**EXPERIMENT-3**

**AIM-Development of SRS document Design Document for the selected project**

**Introduction**

The main objective of this document is to illustrate the requirements of the project Library Management system. The document gives the detailed description of the both functional and non-functional requirements proposed by the client.The purpose of this project is to provide a friendly environment to maintain the details of books and library members.The main purpose of this project is to maintain easy circulation system using computers and to provide different reports. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

**Scope of Development Project**

Library Management System is basically updating the manual library system into an internet-based application so that the users can know the details of their accounts, availability of books and maximum limit for borrowing.

The project is specifically designed for the use of librarians and library users. The product will work as a complete user interface for library management process and library usage from ordinary users. Library Management System can be used by any existing or new library to manage its books and book borrowing, insertion and monitoring. It is especially useful for any educational institute where modifications in the content can be done easily according to requirements. The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules.

The language used for developing the project is PHP & HTML,CS,JAVASCRIPT as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

**Libraries And Kit Used**

* Bootstrap.
* Jquery.
* Materialize UI
* DiapoJs.
* Datepicker JS.

**Scope of Development Project**

Library Management System is basically updating the manual library system into an internet-based application so that the users can know the details of their accounts, availability of books and maximum limit for borrowing.

The project is specifically designed for the use of librarians and library users. The product will work as a complete user interface for library management process and library usage from ordinary users. Library Management System can be used by any existing or new library to manage its books and book borrowing, insertion and monitoring. It is especially useful for any educational institute where modifications in the content can be done easily according to requirements. The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules.

The language used for developing the project is PHP & HTML,CS,JAVASCRIPT as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (freely available), and development process.

**Libraries And Kit Used**

* Bootstrap.
* Jquery.
* Materialize UI
* DiapoJs.
* Datepicker JS.

**Definitions, Acronyms and Abbreviations**

SQL-> Structured query Language

ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment

SRS-> Software Requirement Specification

ISBN -> International Standard Book Number

IEEE ->Institute of Electrical and Electronics Engineers

**Software Configuration:-**

This software package is developed using php&html as front end. MYSQL Server as the back end to store the database.

Operating System: Any

Language: PHP

Database: MYSQL SERVER (back end)

**Hardware Configuration:-**

Processor: Pentium(R) Dual-core CPU

Hard Disk: 40GB

RAM: 256 MB or more

**Data Requirement**

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like create an account, selecting books and putting into account. Now the output will be visible when the user requests the server to get details of their account in the form of time, date and which books are currently in the account.

**External Interface Requirement**

**GUI**

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the book.

* It allows user to view quick reports like Book Issued/Returned in between particular time.
* It provides stock verification and search facility based on different criteria.
* The user interface must be customizable by the administrator
* All the modules provided with the software must fit into this graphical user
* The design should be simple and all the different interfaces should follow a standard template
* The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

Login Interface:-

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

Search:-

The member or librarian can enter the type of book he is looking for and the title he is interested in,then he can search for the required book by entering the book name.

Categories View:-

Categories view shows the categories of books available and provides ability to the librarian to add/edit or delete category from the list.

Librarian’s Control Panel:-

This control panel will allow librarian to add/remove users; add, edit, or remove a resource. And manage lending options.

**System Features**

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* User authentication and validation of members using their unique member ID
* Proper monitoring by the administrator which includes updating account status, showing a popup if the member attempts to issue number of books that exceed the limit provided by the library policy, assigning fine to members who skip the date of return
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member accounts

**5. Other Non-functional Requirements**

**5.1 Performance Requirement**

The proposed system that we are going to develop will be used as the Chief performance system within the different campuses of the university which interacts with the university staff and students. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

* The performance of the system should be fast and accurate
* Library Management System shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid username/password
* The system should be able to handle large amount of data. Thus it should accommodate high number of books and users without any fault

**Safety Requirement**

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

**Security Requirement**

* System will use secured database
* Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
* System will have different types of users and every user has access constraints
* Proper user authentication should be provided
* No one should be able to hack users’ password
* There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

**Requirement attributes**

* There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
* The project should be open source
* The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
* The user be able to easily download and install the system

**User Requirement**

The users of the system are members and Librarian of the university who act as administrator to maintain the system. The members are assumed to have basic knowledge of the computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of:-

* Backup and Recovery
* Forgot Password
* Data migration i.e. whenever user registers for the first time then the data is stored in the server
* Data replication i.e. if the data is lost in one branch, it is still stored with the server
* Auto Recovery i.e. frequently auto saving the information
* Maintaining files i.e. File Organization
* The server must be maintained regularly and it has to be updated from time to time

**Other Requirements**

**Data and Category Requirement**

There are different categories of users namely teaching staff, Librarian, Admin, students etc. Depending upon the category of user the access rights are decided.It means idifferent categories of books available. According to the categories of books their relevant data should be displayed. The categories and the data related to each category should be coded in the