Barter King

Project Proposal

Barter inc.

Author: Jessica Lefkowitz

Team Members:

Sunil Babu 127001211

Siyoun Yang 826006958

Humza Jaffri 626006785

Jessica Lefkowitz 926007592

Department of Computer Science and Engineering

Texas A&M University

April 3, 2020

Table of Contents

1 Executive summary

Customers have many options on where to buy their desired product. Most physical stores have gone online, and other stores have opened as purely virtual shops. With so many options, shops can sell identical products at different price points. Customers are left to choose their favorite online retailer, but this does not mean that they found the item at the best price. That is where “Barter King” comes into play. This application will give the customers an organized list of retailers to choose from. Customers will search for an item by text or using reverse image search, choose filters if desired, and view the results. The application would then show the item’s photo, description, and specifications along with a list of available retailers, ordered by the desired price or filtered item. In addition, the application would provide delivery options, available coupons, price tracking, and related products that the customer could choose from.

In technical detail, the project would use APIs from different retailers such as eBay, along with price tracking data and coupon availability with Honey and CamelCamelCamel. It would also use Google Images to help with the reverse image search. As not all APIs are public, this prototype would include specific examples that would show the complete application. If funding is granted, more APIs could be added, and so the database would grow and include more retailers, thereby making the lives even easier for the customers. Once complete, the website would provide an organized list of sorted retailers that the customer could choose from. They would know which site sells the product at the cheapest price with the best delivery option that works best for them. This would save the customer time from searching through the websites themselves along with money from then buying the cheapest product. “Barter King” would not sell the products themselves. It would instead provide links to the corresponding retailers for the customer to explore. Instead of searching through the vast Internet to buy a single product, customers can use one application that does all of the work for them.

2 Introduction

2.1 Problem background

With the incredible boom of the Internet, everyone and everything moved online. Retailers added virtual stores to their repertoire, and shoppers know that if they cannot find the product they seek in store or if they do not want to physically go to a store, they can order it online. Online-only stores joined the ranks and shoppers have many options to choose from. However, many retailers sell identical products for different prices, and the customers do not know which store to buy the cheapest product from.

2.2 Needs statement

As the growth in online stores leaves customers with many options of which store to buy from, it also leaves them confused on where to find the cheapest product. Many people instead pick a store and stick with it, not knowing that the prices might be significantly cheaper somewhere else. They are then left having spent extra money that could have been used elsewhere.

2.3 Goal and objectives

The uncertainty that comes with the question on if the chosen retailer is the best retailer can be solved with the use of the “Barter King” application. As with any online store, a customer can search for their desired product, and the application will bring up the result along with its photos, description, and specifications. They would also have the option to reverse image search by inputting just a photo of the product that they desire. It will then show a list of retailers that sell this product, ordered by price to show the cheapest option. If customers had a bad experience with specific retailers, they can choose to filter them out during their search. In addition, when applicable, the application will also include a price tracking chart to determine if the price is good relative to other times in the year for that specific retailer, delivery options and prices, available promo codes and coupons to give the best price possible, and other related products for customers

to choose from. With Barter King's use, customers will be able to find the best retailer for their specific needs at just one click of a button. The specific details that will be used to determine the best price and option for each product will also cater towards the websites and APIs used in order to give customers the best options for the specific choices available.

2.4 Design constraints and feasibility

The main constraint that will hold this project back is the lack of public APIs. The website will be most successful if it could include as many retailers as possible, so it would need to be able to access the information for each of those other websites. This constraint could affect the project objectives for if the APIs for the main online retailers are not public, it will only give customers specific options to choose from. However, it will complete the job by giving the cheapest option among the available retailers. In addition, due to the Coronavirus, the team must meet each other and work solely virtually, which will add a pressure to management.

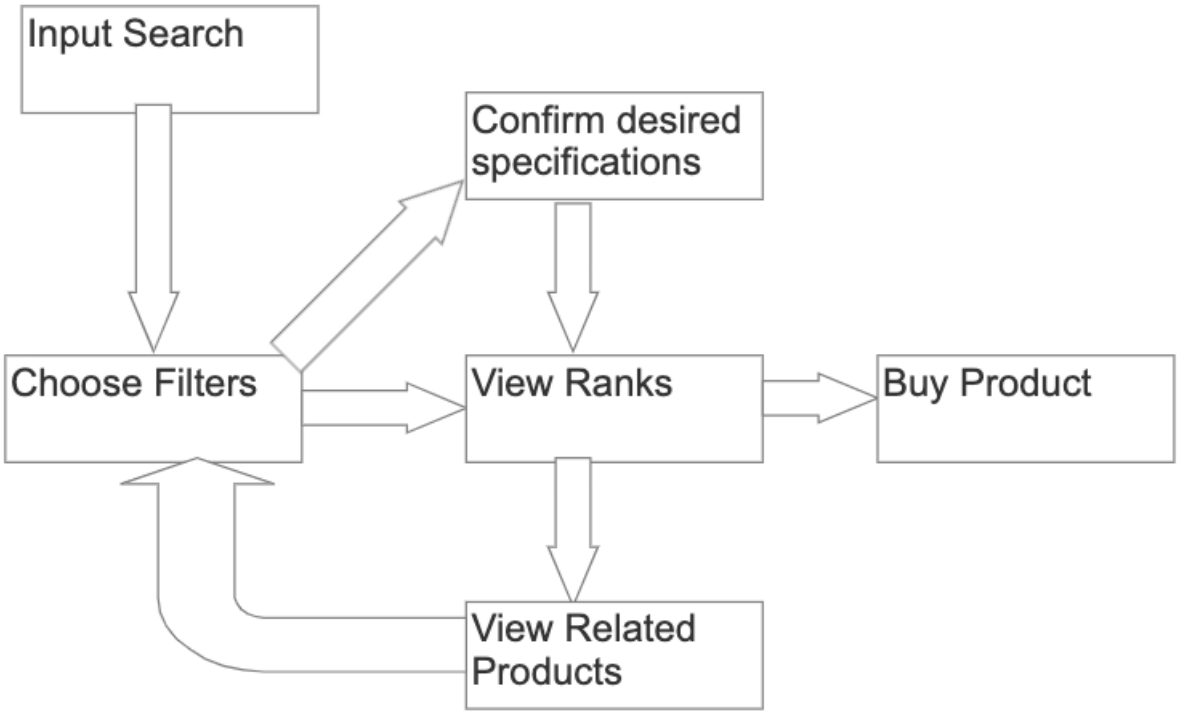
3 Proposed work

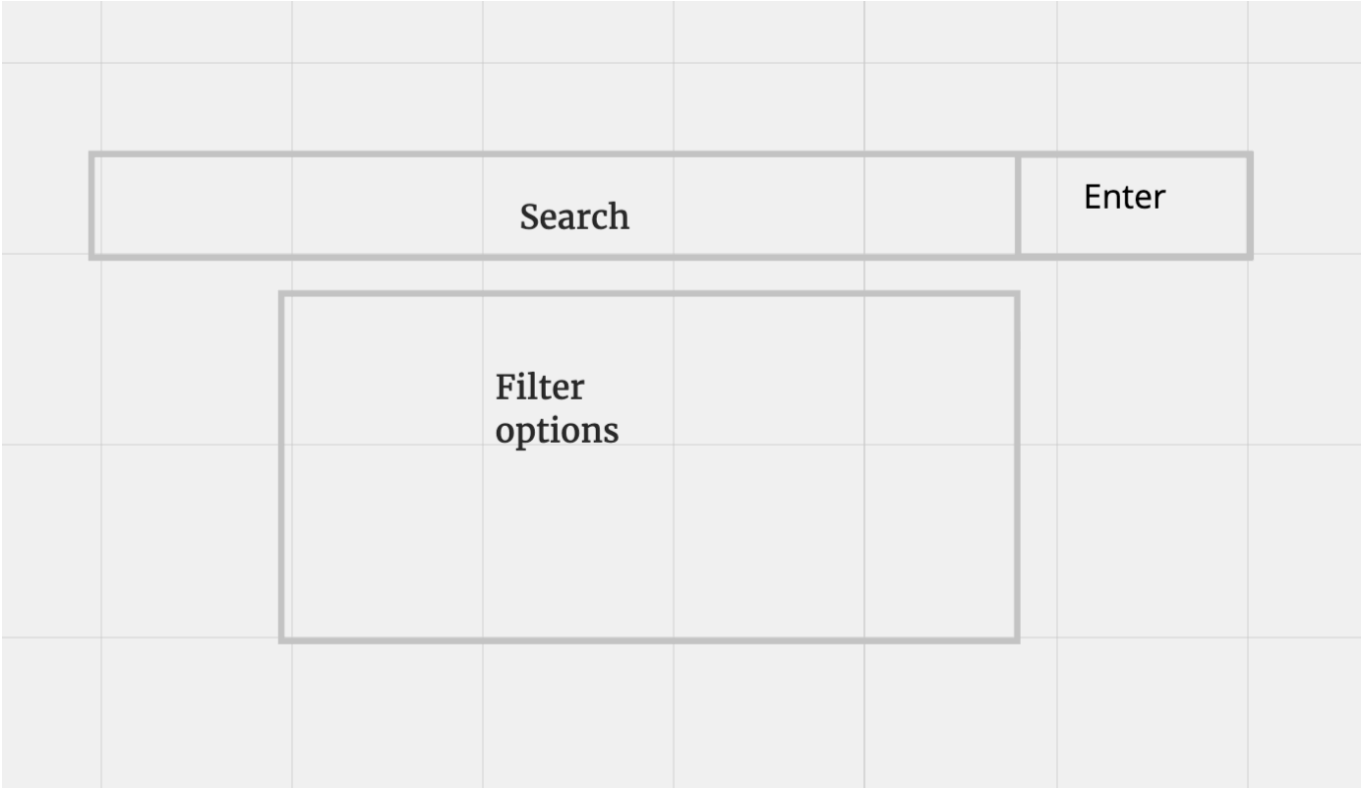
3.1 Evaluation of alternative solutions

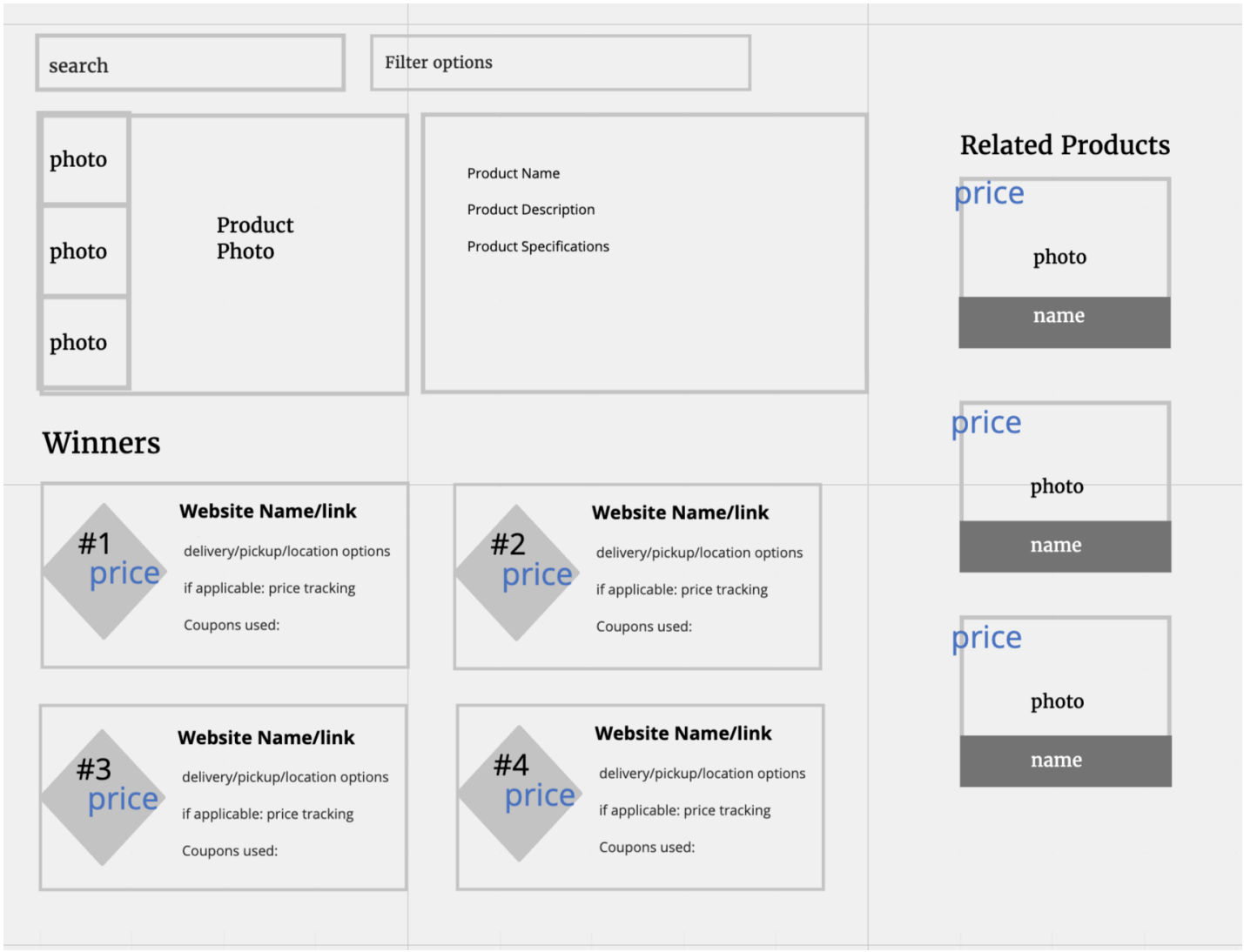
The first alternative solution will cover the issue on if the APIs we desire are not accessible for this project. In this scenario, we will find other APIs that would still give the price ranking and the general idea of the application's purpose. If the project is chosen for future funding, it could then switch to the properly desired APIs very easily.

The second alternative solution will be used if specific features do not work as planned. In this scenario, the website would need to adjust accordingly to add different features that would not fall under the same category that inhibited the original ideas. If the reverse search image or price tracking do not work, the project team will try to add something else to take its place and present an equally appealing product.

3.2 Design interface sketch







The above photos show a block diagram of the customer's experience, a home search page, and the graphical user interface of the results page. The customer would start on the home search page and input the product that they want, either through text or a photo. They could then choose filters on the possible retailers and display order of the results. Next, customers could view the product description and specifications to confirm that this is their desired result. The list of available retailers would show up in order, and it would give the extra information that the customer would want such as delivery options, possible price tracking, and coupons. In addition, the application would provide related products that the customer might be interested in, and they can click on those products to learn more. The customer cannot buy the product from this application, but it would provide the links to the corresponding site. These features would be created by the use of a database and cookies to store the information, sorting algorithms to give the desired results in the correct order, and API access to attain the information from the other websites.

4 Engineering standards

4.1 Project management

Humza Jaffri has experience with leadership, system design, and software development from participation in hackathons, an internship, and class projects. Sunil Babu holds experience in web design, software development/design, database design, and basic machine learning. Siyuan Yang has experience with web application design, software design, database, and team leadership. Lastly, Jessica Lefkowitz has experience with AGILE, project management, and testing from a previous summer internship and software design and development from school projects. Since each team member holds specific qualifications, the roles were assigned to fit each person. Jessica Lefkowitz is the project manager and team leader and works on the user interface, Sunil Babu and Humza Jaffri are in charge of different parts of the systems design such as the database and API access, and Siyuan Yang is in charge of the software design. In order for members to get a variety of responsibility type, the team will approach the other areas of the project and determine who will lead them and who will assist as they come up. In addition, while one person is in charge of a specific area, the team will work together to assist the leader in each area. For example, one person will be in charge of the user testing data, but the entire team will share responsibility in attaining and analyzing the data.

The team leader calls at least three scrum meetings a week and keeps the team on track through encouragement of constant communication. The scrum meetings are set to take place Tuesday and Thursday mornings at 9:30, and Friday morning at 11:30. The team also meets to work during the lab time on Tuesday and Thursday evening and continues the meeting until the goal for that day is complete. In addition, the team holds a retrospective meeting at 11:30 on Saturday mornings and moves on to preparation for the upcoming sprint. All official meetings are held over the Zoom application. Additional meetings are set as needed, and the team is in constant communication over the group chat to answer questions, brainstorm ideas, give individual updates, and make plans.

4.2 Planned Scope

The overall goal of the project is to provide an organized method for customers to search for a product and view a ranking of the cheapest retailers available, along with product information, delivery options, related products, photos, reverse image search, and similar objectives. This would use APIs such as eBay for an example retailer, Honey for available and possible coupons to find the best price, CamelCamelCamel for price tracking, and other API examples that would fall under the same categories, like Google Images for the reverse image search capability. The fall back would be a working website with the price rankings but not including the extra information. This fall back would give the customer the cheapest retailer, but it would not include the extra details that the customer could potentially seek. The stretch goal would include additional APIs that would enhance the database and compare prices across more area and use more retailers. It would also potentially be mobile friendly.

4.3 Product and sprint backlog

The project was split into an initial product backlog to turn the entire project into smaller tasks, and each section was given an expected time frame to work on it for the burn-down chart. A summary of these tasks includes creating an initial drawing of the UI, coding it with each component, creating a database, and connecting to the APIs. The components then need to be brought together to create the website. The additional desired features would be added, and the product would be tested. The first sprint backlog includes the initial steps of the project. Splitting up the overall tasks into these backlogs help show a checklist of everything that needs to get done for the project. This helps both with time management and ensures the proper progress as the sprints go by. Both the product and sprint backlogs can be found in the appendix, along with the burn-down chart.

5 References

No documents were used for this part of the project that need to be referenced. Some possible websites that will be used for assistance in the development will include https://stackoverflow.com/ and https://www.tutorialspoint.com/index.htm. All references will be cited in the location that they are used.

6 Appendices

6.1 Product backlog

1. Design the UI
2. Create the dialog box for the search box
    a. Textfield
3. Create the search button
    a. Create an action listener for the button
    b. Implement functionality to send a stored text to the server for API calls
4. Create the filter
    a. Filter by price
    b. Filter by category
    c. Filter by seller
5. Create a ranking algorithm
    a. Sort by Price low to high
    b. Sort by Price high to low
    c. Sort by Review Score
6. Create a database
    a. Store the product information
    b. Store pricing, pictures
7. Access APIs for data (Repeat for multiple APIs)
    a. Search for product
    b. Get product names
    c. Get product reviews
    d. Get product images
8. Product Description
    a. Textfield
9. A few photos of the product
    a. Image
10. Product Reviews
    a. Textfield

11. Product Price Information
    a. Textfield
12. Related Products
    a. Textfield
13. Coupons
    a. Use Honey for coupon comparison
    b. Get coupons from various APIs
14. Testing to ensure we obtain desired results
    a. Unit testing
    b. Integration testing
    c. System testing
15. User study testing
    a. Create user stories
    b. Create a testing process
    c. Conduct study
    d. Compile data
16. Price tracking over time
    a. Implement Camelcamelcamel API
17. Delivery Information
    a. Shipping prices and times
    b. Pick up locations
    c. Delivery services (such as favor, postmates…)
18. More buying options
    a. Textfield
19. Specifications
    a. Textfield
20. Feedback
    a. FAQs

Sprint Backlog

1. Design the UI
2. Create the dialog box for the search box
    a. Textfield
3. Create the search button
    a. Create an action listener for the button
    b. Implement functionality to send a stored text to the server for API calls
4. Create the filter

    a. Filter by price
    b. Filter by seller
5. Create a ranking algorithm
    a. Sort by Price low to high
    b. Sort by Price high to low
    c. Sort by Review Score
6. Create a database
    a. Store the product information
        i. Product Description
        ii. Product Reviews
        iii. Product Price Information
        iv. Related Products
    b. Store pricing, pictures
7. Access APIs for data (Repeat for multiple APIs)
    a. Search for product
    b. Get product names
    c. Get product reviews
    d. Get product images

6.2 Burn-down chart

This chart shows the expected remaining time that each task will take.