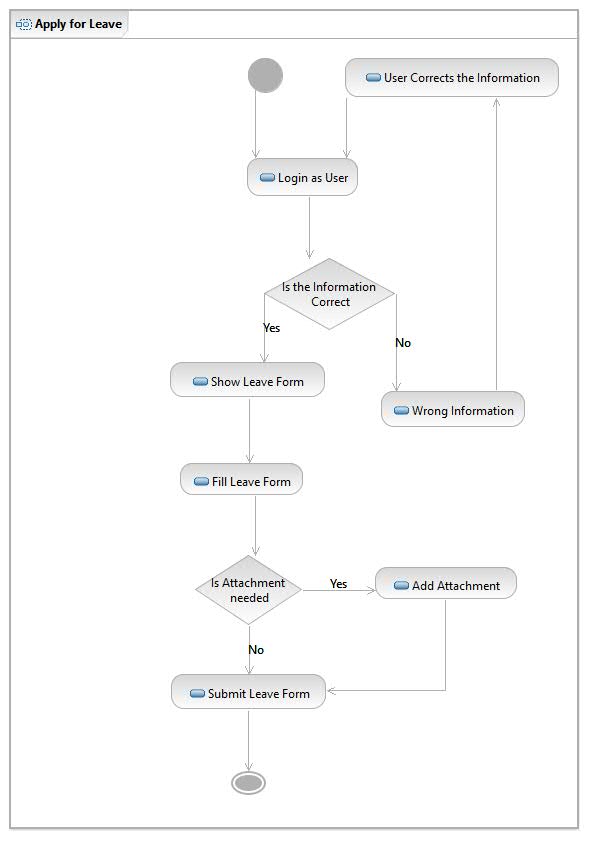
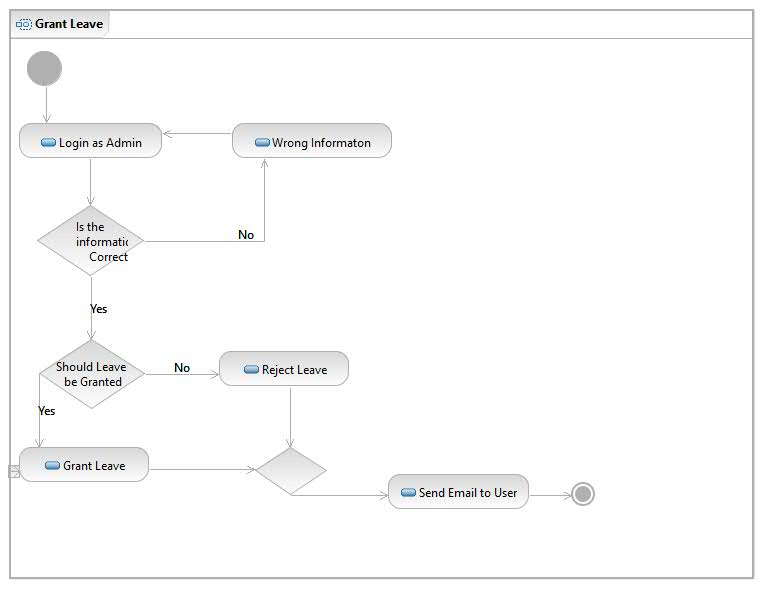
**Brief Description**

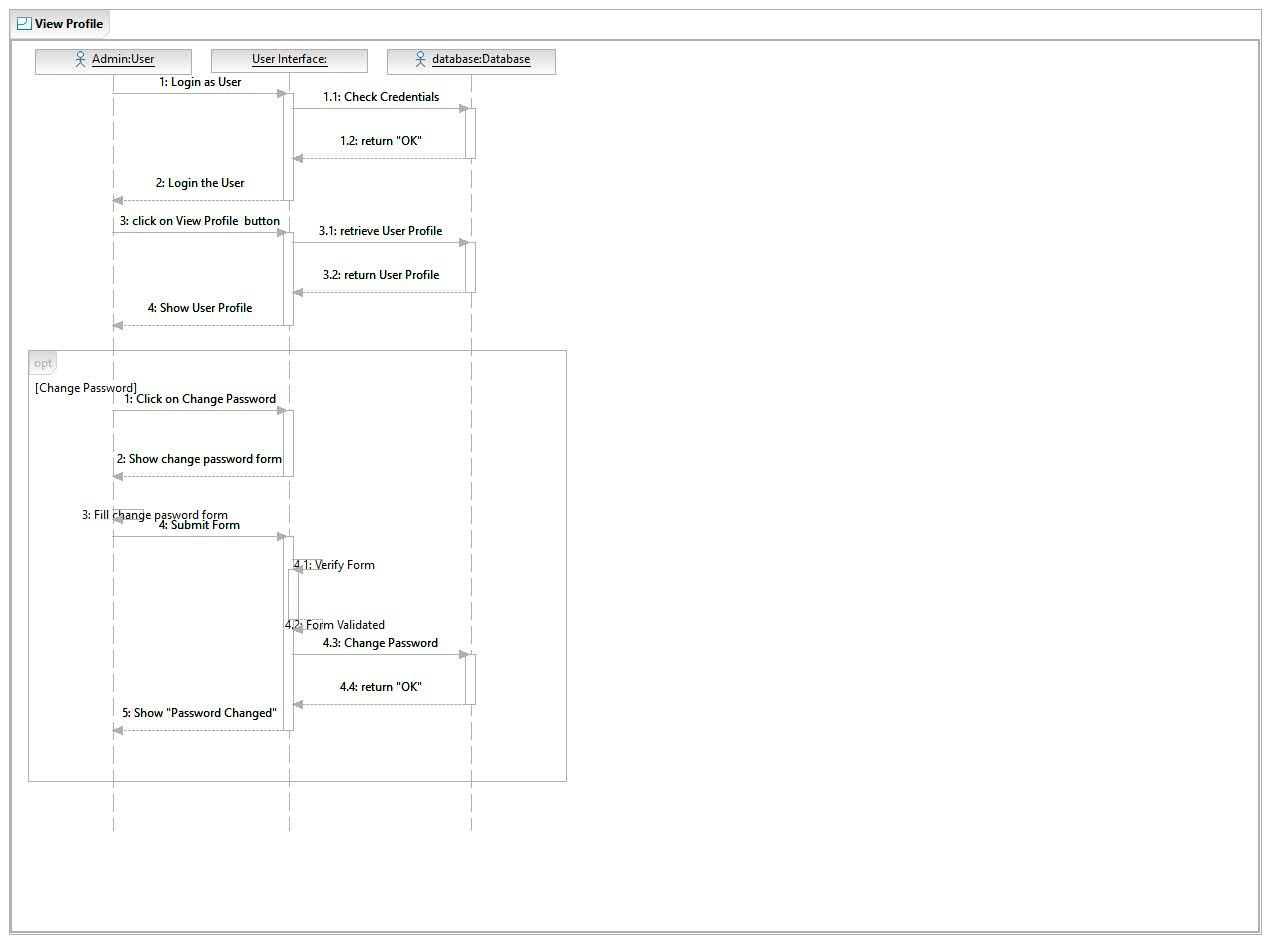
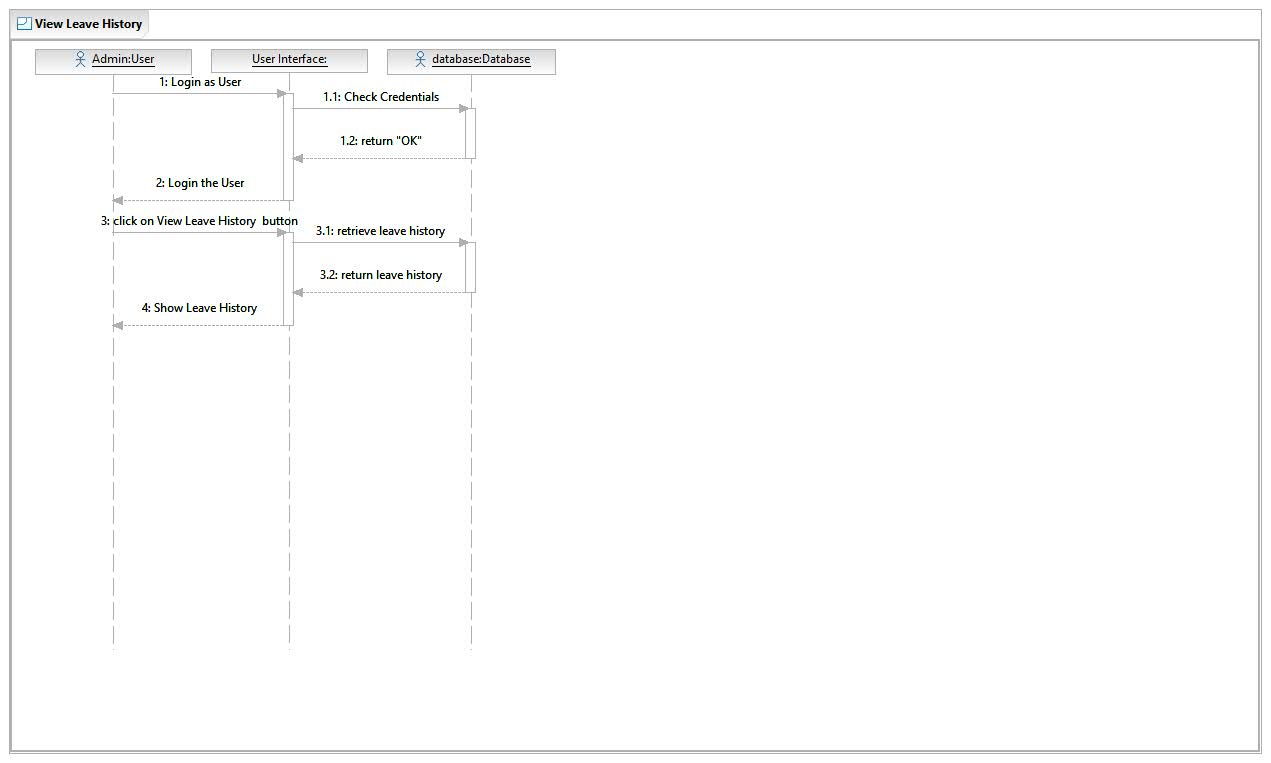
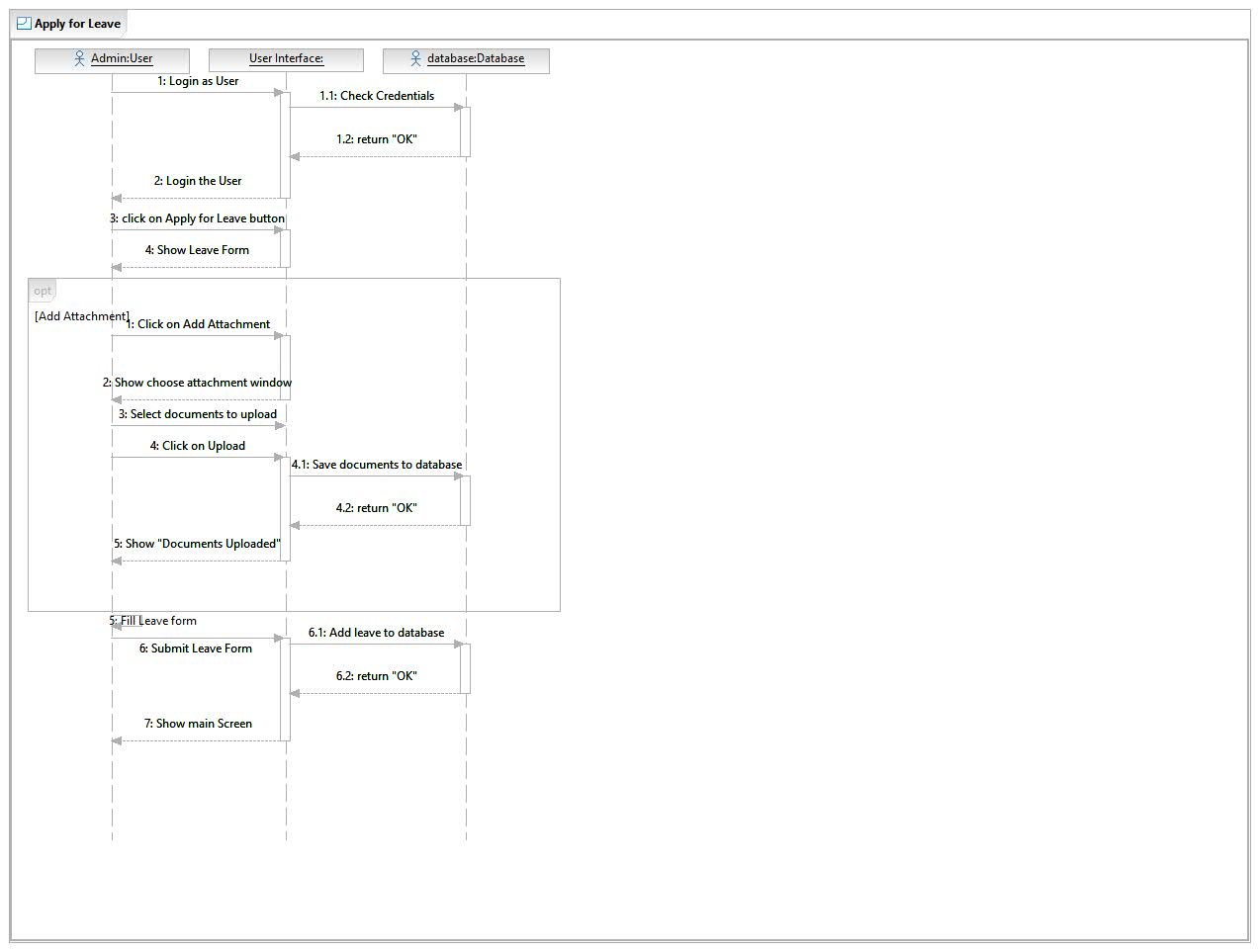
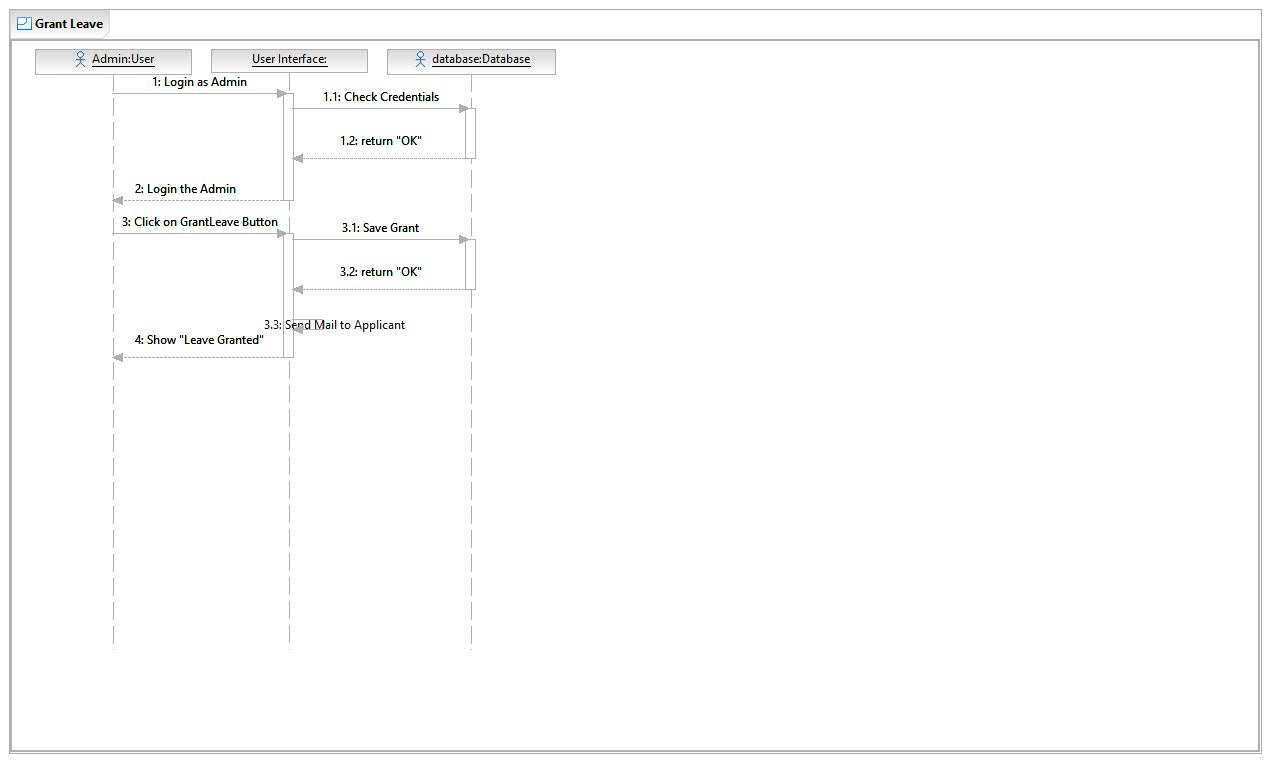
**Initial Step-By-Step Description**

### 2.2.1 User Use cases

#### Use case:



**Diagram**



## User Characteristics

User should be familiar with using computers. User should be internet literate and should know how to use e-mail.

Admin user should be familiar with basic database management.

## 2.4 Non-Functional Requirements

* A Server
* A database application on server
* Internet facility

# 3.0. Requirements Specification

## 3.1 Functional Requirements

### 3.1.1 Add Leave Type

|  |  |
| --- | --- |
| **Use Case Name** | Add Leave Type |
| **Trigger** | Admin clicks on Add Leave Type option |
| **Precondition** | User should be logged in as an admin |
| **Basic Path** | 1. Admin fills the Add Leave Type form 2. System prompts for verification 3. Admin verifies and Leave Type is added |
| **Alternative Paths** | None |
| **Postcondition** | New Leave Type is added to the database |
| **Exception Paths** | The admin may cancel the operation at any time.  In step 3, if Admin does not verify the details then he is prompted to re-enter the details. |
| **Other** | Usually there are 3 types of leaves:   1. Casual Leave (CL) 2. Medical Leave (ML) 3. Vacation Leave (VL) |

### 3.1.2 Remove Leave Type

|  |  |
| --- | --- |
| **Use Case Name** | Remove Leave Type |
| **Trigger** | Admin selects the Leave Type to be deleted |
| **Precondition** | User should be logged in as an admin and View Leave Type page is open |
| **Basic Path** | 1. System shows DELETE as one of the option 2. Admin clicks on DELETE button 3. System prompts for verification 4. Admin verifies and Leave Type is removed |
| **Alternative Paths** | None |
| **Postcondition** | Leave Type is removed from the database |
| **Exception Paths** | The admin may deselect the Leave Type at any time.  In step 2, Admin may select another option.  In step 4, if Admin does not verify then nothing happens. |
| **Other** | None |

### 3.1.3 Modify Leave Type

|  |  |
| --- | --- |
| **Use Case Name** | Modify Leave Type |
| **Trigger** | Admin selects the Leave Type to be modified |
| **Precondition** | User should be logged in as an admin and View Leave Type page is open |
| **Basic Path** | 1. System shows MODIFY as one of the option 2. Admin clicks on MODIFY button and system shows the modification form 3. Admin fills the form 4. System prompts for verification 5. Admin verifies and Leave Type is modified |
| **Alternative Paths** | None |
| **Postcondition** | Leave Type is modified |
| **Exception Paths** | The admin may deselect the Leave Type at any time.  In step 2, Admin may select another option.  In step 5, if Admin does not verify then nothing happens. |
| **Other** | None |

### 3.1.4 View Leave Types

|  |  |
| --- | --- |
| **Use Case Name** | View Leave Types |
| **Trigger** | Admin clicks on View Leave Type |
| **Precondition** | User should be logged in as an admin |
| **Basic Path** | 1. System shows available Leave Types |
| **Alternative Paths** | None |
| **Postcondition** | All Leave Types are shown |
| **Exception Paths** | None |
| **Other** | None |

### 3.1.5 Add User Type

|  |  |
| --- | --- |
| **Use Case Name** | Add User Type |
| **Trigger** | Admin clicks on Add User Type option |
| **Precondition** | User should be logged in as an admin |
| **Basic Path** | 1. Admin fills the Add User Type form 2. System prompts for verification 3. Admin verifies and User Type is added |
| **Alternative Paths** | None |
| **Postcondition** | New User Type is added to the database |
| **Exception Paths** | The admin may cancel the operation at any time.  In step 3, if Admin does not verify the details then he is prompted to re-enter the details. |
| **Other** | Usually there are following types of users in Academic Institutions:   1. Faculty 2. Staff 3. PhD and M.Tech Scholars 4. M.Sc Students 5. B.Tech Students 6. Admin |

### 3.1.6 Remove User Type

|  |  |
| --- | --- |
| **Use Case Name** | Remove User Type |
| **Trigger** | Admin selects the User Type to be deleted |
| **Precondition** | User should be logged in as an admin and View User Types page is open |
| **Basic Path** | 1. System shows DELETE as one of the option 2. Admin clicks on DELETE button 3. System prompts for verification 4. Admin verifies and User Type is removed |
| **Alternative Paths** | None |
| **Postcondition** | User Type is removed from the database |
| **Exception Paths** | The admin may deselect the User Type at any time.  In step 2, Admin may select another option.  In step 4, if Admin does not verify then nothing happens. |
| **Other** | None |

### 3.1.7 Modify User Type

|  |  |
| --- | --- |
| **Use Case Name** | Modify User Type |
| **Trigger** | Admin selects the User Type to be modified |
| **Precondition** | User should be logged in as an admin and View User Type page is open |
| **Basic Path** | 1. System shows MODIFY as one of the option 2. Admin clicks on MODIFY button and system shows the modification form 3. Admin fills the form 4. System prompts for verification 5. Admin verifies and User Type is modified |
| **Alternative Paths** | None |
| **Postcondition** | User Type is modified |
| **Exception Paths** | The admin may deselect the User Type at any time.  In step 2, Admin may select another option.  In step 5, if Admin does not verify then nothing happens. |
| **Other** | None |

### 3.1.8 View User Type

|  |  |
| --- | --- |
| **Use Case Name** | View User Type |
| **Trigger** | Admin clicks on View User Type |
| **Precondition** | User should be logged in as an admin |
| **Basic Path** | 1. System shows available Leave Types |
| **Alternative Paths** | None |
| **Postcondition** | All User Types are shown |
| **Exception Paths** | None |
| **Other** | None |

### 3.1.9 Add New User

|  |  |
| --- | --- |
| **Use Case Name** | Add New User |
| **Trigger** | Admin clicks on Add New User option |
| **Precondition** | User should be logged in as an admin |
| **Basic Path** | 1. Admin fills the Add New User form 2. System prompts for verification 3. Admin verifies and New User is added |
| **Alternative Paths** | None |
| **Postcondition** | New User is added to the database |
| **Exception Paths** | The admin may cancel the operation at any time.  In step 3, if Admin does not verify the details then he is prompted to re-enter the details. |
| **Other** | None |

### 3.1.10 Remove User

|  |  |
| --- | --- |
| **Use Case Name** | Remove User |
| **Trigger** | Admin selects the User to be deleted |
| **Precondition** | User should be logged in as an admin and View User page is open |
| **Basic Path** | 1. System shows DELETE as one of the option 2. Admin clicks on DELETE button 3. System prompts for verification 4. Admin verifies and User is removed |
| **Alternative Paths** | None |
| **Postcondition** | User is removed from the database |
| **Exception Paths** | The admin may deselect the User at any time.  In step 2, Admin may select another option.  In step 4, if Admin does not verify then nothing happens. |
| **Other** | None |

### 3.1.11 Modify User

|  |  |
| --- | --- |
| **Use Case Name** | Modify User |
| **Trigger** | Admin selects the User to be modified |
| **Precondition** | User should be logged in as an admin and View User page is open |
| **Basic Path** | 1. System shows MODIFY as one of the option 2. Admin clicks on MODIFY button and system shows the modification form 3. Admin fills the form 4. System prompts for verification 5. Admin verifies and User is modified |
| **Alternative Paths** | None |
| **Postcondition** | User is modified |
| **Exception Paths** | The admin may deselect the User at any time.  In step 2, Admin may select another option.  In step 5, if Admin does not verify then nothing happens. |
| **Other** | None |

### 3.1.12 View Users

|  |  |
| --- | --- |
| **Use Case Name** | View Users |
| **Trigger** | Admin clicks on View User option |
| **Precondition** | User should be logged in as an admin |
| **Basic Path** | 1. System shows all registered Users |
| **Alternative Paths** | None |
| **Postcondition** | All User are shown |
| **Exception Paths** | None |
| **Other** | None |

### 3.1.13 User Login

|  |  |
| --- | --- |
| **Use Case Name** | User Login |
| **Trigger** | The system is accessed by user |
| **Precondition** | The Login page is open |
| **Basic Path** | 1. User fills the login form 2. System verifies the details 3. User is logged in |
| **Alternative Paths** | None |
| **Postcondition** | User is logged in |
| **Exception Paths** | The user can cancel the login by exiting the system.  In step 2, if credentials are wrong user is asked to re-enter the credentials. |
| **Other** | None |

### 3.1.14 User Logout

|  |  |
| --- | --- |
| **Use Case Name** | User Logout |
| **Trigger** | User clicks on Logout |
| **Precondition** | User is Logged in |
| **Basic Path** | 1. System logs out the user 2. System redirects the user to login page |
| **Alternative Paths** | None |
| **Postcondition** | User is Logged Out |
| **Exception Paths** | None |
| **Other** | None |

### 3.1.15 View Profile

|  |  |
| --- | --- |
| **Use Case Name** | View Profile |
| **Trigger** | User clicks on View Profile option |
| **Precondition** | User is Logged in |
| **Basic Path** | 1. System shows the user profile |
| **Alternative Paths** | None |
| **Postcondition** | User Profile is displayed |
| **Exception Paths** | None |
| **Other** | None |

### 3.1.16 Apply for Leave

|  |  |
| --- | --- |
| **Use Case Name** | Apply for Leave |
| **Trigger** | User clicks on Apply for Leave option |
| **Precondition** | User is Logged in |
| **Basic Path** | 1. System checks available leaves and shows form 2. User fills the leave form 3. System prompts for verification 4. User verifies the details 5. Systems checks the details 6. Leave request is sent to Admin |
| **Alternative Paths** | None |
| **Postcondition** | Leave request is sent to Admin |
| **Exception Paths** | In step 1, if no leaves are available for user then error is shown.  In step 4, if User does not verify the details then he is prompted to re-enter the details.  In step 5, if User may have asked for more number of leaves then he is prompted to re-enter the details. |
| **Other** | None |

### 3.1.17 Cancel Requested Leave

|  |  |
| --- | --- |
| **Use Case Name** | Cancel Requested Leave |
| **Trigger** | User clicks on Cancel Leave button |
| **Precondition** | User is Logged in, View Pending Leaves page is open and Leave Period has not started. |
| **Basic Path** | 1. System prompts for verification if user really wants to cancel 2. User verifies the cancellation 3. System cancels the leave and notifies the admin about cancellation |
| **Alternative Paths** | Admin cancels the requested leave |
| **Postcondition** | Leave is cancelled |
| **Exception Paths** | In step 2, if User does not verify the cancellation then nothing happens. |
| **Other** | None |

### 3.1.18 View Requested Leaves

|  |  |
| --- | --- |
| **Use Case Name** | View Requested Leaves |
| **Trigger** | Admin clicks on View Requested Leaves option |
| **Precondition** | User is logged in as Admin |
| **Basic Path** | 1. System checks and displays if there are any requested leaves |
| **Alternative Paths** | None |
| **Postcondition** | System displays requested leaves |
| **Exception Paths** | In step 1, if there are no requested leaves system displays an error message |
| **Other** | None |

### 3.1.19 Grant Leave

|  |  |
| --- | --- |
| **Use Case Name** | Grant Leave |
| **Trigger** | Admin clicks on Grant Leave button |
| **Precondition** | User should be logged in as Admin, Requested Leaves page is open and there are leaves pending to be granted |
| **Basic Path** | 1. System prompts for verification if Admin really wants to grant leave 2. Admin verifies the Grant 3. System grants the leave and notifies the requestor about it |
| **Alternative Paths** | None |
| **Postcondition** | Leave is Granted to the requestor |
| **Exception Paths** | In step 2, if User does not verify the grant then nothing happens. |
| **Other** | None |

### 3.1.20 Add Attachment

|  |  |
| --- | --- |
| **Use Case Name** | Add Attachment |
| **Trigger** | User clicks on Add Attachment button |
| **Precondition** | User is logged in and he is applying for a leave |
| **Basic Path** | 1. System shows an attachment window 2. User selects the files to be attached 3. System uploads the files |
| **Alternative Paths** | None |
| **Postcondition** | Documents are added as Attachment |
| **Exception Paths** | In step 2, if user does not select any file, then system prompts him to select a file.  In step 2, user may close the attachment window, then nothing happens. |
| **Other** | Following types of attachments are allowed:   1. pdf 2. doc/docx/odt or any other text formats 3. Images |

### 3.1.21 View Attachment

|  |  |
| --- | --- |
| **Use Case Name** | View Attachment |
| **Trigger** | User clicks on attachment |
| **Precondition** | User should be logged in and there should be at least one attachment uploaded |
| **Basic Path** | 1. System opens the attached document in a new window and displays it |
| **Alternative Paths** | None |
| **Postcondition** | Attached document is displayed |
| **Exception Paths** | In step 1, if the attached document can’t be opened in browser, it will be downloaded on user’s computer |
| **Other** | None |

### 3.1.22 Remove Attachment

|  |  |
| --- | --- |
| **Use Case Name** | Remove Attachment |
| **Trigger** | User clicks on Remove Attachment button |
| **Precondition** | User should be logged in and there should be at least one attachment uploaded |
| **Basic Path** | 1. System prompts for verification if user really wants to remove document 2. User verifies the Removal 3. System removes the document. |
| **Alternative Paths** | None |
| **Postcondition** | Attachment is removed |
| **Exception Paths** | In step 2, user may cancel the removal, then nothing happens.  In step 3, server side exception may occur. |
| **Other** | None |

### 3.1.23 View Pending Leaves

|  |  |
| --- | --- |
| **Use Case Name** | View Pending Leaves |
| **Trigger** | User clicks on View Pending Leaves option |
| **Precondition** | User is logged in |
| **Basic Path** | 1. System checks and displays if there are any pending leaves |
| **Alternative Paths** | None |
| **Postcondition** | System displays pending leaves |
| **Exception Paths** | In step 1, if there are no pending leaves system displays an error message |
| **Other** | None |

### 3.1.24 View Remaining Leaves

|  |  |
| --- | --- |
| **Use Case Name** | View Remaining Leaves |
| **Trigger** | User clicks on View Remaining Leaves option |
| **Precondition** | User is logged in |
| **Basic Path** | 1. System checks and displays if there are any remaining leaves |
| **Alternative Paths** | None |
| **Postcondition** | System displays remaining leaves |
| **Exception Paths** | In step 1, if there are no remaining leaves system displays an error message |
| **Other** | None |

### 3.1.25 View Leave History

|  |  |
| --- | --- |
| **Use Case Name** | View Leave History |
| **Trigger** | User clicks on View Leave History option |
| **Precondition** | User is logged in |
| **Basic Path** | 1. System checks and displays if there are any leaves which user has availed |
| **Alternative Paths** | None |
| **Postcondition** | System displays leave history |
| **Exception Paths** | In step 1, if there is no leave history system displays an error message |
| **Other** | None |

### 3.1.26 Send Email

|  |  |
| --- | --- |
| **Use Case Name** | Send Email |
| **Trigger** | Admin clicks on Grant Leave button |
| **Precondition** | User is logged in as Admin and he has granted a leave |
| **Basic Path** | 1. System sends a confirmation email to the requestor of leave |
| **Alternative Paths** | None |
| **Postcondition** | Email is sent to requestor |
| **Exception Paths** | In step 1, a server side error may occur in sending email |
| **Other** | Email is sent to the requestor as a confirmation that his leave request has been accepted. |

### 3.1.27 Send Notification

|  |  |
| --- | --- |
| **Use Case Name** | Send Notification |
| **Trigger** | Admin clicks on Grant/Cancel Leave Request button |
| **Precondition** | User is logged in as Admin and he has Granted/Canceled a leave |
| **Basic Path** | 1. System sends a notification to the related Users |
| **Alternative Paths** | None |
| **Postcondition** | Notification is sent to User |
| **Exception Paths** | None |
| **Other** | A notification signifies that a leave requested by user has been Granted/Canceled. |

### 3.1.28 View Notifications

|  |  |
| --- | --- |
| **Use Case Name** | View Notifications |
| **Trigger** | User clicks on View Notifications option |
| **Precondition** | User is logged in |
| **Basic Path** | 1. System checks and displays if there are any new notifications. |
| **Alternative Paths** | None |
| **Postcondition** | Notifications are displayed |
| **Exception Paths** | In step 1, system displays an error message if there are no new notifications. |
| **Other** | None |

## Detailed Non-Functional Requirements

A fully setup working server will be required to host the application. A database application like MySQL will be required on server to store database. Server should be able to handle all the users. Internet facility is required.

### ***3.4 Logical Structure of the Data***

<< Keep this blank for the time being>>

# 4.0 Supporting information

## 4.1 Table of contents and index

## 4.2 Appendixes