First Iteration Documentation

for

Friendly Car Washers

Version 1.0 approved

Prepared by:
Crandall, Josue
Duong, Michael
Nguyen, Tram
Tran, Hoang
Tran, Huy

Orange Coast College - Fall 2017

CS A220: Software Engineering
Professor Ghadami

October 12, 2017

Table of Contents

Table of Contents 2
Revision History 3
Business Requirements 4
Background 4
Business Opportunity 4
Business Objectives and Success Criteria 4
Customer or Market Needs 5
Business Risks 5
Vision of the Solution 6
Vision Statement 6
Major Features 6
Assumptions and Dependencies 6
Scope and Limitations 6
Scope of Initial Release 6
Scope of Subsequent Releases 7
Limitations and Exclusions 7
Business Context 7
Stakeholder Profiles 7
Project Priorities 8
Operating Environment 8
User Stories (Functional) 8
User Stories (Non-Functional) 9
Use Cases 10
Use Case: See location list 10
Use Case: Sort list 11
Use Case: Write review 12
Use Case: View reviews 13
Use Case: Edit review 14
Use Case Diagram 15
Spring Backlog - Trello 16
Pre-Game Planning 16
Staging or Grooming 17
Development Process 17
User Manual 18

Revision History

Name	Date	Reason For Changes	Version

1. Business Requirements

1.1. Background

Friendly Car Washers is a local start-up company that contracted us to develop software to help them move in on a niche market service even Yelp doesn't provide. They recognized that larger score aggregation services which built out originally and expanded rapidly due to the newly formed online markets did not have local ties to business or a feel for the local environment.

The product idea is to start a score aggregation service which works with and builds ties with the local community in order to arrange deals and provide information that normally would get left by the wayside in the current score aggregation service model. In order to attempt to move into this territory, the Costa Mesa area was chosen, because it is local to the start-up, and car washing was chosen for the test bed. The idea being the product will provide a scoring service like yelp with more awareness of local peculiarities and tips for getting the best service in the Costa Mesa area. Because of the development based on individual areas, ties can be made with the local business sector to provide features a more general score aggregator could not.

1.2. Business Opportunity

There are more than 3 million people living in Orange County in 2017, and at least half of them are using cars for their daily purposes, such as driving to work and school. As a result, cars will get dirty and need to be washed in order to look clean, so car owners will search for car wash locations to clean their cars. Although Yelp is already able to provide car wash locations in Orange County, Friendly Car Washers plans to satisfy users with additional features such as the ability to make appointments for car wash, direct price comparison between certain car wash locations, and readable reviews of each location. Hence, Friendly Car Washers provides a great opportunity for users and customers to find the most reliable car wash location with fair price.

1.3. Business Objectives and Success Criteria

Since Friendly Car Washers is a brand new start-up, the software we are developing is the core of the business and it will be the business's source of income. Although the end goal is a much broader reach, our first objectives focus on the local, Orange County area. Our current target release date is in Mid-December, 2017. The main objective of this product is to establish the business so that it is well known and has many users. Once it has a following, the business will have more success when it expands beyond the Orange County area or to other areas beside car washes. Using the number of users as the measure of success, having 100 or more people use the application per month will be defined as the product being popular and successful. As for what the business will directly gain from the application, the customer data gathered can be analyzed by the business to obtain trends and other useful information. Revenue will be made by having car washes buy ad slots, estimated at $10/month for a slot. Additionally, a $0.99/month plan can be made available to end users to avoid seeing ads. With all of these objectives, the greatest impact on achieving success is whether or not local car washes are willing to cooperate. If twenty car washes in the area are willing to cooperate, we will have enough partnerships so that the planned appointment system will actually be a useful feature.

1.4. Customer or Market Needs

The customer needs an easy to use, easy to access, almost no learning curve, and lightning fast method of discovering what the best way to get their car washed is in their area. They need to have some voice in providing feedback and monitoring good and bad business practices by companies. They need to trust that the service provided to them will be accurate and reliable. They need to be able to reserve space at a car wash in advance while being able to trust that any information given to the application is secure.

Current services can deal with score aggregation, but they cannot provide the customer with local information, which slips through the cracks of the larger aggregators with no ground knowledge of local areas. Current services also do not allow customers to do bookings for services rendered or online payments because they do not directly work with local communities to provide such services. Our application is the first step in bridging those gaps, providing both local expertise and the ability to book car washes on the fly, while possibly expanding into other services or locations if the flagship product proves successful.

At the moment the software runs on any platform which can handle Java applications. However, if the flagship proves itself, the program can easily be ported to work on people's Andriod devices. For the product to take off, high levels of portability would be useful. For many real world services, the more platforms to which you have your service connected, the easier it is for people to use your service and the larger the market you can reach.

<u>Functional customer needs:</u>

1. The customer needs to be able to see reviews local to their area.
2. They need to be able to easily and accurately compare prices and service quality of different nearby locations.
3. They need to be able to participate and have a voice in their community to help maintain and reward high quality car washes near them.
4. They need to be able to reserve services on the fly in order to avoid having to wait in lines to make the car wash experience both pleasant and convenient.
5. They need to be able to change their minds and their reviews in case their expectations were not met or a business has changed its practices.

1.5. Business Risks

There are many risks that might affect the product. Firstly, the biggest risk comes from the competitors. If Yelp starts doing reservations and price comparison, our product will not be as competitive. Secondly, the consumer market might affect the product if not enough people will be interested enough to want to use the application. Moreover, our partners, who are the car wash companies, can be a risk if they are not willing to cooperate with us. Finally, the possibility that a programmer will be unable to stay until the completion of the project might be a lost and delay to this product. To mitigate these risks, we will focus on the appeal to the local community to continue to make our product stand out above Yelp and to convince local businesses to cooperate with us.

2. Vision of the Solution

2.1. Vision Statement

Friendly Car Washers brings the best experience for the car washing customer. It provides the nearest car wash location with comparable prices and quality of service. The program is also competitive with Yelp due to the additional features such as the convenient rewards system, the ability to make appointments, and the direct price comparisons. Hence, we will always develop and modify our program by receiving feedback from customer in order to satisfy their needs and to enhance their user experience. By creating a simple, easy-to-use application for the user, we will gain their trust and increase the amount of users, which will help us develop and become more popular so that the business may expand one day.

2.2. Major Features

1. Appointments: the ability to directly make appointments with local car wash companies.
2. Rewards system: the ability to gain points as a user uses our application so that they can then get a free car wash.
3. Price comparison: the ability to compare prices of each car wash location.
4. Find nearest location based on City: displays the nearest car wash locations according to current city location.
5. Write reviews: the ability to write personal experiences of certain car wash locations.
6. Read reviews: the ability to see other customers' experiences of certain car wash locations.
7. Have ratings out of five: the ability to give certain car wash locations a rating number.

2.3. Assumptions and Dependencies

At the present time that this first iteration documentation is being written, some assumptions are that no other car wash comparison application exists, many people in the Orange County area want an easy way to compare car wash locations, and people actually care about the quality of their car washes. A major dependency is that car wash companies will cooperate with us, since they will gain the benefit of getting more customers. Another dependency is that going to car washes without an appointment has a long wait time, which is necessary so that making an appointment through our application will be beneficial to the user.

3. Scope and Limitations

3.1. Scope of Initial Release

In the first iteration of the product, the intended major features will be the ability to find car washes in an area based on a city, the ability to rate those car washes, the ability to write reviews about those places, the ability to edit those ratings and reviews later on, and the ability to read others' reviews on those places, and the ability to compare the prices of different car washes.

3.2. Scope of Subsequent Releases

Because we are using an agile development model, we plan to have three iterations, with the third being the final release of the product. In our first iteration, we will have the features mentioned in section 3.1 and will defer to future iterations the following features: a log-in system to save rewards and location preferences; a rewards system; the ability to make appointments; a filter for specific car wash options, such as types of washes and if the location has vacuums; and a feedback button for users of the application.

3.3. Limitations and Exclusions

Some features that stakeholders might anticipate but that we do not currently plan to include are finding a location based on a specific address, the ability to pay directly when making appointments, and directly handling disputes between users and specific car washes. The reason we are not using address based location is that we do not have access to a database for addresses, so we cannot use them as a method to compare distance. We are not setting up the ability to pay directly because that requires us to create added security, with the potential of leaks in our program, and since most car washes are cheap and people pay in cash anyway, not much convenience is lost. Lastly, the business has decided not to handle disputes that our end users may have with car washes, so that feature will not be implemented.

4. Business Context

4.1. Stakeholder Profiles

Stakeholder	***Major Value***	***Attitudes***	***Major Interests***	***Constraints***
Friendly Car Washers (Start-up company)	*Increased revenue*	*See product as low investment for a possible viral app*	*Richer feature set than competitors; Time to market*	*Scalable solution in case they expand beyond Orange County*
Users/Custom -ers	*Satisfied car wash experience*	*Want the convenience of finding a quality place*	*Quality of the car wash location; Ease to find a car wash place*	*Must be easy to use and worth their time*
Car Wash Companies	*Increased customer base*	*Interested in being able to get new customers*	*Increased number of customers; Free advertising for their business*	*Should not cost them customers and money for our application*

4.2. Project Priorities

Dimension	***Driver (state objective)***	***Constraint (state limits)***	***Degree of Freedom (state allowable range)***
Schedule	*release 1.0 to be available by 10/12/17; final release 2.0 by mid-December, 2017*		
Features	*application has the core features of writing/reading ratings and reviews, and the ability to find car washes*	*core features must be implemented before additional features can be added*	*70-80% of high priority features must be included in release 1.0*
Quality	*application must be easy to use and not constantly crash*	*the application must rarely crash and overall be user friendly*	*90-95% of user acceptance tests must pass for release 1.0, 95-98% for release 1.1*
Staff	*gather a team dedicated to completing the project*	*maximum team size is 5 people, who are both developing and testing*	*developers and testers are not fixed, so the team can alternate those positions*

4.3. Operating Environment

This application will mainly be used by car owners seeking to wash their car in the Orange County area. The only data available will be of cities in Orange County. This application may be used at any time of the day, so the user should always have quick access to the data. This is done by storing the data of the car washes in text files. Lastly, as of now, the application is used with Java console, so Java is needed on the computer.

5. User Stories (Functional)

1. As a user,
   I want to select the city I'm in
   So that I can find car washes near me.
2. As a user,
   I want to have a list of car wash locations
   So that I can see and choose the best location.
3. As a user,
   I want to sort car wash locations by rating or price
   So that I can focus on what I care about most.
4. As a user,
   I want to be able to review car washes.
   So that I can help keep car washes accountable for their service.

5. As a user,
   I want to see and modify my review of car washes
   So that I can modify my review or rating as my experience changes.

6. User Stories (Non-Functional)

1. Usability:
   As a user,
   I want the application to be easy to use
   So that I don't have to spend time learning.
2. Accuracy:
   As a user,
   I want the information to be displayed accurately
   So that I can reliably make my choice.
3. Security:
   As a user
   I want my information, such as my password, my username, and my email address, to be protected
   So that I can feel safe.
4. Maintainability:
   As a car wash owner,
   I want to to be able to easily update my information when it changes
   So that people know how to properly contact me.
5. Scalability:
   As the product owner,
   I want to be able to scale the product cheaply and quickly
   So that in case the current software proves profitable, I can quickly expand into other locations or markets.

7. Use Cases

Use Case: **Show locations**

Id: UC-01

Description
The customer chooses the city displayed from the app. The app will show the list of car wash locations based on the city selected. The customer can then sort the location by either price, rating or location. The customer can also choose the specific car wash location to see the name, price, rating, and address.

Primary Actor
Customer who uses the app.

Pre-Conditions
Customer must open the app.

Post Conditions
Success end condition

- The app shows the correct car wash locations based on the city selected.

Failure end condition

- The app doesn't display the list of city for the customer to choose.
- The app displays the incorrect car wash location based on the city selected, for example, showing the one in Garden Grove while customer chooses Costa Mesa instead.

Main Success Scenario

1. Customer opens the app from a device.
2. The app interface will appear on the screen with a list of the cities.
3. Customer select a city listed. (Ex: Garden Grove)
4. Customer hits select.
5. The app will display all car wash locations in that city.
6. Customer can then sort the information (UC-02) or see the reviews of a car wash (UC-04).

Use Case: **Sort list**

Id: UC-02

Description
The customer can sort the list by price or rating so they can see the car wash locations in the order from cheapest to most expensive or from highest to the lowest rating.

Primary Actor
User

Pre-Conditions
User needs to choose either sorting by price or by rate.

Post Conditions
Success end condition
The list is sorted in the right order.

Failure end condition
It does not give the proper order.

Main Success Scenario

1. The car wash list is shown based on the city (UC-01).
2. User has the option to sort by price or sort by rating. They click on one of those buttons to sort the information based on the category.
3. The list should be shown up with the price from cheapest to most expensive when clicking “Sort by price” and the average rating from highest to lowest if they choose “Sort by rate.”

Use Case: **Write review**

Id: UC-03

Description
The user selects a car wash he or she went to. Then, the user writes a review explaining his or her experience and gives a rating out of 5. The review is then posted for others to see.

Primary Actor
The end user of the application who has gone to a car wash and wants to review it.

Pre-Conditions
The primary actor has already selected the car wash to review and the system is displaying the information specific to the one car wash (UC-04).

Post Conditions
Success end condition
The review is written and can be seen on the car wash information page.

Failure end condition
The primary actor decided to cancel writing the review and the review is not displayed on the information page.

Main Success Scenario

1. The primary actor accomplishes UC-04 and can see the information about a specific car wash
2. The actor hits “Write a review.”
3. The system brings up the form page to enter the rating and review.
4. The actor enters the rating and writes the review.
5. The actor hits submit.
6. The system checks that the rating is between 1 and 5.
7. The system stores the rating and review.
8. The system loads the car wash information page with the review added.

Extensions

6a. If the actor did not enter in a proper rating in step 4

1. The system notifies the actor that the rating is not between 1 and 5.
2. The actor re-enters a rating.
3. Use case returns to step 5 and may return to this extension if the new rating is still incorrect.

Use Case: **View reviews**

Id: UC-04

Description
User selects a car wash and can view all the reviews of said car wash.

Primary Actor
The end user is the primary actor here since the feature allows them to get information they request.

Pre-Conditions
Currently no login is required, but in the second iteration we may update this Use Case to require log in.
At the moment the precondition is that the user selected a specific car wash to see reviews of.

Post Conditions
Success end condition
All the reviews relevant to them are displayed directly to them in an easy to parse manner.

Failure end condition
There could be no reviews of the location, in which case they can't see any useful reviews of the place. And they get sent one which says "no reviews."

Main Success Scenario

1. User selects car wash to find reviews of
2. Program searches database for reviews of that car wash
3. Program displays the list to user in an easy way to digest form

Extensions

1. User selects car wash to find reviews of
2. Program searches database for reviews available, none can be found
3. Program displays a message where the list normally goes saying no reviews of that car wash are available

Use Case: **Edit review**

Id: UC-05

Description
After having reviewed a car wash or selecting a car wash previously reviewed, the user can edit her or his review of the car wash.

Primary Actor
The end user is the primary actor here since it allows the end user to alter their review. There are also side benefits to other users of the application since it increases the general quality of reviews.

Pre-Conditions
Currently no login is required, but in the second iteration we may update this Use Case to require log in.
At the moment the precondition is that the user selected a specific car wash to see reviews of. Also they must have written a review in order to edit their review, if they have not they go to the "write a review" use case (UC-03).

Post Conditions
Success end condition
The review they wrote has been modified and is currently displayed to them with the changes they wish to make.

Failure end condition
They could have not written a review of the place in which case they get sent to the "write a review" use case (UC-03).

Main Success Scenario

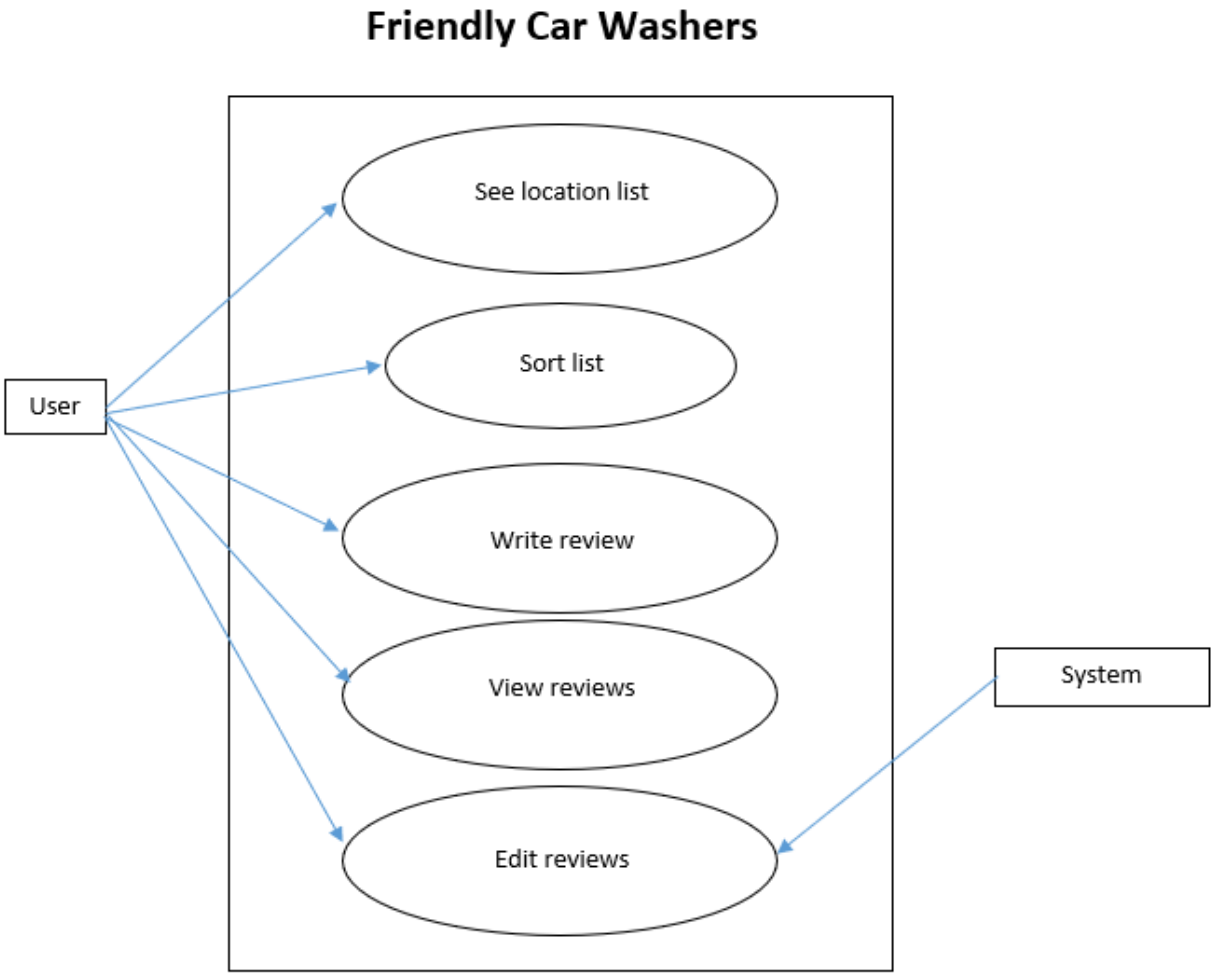
1. User selects a car wash
2. User selects edit review of said car wash (precondition they have written a review prior)
3. Program loads review of that car wash and displays it to the user in an editable form
4. User edits rating or review text body and confirms their changes
5. Program saves and uploads changes to stored data
6. Program brings User back to display of reviews of the car wash including their updated review

Extensions

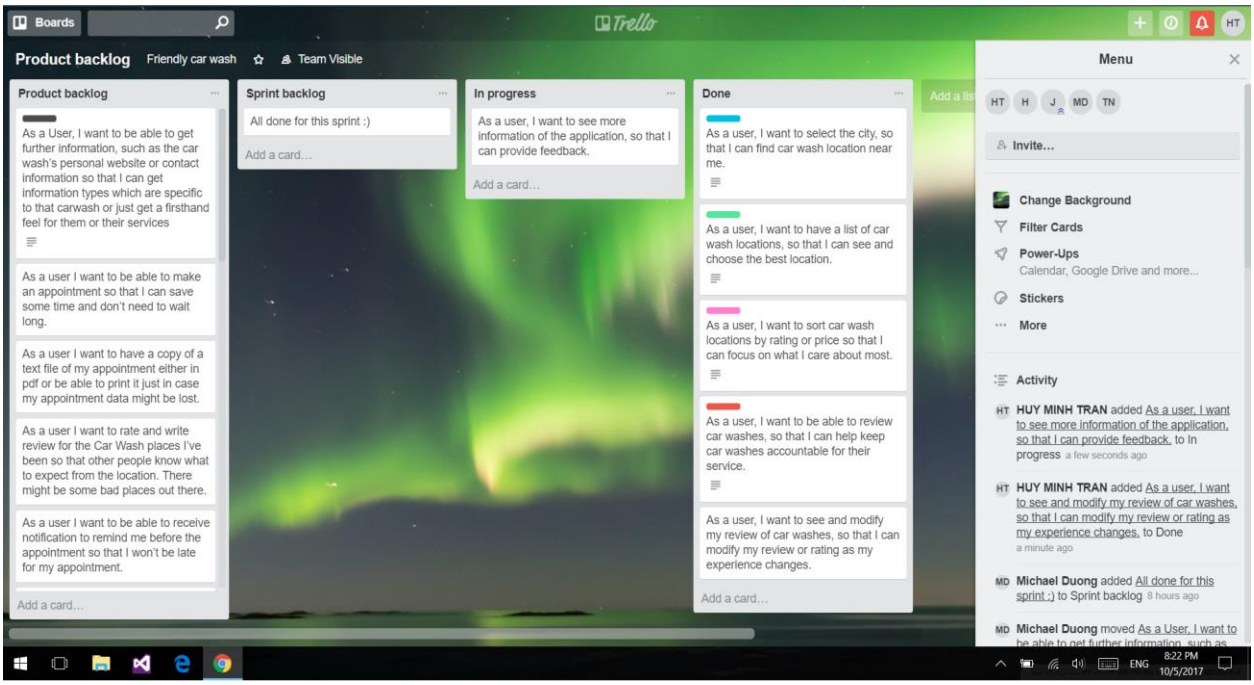
1. User selects a car wash
2. User selects edit review of said car wash
3. Program loads review of that car wash and displays it to the user in an editable form
4. User edits rating and review in a way which is no longer formatted correctly (not using an integer between 1 and 5 for the rating)

5. Program displays a warning message telling user they need to properly format the review in order to continue and doesn't allow them to confirm their response until the formatting is fixed
6. User fixes formatting
7. Program saves and uploads changes to stored data
8. Program brings User back to display of reviews of the car wash including their updated review

8. Use Case Diagram



9. Spring Backlog - Trello



10. Pre-Game Planning

In order to decide which user stories to integrate into our first iteration, we chose user stories which had features that were not dependent on other features being already completed in our product. That way, the chosen user stories could be implemented relatively completely without having to add temporary dependency code. We also outlined a vision for our first iteration that had the base feature set for a functional design.

In our poker game, we focused on one specific user story at time. With each one, we made sure everyone clearly understood what would be necessary to implement the user story so that they had a clear idea of the programming needed behind it. Then, everyone gave their projected time to complete the user story, and if there was a large difference, we had a discussion about the reason for the difference and gave our projected time again. Then, we got the average and rounded it to the nearest hour. We then repeated this until we went through all five chosen user stories and had a good idea of the work needed to complete this first iteration.

11. Staging or Grooming

During the pre-game planning, we figured out which user stories were the most essential components of our application and focused on those. These user stories were the ones that focused on displaying the list of car wash data and writing and editing reviews. Because we had a thorough discussion on the importance of these, we did not need to edit our product backlog much during the development process. There was only one user story that we decided to remove from this iteration because it was not as essential.

The user story that we decided to remove was displaying more information about each car wash. We realized that having reviews displayed was a much higher priority than information, which can be added with relative ease, so we prioritized the review user stories over the display more information user story. Our main goal during this first iteration was having a functional app with all the basic features working, so we kept our product backlog focused on that without having too many "more advanced" features.

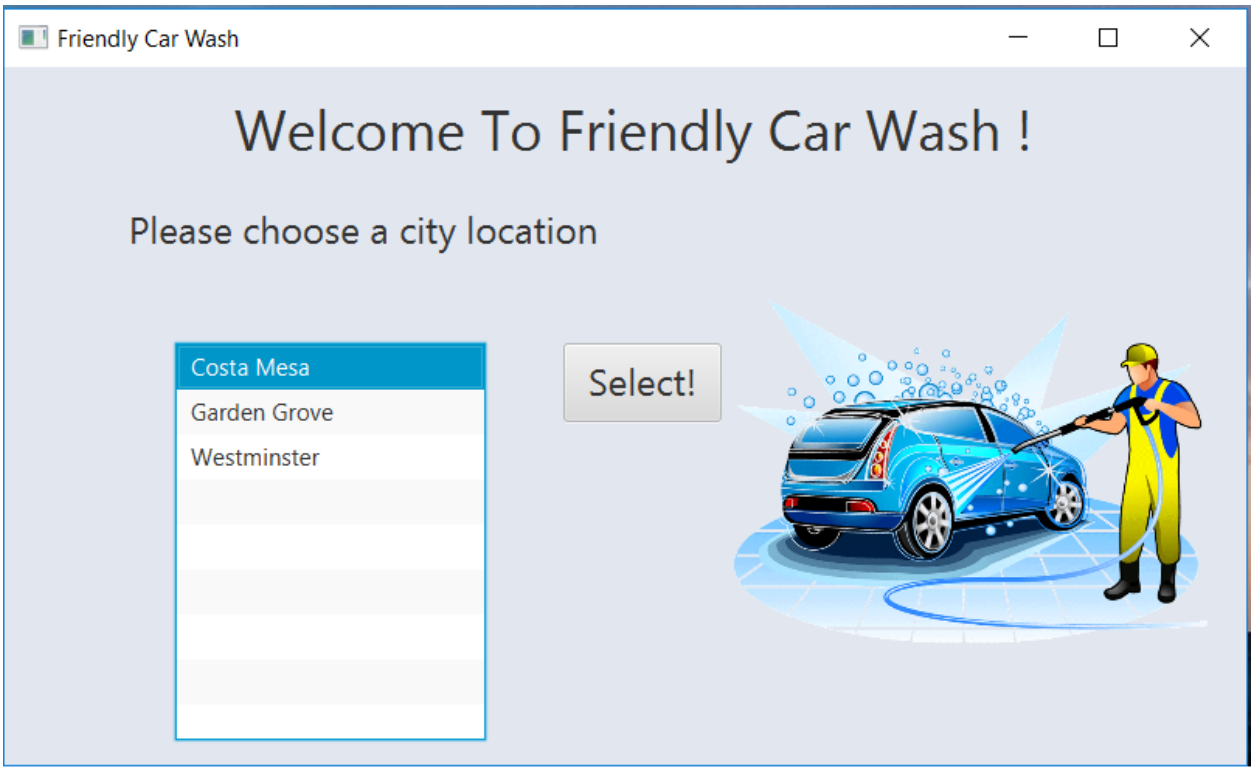
12. Development Process

For our development process, we as a group met up and decided on what idea to do. We all thought that a "Car Wash Yelp" idea, as we first called it, could be interesting to many people and could have many useful features, so we chose that idea. Next, we as a team, huddling around a computer screen, filled out the vision and scope document and built a more solid understanding of what we expected the application to do and where we were going to go with it. We then all wrote five user stories for the application, met up, and selected our top five user stories among all of us. The main focus was on user stories that would focus on the core features necessary for a basic application and upon which future features would be built. Once we had our planning poker game and figured out how long each user story would take to complete, we divided the work and started programming.

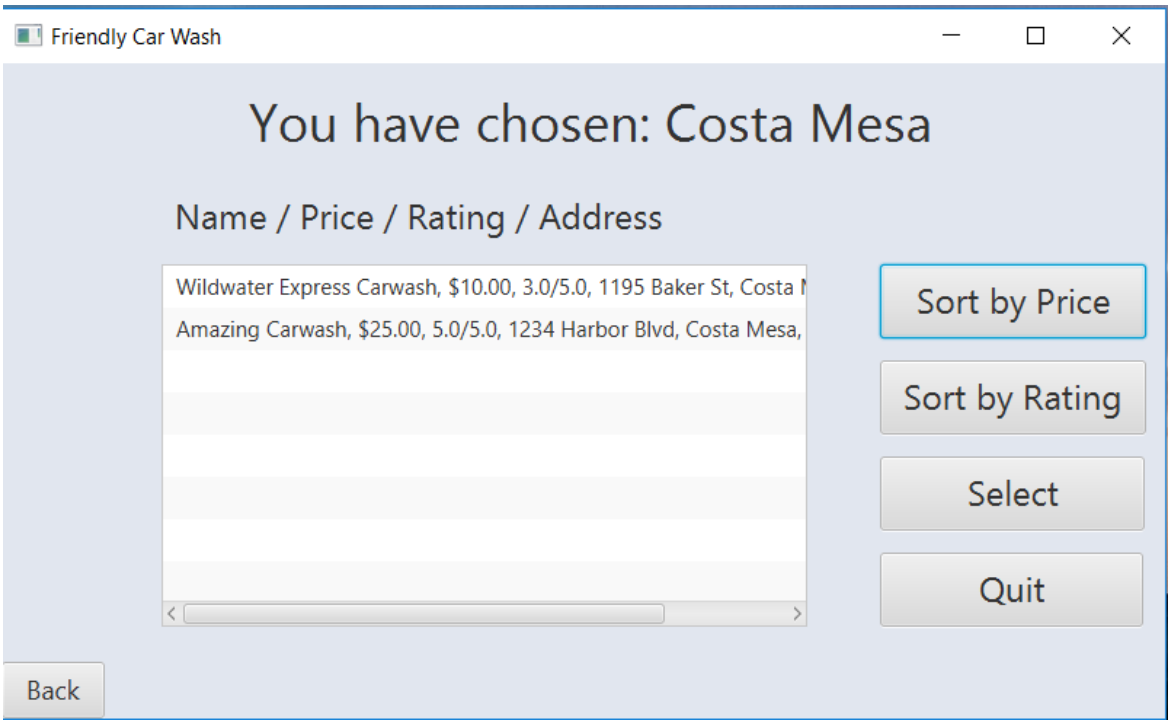
While we were programming, in order to share code, since we were unable to set up Github at the start, for this first iteration, we had to share code using Google Drive. The first week was spent on setting up the work environment on everyone's computers and getting used to the tools, mainly Scene Builder since most of the team had not developed a GUI before. Once that was set up, though, we started working on our parts for the program and finished it in about the estimated times for each user story. During the programming phase, we also wrote the use cases for each respective user story we were assigned. Once we were all finished with our parts, we combined them, tested all of the features, and finished our first iteration product.

As for the rest of the documentation in this first iteration, we evenly divided the work among the team, with each person receiving a section, and everyone wrote out the paragraphs for their section. Our main process during development was figuring out what work to do, dividing it up among the team, having everyone do their part, and combining everything together once it was all finished.

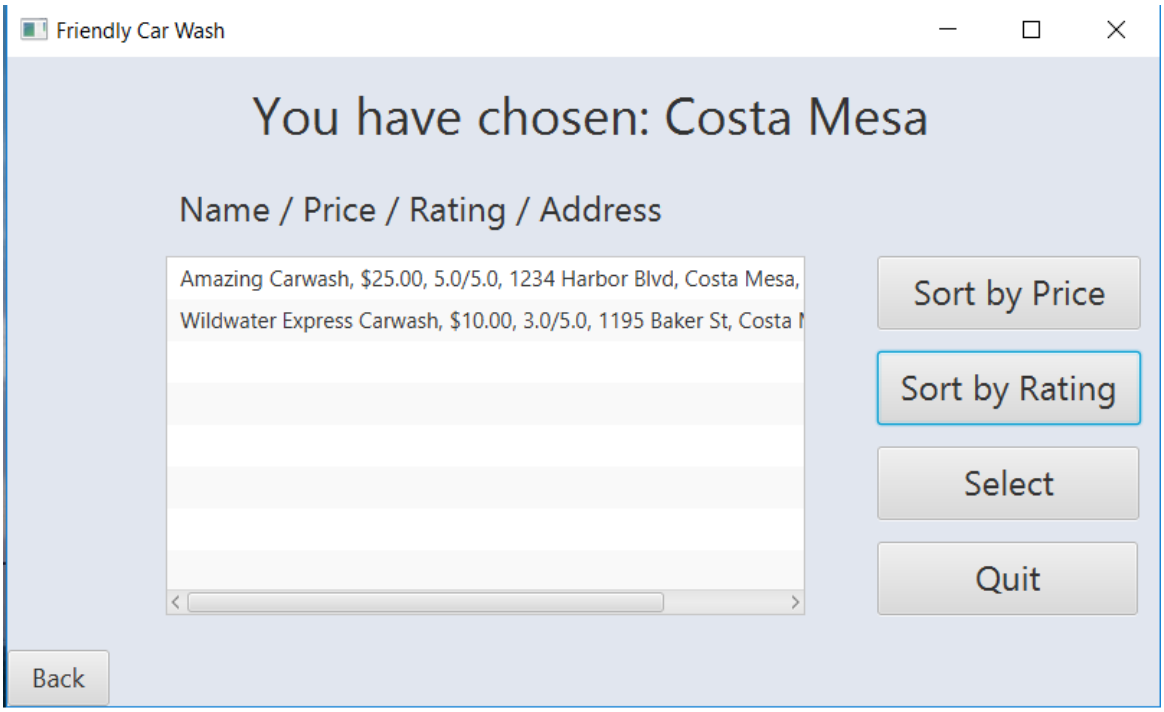
13. User Manual



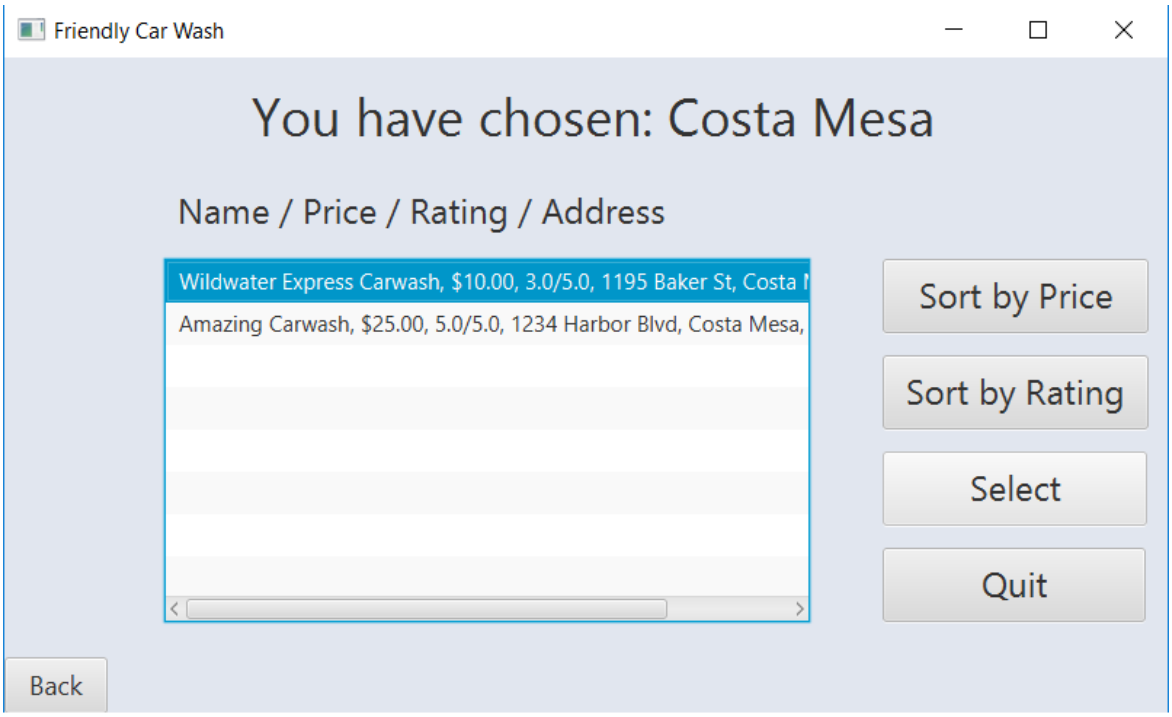
1. User runs the program.
2. The program displays the opening scene.
3. User chooses the city.
4. User hits select.



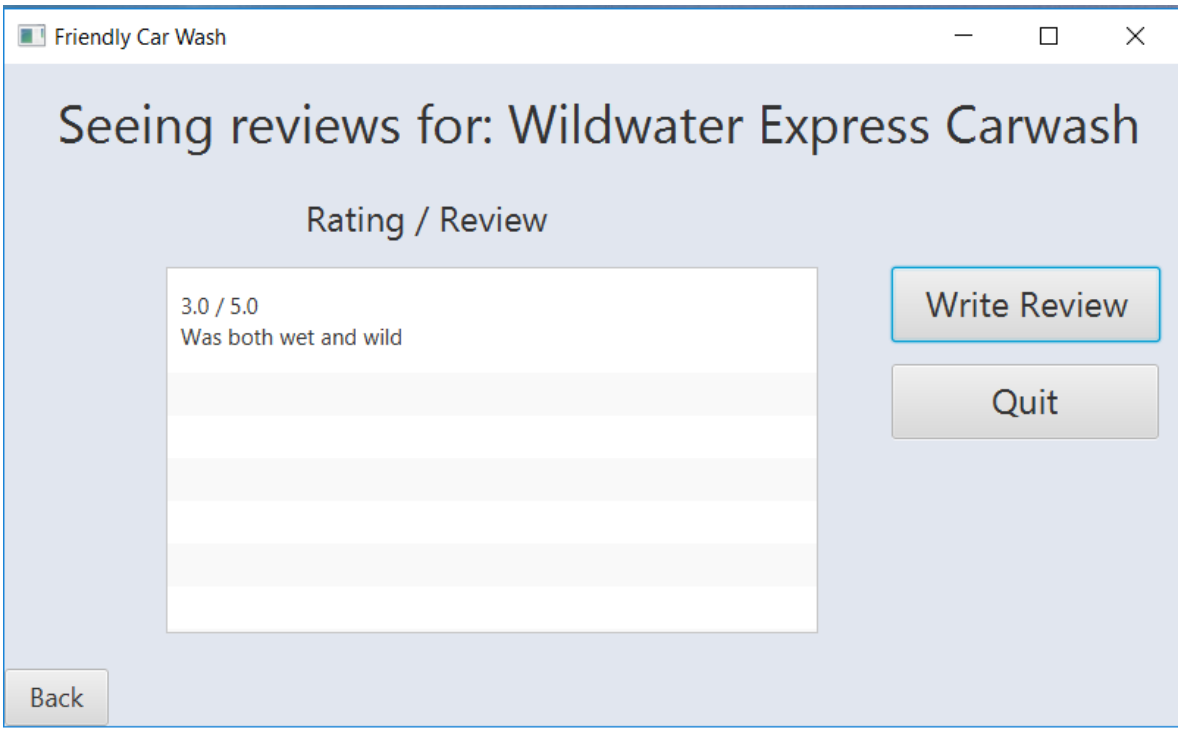
1. User hits "sort by price"
2. Program displays locations that are sorted by price from lowest to highest



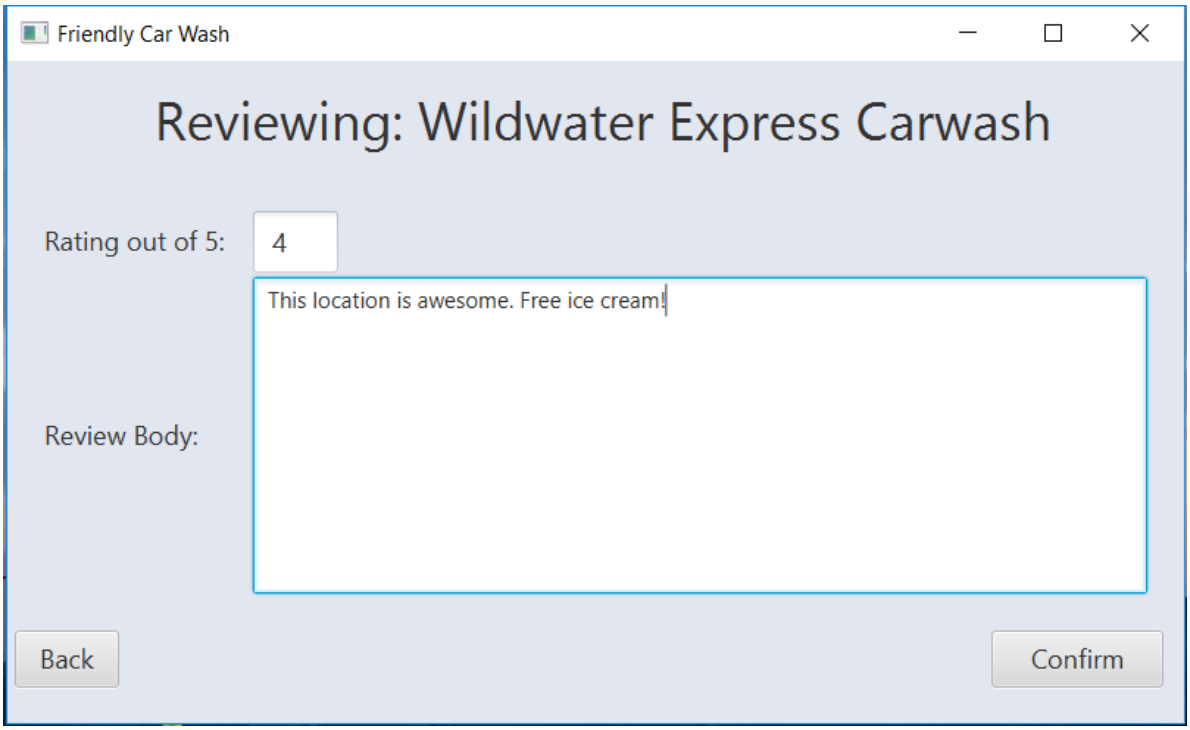
1. User hits "sort by rating"
2. Program displays locations that are sorted by rating from highest to lowest



1. User chooses a car wash location.
2. User hits select.

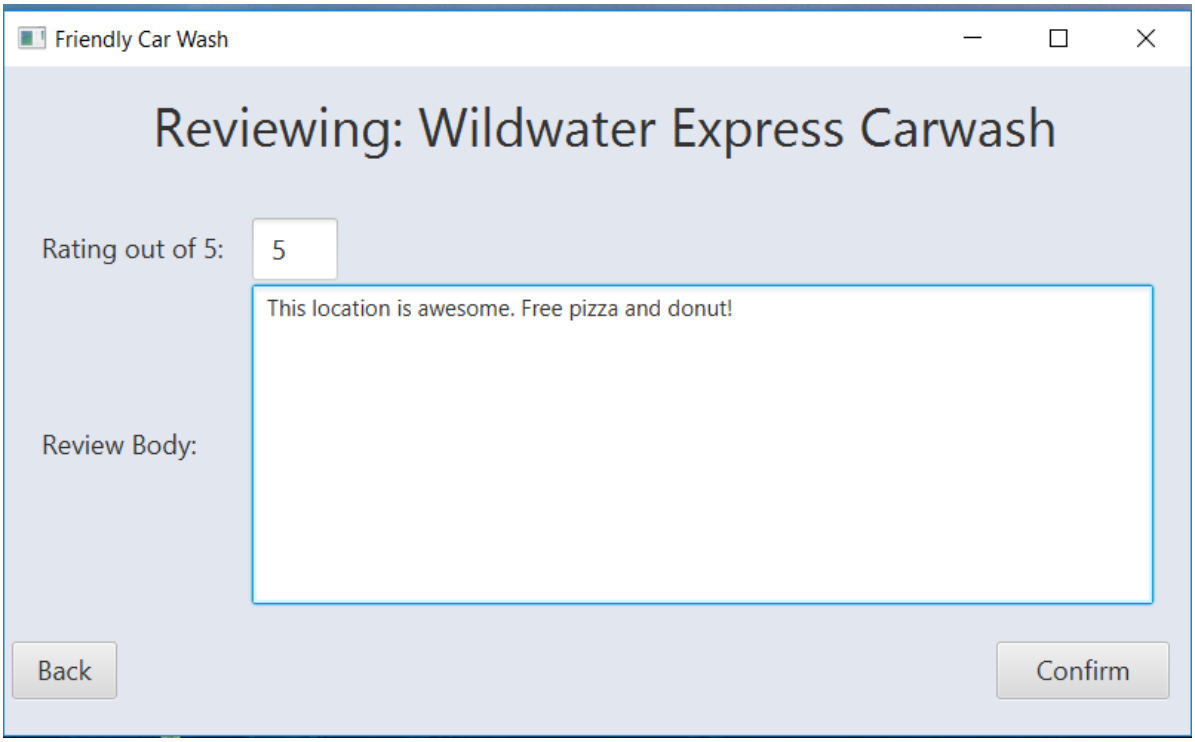


1. The program displays the review(s) of the location.



1. User hits “write review”
2. The program displays the scene for user to write review.

1. The user hits confirm.
2. The program inserts the user review.



1. The user hits “edit review”
2. The program displays the scene for user to edit.

1. The user hits “confirm”
2. The program updates the user edited review.