Software Design Specification

Book Renting Service

Revision 1.0

Group Members:

Piyush Rajput (19BCE0689)

Gaurav Kumar Singh (19BCE2119)

Ayush Gupta (19BCE2422)

Aryan Singh (19BCE2295)

Revision History

Version	Name	Reason For Changes	Date
1.0		Initial Revision	

Approved By

Approvals should be obtained for project manager, and all developers working on the project.

Name	Signature	Department	Date

1. Introduction

This section gives a scope description and overview of everything included in this Report. The Report is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. Its primary purpose is to provide a clear and descriptive "statement of user requirements" that can be used as a reference in further development of the software system. This document is broken into a number of sections used to logically separate the software requirements into easily referenced parts. This Report aims to describe the Functionality, External Interfaces, Attributes and Design Constraints imposed on Implementation of the software system described throughout the rest of the document. Throughout the description of the software system, the language and terminology used should be unambiguous and consistent throughout the document.

1.1 Purpose

The main purpose of the project is to search and rent a book based on title and author. It will also help the users to know about the latest books in the market. The selected books are displayed in a tabular format and the user can order their books online through Cash on delivery.

1.2 System Overview

The software system being produced is called Book rental System or BRS. It is being produced for a customer interested in renting books via the Internet. This system is designed to "provide automation support" for the process of ordering books on rent. This system is largely cross-platform and is available to anyone specially pursuing engineering in Computer Science. The Book Rental System will allow any user to create an account to become a customer. The customer, through the process of account creation, will have the option to become a member of the site. It will enable the users (customers) to view the real time inventory and extract book information, enable users to place orders online and pick up books in-store and access to a personalized account profile etc. The system will allow customers to browse, search, select, and add books to a shopping cart. Then, provided they have books in their shopping cart, check out books in shopping cart and decrement the stock that the inventory the system maintains. The BRS also allows a manager to manage the inventory with full create, retrieve, update and delete (CRUD) functionality with regards to books in the system.

Admin has authority to add/delete users, grant permissions to members and users to rent books and in return is also responsible for generating E-Mail messages for customer regarding book purchase transaction and delivery. Traditionally, customers are used to buy the Book at the real, in other words, factual shops. It needs the customers to show up in the shops in person, and walk around different shopping shelves, and it also needs the owners of shops to stock, exhibit, and transfer the products required by customers. It takes labor, time and space to proceed these operations. Also, the high cost of new books proved a problem for today's students as they needed to buy the books that for short period of times for e.g. to study for semester exams.

1.3 Design Map

SUE - Summarize the information contained within this document or the family of design artifacts. Define all major design artifacts and/or major sections of this document and if appropriate, provide a brief summary of each. Discuss any significant relationships between design artifacts and other project artifacts.

1.4 Definitions and Acronyms

Sign-in
Sign-in is for the session creation

Sign-up
Sign-up is for the registration of customers & lenders.

Sign-out
Sign-out is for ending an already existed session

User profile management
The User Profile Management feature provides central management for user-specific data and settings stored in the users' model.

Book rental system
Users can make the books available for rent from here for other people to borrow.

Book borrower system
Users can browse the books and send a borrow request.

Book return system
This is where the borrower will return the book and pay if any due is there.

2. Design Considerations

2.1 Assumptions

- The application is going to be used locally on small scale for now because it is proof of concept.
- Sufficient user base will follow through to use the application to suffice the independent nature of the application.
- External payment methods are used outside of the application to complete transactions.

2.2 Constraints

- End user cannot use the platform without sign up.
- Renter can validate the borrower and decide whether to rent the borrower or not and same for borrower.

- OTP is generated to end the borrowing time as per the final invoice will be generated.

2.3 System Environment

HTML, CSS and JavaScript are used for front end. Nodejs, Expressjs, Passportjs, Bootstrap, JQuery, Mongoose are used for back end. MVC (Model, View, Controller) is used as architecture. MongoDB (NoSQL) is used for database. The process model used is incremental model. The other software used are google chrome, visual studio code, visual paradigm online and robo3t/compass.

2.4 Design Methodology

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:

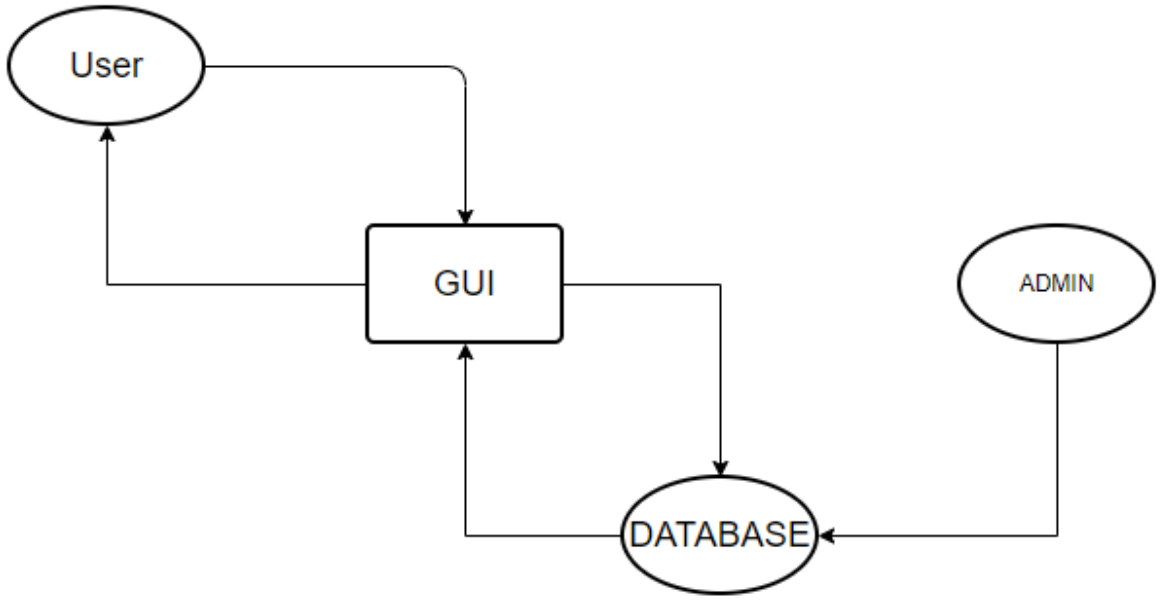
- Database model creation
- Sign-in, Sign-up, Sign-out
- User Profile management
- Book rental system
- Book borrower system
- Book return system

After database model creation, the next 5 modules can be designed and maintained as independent modules. This makes our project scalable, easy to maintain and modify, and makes the development process a lot faster.

2.5 Risks and Volatile Areas

- The application is vulnerable to network attacks such as XSS and SQL mapping.
- Since the application is not hosted on AWS for now, it won't take advantage of the bad request filtering and mitigating techniques which cloud platforms provide.
- The scope of the application is limited by its very nature because, the books are either to be picked-up manually or delivered which can constraint the use case of application to within a small area.
- The application's workflow exploitation can lead to some unrecognized ways of fraud.

3. Architecture



3.1 Overview

The system is structurally and functionally decomposed into the following modules.

Sign-in, Sign-up, Sign-out

Sign-in

Sign-in is for the session creation

Sign-up

Sign-up is for the registration of customers & lenders.

Sign-out

Sign-out is for ending an already existed session

User profile management

The User Profile Management feature provides central management for user-specific data and settings stored in the users' model.

Book rental system

Users can make the books available for rent from here for other people to borrow.

Book borrower system

Users can browse the books and send a borrow request.

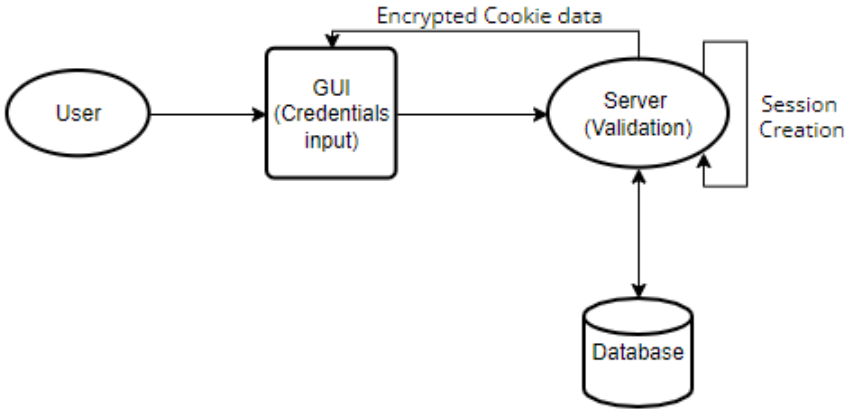
Book return system

This is where the borrower will return the book and pay if any due is there.

3.2 Module 1: Sign-in, Sign-up, Sign-out

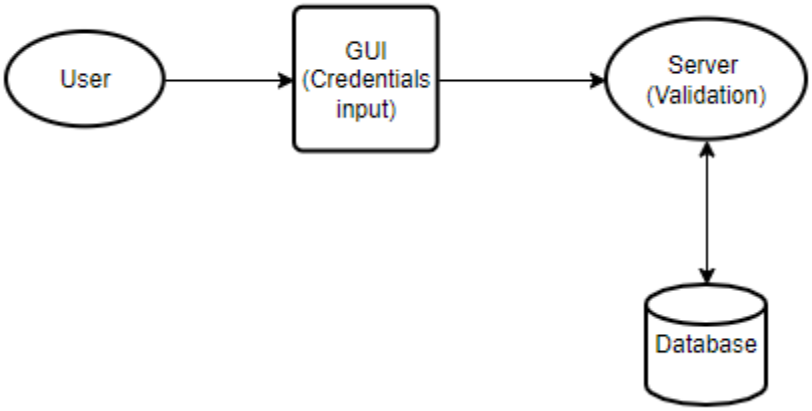
Sign-in

Sign-in is for the session creation. The user credentials are taken and matched in the database. If the credentials match, the session is established on server and a cookie with encrypted user login session data is sent to the user’s device for user identification on subsequent user requests to the server.



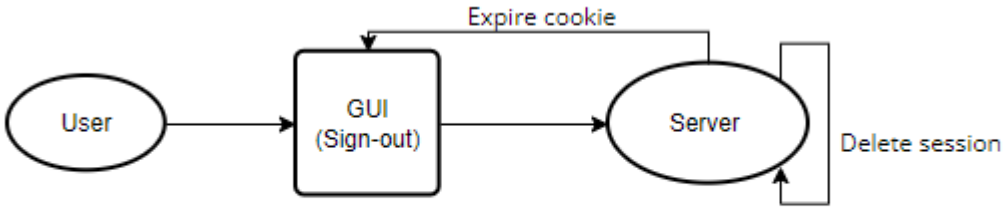
Sign-up

Sign-up is for the registration of customers & lenders. The user details are inputted and checked for validity. If the entries are deemed valid, the data is pushed to the database.



Sign-out

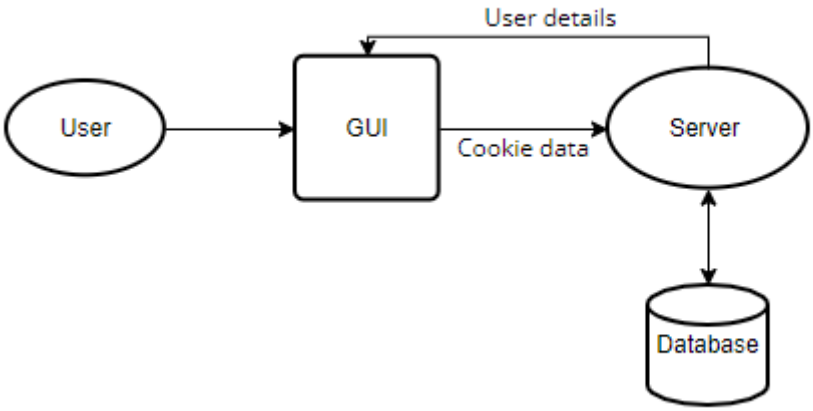
Sign-out is for ending an already existed session



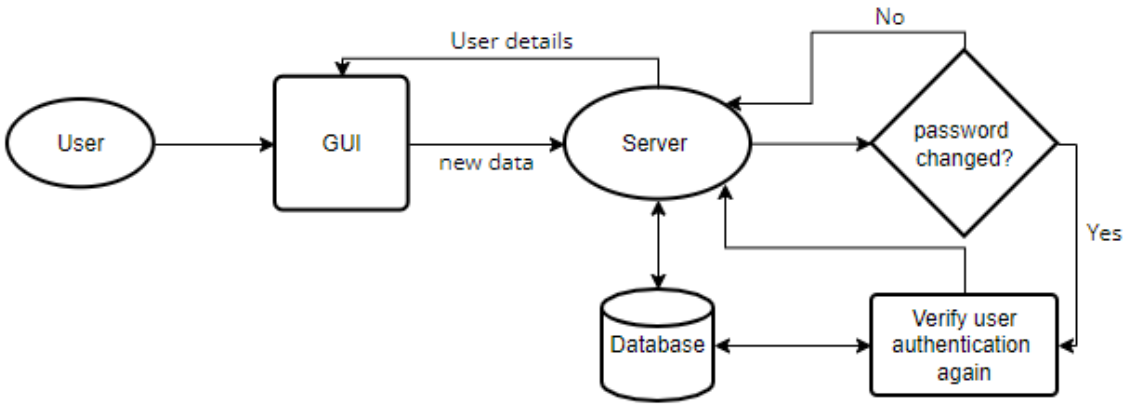
3.3 Module 2: User Profile Management

User can view their personal details from the database here and also update their details including password which requires then one extra step of authentication.

Profile View

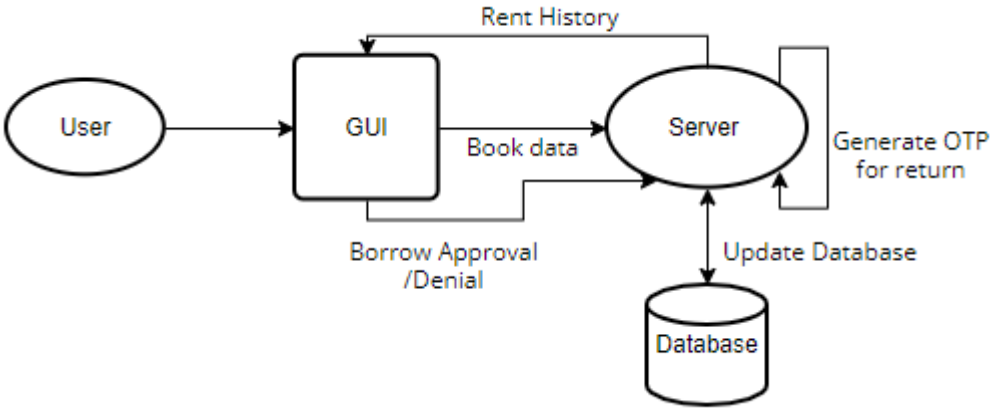


Profile Update



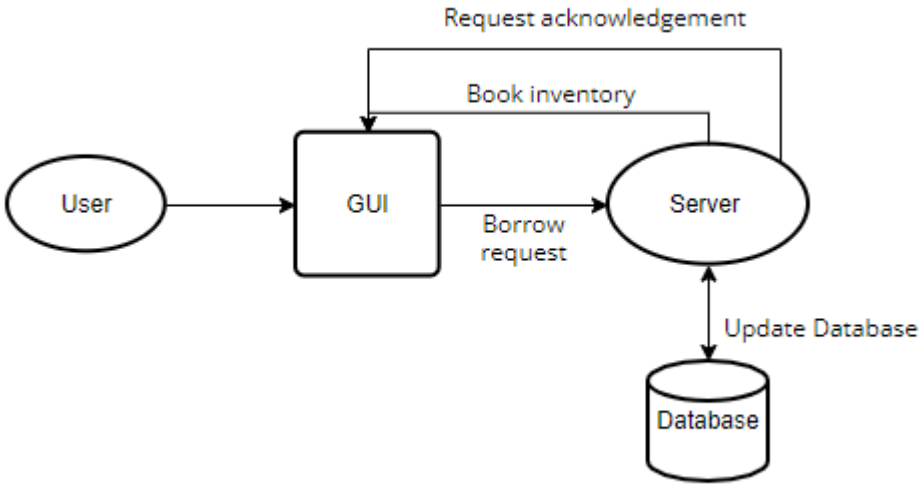
3.4 Module 3: Rental System

Here, user can put up their books for rent and view their rental history and validate borrowers so renters have the flexibility to decide who they want to rent the book to.



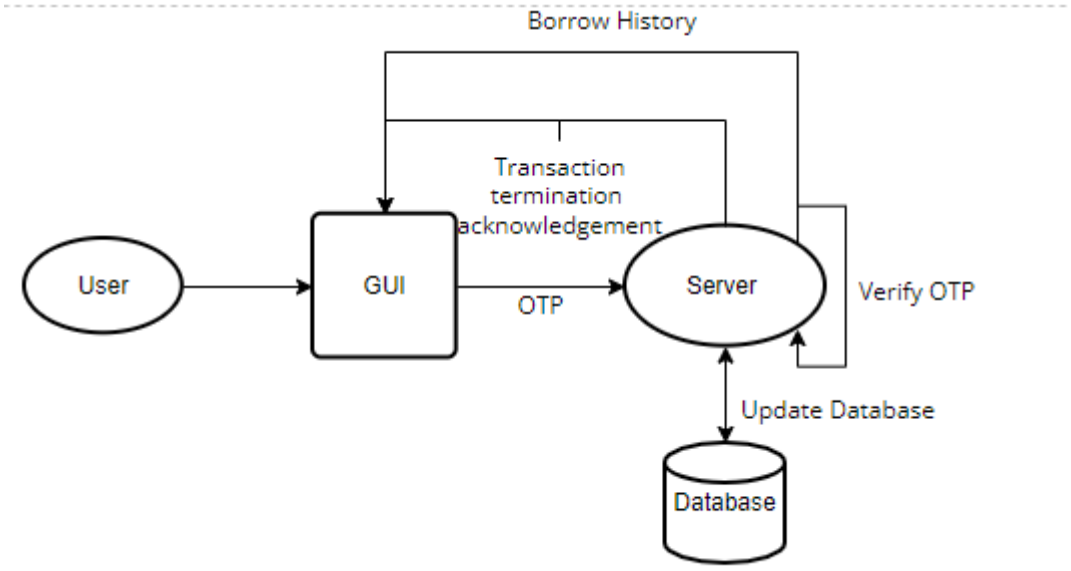
3.5 Module 4: Borrower System

This section of the web application shows the public details of all the books currently available for rent along with who put up the book for rent. Users can send borrow request to the renter and wait for approval from the renter's end to initiate a transaction.



3.6 Module 5: Return System

The borrowers can return the books and close the transactions by entering the OTP which the renter have to convey to the borrower to ensure the transaction is closed from both ends.



3.7 Strategy

We used incremental model for the development process of the application. The main reason for using workflow design we used is to make sure the user experience is not convoluted/confusing and the interface itself becomes intuitive. All the data is classified under the tabs they are relevant to. The sign-in procedure could have been made very simple by using manual authentication which would have simplified the project a lot but is significantly less secure than the alternative we used which is making use of passport.js framework to establish user sessions. The user session creation itself was a question in the start, “Should it be necessary for the user to create an account on the platform to be able to perform any task on it?”, but we concluded that, it is in-fact necessary to prevent frauds to some extent since the records can be maintained and the authenticity of the users can be determined if all the activities they perform can be linked back to their accounts.

The rent and borrow request modules were fairly straightforward in their execution since the modules we added into them were indisputable for their effectiveness, but the problem arrived again during the return procedure. We wanted to ensure that the platform is as fraud-proof as we can and the return procedure is probably the most vulnerable to frauds caused by workflow exploitation. We thought of numerous alternatives such as escrow-based transaction completion, OTPs, advances etc. but the OTP turned out to be the most independent module to implement. Escrow is also great option (also used in micro/non-trivial transactions in cryptocurrencies such as Bitcoin) and we think if the application scales, escrow-based transaction termination alongside OTP based transaction completion can help make the return procedure fraud-proof but the escrow module cannot be implemented properly in the proof-of-concept stages of the project.

4. Database Schema

4.1 Tables, Fields and Relationships

We primarily used two NoSQL database models for our project which are:

1) User Model
2) Book Model

- User Model: This model contains all information about a particular user. Its model is as following:

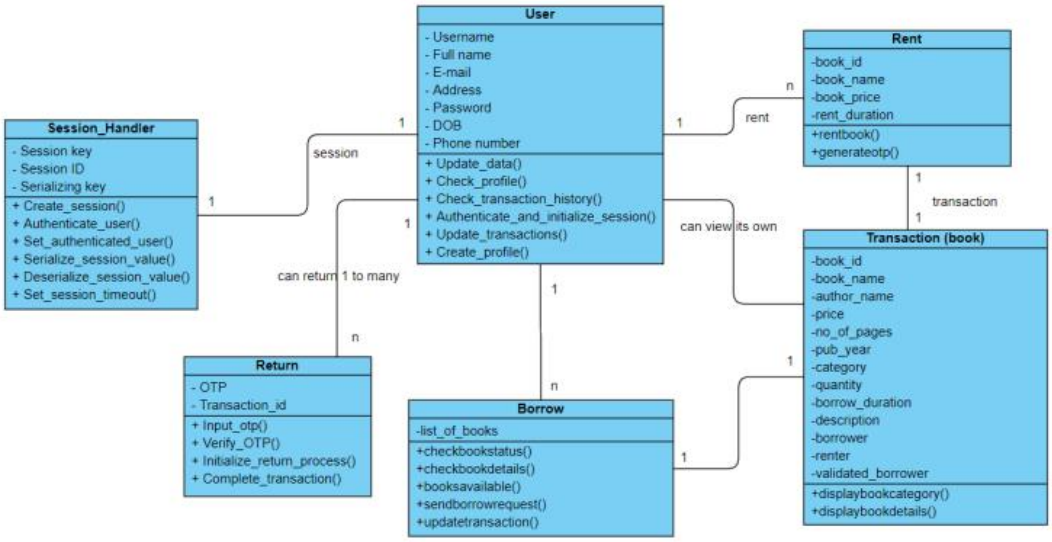
```
Name: {
        type: String,
        required: true
},
E-mail: {
type: String,
        required: true,
        unique: true
},
DOB: {
        type: Date,
required: true,
},
Username: {
        type: String,
        required: true,
        unique: true
},
Password: {
        type: String,
        required: true
},
Phone no: {
        type: Number,
        required: true
},
Address: {
        type: String,
        required: true
}
```

- Books Model: This model contains about all incoming and outgoing books. All the current transactions and history is stored here. This model is used to link the users to transaction and maintain records. Its model is as following:

```
book name: {
   type: String,
   required: true
},
book author: {
   type: String,
   required: true
},
```

```
borrow price: {
   type: Number,
   required: true
},
borrow duration: {
   type: Number,
   required: true
},
compensation price: {
   type: Number,
   required: true
},
phone: {
   type: Number,
   required: true,
},
description: {
   type: String
},
renter: {
   type: Foreign Key to user model,
   required: true
},
borrower: {
   type: Foreign Key to user model
},
Validated borrower: {
   type: Foreign Key to user model
},
borrow date: {
   type: Date,
},
return date: {
   type: Date,
},
total: {
   type: Number
},
returned: {
   type: Boolean,
   default: false
},
OTP: {
   type: Number,
   default:0
}
```

CLASS DIAGRAM



4.1.1 Databases

Mongo DB used as database. MongoDB is a NoSQL database implying that the data stored in it is not structured and is flexible in nature, so any future model updates can easily be accommodated into the database.

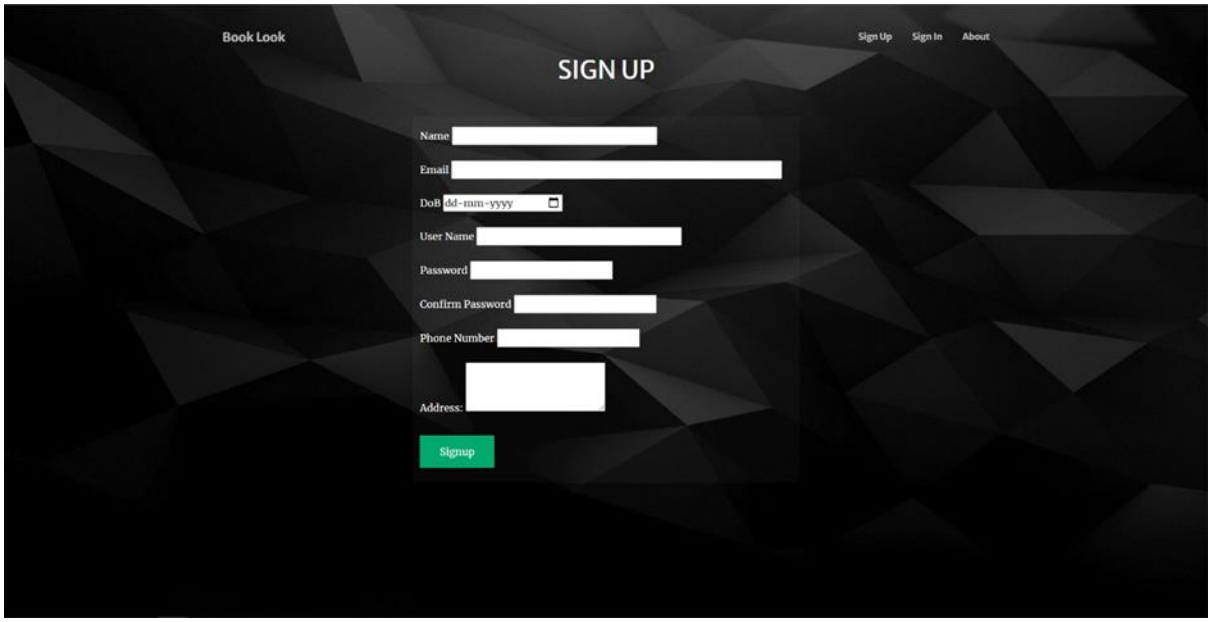
4.1.2 Fields Change(s)

Table Name	Field Name	Field Type/Constraints
User Data	Name	type: String, required
	Email	type: String, required
	DOB	type: Date, required
	Username	type: String, required, unique
	Password	type: String, required
	Phone number	type: Number, required, pattern: \^[0-9]{10}$\
	Address	type: String, required
Book Data	Book name	type: String, required
	Author	type: String, required
	Borrow Price	type: Number, required
	Borrow Duration	type: Number, required
	Compensation Price	type: Number, required
	Phone number (renter)	type: Number, required, pattern: \^[0-9]{10}$\
	Description	type: String
	Renter	type: Foreign key ID, reference: User Data, required
	Borrower	type: Foreign key ID, reference: User Data
	Validated Borrower	type: Foreign key ID, reference: User Data
	Borrow Date	type: Date,

	Return Date	type: Date, required
	Total	type: Number
	Returned	type: Boolean
	OTP	type: Number

5. High Level Design

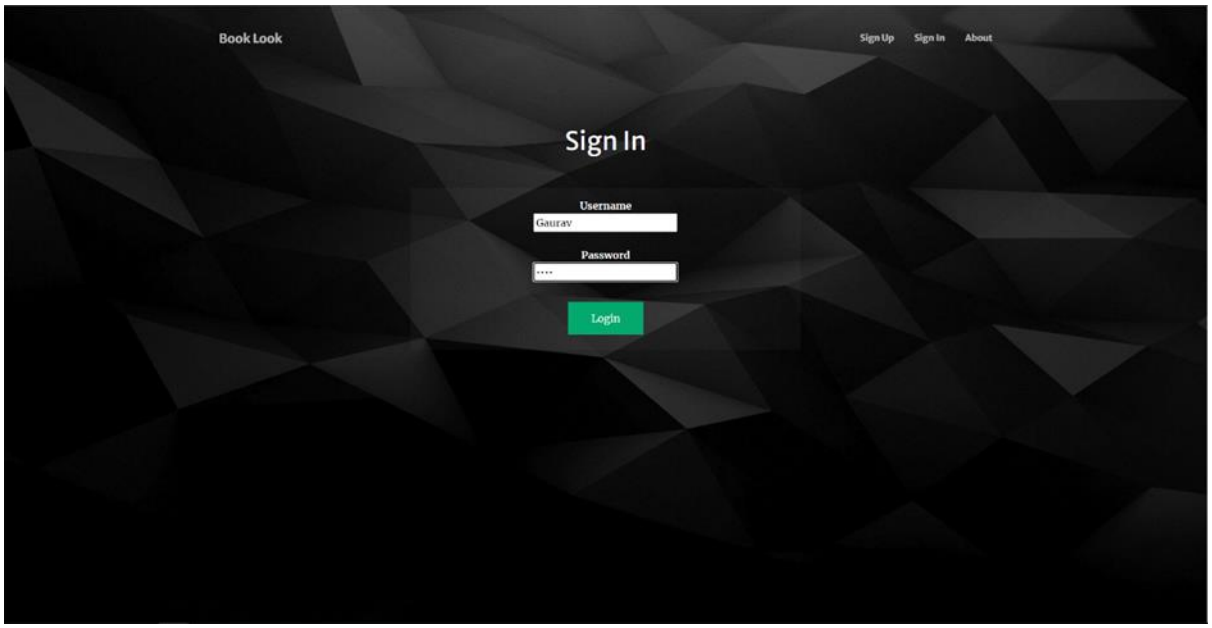
5.1 SIGN-UP



Description

It is used by the users to create new accounts. The users are required to provide their general information such as name, date of birth, email, phone number and address along with a username and a password.

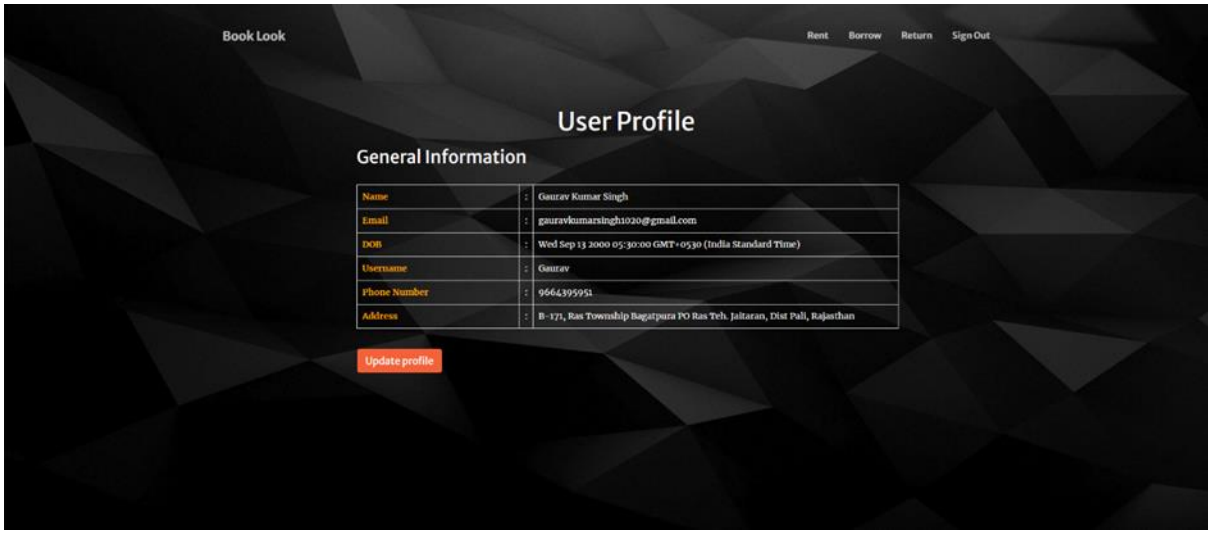
5.2 SIGN-IN



Description

It is used by the users to login into the system. They are required to enter username and password and then the system verifies whether the user exists or not.

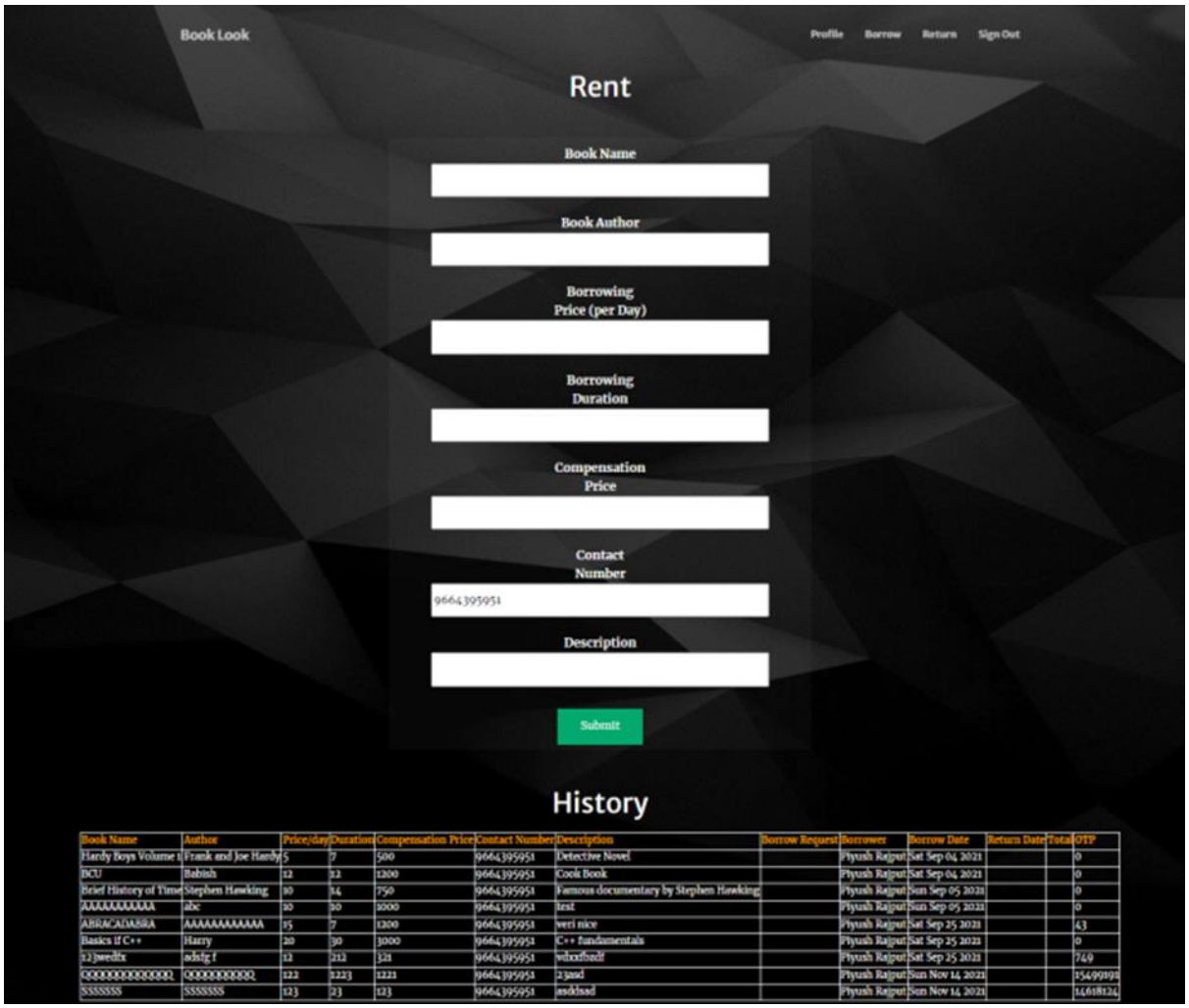
5.3 PROFILE



Description

It provides the general information of the user such as name, email, date of birth, username, phone number and address. Here if the user wants to make any changes in his/her profile, the update profile button has been provided to do the same.

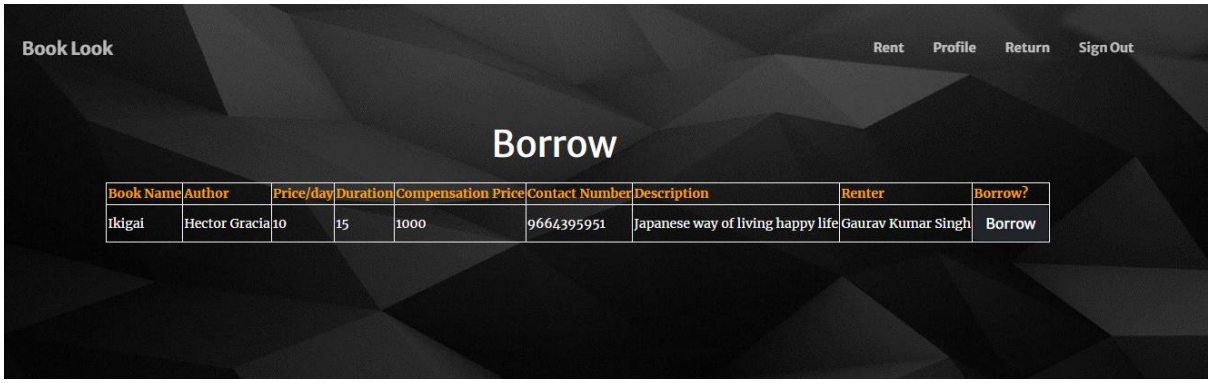
5.4 RENT



Description

It allows users to rent a book, the user has to enter all the details of the book. It also provides a history of the books rented by the user and it also allows user to validate a borrower.

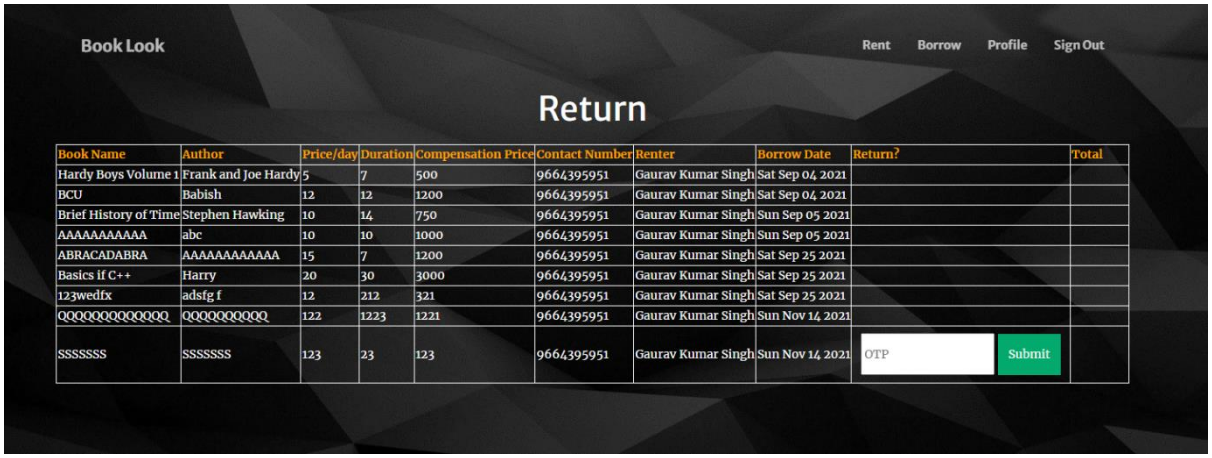
5.5 BORROW



Description

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.

5.6 RETURN

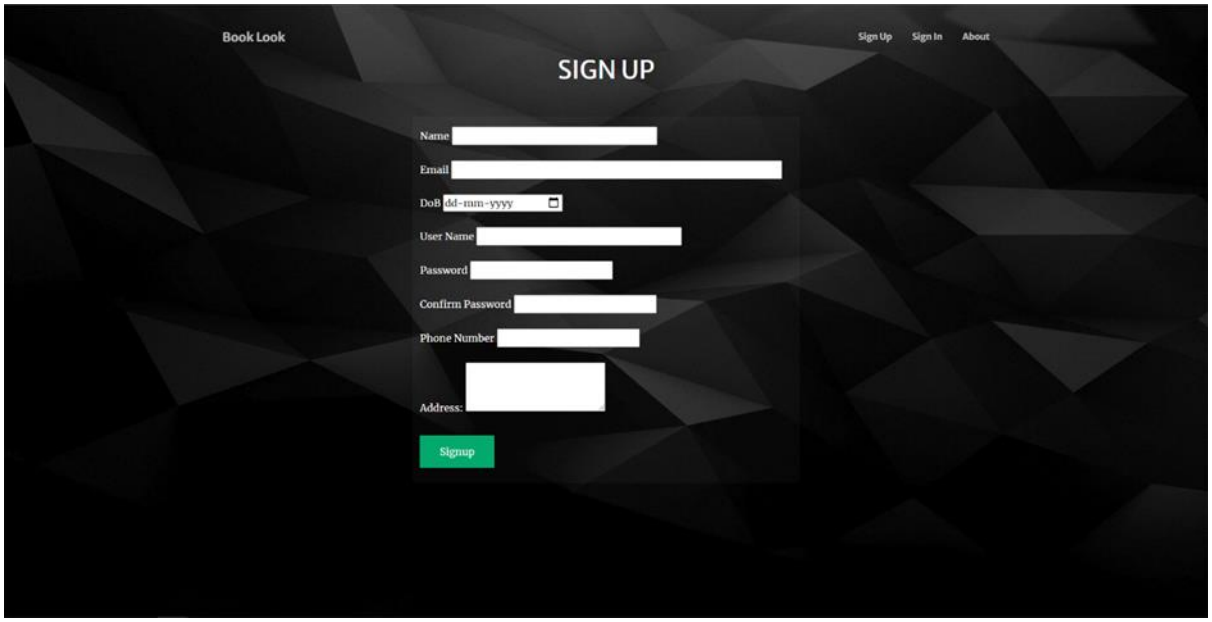


Description

It allows user to return books borrowed by them; it requires an OTP which is provided by the renter (owner of the book). Once the correct OTP is submitted, the book gets successfully returned to the renter.

6. Low Level Design

6.1 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

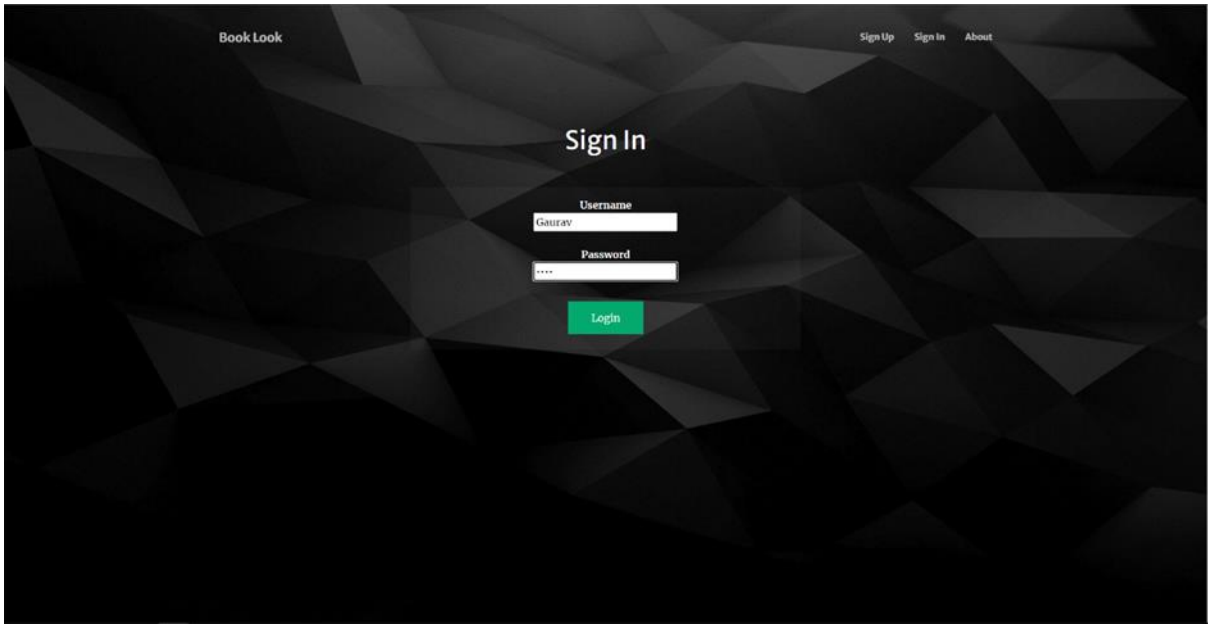
- Confirm Password (string)
  The confirm password input should match the password.

- 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.

6.2 SIGN-IN



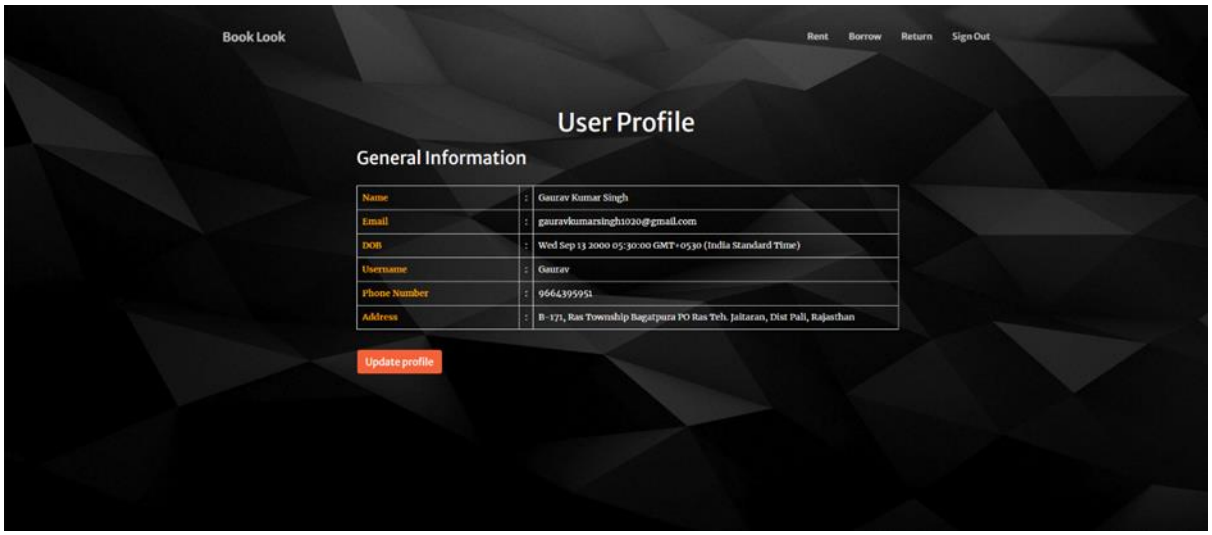
Sign-in page consists of a form which is filled by the user to access their account. The form consists of two fields:

- Username (string)
  The username can contain any character but it has to be unique.

- Password (string)
  The password can contain any character

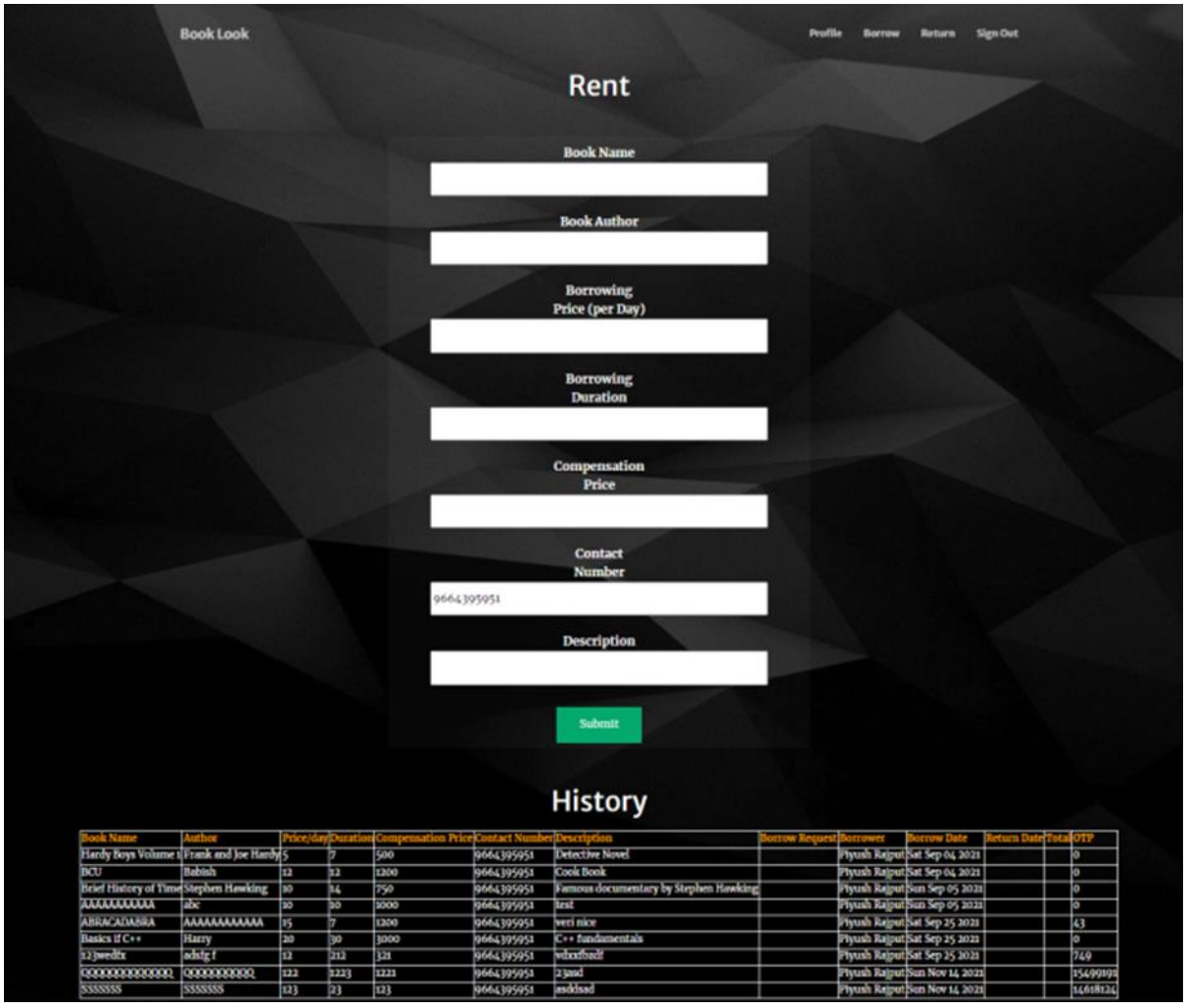
The user has to fill the form and then click on the signin button, after clicking the signing button the user credentials will be verified, if the user is registered in the system, a session will be created and they will be redirected to their profile page otherwise if the credentials are not in the system then they will be redirected to the signin page.

6.3 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.

6.4 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.

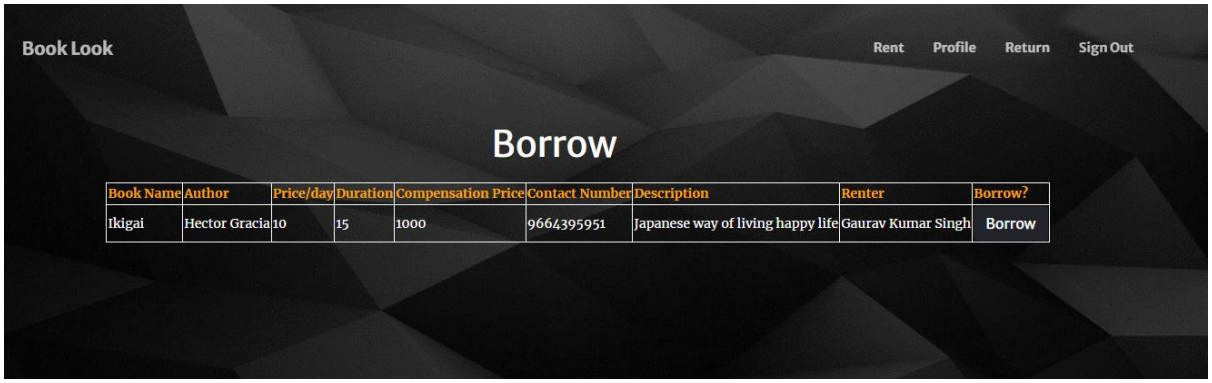
- Compensation Price (string)
  The compensation price should only contain numbers.

- 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).

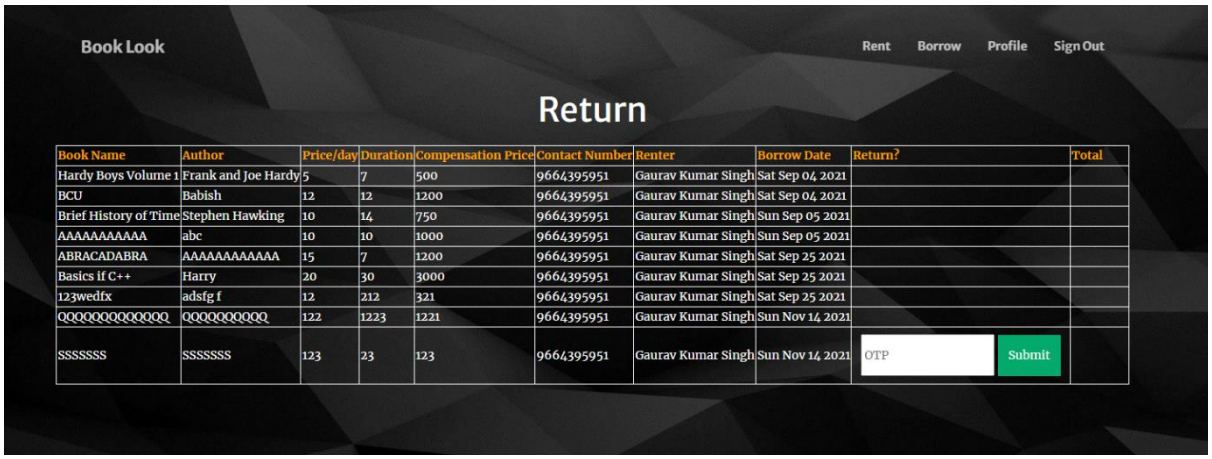
6.5 BORROW



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.

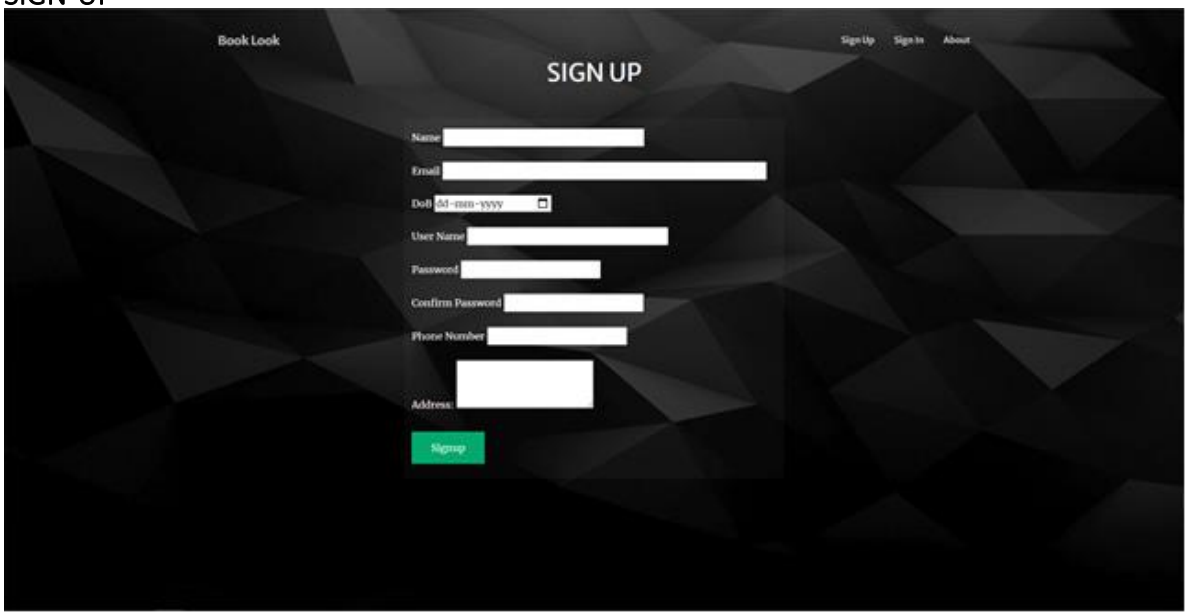
6.6 RETURN



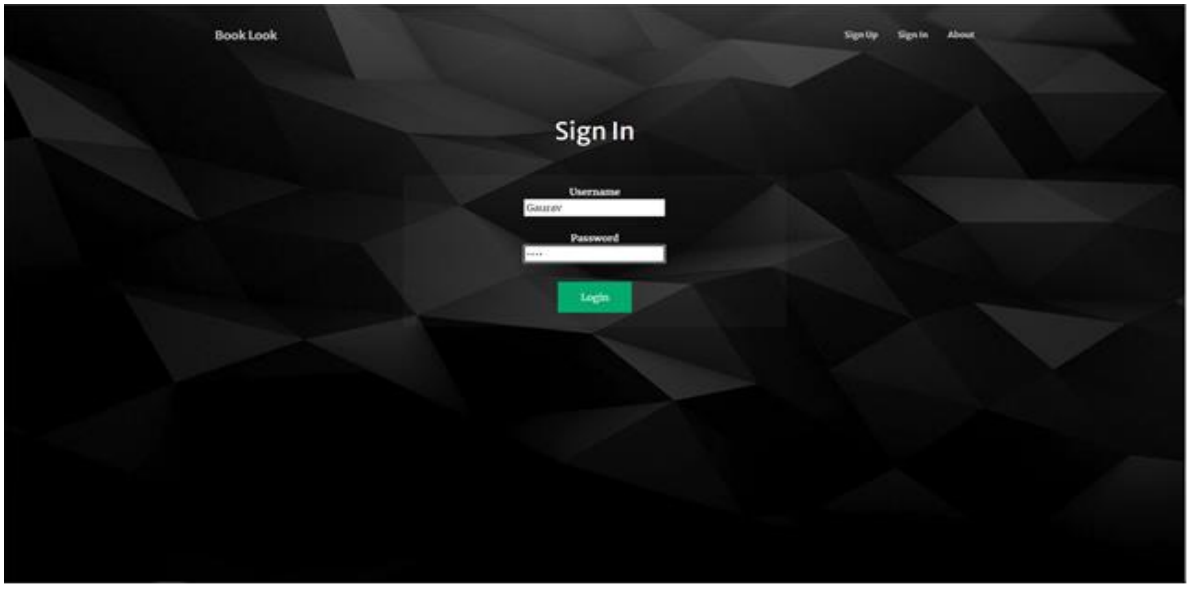
The return page contains the information of the books borrowed by the user, it also contains a section return which allows the user to return the book to its owner. To return the book the user has to enter the OTP given by the owner of the book. The book can only be returned after submitting the correct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

7. User Interface Design

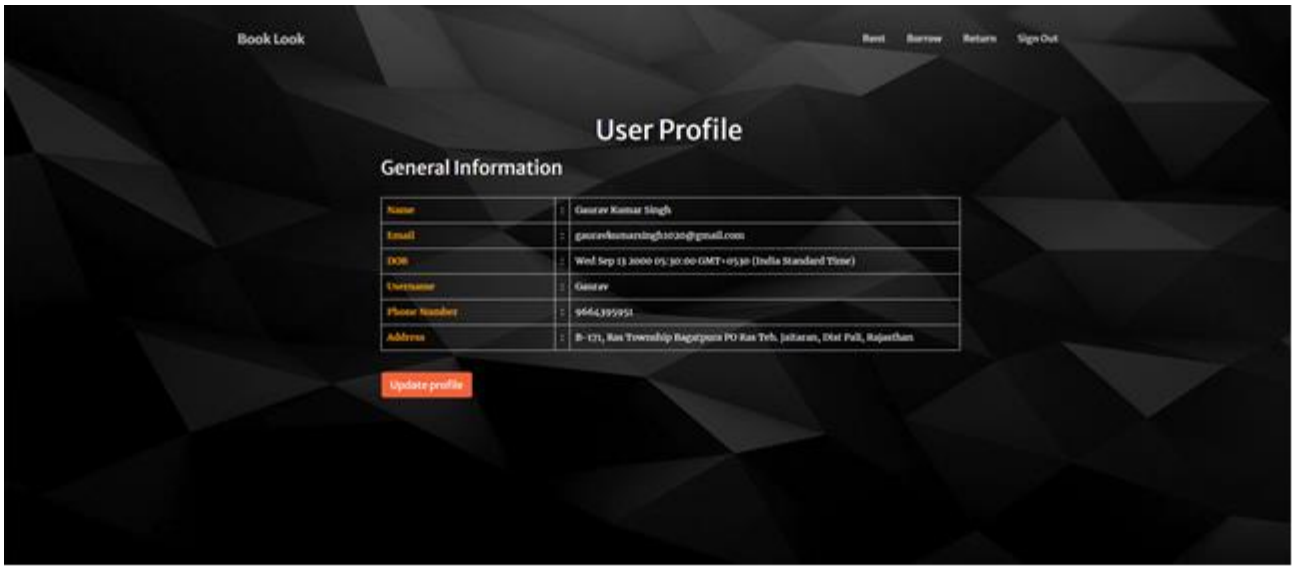
SIGN-UP



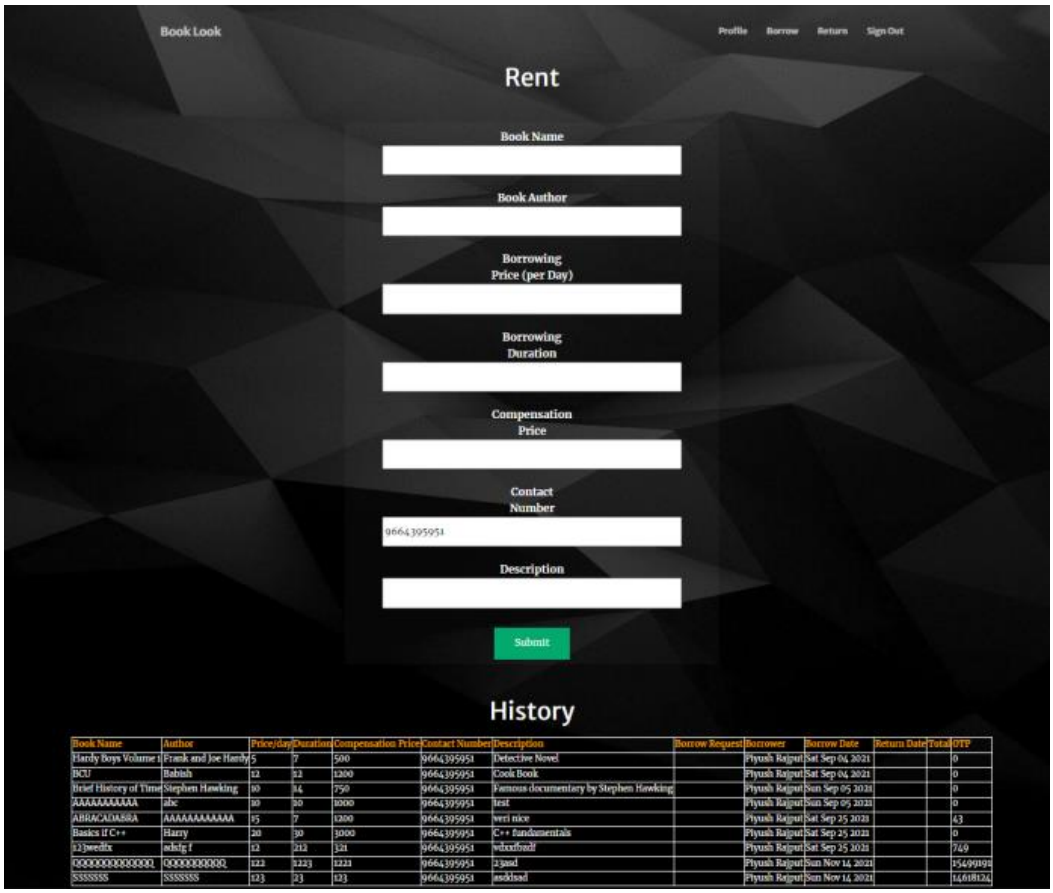
SIGN-IN



PROFILE

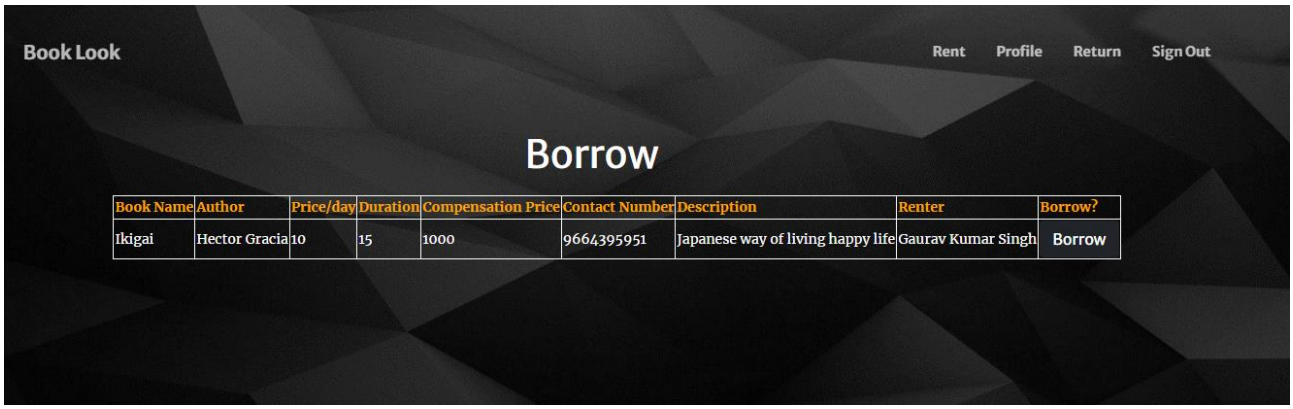


RENT

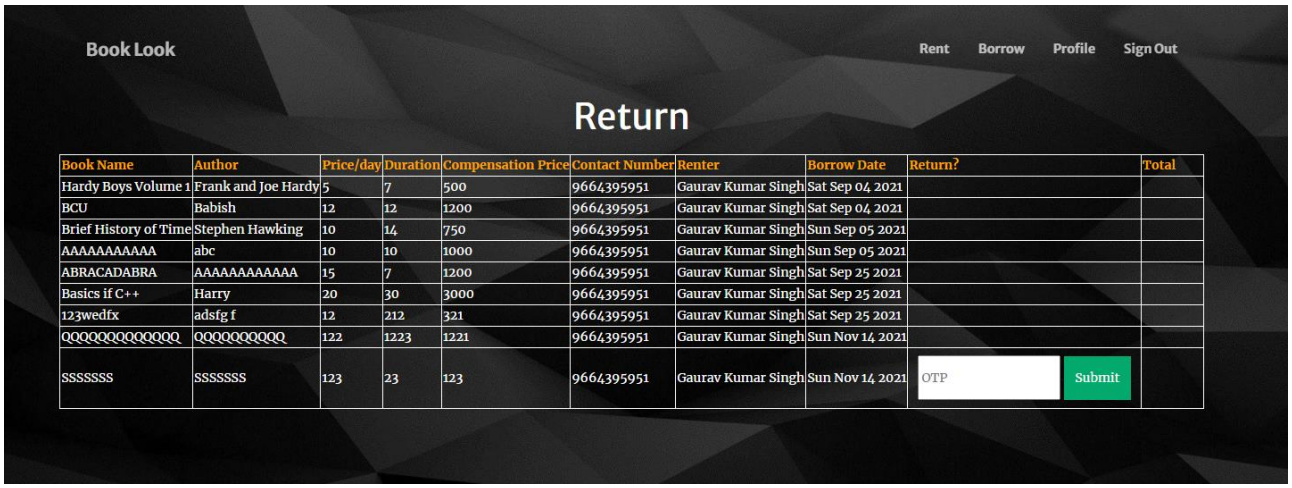


Book Name	Author	Price/day	Duration	Compensation Price	Contact Number	Description	Borrow Request	Borrower	Borrow Date	Return Date	Total	OTP
Hardy Boys Volume 1	Frank and Joe Hardy	5	7	500	9664395951	Detective Novel		Piyush Rajput	Sat Sep 04 2021			0
BCU	Babish	12	12	1200	9664395951	Cook Book		Piyush Rajput	Sat Sep 04 2021			0
Brief History of Time	Stephen Hawking	10	14	750	9664395951	Famous documentary by Stephen Hawking		Piyush Rajput	Sun Sep 05 2021			0
AAAAAAAAAA	abc	10	10	1000	9664395951	test		Piyush Rajput	Sun Sep 05 2021			0
ABRACADABRA	AAAAAAAAAA	15	7	1200	9664395951	veri nice		Piyush Rajput	Sat Sep 25 2021			43
Basics if C++	Harry	20	30	3000	9664395951	C++ fundamentals		Piyush Rajput	Sat Sep 25 2021			0
123wedfx	adsfg f	12	212	321	9664395951	wdxdfmdf		Piyush Rajput	Sat Sep 25 2021			749
QQQQQQQQQQQQQ	QQQQQQQQQQ	122	1223	1221	9664395951	23asd		Piyush Rajput	Sun Nov 14 2021			15499198
SSSSSSS	SSSSSSS	123	23	123	9664395951	asdasd		Piyush Rajput	Sun Nov 14 2021			14618124

BORROW

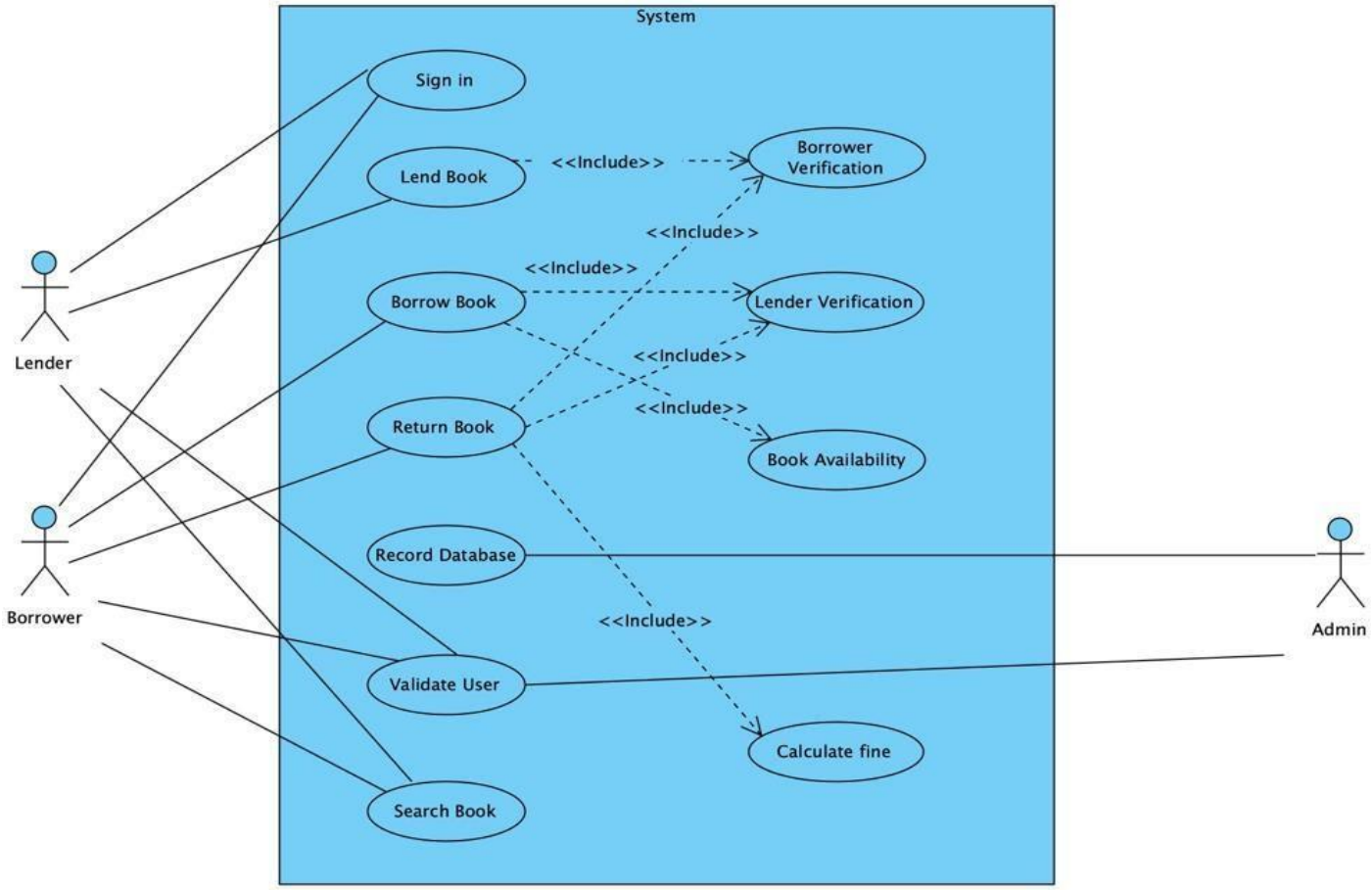


RETURN

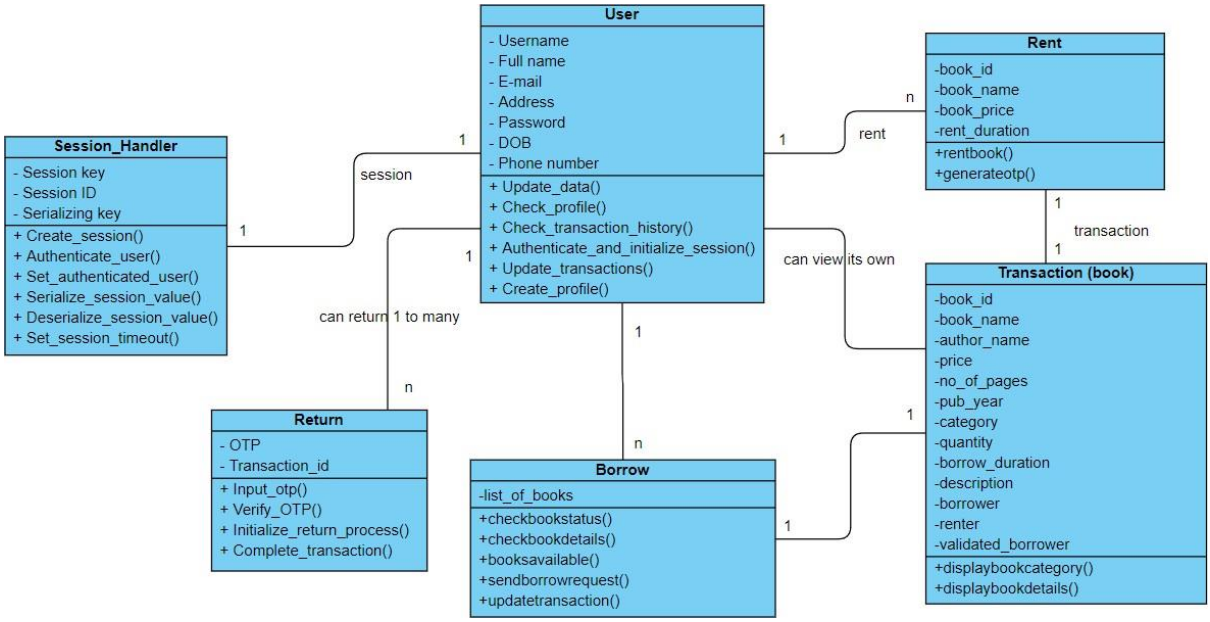


UML DIAGRAMS

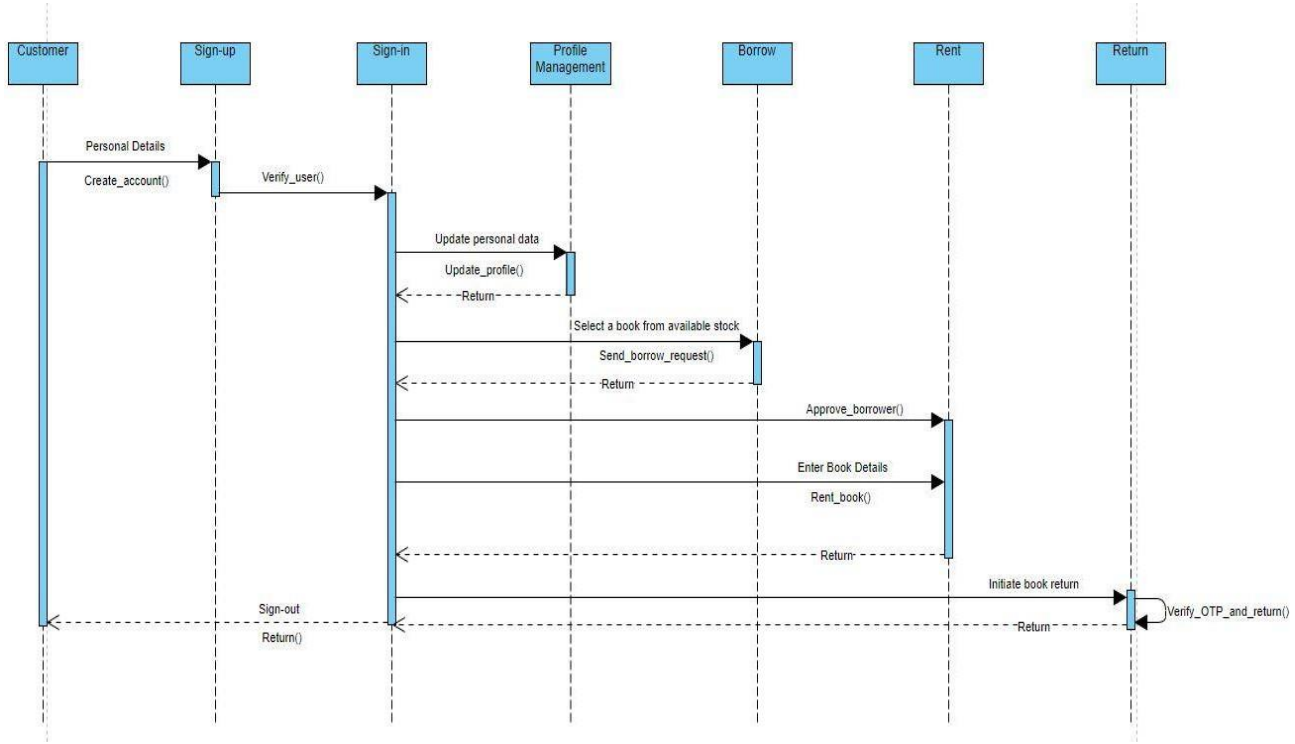
USE-CASE DIAGRAM



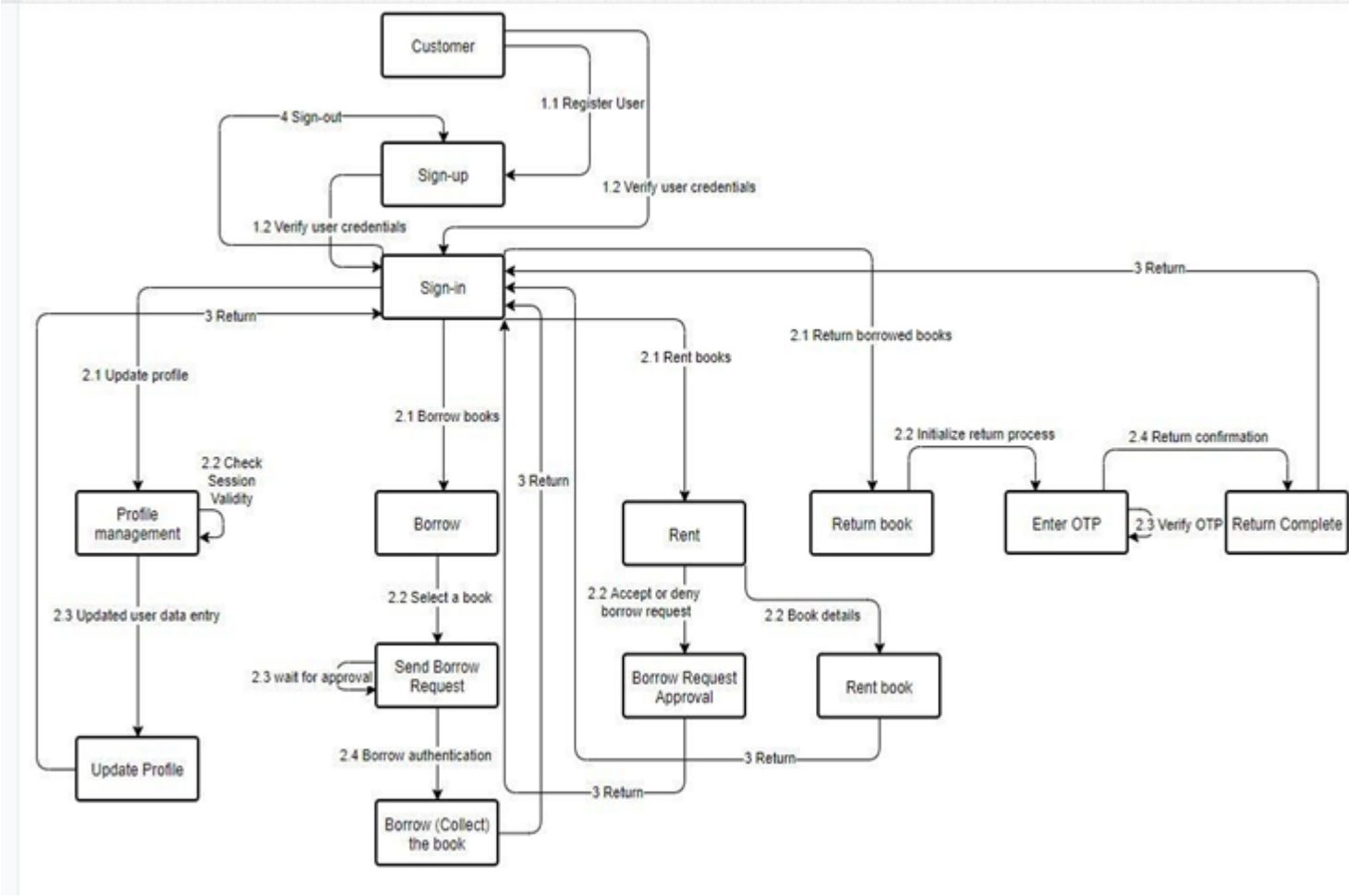
CLASS DIAGRAM



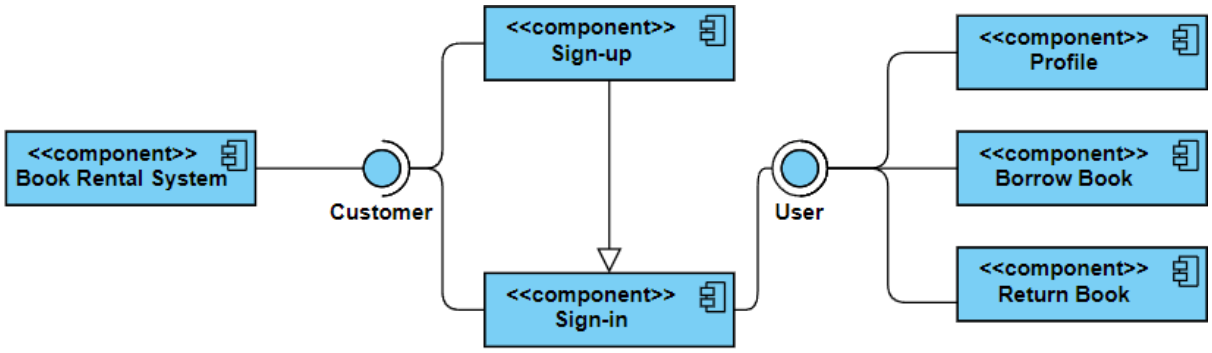
SEQUENCE DIAGRAM



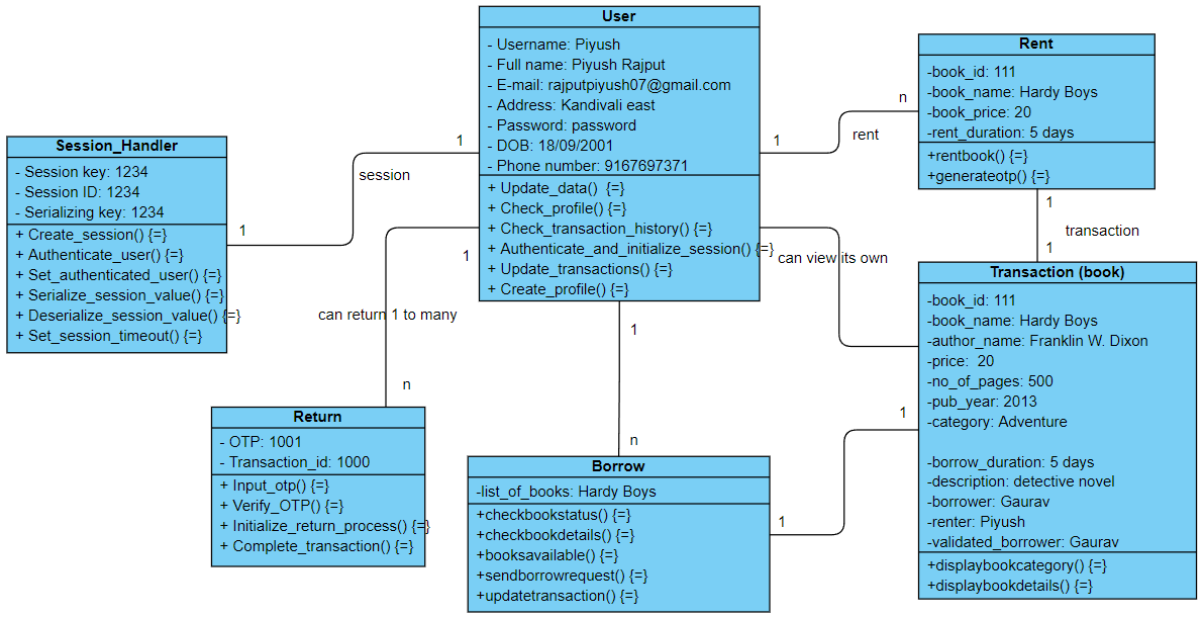
COLLABORATION DIAGRAM



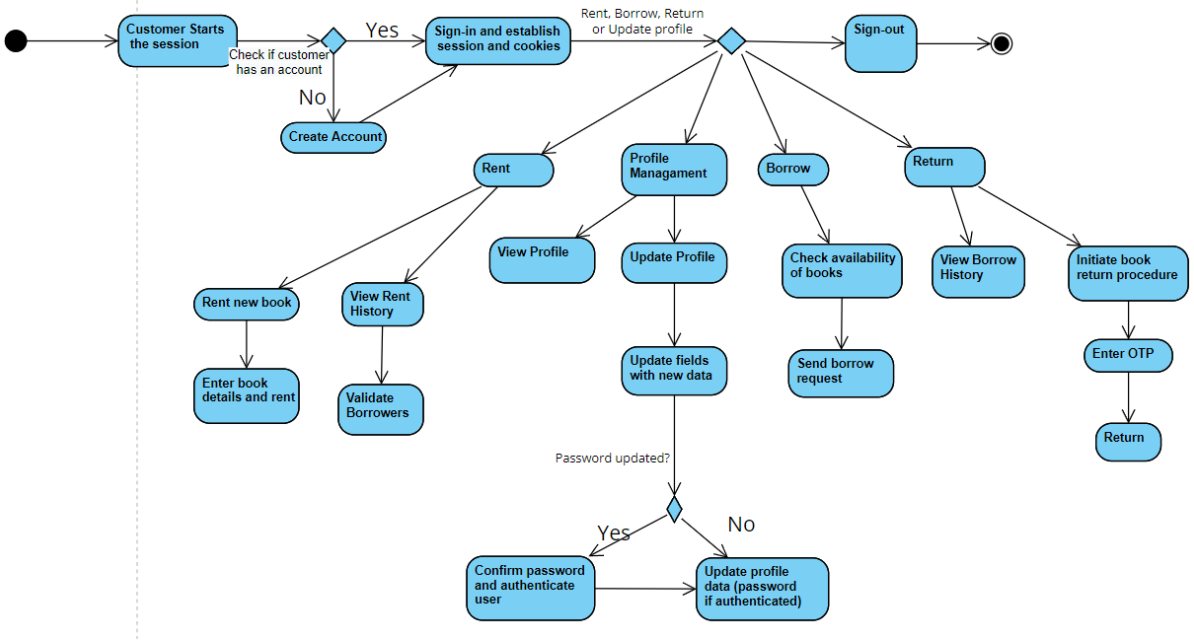
COMPONENT DIAGRAM



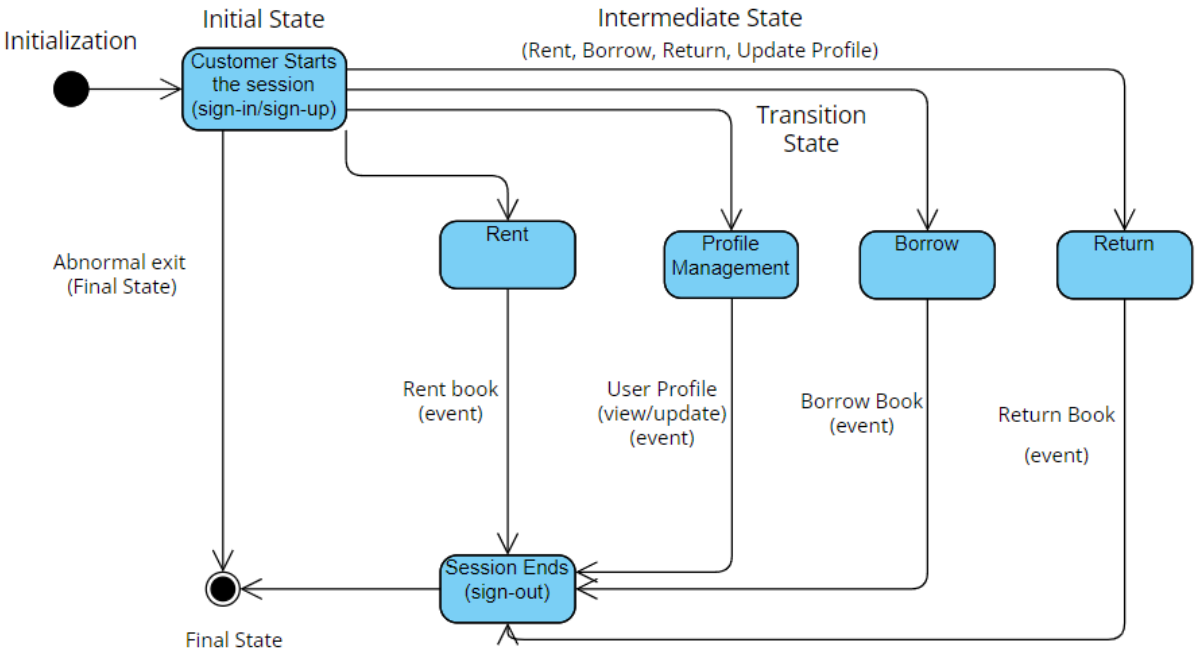
OBJECT DIAGRAM



ACTIVITY DIAGRAM



STATE DIAGRAM



DEPLOYMENT DIAGRAM