**CS 258 - Software Engineering**
**Spring 2018**

# LEAVE MANAGEMENT SYSTEM

## Software Requirements Specification

(1st version) - 25th Jan 2018

| | |
|---|---|
| Rotte Priyanka Ajay | - 160001051 |
| Chakunta Vijay Risheel | - 160001017 |
| Rishabh Kumar Verma | - 160001048 |
| Rishabh Verma | - 160001049 |
| Sunand Agarwal | - 160001057 |
| Neerudi Sai Vikas | - 160001041 |

## Revision History

| Date | Description | Author | Comments |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

| Signature | Printed Name | Title | Date |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The aim of this document is to specify the complete description of the project based on online leave application management at Indian Institute of Technology Indore. This document describes the functionalities, external interfaces, attributes and the design constraints of the system which will be developed as part of this project which is going to help in procedure of leave application as well as provide features can be used for filtering out details based on their submitted information. It is intended to be used by members of the project team that will implement and verify the correct functioning of the system. The client can also use this document to verify the fulfilment of his/her/their requirement.

## 1.2 Scope

1) The IITI Leave Processing System is an online system for leave management at IIT Indore.
2) The system will make it convenient for staff and faculty to apply for leave and get necessary recommendation and approval.
3) It's a one stop solution for addressing issues from multidisciplinary angles related to leave management and services.
4) The existing system uses paper and makes leave application processing a long and slow process, and sometimes due to human error, it can be a huge mess. The proposed system makes leave processing a lot faster as most work is automated.
5) All faculty and staff members, in the campus will have their accounts on the website as well as there will be a provision of super admin, which mainly can control almost all the functionalities. So, a person who wishes to take a leave can apply for it by filling in and submitting the particular leave form, which is then forwarded to the necessary authorities. This eliminates the extra time required for logistics and makes the process paperless.
6) The staff and faculty members can also check the number and type of leaves they have, and that'll be updated automatically that now how many leaves are remaining in their kitty.

## 1.3 Definitions, Acronyms, and Abbreviations

1) **MySQL**:
   It's an open source database management system. SQL stands for Structured Query Language.

2) **PHP**:

Hypertext Preprocessor (earlier called, **Personal Home Page**). PHP is an HTML- embedded. It's a widely used open source general purpose scripting languages that is especially suitable for web development and can be embedded into HTML.

3) **HTML**:

Hyper Text Markup Language is a markup language for designing web pages.

4) **GUI**:

Graphical User Interface, the front end which interacts with the user of the application.

5) **Laravel**:

It is a free, open source PHP framework and intended for the development of web applications following the model–view–controller (MVC) architectural pattern.

## 1.4 Overview

The rest of the SRS document is divided into four main parts:

1) **General Requirements**

This section describes the general requirements for efficient and convenient usage of application. It is an attempt to make understanding the requirements easier.

**2) Specific Requirements**

This section guides through the requirements specific to the design and implementation of this application. A brief overview of the hardware, software and interface requirements is provided in this section.

**3) Functional Requirements**

This includes functional requirements which are to be fulfilled by the application being developed. Basic features of the application are described here.

**4) Non-Functional Requirements**

Non-functional requirements are the performance characteristics of the system. These include requirements like speed, security, recoverability, durability etc.

# 2. General Description

## 2.1 Product Perspective

The application is web based and uses PHP and MySQL server to make access and operate on database. The client interacts with GUI which sends requests to the web server. PHP interpreter interprets the PHP code, queries the database and returns the output. After this, the result is used to perform some other functions.

## 2.2 Product Functions

The product provides an online platform for the user to apply for any type of leave. All users will create an online account and need to log into it to access the features offered in the system. After logging in, the user will be able to view his/her leave history and leave balance. To apply for any type of leave, the users will need to fill an online form asking for all the required details. After this, the application will be forwarded to a Recommending Authority via mail. The recommending authority can either recommend and hence forward the application to the Approving Authority or can suggest changes in the application and send it back to the applicant. The approving authority can either approve the leave or can deny the leave. The applicant will be notified accordingly. On approval, the leave balance is also updated. Now, on the day of the completion of the leave, a notification would prompt the applicant to prepare a joining report and send it to the Recommending authority, who on approval, would send it to the Approving Authority to be approved so that the applicant can join the institute. The objective is to make this process as fast and convenient as possible.

## 2.3 User Characteristics

Users can be broadly classified into four types. Each type having certain authorizations and access to different levels of the database system

1) General Users: This category consists of all users who can apply for leave, for example, the DR Admin, Faculty, the Non - Teaching staff, etc. This is a super category and consists of all the users of the system.
2) Recommending Authority: This category consists of the users who have the authority to recommend leave application of other users.
3) Approving Authority: All users who have the authority to approve leave applications of other users fall in this category. They will receive leave applications for approval and can either accept or reject them.

4) Administrator Section-In-Charge: The administrator will manage the database. He will make necessary changes to keep the database consistent. He will have complete access to the database. He can also filter the database conveniently so as to get specific results and retrieve a particular set of data from the database based on his queries.

## 2.4 General Constraints

The general constraints on this application software include:

1) Fast Response Time:

In near future hundreds of users may be accessing the portal at once and hence the response time of the system must be less in order to avoid any sort of crashes and faults.

2) User Friendly Interface:

The GUI must be user friendly for the convenience of users.

3) Security:

Since an online portal is being designed, security is major constraint for the design. System must be secure as all the data is being kept on the server side and any breach in security can cost loss and misuse of data.

## 2.5 Assumptions and Dependencies

1) Server-Side Dependencies:

- Apache tomcat
- PHP Interpreter
- MySQL Server
- Laravel

2) Client-Side Dependencies: Browser with HTML5 support (higher than Internet Explorer 7 or equivalent)

3) It is assumed that the operating system and other underlying pieces of software on the user side are free from any error which may affect the functioning of this system.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

User Interface will consist of a log in screen. As soon as he logs in, he will be presented with the following options:

- Apply for a leave:

Upon clicking this button, user will be presented with several options representing different kinds of leave, upon selecting the required kind of leave the user will be directed to a form to be filled out regarding the details of leave. As soon as the applicant applies, mails will be sent to the *recommending authority*, i.e., HOD of the department and *approving authority.*

- Leave Balance:

This option will display the number of leaves left over in the bank of the applicant corresponding to the type of leave chosen.

- Leave History:

This option will display the number of leaves taken and dates of the same, corresponding to the leave.

- Account Settings:

As the name suggest, user will be able to edit the details from the above option including a password change.

- Calendar:

The calendar will be visible to all the users, which shows the holidays along with their dates. It may be a grid of a list depicting the holidays round the year.

All the above options are available to all the employees. Below are some options *exclusive* to only some users.

◆ Recommend leave:

The recommending authority, in this case the HOD of the concerned department will be able to recommend the application of leave. Once it is recommended, a mail will be sent to the *approving authority.* The recommending authority may also reject or revert the application back to the applicant with proper changes.

- ◆ Approve leaves:

This option is exclusive only to the *approving authority*, in this case the DR Administration. The person will be able to view the details of the applicant along with the kind of leave, number of days on leave, reason and other details. He has the power to approve or reject the application. After he does his part, a mail will be sent to *recommending authority* and the applicant about the approval of the leave. If the application gets rejected, he can revert it back with proper reason.

- ◆ Approval History:

Upon choosing this option, the user will be asked to select the kind of leave. And after choosing that, a record of number of leaves granted and people whom the leave is granted to along with the date will be presented. This is only for *approving authority.*

- ◆ Recommendation History:

There is an analogous recommending history, which is exclusive to recommending authority, which gives the details of leaves recommended. The interface will the same as that of *approving authority.*

- ◆ Search User:

Finally, this option provides the *administrator* and *approving* authority with six options of search filters of a group of employees. They are salary, name, date of joining, designation(post) and caste. The results of the search will be displayed in the page itself.

There will be only one Administrator who will be given access to the database. He can do the following tasks.

- View Application Status:

The administrator will be able to view the status and details of all the pending applications. By clicking on this button. Appropriate filters will be provided to the administrator to perform the task.

- Modify Leaves:

This enables the admin to modify the leave categories including addition, deletion and altering number of days. These changes will be reflected in other places as well, like in the page where user applies for a leave and in his leave bank.

- Modify Holidays:

The admin may also modify institute holidays including addition, subtraction of holidays.

- Manage Users:

Upon clicking this button, administrator will be able to add new user (admin, teachers, non-teaching etc) and provide them with an initial password. They may also modify the details of existing users and also delete them. There is also option of viewing the details of the employee by entering his/her ID.

- Assign Users:

One of the function of *administrator* is assigning users with special privileges of *recommending* or *approving authority.* The changes made by the admin will be reflected in the database.

### 3.2 Functional Requirements

All users will be provided one of the below mentioned type of accounts depending on their requirements and roles.

1. GENERAL APPLICANTS

This class of users will be provided functionalities like the following:

- Viewing leave balance (checking number and type of leaves remaining in the

various leave categories).

- Applying for one of the various types of leaves if there is sufficient leave balance,

after which it is sent to a recommending authority.

- Facility of encashment of the earned leaves wherever applicable.
- Managing account details like personal information and passwords.
- Accounts will automatically be credited with the necessary number of leave days

at appropriate times (1st January and 1st July).

- Notifications will be sent to the applicant’s registered email-id whenever

necessary (leave approved, leave balance exhausted etc.).

- Also, notification will be sent to the applicant’s registered email-id one day before the requested leave period is over along with a joining report to be submitted.

2. RECOMMENDING AUTHORITY

These types of users will be provided all the functionalities provided to the general applicants along with some special privileges specific to these types of users. These users will be able to check leave balance and apply for leave like general users but
will have an additional option to view the applications which have been referred to them
for recommendation. They shall be able to either forward valid applications to the approving authorities with their recommendations and comments, or revert them to the user with suggested changes
in an application. They can also see the history of applications they have previously
validated.

3. APPROVING AUTHORITY

Users of this type will have the functionalities provided to the general users with additional features relevant to them.

- They shall be able to view all the applications that they have received, for their necessary approval.
- They can approve any pending applications or reject them with proper reason/comment.
- They can also see the history of applications they have approved/rejected.
- Notifications will be sent to their registered email-ids whenever a leave application is validated by the recommending authority.
- They are also notified when the applicant submits the joining report after availing leave.
- They shall be able to Search user by: User type, Salary, Name, Date of joining, Post and Caste.

4. ADMINISTRATOR

- They shall be provided the exclusive access to the database along with the functionalities

provided to the general users.

- They shall be able to view the status of all the pending applications, and also the history

of all the leave applications which been rejected or approved previously.

- They shall be able to view the account information of all the users of the

system (leave balance, personal information etc.).

- They shall be able to manage, add or delete leave categories.
- They shall also be able to manage holidays of the year.
- They shall also be able to give authority to users for recommending or approving leaves.
- The administrator will also be able to manage user system as follows:
    - Create user (Admin, Faculty, Non-teaching, contractual)
    - Modify user
    - Delete user
    - Search user by: User type, Salary, Name, Date of joining, Post and Caste.

### 3.3 Non-Functional Requirements

**Performance**

These are the requirements that are non-functional in nature. Specifically, these
are the constraints the system must work within. The application must be compatible
with almost all type of standard web browsers.

**Usability**

The Screens should be designed for ease of use by non-technical users who do not have any computer knowledge. The GUI design shall be intuitive and task-based without any superfluous design.
Familiarity – The system's interfaces and navigations should be based on other systems that the users are familiar with.

**Standards Compliance**

There shall be consistency in variable names within the system.
The graphical user interface shall have a consistent look and feel.
Performance Requirements

1. The load time for user interface screens should not take a long time.
2. The system shall consume very little primary memory.
3. The server should be capable of handling a large number of requests Simultaneous requests (around 100-150).
4. Thus the internet bandwidth should be as high as possible, minimum of 100 MBPS is required

5.It is recommended that the server hardware should have at least 2 GHz processing speed and

4 GB RAM.
6. The management application should be able to process queries within a very few seconds.

**Reliability**

1. The system should be able to manage to leaves even after some years without errors.
2. The failure rate of the system should be very nominal.
3. The system shall be tested on end cases to offer the user quality and reliable package.

**Availability**

1. As the whole institutes leave processing will be dependent on the system thus, it should not be down for more than few hours.
2. Uptime for the system should be 24X7.
3. Anyone shall be able to connect to it from anywhere.
4. In case of application or a hardware failure, the system should re-initiate immediately.
5. In case of a possible hardware failure or corruption of database, the system administrator should immediately restore the backup.

**Security**

1. Unauthorized creation/modification/publishing of data - through the username and password authentication as defined for relevant user groups.
2. Unauthorized viewing of data - through the username and password authentication as defined for relevant user groups.
3. The software should adhere to security guidelines, standards and policies prescribed by NIC’s Security Division and should be audited & certified for compliance with these standards by Security Division before it is hosted in Production Environment.
4. Privacy of every user should be maintained.
5. The operating system begins used on the server should be the latest version so that it must be free bugs.
6. The leave management should be free from common vulnerabilities such as SQL Injection, Cross-Site-Scripting, Broken Authentication and Session

Management, Insecure Direct Object References, Cross Site Request Forgery.

Etc.

7. The administrator shall also be carrying out a timely inspection of a database to ensure safety.

**Language Support**

1. The application will store data using Unicode representation.
2. Only languages where Unicode font file and keyboard manager/driver are available will be supported.

**Maintainability**

1. Software Code must be modular and well documented.
2. All the artifacts related to the software such as code, SRS, User Manual etc. should be well documented and self-explanatory for any programmer to understand. Detailed documentation shall be available at each stage for easy comprehensions of the application system.
3. All documents shall be prepared as per the defined documentation standards.
4. The database should be accessible to the administrators so as to carry out maintenance. The database should also be periodically backed up to prevent information loss due to system crashes.

**Portability**

1. The platforms on which the system runs should be generic enough to allow substantial amount of portability.

## 3.4 Logical Database Requirements

1. The names, contacts, addresses should be in a valid standard format to maximize correctness of information.
2. Redundancy should be minimum in the database.
3. At least 10 GB storage is recommended for database storage requirements.