

University of Mumbai

RecBooks - Find your next read

Bachelors in Technology

by

Mihir Dholakia

Roll No: 1811074

Jash Shah

Roll No: 1811082

Naman Shah

Roll No: 1811113

Guide

Dr. Archana Gupta



Department of Computer Engineering
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **RecBooks - Find your next read** submitted by **Mihir Dholakia** at the end of semester VII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai

Guide

Head of the Department

Principal

Date:

Place: Mumbai-77

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **RecBooks - Find your next read** submitted by **Jash Shah** at the end of semester VII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai

Guide

Head of the Department

Principal

Date:

Place: Mumbai-77

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **RecBooks - Find your next read** submitted by **Naman Shah** at the end of semester VII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai

Guide

Head of the Department

Principal

Date:

Place: Mumbai-77

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **yBully - A Chrome extension for Cyberbullying Detection** submitted by **Mihir Dholakia, Jash Shah and Naman Shah** at the end of semester VII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai

Guide

Head of the Department

Principal

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **RecBooks - Find your next read** is a bonafide record of project work done by **Mihir Dholakia** during semester VII.

This project work is submitted at the end of semester VII in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

Internal Examiners

External/Internal Examiners

Date:

Place: Mumbai-77

Certificate of Approval of Examiners

We certify that this dissertation report entitled **RecBooks - Find your next read** is a bonafide record of project work done by **Jash Shah** during semester VII.

This project work is submitted at the end of semester VII in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

Internal Examiners

External/Internal Examiners

Date:

Place: Mumbai-77

Certificate of Approval of Examiners

We certify that this dissertation report entitled **RecBooks - Find your next read** is a bonafide record of project work done by **Naman Shah** during semester VII.

This project work is submitted at the end of semester VII in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

Internal Examiners

External/Internal Examiners

Date:

Place: Mumbai-77

Certificate of Approval of Examiners

We certify that this dissertation report entitled **RecBooks - Find your next read** is a bonafide record of project work done by **Mihir Dholakia, Jash Shah, Naman Shah** during semester VII.

This project work is submitted at the end of semester VII in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

Internal Examiners

External/Internal Examiners

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written report submission represents the work done based on our and/or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty, and integrity as we have not misinterpreted or fabricated, or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student Roll No. 1811074	Signature of the Student Roll No. 181082
Signature of the Student Roll No.1811113	

Date:

Place: Mumbai-77

Chapter 1

Introduction

This chapter explains the current situation, the need for, and why we chose the problem statement we did, as well as the scope of the project.

1.1 Background

The following topic was chosen primarily because it addresses a real-world issue. Reading is a common pastime for people of all ages. People read books for a variety of reasons, including entertainment and education. However, there is a catch. It occurs to all of us frequently that when we sit down to watch a movie, we spend half of the time debating which one to watch. We've come up with recommendations for users' future reads, similar to how Netflix solved this problem for movies by proposing their next watch.

1.2 Problem Statement

Reading is an activity that is enjoyed by a lot of people. Reading can increase knowledge, help you relax and reduce stress, make you a better writer, enhance your vocabulary and improve your focus. However, readers often run into the problem of not finding the right book that fits their personal taste and style of reading which can sometimes waste a lot of time and effort. To solve this problem, we will implement a recommendation system and make use of data from various users to recommend new books based on the past reading habits of the reader.

1.3 Scope of the Project

We are going to incorporate the following features:

1. After installation, the user will be asked to register on the app. They can login through gmail. We have used firebase for authentication of the user profiles.
2. For every book there will be a like button through which the user's preferences will be taken into account. These liked/bookmarked books help the app to recommend new books to its users on the homepage.

3. The User will be able to search for a book via:
 - a. User manually searching titles: Advanced book searching filters using Title, Author, Category, Publisher,
 - b. ISBN number through barcode scanning: By scanning the barcode present on the back of the book, we will get the book information.
4. We will provide book recommendations on the following criteria:
 - a. Based on books liked/bookmarked by the user.
 - b. Users will be recommended books similar to the genre or title of the book page they visit
5. Users can create his/her profile in which all the liked/bookmarked books will be saved.
6. We will also show places it is available to buy/download (online).

1.4 Hardware and Software Requirements

Hardware(for development purpose):

Processor - Intel Core i5 or higher

Memory - 8GB Memory or higher

GPU: 2GB Memory or higher

Hardware(for user):

Smartphone with decent camera

Software(for development purpose):

- Framework - Flutter
- Tools - Firebase, Android Studio Code(IDE), Visual Studio Code(IDE), GitHub, Google Collaboratory

Software(for user):

- Mobile versions: Android Jelly Bean, v16, 4.1. x or newer
- Permissions needed: Camera, Media and Storage permissions
- Other requirements: Reliable internet connection

1.5 Brief description of Project

RecBooks is a mobile application which helps users to find their next read. The app recommends books to its users using Content Based and Collaborative Recommendation algorithms. The app also has an Advanced searching module, including an ISBN barcode scanner, to allow users to search for books using the ISBN barcode found on the front cover. The users can bookmark their favorite books which are stored on firebase associated with the user account. If the user wants to buy the book, then an option is provided to the user which redirects them to Amazon.

1.6 Organization of the Report

The project report consists of the research, implementation, the result of each approach, and potential future advancements in the following chapters:

Chapter 2: Literature survey - presents the technical papers reviewed as a part of the literature survey, which has played an important role in deepening our understanding.

Chapter 3: Project Design - the proposed model and project architecture of the system is explained, along with the requirement specifications and related diagrams.

Chapter 4: Implementation & Experimentation of Prototype Model - includes the information and pictures of the implemented modules.

Chapter 5: Testing - includes all different test cases used for feature testing

Chapter 6: Conclusions and scope for further work - concludes the idea behind the project along with indications toward the future scope of the same.

Bibliography and Acknowledgements - includes the references used for the implementation of our application.

This chapter covered the problem definition, scope of project undertaken and how the report is structured in the coming chapters. The next chapter will include the survey performed for the project, with the potential approaches that could be taken.

Chapter 2

Literature Survey

This chapter provides a summary of the research articles that have been read and examined in order to learn more about the development of Scannary and the various strategies that have been utilized in the past to construct this scanning application with text summarisation.

2.1 Research Papers and Articles

2.1.1 Book Cover Recognition

The author of the report tries to create a simple GUI using MATLAB platform where the user would be able to upload an image of the cover of the book and later the algorithm tries to extract and improve the image which could be used to extract the necessary information. This detected information is later checked if it needs any further correction in grammar or spelling and at last, the final string acts as the input for the amazon search box. The final string contains the name of the book whose photo the user has selected and later the details of that book or similar book are displayed to the user. The algorithm got around 86% accuracy for getting the name of the book and successfully arriving at the result with some tolerance. The OCR used in the application is an amalgam of MATLAB OCR and Tesseract OCR where the result of both the string is used to gauge the final string which would be used to perform the search on the e-commerce platform.

Main Idea

MSER stands for Maximally Stable Extremal Region introduced by Chen H which provided a great and satisfying result on the images captured or uploaded from the mobile phone. After doing all the research and studying various OCRs and their outputs, the author decides to go ahead with MESR due to its better accuracy overall.

They developed a simple pipeline that demonstrates how image processing and extraction would be done after the image is uploaded on the application. As Matlab was more popular and gave them a detailed blog on how to extract text from an image, the team decided to choose MatLab over other GUI platforms.

As the GUI would be taken care of by the MatLab platform the main part is now the implementation of the MSER algorithm.

The pipeline consisted of 4 steps:

- 1) Image processing: After the user uploads the image it still needs some work like changing the skewness, color contrast, region detection, identifying the bounding box. The first output from the 1st stage is the boundary detection which would be focused by the OCR to get the characters from the cover image.
- 2) Character recognition: For text extraction, two OCRs were used. One was built-in MatLab and the other one was OpenSourced by Google. In addition to it, both were based on the Tesseract engine but varied in their performance.
- 3) Word correction: There are high chances of a word, or a letter being falsely recognized by the engine which needs attention and correction to get proper results from the web. The wrong results were altered into their correct spellings by downloading the English word package.
- 4) Price searching: Once the string is corrected and they can get the final title or text from the cover image they send that query to the amazon database to fetch the price, reviews, etc, and display it to the user.

2.1.2 Book Recommendation System Based on Combine Features of Content Based Filtering, Collaborative Filtering and Association Rule Mining

This paper presents a new approach for recommending books to the buyers. Purpose of this book recommendation system is to recommend books to the buyer that suits their interest. This recommendation system works offline and stores recommendations in the buyer's web profile.

This system combines the features of content filtering, collaborative filtering and association rule mining to produce efficient and effective recommendations.

Main idea:

Content recommendation system recommends books to the buyers based on the content of the buyers past book buying history. Buying history gives the content of the overview of the books, in which the buyer is generally interested in the large amount of books. Content recommendation system uses Content based filtering for doing the separation.

Limitations: Finding the quality of the content. For example Content based filtering cannot differentiate between good article and bad article if both of them are using the same terminology.

Collaborative filtering systems are used because they are based on the opinion of the other users. Item based collaborative recommendation algorithm looks into the set of items the target user has rated and computes how much similar they are to the target item i and then selects k - most similar items $\{i_1, i_2, \dots, i_k\}$ to the set of items the target user has rated, the recommendation is then computed by taking the weighted average of the target user's rating on these similar items. To compute the similarity between the two items, each item is considered as a vector in m dimensional user-space. The similarity between two items is measured by computing the cosine of the angle between the two vectors.

Association rule mining finds an interesting association and correlation relationship among a large data set of items. In market basket analysis customer buying habits are analyzed for finding association between different items that the customer puts together in their shopping cart. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of items. An association rule can be represented by this form $A \rightarrow B$, where $A \subseteq I$, $B \subseteq I$ and $A \cap B = \emptyset$. Association rule extracts the pattern from the database based on the two measures: minimum support and the minimum confidence.

In general association rule mining can be thought of as a two step process. (i) Generating all item sets having support greater than or equal to the user defined minimum support. (ii) Generate all the rules having the confidence factor greater than or equal to the user defined minimum confidence.

Overall the authors suggest the following 7 steps to build this recommendation system:

Find out the category of the book that the buyer has bought earlier like novel, science, engineering etc from the buyer's web profile.

Find out the subcategory of the book if there is any in the step1 found category.

Perform content based filtering in category / subcategory found in step1 and 2, to find out the books that are much similar to the books that the buyer has bought earlier based on the books overview content from the buyers past history record.

On the result of step 3 perform item based collaborative filtering and find out the list of books in the descending order of recommendations. In this step system actually evaluates the quality of the recommended books based on the rating given to those books by the other buyers.

From the book transaction database find all those transactions whose category and sub category (if there is any) is the same as found in step1 and step2. Apply association rule on those transactions and find out the books that the buyer can buy afterward. Adjust the support and confidence parameters to get stronger rules.

Find out the intersection of the result of step 4 and 5. Arranging the intersection results in the descending order of recommendations as given by the step 4. This step is actually more refining the recommendations generated by the step 4.

Outcome of step 6 is the final recommendations for the buyer. All these steps are performed when the buyer is offline and the results are stored in the buyers web profile. When the buyer comes online next time the recommendations will be generated automatically

This book recommendation has considered many parameters like content of the book and quality of the book by doing collaborative filtering of ratings by the other buyers. This recommender system also uses an associative model to give stronger recommendations. This system does not have performance problems since it builds the recommendations offline.

2.1.3 Development of Book Recommendation Service for Implementation on Android

In this paper, the authors discuss the theme of creating a service that focuses on accounting for reading activities by its users. The goal of the study is to develop classification methods and sorting literary works into categories that can be incorporated into any literary digital resource to increase its functionality. The service that was created is an Android app that allows you to keep a reading diary and also create literary recommendations based on works added to the user's personal library. The feature of expanding the literature base by scanning ISBN codes of books that are not in the Google Books database is one of the app's additional features.

Main Idea:

Accounting and analysis of reading activities, as well as the ability to make new recommendations based on the data collected, are becoming increasingly important. The book database is chosen from Google Books due to the existence of an extensive library, various methods of searching for books and practical use.

The functionalities of the application are:

The user can mark the start and end date of reading a particular book. This information will be available under the user statistics tab.

The users can create the right folders for themselves and not the predefined folders provided by the developers, giving the users flexibility.

Literature collections that are created to all user preferences which are based on assessments and books read, become more attractive and offer relevant information to a user. Therefore, the application will not only combine a collection of satisfying characteristics, but also take into account new user's requests.

Expanding the library with ISBN codes. Using the ISBNdb library, it is possible to add the required book by code, if it is not present in the database.

The window with information about the book provides not only a description of this book, but also allows users to fill in the "characters" and "impressions" fields themselves, as well as edit keywords as desired.

Search window allows you to search for books when accessing connected libraries through the search string. The search string also has a separate section for searching books by ISBN code. In addition, this window displays folders with the most popular genres, where the user can view books of a particular genre.

Steps in creation of the application:

Creation of a single page Android application.

Connecting the Google Books database by obtaining the key for Google Books API.

Creation of the algorithm for literary recommendations relying on the user's doings in the application. The history of his ratings will allow you to find books of a similar genre or content, while connecting keywords that contain books.

2.1.4 Research on Personalized Book Recommendation Model for New Readers

The improvement of computing levels in recent years has made it possible to make personalized recommendation lists for the users based on their behavior. With the growing number of books in the university library, personalized book recommendations are in high demand. The author creates a model that targets students who have never borrowed a book from the library and uses a collaborative filtering algorithm. Using their course selection records and existing borrowing data of known users, the model generates book recommendation lists for the target users. Finally, the paper compares the effects of different parameters setting in the algorithm.

Main Idea:

In this paper, the authors have solved the problem of recommending books to the new users which have no previous records available. While finding a solution, the authors have also tackled

the problem that the common recommendation systems face i.e. this system is not able to cater to the needs of new users since they solely rely on the student's borrowing records. Therefore, to provide personalized recommendation to the new users this paper proposes a book recommendation model combined with students' course selection data and collaborative filtering algorithm.

Build process of the book recommendation model proposed in the paper:

Data preprocessing: It involves the removal of two parts of unused data to reduce the computational complexity. The first part is the relevant data of the users who have only selected courses but have not borrowed books. The second part is the related data of the users who only borrow books but have not selected those courses.

Model Building: Calculate the similarity between known students and target users with three different rulers i.e. calculate with Cosine similarity, Euclidean similarity and Jaccard similarity.

Calculate each book's recommendation coefficient for the target user based on the top n users with the highest correlation with the target user (neighbors). Finally, the book recommendation list is generated by books, which has the highest correlation coefficient.

It carried out recommendation trials with different values of three parameters in the model: the number of book recommendations (5 books, 10 books and 15 books), the number of neighbors (40 values between 5 and 200) and different similarity calculation methods.

2.4.5 Title Extraction from Book Cover Images Using Histogram of Oriented Gradients and Color Information

Pattern recognition and machine learning play increasingly important roles in handling diversity of features in large datasets. The application will recognize characters in the title, and display all the details of that book. In order to recognize characters in the book cover image, the title regions have to be extracted. To improve the accuracy of title recognition, the accuracy of the title extraction needs to be worked on. In this paper, they propose a method that uses the combination

of Histogram of Oriented Gradients and color information to detect the title areas of the book cover.

Main Idea:

The goal of the system is to localize text and extract the title of the book cover images. The proposed method consists of two steps. The first step i.e., text localization is based on oriented gradient of the edge map and color information. The Second Step is the refinement process which is performed to extract the entire title of the book cover image.

Text Localization: Text localization consists of Detecting the lines of text using Histogram of Gradients (HOG). In HOG, local object appearance and shape within an image can be described by the distribution of edge directions. Consequently, areas that are text candidates can be predicted. Then the second task is removing non-title areas using area constraints. In this step, the canny operator for edge detection is applied on the previous result and the area constraint is also used. They then proceed to remove the non-title using color information wherein the cluster that has larger number of connected components is considered to be the title candidate and k-means clustering is applied. Finally, the non-titled area is removed using length as constraint.

Refinement Process: The purpose of this step is to try to recover all characters of the title. First calculate all the bounding boxes of the connected components. If the bounding box satisfies the two conditions in the given equations, it could be considered to be in the text line.

Result: More than 86% of the cover book images title areas are correctly detected. 5% of the cover book images are wrongly detected for title candidates. In addition, the system cannot detect all parts of the title in 2%.

This chapter presents the overview of the research papers that have been studied and analyzed to obtain knowledge regarding the development of Recommendation System and various methods that have been used for Book Recommendation development in the past.

Chapter 3

Project Design

The proposed system, project system design and workflow are presented in this section. There's a software management plan and a software requirement specifications section. To clearly define the proposed system, all of the project design diagrams are shown.

3.1.1 System Design Diagram

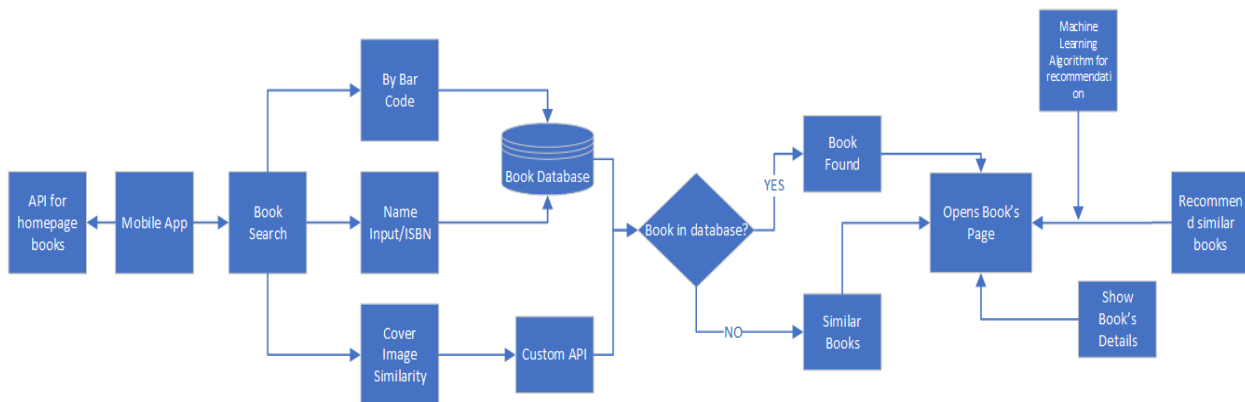


Fig. 1

The diagram above is a simplified representation of the sequence of events that would result in the application working properly. The process would begin with the Mobile App, where after authentication, the user would see a homepage with a collection of books as well as a search bar to find a specific book, which would serve as an initial query for the recommendation engine. The book's title will be used to recommend a few books to the user.

3.1.2 Proposed Functions

1. The User signs up/ logs in from the application.
2. New user is added to the database after successful authentication.
3. The proposed system prompts the user with a welcome message and a search bar on top with various book recommendations on the home page.
4. The user can search for books stored in the database with the ISBN code or the title of the book.
5. Advanced filters are also available in the search to filter and sort through various parameters.
6. If the book is found, it is displayed with the link to buy the respective book.
7. Similar books can be recommended using the title and isbn code with the help of Content based and Collaborative filtering.
8. All the recommendations will also include links to buy the respective books.

3.1.3 Functional requirements

1. Register/sign in using Firebase.
2. Storing a user's profile using Firebase.
3. Redirect to home page
4. Home page:
 - a. Show random books from dataset as trending now
 - b. Show image and title
 - c. Once a user starts using the app, recommended books will be shown here.
 - d. Camera button will be provided to click a photo of the cover page of a physical book present with the user.
5. Based on searching a book using any of the three methods, recommendations will be made on a separate page.
6. For each book:

Once a user clicks on the book, various details about the book are shown like summary (if available), title, author, publication, year etc. Also a like button will be provided to store that book in the user's profile as a liked item.

7. For each user a profile page will be created which will display the user's information and store the user's favorite books (bookmark).
8. Based on the books liked by the user, genre selected and authors selected books will be recommended on the home page.

3.1.4 Non functional requirements

1. Reliability

- Optimal functionality of the application for as long as it is supported on the device
- Minimal maintenance of the system that can be automated as well.
- Adaptable to changing hardware and operating environment, making it versatile over time.
- End user should be able to rely completely on the efficiency of the application.

2. Availability

- The application should be accessible to the user 24/7.

3. Scalability

- The system should be able to perform well even if the volume of data and number of users increase.

4. Accuracy

- The system should have good accuracy for recommending books to the user.

5. Portability

- It should be able to work seamlessly on any version of Android from Android 9.0 onwards providing necessary portability.
- The application is portable and can be accessed using any smartphone device.

3.2 Technology Used

- a. **Firestore:** Firestore is used as a service to assist with Google login or as a database service provider to store user information. Firestore, a cloud-based solution with simple encryption provided by Google, aids in the security of app data. It also functions as a database, storing additional information such as books by user, name, and so on.
- b. **API Hosting:** We used a simple Flask API structure to build the application, which will act as a bridge between the mobile application and the entire backend part of the

recommendation engine, because our app needs to fetch the results provided by the algorithm. To make the API available everywhere, we needed to deploy it on a cloud-based platform, which we chose Python Anywhere for. Python Anywhere is a well-known Python application hosting platform where we will host our flask application, Dataset, and the algorithm that will run to provide the recommended books.

- c. **Flutter:** We chose Flutter as the technology to use for developing a hybrid application. Google created Flutter, an open source software development kit with a variety of features and ease of use, to promote the mobile app development industry. Currently, Flutter aids in the development of applications that can run web-based, mobile-based, and Wear OS-based applications. In our case, the Flutter Application would serve as the frontend component, allowing the user to search for a book while also receiving a few recommendations on the recommended page.
- d. **Python:** Python is a popular programming language for machine learning and data science projects. Thus, for our recommendation algorithm, we used Python to perform the computation while developing the API that would connect the backend to the mobile application using the Flask framework.

3.3 Recommendation Algorithms

Two recommendation algorithms are implemented:

1. Item Based Collaborative Filtering
2. Content Based Collaborative Filtering

Item-Based:

Collaborative filtering systems are used in many recommendation systems because they are based on the opinion of the other users. In our model, this takes into account the title of the books only.

Usually, the Item Based Collaborative recommendation algorithm looks into the set of items the target user has rated and computes how much similar they are to the target item i and then selects k - most similar items $\{i_1, i_2, \dots, i_k\}$ to the set of items the target user has rated, the recommendation is then computed by taking the weighted average of the target user's rating on these similar items.

To compute the similarity between the two items, each item is considered as a vector in m dimensional user-space. The similarity between two items is measured by computing the cosine of the angle between the two vectors.

So in all, it first finds similarity between item pairs using cosine similarity, then calculates the weighted sum using the ratings that particular book receives. Then combine these similar items into a recommendation list.

Content Based:

Content based filtering is based on the content (i.e. on title, summary, author, publication etc. in our model) of the buyer's past book buying history. Book buying history gives the content of the overview of the books, in which the buyer is generally interested in the large amount of books.

The main idea of content-based methods is to try to build a model, based on the available “features”, that explain the observed user-item interactions. Based on the similarity in content of the books, new books are recommended.

This filtering method uses item features to recommend other items similar to what the user likes and also based on their previous actions or explicit feedback.

3.4 Software Project Management Plan

Timeline	Task (to be filled by the students at the time of IRRC approval)
July Second fortnight	Submit initial draft to project committee.
August First fortnight	Initial literature survey performed as per suggestions by project committee and received approval.
August Second fortnight	Scope and detailed problem statement and further elaborate literature survey. Complete description of the intended purpose and behavior of the software to be developed. Identifying the tools and the data required.
Sept. First fortnight	Finish literature survey and propose system model.
Sept. Second fortnight	Start dataset preparation. Identifying the data sources, collection of required data, and grouping of related data required for each part of the process.
Oct. First fortnight	Designing of model (class diagram, Database design, Data Set, UML model) and Elaborate H/W and S/W requirements.
Oct. Second fortnight	Research and Development of recommendation engine. Understanding the technology and concepts related to

	recommendation system.
Nov. First fortnight	Research and development for image similarity required by the application.
Jan. Second fortnight	Design and develop UI of mobile application.
Feb. First fortnight	Integrate recommendation system with mobile application.
Feb. Second fortnight	Complete integration of recommendation systems.
March First fortnight	Integrate image similarity module with mobile application.
March Second fortnight	Start initial testing and compose initial draft for technical paper.
April First fortnight	Complete testing and finalization of technical paper.
April Second fortnight	Preparation of complete Report, presentation and other documentation.

3.5 Software Requirement Specification Document

3.5.1 Project overview

The project contains the following major components.

- Android app
- Database

3.5.2 User Interface Requirement

The major points kept in mind, while designing User Interface:

1. Clean
2. Understandable
3. Responsive

3.5.3 Sequence Diagram

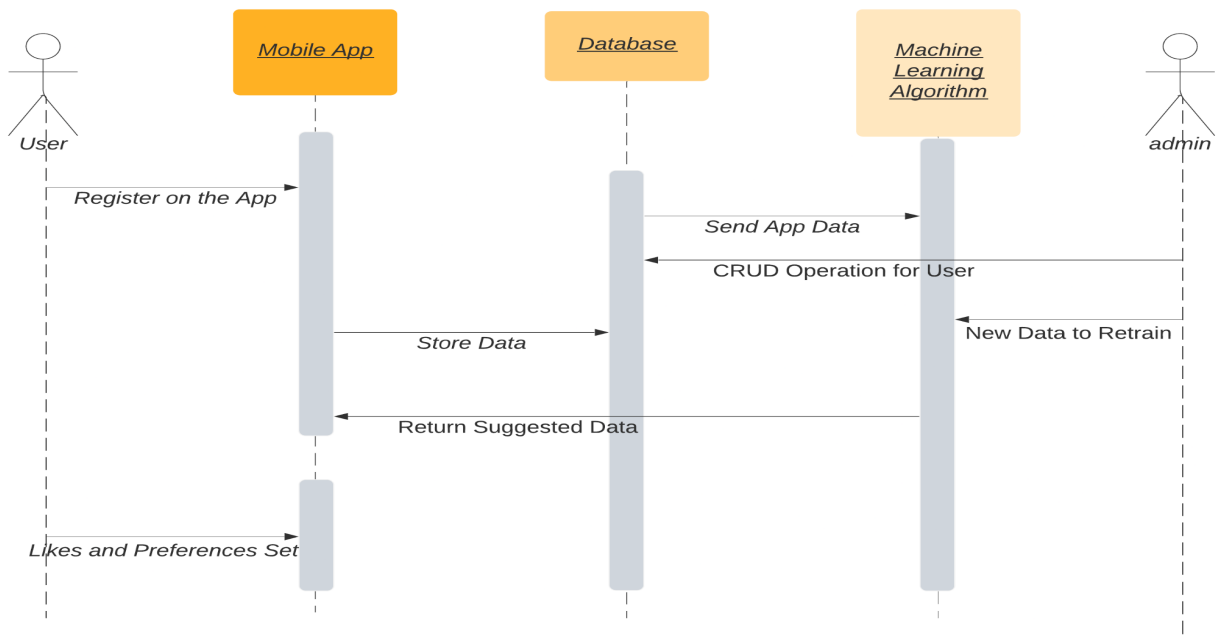


Fig. 2

In this chapter, the necessary requirements were defined, a timeline for the project was presented and a system model was designed. In the next chapter, the actual implementation of the project, with an explanation, is presented.

Chapter 4

Implementation & Experimentation of Prototype Model

This chapter presents the steps taken toward the implementation of the software, both in the frontend as well as the backend.

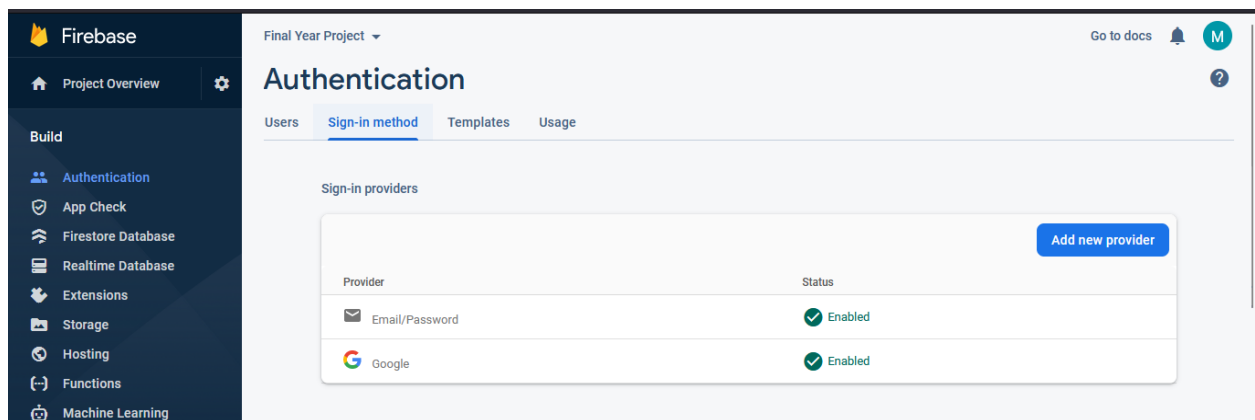
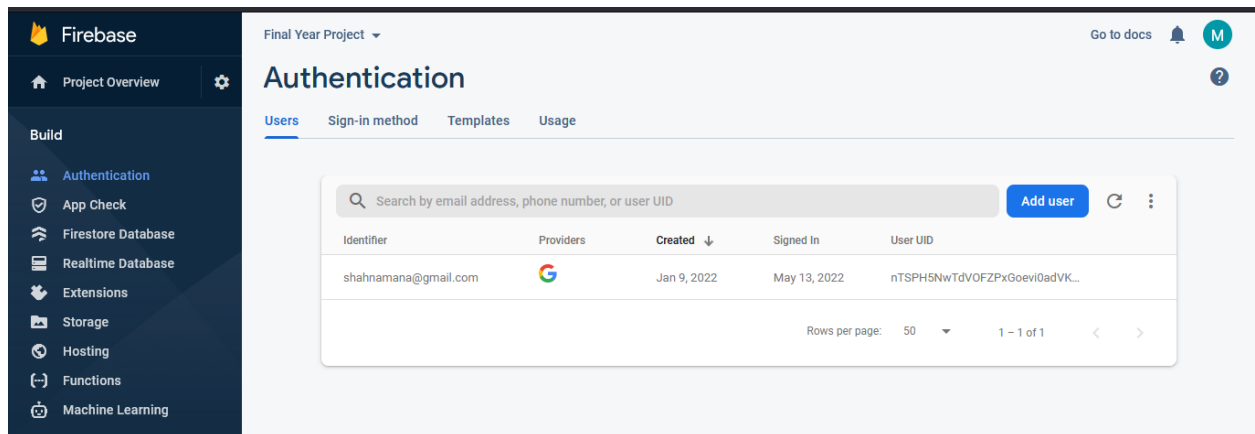
The entire project was implemented using Dart language in Flutter for Mobile Application Development. Multiple Dart Libraries and Plugins were used including `google_sign_in`, `path_provider`, and `firebase_auth` for complete login system with authentication.

4.1 Login Module & Profile Section


Using the Google Log-In and Firebase Authentication supported by Flutter, we were able to create a sign-in and sign-out section. For the most part, Firebase authentication supports a variety of user authentication certifications. It's mostly a mobile backend as a service that offers fantastic features for developing flutter apps. It also helps by providing SDKs and quick UI libraries. This aids in authenticating your clients' eligibility to use the application. There are also exceptional authentication features available. Two plugins were required for this task :

- `google_sign_in`
- `firebase_auth`

To configure, we first created an account on Firebase Website and registered our Application with the Application ID in the app-level build.Gradle file. Firebase then provided a corresponding `google-services.json` file which was to be imported in our app module root directory. Later we added the dependencies and assets, and imported these flutter packages. Similarly, the same logic was used during SignOut.



The login and Sign - up pages are first displayed when you open the application. Traditional sign in and google sign in are both available as options.



RecBooks
Find your next read

Login


✉ Email


🔒 Password

Submit

[Don't have an account? Sign up here](#)

OR

 Sign in with Google



RecBooks
Find your next read

←

Sign Up

👤 Full name

✉ Email

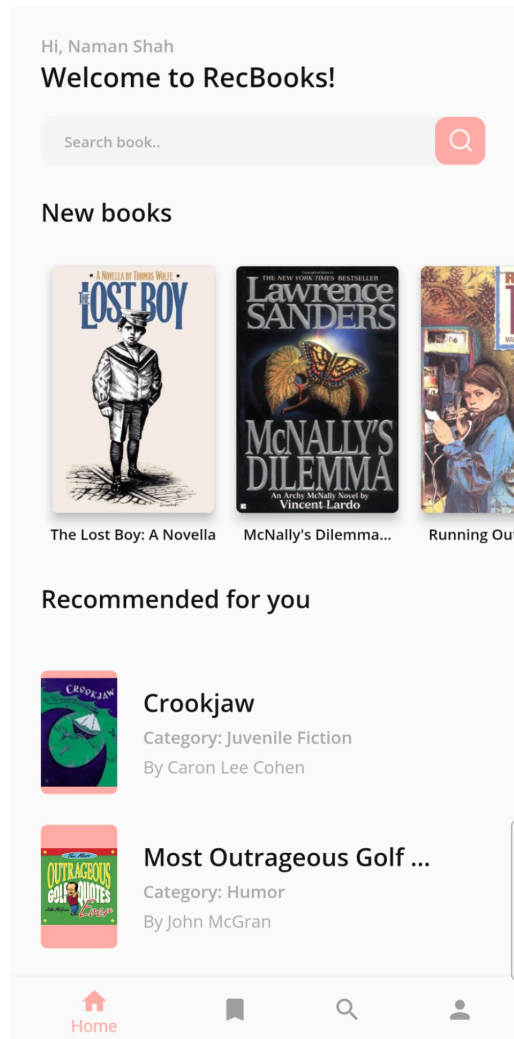
🔒 Password

🔒 Confirm password

Submit


4.2 Home Page

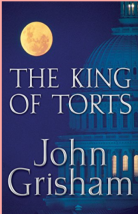
Our Home Page displays a list of all New Books which are randomly displayed and Recommended books for the user by using any one of the following algorithms ; Content based filtering and Collaborative filtering. Search Functionality is provided to the user so that he can find any book given in the dataset.



4.3 Individual Book Section

This Section displays the cover of the book, its title and author in the header. ISBN 10 and ISBN 13 are both present for the user to find more information about the books if he/she wants. The category of the book and the summary is displayed for better understanding of the selected book. All this information is directly taken from the dataset. An option to bookmark the book for storage in the database and link to buy the book directly from Amazon is provided.





The King of Torts

By John Grisham

ISBN 10: 034553199X

ISBN 13: 9780345531995

Category: Fiction

Summary


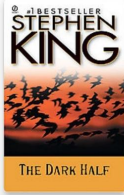
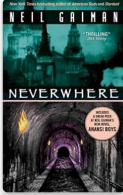
The author of such best-selling legal thrillers as The Summons and The Brethren presents his latest novel of courtroom and legal suspense. 2,500,000 first printing. \$2,500,000 ad/promo.

[Bookmark](#)[Buy on Amazon](#)

Summary

The author of such best-selling legal thrillers as The Summons and The Brethren presents his latest novel of courtroom and legal suspense. 2,500,000 first printing. \$2,500,000 ad/promo.

Recommend 1


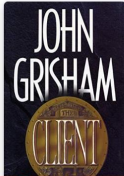
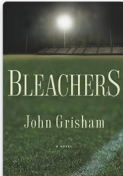


Neverwhere

The Dark Half

While I Wa

Recommend 2

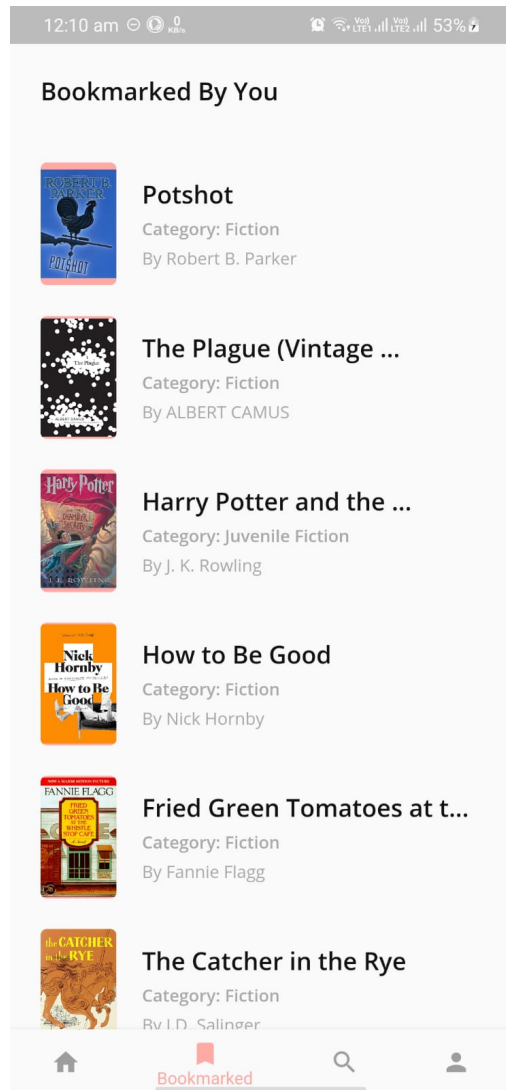


Bookmark

Buy on Amazon

4.4 Bookmark Page Section

The Bookmark Section displays all the books bookmarked by the user.



4.5 Advanced Searching

Advanced Searching page is selected by pressing the magnifying glass icon in the bottom toolbar. Parameters such as Title, Authors, Categories and Publishers are displayed which can be selected as per user choice to search for specific books. The Preference to search is given as Title first then Author then Category and then Publishers. This means that if we search for a book with a Title and category then the Title search parameter will be given more preference.

ISBN scanning is also available for users to directly scan the ISBN barcode given on books.



Search

Titles

Authors

Category


Publisher

Search

OR

Scan ISBN BarCode




Adv Search



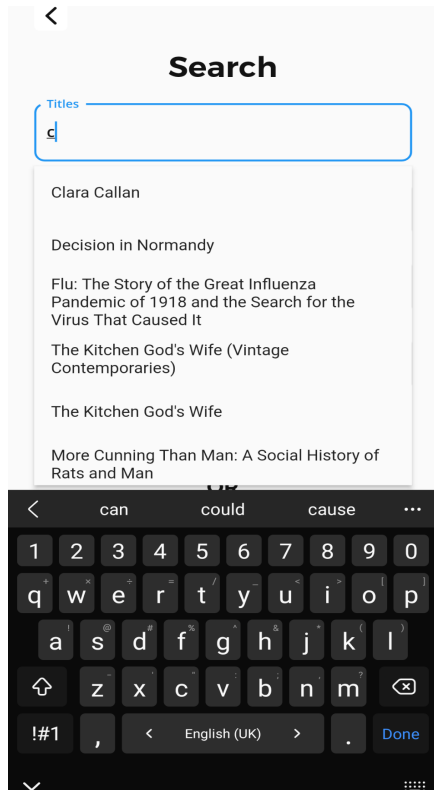


Fig. 9

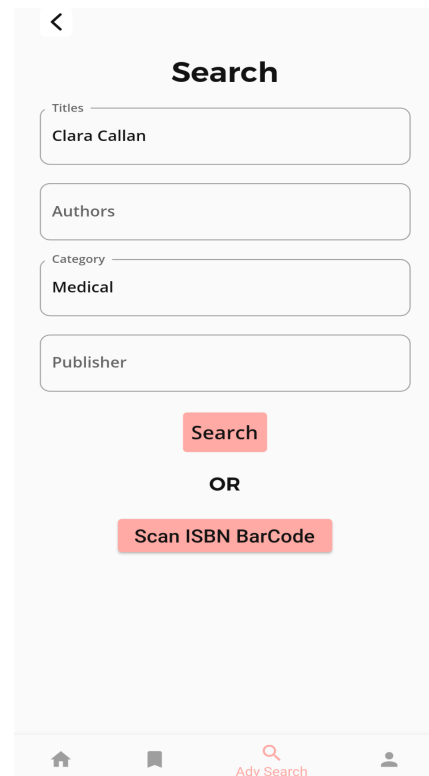


Fig. 10

Barcode Scanner

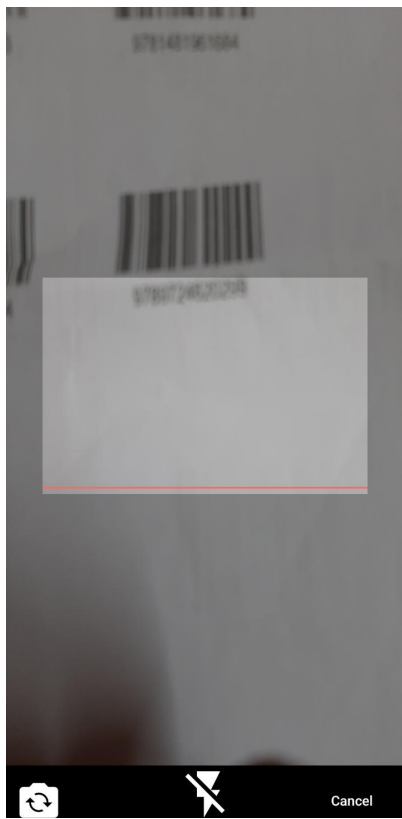


Fig. 11

Search Result Page



Fig. 12

Chapter 5

Testing

This chapter presents the s testing and validation plans.

5.1 Introduction

This chapter documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project's product. The Test Plan document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and testing team.

5.2 Black Box Testing

- Login Screen

Test ID	Test Feature	Test Scenario	Test Steps	Test Data	Expected Output	Actual Output	Pas s / Fail
---------	--------------	---------------	------------	-----------	-----------------	---------------	--------------------

TL0 1	Check using already existing valid email id. (with network)	Scenario: Check Login with valid data and if email is verified	-Make sure Internet is connected -Tap on a valid email id	choose from the list given by the device	User logged In Successfully	As expected	pass
TL0 2	Check using already existing valid email id. (without network)	Scenario: Check Login with valid data and if email is verified	- Make Sure Internet is not connected - Tap on a valid email id	choose from the list given by the device	Login Failed	As expected	pass

- **Advanced Search**

Test ID	Test Feature	Test Scenario	Test Steps	Test Data	Expected Output	Actual Output	Pass / Fail
TM01	Searching for books with parameters of title and author.	Scenario : check if priority of books displayed is correct when multiple parameters are selected.	Make sure the title is given more priority than author	data	Title based search is above author searched for	As expected	pass
TM02	Searching for books using title and category	Scenario : check if priority of books displayed is correct when multiple parameters are selected.	Make sure the title is given more priority than category	data	Title based search is above category searched for	As expected	pass

Chapter 6

Testing

This chapter includes the future work and conclusion.

6.1 Conclusion

In Semester 7, we created a part of the front-end of our application, along with the authentication functionalities for users and organizations. The development of the recommendation system has been completed. Along with this the initial integration of the system has been initiated.

We've almost finished the UI of the application in Semester 8. We created the Advanced searching module, including an ISBN barcode scanner, to allow users to search for books using the ISBN barcode found on the front cover. The recommendation algorithms have also been integrated into the app. As a result, the application now has the ability to display the results of our recommendation system. The users can now bookmark their favorite books which are stored on firebase associated with the user account. If the user wants to buy the book, then an option is now provided to the user which redirects them to Amazon.

6.2 Scope for Future Work

- Working on the technical paper
- Cold start recommendation for new users and New books section backend changes
- Releasing this app on the play store and app store
- Getting application feedback from more users so we can get a conclusive result of our recommendation system

BIBLIOGRAPHY

- Book Cover Recognition
https://web.stanford.edu/class/ee368/Project_Autumn_1617/Reports/report_yang_shen.pdf
- Book Recommendation System Based on Combine Features of Content Based Filtering, Collaborative Filtering and Association Rule Mining
<http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=6779375>
- Development of Book Recommendation Service for Implementation on Android
<http://ieeexplore.ieee.org.library.somaiya.edu/stamp/stamp.jsp?tp=&arnumber=9396414>
- Research on Personalized Book Recommendation Model for New Readers
<https://www.semanticscholar.org/paper/Title-Extraction-from-Book-Cover-Images-Using-of-Do-Kim/1948e0361ce8b8f82b49effb0b655fd991076a59?p2df>
- Extraction from Book Cover Images Using Histogram of Oriented Gradients and Color Information
<https://www.semanticscholar.org/paper/Title-Extraction-from-Book-Cover-Images-Using-of-Do-Kim/1948e0361ce8b8f82b49effb0b655fd991076a59?p2df>
- Kaggle
Available : <https://www.kaggle.com/>
- Flutter documentation
Available: <https://docs.flutter.dev/>
- Firebase documentation
Available: <https://firebase.google.com/docs>
- Python documentation
Available: <https://docs.python.org>
- Python anywhere
Available: <https://www.pythonanywhere.com/>

ACKNOWLEDGEMENTS

We would like to thank our project guide Prof. Kavita Kelkar for her support and encouragement during this project. We would also like to thank Prof. Nirmala Shinde for her inputs and comments during the progress seminar which helped us in improving.

We are grateful for the Computer Department of our college for providing us with the opportunity to build this project.