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. 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. 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. 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.

1.3 Definitions, Acronyms, and Abbreviations

1) MySQL:
   It's an open source database management system. SQL stands for Structured Query Language.
2) PHP:
   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:
   HyperText Markup Language is a markup languages for designing web pages.
4) GUI:
   Graphical User Interface, the front end which interacts with the user of the application.

1.4 References

1) IEEE Software Engineering Standards Committee, "IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

1.5 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 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 output is used to take further steps.

2.2 Product Functions

The product provides an online platform for the user to apply for any type of leave. The objective is to make this process as fast and convenient as possible.

All users will create an online account and need to log into it to access the variety of 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* (chosen by the applicant). The recommending authority can either recommend and hence forward the application to the *Approving Authority* or can suggest chances in the application and send it back to the applicant. The approving authority can either approve the leave or can deny the leave.

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. 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**
   The administrator will manage the database. He will make necessary changes to keep the database consistent. He will have complete access to the database.

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
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 the user proceeds he will be presented with four options:
1) Apply for Leave:

The user on selecting this option will be redirected to *Leave Application* page where he/she will be able to fill the form, provide details for the processing of the leave application and choose from the various types of leaves. After that the applicant will be able to choose a recommending authority and submit the application.

2) Leave Balance:
This option will display the balance of various leaves the user has in account.

3) Leave History:
Upon selecting this option the user will be redirected to *Leave History* page where he/she can view his/her past leave applications.

4) Account Settings:
User can view and modify details associated with his/her account.

Apart from these options, the other categories of users i.e. the Recommending Authority and Approving Authority will have more options.

5) Applications awaiting recommendations (Exclusive to *Recommending Authority* only):
On selecting this, the user will be presented a list of various leave applications awaiting their recommendation. Selecting any of these applications will show them the application and options to either recommend and forward to *approving authority* or suggest edits and return to the applicant.

6) Applications awaiting approval (Exclusive to *Approving Authority* only):
Selecting this option will provide a list of Leave applications forwarded by *recommending authority* and on selecting any of them will present them with the submitted application form and option to either to reject it or to accept the application.

7) Leave Applications Approval History (Exclusive to *Approving Authority* only):
This will display all the applications previously approved by the user.

8) Leave Applications Validation History(Exclusive to *Recommending Authority* only)
This will display all the applications they have previously validated.

3.1.4 Communications Interfaces

The application will have a network server that is web based and created using PHP. Server is used to retrieve information from the database and use it as programmed based on instructions from the website. Database will be managed using MySQL server.

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 leave if there is sufficient leave balance, after which it is sent to a recommending authority (optional).
   - 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.).

2. **RECOMMENDING AUTHORITY**
   These type of users will be provided all the functionalities provided to the general applicants along with some special privileges specific to these type 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. In case the recommending authority is unable to validate an application he/she can forward it to other proper recommending authority to do the same. 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.

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.
   The administrator will also be able to view the account information of all the users of the system (leave balance, personal information etc.).

3.4 Non-Functional Requirements

3.4.1 Performance

Since the system is web based, the server/servers should be capable of handling large number of simultaneous requests(say 100).Thus the internet bandwidth should be as high as possible (200mbps recommended)to handle the large number of requests. It is recommended that the server hardware have at least 2GHz processing speed and 4GB RAM.

3.4.2 Reliability

The system should be able to manage leave systems ever after some years without errors even though new people may be added etc.

3.4.3 Availability

The system should have a downtime not greater than a few hours during which the user may not be able to apply for urgent leave.

3.4.4 Security

The operating system being used on the server should be of the latest version so that it is free of common vulnerabilities. The application should also have security features to protect it from

basic vulnerabilities. The administrator shall also be carry out timely inspection of database to ensure safety.

3.4.5 Maintainability

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.

3.4.6 Portability

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

3.6 Logical Database Requirements

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