EAD

PROJECT

Final Report

Bookshelf

EAD

Group 139

TEAM

S B KOUSHIK

(S20170010131,koushik.s17@iiits.in)

P HEMANTH VARMA

(S20170010108,hemanthvarma.p17@iiits.in)

MOHAMMAD FARAZUDDIN

(S20170010097,farazuddin.m17@iiits.in)

Y NARENDRA REDDY

(S20170010186,narendrreddy.y17@iiits.in)

PROJECT SUMMARY

1. Project Overview

Bookrest is a **web platform** where users can upload their bookshelf and others can see and request books to read.

It also provides the users to **exchange** books with other users.

On our platform, users can get access to read some **rare** books that others have.

Users can keep their book for online auction.

2.Tech Stack

- Nodejs
- MongoDB Atlas
- Html, Css, Bootstrap, Javascript
- Python

3.Use Case Diagram

4.Modules

Common work(not a module)

Homepage

User profile management

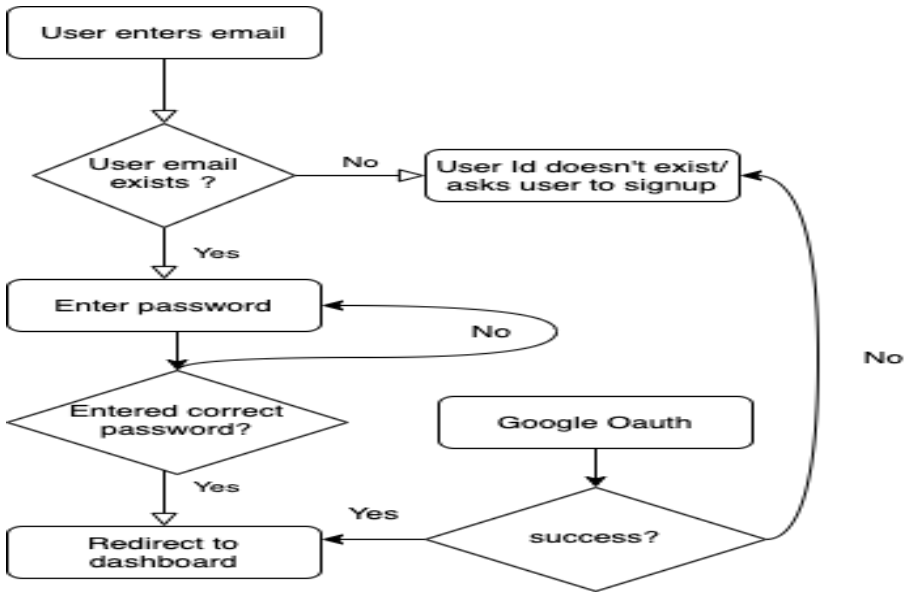
- Login - session management
- Registration,verification and validation
- Profile page

Status: Completed

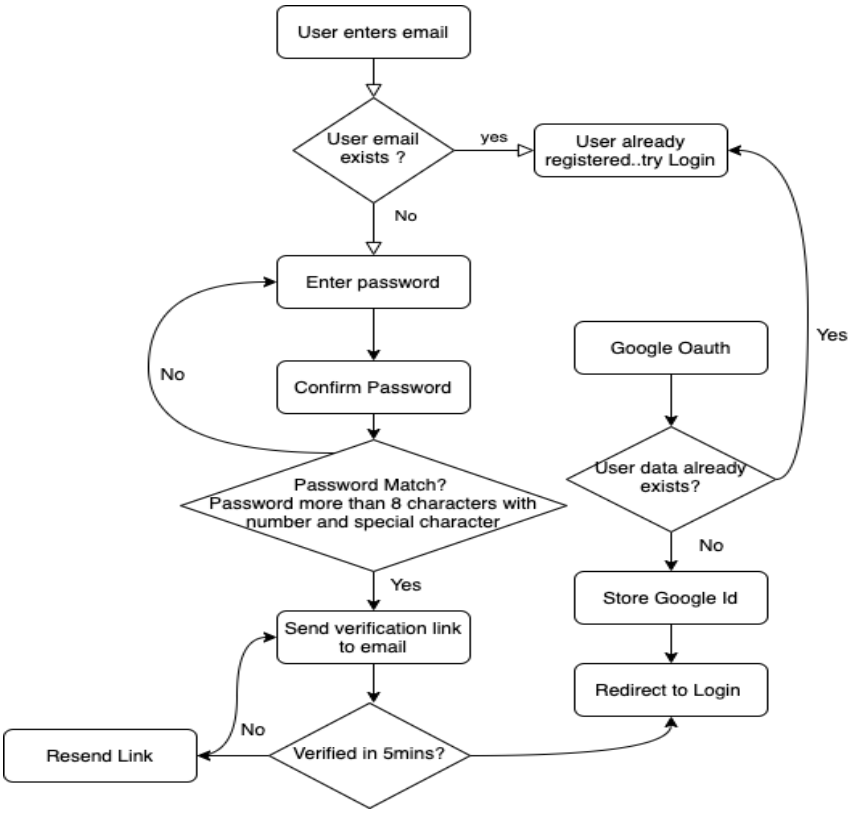
Users can register using their email id and complete email verification to create the account. They can also use google authentication to login to the bookshelf. They will be redirected to a profile page to add some more personal details. They can use forgot password to reset the password. Logged in users will be redirected to homepage where they can see recommended books and other users shelf and have access to many other facilities provided by bookshelf.com

User Authentication Workflow

Login workflow



Signup workflow



Api's developed

- Api to check whether the User Id already exists or not.
- Signup api to store data in db.
- Login api for user credentials verification.

Module 1

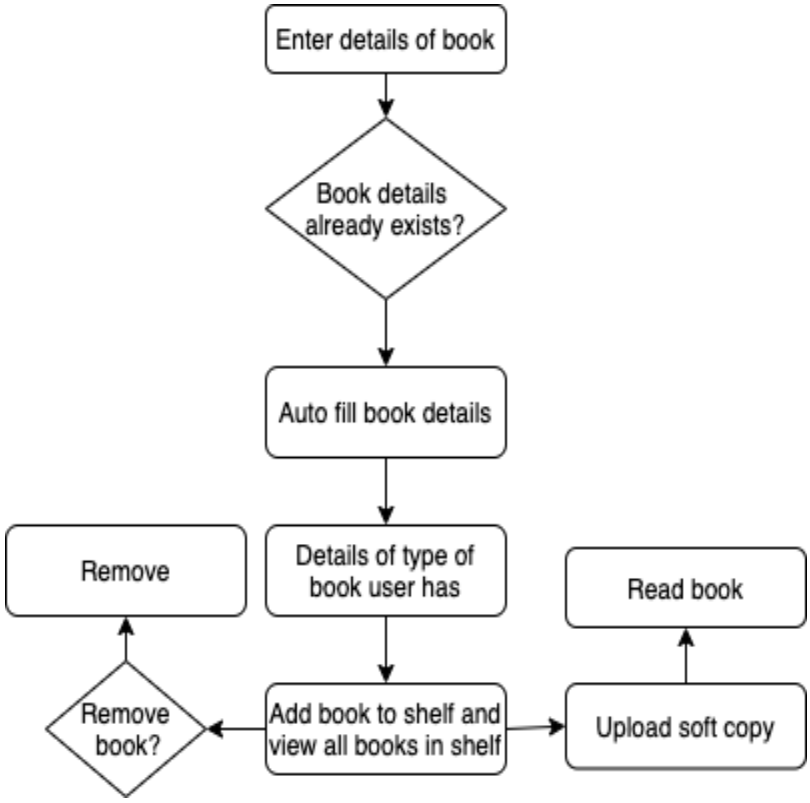
Creating and managing your bookshelf

- Add, edit and delete books from bookshelf
- Uploading a softcopy of a book into aws bucket.
- Book viewer(Users will be restricted to download the book)

Status: Completed

Users can add the books they have into their shelf where other users can see those books. Users can keep their books for auction If they have a softcopy of the book then they can upload it and read using the bookshelf viewer.

Bookshelf workflow



Api's developed

- Api to add books to the shelf .
- display existing books in shelf
- Api to auto fill the details of the book when adding it to the shelf
- remove books from the shelf
- Edit book details

Module 2

Categorizing books based on rarity,popularity and common.

Searching books

- Searching books based on author name,book name.
- Can see book shelves of other users

User to user chat.

Status: Completed

In the book details page, Users will be asked to give feedback about the book as common, popular or rare, Each book is categorised as common, popular or rare based on the count of number of shelves in which that book is present and user feedback.

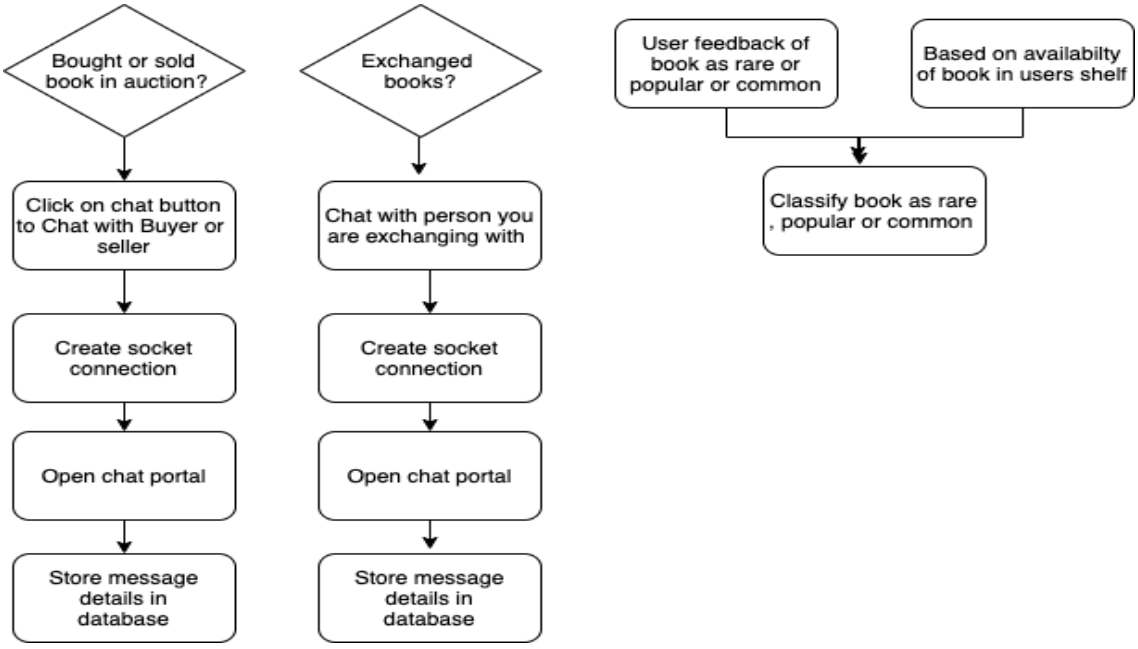
Search books based on book name and book author and retrieve data based on rankings. We used minisearch module for vectorizations.

Users will be able to see the bookshelf of other users in whose shelf that particular book is present.

When a user buys a book or sells a book in auction then a chat portal will be enabled between the buyer and seller where they can chat with each other and get further details of book delivery.

When a user raises or accepts an exchange request then a chat portal will be enabled between both the users who want to exchange books where they can chat with each other and get further details of book delivery.

User to user chat Workflow



Api's developed

- Api to display past user messages.
- Api to display store messages in database

Module 3

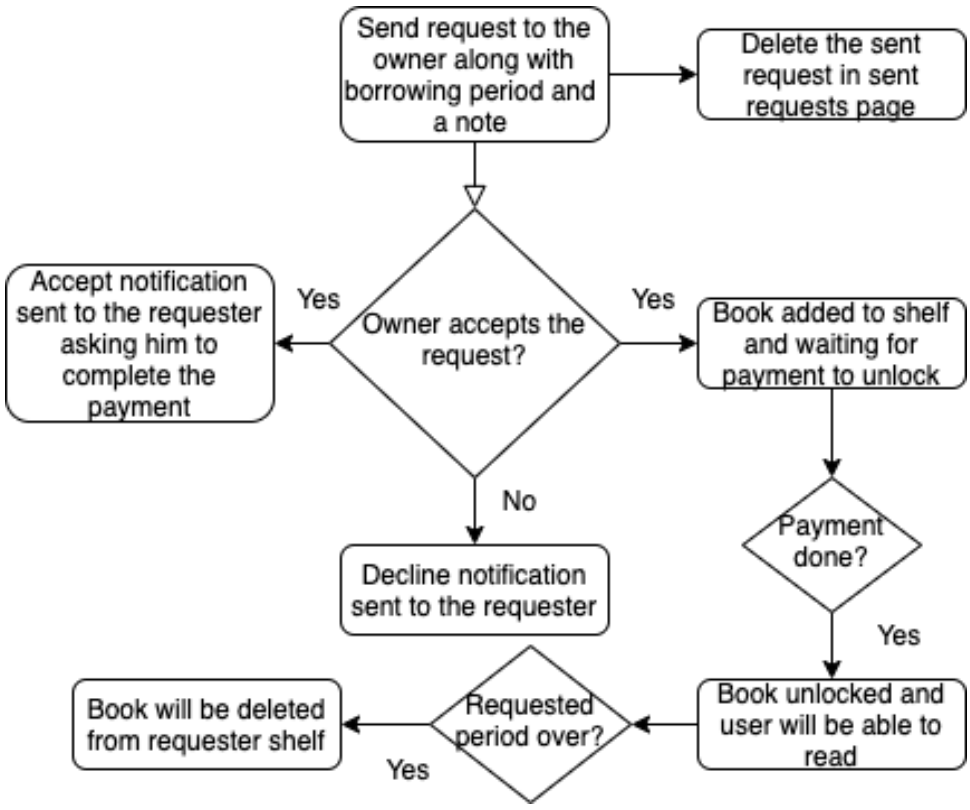
Requesting books of other users

- Users can send a request to the book owner
- Can get access to the book for the desired period of time
- And after that he will lose access to that book.

Status: Completed

The users who have a soft copy of the book will be displayed in the book details page so that the users who would like to read the book can send a read request to the owner so that he will get access to the book for the desired period of time to which he has to pay for the user.

Book Lend Request Workflow



Api's developed

- Api to display users who have a soft copy of the book you want to read.
- Send a loan request to the owner along with a period of borrowing.
- Api to accept or reject the lend requests and update shelf accordingly.
- Api to display sent requests data.
- Api to display old requests.

Module 4

Pay to read others books

- Pay for reading the rare book depending on time you had

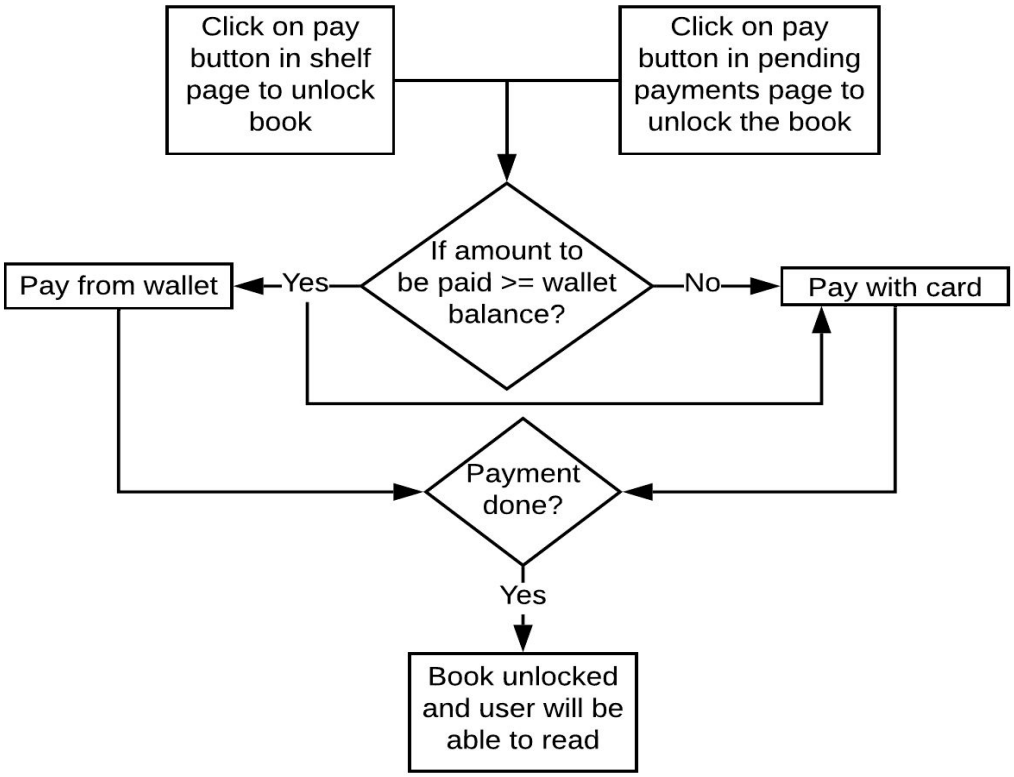
access to it
- payment gateway using stripe.js
- Payments page

Status: Completed

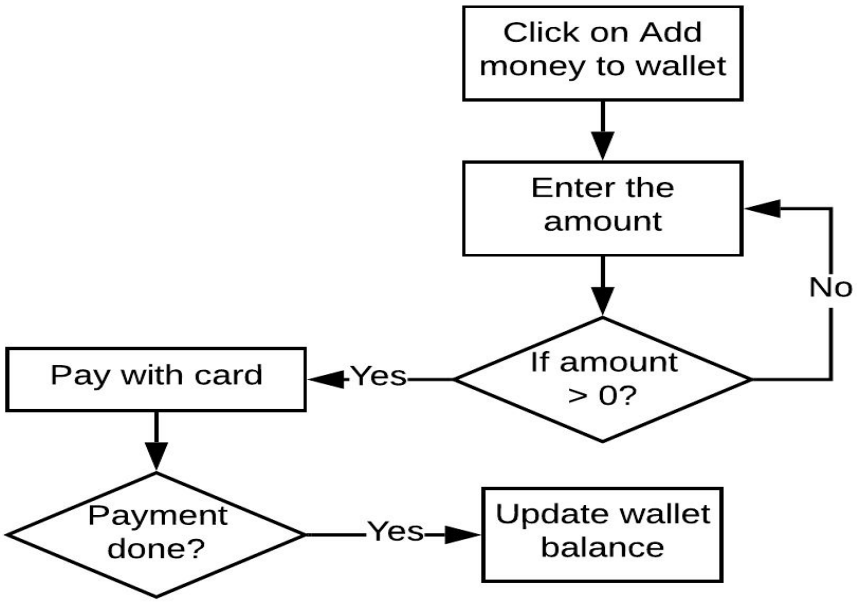
Payment gateway is developed using stripe js module.

Users can pay to the other users from their debit card or the wallet balance.Users can add money to their wallet using this gateway. All the pending payments will be displayed in the pending payments page and all the old transactions can be seen on the transaction history page.

Pay to Unlock books Workflow



Add Money to Wallet Workflow



Api's developed

- Api to finish payment for the borrowed books.
- Unlock borrowed book after payment
- Display pending payments
- Display Payment history
- Add money to wallet

Module 5

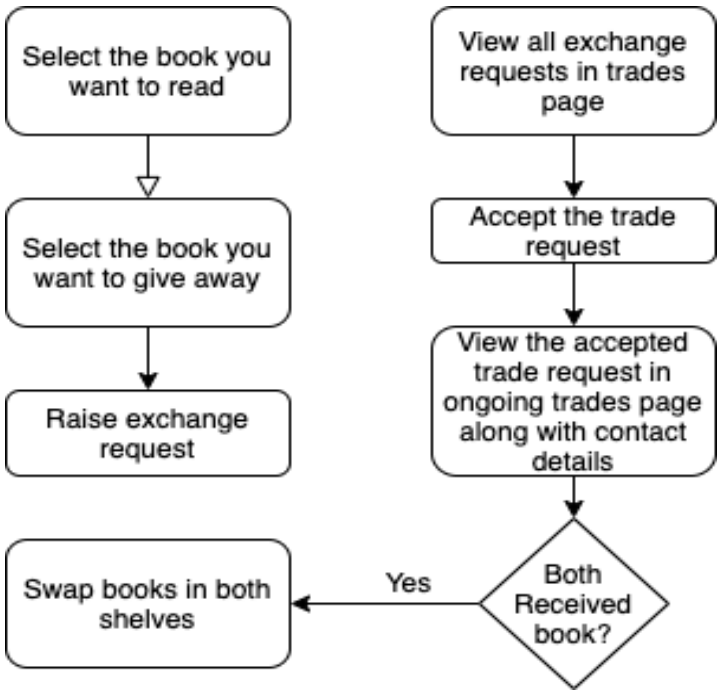
Trading books

- Users can keep a trade request for books he has on his shelf with the book he wishes to read.
- Another user who has that book and wishes to trade can accept the request so that they can exchange books

Status: Completed

User can select the book he wants to read and select one from his shelf to raise an exchange request so that other users can see the request and accept if they are interested. All the accepted requests can be seen in the ongoing trade requests page. User is supposed to click the book received button when he/she receives the book.

Book exchange Workflow



Api's developed

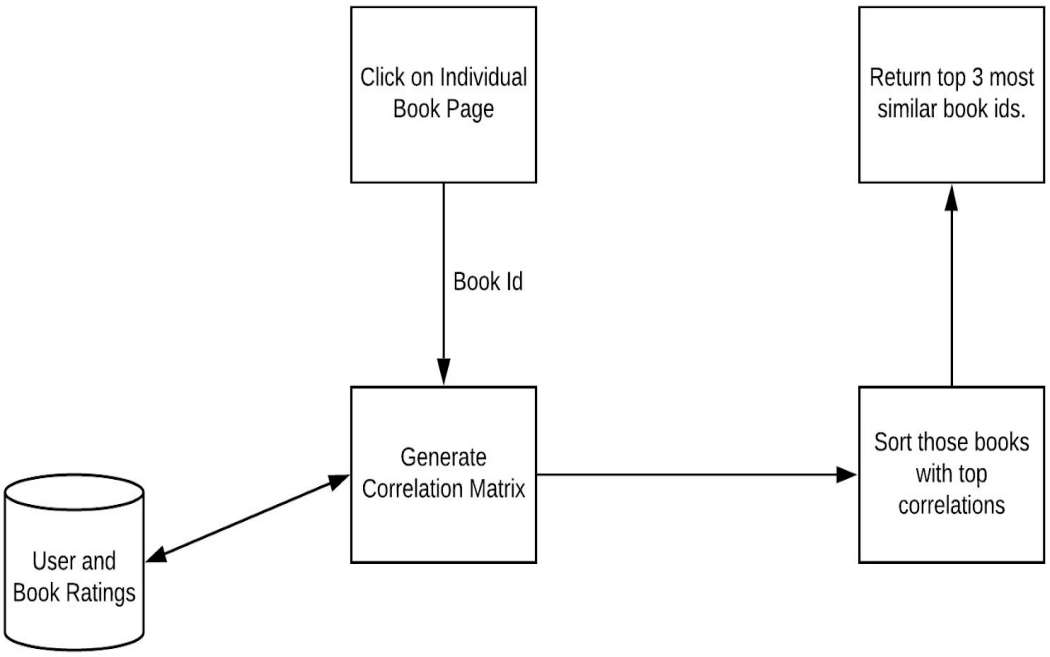
- Api to raise an exchange request for the book you want to read.
- Api to select a book from your shelf for the exchange request you raised.
- Display all exchange requests
- Filter exchange requests based on genre and location
- Api to display ongoing exchanges
- Api to update exchanges request as book received when user receives the book
- Api to exchange books on their shelf when both users received their books.

Module 6

Recommendation system(type 1) RS1

- We will cluster similar kinds of books based on its features.
- Users when searching for some book get recommendations of similar books from the cluster to which that book belongs to.

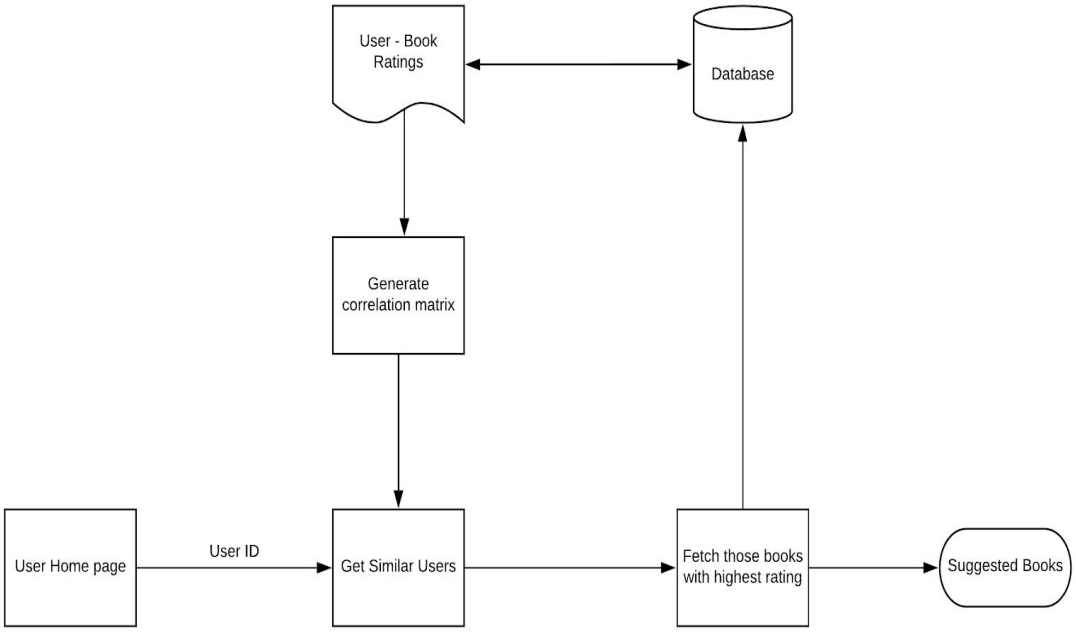
Workflow for RS1



Recommendation system(type 2) RS2

- We will collect the users interest and group all users.
- Users belonging to same group will be considered as similar users
- When a user gives a positive feedback to a book then that book will be recommended to all his/her similar users .

Workflow for RS2



Status: Completed

In the book details page, users will be able to see books similar to the one they are currently watching. We will cluster similar kinds of books based on its features. In the user dashboard, users will be able to see top pick books based on similar users which are classified by clustering the users based on the genres which are interesting for those users.

Api's developed

- API for generating inverted index of user ratings and book id's.
- API for getting similar books to a particular book using the pearson correlation.
- API for getting personalized recommendations based on similar user preferences.

Module 7

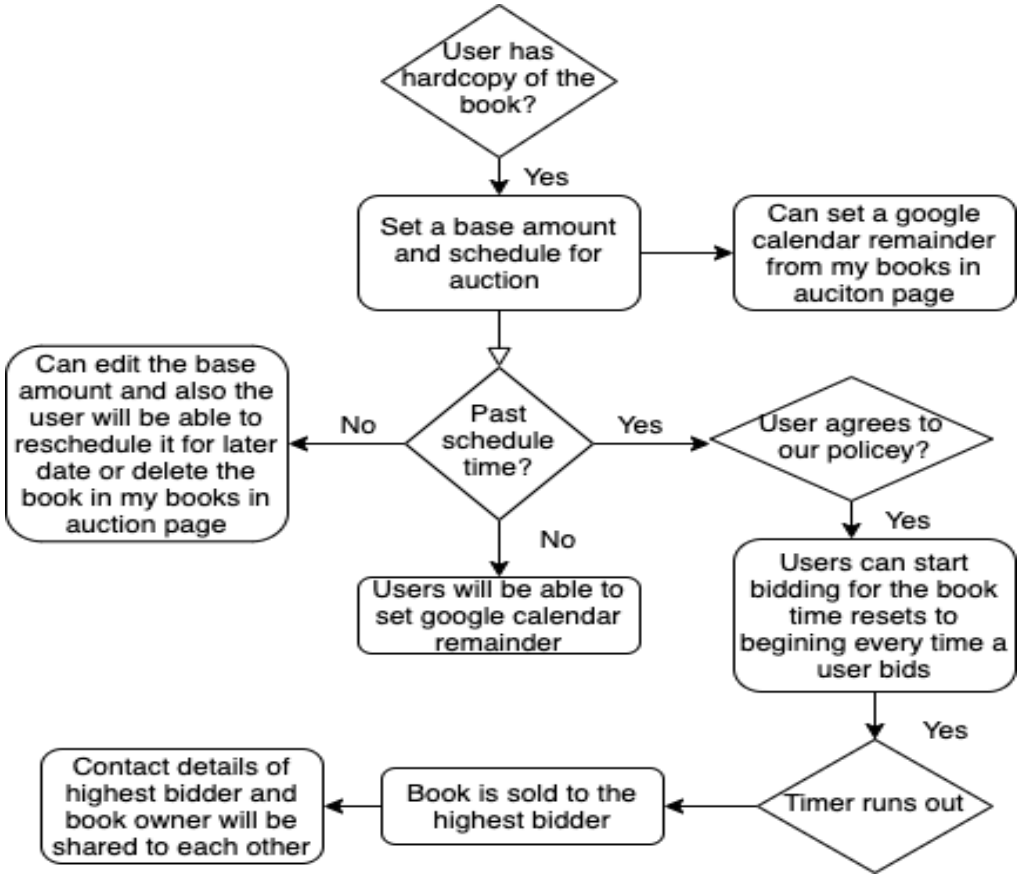
Online bidding for the rare books

- User can open bidding for his book
- Others can give their bids and buy the book

Status: Completed

If the user has the hardcopy of the book then he/she can put the book for auction. Other users can see the books put in auction on all books page in which the user can set google calendar reminders. Users can also edit the scheduled date and time. Users can participate in bidding and the book goes for the highest bidder.

Online Auction Workflow



Api's developed

- Api to put the book for auction.
- Edit auction details.
- Store data of the bid amount of every user .
- Api's to exchange contact details of owner and highest bidder
- Display books that are up for auction.
- Books bought in auction
- Books sold in auction
- Api to filter data based on genre, auction data and base amount.

Module 8

- **Reviews and ratings,** for each book.
- **Notifications** for any updates.
- **Google calendar reminder integration**
- **Data population.**

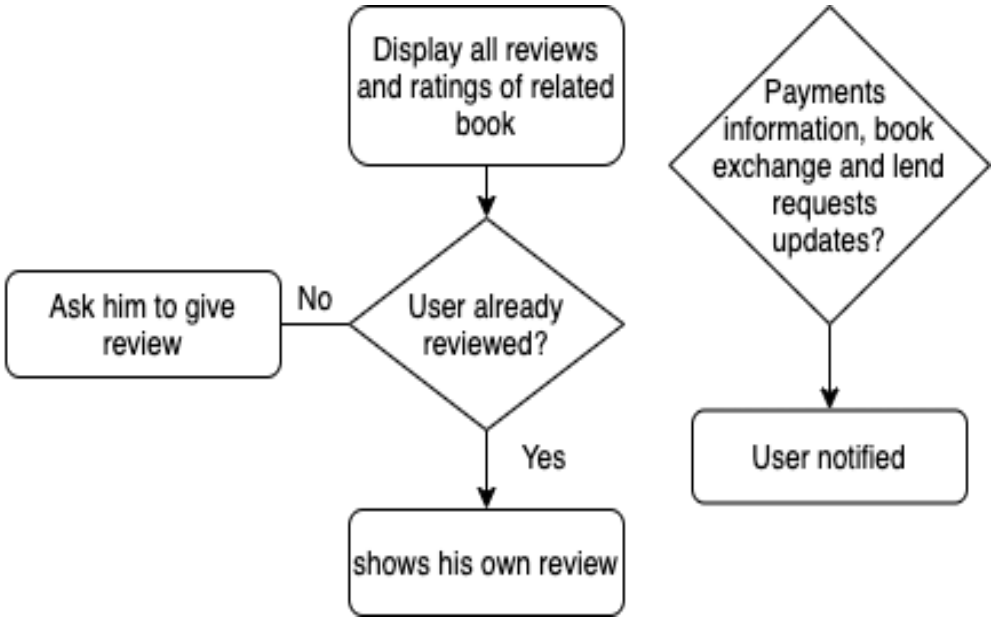
Status: Completed

Users can give a review to the books and they can also edit or delete that review. Users will be notified when some accept his/her read request. When the other user received the book that has been sent as part of exchange requests

Api's developed

- Api to display all notifications.
- Api to change notification status to read
- Api to display all reviews
- Api to store user review

Reviews and Notifications Workflow



Database Schema

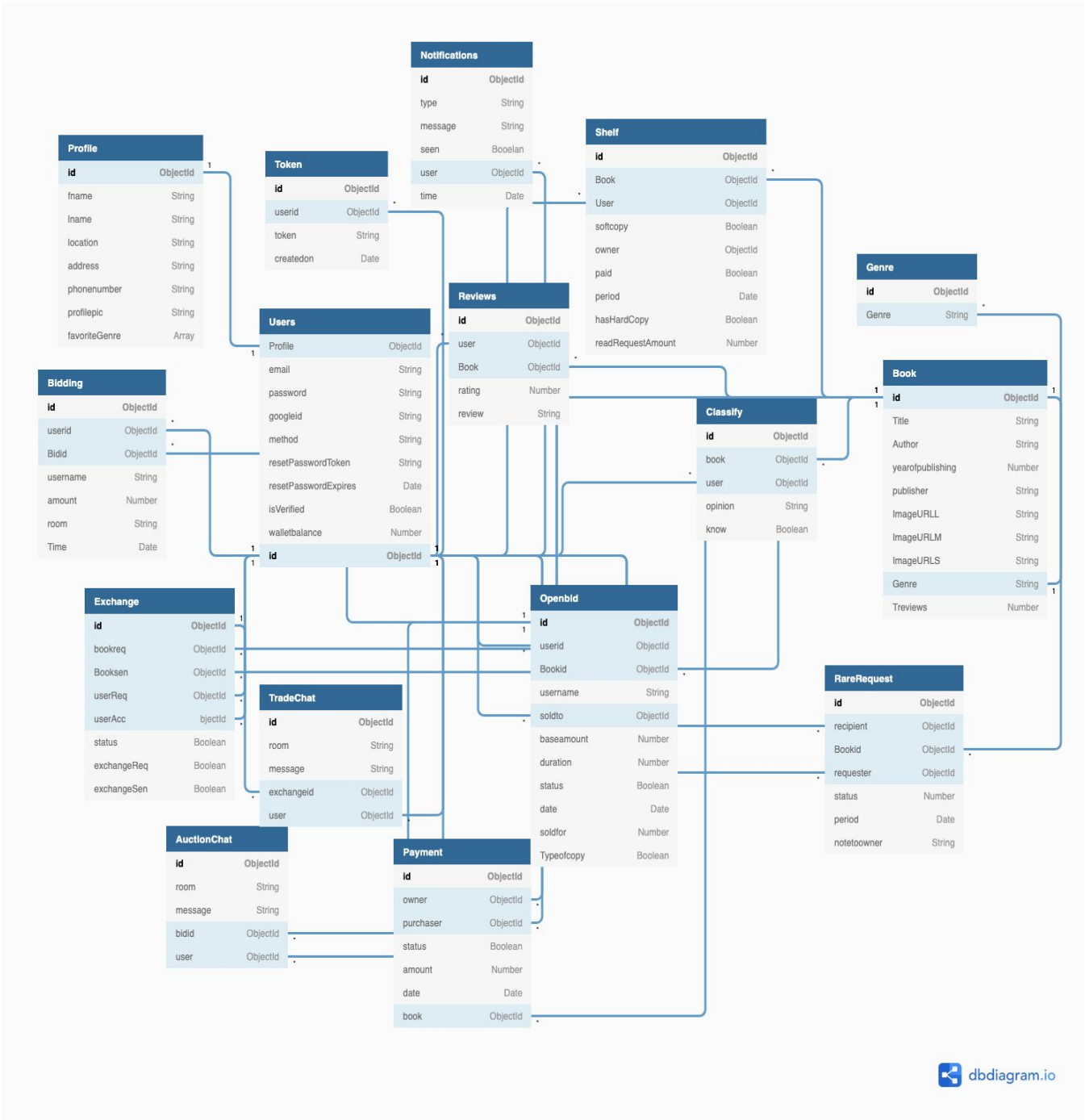
We are using the Mongodb database which is a Nosql database.

We are storing data in a Mongo Atlas server in Aws.

We currently have data of 1000 users, 8000+ books, half a million reviews and ratings required for recommendations.

Link to the Database Schema

Schema on the next page...


Notifications
id ObjectId
type String
message String
seen Boolean
user ObjectId
time Date
Profile
id ObjectId
fname String
lname String
location String
address String
phonenumber String
profilepic String
favoriteGenre Array
Token
id ObjectId
userid ObjectId
token String
createdon Date
Shelf
id ObjectId
Book ObjectId
User ObjectId
softcopy Boolean
owner ObjectId
paid Boolean
period Date
hasHardCopy Boolean
readRequestAmount Number
Genre
id ObjectId
Genre String
Reviews
id ObjectId
user ObjectId
Book ObjectId
rating Number
review String
Users
Profile ObjectId
email String
password String
googleid String
method String
resetPasswordToken String
resetPasswordExpires Date
isVerified Boolean
walletbalance Number
id ObjectId
Bidding
id ObjectId
userid ObjectId
Bidid ObjectId
username String
amount Number
room String
Time Date
Book
id ObjectId
Title String
Author String
yearofpublishing Number
publisher String
ImageURLL String
ImageURLM String
ImageURLS String
Genre String
Treviews Number
Classify
id ObjectId
book ObjectId
user ObjectId
opinion String
know Boolean
Exchange
id ObjectId
bookreq ObjectId
Booksen ObjectId
userReq ObjectId
userAcc bjectId
status Boolean
exchangeReq Boolean
exchangeSen Boolean
Openbid
id ObjectId
userid ObjectId
Bookid ObjectId
username String
soldto ObjectId
baseamount Number
duration Number
status Boolean
date Date
soldfor Number
Typeofcopy Boolean
RareRequest
id ObjectId
recipient ObjectId
Bookid ObjectId
requester ObjectId
status Number
period Date
notetoowner String
TradeChat
id ObjectId
room String
message String
exchangeid ObjectId
user ObjectId
AuctionChat
id ObjectId
room String
message String
bidid ObjectId
user ObjectId
Payment
id ObjectId
owner ObjectId
purchaser ObjectId
status Boolean
amount Number
date Date
book ObjectId
dbdiagram.io