Requirements Analysis and Architecture Design

# Requirements

This section will outline the functional and non-functional requirements for a book tracking application, splitting them into requirements needed for a successful MVP and requirements needed for additional features.

## Functional Requirements

### MVP (Minimum Viable Product) Requirements

Allow the user to make an account or log in

* A section where the user can input their email address and password for account creation is required
* A section where the user can input their email address and password for logging in is required
* A connection to a database where the user’s credentials can be securely stored is required
* An authentication algorithm that will allow or deny the user access to the application is required

Allow the user to look up published books

* A search bar for accepting input associated with book keywords is required
* A connection to an API where book information can be fetched from is required
* A section where individual book objects will be displayed is required
* Allow the user to add books they are interested in into their personal digital library
* A button that will add the books to the library is required
* A connection to a database where the user’s books can be stored is required
* A method that will fetch the user’s books from the database needs to be implemented
* A section where the information about the user’s books will be displayed is required

Allow the user to delete books they don’t want to keep in their library anymore

* A button that will remove the books from the library is required
* A method that will remove the book from the database needs to be implemented

Allow the user to identify which books they have already read in their library

* A button that will set a book in the library as read is required
* A method that will update the details about the book in the database needs to be implemented

Allow the user to set their own reading goal by specifying the number of pages per day that they want to read

* A section where the user can input their reading goal is required
* A button that will let the user indicate that they have reached their daily goal is required
* An algorithm that will reset the user’s goal everyday needs implementing

### Extended Requirements

Allow the user to rate the books they have read

* Interactive star icons that will let the user indicate how much they have enjoyed reading a book are required
* An algorithm that will only let the user rate the books that are marked as read needs to be implemented

Allow the user to recommend the books they have read to their friends

* A section that will display all the user’s friends is required
* A section that will let the user look up other users and add them to their friend’s list is required
* A modification of the database to allow for a connection between users that will represent their friendship is required
* A button that will share a book with another user is required
* A section that will display the books recommended by other users is required

Allow the user to be fed recommendations based on their reading history

* A section that will display the books recommended by the application is required
* An algorithm that will choose books that are not in the user’s library but match the attributes of the books the user has read previously needs implementing

Allow the user to update their reading goal based on the number of pages they want to read and how many books they want to read at the same time

* The reading goal section will need to be modified with an additional field for the number of books the user wants to read at the same time
* An algorithm that will calculate the number of pages per book the user needs to read based on their reading goal and the length of each book need to be implemented

## Non-functional Requirements

Security:

* User login credentials must be encrypted using industry-standard encryption algorithms before storing in the database
* The password complexity should be enforced by introducing minimum requirements for its length and the inclusion of numbers, symbols, uppercase, and lowercase characters
* Ensure that the users can only access their own information and not the other users’ of the application

Availability:

* The application source code and database data need to be backed up to the cloud in case of hardware failure
* Back up the database state regularly in case of database failure or accidental deletion

Reliability:

* Any errors that the application encounters should be handled well and should be displayed to the user in a clear and concise manner
* Ensure database integrity by rolling back to the previous version of it in case of a method interacting with the database not performing the correct action

Performance:

* The user interface should have an average response time of less than 3 seconds when the user interacts with it
* All the data in the database should be properly indexed to ensure quick access

Capacity:

* The database should be designed to handle a large amount of data about the users and their books
* The application should be able to handle multiple users accessing the database at the same time

Continuity:

* Ensure that upon the failure of the system, the errors are fixed within 1 hour of their discovery
* The implementation of version control should be performed, so that the user can access a previous version of the application, in case the latest one fails

Supportability:

* Provide the users with a comprehensive guide to how to use the application that will cover all its features
* There should be a logging system put in place that can track the application’s errors and its availability

Maintainability:

* The source code should be written in compliance with industry standards and utilising the object oriented architecture for easy code modification
* The source code should include unit tests for all the parts of the application and considering all possible cases to decrease the possibility of errors occurring

Scalability:

* The database should be designed in such a way that handles an increase in user and book data without its degradation
* Each method should be designed to run independently, so that the application can handle an increased amount of user requests

# Class Diagram (Partial)

A diagram of a computer

Description automatically generated

# Use Case Diagram

A diagram of a book reader

Description automatically generated

# Activity Diagram (Login System)

A diagram of a computer program

Description automatically generated

# Sequence Diagram (Book Library Management)

A screenshot of a computer program

Description automatically generated

# Entity Relationship Diagram (Database Tables)

A screenshot of a computer

Description automatically generated

# Timing Diagram (Fetching Book Info From an API)

A blue diagram with black lines

Description automatically generated