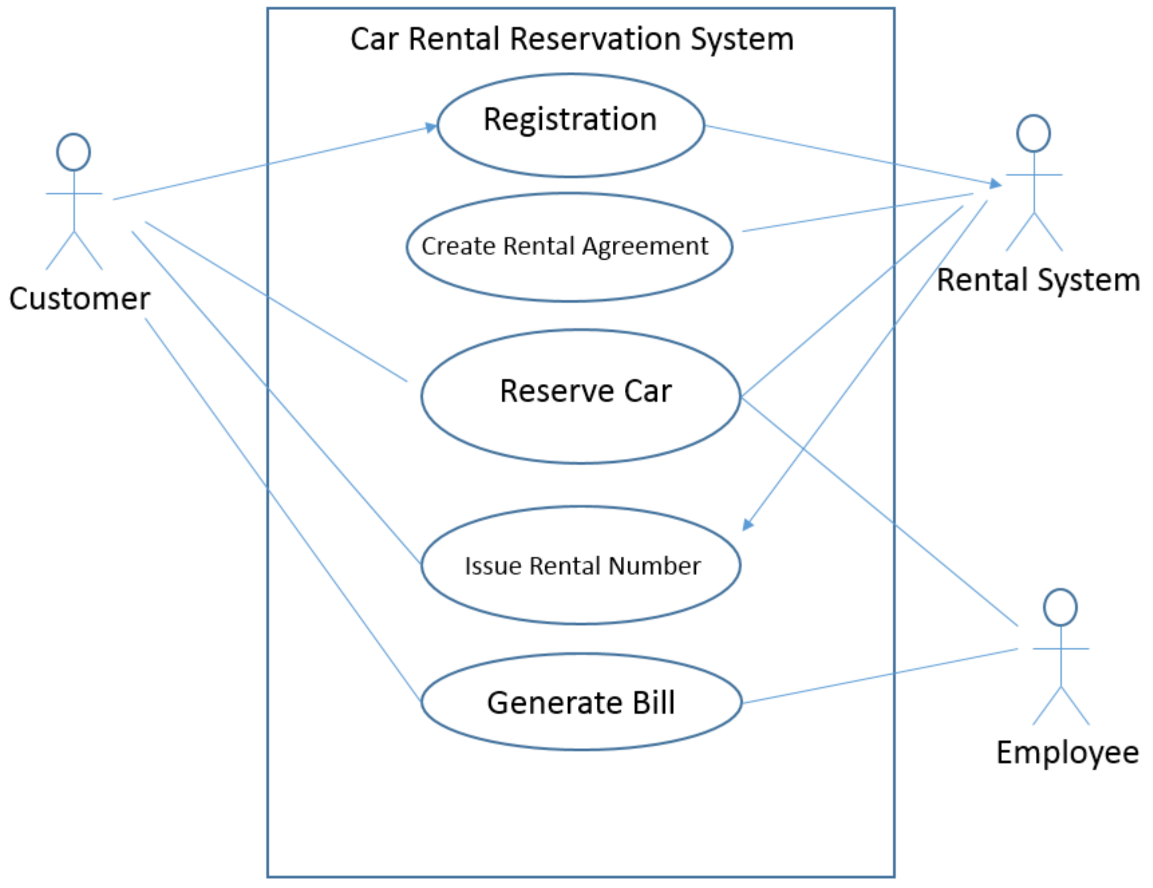
Eduardo Trejo
11/07/14

Use Case Model – Car Rental Example

This is a use case model that describes the behavior of a Car Rental Reservation System. The example is composed of three actors and five use cases.



Use Case – Registration

1. Basic Flow

{Input Information}

(1) The use case begins when the *Customer* starts registering on the Car Rental Reservation System.
(2) *Customer* creates a username and password.
(3) *Customer* inputs their personal information.

{Reads Information}

(4) The Rental System reads in the *Customer* information.
(5) The Rental System adds *Customer* to the database.
(6) The Rental System generates a *Customer* identification number.
(7) The Rental System displays a successful registration message.

{Use Case Ends}

(8) The use case ends.

2. Alternative Flows

2.1 Handle Existing Username

At {Input Information}, if the registering Customer username is already in the system

1. The Rental System prompts the *Customer* to enter another username.
2. Go back to {Input Information}, step 2.
3. System validates the *Customer* username.
4. The use case ends.

2.2 Handle Missing Customer Information

At {Reads Information}, if the Customer mistakenly forgot to fill in a required information field.

1. The Rental System alerts *Customer* that there is a missing field.
2. The Rental System prompts the *Customer* to enter required information.
3. Go back to {Input Information}, step 3.
4. System validates the *Customer* information.
5. The use case ends.

2.3 Handle Failure to Add Customer

At {Reads Information}, if the Rental System failed to add *Customer*

1. The Rental System alerts *Customer* fail to add message.
2. Go back to {Input Information}, step 5.
3. The Rental System auto fills previous entered information.
4. System was able to add *Customer*.
5. The use case ends.

Use Case – Reserve Car

1. Basic Flow

{Customer Login}

(1) The *Customer* logs on.

{Input Information}

(2) The *Customer* information is inputted into the reservation.

{Read Information}

(3) The Rental System reads in *Customer* information.

{Search for Car}

(4) *Customer* searches for available cars in the database.

(5) *Customer* selects car.

{System Updates}

(6) The Rental System shall allow Employee to update *Customer* rent records.

(7) The system shall allow Employee to display all *Customers* rent history.

(8) The Rental System updates the car inventory selection.

(9) The Rental System provides a successful committed reservation.

{Use Case Ends}

(10) The use case ends.

2. Alternative Flows.

2.1 Handle Failure to Read Information
At {Read Information}, if the System fails to read in the *Customer* information.

1. The System prompts a failure to read information message.
2. Go back to {Input Information}, step 2.
3. The System reads the information correctly.
4. Use case ends.

2.2 Handle Car Unavailability
At {Search for Car}, if the certain car requested is already reserved

1. The System alerts the *Customer* that specific car is already out for reserve.
2. The System prompts the *Customer* to select a different car.
3. Go back to {Search for Car}, step 4.
4. The System approves selected car.
5. The use case ends.

2.3 Handle Fail To Update Inventory
At {System Updates}, if the System failed to update the available inventory

1. The System prompts Employee to update inventory once again.
2. Employee makes changes to the inventory.
3. The System prints out a summary of the changes to the Employee.
4. The use case ends.