Team Details:

Imadh Ajaz Banday - 1BM20CS059
Jayam Mouneash - 1BM20CS063
Kotturu Amarnath - 1BM20CS074
Mukesh Kumar N V - 1BM20CS088
Class - 6B
Batch - B1

Date: 19-04-2023

Library Management System

Problem Statement:

Developing a user-friendly Hotel Management System for All Stakeholders

The current library management system lacks user-friendliness, making it difficult for stakeholders to interact with and utilize it to its full potential. This results in reduced efficiency and productivity in managing library resources, causing frustration and inconvenience for both library staff and patrons. Therefore, there is a need to develop a user-friendly library management system that benefits all stakeholders, including library staff, patrons, and administrators. The system should be easy to use, intuitive, and provide seamless access to library resources while minimizing the workload of staff. The system should also offer a range of features that meet the needs of different stakeholders, such as a user-friendly interface for patrons, advanced search functionality, and efficient reporting for administrators. By addressing these issues, the new library management system will improve the overall user experience, increase productivity, and ensure that library resources are utilized to their fullest potential.

Software Requirement Specification(SRS)

1 Introduction

1.1 Purpose of this document

The purpose of this document is to provide a comprehensive Software Requirements Specification (SRS) for the development of a user-friendly library management system that benefits all stakeholders, including library staff, patrons, and administrators.

1.2 Scope of this document

This document outlines the functional and non-functional requirements, constraints, and specifications of the library management system.

1.3 Overview

The library management system is designed to improve the overall user experience, increase productivity, and ensure that library resources are utilized to their fullest potential. It will include features such as an easy-to-use interface for patrons, advanced search functionality, and efficient reporting for administrators.

2 General Description

The library management system will be a web-based application that provides a centralized platform for managing library resources. It will be designed to be user-friendly, intuitive, and customizable to meet the needs of different stakeholders.

3 Functional Requirements

- The system should provide a secure login mechanism for patrons and library staff, with role-based access control.
- The system should allow library staff to manage library resources such as books, journals, and other materials. It should support the addition, deletion, and modification of resources, as well as the ability to track their availability and status.
- The system should support borrowing and returning of resources by different borrowers, with automated notifications and reminders for overdue items.

- The system should allow users to search for resources using multiple criteria, including title, author, subject, and keywords.
- The system should provide administrators with the ability to generate reports and analytics on library resource usage, circulation, and borrowing status.

4 Interface Requirements

4.1 User Interface

The user interface should be intuitive, user-friendly, and responsive, with support for different devices and screen sizes.

4.2 Integration Interface

The system should be designed to integrate with other library systems and technologies, such as RFID, barcode scanners, and library catalogs.

5 Performance Requirements

- The system should be designed to handle a high volume of concurrent users and transactions, with fast response times and minimal downtime.
- The system should be designed to scale horizontally and vertically to handle future growth in users and data volume, with the ability to add additional resources as needed.
- The system should be highly available, with a minimum uptime of 99.9%, and minimal downtime for maintenance or upgrades.
- The system should be designed with robust data security measures to protect against unauthorized access, with encryption of all sensitive data in transit and at rest.
- The system should have a comprehensive backup and recovery plan in place to ensure that data can be recovered in the event of a disaster or system failure. Regular backups should be performed and tested to ensure they are functioning correctly.
- The system should have performance monitoring and reporting capabilities, with the ability to track system performance metrics such as response time, server load, and user activity. This data should be available in real-time for system administrators to monitor and make adjustments as needed to maintain optimal system performance.

6 Design Constraints

- The system should be designed to handle a high volume of concurrent users and transactions, with fast response times and minimal downtime.

- The system should be designed to comply with relevant data protection and privacy regulations
- The system should be designed to be interoperable with other library systems and technologies, such as library catalogs, RFID or barcode scanners, and other library software. This includes ensuring that the system uses standard data exchange protocols and APIs to allow for easy integration with other systems.
- The system should be designed to be accessible to all users with access to the Internet.
- The system should be designed using a modern technology stack that is well-supported and has a strong developer community. This includes using scalable and reliable cloud-based infrastructure, and using open-source libraries and frameworks whenever possible to reduce vendor lock-in and promote interoperability.
- The system should be designed to be intuitive, easy-to-use, and customizable, with support for multiple languages and localization. This includes conducting user research and usability testing to ensure that the system meets the needs of different stakeholders, and incorporating user feedback into the design process.

7 Non-Functional Attributes

- **Reliability:** The system should be reliable, with a low rate of system failures, errors, and data loss. This includes implementing robust error handling and recovery mechanisms, and conducting regular stress testing to identify and address potential failure points.
- **Maintainability:** The system should be designed to be maintainable, with clear documentation, modular architecture, and well-structured code. This includes following industry-standard coding practices and using automated testing and continuous integration to ensure code quality and reduce regression errors.
- **Scalability:** The system should be designed to be scalable, with the ability to accommodate future growth in users, data volume, and system complexity. This includes using distributed computing architectures, load balancing, and horizontal scaling techniques to ensure that the system can handle increased workload without sacrificing performance.
- **Flexibility and Adaptability:** The system should be designed to be flexible and adaptable, with the ability to accommodate changing user needs and requirements.

This includes using agile development methodologies, incorporating user feedback into the design process, and maintaining a modular architecture that allows for easy customization and extension.

- **Security:** The system should be designed with security in mind, with multiple layers of security controls to protect against both external and internal threats. This includes implementing secure authentication and authorization mechanisms, using encryption to protect sensitive data, and conducting regular security audits and vulnerability testing.

8 Preliminary Schedule and Budget

The development of the Library Management System is estimated to take eighteen months. The project will include design, development, testing, and deployment phases. The project will be managed using agile development methodologies.

Requirements Gathering:
Timeline: 1-3 weeks
Cost: $20,000-$30,000

Design and Architecture:
Timeline: 4-6 weeks
Cost: $40,000-$60,000

Development:
Timeline: 5-6 weeks
Cost: $180,000-$280,000

Testing and Quality Assurance:
Timeline: 2-3 weeks
Cost: $40,000-$60,000

Deployment and Maintenance:
Timeline: Ongoing
Cost: $60,000-$120,000 per year