Software

Requirements

Specification

Version 1.0

June, 2024

Table of Contents

[**1** **INTRODUCTION** 5](#_Toc172040831)

[**1.1** **Purpose** 5](#_Toc172040832)

[**1.2** **Document Conventions** 5](#_Toc172040833)

[**1.3** **Intended Audience and Reading Suggestions** 5](#_Toc172040834)

[**1.4** **Project Scope** 5](#_Toc172040835)

[**1.5** **References** 5](#_Toc172040836)

[**1.6** **Overview** 5](#_Toc172040837)

[**2** **OVERALL DESCRIPTION** 5](#_Toc172040838)

[**2.1** **Product Perspective** 5](#_Toc172040839)

[**2.2** **Product Features** 6](#_Toc172040840)

[**2.3** **User Classes and Characteristics** 7](#_Toc172040841)

[**2.4** **Operating Environment** 7](#_Toc172040842)

[**2.5** **Design and Implementation Constraints** 7](#_Toc172040843)

[**2.6** **Assumptions and Dependencies** 7](#_Toc172040844)

[**3** **SYSTEM FEATURES** 8](#_Toc172040845)

[**3.1** **User Management** 8](#_Toc172040846)

[3.2 **Loan Stocks Issuance** 8](#_Toc172040847)

[**3.3** **Product Management** 8](#_Toc172040848)

[**3.4** **Reporting** 8](#_Toc172040849)

[**3.5** **Return Loan** 9](#_Toc172040850)

[**4** **EXTERNAL INTERFACE REQUIREMENTS** 9](#_Toc172040851)

[**4.1** **User Interfaces** 9](#_Toc172040852)

[**4.2** **Hardware Interfaces** 9](#_Toc172040853)

[**4.3** **Software Interfaces** 9](#_Toc172040854)

[**4.4** **Communication Interfaces** 9](#_Toc172040855)

[**5** **NONFUNCTIONAL REQUIREMENTS** 9](#_Toc172040856)

[**5.1** **Performance Requirements** 9](#_Toc172040857)

[**5.2** **Safety Requirements** 9](#_Toc172040858)

[**5.3** **Security Requirements** 9](#_Toc172040859)

[**5.4** **Software Quality Attributes** 9](#_Toc172040860)

[**5.5** **Reliability** 9](#_Toc172040861)

[**5.6** **Scalability** 10](#_Toc172040862)

[**6** **APPENDICES** 10](#_Toc172040863)

[**6.1** **Glossary of Terms** 10](#_Toc172040864)

[**6.2** **Use Case Diagrams** 11](#_Toc172040865)

[**6.3** **Entity-Relationship Diagrams** 12](#_Toc172040866)

[**7** **CONCLUSION** 12](#_Toc172040867)

Table of Images

[Figure 1: The diagram shows the layout of stocks database system – entity-relationship model 5](#_Toc172019122)

[Figure 2*: Use Case Diagram* 10](#_Toc172019123)

[Figure 3 Entity-Relationship Diagrams 11](#_Toc172019124)

# **INTRODUCTION**

## **Purpose**

The purpose of this document is to specify the requirements for the Loan Stocks Management System (LSMS). This system aims to facilitate the management of loans and stocks, allowing users to track loans, manage investments in stocks through loans, and monitor financial performance

## **Document Conventions**

This document uses the following conventions

|  |  |
| --- | --- |
| **DB** | Database |
| **DDB** | Distributed Database |
| **ER** | Entity Relationship |
| **LSMS** | Loan Stocks Management System |
| **Admin** | Administrator, a user with privileged access rights |
| **Staff** | User with administrative privileges but restricted from editing user roles |
| **User** | Standard system user |

## **Intended Audience and Reading Suggestions**

This project is a prototype for the stocks management system and it is restricted within the college premises. This has been done under guidance mentor. This project is very useful for the loan stocks management team as well as for the user.

## **Project Scope**

The LSMS will cover the entire lifecycle of loan stocks, including issuance, tracking, redemption, and reporting. It will provide administrators and users with tools to manage loan stock portfolios efficiently.

The purpose of the online loan stocks management system is to ease stocks management and to create a convenient and easy-to-use application for users, trying to borrow products. The system is based on a relational database with its stocks management and reservation functions. We hope to provide a comfortable user experience.

## **References**

Laravel Documentation: <https://laravel.com/docs>

MySQL Documentation: <https://dev.mysql.com/doc/>

## **Overview**

The remainder of this document provides a detailed specification of functional and non-functional requirements for the LSMS.

# **OVERALL DESCRIPTION**

## **Product Perspective**

The Loan Stocks Management System is a web-based application developed using the Laravel framework, designed to facilitate the management of loans and investments in stocks. It operates as standalone system, interacting with relational database (e.g., MySQL) to store and manage user data, loan details, stock information, and their respective relationships.

The system is designed to provide users with tools to apply for loans, manage their loan portfolios. It includes administrative functionalities for managing users, approving loan applications, adding stocks, and overseeing system operations.

The system provides comprehensive functionalities tailored for three distinct roles:

Admin: Has full control over the system, including managing user roles, approving loan applications, adding stocks, and overseeing all operations.

User: Individual who can apply for loans, manage their loan portfolios, allocate stocks to their loans, and monitor investment performance.

Staff: Similar to admins but with restrictions; they can manage user accounts and assist in loan and stock management but cannot modify user roles or perform administrative tasks.

## **Product Features**

The major features of the stocks database system are depicted in the entity-relationship model (ER model).

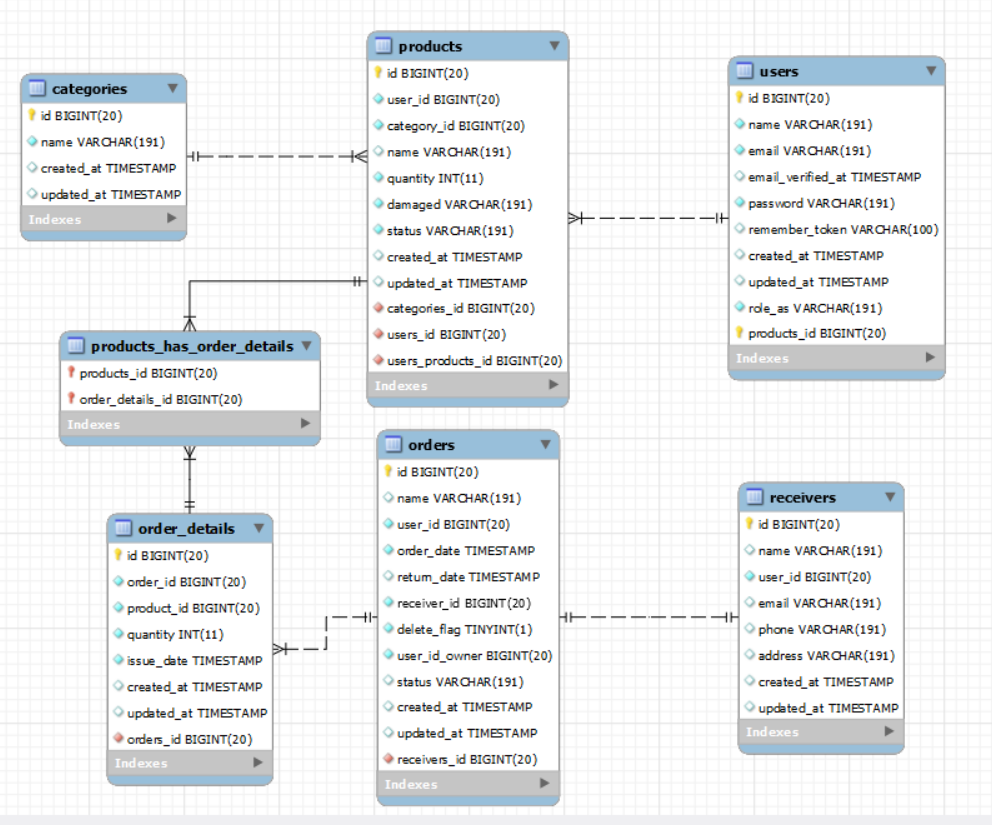


Figure 1: The diagram shows the layout of stocks database system – entity-relationship model

## **User Classes and Characteristics**

The system accommodates the following user classes:

Admin: Has the highest level of access and can perform administrative tasks such as managing user roles, approving loan applications, adding stocks, and generating reports.

User: Applies for loans, manages their loan accounts, allocates stocks to their loans, and views investment performance metrics.

Staff: Assists in user management, loan application processing, stock management, and provides support to users. They do not have access to administrative functions like role management.

Each user class benefits from role-specific functionalities and permissions tailored to their responsibilities within the organization.

## **Operating Environment**

The system operates within a web-based environment accessible via modern web browsers (e.g., Chrome, Firefox, Safari, Edge) on desktop and mobile devices. It is deployed on a web server capable different operation systems and devices.

## **Design and Implementation Constraints**

The system adheres to the following design and implementation constraints:

Laravel Framework: Developed using Laravel version 10.x to leverage its MVC architecture for scalability, modularity, and maintainability.

Database: Utilizes a relational database management system (e.g., MySQL) for data storage, ensuring data integrity and efficient query handing.

Security: Implements robust security measures, including data encryption (HTTPS), secure authentication mechanisms, and role-based access control (RBAC) to protect sensitive information and functionalities.

Responsive UI: Follows responsive web design principles to provide a seamless user experience across various screen sizes and devices.

## **Assumptions and Dependencies**

Assumptions and dependencies include:

Internet Connectivity: Users and administrators require stable internet connectivity to access and interact with the system.

Browser Compatibility: The system supports major web browsers (Chrome, Firefox, Safari, Edge) for optimal performance and user experience.

Third-Party APIs: Integration with external APIs (e.g., for financial data) assumes reliable API availability and compatibility

# **SYSTEM FEATURES**

## **User Management**

**Description:** Manage users within the system.

* Admin Registration and Login
* User Role Management

**Functional Requirements:**

**FR1.1:** The system shall allow the admin to read, update users.

**FR1.2:** The system shall support role-based access control.

## **Loan Stocks Issuance**

**Description:** Manage loan stocks issuance and records.

* Loan Application Submission
* Loan Approval Process

**Functional Requirements:**

**FR2.1:** The system shall allow users to submit loan applications.

**FR2.2:** The system shall allow product owner to approve or reject loan applications.

## **Product Management**

**Description:** Manage products related to loans

* Product Inventory Management
* Product Issuance and Return
* Product Valuation

**Functional Requirements:**

**FR3.1:** The system shall allow stock managers to manage stock inventory.

**FR3.2:** The system shall track the issuance and return of products

## **Reporting**

**Description:** Generate various reports for analysis

* Loan Reports
* User Activity Reports

**Function Requirements:**

**FR4.1:** The system shall generate detailed loan reports.

**FR4.2:** The system shall generate user activity reports

## **Return Loan**

**Description:** Manage the return of loaned products.

* Return Loan Management

**Functional Requirements:**

**FR5.1:** The system shall allow users to return loan products.

**FR5.2:** The system shall update damage, record details of returned products

# **EXTERNAL INTERFACE REQUIREMENTS**

## **User Interfaces**

**Web-based UI:** User-friendly interfaces for all system functionalities.

## **Hardware Interfaces**

None required beyond standard server and client hardware.

## **Software Interfaces**

**Database:** MySQL for data storage.

## **Communication Interfaces**

**HTTP/HTTPS:** For secure communication between client and server.

# **NONFUNCTIONAL REQUIREMENTS**

## **Performance Requirements**

* The system should support concurrent access by multiple users without performance degradation.
* Response times for typical operations should be less than 2 seconds.

## **Safety Requirements**

* The system should ensure data integrity during transactions.

## **Security Requirements**

* User authentication should be based on secure login mechanisms (e.g., username/password).
* Data transmission should be encrypted using industry-standard protocols.

## **Software Quality Attributes**

* **Usability:** The system should be easy to use for all user roles.
* **Reliability:** The system should be available 99% of the time.
* **Scalability:** The system should be able to scale to accommodate more users and data

## **Reliability**

* The system should maintain data integrity and consistency at all times.
* Regular backups should be performed to prevent data loss.

## **Scalability**

The Laravel application architecture should support scaling to accommodate future growth in users and data volume.

# **APPENDICES**

## **Glossary of Terms**

#### ****Admin****

A user with privileged access rights who can perform administrative tasks such as managing user roles, approving loan applications, adding stocks, and generating reports.

#### ****Database (DB)****

A structured collection of data stored and accessed electronically from a computer system.

#### ****Distributed Database (DDB)****

A database in which storage devices are not all attached to a common processor.

#### ****Entity Relationship (ER)****

A data modeling technique that creates a graphical representation of the entities and their relationships to each other.

#### ****Loan Stocks Management System (LSMS)****

A web-based application designed to facilitate the management of loans and investments in stocks. It provides tools for applying for loans, managing loan portfolios, and overseeing system operations.

#### ****Product Owner****

A user who owns a product and has the authority to approve or reject loan applications for that product.

#### ****Responsive UI****

User interfaces that adapt to different screen sizes and devices to provide a seamless user experience.

#### ****Role-Based Access Control (RBAC)****

A method of regulating access to a computer or network resources based on the roles of individual users within an enterprise.

#### ****Secure Authentication Mechanisms****

Methods used to verify the identity of a user, typically involving a combination of a username and password or other secure credentials.

#### ****User****

A standard system user who can apply for loans, manage their loan portfolios, allocate stocks to their loans, and monitor investment performance.

#### ****User Interface (UI)****

The means by which the user and a computer system interact, including the use of input devices and software.

#### ****Waiting for Approval****

A status indicating that a user has submitted a loan application or product borrowing request and is awaiting the approval of the product owner.

## **Use Case Diagrams**

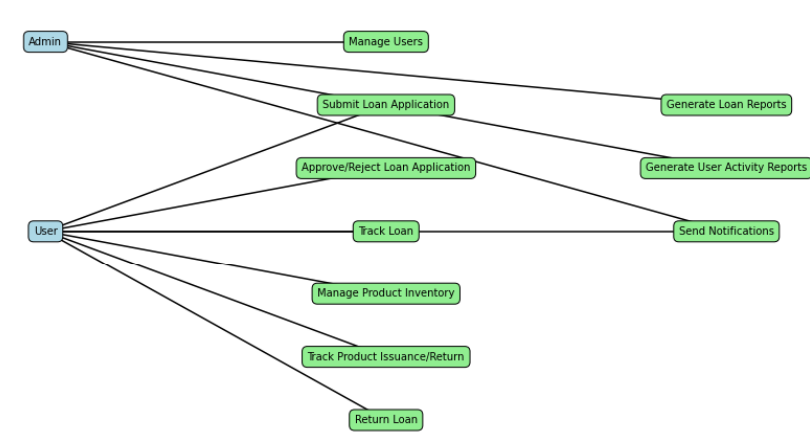


Figure 2*: Use Case Diagram*

## **Entity-Relationship Diagrams**

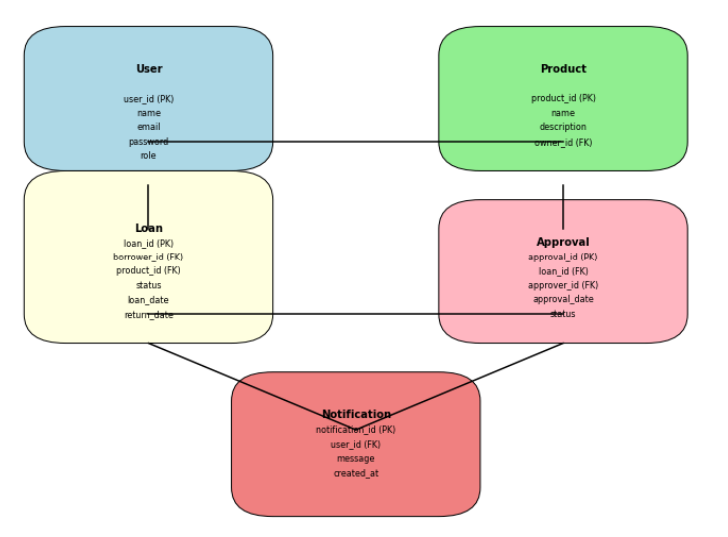


Figure 3 Entity-Relationship Diagrams

# **CONCLUSION**

The Software Requirement Specification (SRS) document for the Loan Stocks Management System provides a detailed and structured outline of the functional and non-functional requirements necessary to develop a robust and efficient system. This document serves as a critical foundation for the development process, ensuring that all stakeholders have a clear understanding of the system's objectives, features, and constraints.

The Loan Stocks Management System is designed to streamline the management of loan stock items, providing a seamless experience for both users and administrators. Key functionalities such as loan issuance, returns, inventory tracking, user management, and reporting are meticulously detailed to ensure comprehensive coverage of the system's capabilities. Additionally, the SRS addresses crucial non-functional requirements including performance, security, usability, and scalability, ensuring that the system meets high standards of quality and reliability.

By adhering to the specifications outlined in this document, the development team can create a system that not only meets the immediate needs of users but also provides a scalable solution for future growth and enhancements. The SRS serves as a contract between stakeholders and developers, ensuring alignment on the system's functionality and performance expectations.

In conclusion, the Loan Stocks Management System SRS sets the stage for a successful development process by providing clear, detailed, and agreed-upon requirements. This document will guide the project from inception through to completion, ensuring that the final product is robust, user-friendly, and capable of meeting the demands of its users. With a strong foundation in place, the Loan Stocks Management System is poised to become an invaluable tool for managing loan stock items efficiently and effectively.