Software Requirements Specification

for

BookDiary project

Prepared by

Yuliana Yurchenko

2023

Contents

1 Project Description

1.1 Project Scope

The goal of the project is to provide a platform for storing information about books that have been read or are still being read. The system allows you to add book titles, book author, book images, book reviews, and edit all this information. It helps to track the progress in reading, namely, you can record the number of read pages of the selected book on a certain day and then see the statistics of the read books for a certain period of time. There is an opportunity to set a reminder of the planned reading and track progress.

1.2 Product Features

The system has the following major features:

1. Storing a list of books that user plans to read;
2. Noting the book and the number of page on which user stopped reading last time;
3. Saving notes and quotes of pages or lines that interested the user;
4. Writing own review and putting a mark to book that user already read;
5. Checking statistics of read pages during day/week/month/year;
6. Setting a reminder of the planned reading.

1.3 Operating Environment

Operating environment for the BookDiary system is as listed below:

1. Database created using SQLite;
2. Android operating system;
3. .Net platform.

1.4 Assumptions and Dependencies

The system will be developed by using the .NET platform with C# language. Other dependencies will also include:

1. NuGet;
2. SQLite;
3. Entity Framework Core.

2 Requirement specifications

2.1 Functional Requirements

This section contains all functional requirements of this project. They describe how the system should work and behave, so these requirements are very important and must be implemented in full.

2.1.1 User Functional Requirements

The list of the post office employee functional requirements descriptions:

1. The user should be able to register and log in;
2. The user should be able to switch between ToRead/InProgress/Done books;
3. The user should be able to add a book to any of these lists;
4. The user should be able to remove a book from any of these lists;
5. The user should be able to move a book from ToRead list to InProgress list or from InProgress to Done list;
6. The user should be able to search among existed books by their title or author;
7. The user should be able to note the number of page she/he stopped reading last time for any book from InProgress list;
8. The user should be able to write own review for any book from Done list;
9. The user should be able to put a mark (on a five-point scale) for any book from Done list;
10. The user should be able to sort books from Done list by their marks;
11. The user should be able to check the statistics of total pages read during day/week/month/year;

12. The user should be able to create a reminder of the planned reading for chosen date and time.

2.2 Non-functional Requirements

This section contains all non-functional requirements of this project. They describe what the system is supposed to be, different system properties. They also concentrate on the user's expectations, so it is good to have them, although these requirements are non-mandatory.

2.2.1 Performance

The list of the performance requirements descriptions:

1. The application should load and be usable within 2 seconds;
2. The system should be able to work without Internet connection;
3. The system should be able to support any number of users;
4. Response time for queries and updates should be not over 500 ms.

2.2.2 Reliability

The list of the reliability requirements descriptions:

1. The system should work even if errors have occurred;
2. The system should always be available;
3. Downtime after a critical failure should not exceed 4 hours.

2.2.3 Usability

The list of the usability requirements descriptions:

1. The interface shall be easy to learn without a tutorial and allow users to accomplish their goals without errors.

2.2.4 Supportability

The list of the supportability requirements descriptions:

1. The system should be available for IOS operating system.

2.2.5 Maintainability

The list of the maintainability requirements descriptions:

1. The application should use continuous integration so that features and bug fixes can be deployed quickly without downtime;
2. The system should be covered with unit tests.
3. The system should be easy for testing.

2.2.6 Security

1. User's password should be stored in database as an encrypted hash.

3 Other Requirements

3.1 Use case diagram

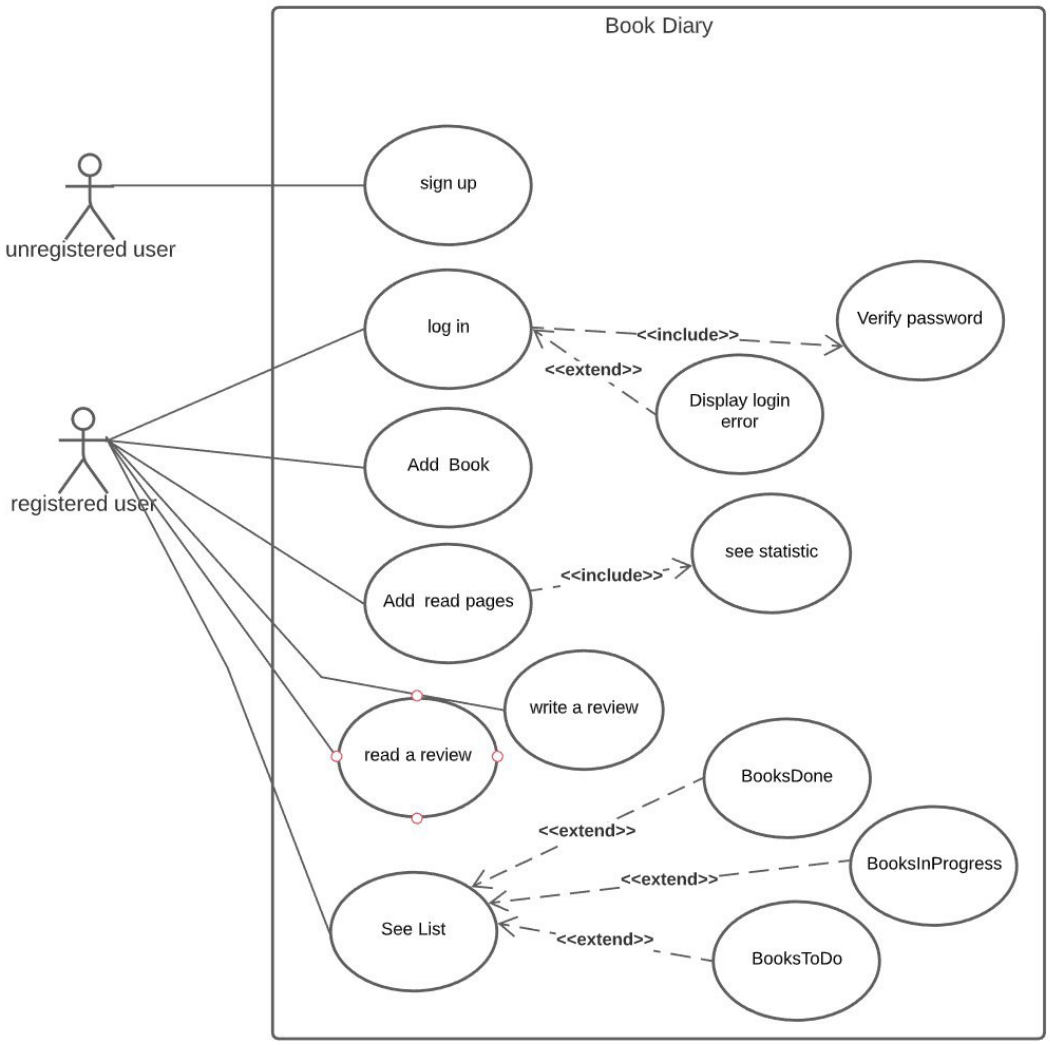

Figure 1: Use case diagram of the project

3.2 User Stories and Acceptance Criteria

User story:

As a user I want to register.

Acceptance criteria

- The user should fill in all required fields (nick name, first name, last name, email, password);
- The user should write valid email;

- The user should repeat password twice;
- The user should re-enter via invitation link which is sent by specified email address.

User story:

As a user I want to be able to see all added books and search among them.

Acceptance criteria

- The user is able to see all added books;
- The user is able search a certain book by its title or author.

User story:

As a user I want to move a book to Done list.

Acceptance criteria

- The user should choose a book from InProgress list;
- The user is able to manually move a book from InProgress list to Done list; or increase the number of read pages to the book's total number of pages to move the book to Done list automatically.
- Since a book is in Done list the user is able to write a review and/or put a mark for it.