T.C

MANİSA CELÂL BAYAR UNIVERSITY

ENGINERRING FACULTY

COMPUTER ENGINEERING DEPARTMENT

Personal Bookshelf Web Application with Gutenberg Project

Requirements Document

Prepared by

Mustafa Deniz Demir 180316043
Okan Hoşyılmaz 180316046
Berkay Şahin 170316012
Ayşenur Büyükbal 170316007

Instructor

Assist. Prof. Dr. Didem ABİDİN

FALL 2021-2022

TABLE OF CONTENT

Preface

This is the first version of the system and this document focuses on the requirement specification phase of the project. Prepared for end user and associated stakeholders by software development team.

Introduction

The project *"Personal Bookshelf Web Application with Gutenberg Project"* provides more visual and personal reading experience for books are available in *Gutenberg Project* library to user. Our methodology is KANBAN and while this project developing, "Jira Software" (CASE Tool) is used to manage the project. Last view of KANBAN board on Jira is available in the "Appendices" part of the document.

Glossary

scenario: A scenario is a scene that illustrates some interaction with a proposed system.

.txt format: Text is a human-readable sequence of characters and the words they form that can be encoded into computer-readable formats.

responsive design: Responsive design is an approach to web page creation that makes use of flexible layouts, flexible images, and cascading style sheet media queries. The goal of responsive design is to build web pages that detect the visitor's screen size and orientation and change the layout accordingly.

1. User Requirements Definition

The user shall register and **create an account** in the system. When the user has been created an account, his/her bookshelf created in the profile as well. The user can **search** through the books as part of the *Gutenberg Project* and **add** them to his/her bookshelf. When a user logs in to the system, he/she should be able to see the **last pages** which she/he has stopped for each book. The user can **see all the books on screen** which are she/he has added to own bookshelf at the same time. Books should be on the screen with their **name, author, and cover**. When the user clicked one of the books, that book should be **continued** from the page that user has read lastly. Books should be in **text format**. When the user finishes the book or he/she doesn't want to read anymore, he/she should be able to **remove** the book. Each user should be able to **access only his/her bookshelf** from his/her account. The system shall be **easy** to use and learn.

Scenarios

- **Creating Account**

Initial assumption: User doesn't have account yet and this is the first meeting system.

Normal: A user wants to read a book from *Gutenberg Project,* so he/she need to be created and account first. They need to create their account with using name, surname, and e-mail address and also they need to set a password for their account. On completion of the creating account, he/she needs to register with their e-mail address and password. After that they can access any book inside of the *Gutenberg Project*.

What can go wrong: Users e-mail address may be invalid so they can't create an account with this e-mail address. If user tries, s/he cannot create a different account with same e-mail address they need to use new e-mail for multiple accounts. Users' passwords need to be minimum length and safety criteria if their password does not match with these rules they cannot create an account.

System State on Completion: User created an account and now s/he can log in to the system to access this bookshelf by entering his/her information.

- **Logging-in**

Initial assumption: User has an account which is created before.

Normal: User wants to read book from *Gutenberg Project,* so he/she need to log-in their account with using their e-mail address and passwords which set before. After they typed their information system will check their e-mail address and password if match he/she can log-in successfully. On completion of the log-in he/she can read any book from inside the *Gutenberg Project.*

What can go wrong: If the password and e-mail address cannot match user cannot log-in his/her account.

System State on completion: User logged-in successfully in his/her account and s/he can pick any book which he/she wants to read from inside the *Gutenberg Project.*

- **Adding new book to the bookshelf**

Initial assumption: A user open the website and log-in his/her account successfully.

Normal: User goes for search a book, types book name or author name to search bar. Results come up to bottom of the bar. User can choose options to go to that book.

What can go wrong: User cannot find the book which they're looking for. User may do typo while typing the name of the book or author, thus results may not be accurate.

System State on Completion: Either the searching book is added to the bookshelf for reading anytime or after seeing the book's info, user decided not to add this book to the bookshelf, so the book is not added.

- **Reading book**

Initial assumption: A user open the website and log-in his/her account successfully.

Normal: User wants to read book and display his/her account. Then s/he chooses one of the books from display screen and continues where s/he stopped reading lastly. If this is the first time of reading, so book's first page will be showed.

What can go wrong: Users may forget the add a book which they read for a while and after that they lose their all reading progresses in this book.

System State on Completion: User reads the book for a bit or doesn't make any changes in his/her position. After this reading process, the place the reader stopped is updated.

- **Removing a book from the bookshelf**

Initial assumption: A user open the website and log-in his/her account successfully. User has books in bookshelf.

Normal: User displays his/her books in bookshelf. S/he can remove one of the books while s/he is in this displaying screen.

What can go wrong: User may remove a book from bookshelf unintentionally but still user lose the saved reading progress of the book.

System State on Completion: The removed book doesn't be shown in the display screen anymore (until it added again). Also, that book's information is not keeping anymore.

2. System Architecture

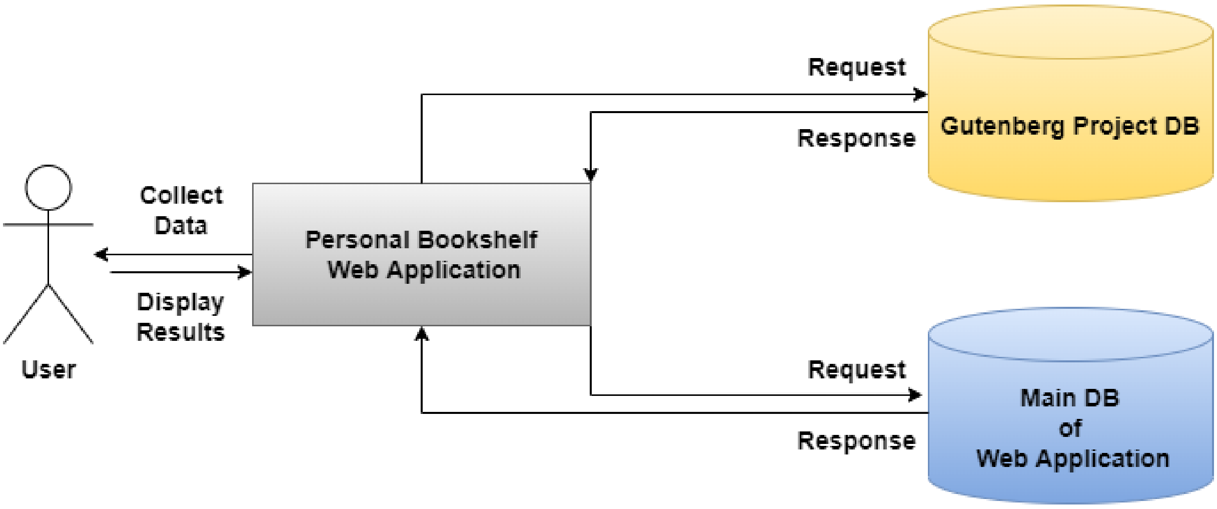

Figure 2.1 System Architecture Model

3. System Requirements Specification

The system's user requirements are described above. We can list them in two categories:

Functional Requirements

1) User shall create account in the system.
2) User shall search book from *Gutenberg Project*'s available books.
3) User shall add book to bookshelf
4) User can see the last read pages of each book when the bookshelf is opened.
5) Use can display all the added books at the same time
6) When the user opened the book, user shall continue reading from where s/he stopped lastly.
7) User shall remove a book anytime.

Non-Functional Requirements

1) Books in the bookshelf should be shown with names, authors, and covers.

2) Bookshelves must be private, so account owner only access his/her bookshelf.

3) The system shall be easy to use.

4) Books shall be in .txt format.

3.1 System Requirements

System requirements is a structured document which setting out detailed descriptions of the system's functions, services, and operational constraints. Defines what should be implemented. By using "User Requirements" we can obtain system requirements. After some investigation of user requirements, we can detail them and list our system requirements.

In this project, system's stakeholder is reader.

[1]. User shall create account in the system

a) Users enter e-mail, password, name, surname, and date of birth information in the system.

b) If e-mail is valid and password is strength enough then the system shall register this account with given information. Otherwise, system shall warn user.

c) E-mail validation shall depend on if e-mail already used or not, if that e-mail exists or not.

[2]. User shall search book from Gutenberg Project's available books.

a) User can use search bar in the system and access books from *Gutenberg Project.*

b) User must give some keywords to system to search for a book.

c) User can search books with book name or author name.

d) If there is a typo in keywords, system doesn’t correct the given keyword.

e) Searching function only covers books which are available in *Gutenberg Project.*

[3]. User shall add book to bookshelf

a) User must search for a book before adding operation.

b) Adding operation is only accessible from results of searching.

c) User shall display one of the books from results and add it to bookshelf.

d) This operation has a constraint which is addible books are limited with Gutenberg Project’s scope.

[4]. User can see the last read pages of each book when the bookshelf is opened

a) System records the user’s last page information for each book.

b) When the account is opened, this information table shall be shown on the screen.

c) This table includes only book name and page number.

[5]. User can display all the added books at the same time

a) For each account, system keeps information about bookshelf content.

b) When the web application is opened each book will be shown on the main page screen.

[6]. Books in the bookshelf should be shown with names, authors, and covers.

a) Covers of the books shall be shown on the screen in a responsive structure. Which means that cover sizes must be clear on different screen sized devices.

b) Book’s names shall be shown between cover and author name. Book name’s size must be bigger than author name.

c) Author names shall be placed at the bottom.

[7]. When the user opened the book, user shall continue reading from where s/he stopped lastly.

a) When the reader opens a book system must remember where user stopped and automatically continue from there to read.

b) If it is first time to read that book, book must start from the first page.

c) If the book is finished and still exist in the bookshelf that book always opens at the last page.

[8]. Books shall be in .txt format.

a) Users only access books .txt versions. Other formats are not allowed in the system.

[9]. User shall remove a book anytime.

a) Removal operation requires adding book before.

b) Users do not need to finish the book to remove book.

c) One book can be removing at the same time. Multiple removal is not allowed.

[10]. Bookshelves must be private, so account owner only access his/her bookshelf.

a) Bookshelves shall be dedicated to account.

b) It can use multiple accounts on one device but not at the same time.

c) Same account can be open on many devices at the same time.

[11]. The system shall be easy to use

a) System interface must be spacious, and texts must be easy to read. As a readable text criterion, sans-serif fonts shall be used.

b) Font sizes must be responsive for devices.

c) Interface shall be basic which means that only essential functions placed on the screen.

d) Button/menu texts shall be clear and reflects its function.

e) System colors must not be arbitrary and overly colorful, there must be a light and clear color theme.

4. System Models

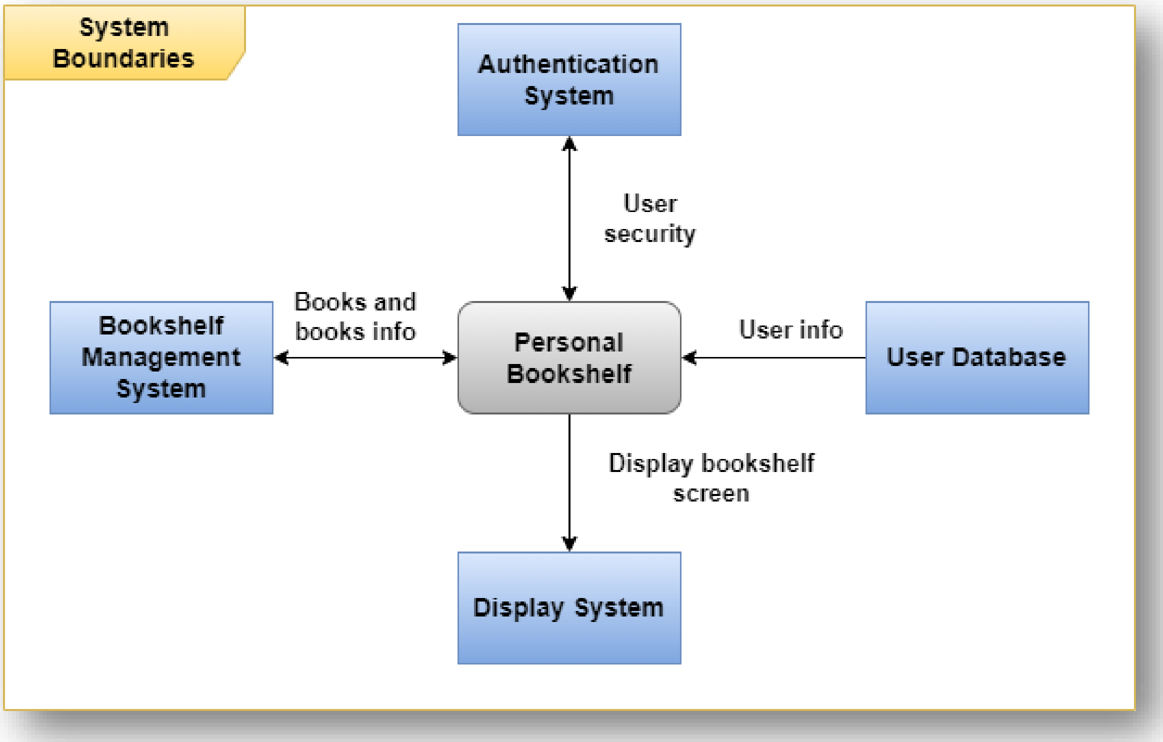

Figure ***Error! No text of specified style in document.****1 System Boundaries Model*

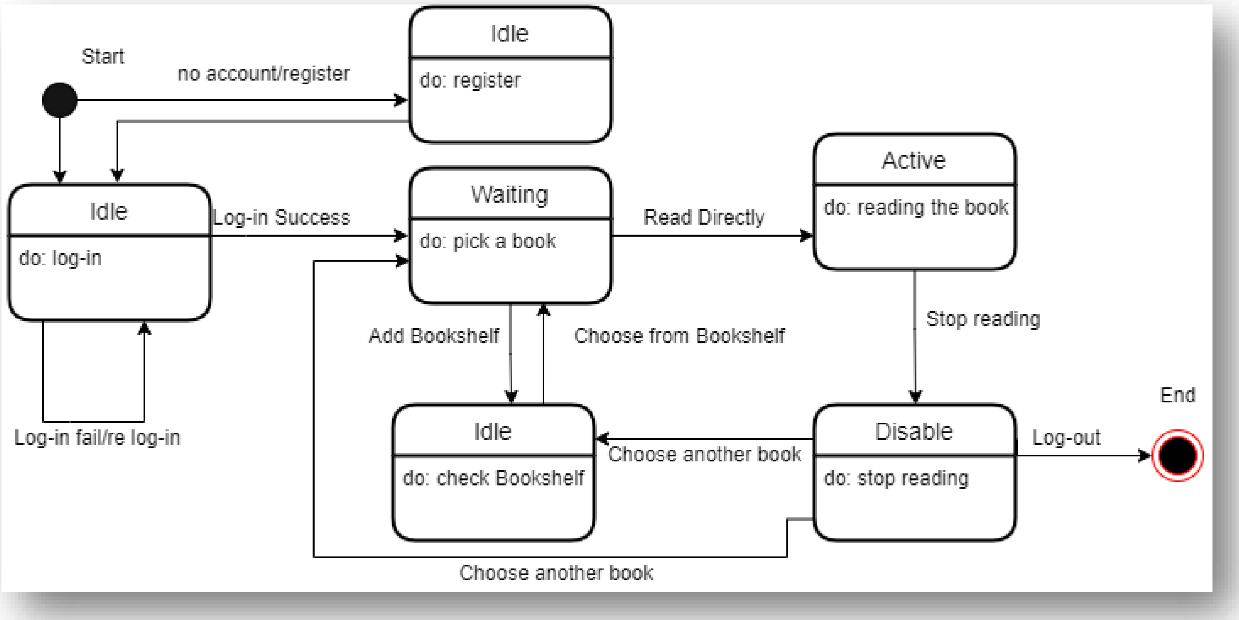

Figure ***Error! No text of specified style in document.****2 State Diagram*

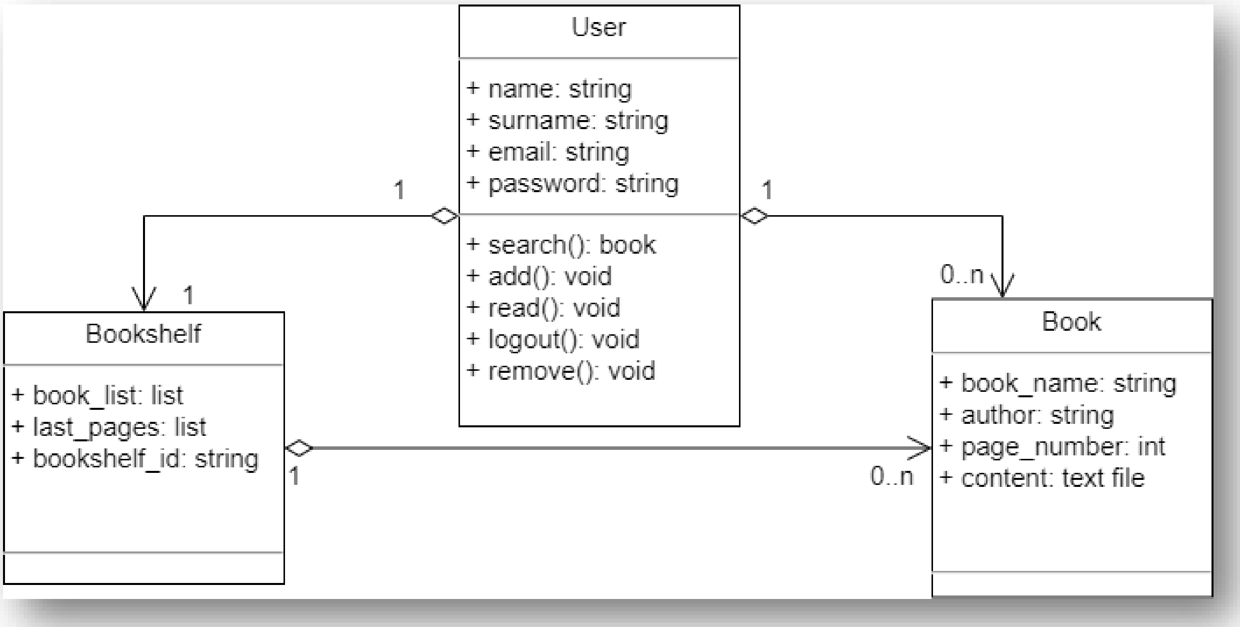

Figure 4.3 Class Diagram

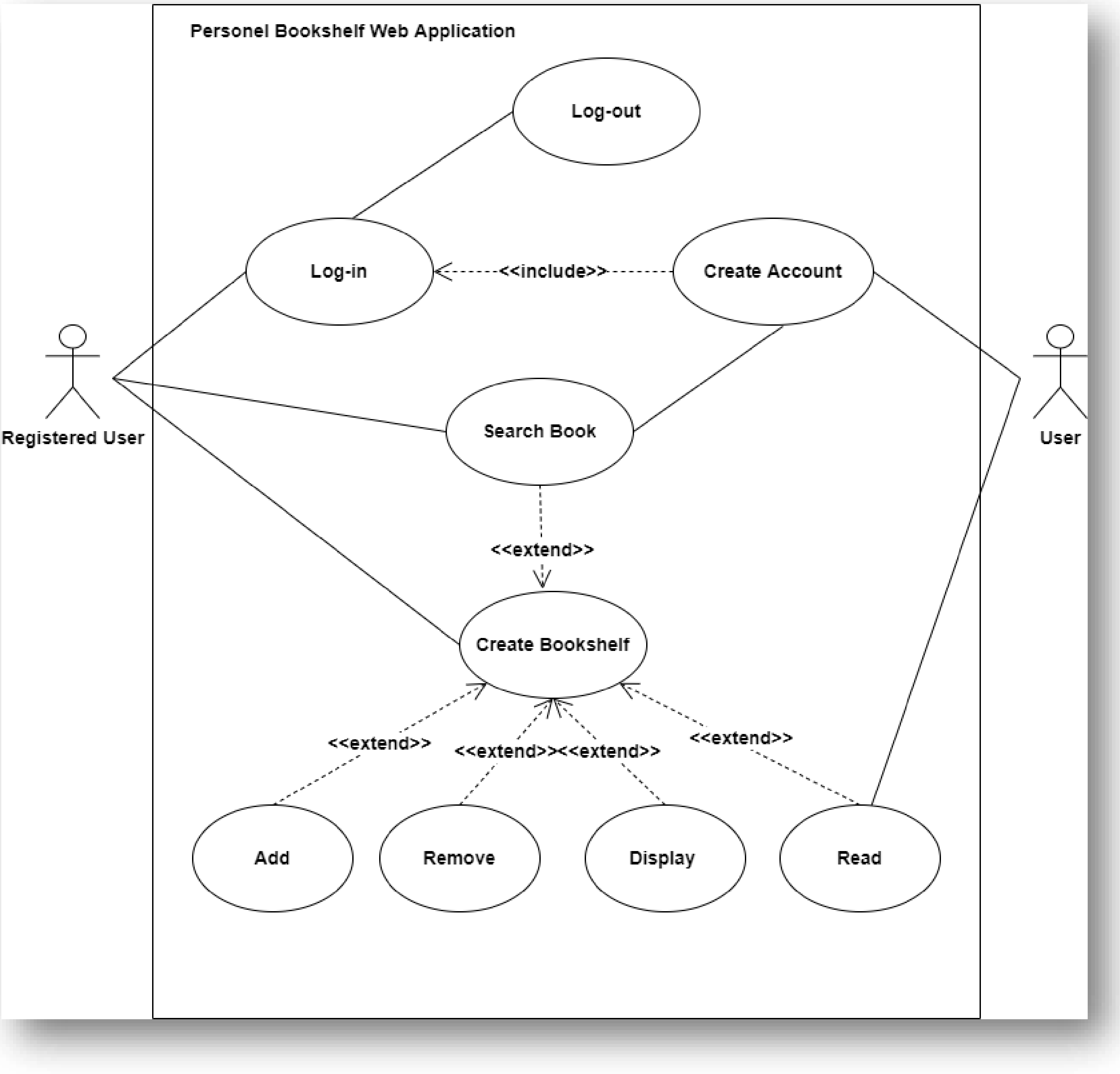

Figure 4.4 Use Case Diagram

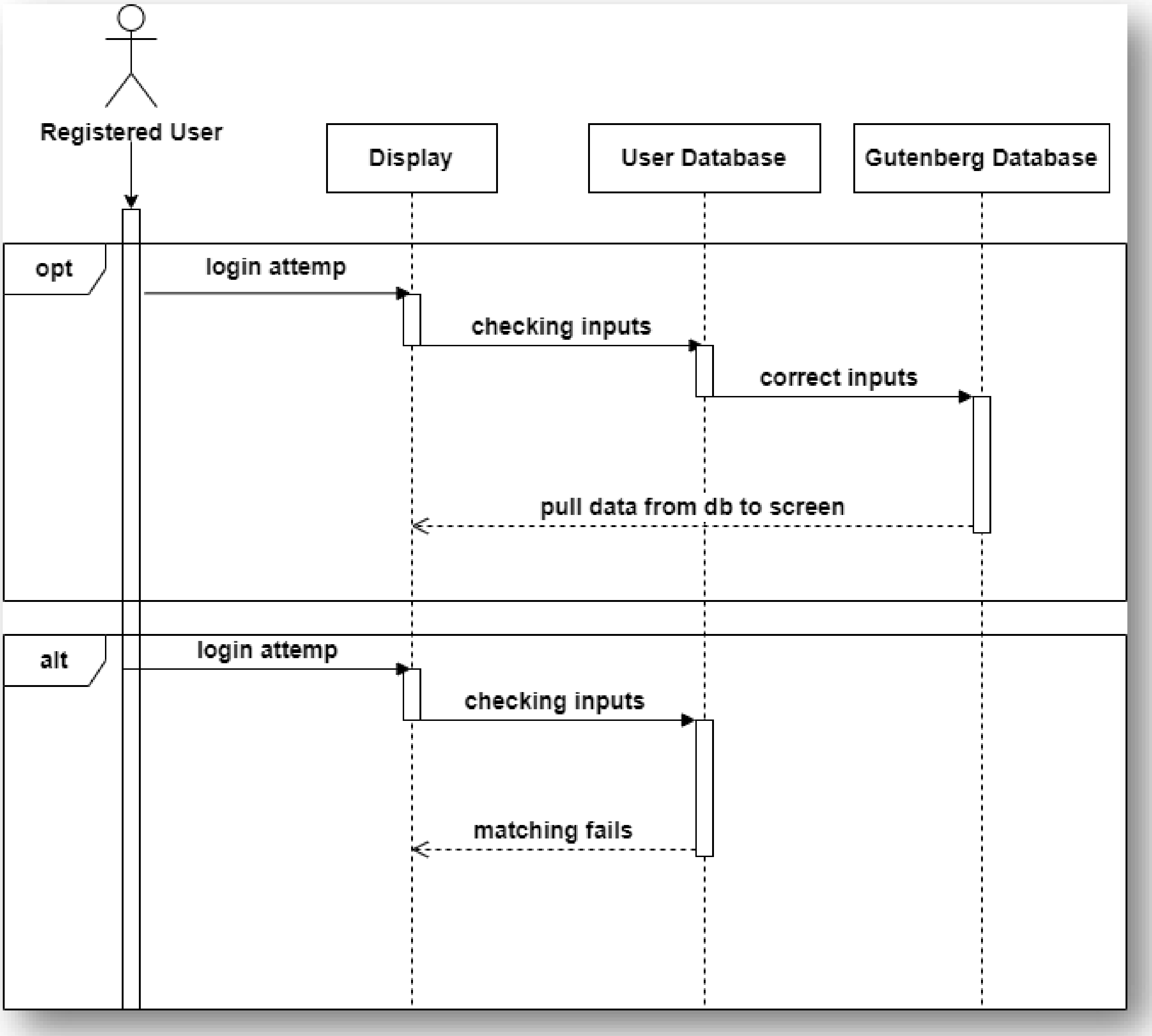

Figure 4.5 Sequence Diagram of Login Attempt

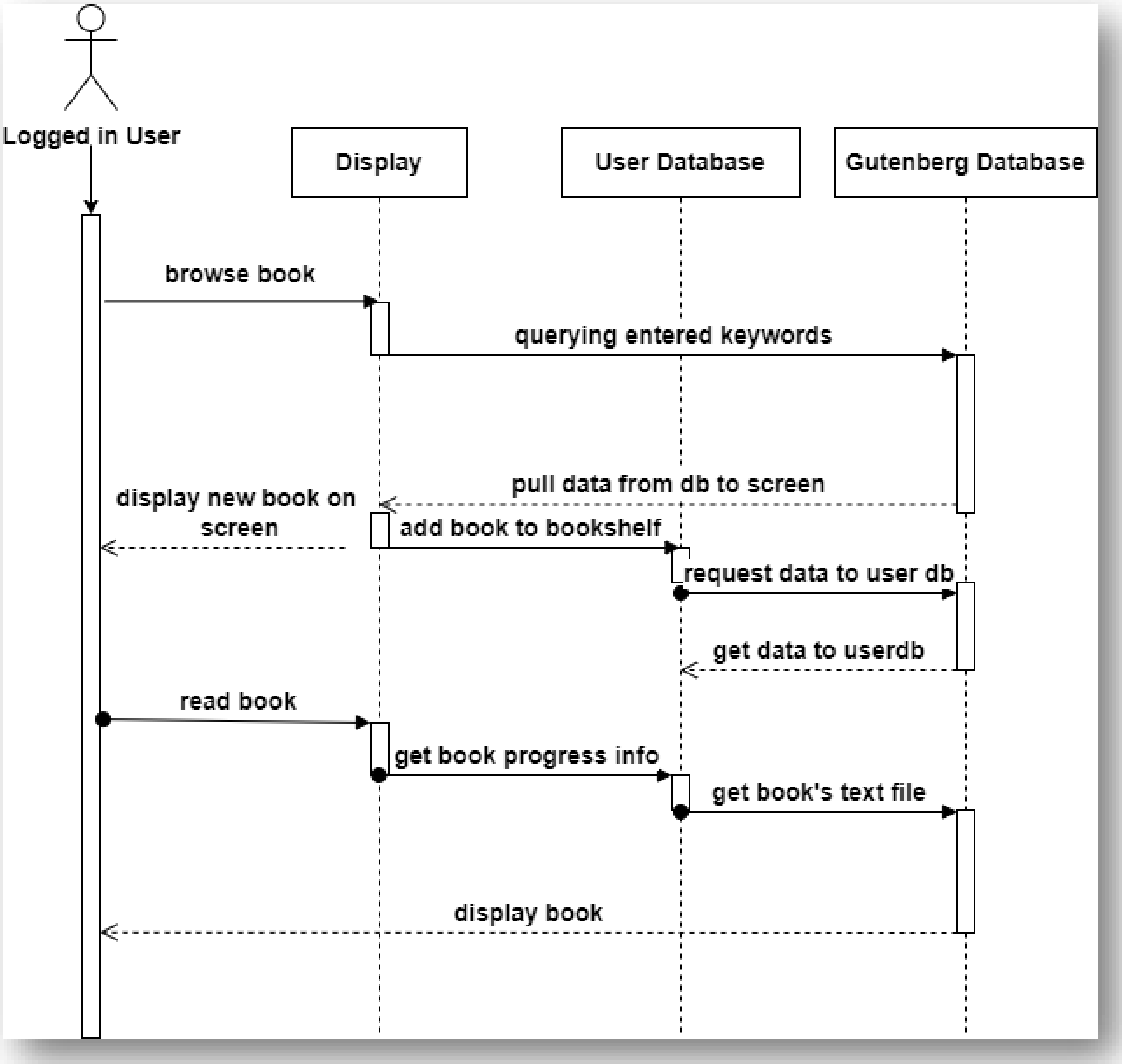

Figure 4.6 Sequence Diagram of Book Adding and Reading

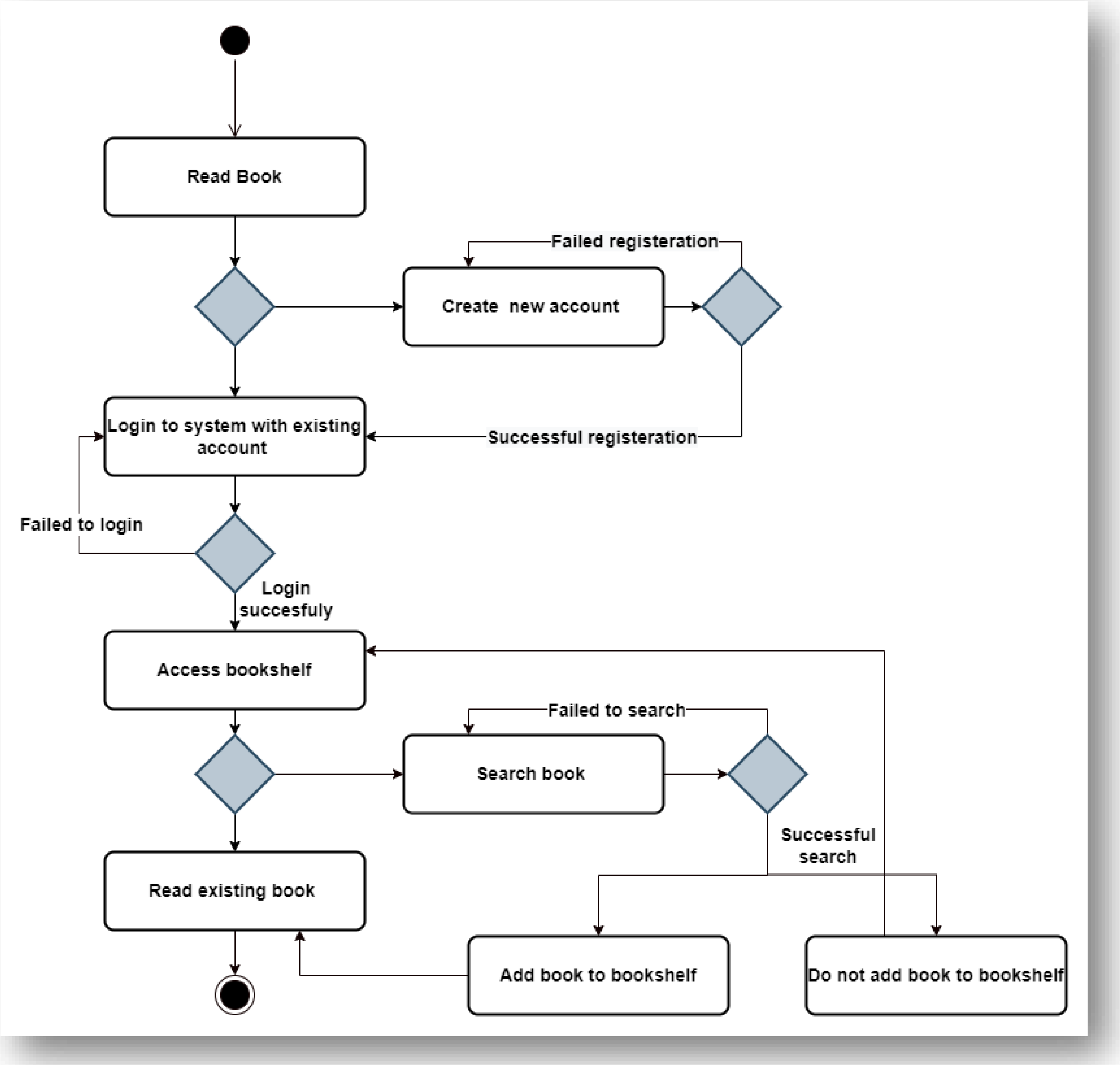

Figure 4.7 Activity Diagram

5. System Evolution

Audible Books

Users may want some changes about reading a book in Gutenberg Project. We can integrate an audible reading experience to our website. Users just need to pick a book and select audible read button and then start to read user's last checkpoint in that book, or user can choose to read from beginning. This improvement is so effective for users who like to listen much more than read or children who not able to read yet. During the audible read, you can stop or close that function, or you can continue with reading yourself. After the close, you will not lose any progress if you add that book in your Bookshelf.

Database Optimization

We may update and renovate our database parallelly with *Gutenberg Project*. If *Gutenberg Project* adds new books in his database, we need to add these books to our database as soon as possible, so we can offer much more efficient reading experience in our website. We will always be on the watch about these updates. Also, we can upgrade our user database if we approach more user than our database and server capacity we can upgrade and update our database and server services when there is a need.

Displaying Other Users and Adding Friends

In our web application you can create a profile which includes your special information and books which you like to read in your bookshelf. We can add some features which are similar to social media platforms. Users can add friends in our website, and then they can chat. They can see their friend's special information like name, surname, age, etc., and also they can see their friend's bookshelves and take book advice from there. Also, friends can communicate with each other and talk about books which they are interested in.

Appendices

Last view of KANBAN Board on Jira: https://ibb.co/dQDTSQJ