Software Requirements Specification (SRS) for Library Management System

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the Library Management System. It outlines the system's functionality, performance, and design constraints, serving as a blueprint for development and validation.

1.2 Scope

The system is a single-page web application for managing library resources. Its core functions include managing books, members, and transactions (borrowing and returning books). The system is intended for use by library staff to streamline daily operations.

1.3 Definitions and Acronyms

- **System**: The Library Management System.
- **User**: A library staff member or administrator.
- **Book**: A single physical or digital resource in the library's collection.
- **Member**: A registered patron of the library.
- **Transaction**: A record of a book being borrowed, returned, or renewed.
- **UI**: User Interface.
- **API**: Application Programming Interface.

2. Overall Description

2.1 Product Perspective

The system is a standalone web application designed for a web browser. It is built using HTML, Tailwind CSS, and JavaScript, with a focus on creating a single-file, self-contained application. It uses a Firebase Firestore database for data persistence.

2.2 Product Functions

The system provides the following major functionalities:

- **Book Management**: Allows the user to view, add, edit, and delete book records.
- **Member Management**: Enables the user to view, add, edit, and delete member records.
- **Transaction Management**: Facilitates the borrowing and returning of books by members. It also tracks overdue books.
- **Dashboard**: Provides a summary of key metrics, such as total books, total members, and overdue books.

2.3 User Characteristics

The target user is a library staff member with basic computer literacy. They will use the system to perform daily tasks like checking books in and out, registering new members, and managing the library's catalog.

2.4 Design and Implementation Constraints

- **Technology**: Must be implemented as a single HTML file with embedded JavaScript and CSS.
- **Styling**: Use Tailwind CSS for all styling and responsive design.
- **Data Storage**: Data must be stored and retrieved from a Firebase Firestore database.
- **Interaction**: The application shall not use alert() or confirm() for user notifications. Modals or custom UI elements should be used instead.
- **Responsiveness**: The UI must be fully responsive and optimized for various screen sizes, including desktop and mobile.

3. Specific Requirements

3.1 Book Management Module

- **3.1.1 View Book Catalog**: The system shall display a list of all books, including title, author, ISBN, and availability status.
- **3.1.2 Search Books**: The user shall be able to search for books by title, author, or ISBN.
- **3.1.3 Add New Book**: The system shall provide a form to add a new book record.
- **3.1.4 Edit Book Details**: The user shall be able to modify the details of an existing book.
- **3.1.5 Delete Book**: The user shall be able to delete a book record from the system.

3.2 Member Management Module

- **3.2.1 View Members**: The system shall display a list of all registered members.
- **3.2.2 Search Members**: The user shall be able to search for members by name or member ID.
- **3.2.3 Add New Member**: The system shall provide a form to add a new member record.
- **3.2.4 Edit Member Details**: The user shall be able to modify a member's details.

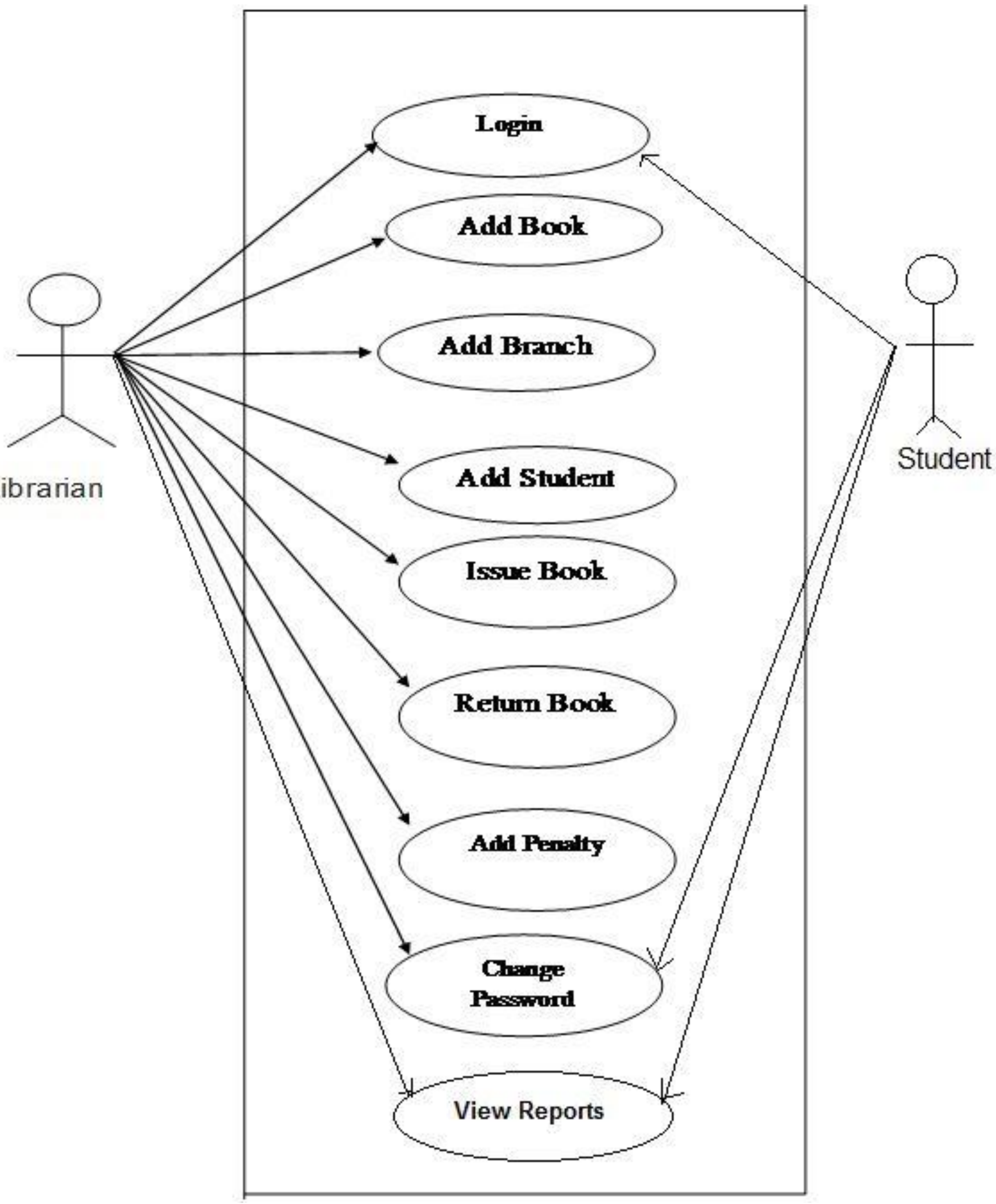
3.3 Transaction Management Module

- **3.3.1 Borrow Book**: The system shall record a transaction when a member borrows a book, linking the member and book records. The book's status must be updated to Borrowed.
- **3.3.2 Return Book**: The system shall record a transaction when a member returns a book. The book's status must be updated to Available.
- **3.3.3 Track Overdue Books**: The system shall automatically identify and list books that have not been returned by their due date.

4. System Diagrams

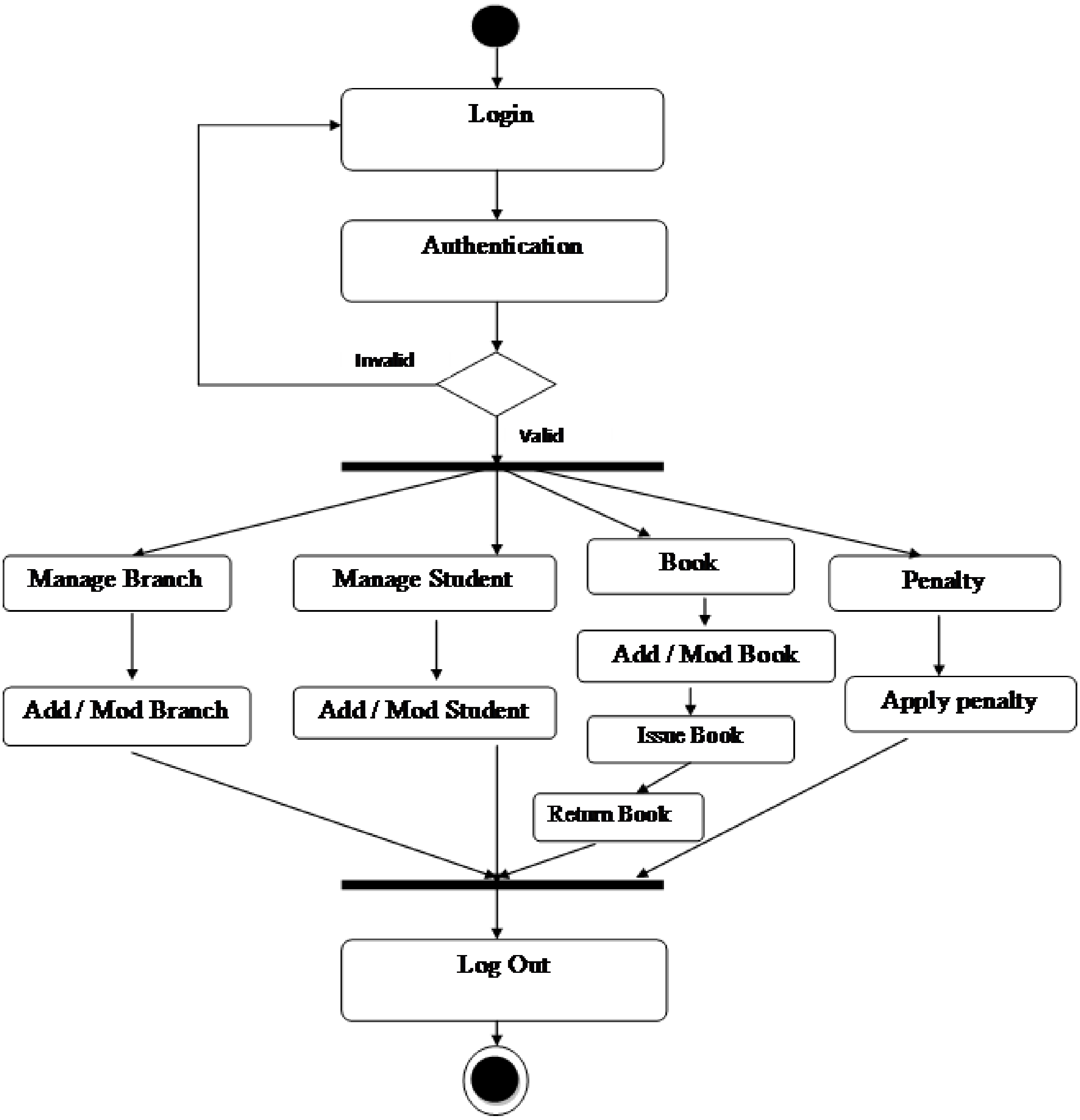
4.1 Use Case Diagram

This diagram details the primary functions and their related actions that a library staff member can perform.



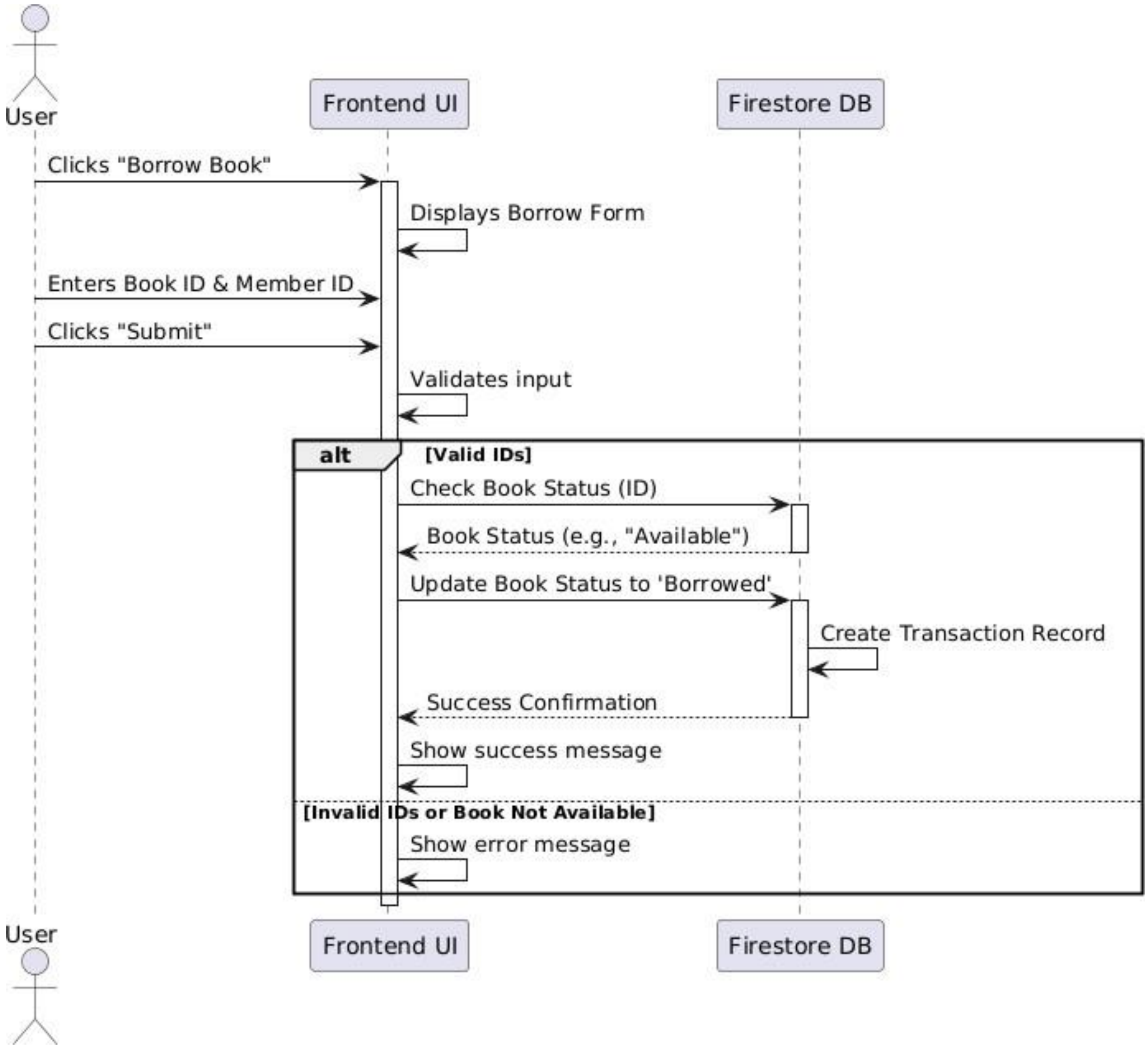
4.2 Activity Diagram

This diagram visualizes the comprehensive workflow for a staff member returning a book.

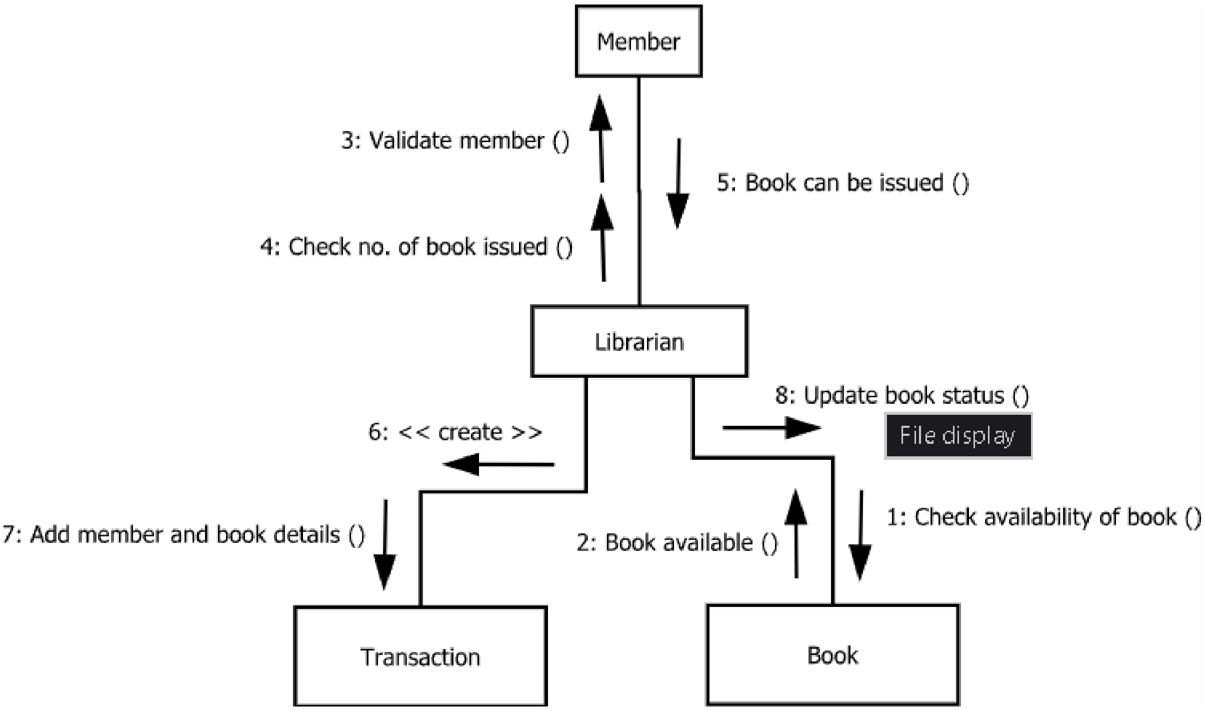

Login
Authentication
Invalid
Valid
Manage Branch
Manage Student
Book
Penalty
Add / Mod Book
Add / Mod Branch
Add / Mod Student
Apply penalty
Issue Book
Return Book
Log Out

4.3.1 Sequence Diagram

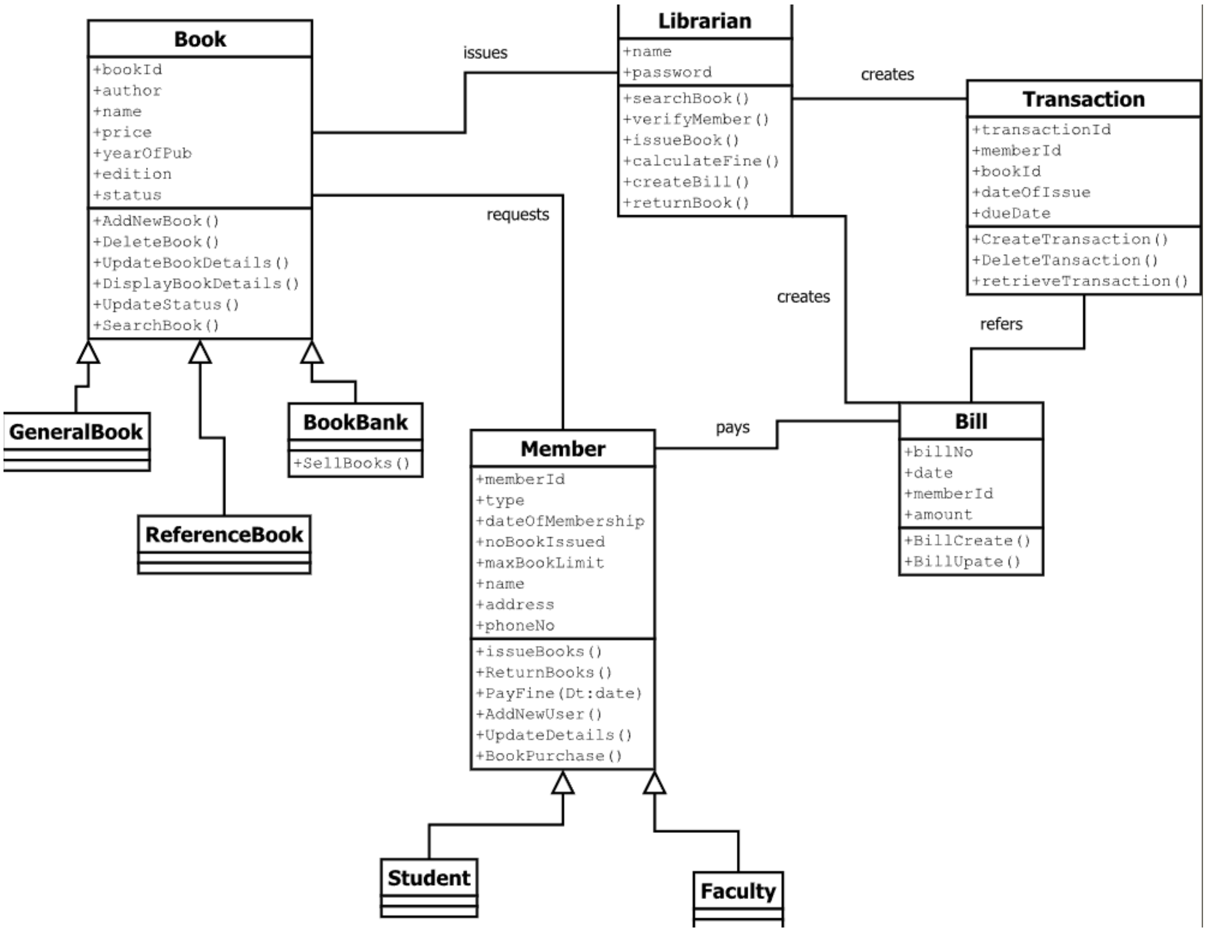
This diagram shows the sequence of events for a user borrowing a book, from the UI to the database.



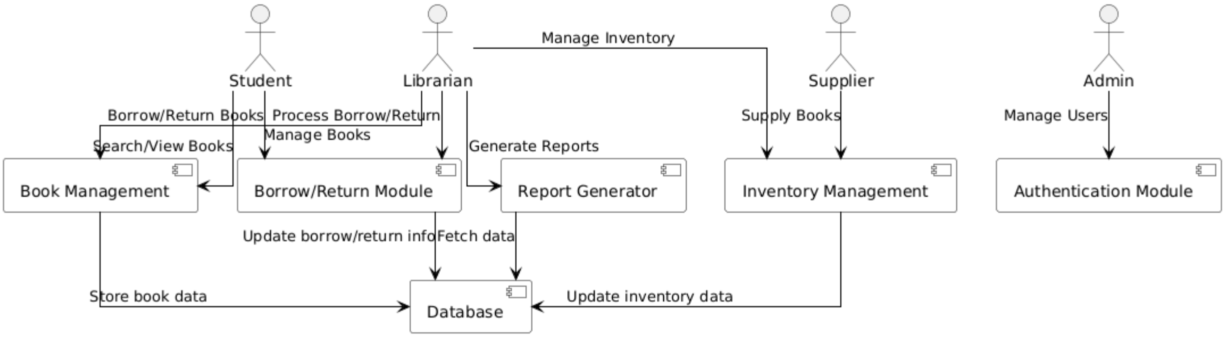
4.3.2 Collaboration Diagram



4.4 Class Diagram



4.5 Component/Deployment Diagram



5 Conclusion

This SRS provides a detailed and structured overview of the requirements for the Library Management System. The document and its accompanying UML diagrams will serve as a foundational guide for the development team. By clearly defining the scope, functionalities, and technical constraints, we ensure that the final product will meet the needs of the library staff and patrons effectively. This document is a critical reference for all project stakeholders and will be maintained throughout the development lifecycle.