Third Iteration Documentation

for

Friendly Car Washers

Version 1.2 approved

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

Orange Coast College - Fall 2017

CS A220: Software Engineering
Professor Ghadami

December 14, 2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team	10/26/17	- Added Functional User Stories numbers 6-10. - Updated Use Case UC-01 to "Show locations" in the section and in the Use Case Diagram. - Added five new use cases (UC-06 ~ UC-10)	1.1
Tram Nguyen	10/27/17	-Added ability to resize the window	1.1
Team	10/31/17	- Added Class Diagram.	1.1
Team	11/02/17	- Added CRC Cards.	1.1
Huy Minh Tran	11/04/17	- Added "Change City" button.	1.1
Hoang Tran	11/05/17	- Updated use cases in Use Case Diagram. - Added status if reviews are "edited.".	1.1
Josue Crandall and Huy Minh Tran	11/07/17	- Added ability to see more information about specific car wash locations	1.1
Michael Duong	11/07/17	- Added indicator for how data is sorted in the list of car wash locations	1.1
Team	11/14/17	- Added 5 new User Stories and Use Cases (UC-11~UC-15)	1.2
Team	11/17/17	- Added Test Cases (TC_01~TC_10)	1.2
Team	11/21/17	- Added State Diagrams	1.2
Michael Duong	12/04/17	- Revised Project Plan	1.2
Team	12/05/17	- Revised Class Diagram and CRC Cards	1.2

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 Orange County 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 Orange County 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 car wash locations, and the ability to save favorite locations. 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 Android 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.

Functional customer needs:

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 exclusive coupons, 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. Saving favorite locations: the ability to have a list of favorite car wash locations to which users can refer and remember which places were good.
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 third 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 release of the product, the intended major features will be the ability to find car washes in an area based on a city, to rate those car washes, to view, write, and edit reviews about those places, to compare the prices of different car washes, to sign in and out of accounts, to save and view a list of favorite car washes, and to see available coupons.

3.2. Scope of Subsequent Releases

Because we are using an agile development model, we are able to quickly and effectively build upon the codebase for our application. The only major feature that we are still missing is the appointments system. The main reason for this the fact that we have not gone out and talked with many car washes yet and we delayed it to avoid having too much work in a single iteration. With the account features already written, it will be easy to add a rewards system in. Since the core features are working, we can add more user-friendly features, such as a filter for specific car wash options, such as types of washes and if they have 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 1.2 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 select 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 serialized 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 am 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 add a review of a car wash,
   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. As a user,
   I want to see the "Edited" caption next to an edited review
   So that I can know which reviews have been revised.
7. As a user,
   I want to have an indication of how the car washes list is sorted when I hit a sort button,
   So that I know how the information is organized.
8. As a user,
   I want to see more information about specific car wash locations,
   So that I can contact them or gather more information myself.
9. As a user,
   I want to change the city from any main part of the program,
   So that I can quickly search for more car wash locations.
10. As a user,
    I want to create an account,
    So that I can login to use the login-only features.
11. As a user,
    I want to log in,
    So that I can save my favorite car wash locations and write reviews.
12. As a user,
    I want to see available coupons in a city,
    So that I can find good deals available right now.
13. As a user,
    I want to save a car wash to my list of favorite car washes,
    So that I can remember them for future washes.
14. As a user,
    I want to see my favorite car wash locations,
    So that I can immediately choose the ones I like when I login.
15. As a user,
    I want to be able to log out,
    So that I can log in to a different account and protect my information.

6. User Stories (Non-Functional)

1. <u>Usability:</u>
   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 displayed city from the application. The application 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 alphabetical. The customer can also choose the specific car wash location to see the name, price, rating, and address.

Primary Actor
Customer who uses the application.

Pre-Conditions
Customer must open the application.

Post Conditions
<u>Success end condition</u>

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

<u>Failure end condition</u>

- The application doesn’t display the list of city for the customer to choose.
- The application 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 application 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, rating, or alphabetical, so they can see the car wash locations in the order from cheapest to most expensive, from highest to the lowest rating or alphabetical order.

Primary Actor
User

Pre-Conditions
User has to select a city already and needs to choose either sorting by price or by rating.

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, sort by rating, or sort by alphabetical.
3. They select on one of sorting method to sort the information based on the category from the dropdown box.
4. The list should be shown up with the price from cheapest to most expensive when clicking “Price,” the average rating from highest to lowest if they choose “Rating” and from A to Z by car wash name if they choose “Alphabetical.”

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 logged in (UC-11), 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 review 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-11and UC-04 so they can see the information about a specific car wash
2. The actor hits "Write 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 review 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 see all the reviews of the car wash.

Primary Actor
The end user who wants to see the review of the car wash.

Pre-Conditions
The user already selects a city and a car wash in that city.

Post Conditions
Success end condition

- All the reviews relevant to the car wash 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, and 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
The user is already logged in (UC-11), chooses a city and car wash in that city.
The user 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
<u>Success end condition</u>

- The review they wrote has been modified and is currently displayed to them with the changes they wish to make.

<u>Failure end condition</u>

- 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 the 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 the 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.

Use Case: **Show revision status**

Id: UC-06

Description
Users can see the “Edited” caption next to the reviews to know if a review has been revised. When a user edits a review, the system adds “Edited” caption to the review.

Primary Actor
The system itself is the primary actor since it has to insert the “Edited” caption into the review.

Pre-Conditions
User has opened the application and chosen a specific car wash location to show the reviews of (UC-04). Also, the user has already written a review (UC-03).

Post Conditions
Success end condition

- User can see the “Edited” caption next to the review.
- System successfully adds the “Edited” caption to the review.

Failure end condition

- The app does not show the “Edited” caption next to the review.
- The app show the “Edited” caption next to the review that has not been modified.

Main Success Scenario

1. The list of car washes is shown based on the city selected (UC-01).
2. User chooses a specific car wash to see the reviews (UC-04).
3. User has already written a review (UC-03) and chooses to edit the review.
4. User edits the review normally (UC-05).
5. The system detects that the review was edited and inserts the “Edited” caption.
6. User will be able to see the “Edited” caption next to the review that has been revised.

Extensions

3a. If User has not edited a review

1. User edits their review if they feel the need to change it (UC-05).
2. Now that their review is edited, the caption is displayed next to the review in the list.

Use Case: **Indicate sorting criteria**

Id: UC-07

Description
When seeing the list of car washes after choosing a city, if the user chooses to sort the list of car washes by either price or rating, how the list is sorted is indicated by the system.

Primary Actor
The system so that it can change the titles to indicate to the user how the car washes are sorted.

Pre-Conditions
The user has chosen a city and is able to see the list of car washes for that city (UC-01).

Post Conditions
<u>Success end condition</u>

- The user changes the text box to select the sorting criteria and the app will show the indicator next to Price or Rating to show how the list is sorted.

<u>Failure end condition</u>

- The app does not show the indicator next to Price or Rating.
- The app shows an incorrect indicator for the sorting type (e.x. showing a down arrow for ascending order).

Main Success Scenario

1. Car wash locations are shown based on the city (UC-01).
2. The user is able to choose between “Alphabetical,” “Price”, and “Rating.”
3. The user chooses to sort the list of car washes by one of the criteria (UC-02).
4. Based on what the user chose in step 3, the system will insert an indicator next to the title of the criterion they chose. If they chose “Sort by Price,” the indicator will appear next to “Price” and if they chose “Sort by Rating,” it will be next to “Rating.” If they choose alphabetical, there is no indicator.

Use Case: **See more information**

Id: UC-08

Description
After selecting a specific car wash, the user can click on "See more information" to get more information about the car wash, such as a website, phone number, and email.

Primary Actor
The user who has used the car wash application.

Pre-Conditions
The user has chosen a specific car wash and is looking at the list of car washes (UC-01 or UC-14).

Post Conditions
Success end condition
- More information about that car wash is displayed for the user to see and use.

Failure end condition
- No other data about the car wash is available.
- Incorrect data, like the wrong car wash, is displayed to the user.

Main Success Scenario

1. The actor has already seen the car wash locations (UC-01) and selected a specific car wash.
2. The user clicks on the "See more information" button for that car wash.
3. The application then displays what additional information about that car wash is available, such as a business website, contact phone number, and/or email.
4. The user can do whatever they want with this given information.

Use Case: **Change city**

Id: UC-09

Description
The customer can change the city from any point in the application except for when writing or editing a review so that they have to confirm if they want to save the review or not. The user can change the city immediately just by clicking the button.

Primary Actor
The customer who uses the application.

Pre-Conditions
The customer must already have selected a city or else there would be no selected city to change.

Post Conditions
<u>Success end condition</u>
- The customer is able to change the city immediately without going back through each scene.

<u>Failure end condition</u>
- The customer cannot change the city without going back through every scene to get back to the first scene.

Main Success Scenario

1. The user selects a city from the first scene (UC-01).
2. The user selects a car wash location from that city (UC-04).
3. The user goes to the review scene (UC-04).
4. The user clicks on the “change city” button.
5. The program displays the cities and the user can select the city again.

Extension

2a. After seeing the list of car wash locations, if the user wants to change city from there,

1. The user clicks on the “Change City” button.
2. The program displays the cities and the user can select the city again.

Use Case: **Create account**

Id: UC-10

Description
The customer can create an account in the application so that they can use their account to log in and add and see favorite car wash locations.

Primary Actor
The customer who uses the application.

Pre-Conditions
The customer must have the application open.

Post Conditions
<u>Success end condition</u>
- The customer is able to create an account.

<u>Failure end condition</u>
- The customer cannot create an account.

Main Success Scenario

1. The user runs the program.
2. The user selects "Create an account."
3. The program displays the blank username and password field as well as other user information.
4. The user inputs their desired username and password and information.
5. The user clicks "Create."

Extension

4a. If the user changes his or her mind while making the account,
   1. The user can click "Cancel" to stop the process if they do not want to create an account.

5a. If the username is already taken
   1. The system displays the warning message "username already taken."
   2. The system returns to the page with the information still filled out.

Use Case: **Log in**

Id: UC-11

Description
With a created account, the user can login and access login-only features, such as writing reviews and having a list of favorite car washes.

Primary Actor
The end user with a desire to have an account and use the special features.

Pre-Conditions
The actor has already created an account (UC-10).

Post Conditions
<u>Success end condition</u>
- The actor is logged in and can use the special features.

<u>Failure end condition</u>
- The actor is not logged in.

Main Success Scenario

1. The actor clicks the “Log in” button.
2. The system takes the actor to the login page.
3. The actor enters in the username and password.
4. The actor clicks “login.”
5. The system logs the user in.
6. The user can now write reviews (UC-04) or do things with their favorites list (UC-12 and UC-13).

Extension

4a. If the username and password is not found in the system

1. Display a message “User and password combination not found.”
2. The actor has to go back to step 3 and re-enter the username and password.

Use Case: **See coupons**

Id: UC-12

Description
User is able to see available coupons for all car wash locations offering those coupons.

Primary Actor
User who uses the application.

Pre-Conditions
User must select the city they are in and is able to see the “See coupons” button.

Post Conditions
<u>Success end condition</u>

- User is able to see all available coupons of all car wash locations in the city.
- The information of the coupon is displayed accurately including the promotion details and the info of that car wash location.

<u>Failure end condition</u>

- User hit the “See coupons” and the system does not respond.
- The coupon information is displayed inaccurately.

Main Success Scenario

1. User selects a city from the list displayed (UC-01).
2. User hits the “See coupons” button.
3. The program displays the list of coupons for all car wash locations in the city with promotion details and the info of that car wash location.

Use Case: **Save favorites**

Id: UC-13

Description
The customer can save their favorite car wash locations so that they do not have to look for it each time they use the application.

Primary Actor
The customer who uses the application.

Pre-Conditions
The customer already logged in and selected a city.

Post Conditions
<u>Success end condition</u>

- The customer is able to add their car wash to the favorite list

<u>Failure end condition</u>

- The customer cannot add the car wash to the favorite list

Main Success Scenario

1. The user selects a car wash location.
2. The user selects "Add to favorites."
3. The program adds the car wash location to the favorite list.
4. The program indicates that the car wash location has been added.

Extension

3a. If the car wash location is already in the favorite list and the user still tries to add it, the car wash will not be added to the list and there will be no indication of the wash being added.

Use Case: **See favorites**

Id: UC-14

Description
Users can choose to see their previously selected favorite car wash locations rather than the available car washes in a city.

Primary Actor
The end user.

Pre-Conditions
The user must already ran the program and logged in to the account.

Post Conditions
Success end condition
- The user can see their favorite car wash location list.

Failure end condition
- The user is not logged in so the feature fails.
- The favorite list does not display the correct car wash locations.

Main Success Scenario

1. The user is in the scene displaying car washes in a city.
2. The user clicks “See favorite.”
3. The program displays a list of user’s favorite car wash locations rather than the car washes in the city.
4. The user can then interact with those car washes as if they’re the normal car wash list.

Use Case: **Log out**

Id: UC-15

Description
If the user is signed into an account they can log out or swap users.

Primary Actor
The end user who’s logged in and wants to logout or swap users.

Pre-Conditions
The user needs to be logged in.

Post Conditions
Success end condition

- The user becomes logged out, or logged into a different account.

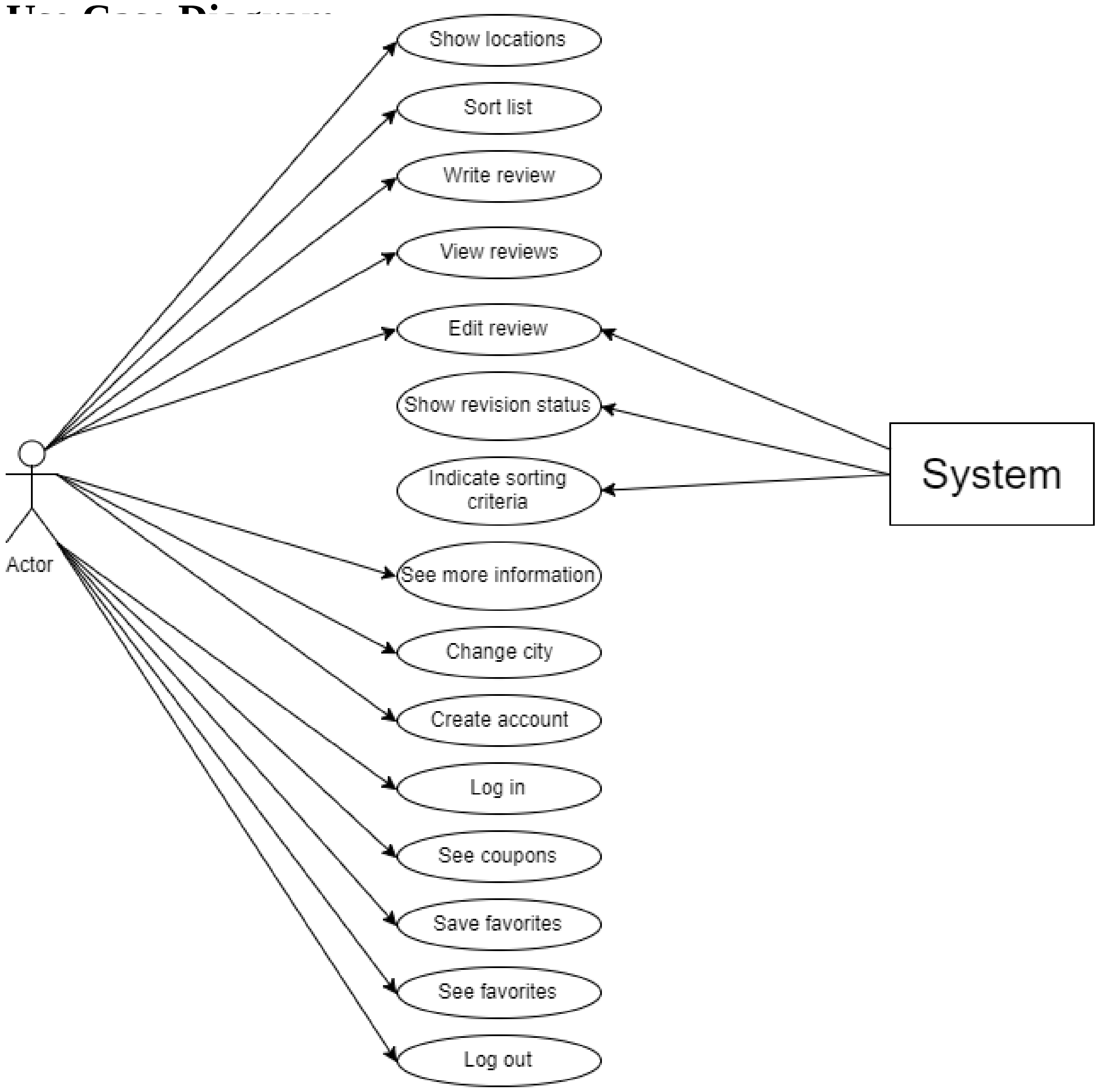
Failure end condition

- The user fails logging into the new account and stays logged into their current account.

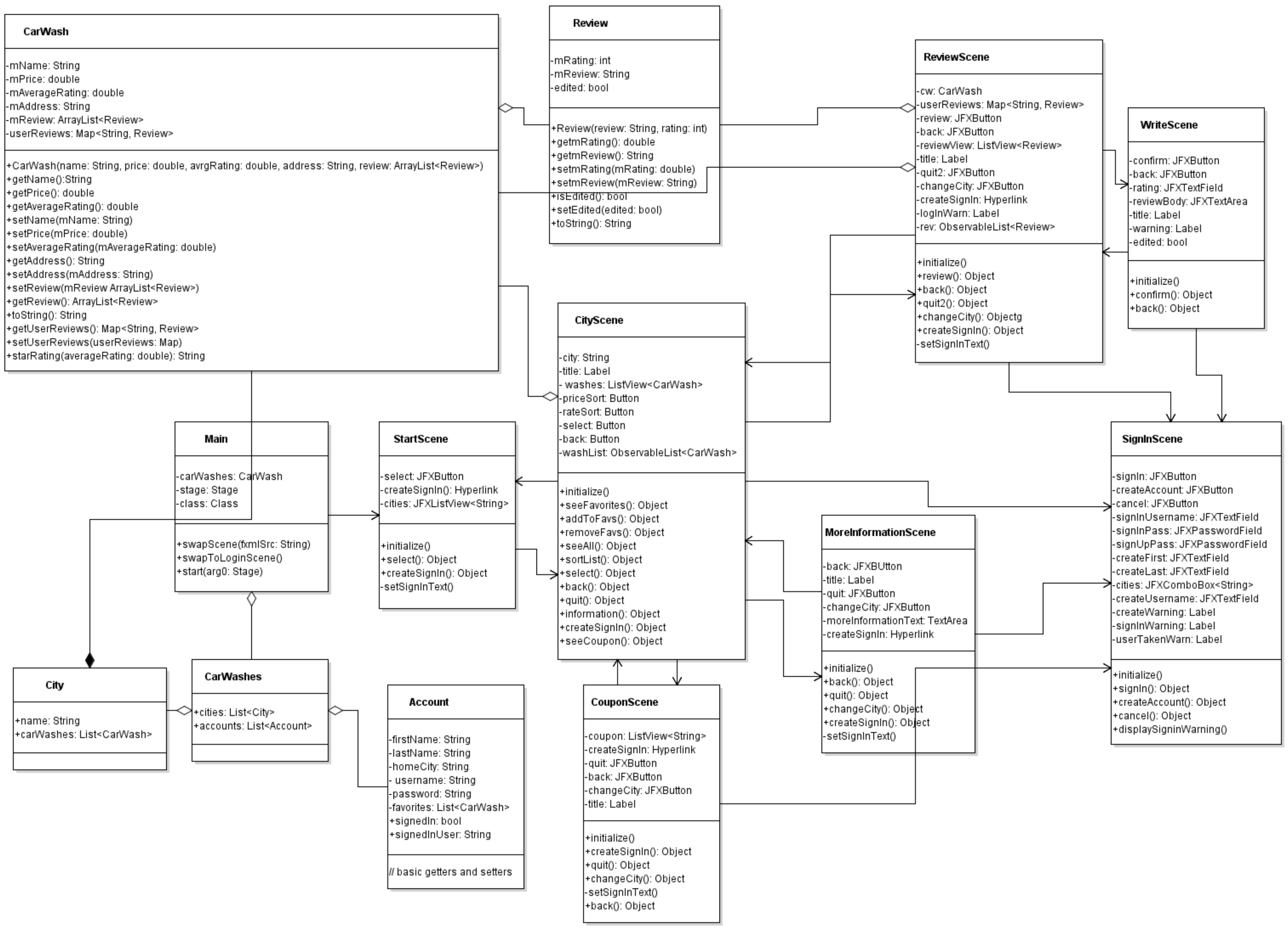
Main Success Scenario

1. The user presses the logout button.
2. The system logs the user out.
3. The system returns the user to the scene they were on before clicking log out.
4. The user has the option to log in again by following UC-11.

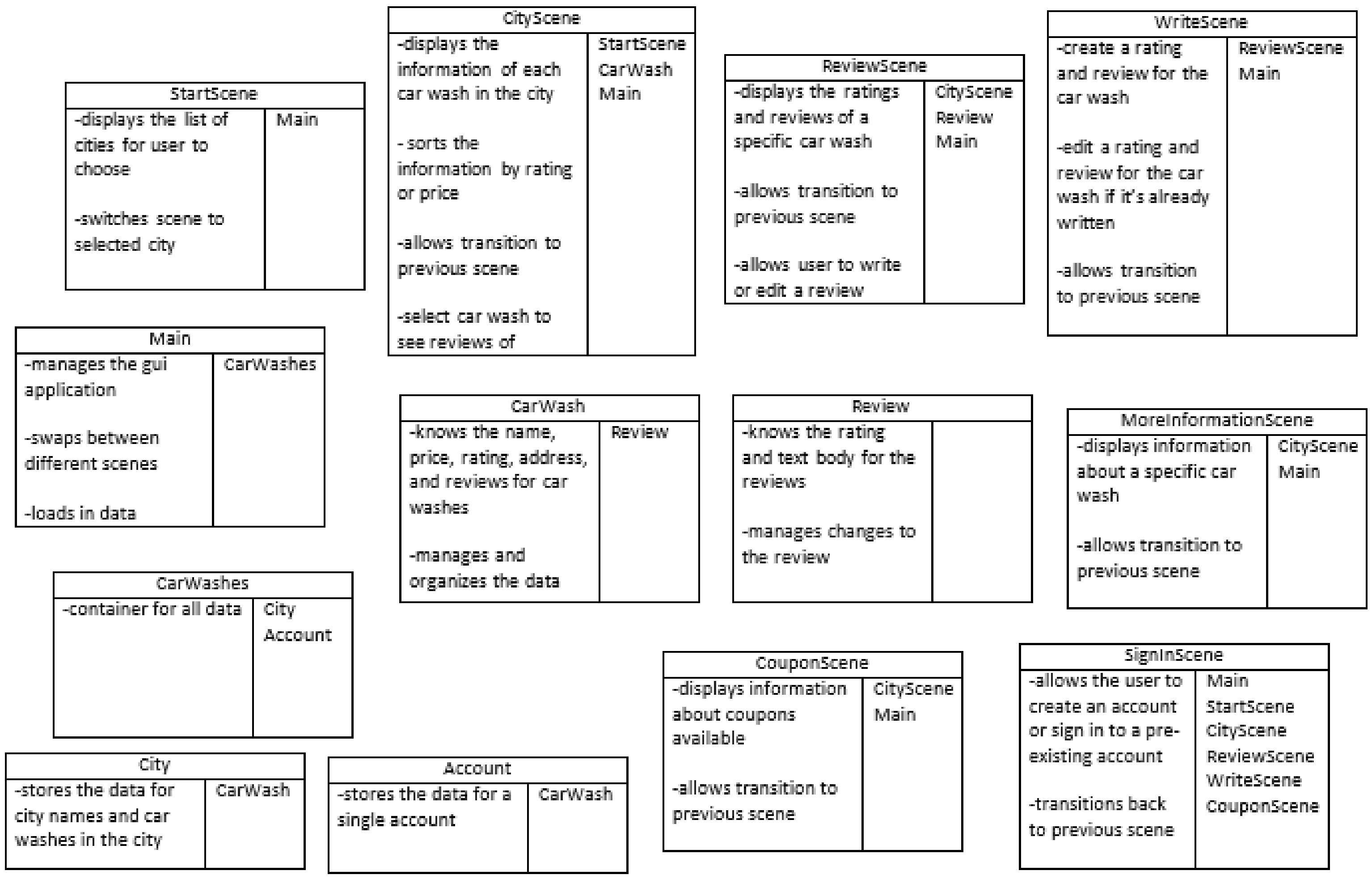
8. Use Case Diagram



9. Class Diagram



10. CRC Cards



StartScene	
-displays the list of cities for user to choose -switches scene to selected city	Main

CityScene	
-displays the information of each car wash in the city - sorts the information by rating or price -allows transition to previous scene -select car wash to see reviews of	StartScene CarWash Main

ReviewScene	
-displays the ratings and reviews of a specific car wash -allows transition to previous scene -allows user to write or edit a review	CityScene Review Main

WriteScene	
-create a rating and review for the car wash -edit a rating and review for the car wash if it's already written -allows transition to previous scene	ReviewScene Main

Main	
-manages the gui application -swaps between different scenes -loads in data	CarWashes

CarWash	
-knows the name, price, rating, address, and reviews for car washes -manages and organizes the data	Review

Review	
-knows the rating and text body for the reviews -manages changes to the review	

MoreInformationScene	
-displays information about a specific car wash -allows transition to previous scene	CityScene Main

CarWashes	
-container for all data	City Account

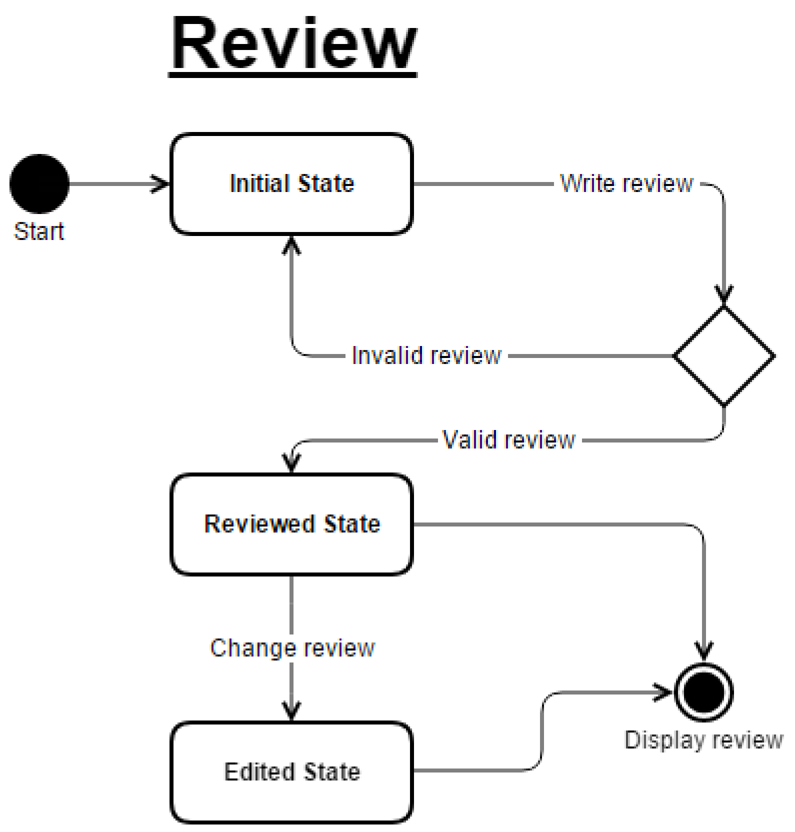
CouponScene	
-displays information about coupons available -allows transition to previous scene	CityScene Main

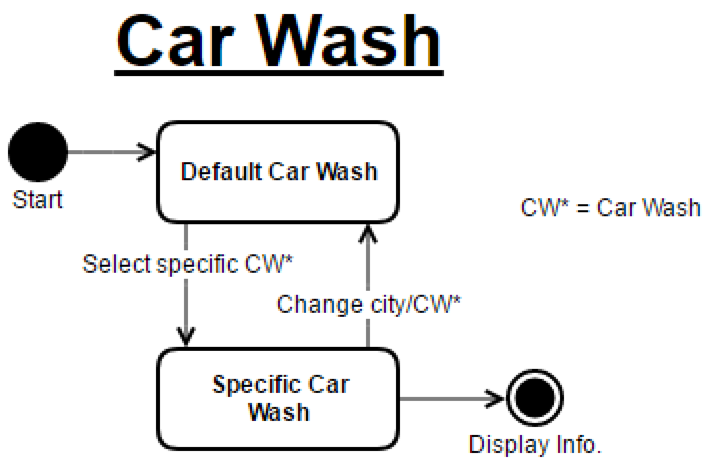
SignInScene	
-allows the user to create an account or sign in to a pre-existing account -transitions back to previous scene	Main StartScene CityScene ReviewScene WriteScene CouponScene

City	
-stores the data for city names and car washes in the city	CarWash

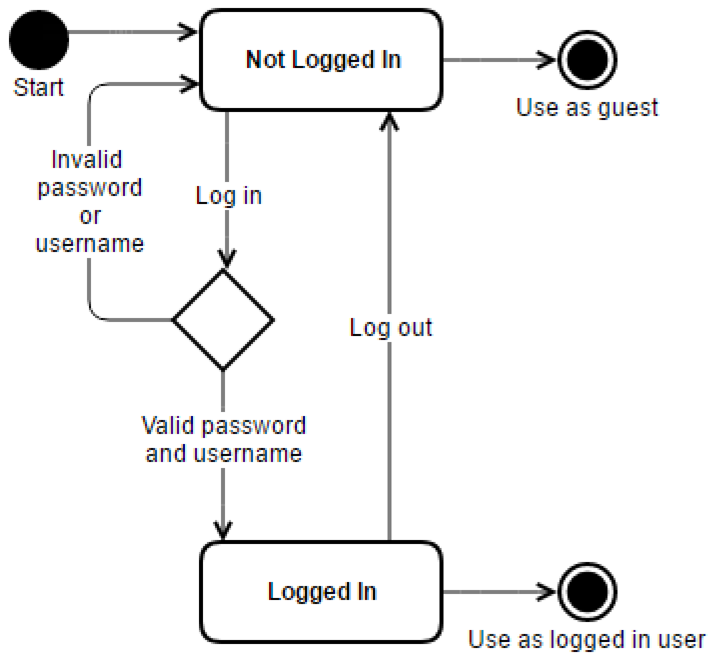
Account	
-stores the data for a single account	CarWash

11. State Diagrams

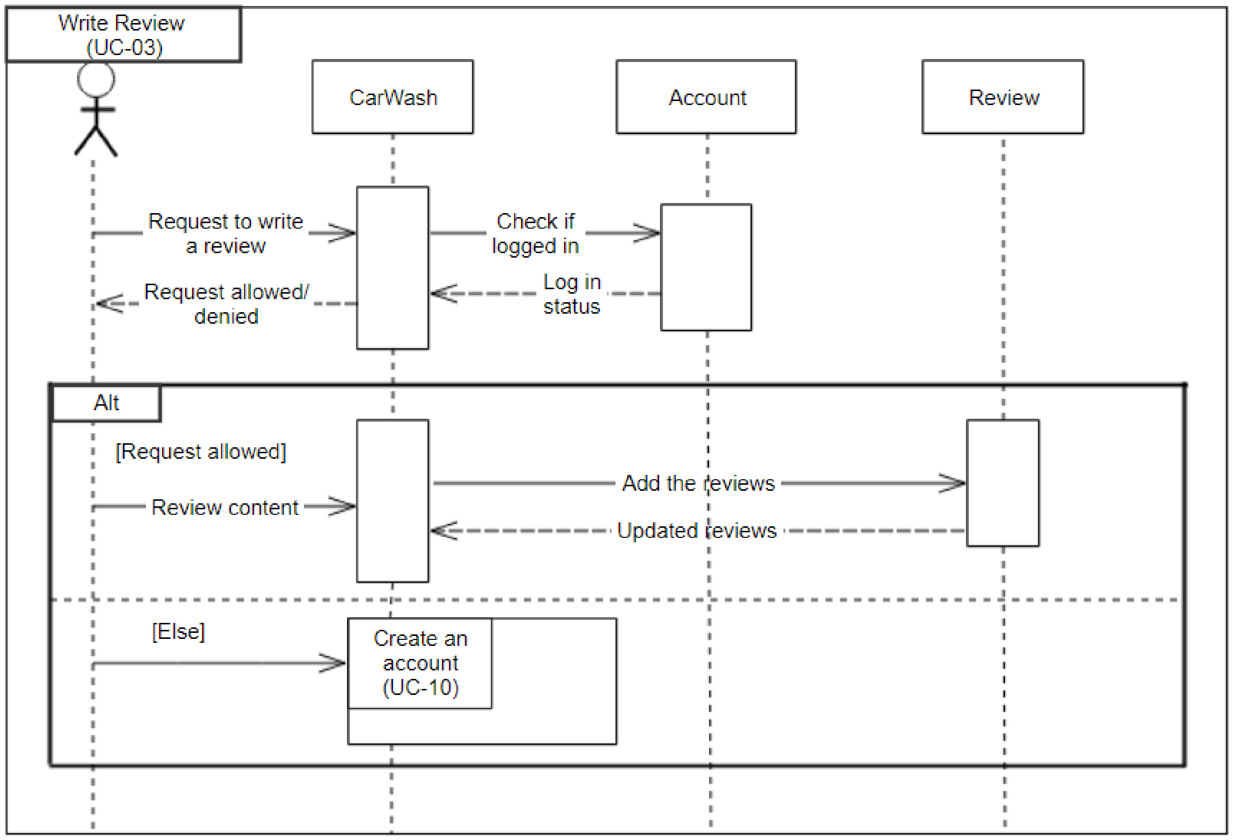




Account


Start
Not Logged In
Use as guest
Invalid password or username
Log in
Log out
Valid password and username
Logged In
Use as logged in user

12. Sequence Diagram



13. Test Cases

Test Case 1

Test Case #: TC_01	**Test Case Name:** Test city selection	Page 1 of 1
System: Friendly Car Wash	**Subsystem:** City selection	
Designed by: Josue Crandall	**Design Date:** 11/16/2017	
Executed by:	**Execution Date:**	
Short Description: Tests selecting city function.		

Pre-Condition:
Program has loaded data successfully.

Step	Action	Expected System Response	Pass /Fail	Comment
1	User selects city from menu.	System should highlight the city selected.		
2	User presses Select button after selecting a city.	Scene transitions to the car wash display scene		
3	**Check post-condition**			

Post-Condition: That the data loaded is for the proper city and the city name is in the title

Test Case 2

Test Case #: TC_02	**Test Case Name:** Test carwash sorting	Page 1 of 1
System: Friendly Car Wash	**Subsystem:** Carwash display	
Designed by: Josue Crandall	**Design Date:** 11/16/2017	
Executed by:	**Execution Date:**	
Short Description: Tests sorting carwashes by price or rating.		

Pre-Condition:
User selected city and the proper data was loaded and displayed.

Step	Action	Expected System Response	Pass /Fail	Comment
1	Check that the car washes are displayed in alphabetical order.			
2	Press sort by rating button.	A graphical change in the heading should display that the sorting has been done. And the carwashes should now be sorted by rating in descending order.		
3	Press sort by price button.	A graphical change in the heading should display that the sorting has been done. And the carwashes should now be sorted by price in ascending order.		
4	**Check post-condition**			

Post-Condition: The display of carwashes should be sorted, and the a visual marker of how the list has been sorted should be visible.

Test Case 3

Test Case #: TC_03	**Test Case Name:** View reviews	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** Reviews	
Designed by: Michael Duong	**Design Date:** 11/16/17	
Executed by:	**Execution Date:**	
Short Description: Test the ability to see reviews		

Pre-Condition:
The user has selected a city and is viewing the list of car washes for that city.

Step	Action	Expected System Response	Pass /Fail	Comment
1	Click on a car wash in the list of car washes.	The system highlights the specific car wash.		
2	Click on “Select” button	The system changes to the review scene.		
3	**Check post-condition**			

Post-Condition: A list of reviews for the correct car wash is shown.

Test Case 4

Test Case #: TC_04	**Test Case Name:** Edit review	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** Reviews	
Designed by: Michael Duong	**Design Date:** 11/16/17	
Executed by:	**Execution Date:**	
Short Description: If a review has already been written, the user should be able to change it.		

Pre-Condition:
The user has already gone through the steps to write a review and is now in the Reviews scenes of a specific car wash that already has a review written.

Step	Action	Expected System Response	Pass /Fail	Comment
1	Click on “Edit review”	The system brings up the previously made review by switching to the “Write Review” scene.		
2	Change the rating to a different number.	The new number is displayed in the box.		
3	Change the review text to something different.	The new text is displayed in the box.		
4	Click on “Submit”	The system takes the user back to the Reviews scene.		
5	**Check post-condition**			

Post-Condition: The user’s review has been changed to the new text and rating.

Test Case 5

Test Case #: TC_05	**Test Case Name:** “Edited” caption	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** Write review scene	
Designed by: Hoang Tran	**Design Date:** 11/16/2017	
Executed by:	**Execution Date:**	
Short Description: Show “Edited” caption next to revised review		

Pre-Conditions: User uses the Friendly Car Washer app
- User chooses the city and a specific car wash.
- User chooses “Write review”, writes review and hit “Submit” button.
- User chooses “Edit review”, edits the review and hit “Submit” button.

Step	Action	Expected System Response	Pass /Fail	Comment
1	After user edits review and hits “Submit” button	The system will display the “Edited” caption on the review that has been revised		
2	**Check Post condition**			

Post-Condition: The edited review is saved to the database.

Test Case 6

Test Case #: TC_06	**Test Case Name:** See available coupon	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** City scene	
Designed by: Hoang Tran	**Design Date:** 11/16/2017	
Executed by:	**Execution Date:**	
Short Description: See available coupons at all car wash locations		

Pre-Conditions:
- User uses the Friendly Car Washer app
- User chooses city.
- User hits “See available coupons.”

Step	Action	Expected System Response	Pass /Fail	Comment
1	User hits “See available coupons”	The system will display coupons of all car wash locations		
2	**Check Post condition**			

Post-Condition: User can choose another city to view more coupons available for that city.

Test Case 7

Test Case #: TC_07	**Test Case Name:** Create an Account	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** Create account	
Designed by: Tram Nguyen	**Design Date:** 11/16/17	
Executed by:	**Execution Date:**	
Short Description: User can create their account with some of their information.		

Pre-Condition:
The user did not log in to the system.

Step	Action	Expected System Response	Pass /Fail	Comment
1	The user clicked the "Create an account/ Sign in" link on the top right.	The system will give user a form asking them to fill out the user name, password they want for their account.		
2	The user fills in all the fields.	The system created an account and bring the user back to the previous scene with the sign in account they just created.		
3	**Check post-condition**			

Post-Condition: An account contains user's information has been made.

Test Case 8

Test Case #: TC_08	**Test Case Name:** Log In	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** Login	
Designed by: Tram Nguyen	**Design Date:** 11/16/17	
Executed by:	**Execution Date:**	
Short Description: User can log in to the system by using the account they created.		

Pre-Condition:
The user did not log into the system.

Step	Action	Expected System Response	Pass /Fail	Comment
1	The user did not log in to the system and they clicked the “Create an account/ Sign In” link on the top right.	The system prompts them a form asking to fill in username and password.		
2	The user fills in the correct password and usename.	The account is signed in and the link “Create an account/ Sign in” changed into “Log out”.		
3	**Check post-condition**			

Post-Condition: The user logged in to the system by using their account.

Test Case 9

Test Case #: TC_09	**Test Case Name:** See more information	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** More information	
Designed by: Huy Tran	**Design Date:** 11/16/2017	
Executed by:	**Execution Date:**	
Short Description: Test the “See more information” button		

Pre-Conditions:
The user has chosen a city
The user has chosen a car wash location

Step	Action	Expected System Response	Pass /Fail	Comment
1	Select “See more information”	The system displays additional information of the car wash location		
2	**Check post-condition**			

Post-Condition:
More information about the car wash location is displayed for the user to see and use.

Test Case 10

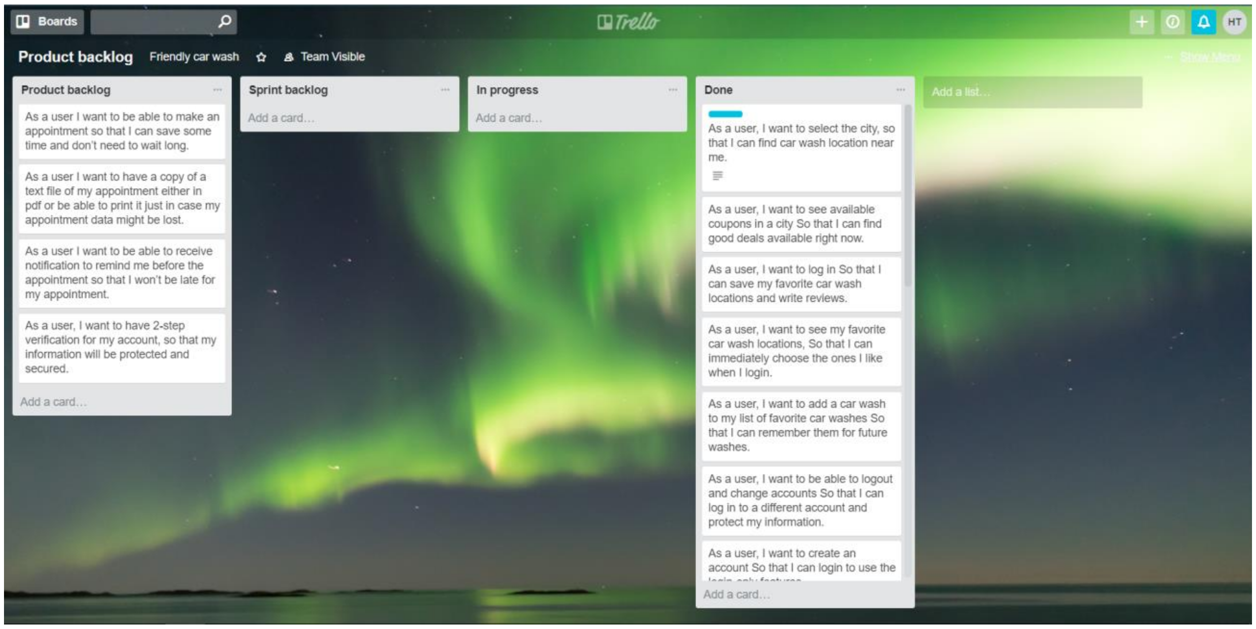
Test Case #: TC_10	**Test Case Name:** Change city	Page 1 of 1
System: Friendly Car Washers	**Subsystem:** City selection	
Designed by: Huy Tran	**Design Date:** 11/16/2017	
Executed by:	**Execution Date:**	
Short Description: Test the “Change city” button		

Pre-Conditions:
The user must already selected a city or a car wash location

Step	Action	Expected System Response	Pass /Fail	Comment
1	Click “Change city”	The system displays the list of cities for user to choose		
2	**Check post-condition**			

Post-Condition:
The user can change the city immediately without going back in each scene

14. Sprint Backlog - Trello



Product backlog

- As a user I want to be able to make an appointment so that I can save some time and don't need to wait long.
- As a user I want to have a copy of a text file of my appointment either in pdf or be able to print it just in case my appointment data might be lost.
- As a user I want to be able to receive notification to remind me before the appointment so that I won't be late for my appointment.
- As a user, I want to have 2-step verification for my account, so that my information will be protected and secured.

Add a card...

Sprint backlog

Add a card...

In progress

Add a card...

Done

- As a user, I want to select the city, so that I can find car wash location near me.
- As a user, I want to see available coupons in a city So that I can find good deals available right now.
- As a user, I want to log in So that I can save my favorite car wash locations and write reviews.
- As a user, I want to see my favorite car wash locations, So that I can immediately choose the ones I like when I login.
- As a user, I want to add a car wash to my list of favorite car washes So that I can remember them for future washes.
- As a user, I want to be able to logout and change accounts So that I can log in to a different account and protect my information.
- As a user, I want to create an account So that I can login to use the

Add a card...

Add a list...

15. Pre-Game Planning

<u>First Iteration</u>

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.

Second Iteration

For the second iteration, we originally had some features we planned to implement based on the excess user stories from our initial pre-game planning. After meeting with the customer and hearing the customer's suggestions, we realized that we should include some of their suggestions to keep them satisfied at the end of this iteration. At the same time, though, we wanted to continue to improve our software since the customer's suggestions were mostly aesthetic changes. As a compromise, we added some changes suggested by the user and some changes we initially wanted to make. Just as before, we played a poker game with all of the possible user stories to pick the ones that would take the least amount of time. Then, we chose the quickest ones from both piles: user suggestions and original ideas. Because of this though, we may not have added as many features as we initially wanted and not all of the customer's desires were met, but we felt that we still made good forward progress with the chosen user stories.

Third Iteration

For the third iteration, we had to decide which features made the final cut. During the second iteration, the customer had no immediate suggestion for improvement but we needed to have a richer feature set. We pooled together the features our team members wanted to see in the final product. Like the last two iterations, we played a planning poker game to decide on which feature would make it to the final iteration. During the game, we chose the most important ones from both piles: original ideas and user suggestions. Throughout the game, we made some tough cuts. As a result, our final product has the necessary features, and also maintains the concise and effective user interface. We also set out the substantial graphical and quality of life improvements because the end-user is going to operate the final cut.

16. Staging or Grooming

First Iteration

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.

Second Iteration

For this iteration, our biggest difficulty was balancing the user stories suggested by the customer with those we wanted to implement ourselves to add more functionality to our application (which we had already thought of since the first iteration). At first, we focused more on the user stories relating to features we wanted to implement, but as we were working, we realized that we neglected many of the desires of the customer. Since the customer would be looking at the application at the end of the iteration, we chose then to leave what features we had already implemented, which were see more information and resizing the window, and then switch gears to focus on the desires of the customer, making them a bigger priority. In the end, that resulted in our iteration having a hybrid of both categories, which we felt was a good decision since we made progress and it appeased both the customer and developers.

Third Iteration

For this iteration, we were expecting difficulties because of the substantial number of new features being added. However, because of the refactoring we had done during our second iteration, due to the Agile development process. We found the process of adding new features streamlined. We found that not only could we add the features that we had user stories for, we could also spend development time improving non-functional attributes, such as adding CSS and background image as well as many quality features like a theme.

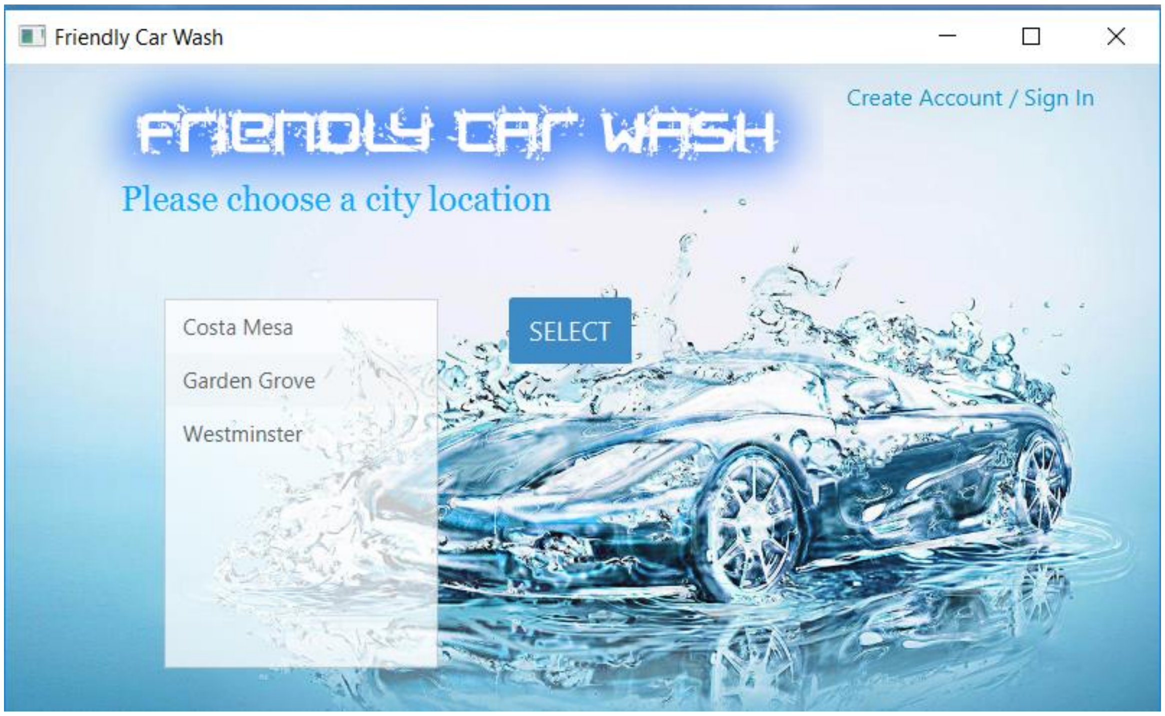
17. Development Process

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.

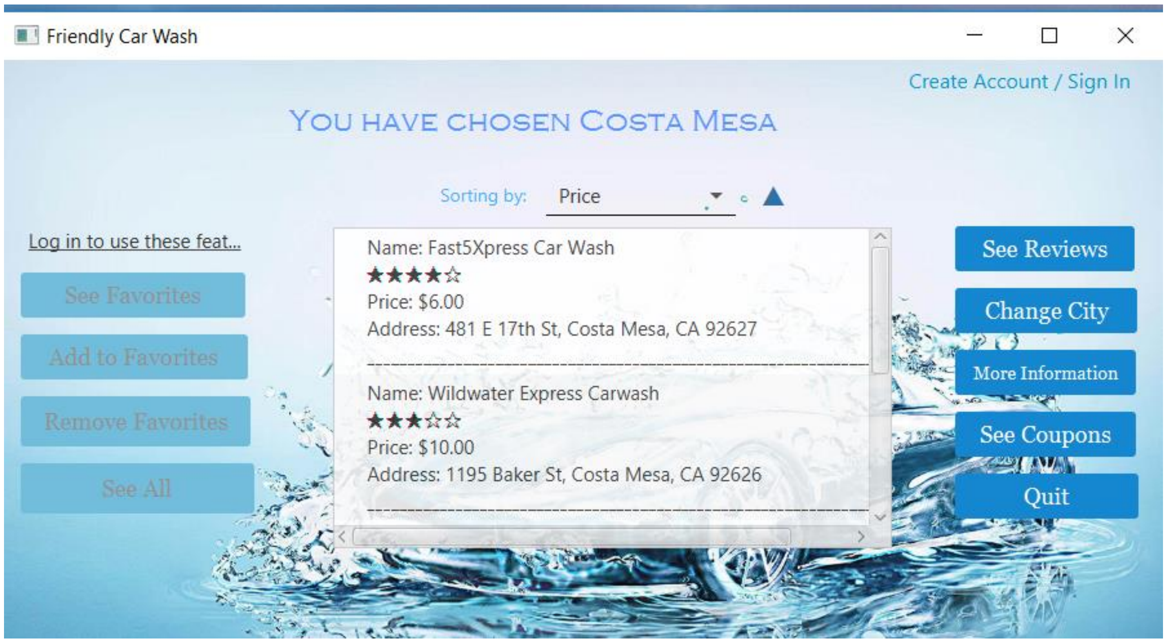
While we were programming, in order to share code, since we were unable to set up Github at the start, for this first iteration, we had to share code using Google Drive. The first week was spent on setting up the work environment on everyone’s computers and getting used to the tools, mainly Scene Builder 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, 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.

18. User Manual



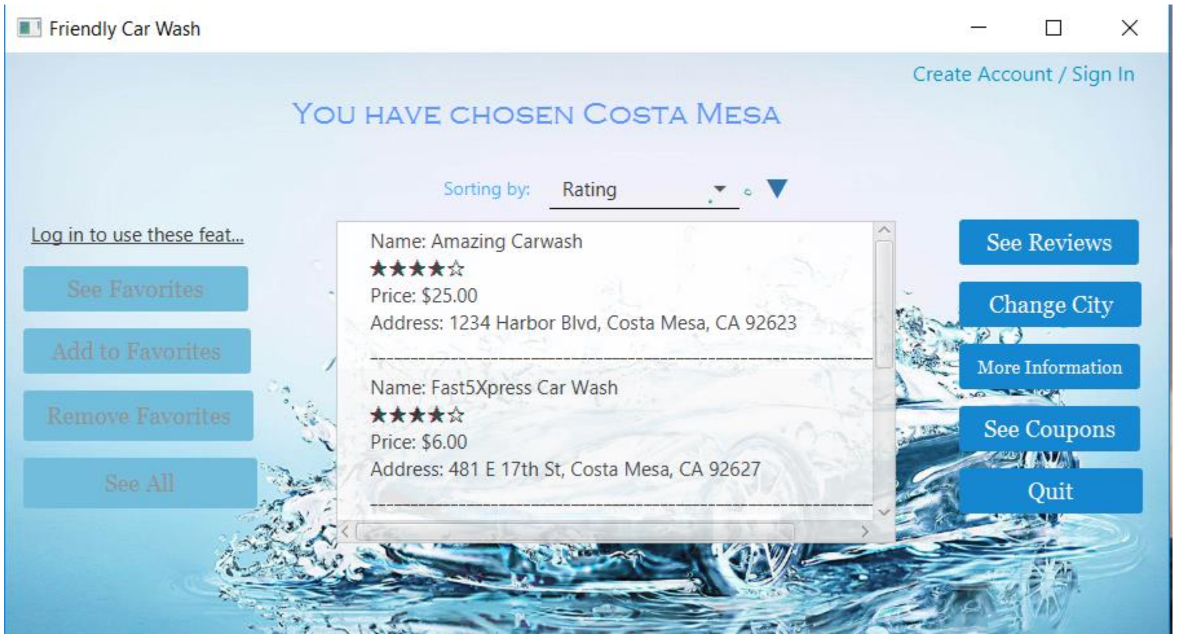
1. User runs the program.
2. The program displays the opening scene.
3. User chooses the city.
4. User hits select.



Name: Fast5Xpress Car Wash
★★★★☆
Price: $6.00
Address: 481 E 17th St, Costa Mesa, CA 92627

Name: Wildwater Express Carwash
★★★☆☆
Price: $10.00
Address: 1195 Baker St, Costa Mesa, CA 92626

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

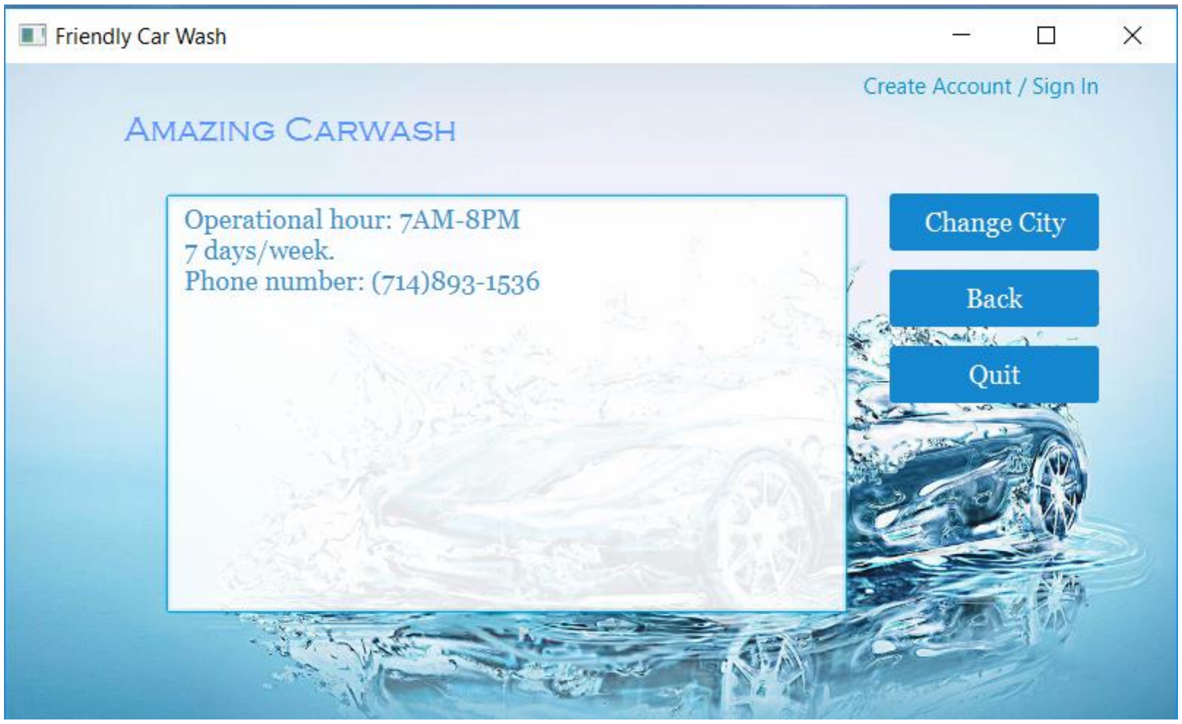


Sorting by: Rating

Name: Amazing Carwash
★★★★☆
Price: $25.00
Address: 1234 Harbor Blvd, Costa Mesa, CA 92623

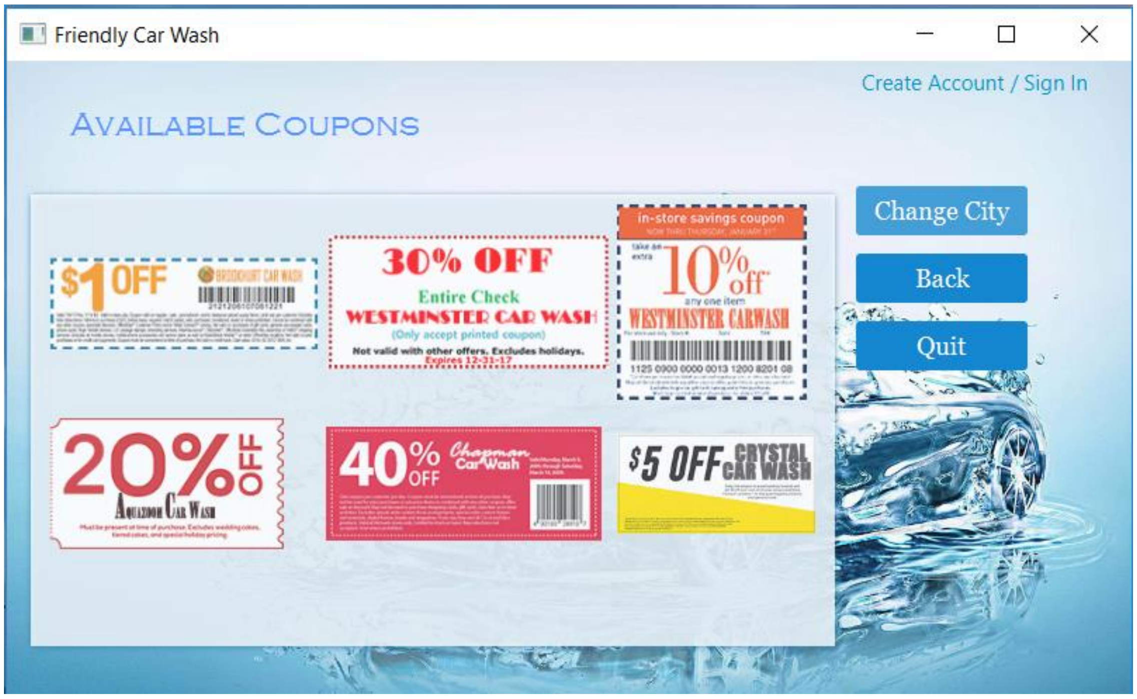
Name: Fast5Xpress Car Wash
★★★★☆
Price: $6.00
Address: 481 E 17th St, Costa Mesa, CA 92627

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

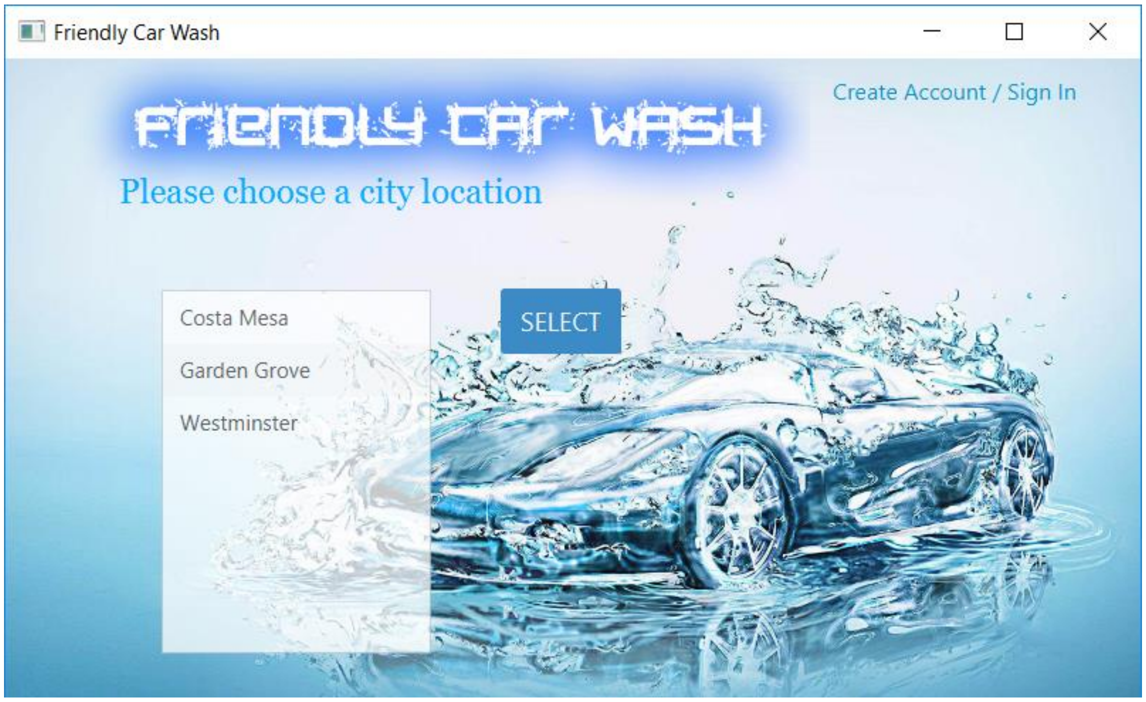


Operational hour: 7AM-8PM
7 days/week.
Phone number: (714)893-1536

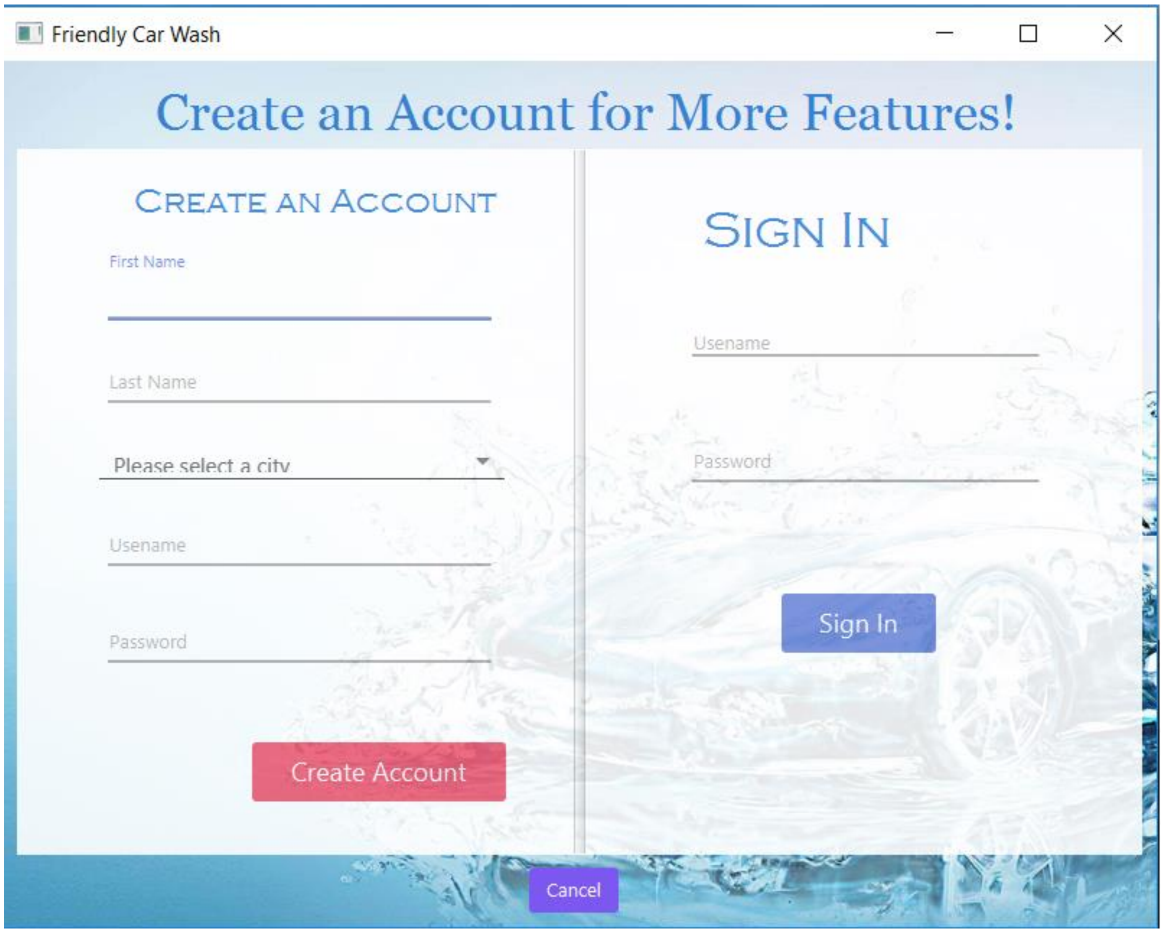
1. User selects a car wash and hits "More Information"
2. The program displays further information about the car wash



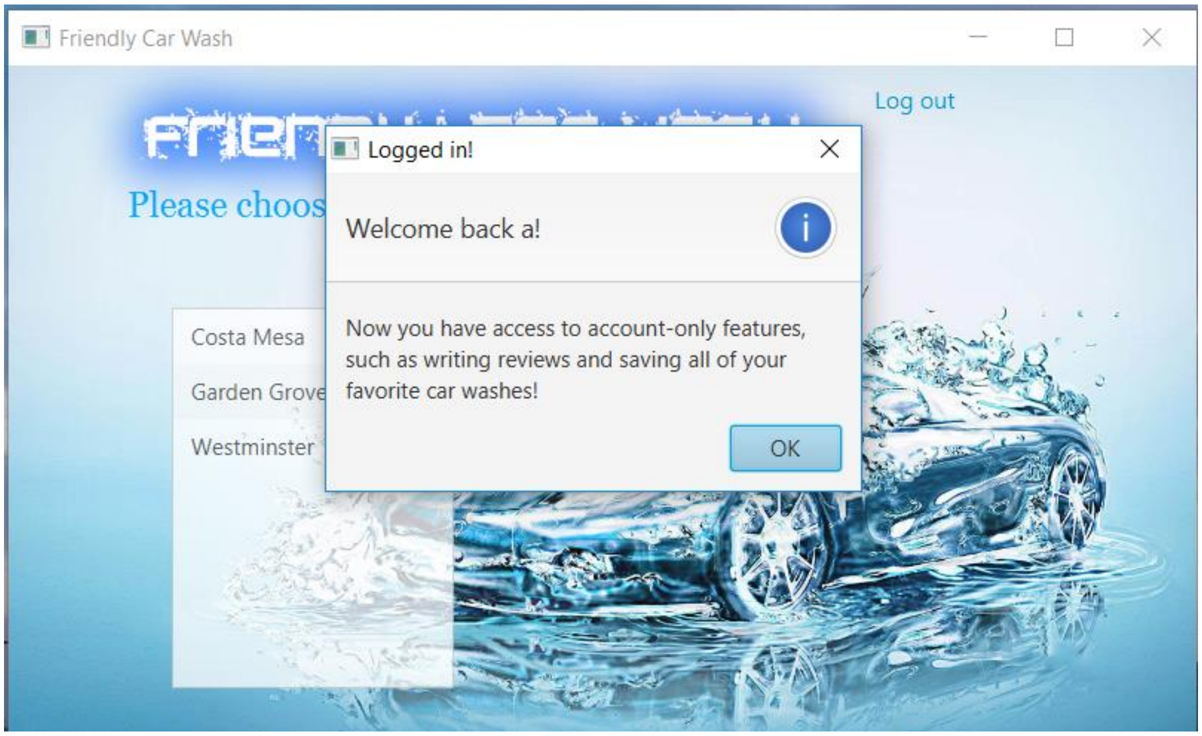
1. User selects “See Coupon”
2. Program displays available coupons in the city



1. The user hits “Change City.”
2. The program goes back to the start scene and displays list of cities.



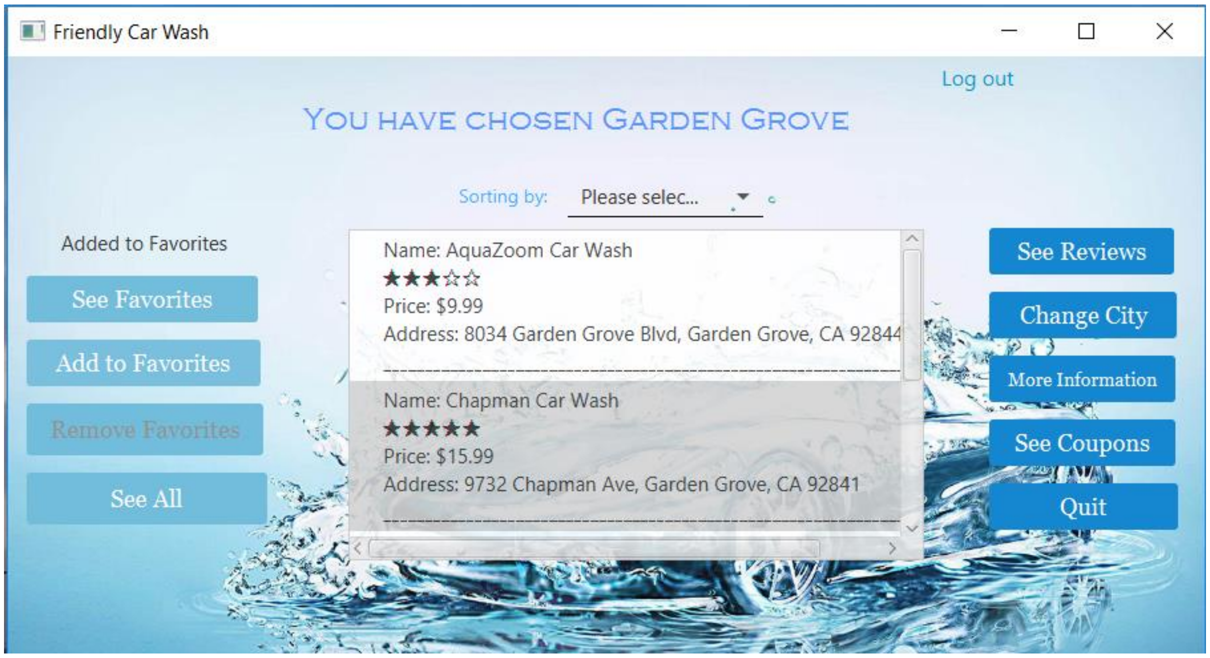
1. User chooses "Create Account/ Sign in."
2. The program displays a scene for user to create an account or sign in.
3. The user fills all the blank and hits "Create Account."



Welcome back a!

Now you have access to account-only features, such as writing reviews and saving all of your favorite car washes!

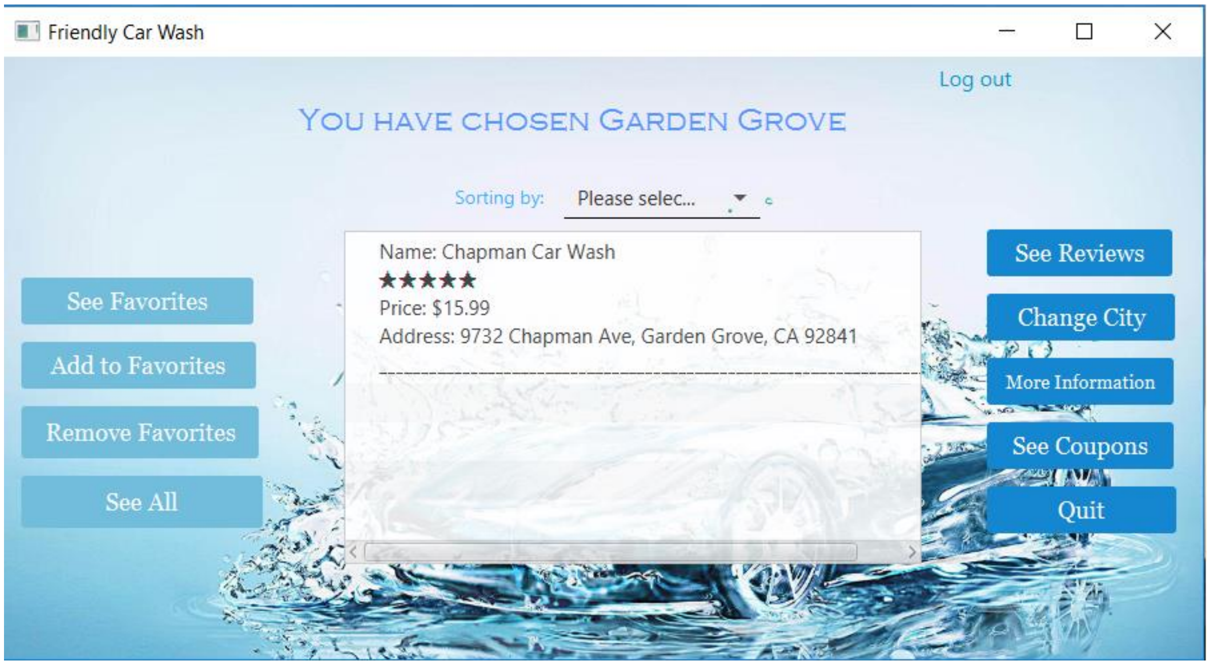
1. The user enters the username and password and clicks “Sign In.”
2. The program displays welcome scene



Name: AquaZoom Car Wash
★★★☆☆
Price: $9.99
Address: 8034 Garden Grove Blvd, Garden Grove, CA 92844

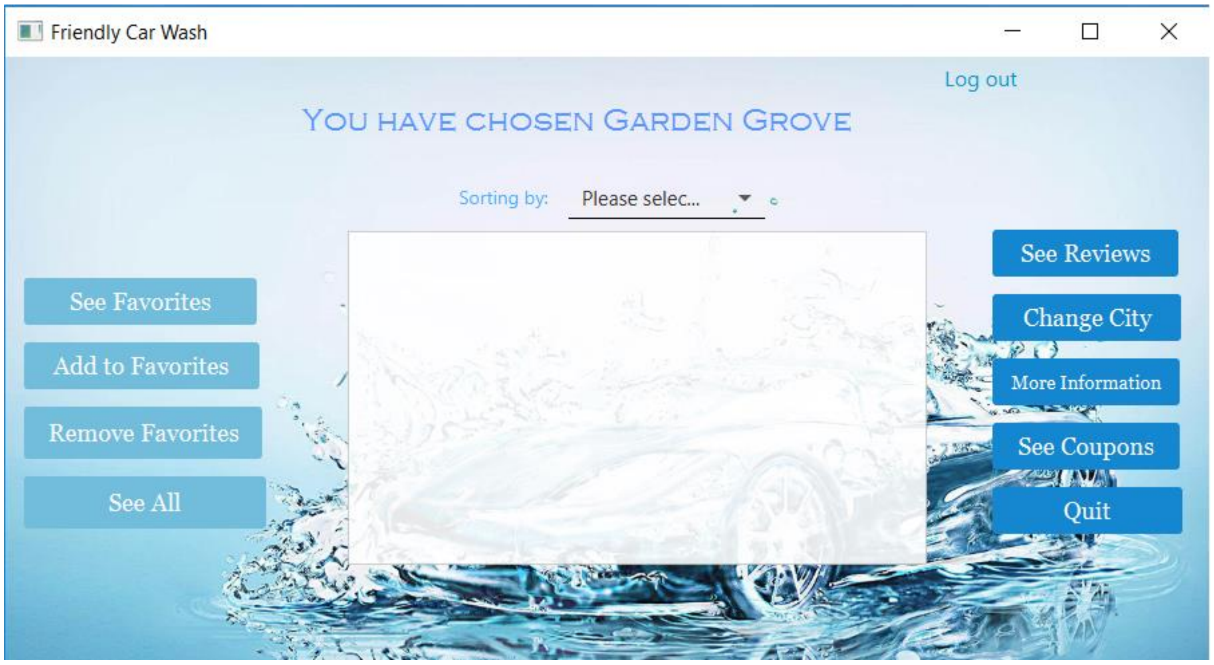
Name: Chapman Car Wash
★★★★★
Price: $15.99
Address: 9732 Chapman Ave, Garden Grove, CA 92841

1. The user chooses a car wash and clicks "Add to Favorites."
2. The program adds the car wash to favorites.

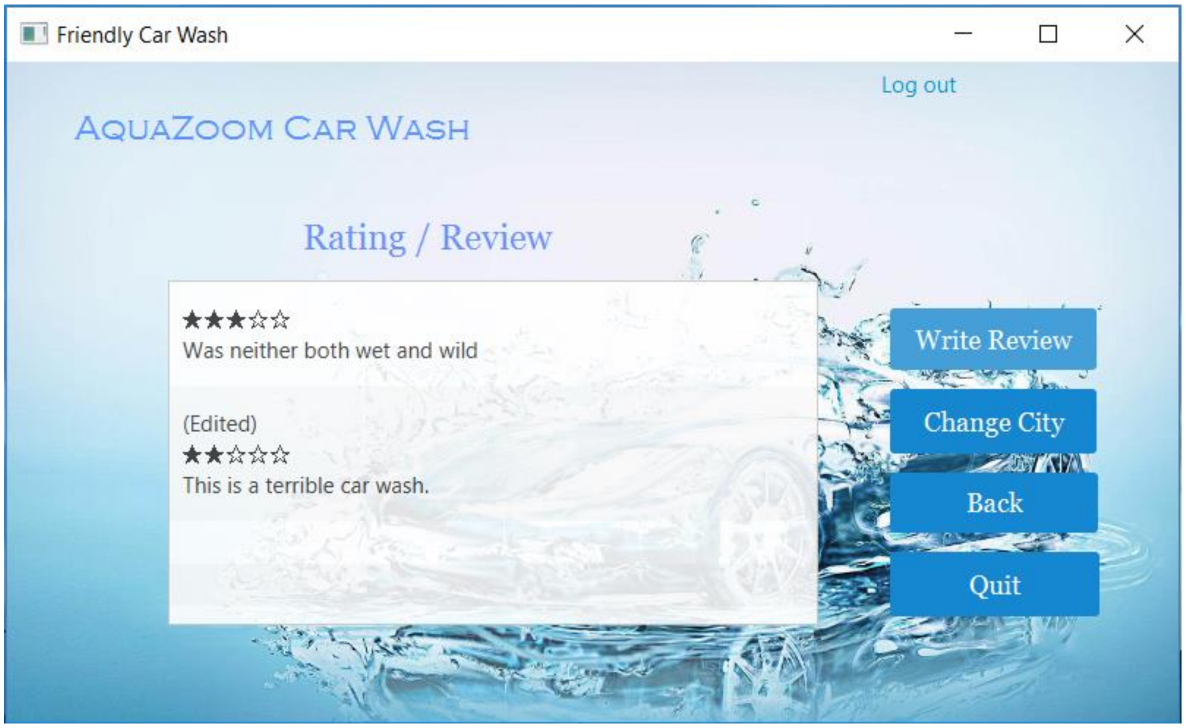


Name: Chapman Car Wash
★★★★★
Price: $15.99
Address: 9732 Chapman Ave, Garden Grove, CA 92841

1. The user hits “See Favorites”
2. The program displays the favorite car wash.



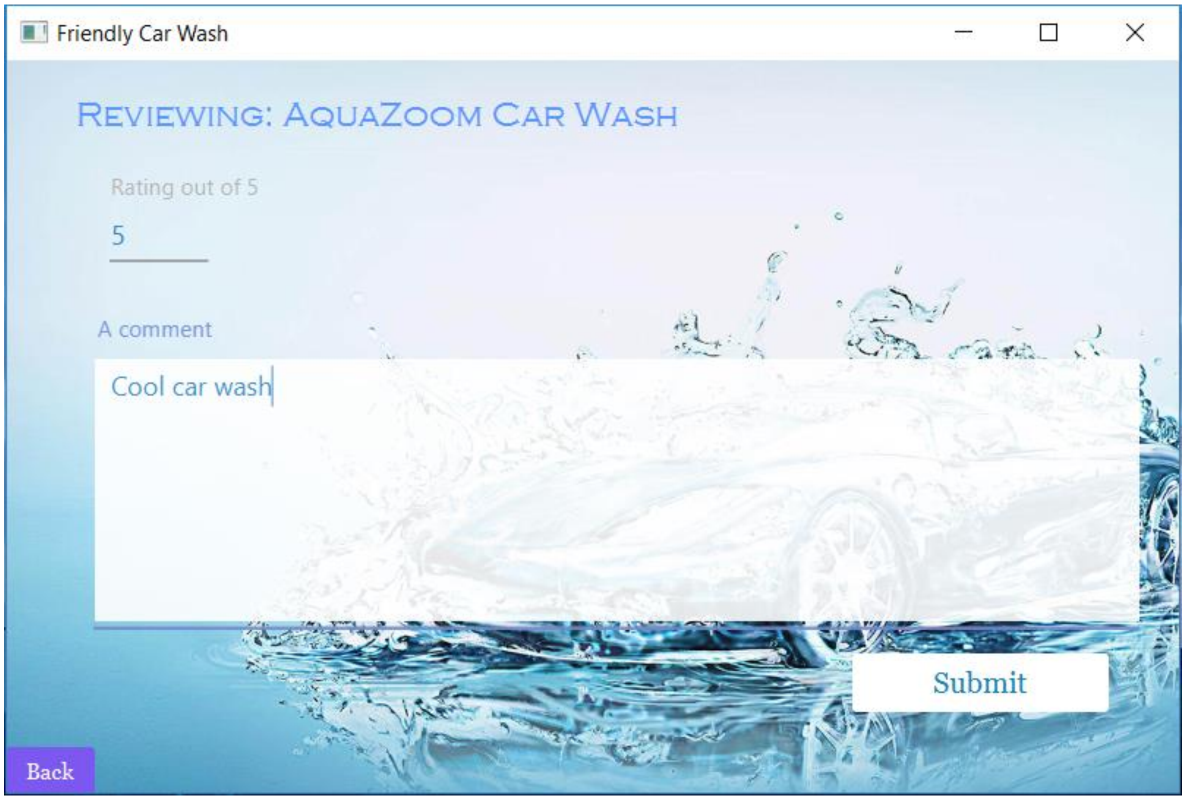
1. The user chooses a car wash and hits “Remove Favorites.”
2. The program removes the favorite car wash



★★★☆☆
Was neither both wet and wild

(Edited)
★★☆☆☆
This is a terrible car wash.

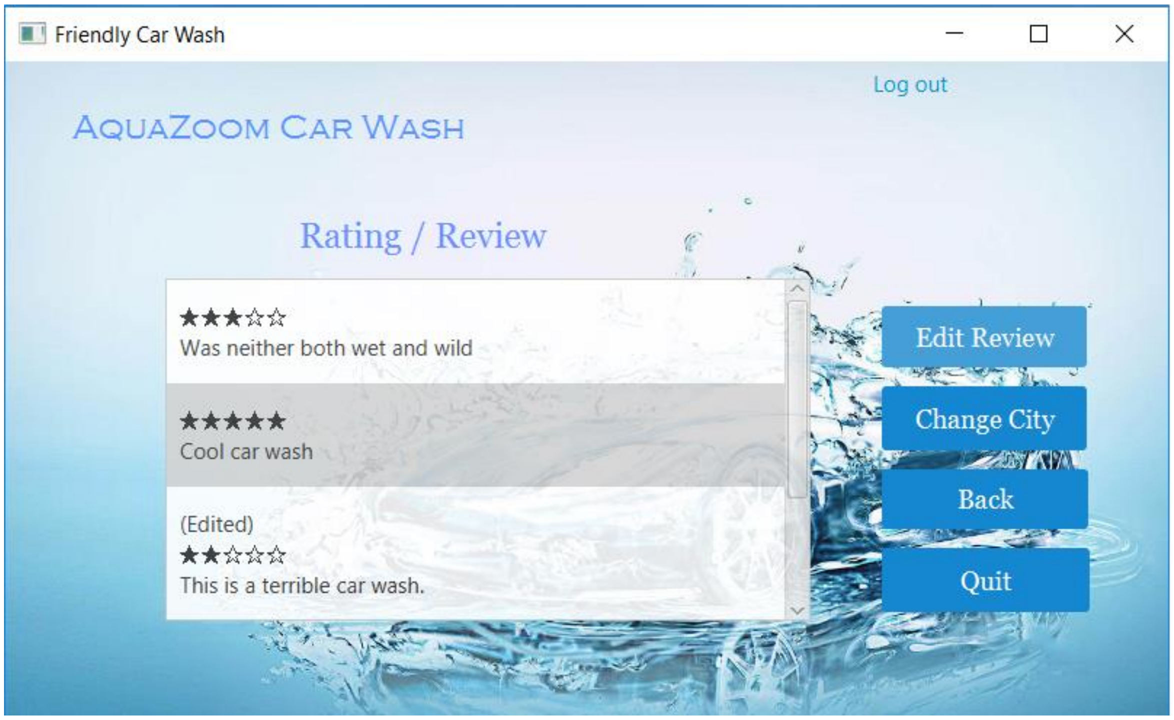
1. The user clicks “See Review”
2. The program displays all the review of the car wash



5

Cool car wash

1. The user clicks "Write Review."
2. The program displays a scene for user to write review



Friendly Car Wash

Log out

AquaZoom Car Wash

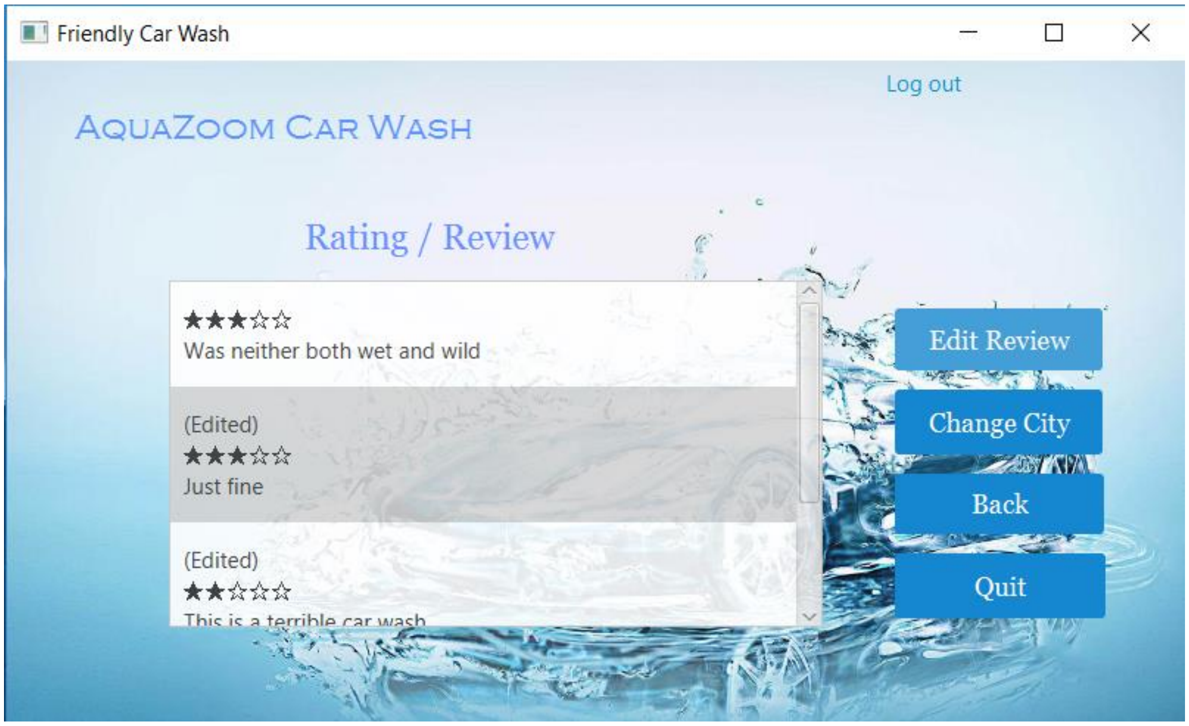
Rating / Review

★★★☆☆
Was neither both wet and wild

★★★★★
Cool car wash

(Edited)
★★☆☆☆
This is a terrible car wash.

1. The user clicks “Submit.”
2. The program displays the user’s review.



★★★☆☆
Was neither both wet and wild

(Edited)
★★★☆☆
Just fine

(Edited)
★★☆☆☆
This is a terrible car wash

1. The user hits “Edit Review.”
2. The program displays a scene for user to edit the review.
3. The user hits “Submit.”
4. The program displays the latest review, marked by “Edited.”