

CS530 – Developing User Interfaces

Barter Books

Madhu Anumula

Computer Science

Drexel University

Philadelphia, PA USA

mla332@drexel.edu

ABSTRACT

Availability and costs make books inaccessible to many book buyers, on the other hand after reading they are usually left stacked building skyscrapers. Combining the best of both scenarios is the idea of Barter books where users once they finish their book and post its availability in exchange to get a book of his preferences. The application provides a medium for the users to browse through a list of books available along with customized search to the user that enlist books of his preference and interests.

1. Introduction

The idea behind the paper was inspired by the fact that nobody wants to read a book twice. Most of the book lovers cherish their stories but the books are left stacked in a rack losing its value, students buy high priced textbooks and they remain of no use after their curriculum. When given a thought, a person having some books unused might seem trivial but just an imagination of all those books cumulative from all over the world could give overwhelming prospective on all those stacked used books forming skyscrapers while a potential reader is inaccessible because of factors like cost, availability and more. Though it surprises, the digital transformation of books did not really disrupt readers love for physical touch of a book. Publishers of books from markets like educational books, fiction, trade made almost \$26 billion in revenue last year in the U.S., with print making up \$22.6 billion and e-books taking \$2.04 billion, according to the Association of American Publishers' annual report 2019[1]. For students, the emergence of e-textbooks though made their life easy, but the cost and adaptability often make online versions infeasible, at the same time the printed copies too come with a price distortion often pushing student's life over a barrel. This paper proposes a feasible and accessible application that provides a medium to exchange such used books among like-minded people over small proximities. Cultivating the idea of barter system from former era, users can trade their personal books with those books they are searching for from other prospective users upon negotiation, thus, expanding the range and sources of usage and reducing monetary costs of buying and selling.

The model helps users to publish availability of their books and search a catalog of books available for them to exchange that are retrieved and searched through all available sources and best matched to their preferences. It uses information retrieval system search to present one-on-one matching of the exchanger and borrower as initial best results for the user. Further, the model proposes "For You" page, a tab beside the catalog, based on the user preferences and history, recommends possible future categories and genres of user's interest that potentially could be an interest for future exchanges. Finally, when the user finds a potential party from available exchangers, he initiates a negotiation request and post acceptance a chat option is presented to both parties to agree upon the modes of exchange and complete the process.

This application contains modules and features that provide optimal results with ease in navigation, ability to customization the search using filters and communication tools to negotiate exchange and further. Along with basic filters like Title, Author, Year Published, Category etc., location filter is provided to allow a convenient exchange since users typically prefer to trade within close localities. The application presents a user profile that lists the preferences, available books for exchange and his exchange history for others to see. Features like "Will be available in(days)" option allows user to give an estimate of how soon a book of his possession would be available for exchange helping a potential searcher to save the availability.

2. Background

Mechanism design without money has been a major subject of study in economics and mechanism design. This line of research has been studied in the economics literature in the context of two-sided CS530 -Developing User Interface Barter Books Madhu Anumula Dept of Computer Science Drexel University matching markets, markets where monetary transactions are repugnant, and house allocation problems. Recently this field has gained more attention in the computer science literature due to the fact that monetary compensations are not always easily applicable. In some cases, payments are hard to implement or to collect, e.g., implementing secure money transaction systems is costly in general and some people do not feel safe enough sharing sensitive information online fearing internet fraud. Moreover, in some

repugnant markets, there may be legal or ethical issues with monetary transactions, e.g., in the case of kidney donation [9].

Motivated by concerns about money transaction on the Internet, and simplicity and convenience of swapping items in local economies, barter websites (also referred to as barter economy sites) have become more popular in the recent years. Such barter websites help users exchange items with each other. Various types of items may be exchanged in these websites: from smaller used items like books, DVDs, cellphones, or children's clothing, to bigger items like boats, vehicles and vacation rentals. Some of these sites also support exchanging services like dental work and installing hardwood flooring [9]. Numerous websites are dedicated to swapping items of various categories, such as swapacd.com, swapadvd.com, read-itswapit.co.uk, bookmooch.com, etc. [2].

User Interface (UI) is one of the most influential components of mobile applications and strongly catalyzes user's perception of the application [4]. It plays a very important role in increasing usability of an application serving as a medium for human computer interaction [3].

Applications that solemnly works on user input data result in a massive database repository and requires a constant need for optimal interaction. Performance of such match-based interfaces need robust search mechanisms and features that optimize and conjugate user interaction and adaptability. Major content-based web applications like Amazon developed their own interface strategies for optimal search results using search-oriented user interfaces. Search User Interfaces (SUI) are usually designed and optimized for generic users or for a certain user group. Users within the group are similar, e.g. concerning their information need, search goals or cognitive skills. These properties influence decisions made in the user interface (UI) design process [5]. Using SUIs, Web applications are optimized and adapted to generate user group strategies.

There are many strategies to filter the requirements based on user preferences, straight forward search based on keyword being the base and primary strategy, there are advanced mechanisms like collaborative filtering that influence the search results based on the reaction of similar users. User based collaborative filtering is one of the most widely used recommender methods. It recommends items to a user according to her similar users' opinions. The key point of user-based collaborative filtering is to compute users' similarities. In traditional user-based collaborative filtering, the similarity between two users is determined by their ratings to co-rated items [6].

Internet selling through eBay is becoming more and more popular. Except for the traders, making their living by selling products on eBay, there are lots of people trying to sell things they have and no longer need. The eBay Application Programming Interface (API) offers a set of system calls, allowing simple, but effective tools to be created. These tools provide ways to gather and provide reasonable information to the seller that aids him to construct well

defined auctions.[7] The eBay API provides methods and calls like FindPopularSerches, FindPopularItems, getMostWatchedItems, GetPopularKeywords and more that acts as diverse tools that can be helpful to sellers optimizing the search results [7].

The adoption of World of Things [WoT] has introduced new challenges for collecting, integration and processing real-world data. Among these challenges, the massive volume of devices connected to the Internet stands out. As the amount of resources that can act as data producers is increasing it is complex task to discover the most suitable producer to provide a certain information. Moreover, additional challenges arise with the heterogeneity and autonomy of data producers [8].

3. System

The aim of the application is to develop an interface that provides availability of book exchange options to the user along with customary views and options to post the user's data. Envisioning the same the application of Barter system was built in modules using prominent web developer tools available in the market.

The application opens with an introduction page that imposes a visual affect of books welcoming to the user that directs the users towards, one, a catalog view of all the books available for the user to exchange to, and then a registration gateway to start with the process. Along with them, all the pages of the application have a common thread header available all along to navigate to any page any time. Such navigational band was created using html, CSS and bootstrap. The navigational band, before logging in provides access to three pages- First being the home page itself, a consecutive next page that lists the available book and then a simple about page describing the functionality of the application. Towards the right end comes the gateway, Log In option, that takes the users into the world of barter systems. Below is a snippet of the page.



Figure 1: Home page of the web application

Clicking on the button "Register" will take the users to a login form where the user input the name, username and password, once inputted the application takes the data and saves them to create a user login database that saves these details with password in encrypted form. Further user is asked to input some further

additional profile details like address, email, book preferences and etc and finally upon successful registration user can enter his logged session. All the mentioned procedures are rendered using Python Flask and supporting data flow between the application interface and the backend flask procedures are carried with the help of Java script, JQuery.

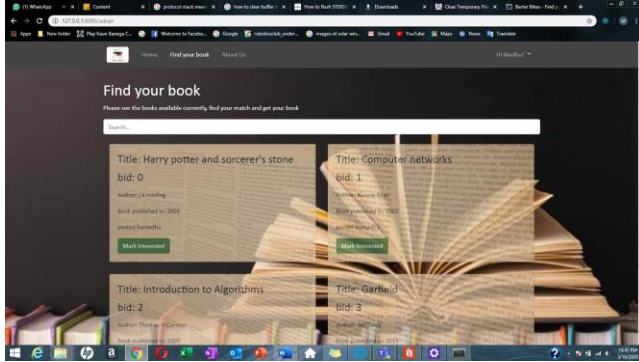


Figure 1: Catalog of books

Once the user creates his profile his inputs will be displayed in the first page after login with a side by side view of personal details and user's inputted preferences, both have an option to update or change whenever the user wants to. After this, application takes it role and populates personal tabs – '*For you*' and '*Pinned Books*', the first mentioned tab, is populated by the application based on the book preferences user has inputted when registering to the service. Further, using the books catalog page that is generic to everybody, logged in users can pin some books of their interest that will be populated in user's pinned page for future conversations, these two tabs one being populated by the application and the other by the user serves flexibility to the user in choosing exchange parties. Lastly users need to post their book that will be uploaded to the database and are made available for others to see.

All the logged in pages are styled and presented to provide a welcoming interface experience, these pages are structured, styled using bootstrap and CSS to give a smooth navigation experience. The session opens with a left vertical and top horizontal navigation bars that will be available through all pages to the users helping in easy page access at any point of time. The top bar enlists the aforementioned common tabs, while the left vertical bar provides links to pages that are specific to the user activity starting with user profile. Then the heart of the application appears titled '*Your books*' that directs to two tabs, one showing the pinned books by the user and the next populated by the system. The pinned tab is populated when the user clicks on pin button on the catalog page, the event triggers with the book id and this information is carried to the backend Flask environment and consequently to database storing the user picked books. Next based on the user's preferences given during the registration, i.e. the book, author, genre, published year, edition, books are populated for the user and again user can pin from these books directing these books to pinned books again.

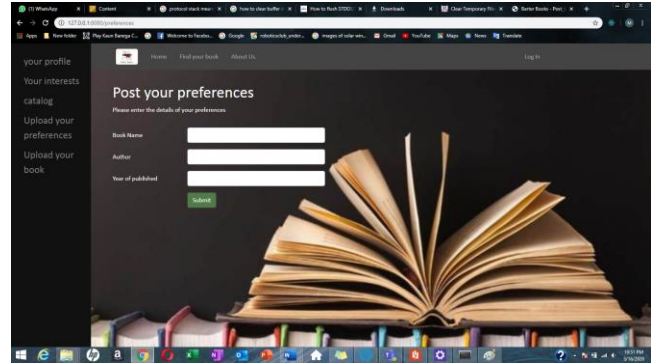


Figure 1: Form to submit preferences

These filtering of books based on preferences is done using individual preference rather than a combination to expand the scope of the search. All these pages have a common search bar to filters the based on any features, ranging to book names, authors, genres even by posted user, helping the user to skim the search results. Finally, a simple log out option to exit from the application and the application return to the home page.

4. Discussion

The application achieves the purpose of its development as it provides interface for users to post their books availability and find books of their choice, the content flow among the pages makes the interaction smooth for the user. But the application could use some features and styling that could amplify the contribution for the users, starting with more filter options in catalog pages that can give more power to the user to segregate the books and find the required easier. Many attributes could be added to the books like estimation of when the book would be available in future enabling interested user to pin the books for next exchange. An obvious more robust recommendation system can be great add to the application, although it might increase the application maintenance costs. Lastly when the application takes input from the users for personal details, book details, preference details etc., all such raw information have high chances of erroneous data which may not yield in right search results, furthermore, could overburden the database. Such scenarios could be avoided with drop down filters populated using separate database that contains possible choices avoiding using data entry.

Styling can always be improved with fancy elements any time, features like drag and dropping books between catalog to interested page would definitely improve the accessibility of the application. Adding alerts, notifications, group based sharing buying options if not available and more could be the extensions for the project.

5. Conclusion

An application for the purpose of providing books exchange choices for book readers was successfully developed using

common prominent designing tools like Flask, Javascript, AJAX, HTML, CSS. Although the application could use may more additions, it serves as a strong base for the requirement.

REFERENCES

- [1] Physical Books still outsell e-books – and here's why DOI: <https://www.cnbc.com/2019/09/19/physical-booksstill-outsell-e-books-and-heres-why.html>
- [2] Bartering Books to Beers: : A Recommender System for Exchange Platforms DOI: <https://dl.acm.org/doi/10.1145/3018661.3018696>
- [3] User Interface Design Issues for Easy and Efficient Human Computer Interaction: An Explanatory Approach DOI: <https://www.researchgate.net/publication/294428623>
- [4] Adaptation of a Search User Interface towards User Needs – A Prototype Study with Children & Adults DOI: <https://dl.acm.org/doi/pdf/10.1145/2528394.2528397?download=true>
- [5] Using inferred tag ratings to improve user-based collaborative filtering DOI: <https://dl.acm.org/doi/abs/10.5555/3106028.3106030>
- [6] Researching product tiles in eBay using the eBay API DOI: <https://dl.acm.org/doi/pdf/10.1145/1731740.1731762?download=true>
- [7] Data Producer catalogs for the Web of Things: A study on NoSQL Solutions
- [8] Characterizing and Leveraging Social Phenomena in Online Networks, ZeinabAbbass