Grocery Deal Hunter Application: Finding Ways to Cut Costs

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Sanha Kim

Spring, 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

Grocery Deal Hunter App

CS4991 Capstone Report, 2023

Sanha Kim
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
sk7pr@virginia.edu

ABSTRACT

The nationwide increase in the prices of basic necessities, especially gas and food, have created financial hardship for American families. The proposed solution to this is the development of a mobile app that can scour through nearby grocery stores' apps to find which items to buy at which stores to get the overall lowest cost, including gas costs. An iterative UX wheel method would be used to approach this problem, and it would be developed in Kotlin & Swift. The anticipated outcome is an app that would help people save money on groceries. Future work may include receiving feedback from users to improve the app by making it easier to use or adding new futures.

1. INTRODUCTION

Most people in the US live paycheck to paycheck, so a few dollars a week saved on groceries and gas would be an immense help. Grocery stores have hundreds to thousands of different items, all priced differently than other grocery stores. It is impractical to keep track of all of them, especially when the prices can change on any day. These stores cannot always carry everything, so a shopper would need to go to multiple stores to get everything they need. How are they supposed to keep track of the differing prices and availability between those stores to ensure they get the best deals? And how are they supposed to know whether the fuel costs and travel time justify going to a different grocery store for the lower prices? The proposed Grocery Deal Hunter app would help people save both time and every cent possible while grocery shopping.

2. RELATED WORKS

According to Kathy W. Warwick (2022), a similar app, Instacart, focuses on customer convenience more than customer cost. It scrapes through grocery stores' information and offers delivery and pickup from those stores with an additional fee instead of providing the customer the cheapest options. I chose to give the customer the additional information instead because going there personally is generally cheaper than delivery. I also chose to show prices between stores so the customer can more easily tell where they should shop.

Hartson & Pyla (2012) detail the UX Wheel design process, focusing on the importance of an iterative approach to UX design. They identify the main stages as Design, Prototype, Evaluate, and Analyze and break each of these into smaller steps. The design of the app involves following this design process.

3. PROJECT DESIGN

This section will describe the process of designing the Grocery Deal Hunter app, including a review of the systems requirements, components, internal logic, and overall design philosophy.

3.1 Review of System Requirements

Below is a list of the requirements the system must meet. Broad requirements are enumerated, and specifications are in sub-lists underneath the broad requirements.

1. The app will need to find grocery stores within a certain radius of the user.
   a. If the user has a different start and end point, the center of the search radius would be the middle point between the start and end.
2. The app will need to receive user input of a grocery list and return a list of the cheapest choice from all the supermarkets.
   a. The user must have the option to block certain brands or stores from appearing.
   b. The user must be able to limit the maximum number of different stores to go to, with a default of three stores.
3. The app will need to calculate final cost of a grocery list including gas prices.

3.2 Interaction of System Components

For the mobile application, the only required component is the app on the Android phone itself. The user would create a request on their app, which would then be sent to the server. The request would be comprised of a shopping list, the user's starting and ending location, maximum number of stores to visit, maximum travel distance, mode of transport, and cost of transport. The server would receive that request, and search its cache to check whether it already has up-to-date information. If not, it would send an API request for data on prices of nearby, and another API request for routes. Then it would run calculations on the data retrieved from the APIs, and send the results to the client-side application.

3.3 Algorithm for Location and Prices

Data needs to be collected from the plethora of different grocery stores in the country in order to show accurate information reliably. It would normally be a massive undertaking to get the data from each of these stores. Luckily, there is an aggregate API for grocery stores, MealMe, which automatically searches grocery stores for a product within a given radius of a location.

In order to calculate the optimal route between a given start and end point with an indeterminate number of stops in between, the distance between each stop needs to be known in advance. Since roads are rarely a direct route between two points, it is impossible to calculate the actual distance travelled from the latitude and longitude alone. It is possible to send a Google Maps API request for the routes between each individual location every time, but sending so many API requests would take a long time, and incur high costs. Therefore, there would need to be a database on the server which stores the information permanently after a Google Maps API request. Fortunately, grocery stores rarely move, so the database should only need occasional updates. There are only around "40,000 stores* that sell grocery items in the US", so the database wouldn't need to keep track of too many distances between stores (FoodIndustry, 2019).

There needs to be a limit of how far grocery stores are from each other to store the distance relation in a database since most relations would never be used, and storing every relation would mean 1.6 trillion data entries. I decided the limit should be 25 miles since "the average American commutes 41 miles a day to and from work," or about 20.5 miles one way (Flynn, 2023). The extra five miles is padding for people who have farther than average commutes. Although it is difficult to find exact numbers for the density of grocery stores per square mile, "researchers found that in 2015, the median distance to the nearest food store for the overall U.S. population was 0.9 miles, with

40 percent of the U.S. population living more than 1 mile from a food store" (Rhone, 2019). This seems to suggest that the average store would not have more than 25 stores within a 25-mile radius, so even in the worst-case scenario, only a million relations would need to be stored.

To construct a cheapest list between stores, the server will first create an array of Shopping List objects to store the prices at each grocery store. It would then search a separate database for prices of items for each store. Prices are time-sensitive information since they are constantly in flux, so this database would act as a cache, and automatically disregard information as outdated after a set period. If up-to-date information is not found in the database, the server will then send an API request to MealMe for each item. The MealMe API requires, among other information which remains unchanging for our purposes, latitude and longitude, a query term, and a max distance in miles (MealMe, 2023). Each of these fields will be fed information from the client's request. The response will contain a list of products from stores in the specified radius, listed from cheapest to most expensive. The price, name, store address, size, and other data relevant to display to the user also comes along with each product. The size measure would be used to calculate size per unit since otherwise, the cheapest item might just show the store which sells the smallest amount of an item. After filtering out the user's specified banned brands and stores, the relevant data from each request would be added to each store's Shopping List. Then, the server will send a distance-matrix request from the user's location to each of those stores to Google Maps API, which accepts an origin point and "one or more locations to use as the finishing point for calculating travel distance and time" (Google, 2023). Distance information is needed at this point to calculate the cheapest price since gas price will also be factored in.

The shortest path can be found by treating the start point, end point, and stores as nodes of a graph, and the distances as weighted edges between them, as seen in Figure 1. Then, we can do a depth first search starting from the start node, and set a requirement that the path must travel through all of the nodes first before reaching the end node to qualify as a valid path. For cases in which the start and end positions are the same, we can just make the end node and its edges a copy of the start node. If in the example shown in Figure 1 the client specified they would visit a fewer number of stores such as two, the shortest path can still be derived from this graph. Each of the store nodes would take turns being set as visited before the algorithm starts, making the algorithm only visit the other two nodes before reaching the end point. Then, we would take the total mile length of the trip and use the user-provided miles per gallon to calculate how much fuel they would spend on the trip, and add it onto the price for visiting those stores.

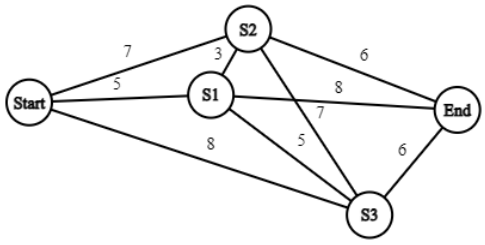

Figure 1: Destinations represented as Nodes.

No algorithm can simplify finding the cheapest combination of stores especially since each new group of stores would have a different mileage associated with it. Therefore, to find the optimal combination, there is no choice but to check each possibility by brute force. However, since none of the Shopping Lists would change while searching for the cheapest option, this problem can be sped up by multithreading. After the results are calculated, the top few

combinations of stores, separate shopping lists for each store, final prices, and total travel time will be sent to the client and displayed on their application.

3.4 Design of Interface

An intuitive user interface is necessary with the amount of information the user has to be able to understand and use. The UX Wheel Design process would work well for this project. The UX Wheel is an iterative approach to design with four main steps: analyze, design, prototype, & evaluate. Those main steps can be briefly summarized as understand user work and needs, create interaction design concepts, make a testable model of design concepts, verify and refine those ideas, then repeat (Hartson & Pyle, 2012, p. 90). The steps are flexible, but the important point to note is the iteration. Iteration in this project is required since "the design domain is so vast and complex that there are essentially infinite design choices along many dimensions" (Hartson & Pyle, 2012, p. 90).

4. ANTICIPATED RESULTS

The Grocery Deal Hunter App would help American consumers reduce how much they are spending on groceries every week. When the app quantifies how long the trips between stores would take, and how much it could cost for gas, it could discourage users from taking long car trips which might be costing them more money.

The app would increase the ease of access to information across stores. Currently, it is difficult to know how much an expensive food like beef costs in an area. With the app, a shopper might decide to check the prices of food in their area, and wait to buy an item until their trip to another store, even if they do not do it immediately. A wide enough adoption of the app could force grocery stores to lower their prices to stay competitive with nearby stores and retain customers.

5. CONCLUSION

The Grocery Deal Hunter App's main benefit would be helping shoppers save time and money. People could save some money every week to save for bigger purchases, or even indulge in small luxuries every week. It will be their choice what they can do with money. In this age of information, it only takes a second to learn about celebrity drama, but we do not even know how much we are losing out on savings by not going to the grocery store just five minutes away. This app aims to change that and give power back to the consumer.

6. FUTURE WORK

The actual implementation of the app described in this report is the biggest work that is needed in the future. After a working version has been made, there also needs to be user testing for feedback. This feedback would be used to see how the app could be improved in the future.

REFERENCES

Flynn, Jack (2023). 15+ Average Commute Time Statistics (2023): How Long is the Average American Commute? Zippia, Inc. https://www.zippia.com/advice/average-commute-time-statistics/#:~:text=The%20average%20American%20commutes%2041,workers%20drive%20alone%20to%20work.

FoodIndustry (2019). How many grocery stores are there in the United States? FoodIndustry.com. https://www.foodindustry.com/articles/how-many-grocery-stores-are-there-in-the-united-states/

Google (2023). Distance Matrix API request and response. Google. https://developers.google.com/maps/documentation/distance-matrix/distance-matrix

Hartson, Rex and Pyla, Pardha (2012). The UX Book: Process and Guidelines for Ensuring a Quality User Experience. Elsevier, Inc.

Rhone, Alana and Ploeg, Michele Ver. (2019) U.S. Shoppers' Access to Multiple Food Stores Varies by Region. USDA. https://www.ers.usda.gov/amber-waves/2019/june/us-shoppers-access-to-multiple-food-stores-varies-by-region/

MealMe. (2023). https://docs.mealme.ai/

Warwick, Kathy W. (2022). Instacart Review: Pros, Cons, and Is It Worth the Price? Healthline Media LLC. https://www.healthline.com/nutrition/instacart-review