PEER TO PEER BOOK RENTAL SERVICE

A Project Report

Submitted

By

Gaurav Kumar Singh: 19BCE2119

Piyush Rajput: 19BCE0689

Ayush Gupta: 19BCE2422

Aryan Kumar: 19BCE2295

In

Software Engineering

Under the Guidance of

Prof. Akila Victor

As part of

Final Project Report for the Course CSE3001 (Software Engineering)

INDEX

Abstract

The project entitles "Peer-to-Peer Book Rental Portal" is a very effective, feasible online portal that facilitates the readers to read the books and magazines on rent reliably. If a reader wants to read a book, he/she has to purchase the book, which is not worth it for most users, especially for those who want to refer to only a part of the specific book, or does not want to keep the book after purchase or they can go to the library to lend a book where firstly they have to be a member and have to follow the library rules and regulations - like to return the book at the specified date that is given by the library. There is not much flexibility with the library management system and many places do not even have libraries and many of them are not very well maintained. So, a lot of time and money gets wasted on these. But with our application, the situation becomes very easy. The person which is having the book can give the book on rent for some days at a reasonable rent and can gain money and on the contrary, those who want to rent a book can communicate with the book giver through our portal. So, in general, our online book rental system will allow a user to rent a book online from the website. Once a book is rented it is no longer available to rent again by other users until the book is returned. The book which has already been rented will show the name of the renter and the date until when it is rented for the main objective of the project is to create an online book rental site that allows users to search and rent a book online based on title, author and subject. Using this Website, the user can rent a book online instead of going out to a book store and library spending a lot of their time meanwhile also ensuring giving them the opportunity to passively earn money in exchange for their books.

Aim

If a reader wants to read a book, he/she has to purchase the book, where they have to give much money or they can go to the library to lend a book where firstly they have to be a member and have to follow the library rules and regulations - like to return the book at the specified date that is given by the library. So, a lot of time and money gets wasted on these. With our portal situation becomes very easy. The person which is having the book can give the book on rent for some days at a reasonable rent and can gain money and on the contrary, those who want to rent a book can communicate with the book giver through our portal. So, in general, our online book rental system will allow a user to rent a book online from the website.

Objective

To create an application which satisfies all basic requirements of a decentralised peer to peer book rental service.

Scope

The Book Renting Service that is developed provides the users with renting books, borrowing books and many other facilities. The Book Renting Service is supposed to have the following features:

- The system provides a Sign-In and Sign-Up feature to the users.
- The system provides users to an option to check their profile page.
- The system provides a rental section which is composed of two components, the book rental system, and the history of previously rented books.
- The system provides a borrower section, the user can send a borrow request to the vendor of the book and browse through all the books which are up for borrowing.
- The system provides a return section where users can see the list of all the books, he/she had ever borrowed from the portal along with the option to return the books he currently has not returned to its owner.
- The system provides the user to manually sign out.
-

The scope of the project is localised to a small area up to the extent to which travelling for a book is feasible for the users. This project won't focus on online payment gateways but rather on the functional flow of the application. The database used (Mongo dB) can handle a lot of requests so the scope of application is not constrained by database used, however the use of synchronous functions for the most part can lead to some server performance invoked internal errors although except request time out error, most of the internal errors are handled manually.

Introduction

The main leverage of this project is the fact that many people like collecting books, including ourselves whereas many people don't have access to books for cheap, or do not particularly want to read the entire book but only a segment of that particular book. Also, many people like to read stuff in hardcopy format instead of softcopies. These two groups of people can massively benefit each other through the use of an application such as ours. Borrowing and renting such books makes a lot of sense to us as in most cases, after we read a particular book, we don't refer to most books again very often and they just accommodate with not a lot of use. Through our application, people can make their book available for rent for other people to borrow for some time which gives the renter a passive income source and the borrower a cheap alternative to read from a book instead of buying a new book every time. It also helps to create a community of people having similar interests.

The convenience of an online book rental is a major consideration for most of us, especially those living in areas where access to a large bookstore is limited or is quite far away.

Many books are too much expensive and for these types of books, this system is most preferable. If a book is too expensive, then one can rent that Book for some amount and can read that book by spending a very small amount of money.

After reading the whole book they can return a book and another one can take Benefit of this type of expensive books.

Using our website both renter and borrower will be benefitted.

Problem Statement

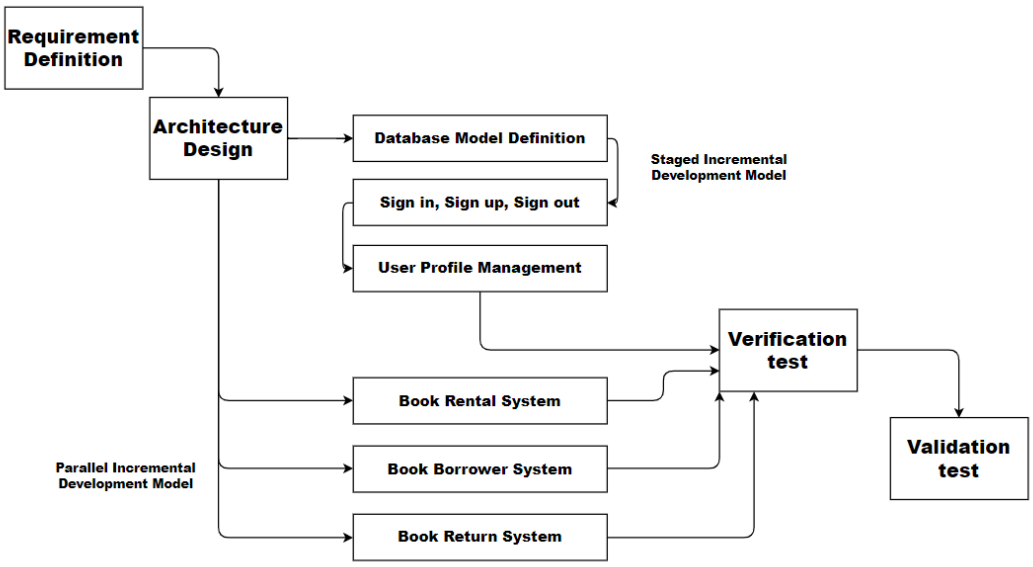
If a reader wishes to read a book, he or she can either buy it or borrow it from the library, where they must first become a member and adhere to the library's rules and regulations, such as returning the book before the due date. As a result, a significant amount of time and money is spent on them.

Process Model

Incremental Model (Parallel and Staged Development Model Hybrid)

We have picked the incremental process model for our project because our project is modular in the sense that each module of our project can be developed independently with only the routes to be established to integrate a module into our project. The entries in the database for our project are interconnected to each other to facilitate the reusability of models amongst the modules. Our project in the simplest case can be divided into 6 modules

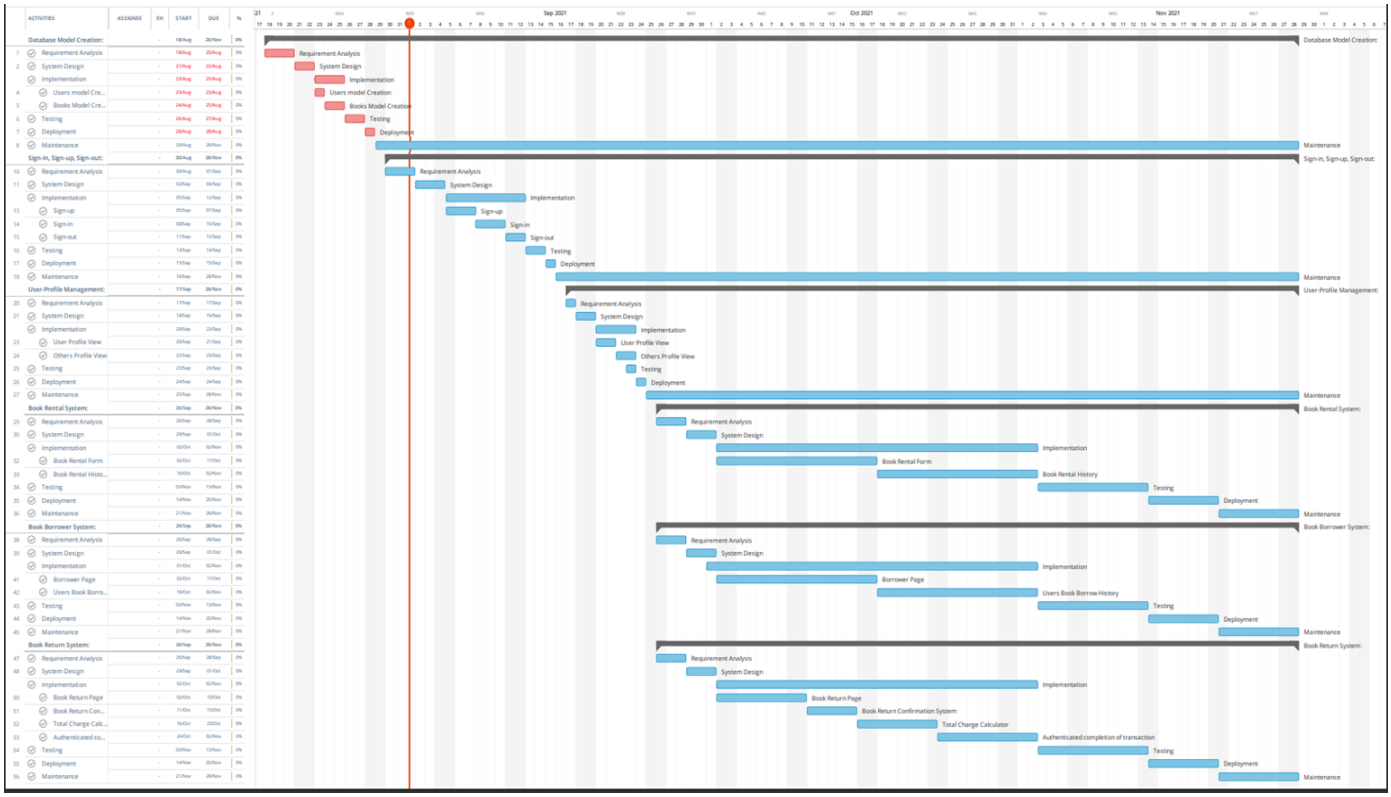
- Database model creation
- Sign-in, Sign-up, Sign-out
- User Profile management
- Book rental system
- Book borrower system
- Book return system



Project Plan

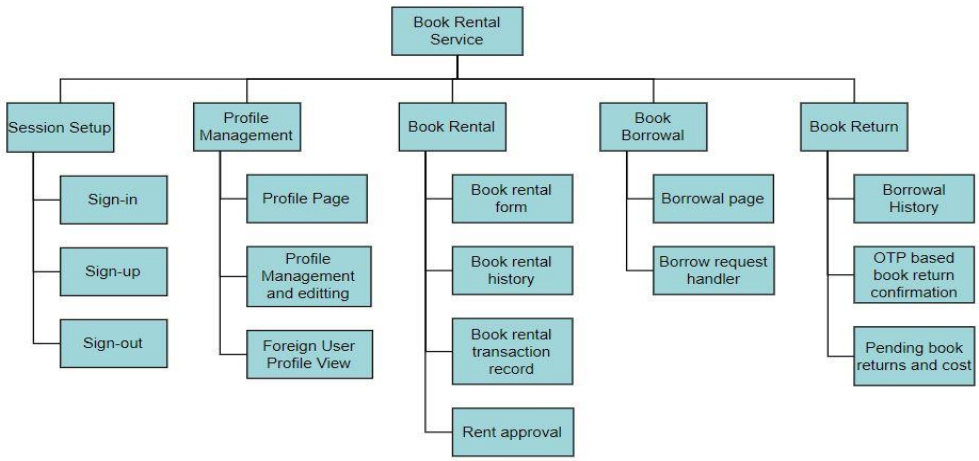
Project Module	Module Stage	Initiation Date (2021)	Completion Date (2021)
Database Model Creation	Requirement Analysis	18th August	20th August
	System Design	21st August	22nd August
	Implementation	23rd August	25th August
	Testing	26th August	27th August
	Deployment	28th August	28th August
	Maintenance	29th August	--- --- --- --- --- ---
Sign-in, Sign-up, Sign-out	Requirement Analysis	30th August	1st September
	System Design	2nd September	4th September
	Implementation	5th September	12th September
	Testing	13th September	14th September
	Deployment	15th September	15th September
	Maintenance	16th September	--- --- --- --- --- ---
User-Profile Management	Requirement Analysis	17th September	17th September
	System Design	18th September	18th September
	Implementation	19th September	22nd September
	Testing	23rd September	23rd September
	Deployment	24th September	24th September
	Maintenance	25th September	--- --- --- --- --- ---
Book Rental System	Requirement Analysis	26th September	28th September
	System Design	29th September	1st October

	Implementation	**2nd October**	**2nd November**
	Testing	**3rd November**	**13th November**
	Deployment	**14th November**	**20th November**
	Maintenance	**21th November**	**--- --- --- --- --- ---**
Book Borrower System	**Requirement Analysis**	**26th September**	**28th September**
	System Design	**29th September**	**1st October**
	Implementation	**2nd October**	**2nd November**
	Testing	**3rd November**	**13th November**
	Deployment	**14th November**	**20th November**
	Maintenance	**21th November**	**--- --- --- --- --- ---**
Book Return System	**Requirement Analysis**	**26th September**	**28th September**
	System Design	**29th September**	**1st October**
	Implementation	**2nd October**	**2nd November**
	Testing	**3rd November**	**13th November**
	Deployment	**14th November**	**20th November**
	Maintenance	**21th November**	**--- --- --- --- --- ---**

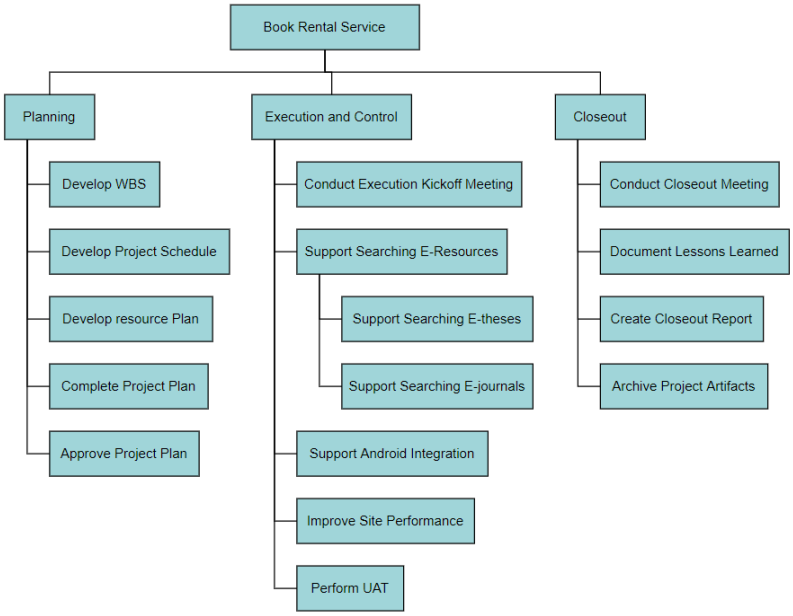


Work Breakdown Structure

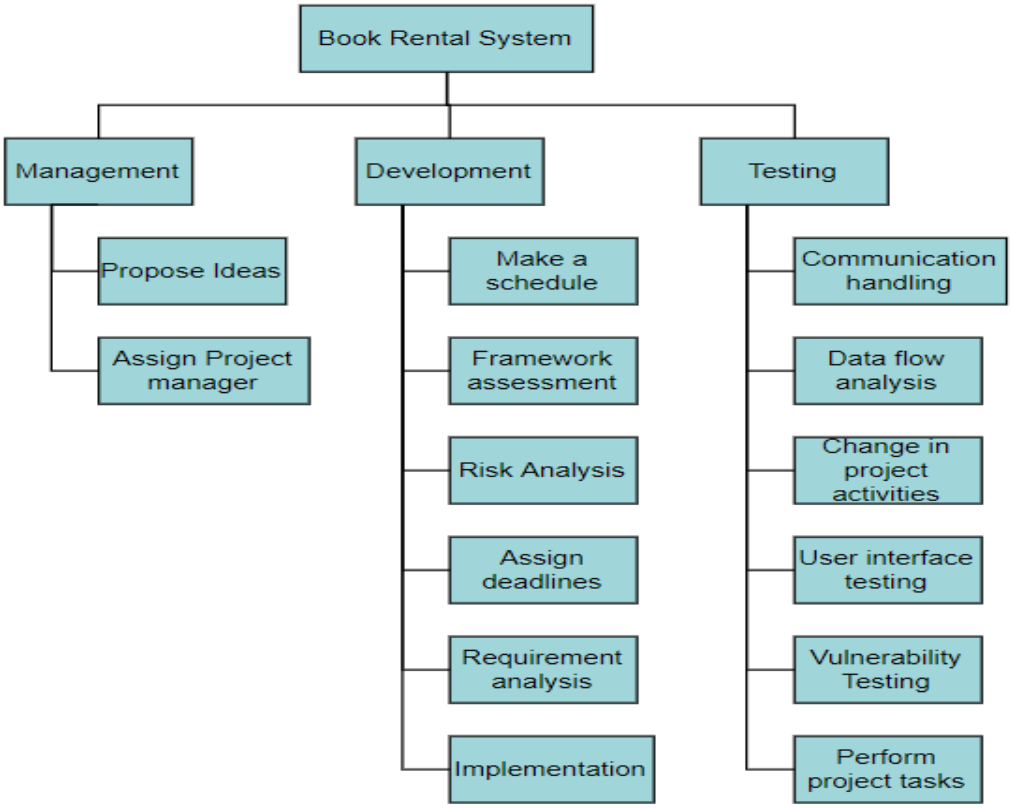
Deliverable Based WBS



Phase Based WBS



Responsibility Based WBS



Requirements

Functional

- **Session Management:** This includes **Sign-in, Sign-out and Sign-up** features for user's session creation and management. It is a high priority task because it ensures the security of the users.
- **Profile Management:** This feature enables logged-in users to view and modify their profile personal information. The users can also visit other users' profile to see their public details but can't modify other's details. Profile page is the first page the user is redirected to after successfully signing in. It helps the users to keep track of who borrowed or rented the book and can help leverage reliability of certain vendors and borrowers.
- **Book Exchange Management:** The feature enables logged-in users to **rent, borrow or return** books.
- If the user chooses the rent a book, there will be a form in which the details of the book which the user wants to give away for rent will be submitted and the book will be made available in the borrow section of the application.
- If the users choose to borrow a book, the user can send a borrow request to the vendor of the book and browse through all the books which are up for borrowing.
- If the user wants to return the books (which the user borrowed), user users can see the list of all the books he/she had ever borrowed from the portal along with the option to return the books he currently has not returned to its owner.

Non-Functional

- **Performance Requirements:** User Sessions must be maintained to reduce inconvenience of signing in every time they visit the website for the user to maintain a proper work flow for active users.
- **Safety Requirements:** Transactions must be safely completed to reduce risk of frauds. The session information should be properly encrypted to safeguard the identity of the user. The cookies must be set to expire periodically too to ensure proper authentication of the user periodically.
- **Software Quality Attributes:** The application must be robust and assume that wrong entries can be entered by user and the application therefore should not crash when inappropriate details were entered instead of letting them pass through. The data in database should be maintained and cleaned.
- **Business Rules:** Admins can manipulate transaction in dire situations to avoid well thought out frauds.

Confirmation from the client side will still be required to make changes to user's database.

Admins also have control to block certain users if they violate a lot of policies of the application or add nsfw content.

Design

Prototype Design

On the home page, the user has to login into their account. If they do not have an account then the user has to sign up to the portal. After signing up, the user is redirected to the sign-in page where the user credentials are verified. If the credentials are correct, the website creates a session cookie to keep the user signed in and prevents access to sign-in and sign-up pages to prevent malicious users from using their generated cookie to access other's accounts. Also, a middleware is configured to check if the user is trying to bypass login to access restricted pages and if they are; the user is redirected to the login page to log in with their credentials and session cookie. After signing in, the user can see his/her profile. Users can update their profiles here. From here user gets an option to perform one of many activities. Users can choose to make a book available for rent, borrow books, return the borrowed books, or either sign out.

If the user chooses to check the rental system, he will be redirected to another page that is composed of two components, the book rental system, and the history of previously rented books. In the book rental system, there will be a form in which details of the book which the user wants to give away for rent will be submitted and the book will be made available in the borrow section of the application. The history section of the page is mainly used for two tasks. Firstly, to show details of all transactions. Secondly, if any other user sends a request to borrow the book, the renter gets to choose if he wants to lend the book to that particular user or not. The book transaction gets through the initiation state only after the borrower is validated by the renter.

On the borrower section, the user can send a borrow request to the vendor of the book and browse through all the books which are up for borrowing. The restriction is that the user cannot borrow the book which he made available to borrow.

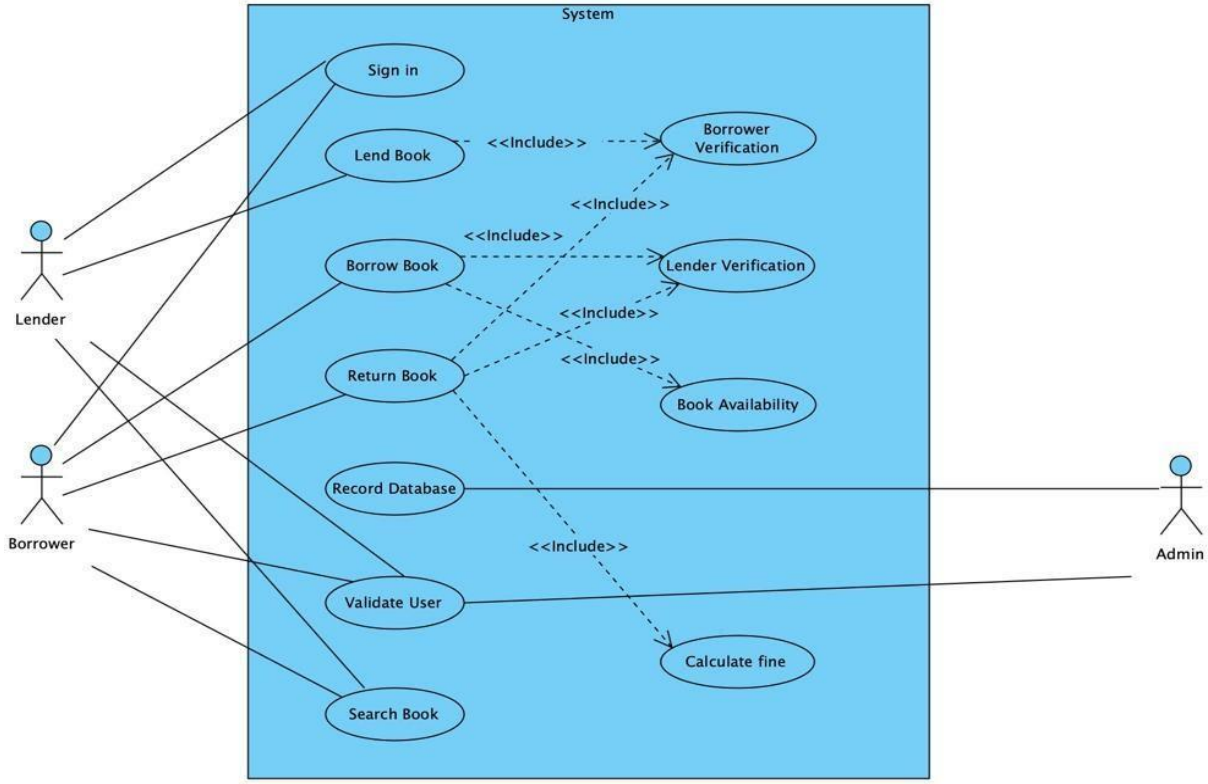
On the book return section, users can see the list of all the books he/she had ever borrowed from the portal along with the option to return the books he currently has not returned to its owner. For the transaction to be safely finalized, we will be implementing an OTP-like system. Whenever a book is made available for rent, a random number is also generated as an OTP which only the renter has access to. For the borrower to be able to complete the transaction, he has to enter that OTP for confirmation which the vendor will tell him/her. If the OTPs don't match, the

transaction won't close. If the OTPs match, it can be deduced that both the borrower and vendor approve the completion of a transaction and therefore can be terminated.

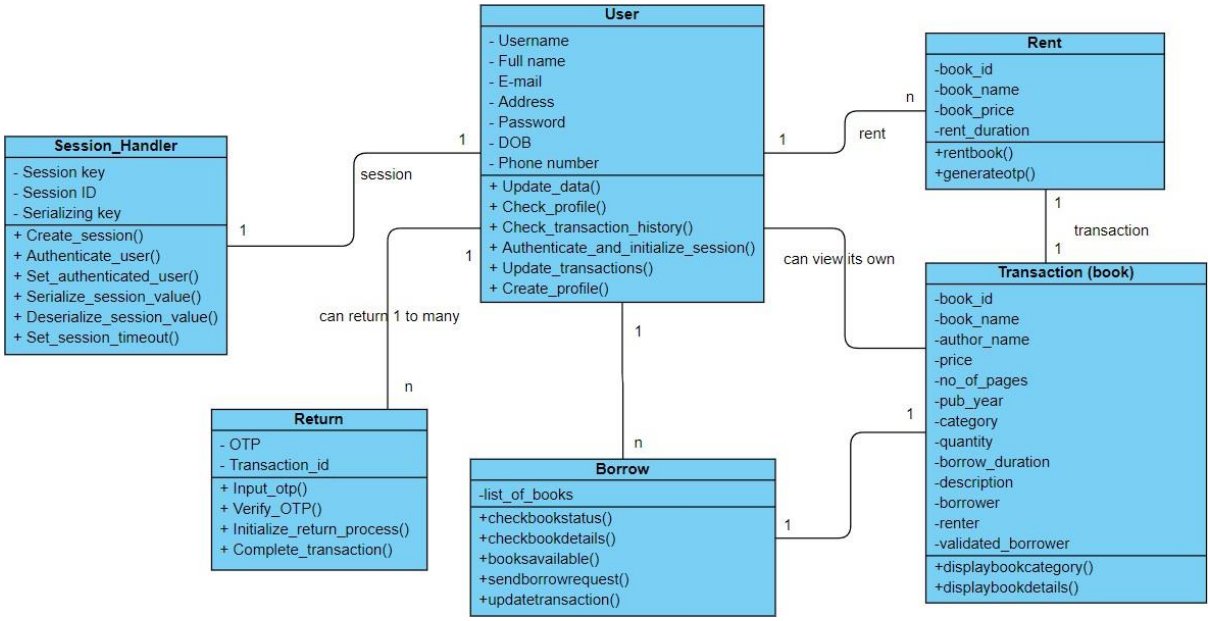
The user can also manually sign out which expires the session cookie, otherwise, the session cookie will automatically expire after some time, making the user log in again for security reasons.

UML Diagrams

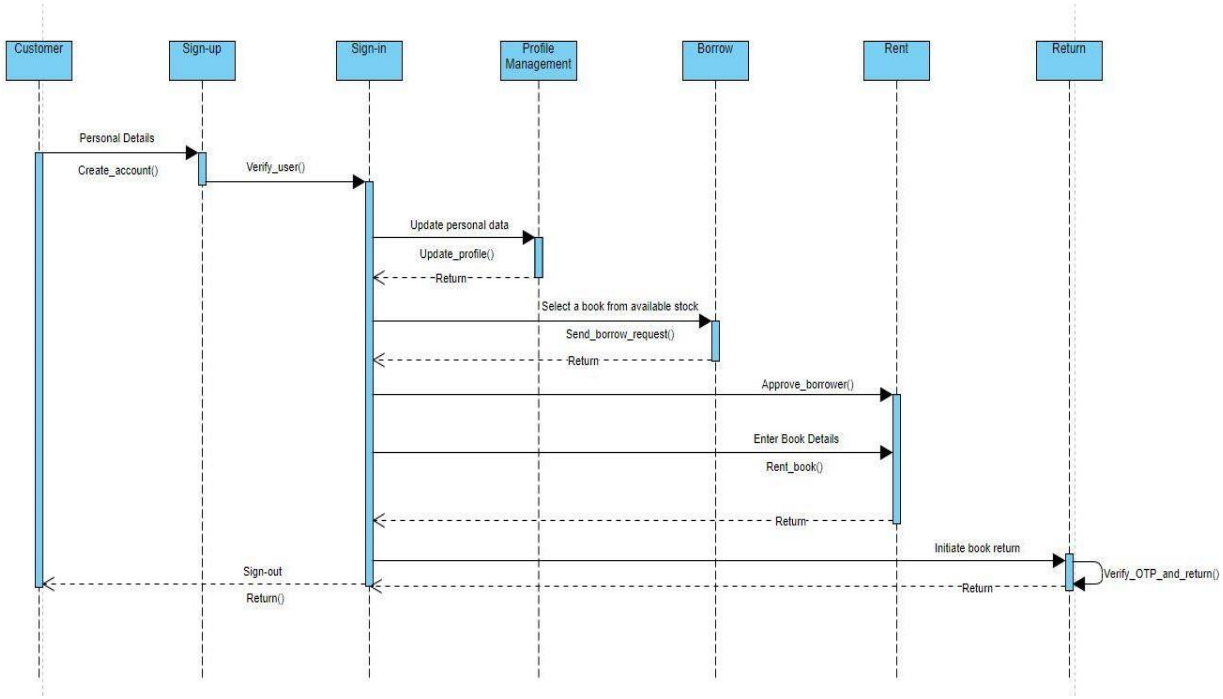
Use-Case Diagram



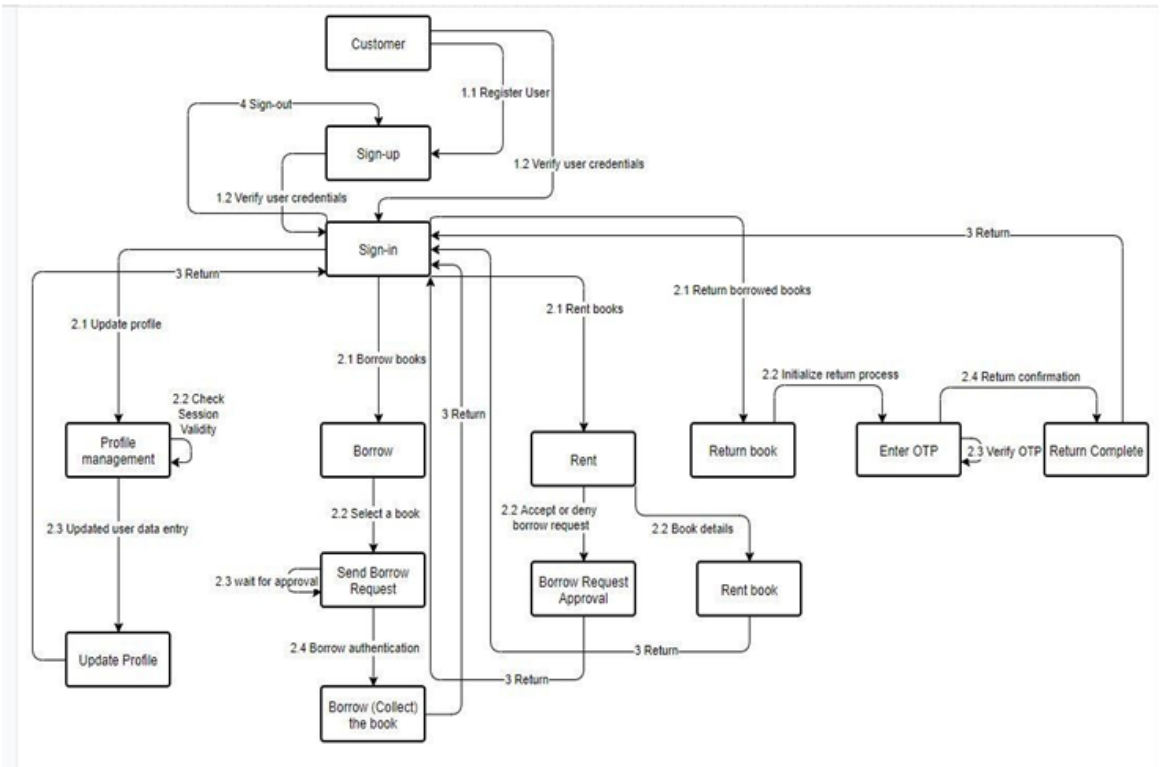
Class Diagram



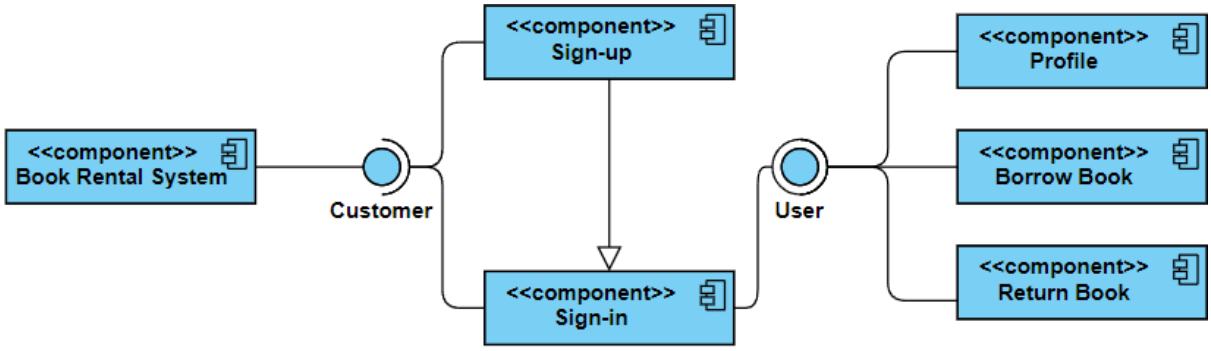
Sequence Diagram



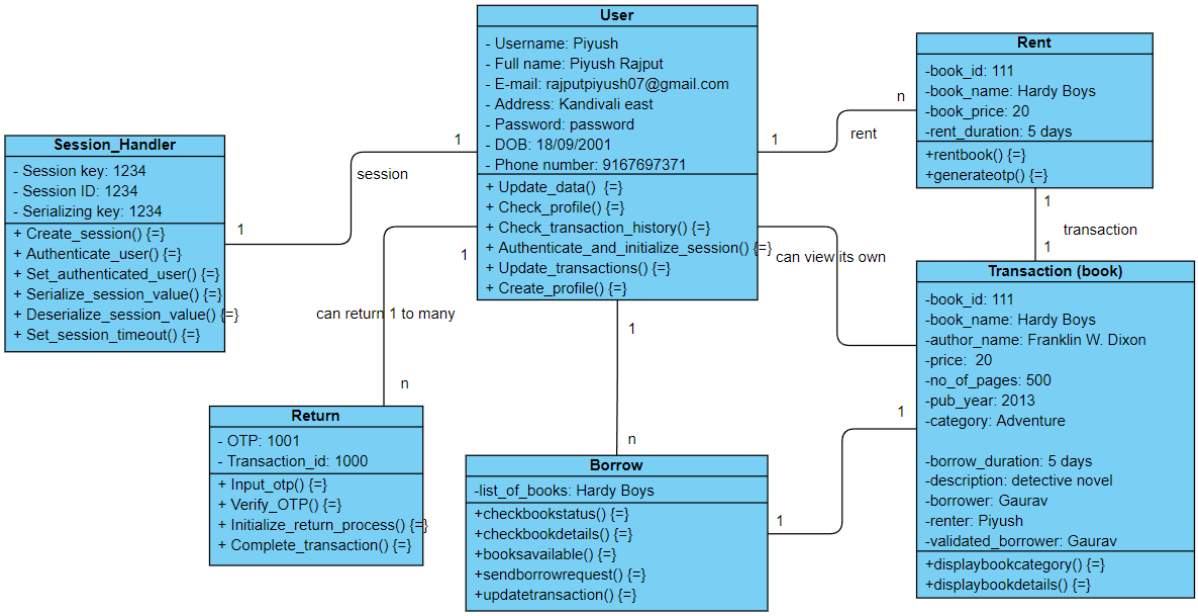
Collaboration Diagram



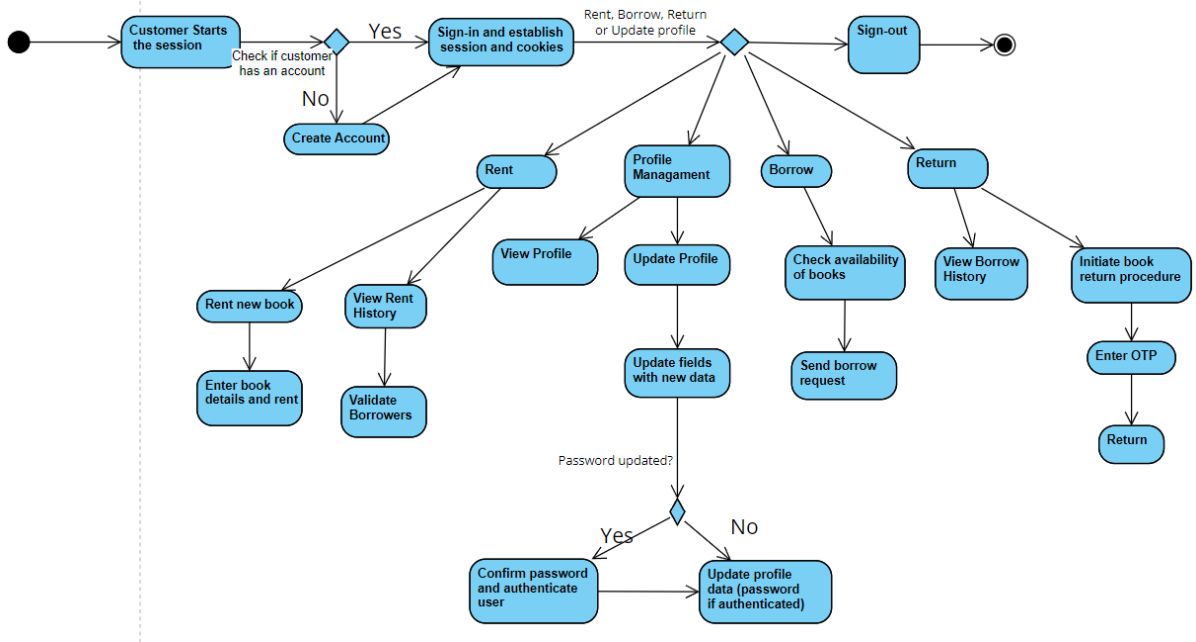
Component Diagram



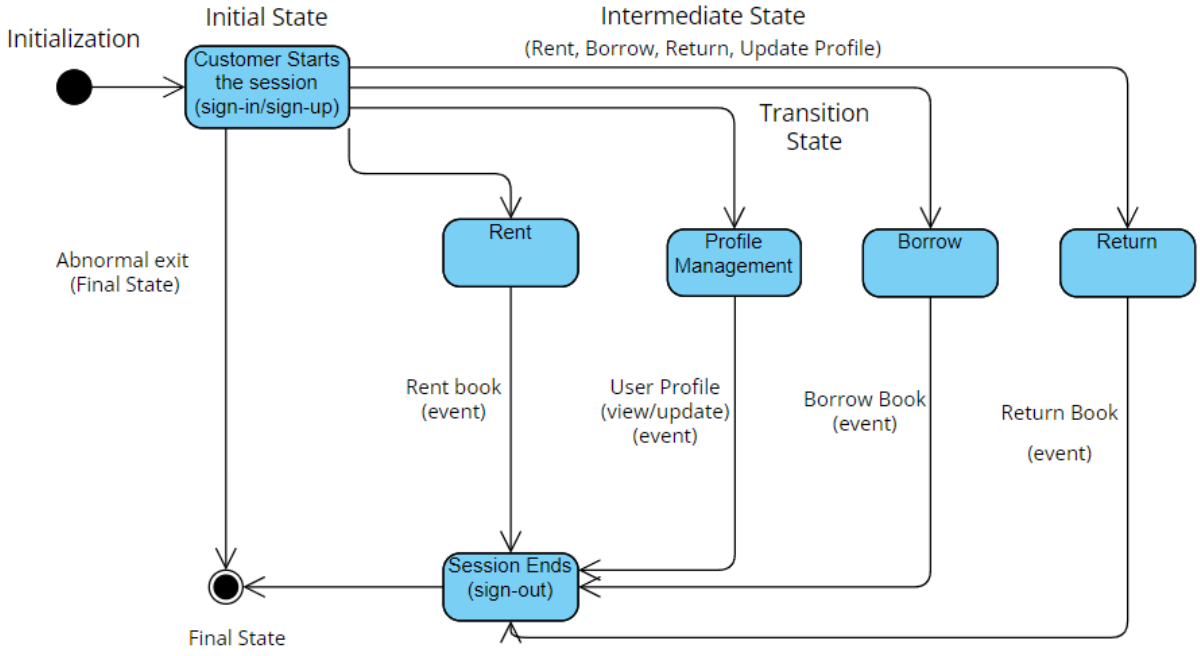
Object Diagram



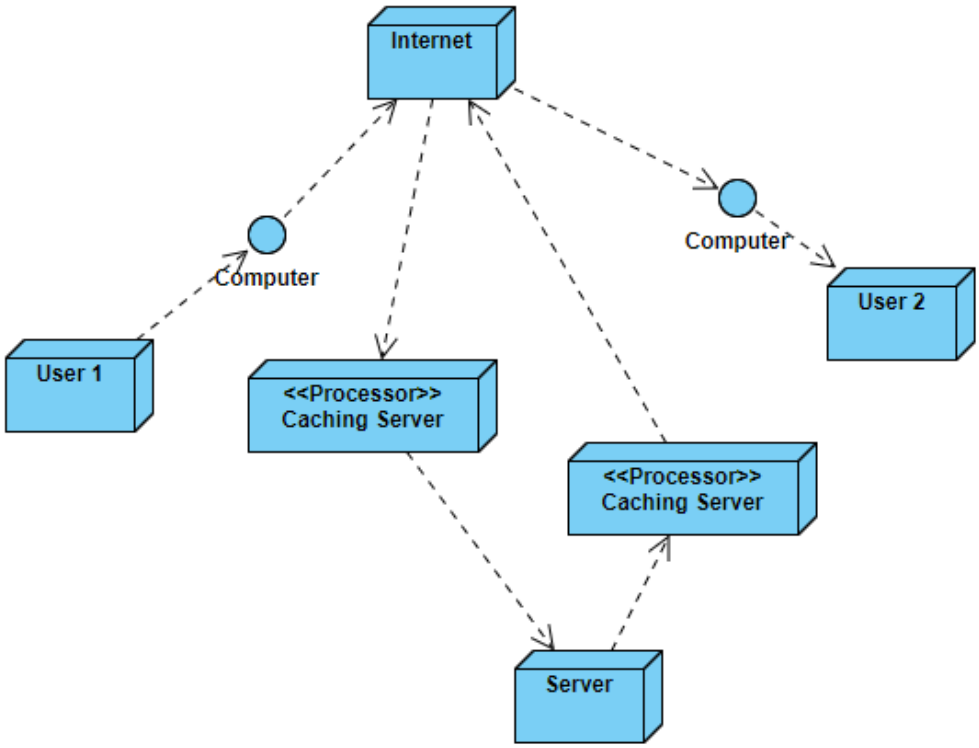
Activity Diagram



State Diagram

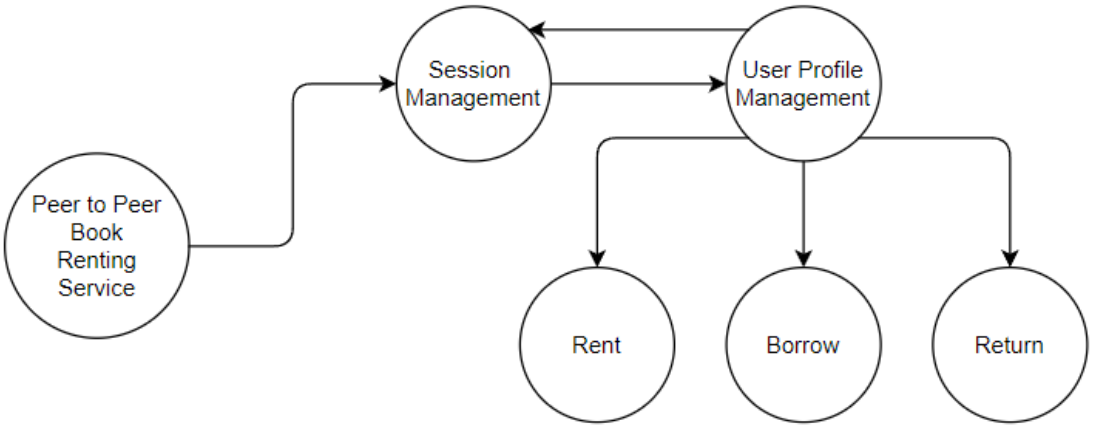


Deployment Diagram



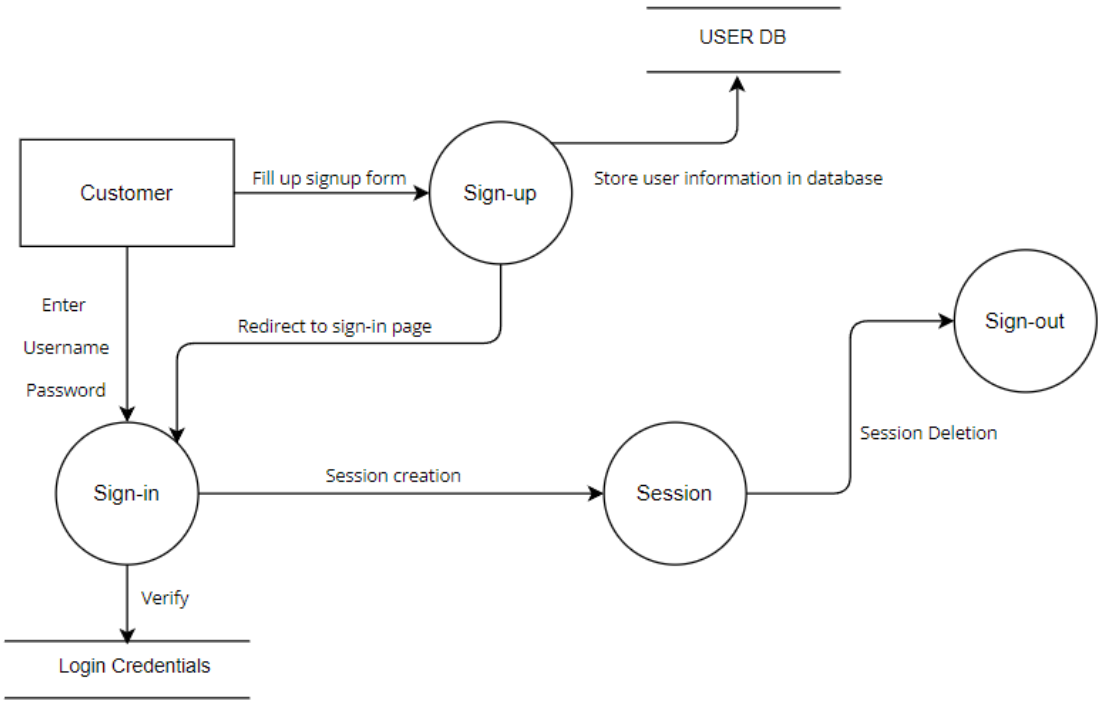
Data Flow Diagrams

Level 0

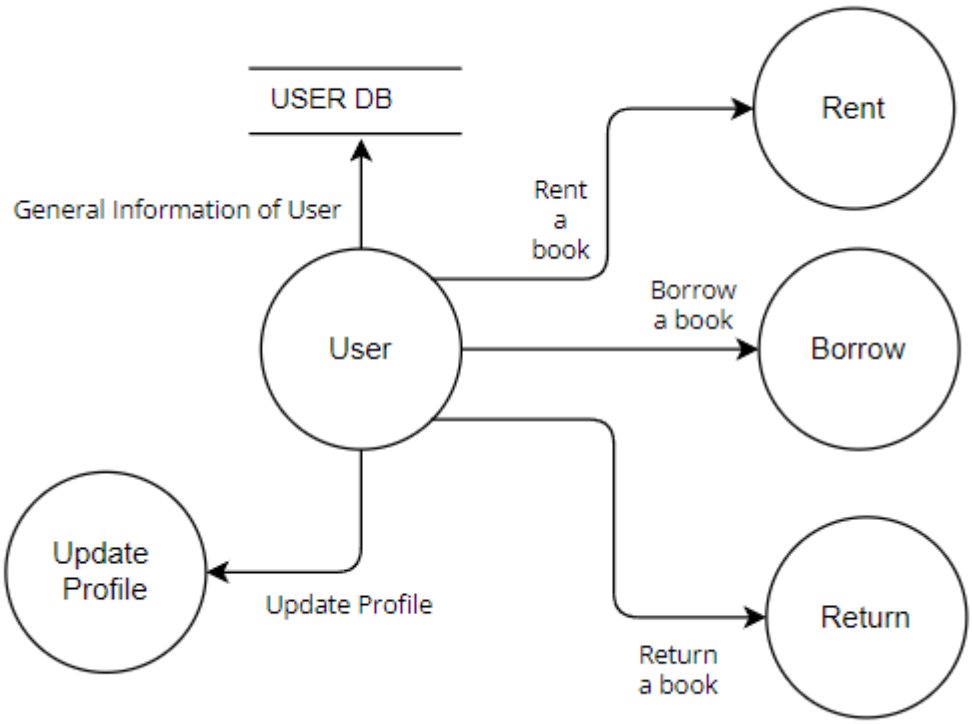


Level 2

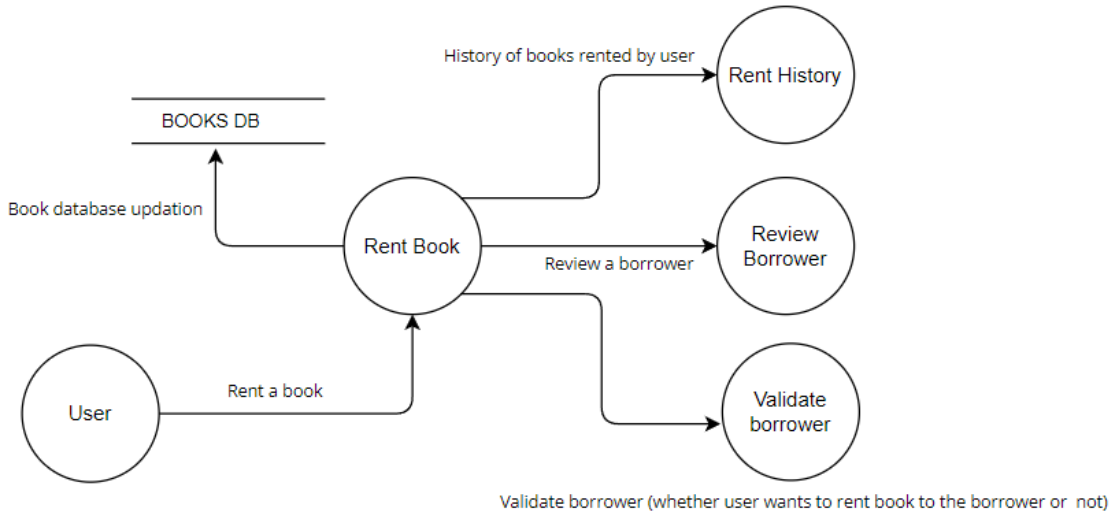
Session Management



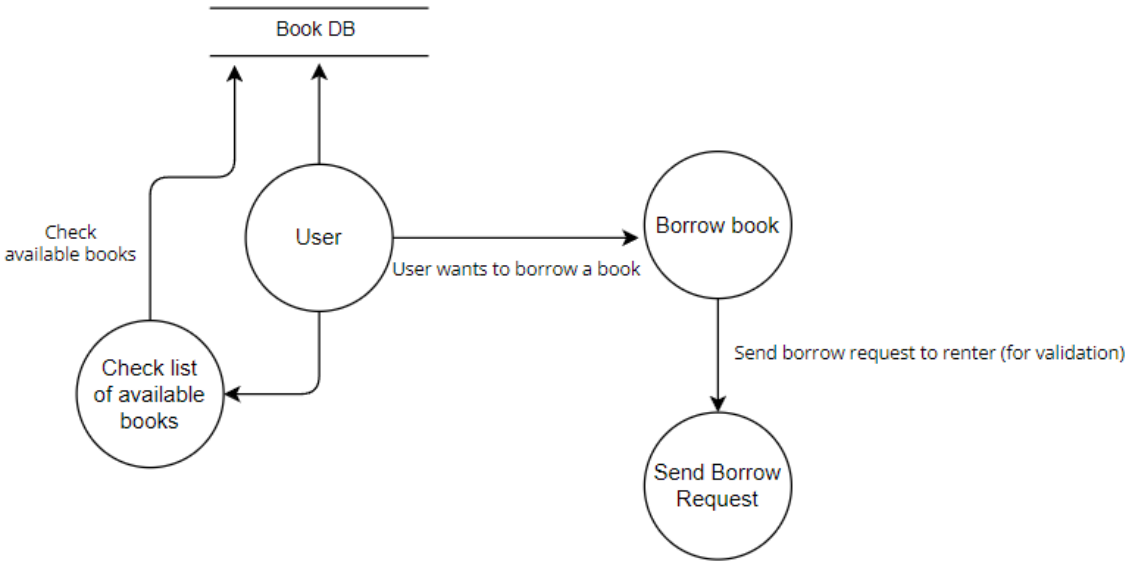
User Management



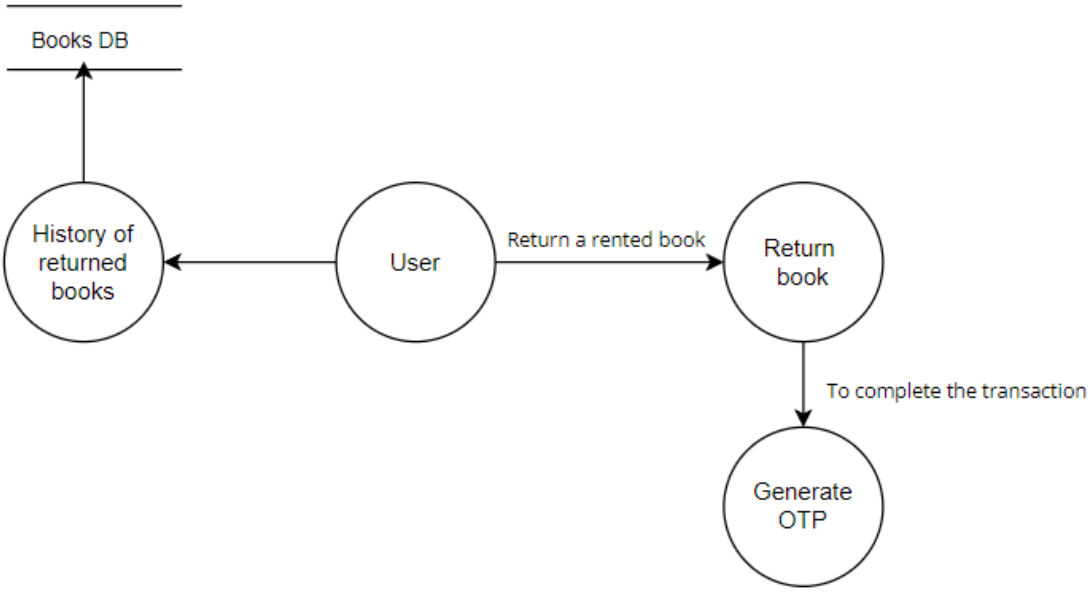
Rent



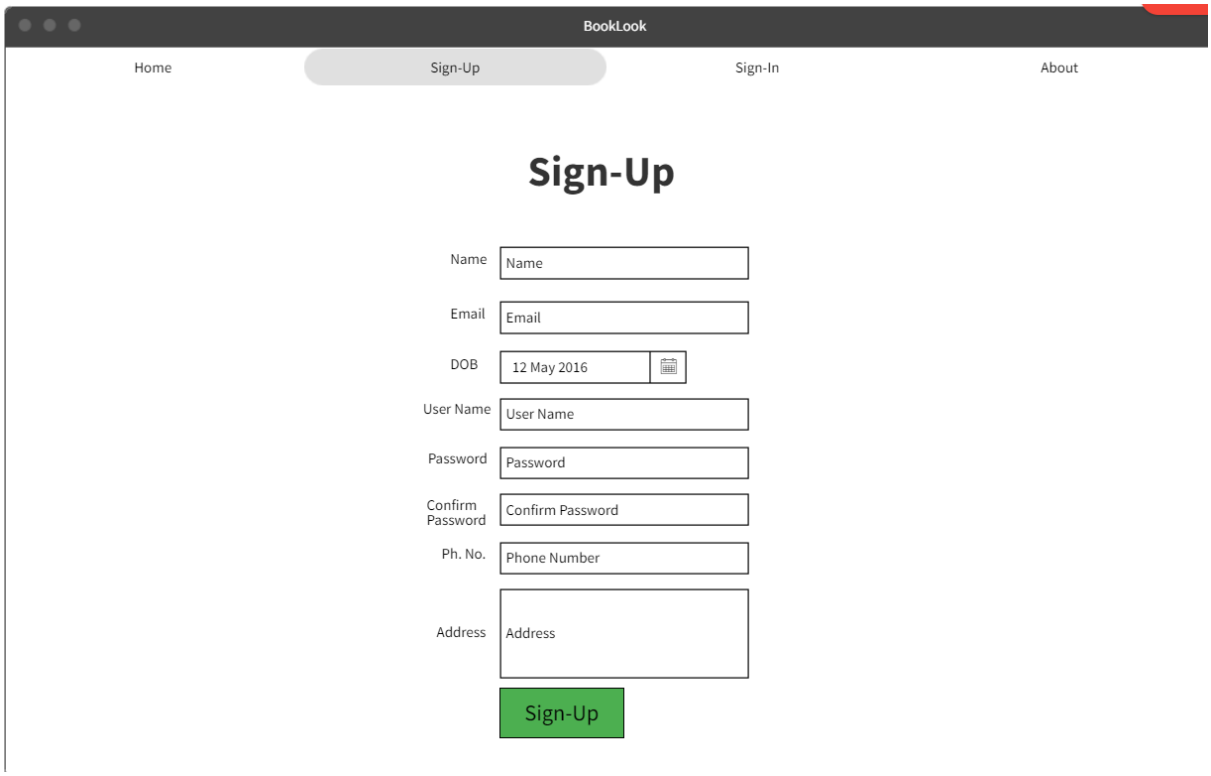
Borrow

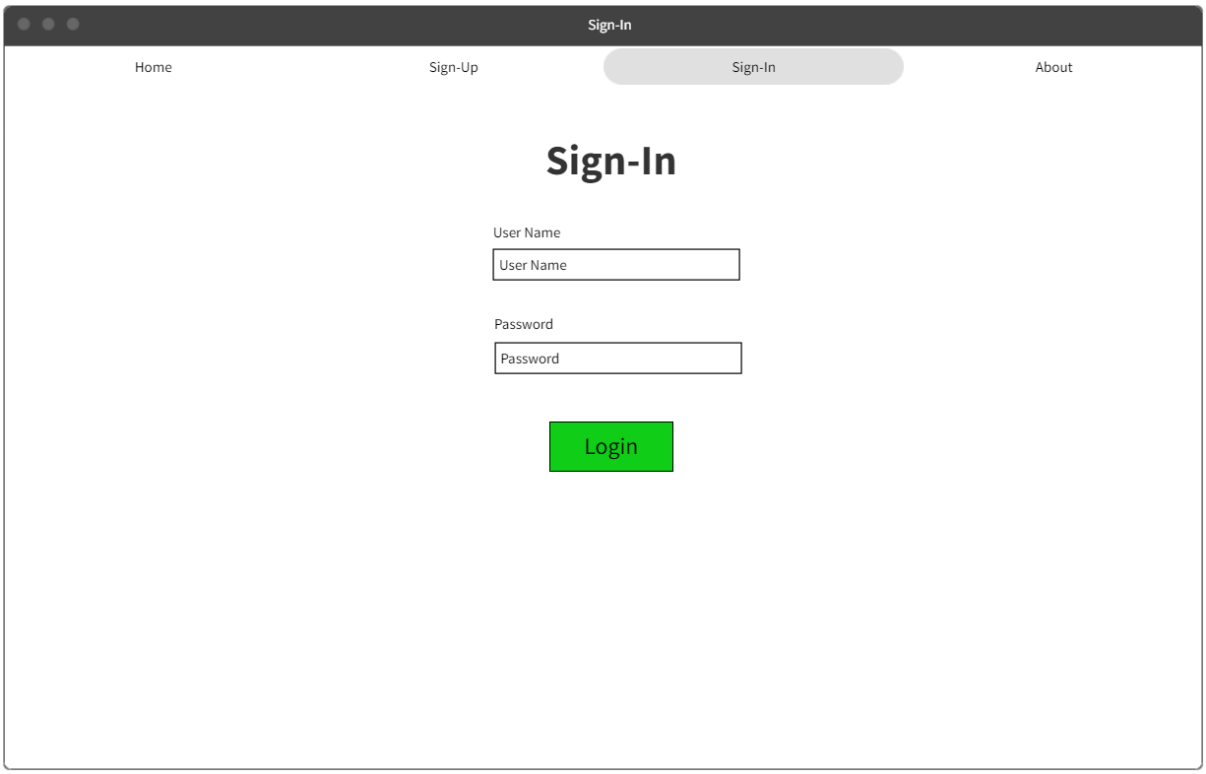


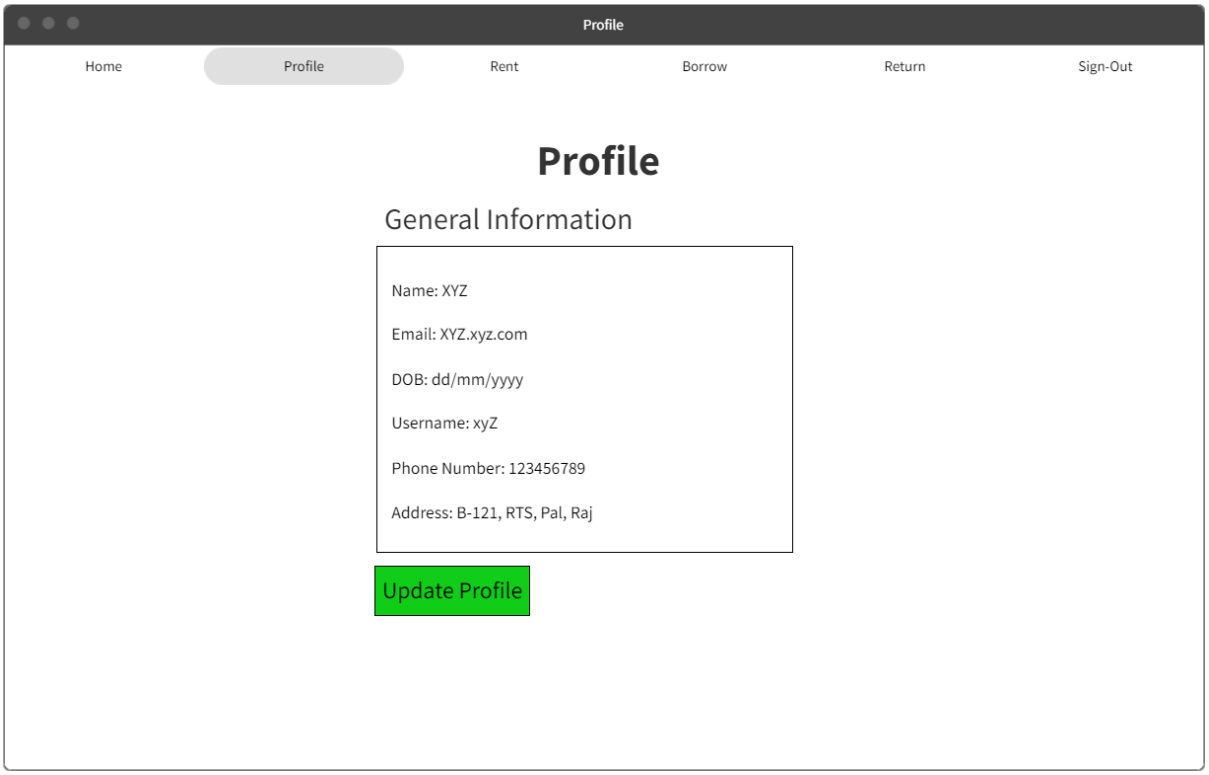
Return

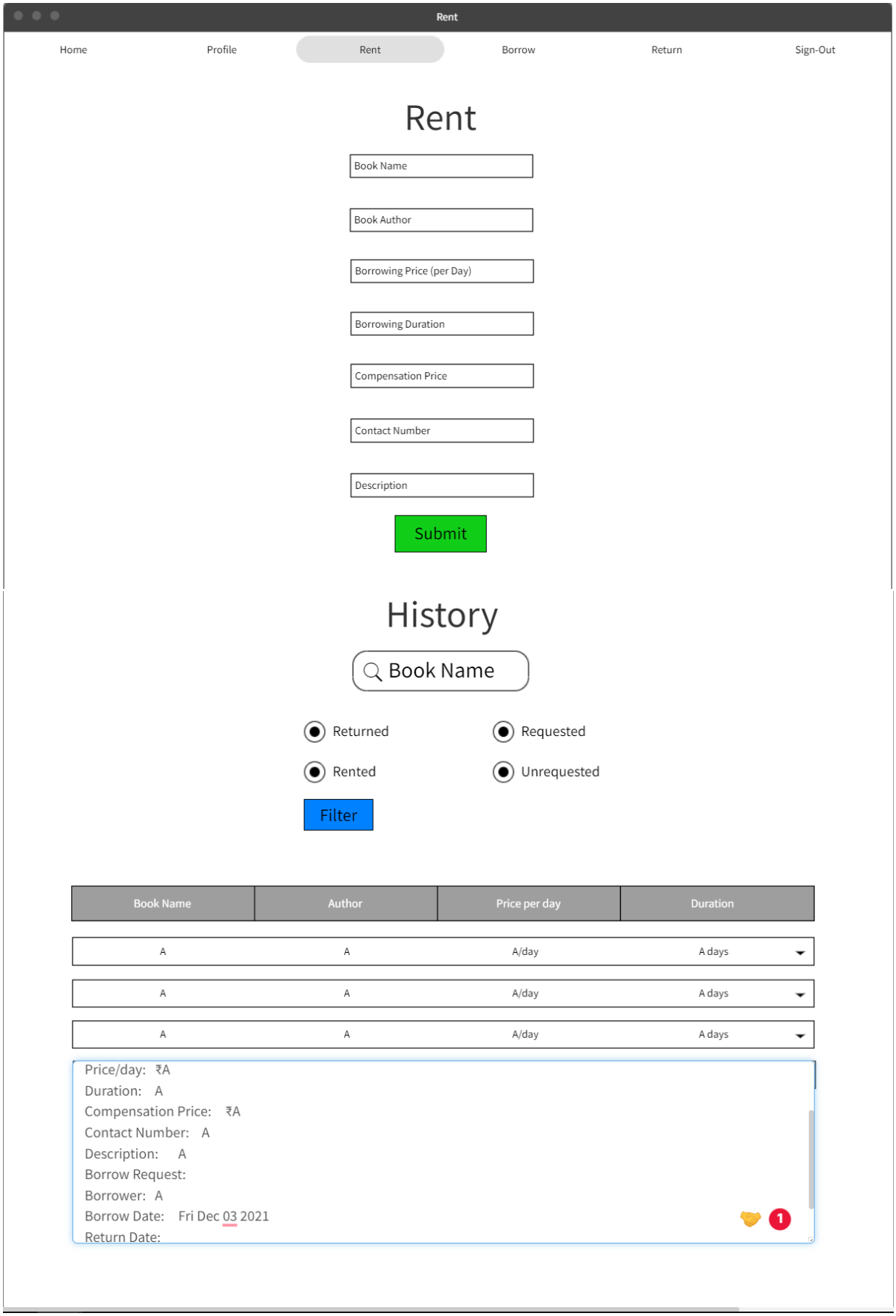

Books DB
History of returned books
User
Return a rented book
Return book
To complete the transaction
Generate OTP

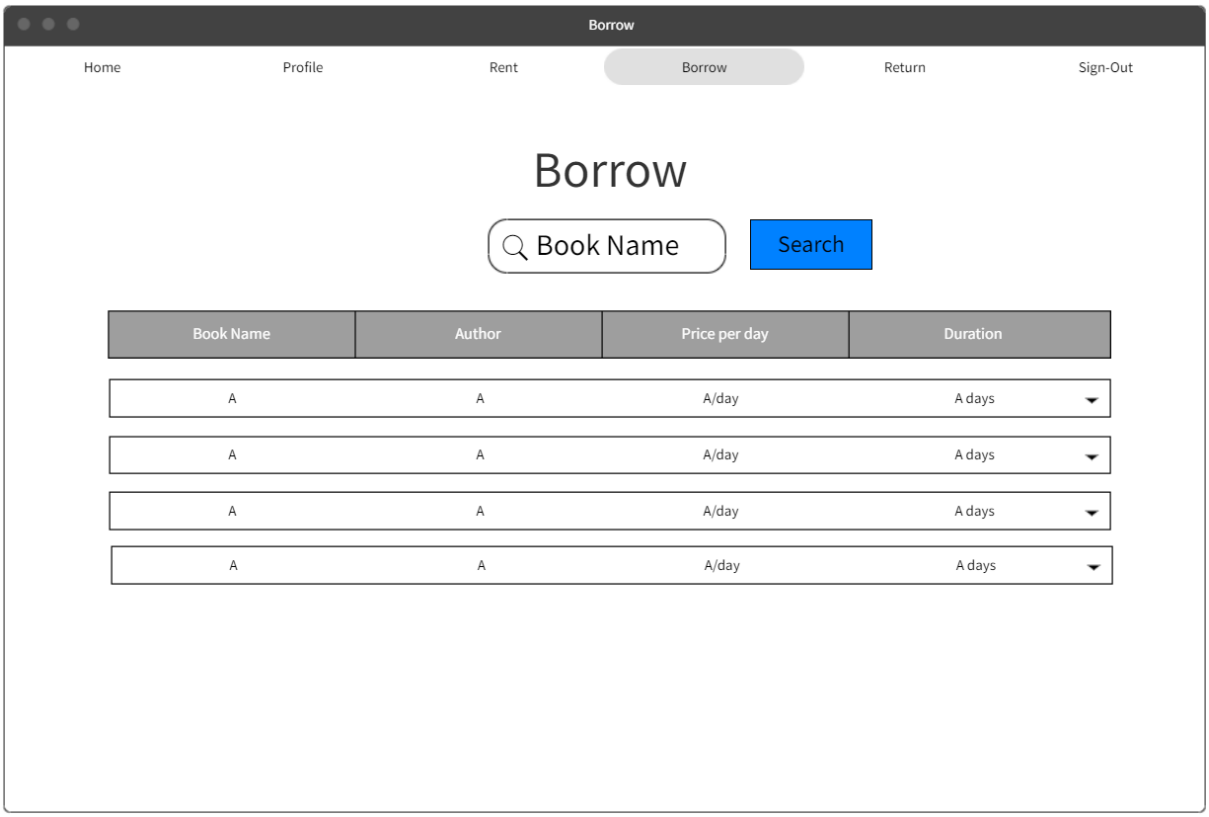
Wireframe

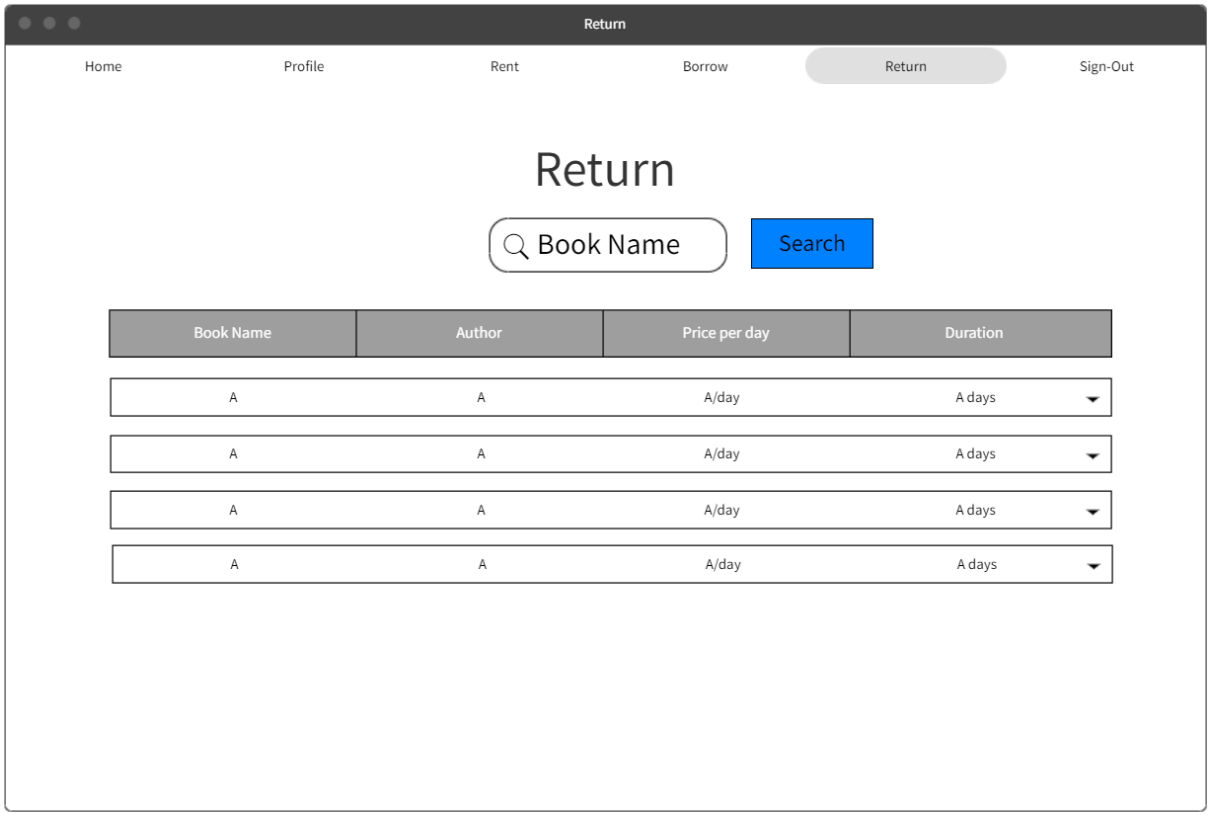

BookLook
Home
Sign-Up
Sign-In
About
Sign-Up
Name
Name
Email
Email
DOB
12 May 2016
User Name
User Name
Password
Password
Confirm Password
Confirm Password
Ph. No.
Phone Number
Address
Address
Sign-Up


Sign-In
Home
Sign-Up
Sign-In
About
Sign-In
User Name
User Name
Password
Password
Login


Profile
Home
Profile
Rent
Borrow
Return
Sign-Out
Profile
General Information
Name: XYZ
Email: XYZ.xyz.com
DOB: dd/mm/yyyy
Username: xyZ
Phone Number: 123456789
Address: B-121, RTS, Pal, Raj
Update Profile


Rent
Home
Profile
Rent
Borrow
Return
Sign-Out
Rent
Book Name
Book Author
Borrowing Price (per Day)
Borrowing Duration
Compensation Price
Contact Number
Description
Submit
History
Book Name
Returned
Requested
Rented
Unrequested
Filter
Book Name
Author
Price per day
Duration
A
A
A/day
A days
A
A
A/day
A days
A
A
A/day
A days
Price/day: ₹A
Duration: A
Compensation Price: ₹A
Contact Number: A
Description: A
Borrow Request:
Borrower: A
Borrow Date: Fri Dec 03 2021
Return Date:
1


Borrow
Home
Profile
Rent
Borrow
Return
Sign-Out
Borrow
Book Name
Search
Book Name
Author
Price per day
Duration
A
A
A/day
A days
A
A
A/day
A days
A
A
A/day
A days
A
A
A/day
A days


Return
Home
Profile
Rent
Borrow
Return
Sign-Out
Return
Book Name
Search
Book Name
Author
Price per day
Duration
A
A
A/day
A days
A
A
A/day
A days
A
A
A/day
A days
A
A
A/day
A days

Code

GitHub Link: https://github.com/Gaurav1020/BookLook

Testing

Testing Tool: Selenium python (Chromium Driver)

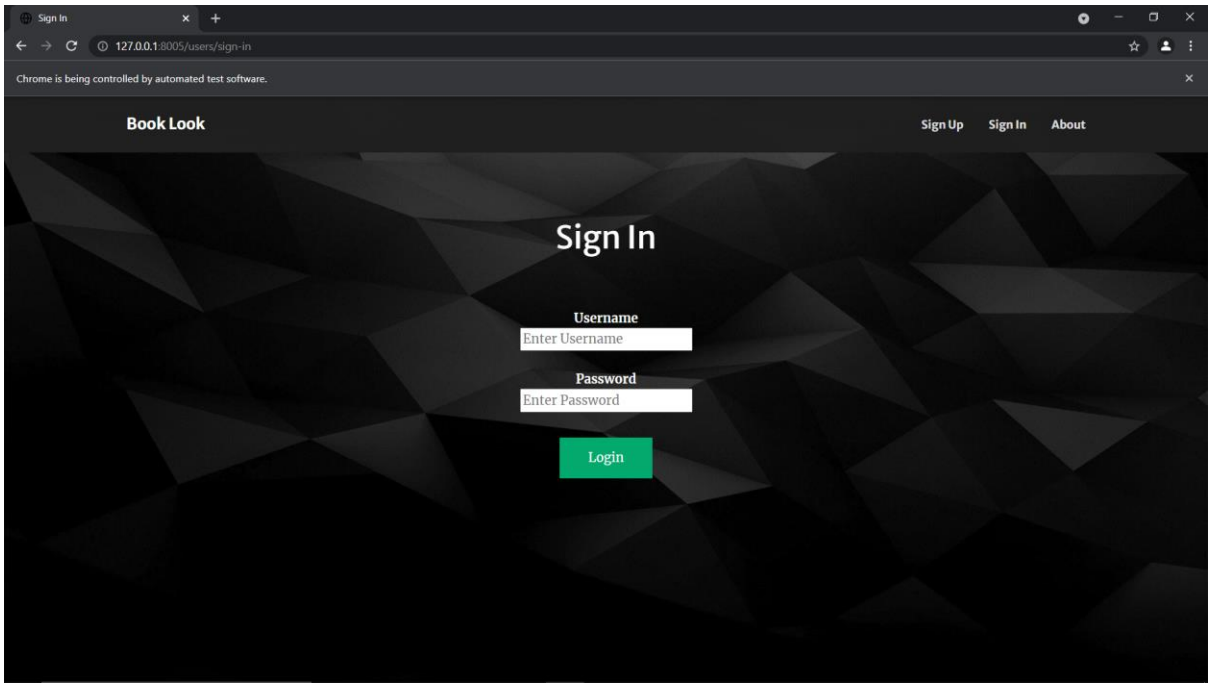
Code for testing login: (Each subsequent action occurs at a 10 second delay and the source code is also available on the same GitHub link named test.py)

```
from selenium import webdriver

import time

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.keys import Keys

from webdriver_manager.chrome import ChromeDriverManager

from selenium.webdriver.common.by import By

s=Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=s)

driver.maximize_window()

driver.delete_all_cookies()

driver.get('http://127.0.0.1:8005')

time.sleep(10)

driver.find_element_by_xpath("//a[contains(text(),'Sign In')]").click()

time.sleep(10)

driver.find_element_by_xpath("//header/div[1]/form[1]/div[1]/input[1]").send_keys('Gaurav_incorrect_username') #incorrect username

time.sleep(10)

driver.find_element_by_xpath("//header/div[1]/form[1]/div[1]/input[2]").send_keys('1020')

time.sleep(10)

driver.find_element_by_xpath("//button[contains(text(),'Login')]").click()

time.sleep(10)

driver.find_element_by_xpath("//header/div[1]/form[1]/div[1]/input[1]").send_keys('Gaurav')

time.sleep(10)

driver.find_element_by_xpath("//header/div[1]/form[1]/div[1]/input[2]").send_keys('102030_incorrect_password') #incorrect password

time.sleep(10)
```

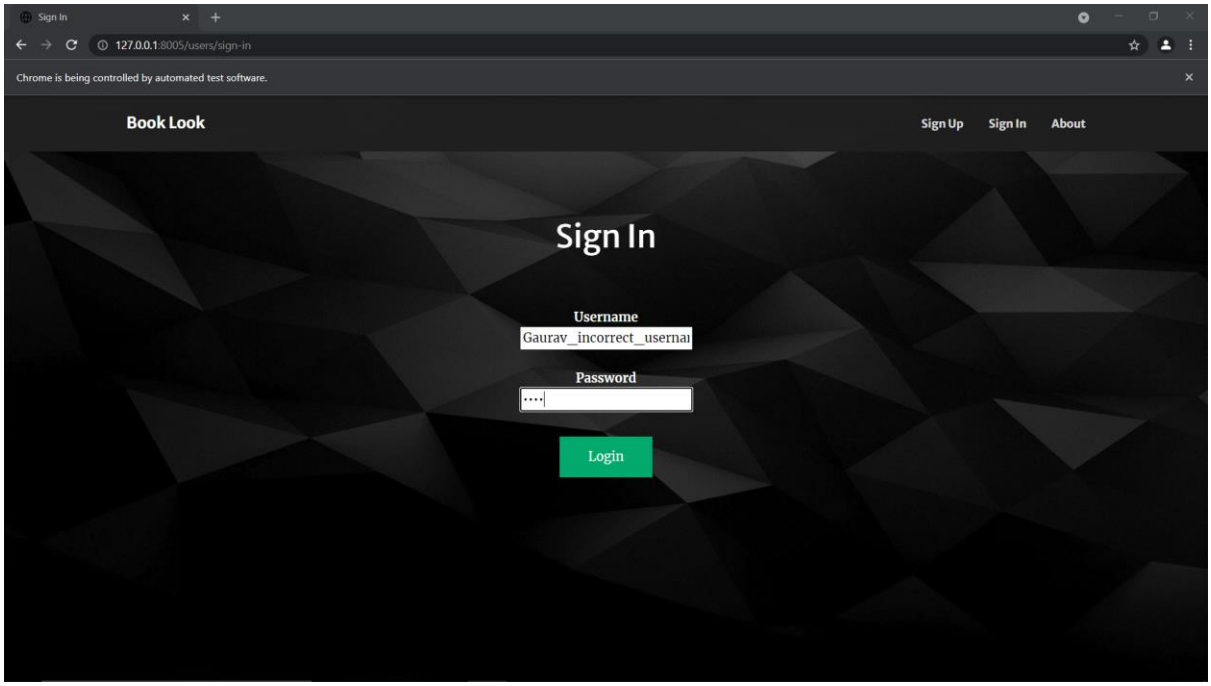
```
driver.find_element_by_xpath("//button[contains(text(),'Login')]").click()

time.sleep(10)

driver.find_element_by_xpath("//header/div[1]/form[1]/div[1]/input[1]").send_keys('Gaurav')
#correct username and password

time.sleep(10)

driver.find_element_by_xpath("//header/div[1]/form[1]/div[1]/input[2]").send_keys('102030')

time.sleep(10)

driver.find_element_by_xpath("//button[contains(text(),'Login')]").click()

time.sleep(10)

driver.find_element_by_xpath("//a[contains(text(),'Sign Out')]").click()

driver.close()
```

Screenshots:

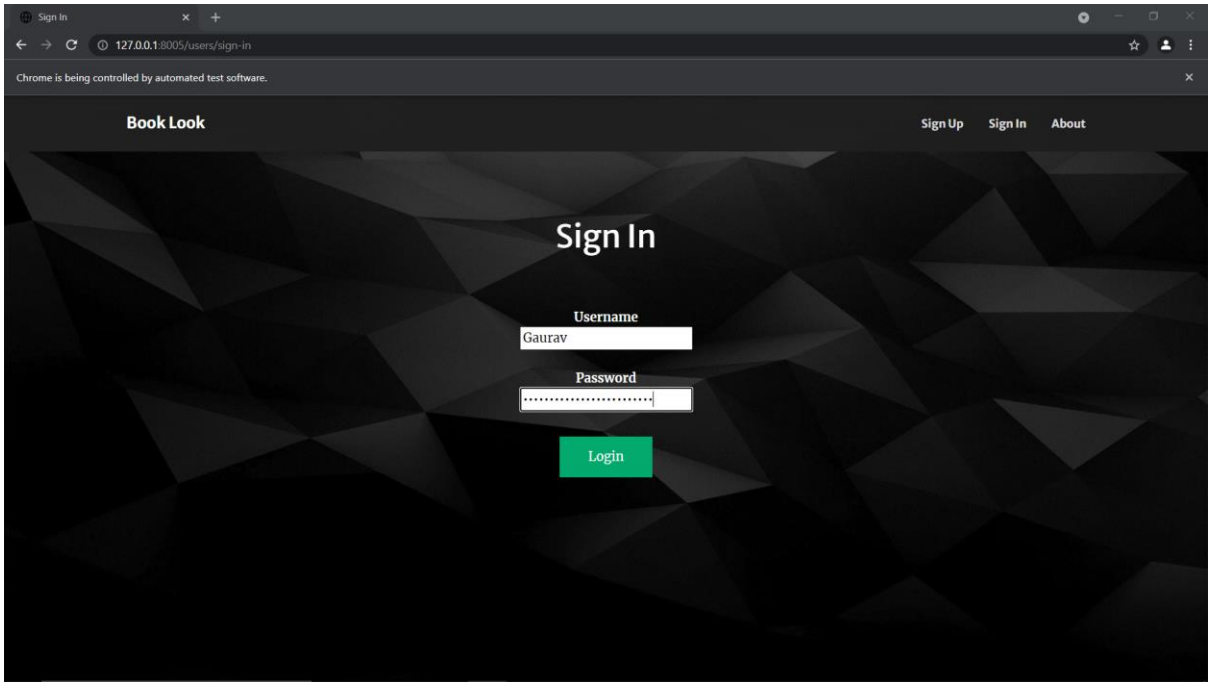
Starting sequence of the automated test software.



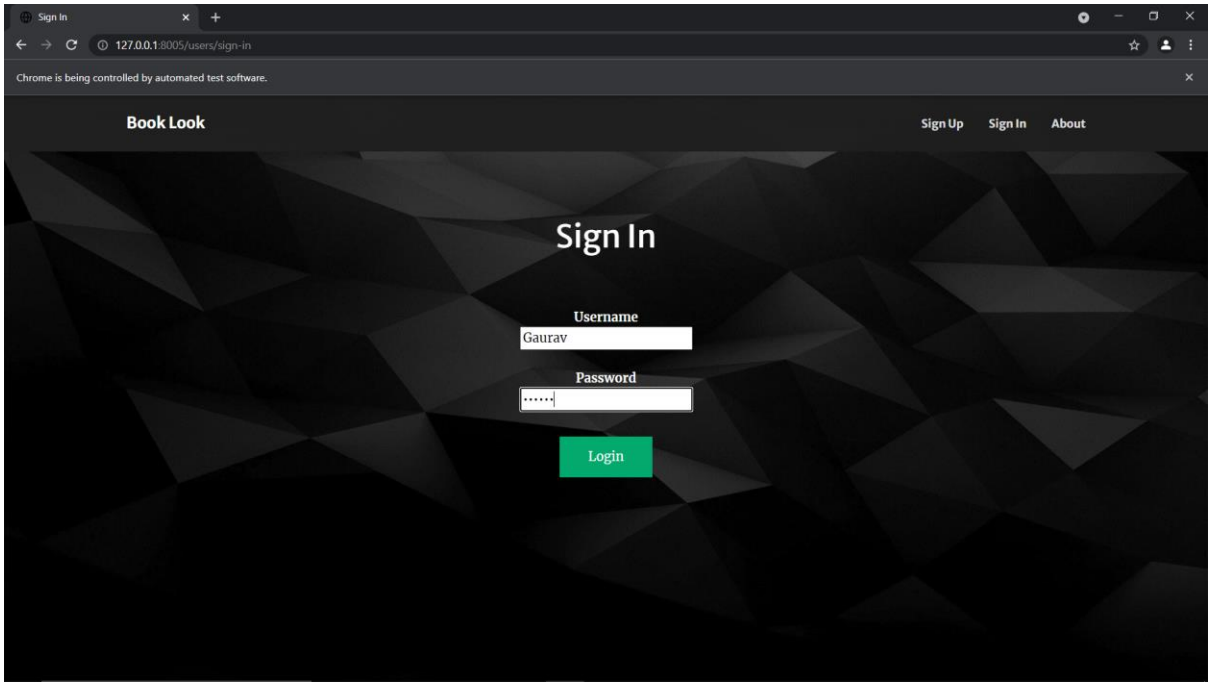
Incorrect Username, Correct Password=> Redirected back to sign-in page

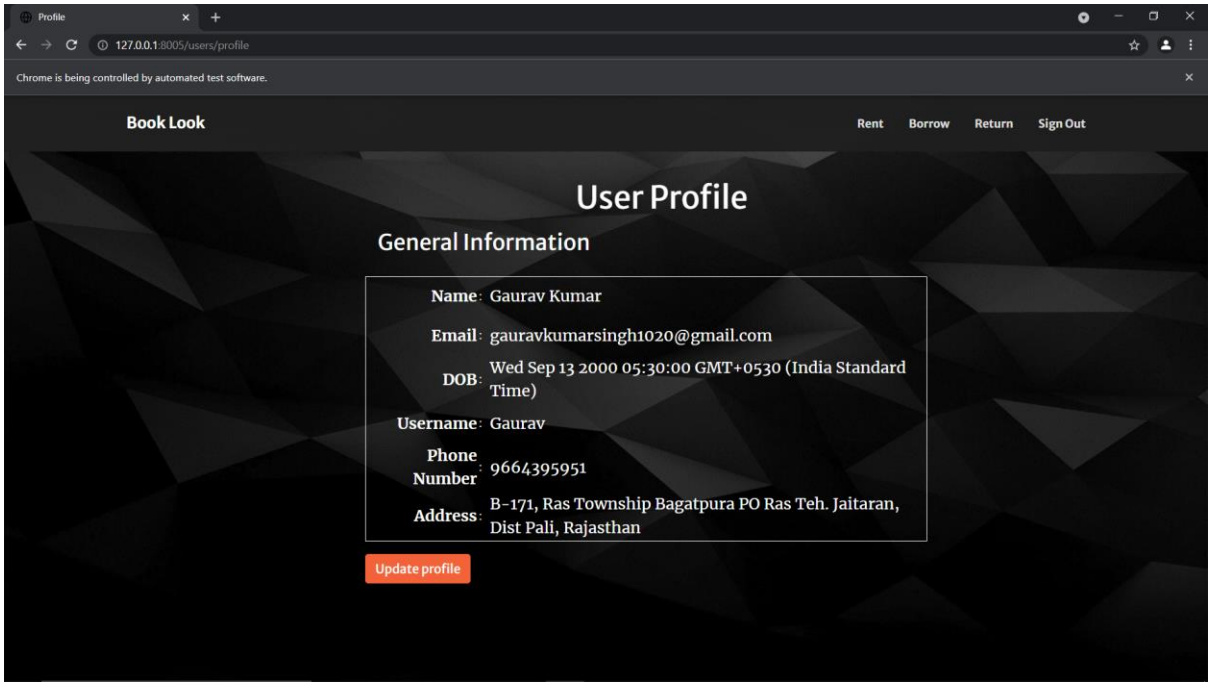


Correct Username, incorrect Password=> Redirected back to sign-in page



Correct Username, Correct Password=> Logged in and session established





Software Maintenance Analysis

Suggestions

1. The user can read the whole SRS document uploaded on GitHub link provided above.
2. For the project manager the system features are very important.
3. The developer must go through the whole SRS for understanding the requirement and functioning of software.
4. The designer and coder must see the class and object diagram and state transmission diagram for coding the modules.
5. A tester must be aware of coding language and visit through the code section and check the required output.
6. The document writer should write the qualitative document so that it becomes easy and understand to everyone.
7. New recruits should be trained to follow better coding practices which work in cohesion with existing codes.
8. More requests and responses must be converted to asynchronous requests and responses.

Conclusion

Book Look is an online web application where the customer can rent books online. Through a web browser, the customers can search for a book by its title or author, in general by this project one can rent the book from anywhere at any time. The convenience of an online book rental is a major consideration for most of us, especially those living in areas where access to a large bookstore is limited or is quite far away. Many books are too much expensive and for these types of books, this system is most preferable. If a book is too expensive, then one can rent that Book for some amount and can read that book by spending a very small amount of money.

After reading the whole book they can return a book and another one can take Benefit of this type of expensive books. Also, Textbook rentals are great for when you only need the textbook for a small section of the course. In such cases, purchasing a book just for one chapter could be a waste of money. So overall by this system one can read the book of their choice at any time by spending a very small amount of money. Using our platform both parties' problems will get solved. The person which is possessing any book will get some sort of rent price so he will get benefited and the person who wants to read the book can get a book at a much less price and should return that book as per the norms and conditions decided.

The user can also give feedback on the condition of the book and convey its overall renting experience by giving ratings on a score of five and can add beneficial comments so that the buyers get to know the genuine sellers. This also prevents the buyers from fraud and hence creating the best experience.

References

Technologies Used

The application is based on HTML, CSS and JavaScript with JavaScript specific frameworks such as:

Node.js (version), Passport.js, Express.js and mongoose.js (MongoDB)

cookie-parser: ^1.4.5,

ejs: ^3.1.6,

express: ^4.17.1,

express-session: ^1.17.2,

mongoose: ^5.13.7,

nodemon: ^2.0.12,

passport: ^0.4.1,

passport-local: ^1.0.0,

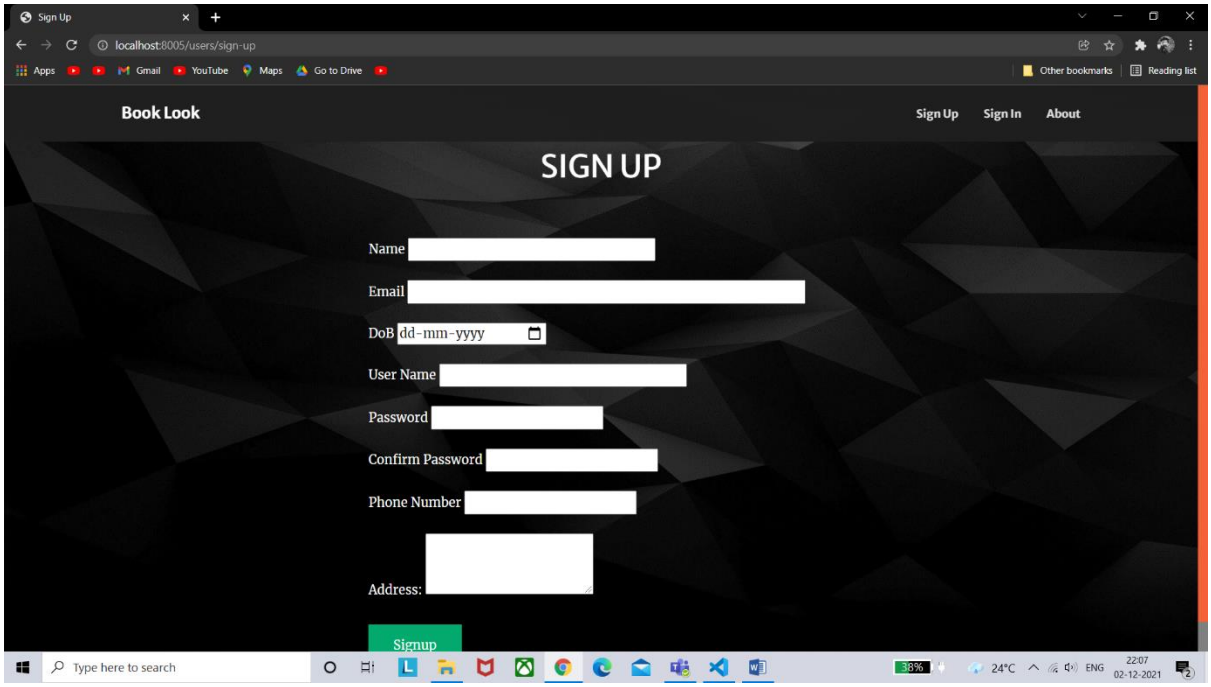
node: ^6.14.13

Research Papers

1. Ansh Agrawal, Rishabh Mathur (2020) Online Vehicle Rental System (https://ijsret.com/wp-content/uploads/2020/05/IJSRET_V6_issue3_362.pdf)
2. Willian Cerveny (2008) Implementing a Computerized Film Rental System (https://www.tandfonline.com/doi/abs/10.1300/J107v05n02_05?journalCode=wjcl20)
3. Mr. Suppasit Trakulrangsi (2003) Practical development of information system in business context: book rental system for N. Book Company (https://repository.au.edu/server/api/core/bitstreams/3c956109-32ec-4310-b71c-549bf43fb4fa/content)

Application Snapshots

SIGN-UP



Sign-up page consists of a form which is filled by the customer to get registered on the website. The form consists of 8 fields:

- Name (string)

  The name can contain any character

- Email (string)

  The email should be in a proper email address format

- Date of birth (date)

  The customer has to add their date of birth using the calendar.

- User Name (string)

  The username can contain any character but it has to be unique.

- Password (string)

  The password can contain any character

- Confirm Password (string)

  The confirm password input should match the password.

- Phone Number (string)

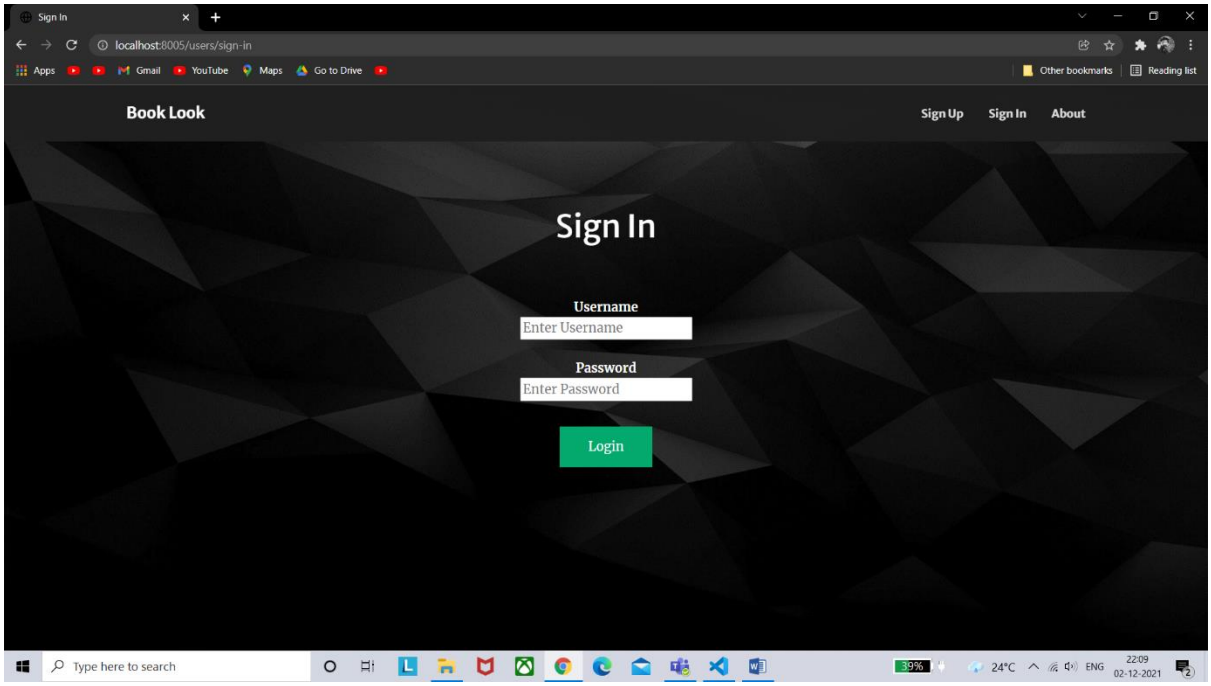
(+91) by default, enter a ten-digit phone number (no landline).

- Address (string)

  The address should contain the home address of the customer.

The customer has to fill each field of the form and then click on the signup button, after clicking on the signup button the customer will be registered as a user and their information will be stored in the database, and then the customer will be redirected to sign-in page.

SIGN-IN



Sign-in page consists of a form which is filled by the user to access their account. The form consists of two fields:

- Username (string)

  The username can contain any character but it has to be unique.

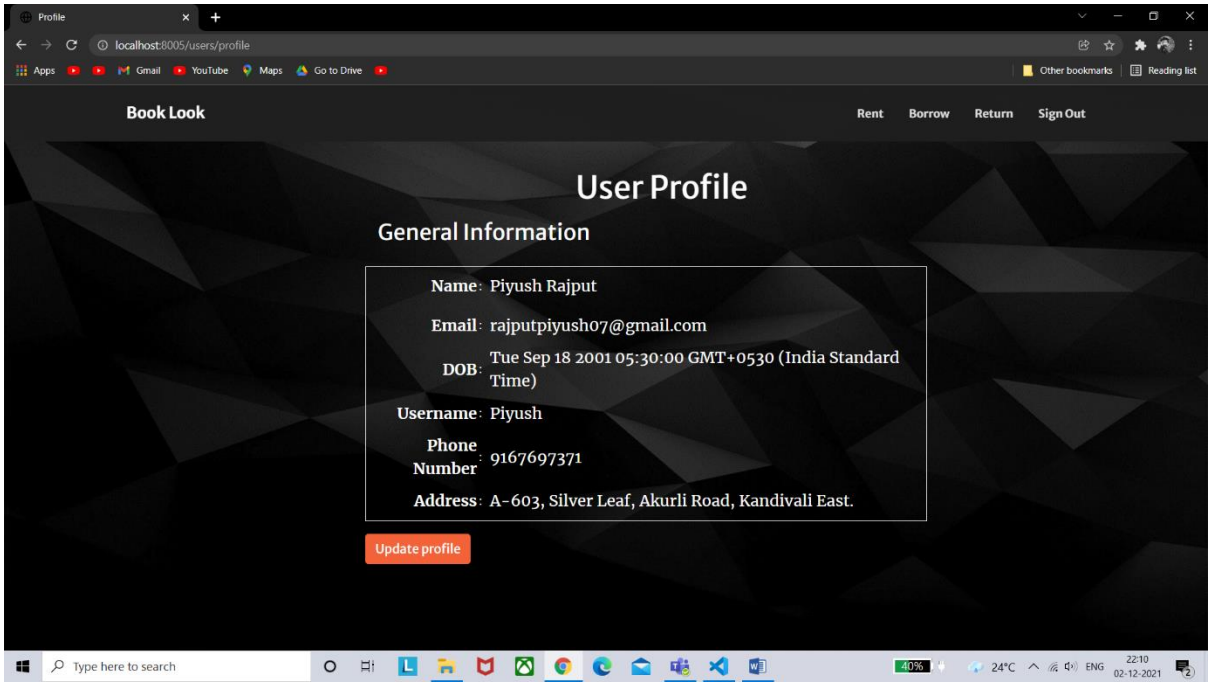
- Password (string)

  The password can contain any character

The user has to fill the form and then click on the signin button, after clicking the signing button the user credentials will be verified, if the user is registered in the system, a session

will be created and they will be redirected to their profile page otherwise if the credentials are not in the system then they will be redirected to the signin page.

PROFILE

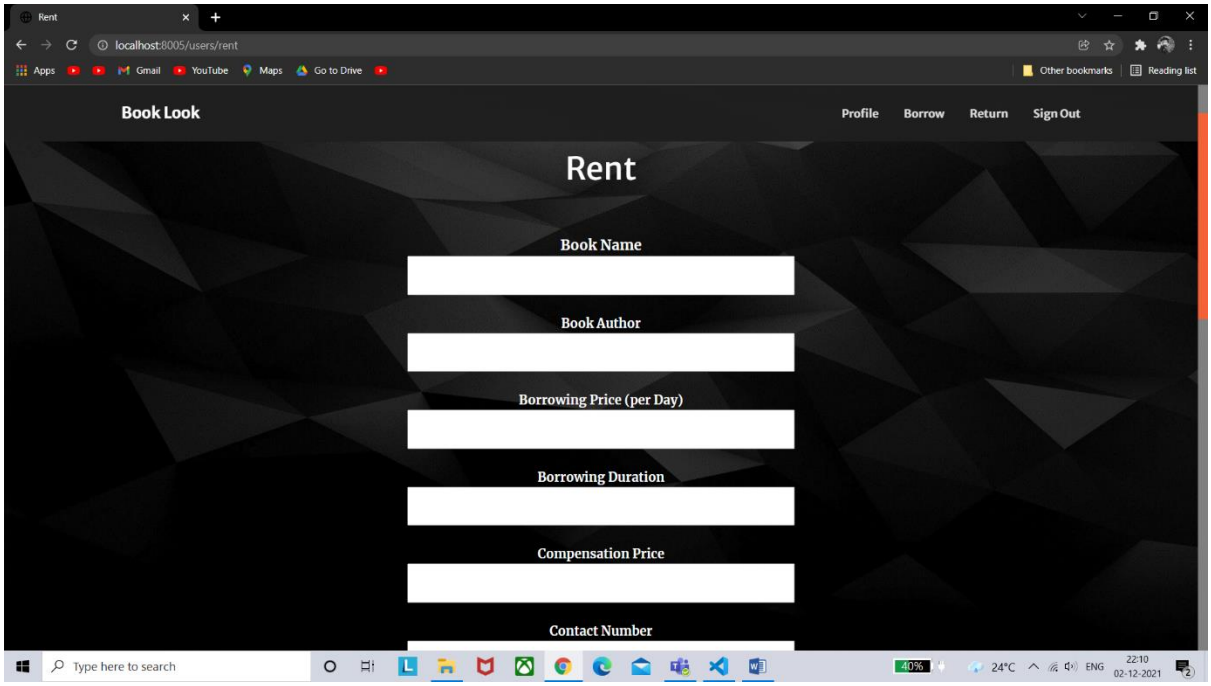


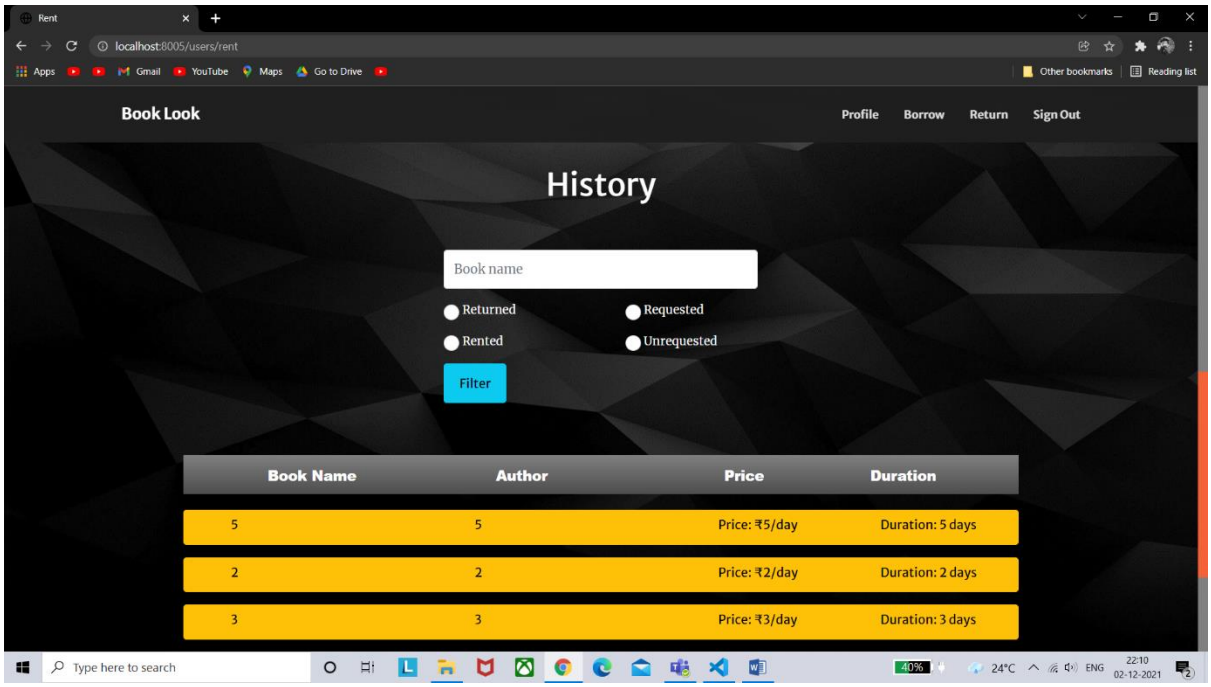
The profile page consists of a table and an update profile button. The table contains the information of the user such as name, email address, date of birth username, phone number and address.

If the user clicks on the update profile button, they will be redirected to the update profile page where the user can update their information (given in the table).

On the navigation bar there are three buttons rent, borrow, return and sign-out. If the user clicks on the rent button, then he/she will get redirected to the rent page, similarly for return and borrow the user will get redirected to the respective pages. If the user clicks on the signout button, the session will be deleted and they will get redirected to the signin page.

RENT





The rent page consists of a form and a rent history, the form is used to enter the details of the book the user wants to rent and the rent history is used to see the books rented by the user in the past.

The form contains seven fields:

- Book Name (string)

  The book name can contain any only alphabets and numbers.

- Book Author (string)

  The book author can contain any character.

- Borrowing Price per day (string)

  The borrowing price should only contain numbers.

- Borrowing Duration (string)

  The borrowing duration contains the amount of time the book will be rented for. It should only contain alphabets and numbers.

- Compensation Price (string)

  The compensation price should only contain numbers.

- Contact Number (string)

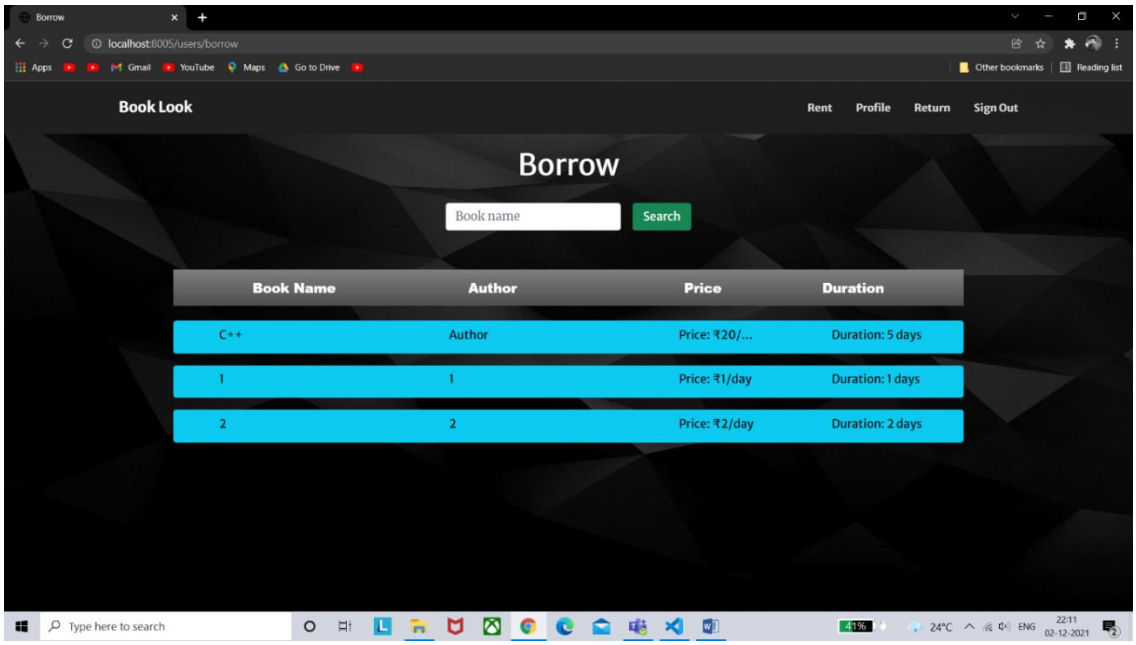
  (+91) by default, enter a ten-digit number.

- Description (string)

  The description can contain any character.

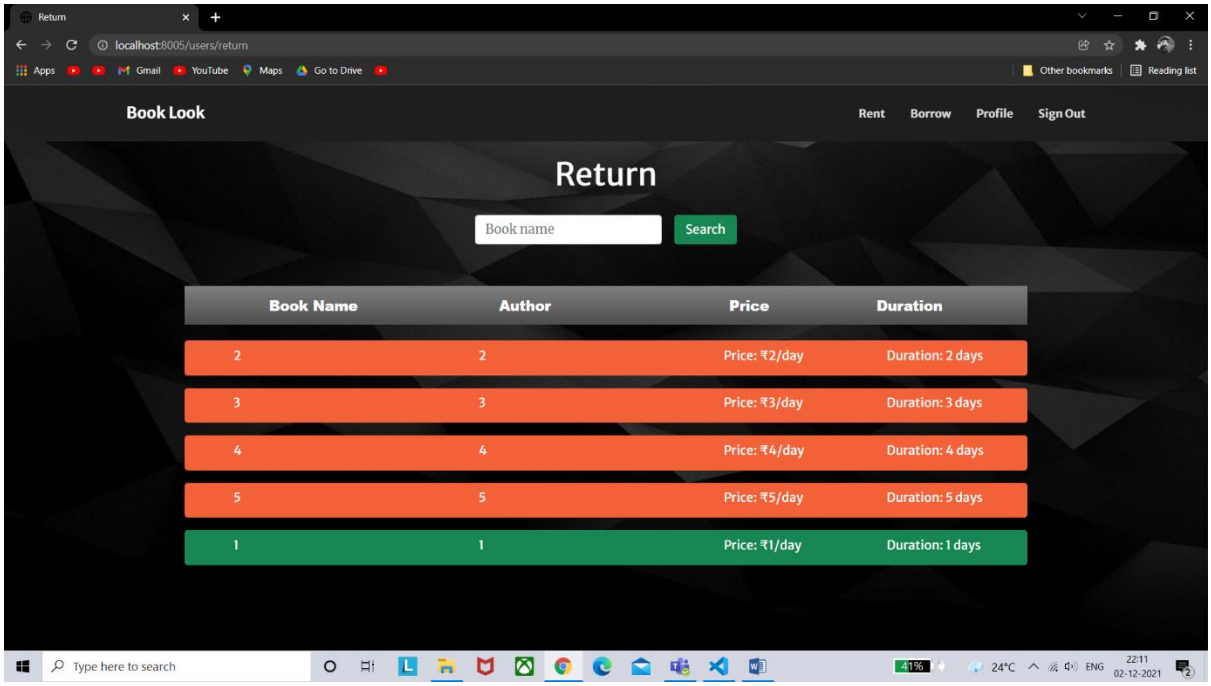
The rent history contains the history of the books rented by the user. The user can validate the borrower and it also contains an OTP field. The OTP is used by the borrower to return the book which is randomly generated and provided by the owner of the book (renter).

BORROW



The borrow page consists the borrow information of the books. It also contains a borrow button which sends the request to the renter for borrowing the book. It shows all the books which are available for borrowing, it shows all the details of the books such as book name, author name, price, duration, name and contact number of the renter and a compensation price if the book is damaged.

RETURN



The return page contains the information of the books borrowed by the user; it also contains a section return which allows the user to return the book to its owner. To return the book the user has to enter the OTP given by the owner of the book. The book can only be returned after submitting the correct OTP by clicking on the submit button.