Software Requirements Specification

for

Book Renting Service

Version 1.0 approved

Prepared by

Piyush Rajput(19BCE0689)

Gaurav Kumar Singh(19BCE2119)

Ayush Gupta(19BCE2422)

Aryan Singh(19BCE2295)

Vellore Institute of Technology

22/09/2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The main objective of this document is to illustrate the requirements of the project Book Renting Service. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The document is developed after a number of consultations with the client and considering the complete requirement specifications of the given Project. The final product of team will be meeting the requirements of this document.

1.2 Document Conventions

	Font	**Style**	**Size**
Heading	Times New Roman	Bold	18
Sub-Heading	Times New Roman	Bold	14
Other's	Times New Roman	Regular	12

1.3 Intended Audience and Reading Suggestions

We are developing our project Book Renting Service for the college or any Institute. But this project can be deployed in any organization. This SRS is mainly developed for the project development team. In this team there are the project manager, developer, coder, tester and documentation writer and the user of the project also.

User (Customer)

This document is intended to user and customer to make them ensure that this document is well meeting the need of the users.

Project Manager

This SRS document is also very important for the project manager to ensure that can estimate the cost easily by referring to the SRS document and that it contains all the information require planning the project.

Project Developer

The project developer will refer to the SRS document to make sure that they developed exactly what the customer requires.

Tester

The tester will read this SRS document and he will ensure that the requirements are understandable from functionality point of view so that he can test the software and validate it's working.

Document Writer

The document writer is reading the SRS document is to ensure that they understand the document well enough to be able to write the users manuals.

Maintenance

The SRS document helps the maintenance engineers to understand functionality of the system, a clear knowledge of the functionality can help them to understand design and code.

Suggestions

1. The user can read the whole SRS document but for him Introduction, Overall description and System features is much required the software performance.
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.

1.4 Product Scope

The Software Requirements Specification captures all the requirements in a single document. The Book Renting Service that is to be 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 features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criteria for the system is based in the level up to which the features described in this document are implemented in the system.

1.5 References

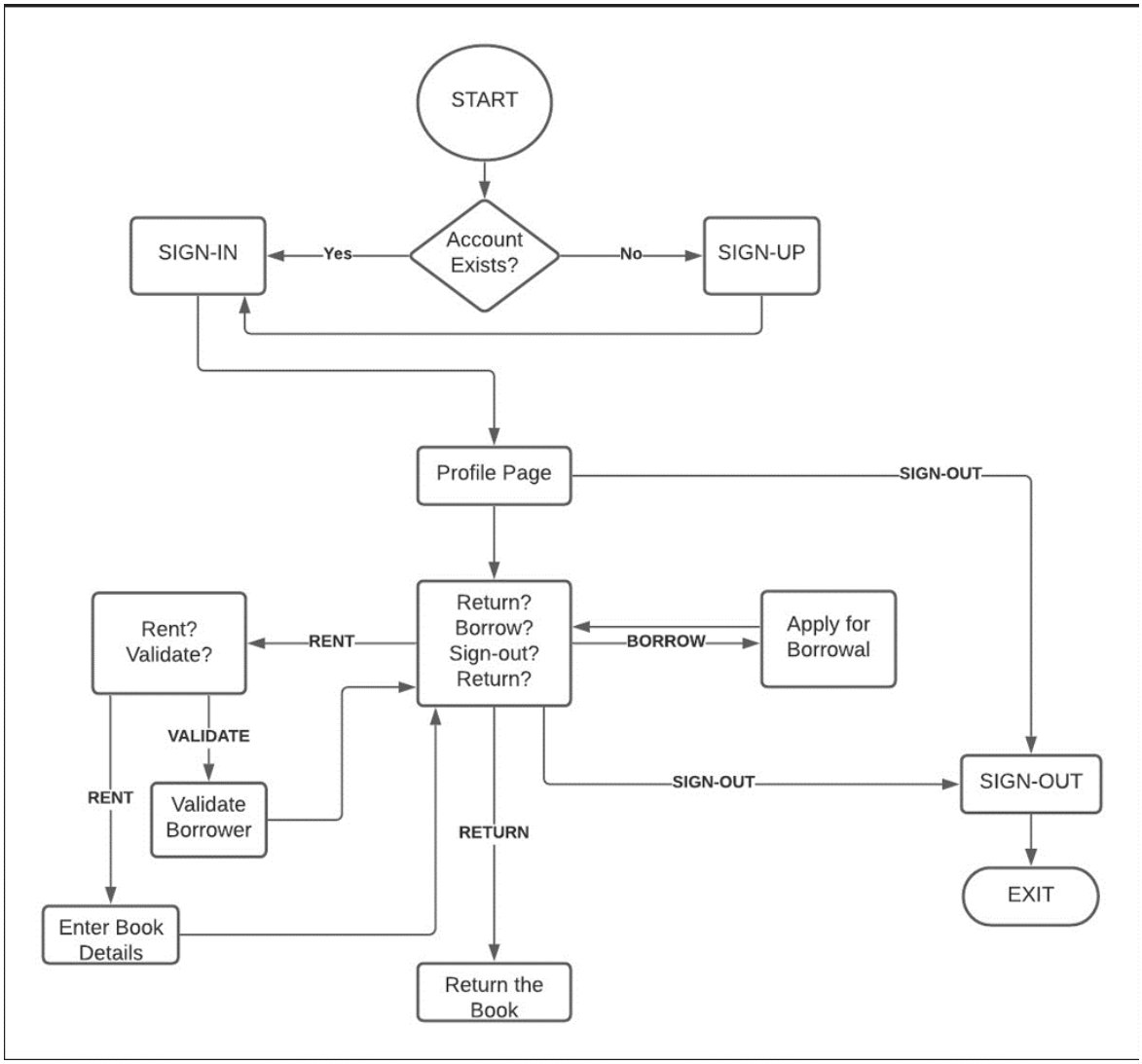
The application is based on HTML, CSS and JavaScript with JavaScript specific frameworks such as Node.js (version), Passport.js, Express.js and mongoose.js (MongoDB)

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

2. Overall Description

2.1 Product Perspective

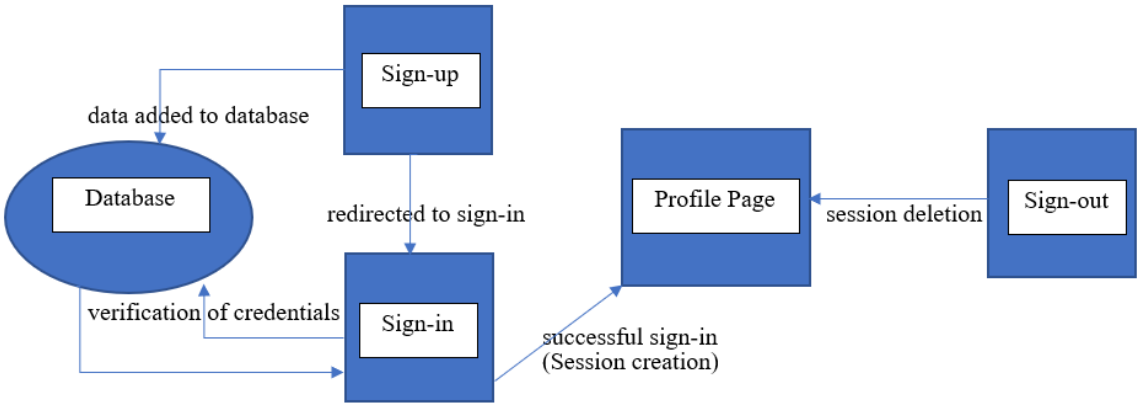
The application is an advanced version of library management system, but instead of the library managing its own resources, this application is more user centric in its functionality. It does not replace any existing infrastructure; it just adds to it as an additional feature. Both the borrower and vendor are normal users, not an organization, so it can function as both library management system and as peer-to-peer book rental service since organizations can also make use of this software.



2.2 Product Functions

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

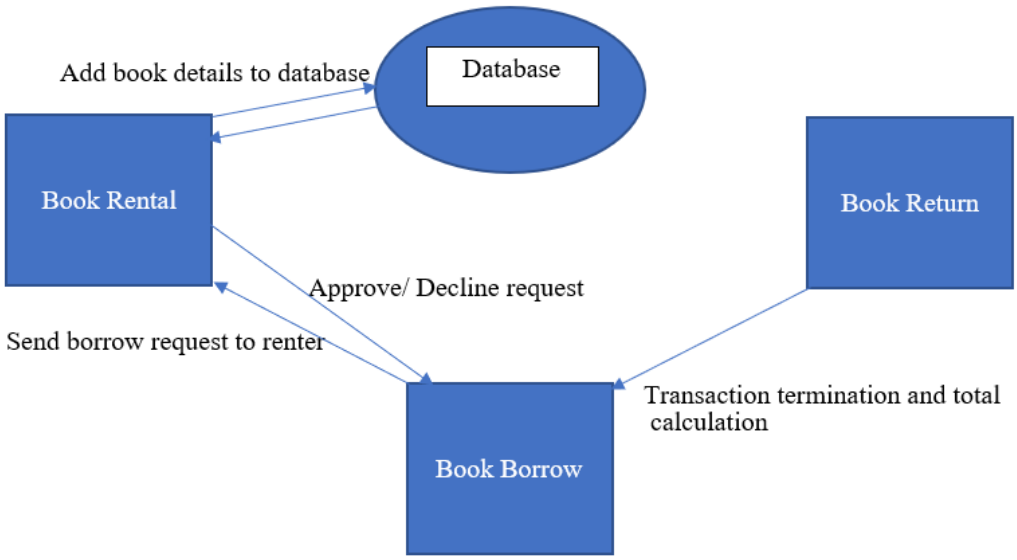


User profile management
The User Profile Management feature provides central management for user-specific data and settings stored in the users' model.

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.



2.3 User Classes and Characteristics

- Collectors: The people who like to read often and prefer owning the books they read. These people can put their books up for rent if they are not currently using the book and generate a passive income for themselves so they have the incentive to put their book up for borrowing.

- Borrowers: People who prefer to save money on buying books or renting books. These people just want access to the resources they want to read and does not necessarily want to buy the book because it can be expensive at times. These people can borrow books from the collectors and save money over buying the new book for themselves.

2.4 Operating Environment

For Hosting
- ✓ Operating Systems: Windows, Linux, MacOS
- ✓ Node.js, express.js, passport.js modules
- ✓ Internet connectivity

For Using Application
- ✓ Web Browser
- ✓ Operating Systems: Windows, Linux, MacOS
- ✓ Internet Connectivity

2.5 Design and Implementation Constraints

Cloud computation in case of deployment can be an issue since some cloud facilities do not facilitate collector collection for storage of data which means asynchronous functions might not work on some cloud deployment websites. The database can also grow big very quickly and anything except cloud storage will be difficult for our team to maintain and the consistency of the application on that scale could also be an issue for us. Otherwise, it is supposed to be a intuitive application for the users to use without much hassle.

2.6 User Documentation

Online user help section will be developed on the website to tell user about the functionalities of the application and how to use them. Advantages of using help files are:
- User friendly
- really fast
- easy to use
- larger amount of information
- easier to update
- can access from any device

The help section will contain step-by-step procedure of how-to sign-up to the website and use all the functionalities of the website with proper visualizations to help even a non-tech-savvy user to be able to use the website.
It will initially be divided in to the same number of modules as product functions to guide user through each step.
- Sign-in, Sign-up, Sign-out
- User profile management
- Book rental system
- Book borrower system
- Book return system

2.7 Assumptions and Dependencies

Assumptions are:

- The coding should be error for free.
- The system should be user-friendly so that it is easy to use for the users.
- The information of all users, and books must be stored in a database that is accessible by the website.

- The system should have more storage capacity and provide fast access to the database.
- The system is running 24 hours a day.

Dependencies are:

- The specific hardware and software due to which the product will run.
- On the basis of listing requirements and specification the project will be developed and run.
- The end users (admin) should have proper understanding of the product.
- The information of all the users must be stored in a database that is accessible by the Book Renting Service.

3. External Interface Requirements

3.1 User Interfaces

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as rent, borrow or viewing the details of the book.

- It allows the user to rent or borrow books.
- The user interface must be customizable by the administrator.
- All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined.
- The design should be simple and all the different interfaces should follow a standard template.
- The user interface should be able to interact with user management module and a part of the interface must be dedicated to the sign-in/sign-out module.

3.2 Hardware Interfaces

Processor: Pentium(R) Dual-core CPU

Hard Disk: 40GB

RAM: 256MB

3.3 Software Interfaces

This software package is developed using HTML and CSS as front end. Node.js as the back end.

Operating System: Windows 8 and higher versions.

Language: Node.js, HTML and CSS.

Database: MongoDB

3.4 Communications Interfaces

The communication architecture must follow the client-server model. Communication between client and server is HTTP based. The application itself can be hosted on cloud

services such as Heroku upon completion and anyone can access the application through there using HTTP protocols. The client-server protocol must be stateless. The requests and responses are managed through the backend using node.js and express.js frameworks and the data is managed in JSON format using MongoDB NoSQL database and mongoose.js.

4. System Features

4.1 Session Management

The priority ratings are rated from 1 to 9 (1 being very low and 9 being very high)

4.1.1 Description and Priority

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.
Feature Priority: 7
Benefit: 7

4.1.2 Stimulus/Response Sequences

If the user does not have an account registered, the user can Sign-up by filling the form with proper details. If the sign-up is successful, user will be redirected to the Sign-in page where user can create a new session for themselves by providing their username and correct password.
At any time after signing-in, the user can manually destroy the session cookie by simply signing-out to prevent other users from using you account and session to access the application.
All these features can be initialized via buttons on the top right of our application.

4.1.3 Functional Requirements

The session information (cookie) must be encrypted with user's id securely to ensure the security of the end users of the application. The user must be restricted from accessing sensitive pages without logging in to maintain integrity of the application and prevent frauds via spoofing. The application will be robust and handle invalid inputs from users accordingly taking fair margin of error from the user's end. In case of error during Sign-up or Sign-in user can be prompted of the input issues and redirected back to the home/sign-in/sign-up page. If the sign-out is unsuccessful, it can be easily noticed by the continued presence of the option to sign-out.

REQ-1: Session-key encryption
REQ-2: Page-Access restriction structure
REQ-3: Session management (creation, deletion, modification, expiry)
REQ-4: Interactive and Intuitive GUI

4.2 Profile Management

The priority ratings are rated from 1 to 9 (1 being very low and 9 being very high)

4.2.1 Description and Priority

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.
Feature Priority: 8
Benefit: 9

4.2.2 Stimulus/Response Sequences

After signing in the first page user encounters is profile page where the user can see all their details. By clicking on profile update button, a user can update their profile and save it. On the rental, borrowal and return pages, the user can click on usernames of other users concerned with a specific transaction to view their profile to judge someone's reliability.

4.2.3 Functional Requirements

The profile page must be locked for non-logged-in users or if the session cookie expires. Other users must not be able to modify any other persons' personal details and the profile modification must be linked to the logged in user's user id.

REQ-1: Page Access Restriction for non-logged-in users.
REQ-2: Personal Profile management and modification
REQ-3: Foreign Profile View
REQ-4: Interactive and Intuitive GUI

4.3 Book Exchange Management

The priority ratings are rated from 1 to 9 (1 being very low and 9 being very high)

4.3.1 Description and Priority

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.

Feature Priority: 9
Benefit: 9

4.3.2 Stimulus/Response Sequences

In the book rental system, 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.
In the borrowal system, the restriction is that the user cannot borrow the book which he made available to borrow.

In the return section, for the transaction to be safely finalized, there will be 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.

4.3.3 Functional Requirements

The borrower has to enter the OTP to complete the transaction (return the book) which will be provided by the vendor. 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.

REQ-1: Page Access Restriction for non-logged-in users
REQ-2: Renter Validates the borrower
REQ-3: User cannot borrow the book which he made available to borrow.
REQ-4: OTP for transaction to be safely finalized

5. Other Nonfunctional Requirements

5.1 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. The sign-in procedure must be fast and intuitive and transactions must be optimized to be secure.

5.2 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.

5.3 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. The application must work on most of the web browsers and the user interface must be intuitive so that even non-tech savvy users can understand the standard work flow of the application.

5.4 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. Development team can change some functionalities and add or remove functionalities to cater the needs of the clients at any time of the applications life.

6. Other Requirements

There are no constraints on the database. It could be SQL or NoSQL database. The application is designed for small area deployment(local) but the application can be scaled later to cater larger

audience across larger areas. In case of direct payment option integration requirement later, some legal requirements have to be satisfied.

Appendix A: Glossary

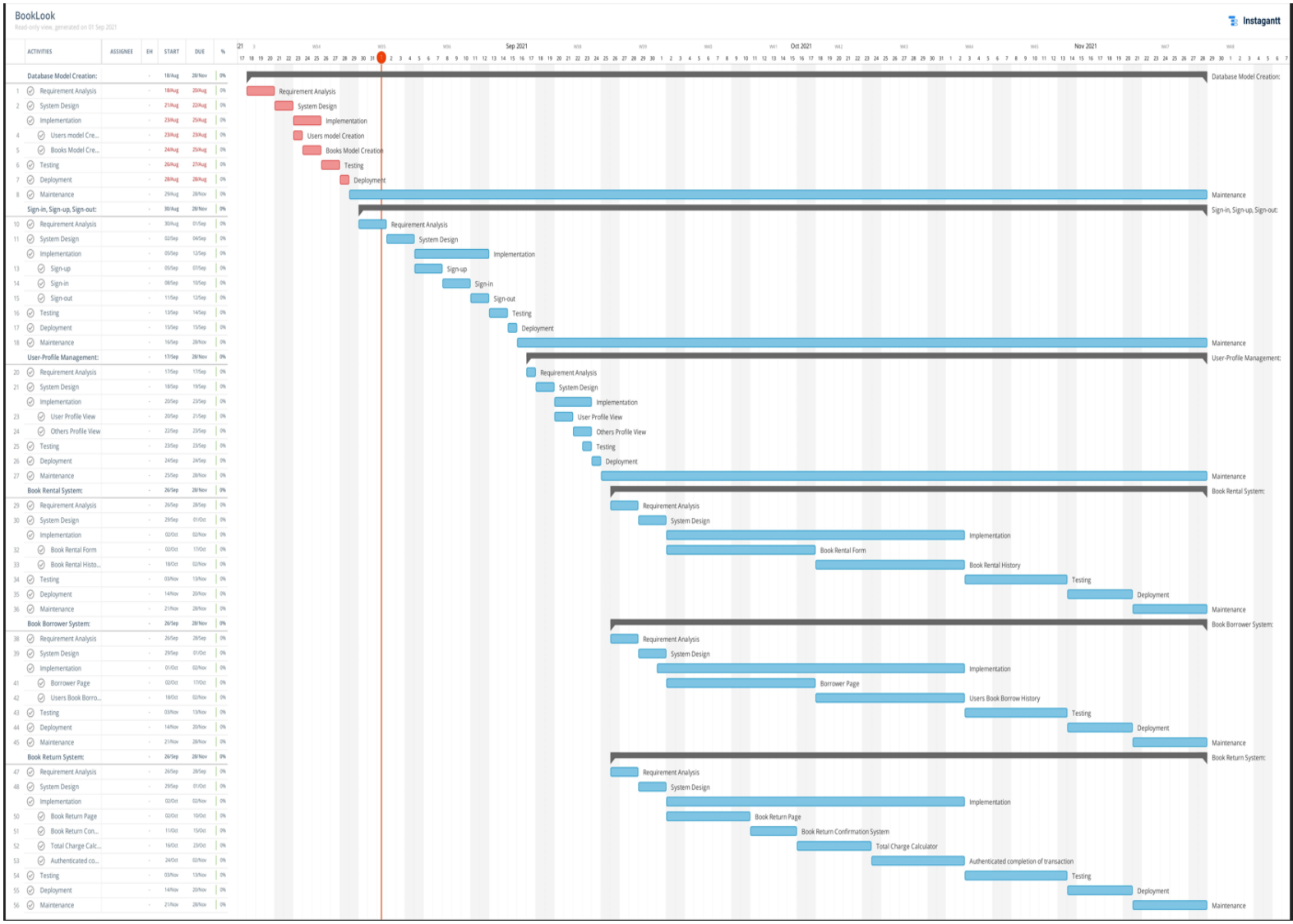
We use the term module in the project to depict all the independent functional sectors of the project. The modules thereby mentioned can be developed separately and integrated into the project whenever they are ready.
^ in the document signifies version of software used.

Appendix B: Analysis Models

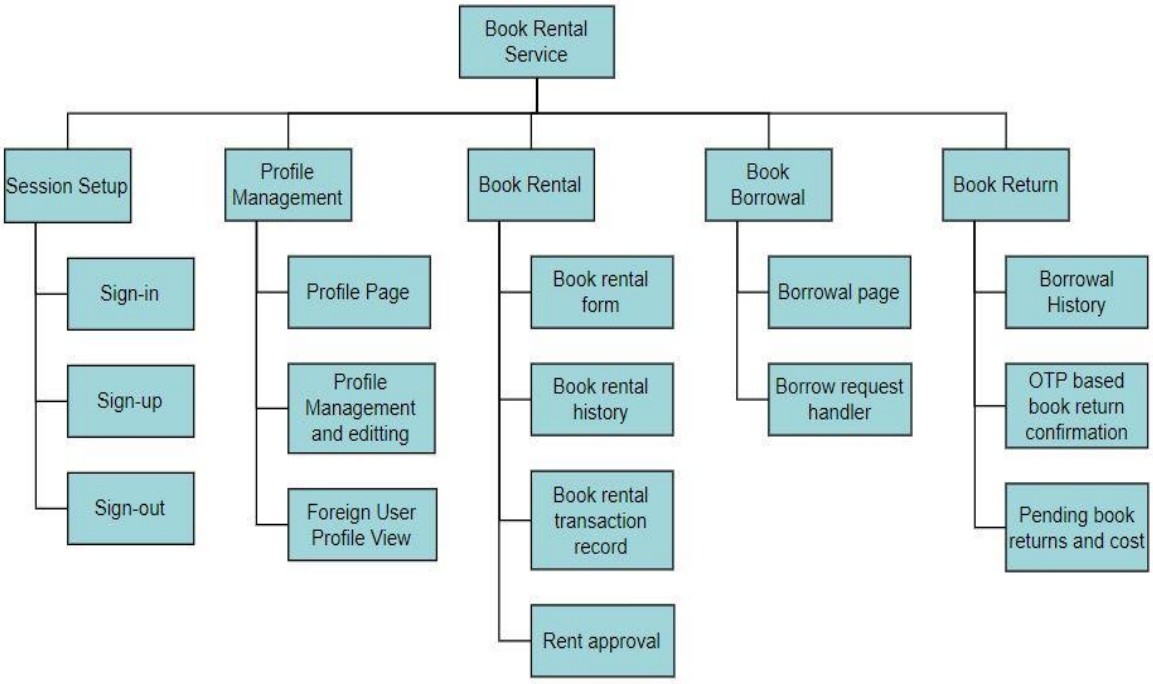
PROJECT'S SDLC

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	--- --- --- --- --- ---



Deliverable Based WBS



Phase Based WBS