Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Fall 2021

UTA STeam

UTA STeam
Shopping Aide

Patrick Faulkner
Hozefa Tankiwala
Andrew Hands
Kiran Karki
Uyen Do

Revision History

Revision	Date	Author(s)	Description
0.1	09.17.2021	AH	document creation
0.2	09.24.2021	PF, UD	first draft
1.0	10.01.2021	AH, PF, HT, KK, UD	complete version 1
2.0	12.03.2021	HT	Changes required from other documents

Contents

List of Figures

1 Problem Statement

This Project Charter establishes an agreement between UTA STeam and Tim Dockins to develop software to aid groceries shopping by finding the optimal route with the most savings. The existing software can only fulfill partial requirements of finding the lowest prices of the items but do not take into consideration of time and distance saving by suggesting the optimal route. UTA STeam is to develop software that accomplishes both tasks.

2 Methodology

UTA STeam is going to build a web-based application. The application will search for the items input by a user based on preference such as locations and brands to suggest the best places to shop these items with the lowest prices and fastest route. UTA STeam will utilize the Agile methodology which allows frequent alteration during the development process to accommodate continuous client feedback.

3 Value Proposition

Technology has revolutionized all platforms including shopping. Furthermore, the thriving of web-based applications has significantly changed the way people shop today. Shoppers do not just go to the store and buy what is available there anymore. Now, they have an option to go online and check for availability and prices before they go. Many applications help with grocery shopping; however, because the shoppers have many choices, it could be overwhelming and time-consuming since they have to look through many deals to compare and decide. There is also a high chance that the selected one might not be the best deal. This is when this application becomes helpful and necessary. The application will take into consideration of time, distance, and cost to suggest the optimal choice. By doing this, the application will help users to save money, time and avoid frustration such as going through bad traffic just to arrive at the store that do not carry the products that they want. Because this application will greatly benefit users, it will soon attract a lot of users, sponsors, and investors. It can also be expanded to get subscriptions, referral marketing, advertising, collect and sell data.

4 Development Milestones

The project is award on September 15, 2021 and the deadline to release is by April 21, 2022.

- Project Charter first draft - October 4, 2021
- System Requirements Specification - October 25, 2021
- Architectural Design Specification - November 15, 2021
- Demonstration of wireframes - December 6, 2021
- Demonstration of non-functional front-end - February 18, 2022
- Detailed Design Specification - February 21, 2022
- Demonstration of <feature 1> - March 11, 2022
- Demonstration of <feature 2> - April 1, 2022
- CoE Innovation Day poster presentation - April 18, 2022
- Demonstration of <feature 3> - April 18, 2022
- Final Project Demonstration - April 21, 2022

5 Background

The client, ***Tim Dockins***, is an alumnus of the University of Texas at Arlington. Tim was trying to buy ingredients for his weekly meals and after going to a couple of stores, he still could not buy all of the ingredients he needed. That was when Tim realized that it would be very helpful if there is a groceries shopping application that can tell him where he can get what he needs with the most savings and shortest trips.

Currently, Tim uses the Alexa application to keep his shopping list but he does not use any shopping application to help with comparing prices or distances. Tim acknowledged that many shopping applications help to find sales and coupons; however, no application can combine all aspects and suggest the optimal choice regarding both savings and distance. Therefore, this is a good opportunity for the team to build an application that can solve the problem, gain users and bring value to the community.

The client does not have a preference for technical requirements such as a platform or database. However, the client wants to be able to access the application on multiple platforms. The application will allow manual entry to search for items and have a shopping list component. A user will have an account to save his or her shopping list and other settings. The application can access multiple stores and has a map component to navigate users to the stores. The client does not have a preference for the favorite stores, any nearby stores that have the items are accepted. Searching items by brands is not the client's priority. The desired component is to have in-application recipes to pick from or recipes imported from another website. Regarding the documentation and manual, Tim prefers the in-application manual and wants to have access to source code and documents but does not have a preference of how they should be done.

6 Related Work

Currently, no application fully satisfies the clients' requirements. Many shopping applications satisfy a part of the requirements. The most popular applications are as below.

- ***Flipp*** [4] and ***Coupons*** [6] keep track and display local stores' sales and coupons so that users can shop these items at lower prices. The application also allows users to clip and save the digital coupons inside the application to scan them at actual stores.

- ***FetchRewards*** [3] ***Ibotta*** [5] and ***Rakuten*** [1] allow users to see which products are on sale and give cashback. The users can shop for these products and scan the receipts to get cashback inside the app. Some applications will give cashback as rewarding points while others will give actual cash.

- ***Honey*** [7] allows users to search for items and display stores that carry the items and their prices with different brands. The application also allows users to sort and filter the available options with prices, brands, colors, categories, and stores' names.

- ***Basket*** [2] allows users to enter zip code to set up default nearby stores then the users can enter a keyword to search for grocery items. The application will display the default stores' available brands with their prices and the distance to each store.

Among those available applications, Honey and Basket are most like the requirements; however, they both have their limitations. For example, Honey does not calculate the distance to the store, the user will be redirected to the actual store website to complete the shopping and it is not for groceries. Honey focuses more on retail products such as electronic devices, furniture, etc.

Basket seems to be closest to the client's requirements, but the user interface is confusing because there are many hidden sub-menus. The application also limits grouping the shopping items into two stores at max and left out the unavailable items; therefore, the user must figure out what to do with the remaining items. There are a lot of unavailable items and replacement suggestions since the application shows all available brands in default stores and there is a high chance that the selected brands are not in the two grouped stores.

7 SYSTEM OVERVIEW

Our challenge for this application is to find the best available deals for the items in a given shopping list. It will also be difficult to synchronize user profiles over various platforms, as indicated by the client wanting to be able to input recipes on one device, and using it on another. Furthermore, there will be difficulties with concurrency issues regarding keeping prices up to date and managing recipes in databases.

The application will generally be broken into four parts: the front end, the back end, the database, and the data collection system. The following is a brief overview of each system.

- The front end of the application will have to be able to present fairly complex types of data, in terms of recipes, search results, application activity, and navigational information.
- The back end will have to handle user queries, such as item location/price searches and user profiles. For the searches, it will be a relatively difficult task as the user will create a profile to indicate relative importance of distance versus cost for items, thus making it a nontrivial optimization problem. In practice, there are various ways of solving this, but they will require a fairly computationally capable back end.
- The database will need to store all user information, such as shopping preferences and all recipes. For the recipes, the application will require relational data as each recipe will have at least one step and ingredient, requiring the use of at least three tables.
- There will also be a data retrieval system, which will likely be a combination of web application programming interfaces (APIs) and web scraping tools. This is highly dependant on the terms of service for the grocery stores being considered, as some of them allow web scraping while others only allow for access to be done through their API.

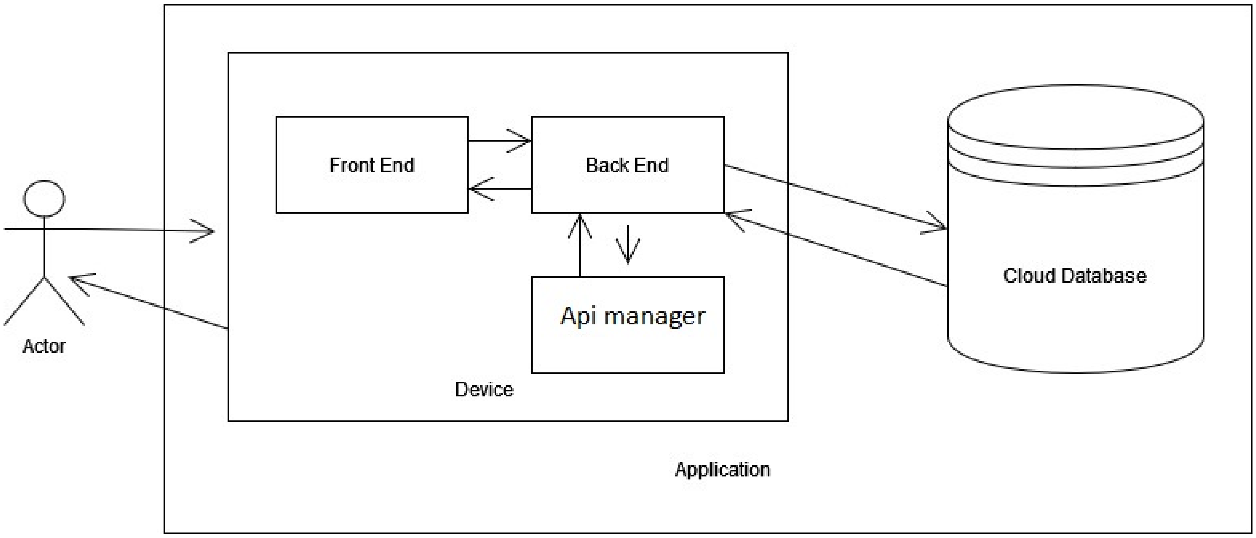


Figure 1: High level overview of the system

Another item that will have to be considered is security and authentication. Since users will want their profiles to persist across platforms, it will be necessary to have some sort of login mechanic. For a rough outline of what the system architecture will look like, consider figure 1.

8 Roles & Responsibilities

The stakeholders for this project are our software development team, the customer and the users of this software.

Tim Dockins is the customer for this project and he will be the main point of contact.

The team members for this project are Patrick Faulkner, Kiran Karki, Uyen Do, Andrew Hands, and Hozefa Tankiwala.

For the initial phase of the project all members will research the technology required for the project and complete the project charter. Patrick will be the team lead. Andrew will compile our final charter on over leaf and submit it. For the development phase all members will act as software engineers. We will most likeley maintain the product owner and the scrum master

9 Cost Proposal

As this is a software project with no hardware component most of our expenses will be for licensing and hosting the application. We will be hosting the back-end of the application on AWS (EC2) which would be another one of the major costs. We would also need to buy a domain name for the web application part of the project. The cost for these items should be covered by the senior design spending budget and AWS costs should be covered by the credit offered to UTA students by AWS.

9.1 Preliminary Budget

Sr. No	Item	Cost ($)
1.	AWS EC2 Hosting	200
2.	Domain Name	20
3.	API's	200
	Total Cost	420

Table 1: Overview of the cost proposal

9.2 Current & Pending Support

The primary source of funding for this project is the default $800 budget provided by the CSE department for senior design teams. As this project is not sponsored, there is no external funding for the project. We would also be using the $100 credit which is provided to each member of the team by The University of Texas at Arlington for using AWS services. There is no other support, expected or needed, pending.

10 Facilities & Equipment

- Facilities:
  As this project is completely independent of any hardware components, the team does not require any lab space, makerspaces, or testing grounds. For the purpose of the team meetings and to discuss sprint progress the team has reserved a room in the Engineering Library for the rest of the semester.

- Equipment:
  The only physical equipment that will be required for the project will be the Windows PCs, and Apple Mac to test and develop the application.

- Collaboration:
  The team will be using Microsoft Teams for continuous team communication, GitHub to share the source code for the application, and Jira to track all project sprint progress.

11 Assumptions

Throughout the development lifecycle of this application, it is highly likely that many new issues and assumptions will arise. However, at present the following is a list of known assumptions for this application and its development process.

- No members of the team will be lost.
- Time and schedule availability will persist for the duration of the project.
- The application will not have to be maintained after the final developmental increment.
- Developmental platforms will remain relatively constant.
- Regional and userbase scaling will not be a major issue during the lifetime of the application.
- The stores will provide us with the latest product information such as price and availability
- All users will have email addresses.
- Users will visit the stores in the consecutive order suggested by the application.

12 Constraints

Consider the following list of developmental and deliverable constraints for this project. This list is not exhaustive, but does contain the main identified constraints for this project.

- Final prototype demonstration must be completed by April 21, 2022.
- Total development costs must not exceed $800.
- Only stores which have been confirmed to allow scrapers or have affordable API's will be used within the app.
- Generally only major stores will be considered.
- The web-application should be compatible on mobile devices.
- Lists and such should be transferrable through the cloud.
- Users should be able to add their own recipes.

13 Risks

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	Probability	Loss (days)	Exposure (days)
Change in TOS for grocery store website used	0.15	5	0.75
Network issues requiring reformulation of app	0.05	9	0.45
Oversaturation requiring migration of service to a different platform	0.10	20	2.0
Difficulty acquiring pseudo data, requiring manual entry	0.40	4	1.6

Table 2: Overview of highest exposure project risks

14 Documentation & Reporting

14.1 Major Documentation Deliverables

14.1.1 Project Charter

Version 1 of the Project Charter will be delivered on October 4, 2021 with Version 2 being delivered on December 6, 2021 and the final version being delivered on April 21, 2022. The Project Charter will be reviewed every week during team meetings and will only be updated if four out of five team members agree on the update. The team will review and approve the Project Charter in the meeting that most immediately precedes the Charters due dates.

14.1.2 System Requirements Specification

Version 1 of the System Requirements Specification will be delivered on October 25, 2021 with Version 2 being delivered on December 6, 2021 and the final version being delivered on April 21, 2022. The System Requirements Specification will be reviewed every week during team meetings and will only be updated if four out of five team members agree on the update. The team will review and approve the System Requirements Specification in the meeting that most immediately precedes the System Requirements Specification due dates.

14.1.3 Architectural Design Specification

Version 1 of the Architectural Design Specification will be delivered on November 15, 2021 with Version 2 being delivered on December 6, 2021 and the final version being delivered on April 21, 2022. The Architectural Design Specification will be reviewed every week during team meetings and will only be updated if four out of five team members agree on the update. The team will review and approve the Architectural Design Specification in the meeting that most immediately precedes the Architectural Design Specification due dates.

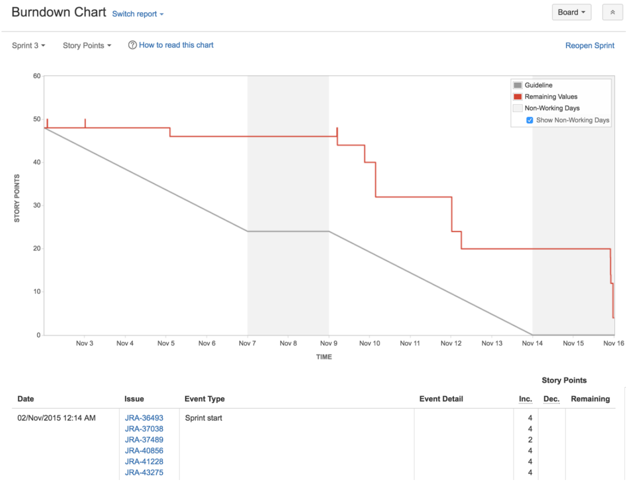
14.1.4 Detailed Design Specification

Version 1 of the Detailed Design Specification will be delivered on February 22, 2022 and the final version being delivered on April 21, 2022. The Detailed Design Specification will be reviewed every week during team meetings and will only be updated if four out of five team members agree on the update. The team will review and approve the Detailed Design Specification in the meeting that most immediately precedes the Detailed Design Specification due dates.

14.2 Recurring Sprint Items

14.2.1 Product Backlog

Items will be added to the product backlog based on requirements from the customer and will only be done as a group. No members will be permitted to add items without the rest of the teams knowledge and approval. Priority will be given to backlog items based on the customer's input. All other items will then be given priority by the team through majority vote of four to one. The teams backlog will be maintained and accessible to all members via Jira.



				Story Points		
Date	Issue	Event Type	Event Detail	Inc.	Dec.	Remaining
02/Nov/2015 12:14 AM	JRA-36493	Sprint start		4		
	JRA-37038			4		
	JRA-37489			2		
	JRA-40856			4		
	JRA-41228			4		
	JRA-43275			4		

Figure 2: Example sprint burn down chart

14.2.2 Sprint Planning

This project will have eight confirmed sprints with the possibility to add any during the winter break. All sprints will be planned by the team on the Fridays immediately preceding the beginning of the sprint.

14.2.3 Sprint Goal

The customer will be given the option to set the team's sprint goal(s). In the case that the customer does not provide any, the sprint goal(s) will set by the team during the sprint planning meeting. These team set goals will be based on the priority of product backlog items and due dates of required documentation.

14.2.4 Sprint Backlog

The team's sprint backlog will be set before every sprint. The sprint backlog will be set based on the priority of items on the product backlog as well as estimated person hours needed to complete those items. The team will maintain the sprint backlog along with a scrum board via Jira.

14.2.5 Task Breakdown

Items from the sprint backlog will be assigned on Jira on a volunteer basis. This will allow for members to volunteer for items that would provide them an opportunity to learn something new while also creating the ability for team members to take on items they already know how to do when time becomes a bigger factor.

14.2.6 Sprint Burn Down Charts

The team manager will be responsible for the sprint burn down chart. The manager will use Jira to populate the burn down chart, but all members will be responsible for updating the progress of their assigned backlog items and logging the hours they put into completing those items. See Figure 1 for an example.

14.2.7 Sprint Retrospective

The sprint retrospective will be created by all team members during the sprint review session that will take place during the Friday meetings at the end of each sprint. During these sprint review sessions, all team members will discuss the status of each sprint backlog items, the person hours put into them, and the team manager will populate the sprint burn down chart.

14.2.8 Individual Status Reports

Individual Status Reports are to be completed by each individual team member immediately following the end of sprints. Team members will use the sprint goal, sprint backlog, and sprint burn down chart that is agreed upon by the team. Team members will also use the sprint review session to help in the assessment of other team members. The sprint review session can also be used to create individual goals for the upcoming sprint.

14.2.9 Engineering Notebooks

Each team member will be solely responsible for filling out their engineering notebook. Due to the setting of this project, all team members will disregard the witness signature for the engineering notebook.

14.3 Closeout Materials

14.3.1 System Prototype

The aim of this project is to meet all of the customer's requirements in the final system prototype. The final system prototype will first be demonstrated to the customer during the Prototype Acceptance Test. The final demonstration will take place on April 21, 2022.

14.3.2 Project Poster

The project poster will be delivered on April 21, 2022 during the final demonstration. The project poster will contain the system requirements, architectural design and design specifications and be 24" x 36".

14.3.3 Web Page

The project web page will be hosted on the University's senior design project repository. The web page will be accessible to the public and be delivered on April 21, 2022. The web page will contain: the team name, the product name, the timeline that the team worked on the project, all team member's names, an abstract, background, project requirements, system overview, demo video, future plans and improvements for the project, and links to all documents and source code.

14.3.4 Demo Video

The demo video will walk viewers through the entire user interface while also showing all the requirements that were met. The demo video should be under five minutes.

14.3.5 Source Code

All source code will be maintained in a private GitHub repository that will be accessible by all team members. Due to our customer's experience and knowledge of this project, they will have access to all source code upon request. This project will not be open sourced to the general public.

14.3.6 Source Code Documentation

Each team member will be responsible for properly documenting all code. Formal documentation will be generated with Doxygen and will be provided in LATEX format.

14.3.7 Installation Scripts

In the case that this project is to be a desktop application, installation scripts will be provided to allow for the automatic installation for the customer.

14.3.8 User Manual

A short, easy to read user manual will be provided in the application that will be accessible to all users. The customer does not require any additional user manual documentation.

REFERENCES

[1] Inc dba Rakuten Rewards Ebates Performance Marketing. Rakuten: Earn Cash Back & Save. Version 9.13.1 , 2021.

[2] Basket Savings Inc. Basket - Grocery Shopping. Version 3.9.189 , 2018.

[3] Fetch Rewards Inc. Fetch Rewards. Version 2.55.0 , 2021.

[4] Flipp Corporation Inc. Flipp - Weekly Shopping. Version 18.0.0, 2021.

[5] Ibotta Inc. Ibotta: Save & Earn Cash Back. Version 6.105.0 , 2021.

[6] Quotient Technology Inc. Coupons.com: Shop & Earn Cash. Version 6.9.0 , 2021.

[7] Honey Science LLC. Honey: Coupons & Cash Back. Version 2.2.3 , 2021.