**Leave Management System**

**Software Requirements Specification**



*Submitted By:*

DEEWAN SINGH(B15CS016)

RAJESH MEENA(B15CS030)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **People** |
| 16 Jan 2017 | 1.0 | First draft:SRS | Project Group: WITSF |
| 3 feb 2017 | 2.0 | Second draft : modeling |  |
| 1 March 2017 | 3.0 | Third draft: Coding |  |
| 21 April 2017 | 4.0 | Fourth draft: Testing Correction |  |

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Constraints

1.4 Assumptions and Dependencies

1.5 Definitions, Acronyms and Abbreviations

1.6 References

1.7 Organization of the Document

2. Overall Description

2.1 Product Functions

2.2 User Characteristics

3. Specific Requirements

3.1 Use case description

3.2 Reliability

3.3 Performance Requirements

3.4 Supportability

3.5 Design Constraints

3.6 Online User Documentation and Help System Requirements

3.7 Interfaces

4. Supporting Information

**Software Requirements Specification**

**1.Introduction**

**1.1Purpose**: The purpose of this document is to show the requirements of the leave management software. It illustrates the complete declaration of the software. It also explains about the interaction with the user and the admin. There will be certain constraints in the system which are described in this document. This document describes about the initial idea on which the software will be developed and the requirements of the software.

**1.2Scope**: The Leave Management software aims at helping the user to address issuesrelated to leave management system and services.It is a unique software which helps to organize event without any paperwork and help in easier records and maintenance. With the help of the system users can apply leave online and get approved/rejected. By just few clicks users can check the leave status, leave balances and apply for leave accordingly.

**1.3Constraints:**

* This system might fail when internet connection got failed.
* If the user has less number of leaves available than applied than his/her leave application will get terminated.
* The number of users that will be handle by database can be limited.

**1.**4 **Assumptions and Dependencies:**

* The headings are numbered 1,2,3... and so on and sub-headings are numbered as 1.1,1.2,1.3…
* There is only one admin.
* User should have internet connection.
* User should have valid user-id and password.
* User can’t apply for new leave if there is any pending leave.
* System will work 24\*7 hours.
* System have enough space to hold the database.
* User should be familiar with computer and internet.

**1.5 Definitions, Acronyms and Abbreviations:**

|  |  |
| --- | --- |
| Term | Definition |
| Admin | An Admin is the person who has authority to monitor the system. |
| User | A user is anyone who works in the organization. |
| Leave types | Types of leave in an organization are casual leave, medical leave and others. |
| User type | Types of user in an organization are staff, students and faculty. |
| LMS | Leave management system |
| LB | Leave balance |
| SRS | Software requirement specification |

**1.6References:**

[1]   Pankaj Jalote , An Integrated Approach To Software Engineering.

[2] Internet.

**1.7Organization of the Document:**

In this document section 1 describes the purpose of the document and scope of the software and the assumptions & dependencies.

section 2 includes the product’s functional and non-functional requirement.

Section 3 describes about the specific requirement which includes use case description, reliability, performance requirements, Supportability, Design constraints, Interface etc.

**2 Overall Description:**

**2.1 Product Functions**

*2.1.1Functional Requirements:*

* Apply for leave.
* System will keep student, staff records.
* System provides information about the leave approval/rejection.
* Check for leave availability.

2.1.1.1 Use Case Diagram





* + - 1. Class Diagram



* + - 1. Sequence Diagram







* + - 1. Activity Diagram





* + 1. *Non-Functional Requirements:*
* A server
* A database
* Internet facility
* Security: Access to the system will be protected by a user log in screen that require a valid Userid
* Availability: The system should be available 24\*7 hours.
* Maintainability: The leave Management System is being developed in PHP, HTML which are easy to maintain.
  1. **User Characteristics**

1. **Specific Requirements** 
   1. **Use case description**

3.1.1 User Login

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

3.1.2 User Logout

|  |  |
| --- | --- |
| **Use Case Name** | User Logout |
| **Trigger** | User clicks on Logout |
| **Precondition** | User is Logged in |
| **initial 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** | The user can cancel the logout by exiting the system |
| **Other** | None |

3.1.3 Grant Leave

|  |  |
| --- | --- |
| **Use Case Name** | Grant Leave |
| **Trigger** | Admin clicks on Grant Leave button |
| **Precondition** | User should be logged in as Admin and there are leaves pending to be granted. |
| **initial Path** | 1. Admin clicks on Grant Leave  2. System grants the leave and notifies the requestor about it |
| **Alternative Paths** | None |
| **Postcondition** | Leave is Granted to the requestor |
| **Exception Paths** | if admin does not take action then nothing happens. |
| **Other** | requestor get notified when his/her leave's action is taken |

3.1.4 reject Leave

|  |  |
| --- | --- |
| **Use Case Name** | Reject Leave |
| **Trigger** | Admin clicks on Reject Leave button |
| **Precondition** | User should be logged in as Admin and there are leaves pending to be granted or rejected. |
| **initial Path** | 1. Admin clicks on Reject leave  2. System reject the leave and notifies the requestor about it |
| **Alternative Paths** | None |
| **Postcondition** | Leave is rejected to the requestor |
| **Exception Paths** | If admin does not take action then nothing happens. |
| **Other** | requestor get notified when his/her leave's action is taken |

3.1.5 View all Users

|  |  |
| --- | --- |
| **Use Case Name** | View all Users. |
| **Trigger** | Admin clicks on View all Users option |
| **Precondition** | User should be logged in as an admin |
| **initial Path** | 1. System shows all registered Users |
| **Alternative Paths** | None |
| **Postcondition** | All registerd Users are shown. |
| **Exception Paths** | None |
| **Other** | Admin can check any user's profile |

3.1.6 Apply for Leave

|  |  |
| --- | --- |
| **Use Case Name** | Apply for Leave |
| **Trigger** | User clicks on Apply for Leave option |
| **Precondition** | User is Logged in |
| **initial 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** | 1. if no leaves are available for user then he/she can't apply.  2. if User does not verify the details then he is prompted to re-enter  the details.  3. if User have asked for more number of leaves then he is  prompted to re-enter the details. |
| **Other** | None |

3.1.7 Leave History

|  |  |
| --- | --- |
| **Use Case Name** | View Leave History |
| **Trigger** | User clicks on View Leave History option |
| **Precondition** | User is logged in |
| **initial Path** | System checks and displays if there are any leaves which user has availed till now or this acedmic year. |
| **Alternative Paths** | None |
| **Postcondition** | System displays leave history |
| **Exception Paths** | if there is no leave history system displays that no leave takenin this acedmic year. |
| **Other** | None |

3.1.8 Leave status

|  |  |
| --- | --- |
| **Use Case Name** | Leave Status |
| **Trigger** | User clicks on Leaves status option |
| **Precondition** | User must be logged in |
| **initial Path** | System checks and displays the leave status |
| **Alternative Paths** | None |
| **Postcondition** | System displays the status of the requested leave. |
| **Exception Paths** | if there are no requested leaves system displays some message. |
| **Other** | None |

* 1. **Reliability**

*3.2.1 Maintenance*

*3.2.2 Maximum bug rate*

There will be a maximum of 1 bug/KLOC.

*3.2.3 Security Considerations*

**3.3Performance Requirements**

* + 1. *Response time*
    2. *Capacity*
    3. *Deadline sensitivity*
  1. **Supportability**

*3.4.1 Naming Convention*

All code will be written as specified by the Hungarian Naming Convention.

* + 1. *Coding Standards*
  1. **Design Constraints**
     1. *Software Language*
  2. **Online User Documentation and Help System Requirements**
  3. **Interfaces**
     1. *User Interfaces*

*3.7.2 Hardware Interfaces*

* + 1. *Software Interfaces*
    2. *Communications Interfaces*

1. **Supporting Information**
   1. *Appendix A – Data Flow Diagrams*
   2. *Appendix B – Data Dictionary*