

Frontier Rental Car Service

Every year car rental companies handle over millions of reservations. A large number of customers are not local, so looking for a good deal on a car rental can be difficult if the customer does not know where to start. The high cost and time to find a rental in some areas also affects the user's ability to find a suitable car that he or she needs. The car rental services usually fill up fast and are quite expensive. It is easy to find a round-trip rental, but looking for a one-way or long-term car rental can be hard. Once summer and/or holiday season begins, they are in high demand and it is a challenge to find the right car. As of now, customers have to rely on big car rental companies to find a suitable and affordable accommodation for what they need. Therefore, this web application's goal is to create an online platform for locals and foreigners to share cars where ever they are. This application aims to let users manage their account easily, and with an ease of use, save time and money.

Frontier Car Rental Service will be for all and user-friendly, that is why we have made a few functional requirements to assist the user:

1. User accounts (customer)
 - a. To rent a car, the app shall require users to create an account using their name, phone number, address, email and driver license number.
2. Sorting Algorithm
 - a. The application must have a functionality to sort searching based on the user's criteria such as car type, model, number of doors, etc.
3. Matching Algorithm
 - a. This functionality shall identify the right vehicle that matches the user's search
4. Queuing System (FCFS-first come first serve)
 - a. The application shall be using the queuing system to prioritize users who have reserved a car first, in the order that the reservations were placed
5. Interactive Map View
 - a. The search view must have the option to switch between a listview and an interactive map, which places pins at the location of the cars.
6. User accounts (host)
 - a. Additionally to the customer account the app shall allow an account to be switched to a host, with which one can offer their own car for rent

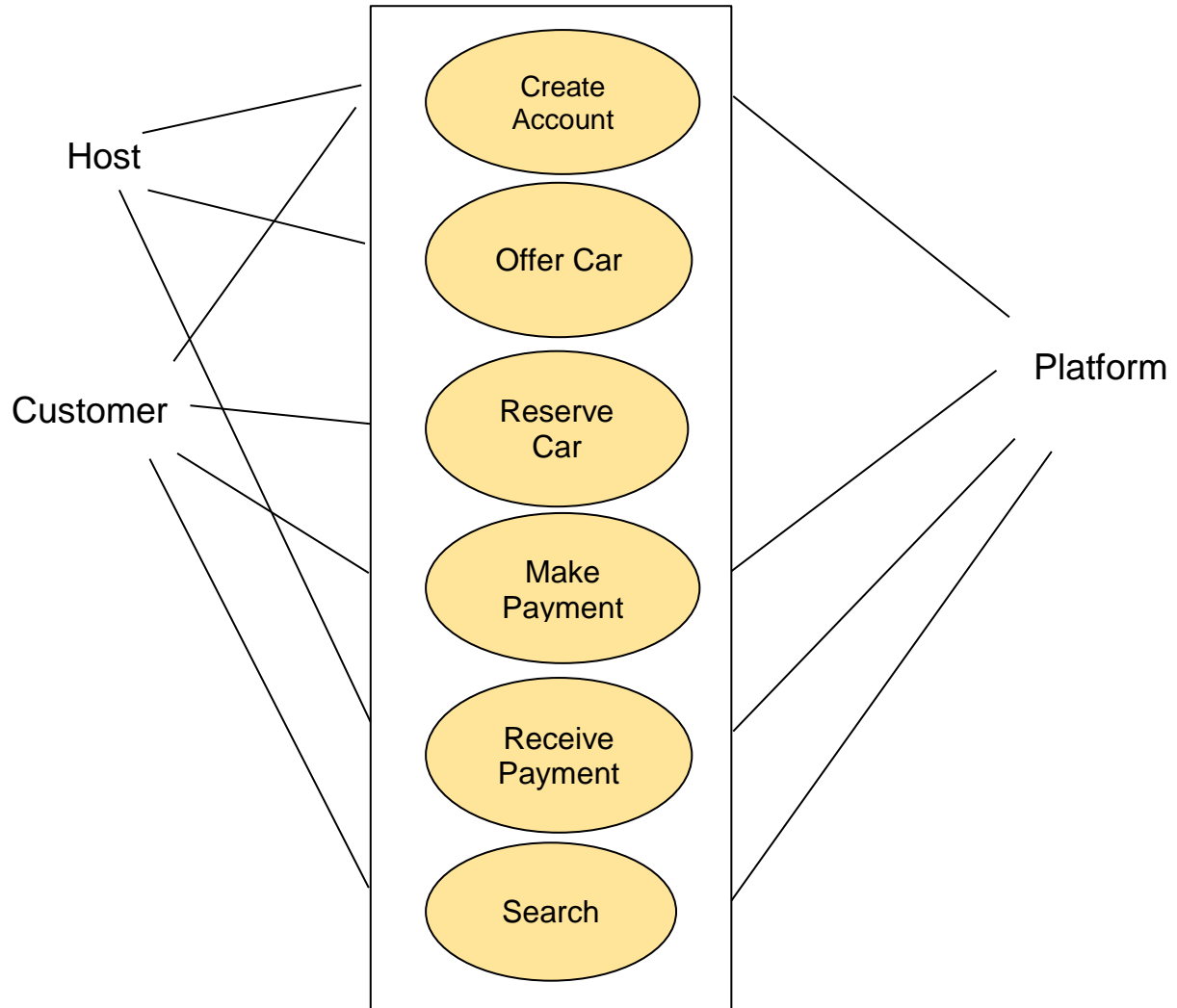
Additionally, to the six functional requirements mentioned above, the application also required to have four nonfunctional requirements to supports the users.

1. GUI
 - a. The application's look must be streamlined with one or more accent colors throughout all pages
2. Load times
 - a. Load times of search results shall be kept to a minimum, this is achieved through dynamical loading and limiting to ~20 cars
3. Navigation
 - a. There must be a navigation bar on the top which lets the user switch between different views
4. Responsive Design
 - a. The app shall offer all functions no matter what platform (Desktop, Mobile or Tablet) it is used from

The functional specification also includes a UML (use case diagram) that contains six use cases with a minimum of two actors and use case forms with three descriptions for each case.

The goal of this web application is to enable the users' rental experience as easy, safe, efficient and affordable as possible. With the successful implementation of the Frontier Car Rental web application, users will have the ability to find a suitable and right car to rent that he or she needs including car availability, price range, and location.

UML (use case diagram)



Use Case Description

Use Case name:	Create Account
Product name:	Frontier Rental Car Service
Team:	Frontier
Date:	02/22/2017

1. Goal

The customer creates an account on the website.

2. Summary

A customer is required to fill out name, phone number etc. to create an account.

3. Actors

Actor 1: Customer
Actor 2: The platform
etc.

4. Preconditions

- The website is not in logged-in status

5. Trigger

The customer selects the “Sign Up” option

6. Primary Sequence

Step	Action
1	The webpage is directed to the member registration page.
2	The customer enters his/her name, phone number, address, email address, and driver license number into the blanks.
3	The customer confirms the correctness of the information.

4	The customer submits the information.
---	---------------------------------------

7. Primary Postconditions

- The customer can login to the website.
- The logged-in customer can rent a car (See use case “Reserve Car”).

8. Alternate Sequences

Alternate Trigger

The customer enters the information in wrong format (e.g. name contains number or symbol.).

Step	Action
1	The system displays a message to the customer.
2	The customer shall enter the information again.
<i>etc.</i>	

Alternate Postconditions

- The customer can login to the website.
- The logged-in customer can rent a car (See use case “Reserve Car”).

Alternate Trigger

Step	Action
1	
2	
<i>etc.</i>	

Alternate Postconditions

- -
- etc.*

9. Nonfunctional Requirements

- The system responds to each customer input within 15 seconds.
- The system displays messages in English

10. Glossary

- Customer = a person who wants to register to create an account.
- The platform = a system that allows the customer to create an account, and maintains the information.

Use Case Description

Use Case name:	Make Payment
Product name:	Frontier Rental Car Service
Team:	Frontier
Date:	2/22/17

1. Goal

To make a payment after customer has selected cars to purchase.

2. Summary

The customer will provide payment information, duration of the trip, and pick up & return location.

3. Actors

Actor 1: Customer

Actor 2: Platform

4. Preconditions

- Customer is on the web page browsing the cars.

5. Trigger

The customer selects the “rent this car” option

6. Primary Sequence

Step	Action
1	The customer browse through the cars that are offering on the website.
2	The customer selects a car to rent.
3	Enter the information of the trip date, pick up & return location, and choose a payment method.

4	Make a payment with credit card, cash, or check
---	-------------------------------------------------

7. Primary Postconditions

- Customer is required to sign in to his/her account to rent a car
- The system will send out the confirmation automatically to the customer's email.

8. Alternate Sequences

Alternate Trigger

Unsuccessful order

Step	Action
1	The system displays the error message
2	The customer must redo the renting procedure again
<i>etc.</i>	

Alternate Postconditions

The customer will see the display message of the unsuccessful order, please try it again

Alternate Trigger

Step	Action
1	
2	
<i>etc.</i>	

Alternate Postconditions

-
-
- etc.*

9. Nonfunctional Requirements

- The system sends the confirmation to the customer within minutes.
- The system displays the message to the customers if there is an error occur.

10. Glossary

- Customer = a user who tries to make a payment to rent a selected car.
- The platform = a system that receives payment from customer.

Use Case Description

Use Case name:	Offer Car
Product name:	Frontier Rental Car Service
Team:	Frontier
Date:	02/22/2017

1. Goal

Edit and modify the database of the cars.

2. Summary

This enables the customer to browse different kinds of cars that are available

3. Actors

Actor 1: Host

Actor 2: The platform

4. Preconditions

- The host should login with their username and password.
- Go to the web page

5. Trigger

Update the database

6. Primary Sequence

Step	Action
1	Host needs to login to modify the database.
2	Add new car to the company's database.

3	Save the changes
<i>etc.</i>	

7. Primary Postconditions

- The customers can browse the updated database.
- The message will display to the host for successful update.

8. Alternate Sequences

Alternate Trigger

Fail to edit or modify the data on the website

Step	Action
1	The system displays the error message.
2	Try to reconnect to the server
<i>etc.</i>	

Alternate Postconditions

- The host is required to login in order to update the database.

Alternate Trigger

Step	Action
1	
2	
<i>etc.</i>	

Alternate Postconditions

-
- etc.*

9. Nonfunctional Requirements

- The system updates the database within few minutes.
- The system displays the message if the car is or is not successfully updated to the database.

10. Glossary

- Database = collection of information that is organized by the host, it is easy to access, modify and update.
- Host = a user who wants to offer a car for rent.