# **Software Requirements Specification (SRS)**

## **1. Introduction**

### **Purpose**

This document provides a comprehensive specification for the Book Recommendation System. It is intended for developers, project managers, and stakeholders involved in the development and deployment of the system.

### **Scope**

The Book Recommendation System is designed to enhance user experience by providing personalized book suggestions and automated email notifications. It includes capabilities such as secure user registration and authentication, tracking of user interests based on browsing history and preferences, and a recommendation engine that uses collaborative and content-based filtering to offer personalized book recommendations.   
  
The system also sends regular email notifications with book suggestions tailored to user preferences, allows users to search and browse books by title, author, or genre, and enables users to rate books and provide feedback to improve future recommendations.

### **Definitions, Acronyms, and Abbreviations**

* **Admin**: Administrator with privileges to manage book entries.
* **User**: Standard user with access to browse and filter books.

### **Overview**

This SRS document details the functional and non-functional requirements of the Book Recommendation System, including user roles, system features, data handling, and external interface requirements.

## **2. Overall Description**

### **Product Perspective**

### The Book Recommendation System is an independent module that integrates with an existing library management system to provide personalized book recommendations based on user activity and preferences.

### **Product Functions**

* User authentication and role management
* Book catalog browsing and filtering by genre
* Regular email notifications with book suggestions
* Administrative functions for book management

### **User Classes and Characteristics**

* **Admins**: Manages book entries (add).
* **Regular Users**: Browse, filter, and view books.

### **Operating Environment**

The system will operate on web browsers and mobile devices, interfacing with a backend server and a MySQL database.

### **Design and Implementation Constraints**

* Must support secure authentication mechanisms.
* Should handle high volumes of real-time data processing via Kafka.
* Must ensure data consistency and integrity within the MySQL database.

### **User Documentation**

User manuals and online help guides will be provided, covering both Admin and Regular User functionalities.

### **Assumptions and Dependencies**

* Users have basic internet access and web browser capabilities.
* The system relies on the proper functioning of the Kafka and MySQL services.

## **3. System Features**

### **3.1 User Roles and Authentication**

#### **Description and Priority**

Support two user roles: Admin and Regular User. High priority as it forms the basis of system access control.

#### **Functionality**

* **Admin**: Can add books.
* **Regular User**: Can browse and filter books.

### **3.2 Book Catalog and Filtering**

#### **Description and Priority**

Allow users to browse and filter books by genre. High priority for user engagement.

#### **Functionality**

* List books.
* Filter books by genre.

### **3.3 Book Recommendations**

#### **Description and Priority**

Generate dynamic book recommendations based on user activity. High priority to personalize the user experience.

#### **Functionality**

* Process user data via Kafka.
* Recommend books based on processed data.

### 

### **3.4 Book Management (Admin Only)**

#### **Description and Priority**

Enable Admins to manage book entries. Medium priority to maintain an updated catalog.

#### **Functionality**

* Add new books.

### **3.5 Book Details**

#### **Description and Priority**

Display detailed information about each book, including summary, author bio, and reviews. Medium priority for comprehensive user experience.

#### **Functionality**

* View book summary, author bio, and reviews.

## **4. Detailed System Design**

**User Interface**

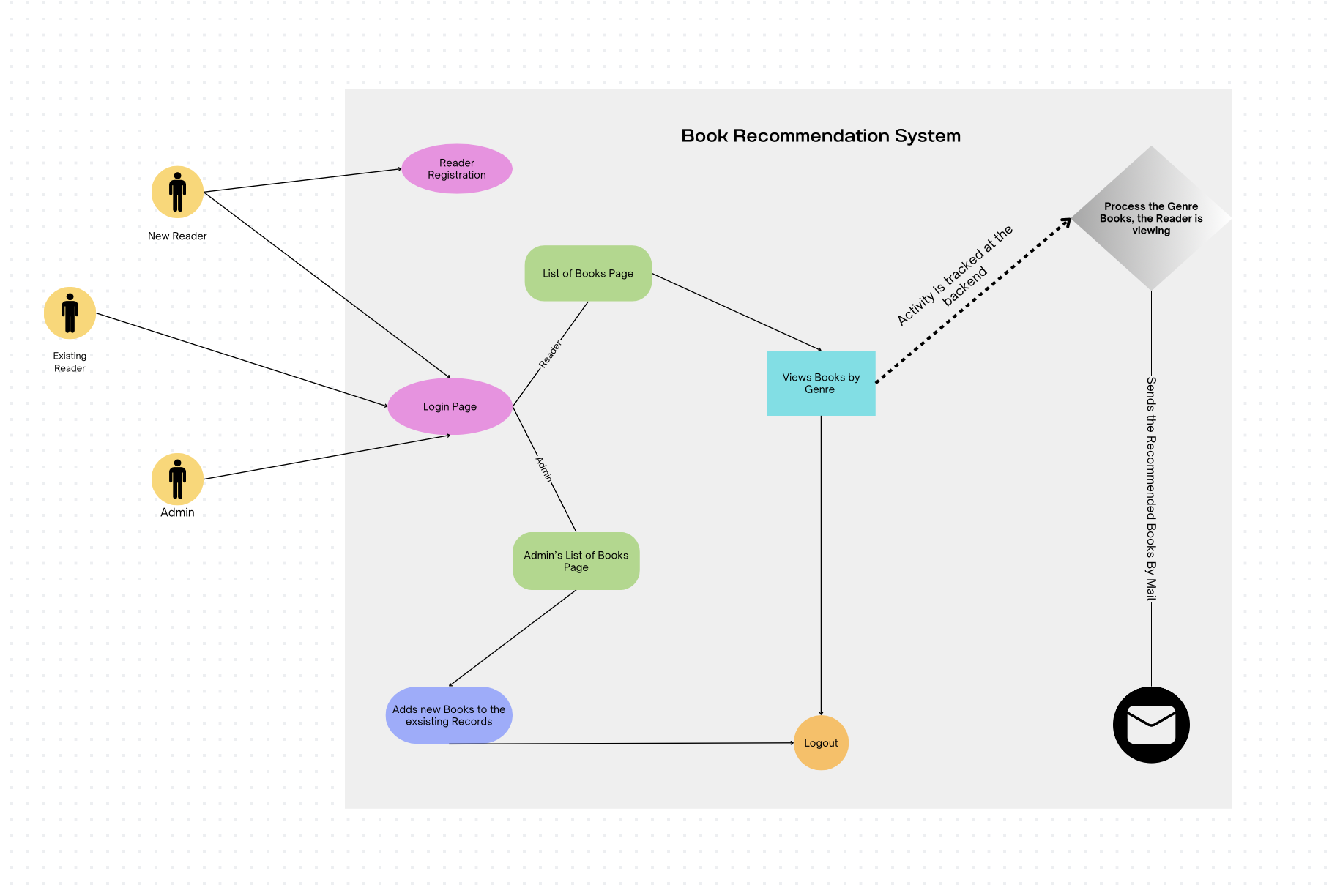
* **Purpose**: Provides interaction points for users and admins.
* **Interface**: Web pages and mobile views.
* **Functionalities**: User login, registration, browsing books, viewing recommendations.
* **Performance Characteristics**: Responsive design, fast loading times.

#### **Backend Server**

* **Purpose**: Handles business logic and API requests.
* **Interface**: RESTful APIs.
* **Functionalities**: User authentication, book management, recommendation generation.
* **Performance Characteristics**: High availability, low latency.

#### **Recommendation Engine**

* **Purpose**: Generates personalized book suggestions.
* **Interface**: Interacts with Kafka and database.
* **Functionalities**: Process user data, generate recommendations.
* **Performance Characteristics**: Real-time processing.

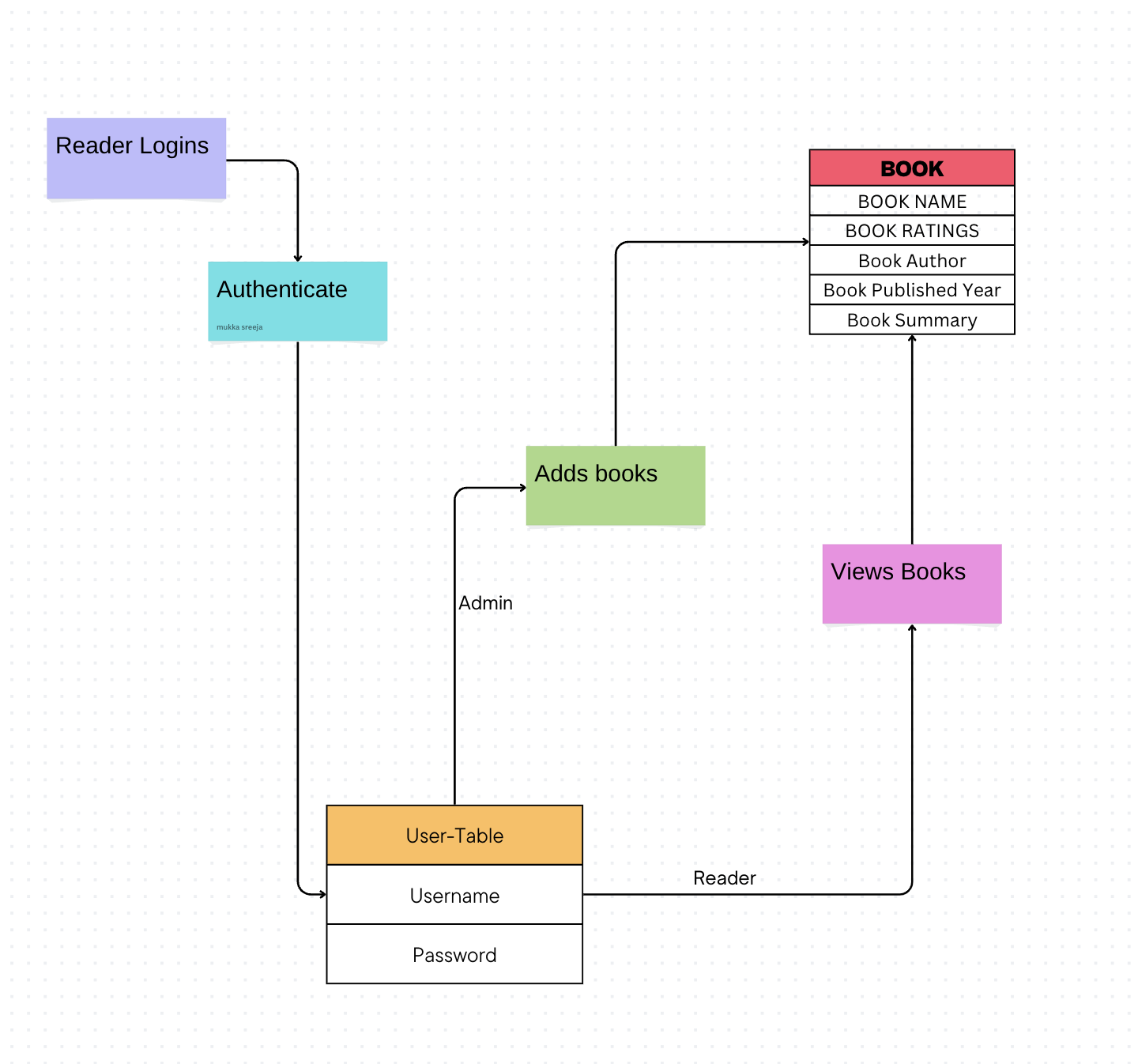


## **5. Data Design**

### **Data Architecture**

The system uses a relational database to store user and book data. Apache Kafka processes real-time user interactions.  
  
**Database Design**

* **Users Table**: Stores user information.
* **Books Table**: Stores book information.



## **6. External Interface Requirements**

### **User Interfaces**

* **Admin View**: Options for adding books.
* **User View**: Options for browsing and filtering books and viewing detailed book information.

### **Hardware Interfaces**

No specific hardware requirements, as the system will be web-based.

### **Software Interfaces**

* **Backend Server**: Handles API requests.
* **MySQL Database**: Stores user and book data.
* **Kafka**: Processes real-time data for recommendations.
* **GCP**: To host the website online.

### **Communications Interfaces**

Standard HTTP/HTTPS protocols for communication between client and server.

## **7. Other Nonfunctional Requirements**

### **User-Friendly Interface**

* The system should have an intuitive and easy-to-use UI design for seamless book exploration.

### **Performance**

* Fast loading times for book listings and details.
* Authentication and recommendations should meet the specified performance requirements.

### **Scalability**

* The system should handle a growing database of books and increasing user interactions without performance degradation.

## **8. Detailed Requirements**

### **Functional Requirements**

* **FR1**: The system shall authenticate users using secure login credentials.
* **FR2**: The system shall allow Admins to add books.
* **FR3**: The system shall enable users to browse and filter books by genre.
* **FR4**: The system shall provide book recommendations based on user activity data processed by Kafka.
* **FR5**: The system shall allow users to specify their favorite genres.
* **FR6**: The system shall capture user reading habits and preferences over time.
* **FR7**: The system shall display detailed information about each book, including summary, author bio, and reviews.

### **Non-functional Requirements**

* **NFR1**: The system shall process user data securely.
* **NFR2**: The system shall be accessible on various web browsers and mobile devices.
* **NFR3**: The system shall have an intuitive and user-friendly interface.
* **NFR4**: The system shall ensure fast loading times for book listings and details.
* **NFR5**: The system shall be scalable to handle a growing database and user interactions.

## **9. Validation Criteria**

### **Validation Methods**

* User acceptance testing to validate functionality.
* Performance testing to ensure response times.

### **Success Criteria**

* Successful login for both Admin and Regular User roles.
* Accurate and relevant book recommendations.
* Efficient book management by Admins.
* User satisfaction with the browsing, filtering, and recommendation features.