

CMPE 272 Final Project Report

BookSwap Application

Team BookSwap: Raaga Pranitha Kolla, Somya Menghani, Praneetha Sripada, Srinivas Anudeep Chitta

Department of Software Engineering , San jose State University

San jose 95192, U.S

Abstract— A paymentless, Barter System based marketplace where people can swap their books . Users can upload the books they are willing to swap and can post a request list of books or genre of books they are interested in . Using ML, based on the user's interest the app will recommend books if available in the system for swapping. Users can decide time and location of exchange in the message window or use the application generated google map suggestions. All the conversations of the users can be seen in their respective message tabs. Users can search for books by title or author or genre in the dashboard and see a list of all available books within a selected radius . Users can view book details as mentioned by the book owner and send swap requests to the owner through messages if interested.

Keywords: BookSwap, book, author, barter system , machine learning, recommendation , ISBN , Tensorflow

1 INTRODUCTION

The application BookSwap excites all the bibliophiles out there! The users can have access to a wide category of books from various genres, penned by famous authors.

Many readers prefer the joy of having a solid book in their hands for a read over digitised ebooks. It can become quite expensive to buy every book one wants to read. Instead, readers can opt to lend and borrow books with other book enthusiasts who are willing to share. But how do such people connect with each other? Is this possible? Well, the answer would be yes! All they need is a platform that can aid them in connecting with other book lovers in their vicinity , anywhere across the globe.

BookSwap application is the platform developed for achieving this simple principle. All the user has to do is signup for BookSwap. With this, the user has the ability to search for the books he/she wants to read and connect

with people who have the book and are willing to barter for it. This application also recommends the users similar books that match their interests. The users can also search the books within a specified radius of their choice.

When the user does not have any book in mind and is

willing to explore books, our application can suggest some books that match their interest. The user has to just provide the details of their reading interests to BookSwap. The application uses this information to recommend similar books to the user that are available to swap. The swap decision is entirely up to the user. The user has the option to upload the details of the books he/she is willing to share..

The BookSwap users can contact each other through in-app messaging and decide on a place where they could meet and exchange books or could use the nearby public meeting points suggested by the application. Isn't this quite simple and comfortable!

Revenue can be generated by charging the businesses at the meeting points provided in the suggestions as it allows the owners to increase their customer base.

2 ARCHITECTURE AND IMPLEMENTATION

The BookSwap application is implemented with the following technologies (Fig 2.1):

- Front end: Html5+Css, Javascript and ReactJS
- Backend: Python, NodeJS, Flask
- Databases: Mongo DB
- Deployment: Amazon EC2 Instance
- Machine Learning: Tensorflow

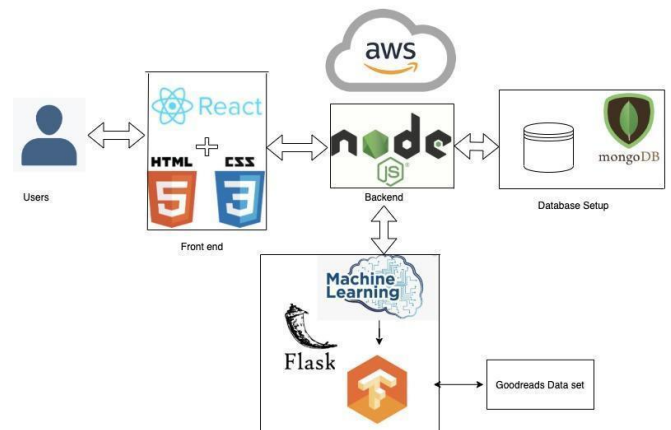


Fig 2.1: Architecture Diagram of BookSwap Application

3 PROCESS FLOW

Some use case scenarios below explain how BookSwap works:

#1. Linda is new to BookSwap. She wants to read the book *Alchemist*. All Linda has to do is navigate to the dashboard of the BookSwap application and search for 'Alchemist'. Our application also has typeahead search in case Linda does not remember the exact title of the book she wants. If any other user has put up this book for sharing on BookSwap, Linda can view their details and request them to swap the book and discuss any other questions by sending a message to them. If no user has shared the book, Linda cannot see any search results. Linda can also narrow her search range to view books available in her neighbourhood within walking distance and get the book she wants. Linda does not need to worry about meeting strangers, our application also suggests good public places like coffee shops in case she wants to swap the book.

#2. John, a BookSwap user, wants to explore new books. He gives details of his favorite books on his BookSwap profile. On navigating to the recommendations page of BookSwap, John can see books recommended by BookSwap, based on his interests. He can choose a book which is available on BookSwap and connect with the user who is sharing it.

A detailed process flow can be seen in the diagram (Fig. 3.1)

4 TECHNICAL ASPECTS IN BOOKSWAP

BookSwap collects the user's geolocation to bind the latitude and longitude of the book's upload location as a property to the book. When a user selects a range to limit the radius of the available books, the app captures user's live location and haversine's formula[5] is used to calculate the distance between the user and the book.

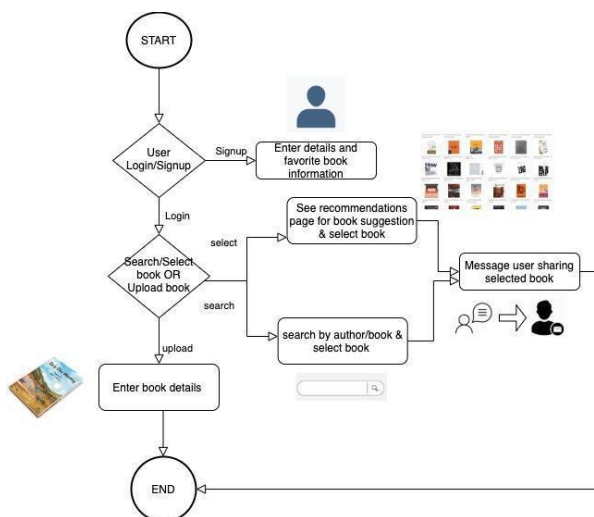


Fig 3.1: BookSwap Process flow diagram

5 HOW BOOKSWAP RECOMMENDS YOU BOOKS?

Can BookSwap read your mind? How can BookSwap recommend books its users may like to read? This is possible through Machine Learning.

5.1 Machine Learning Implementation for BookSwap

BookSwap Machine Learning model uses GoodReads dataset to give recommendations to the user, based on the genres of the selected book. Matrix Factorization algorithm has been used to find nearest neighbors of a given book based on the genres. Cosine score was calculated to recommend the books based on their highest score. This model was run on TensorFlow.

The Machine Learning inference for recommendations runs as a micro service which is exposed through REST API via Python Flask web server.

6 CONCLUSION

Thus, the BookSwap application provides its users with a convenient platform where they can share and borrow books with other bibliophiles like them!

7 ACKNOWLEDGMENT

We wish to thank our Professor, Mr. Rakesh Ranjan for his constant support and guidance throughout our project.

8 REFERENCES

- [1] How to deploy MERN stack app on amazon ec2:
instance <https://jasonwatmore.com/post/2019/11/18/react-nodejs-on-aws-how-to-deploy-a-mern-stack-app-to-amazon-ec2>
(Referred for application deployment)
- [2] Isbn-book-coverpage-images-open_api:
<http://covers.openlibrary.org/b/isbn/>
(Application development and functionality)
- [3] Datasource: <https://www.kaggle.com/jealousleopard/>
(For machine learning model)
- [4] Datasource: <https://github.com/zygmuntz/goodbooks-10k>
(For machine learning model)
- [5] https://en.wikipedia.org/wiki/Haversine_formula
- [6] Git Repo: [SISUSpring2020-CMPE272/BookSwap](https://github.com/SISUSpring2020-CMPE272/BookSwap)