

Part A: Problem Domain Classes

Please see *Domain Model Class Diagram - Car Sharing IS Project.jpg* for the domain model class diagram.

Part B: Use Cases

Please see *Use Case Diagram - Car Sharing IS Project.jpg* for the overall use case diagram.

Fully developed use case:

Use Case Name:	<i>Member makes reservation</i>	
Scenario:	Member makes on-line reservation	
Triggering Event:	Member logs on to Car Sharing website and requests a vehicle for use	
Brief Description:	Member logs onto system and requests the reservation form. The member enters the date, time, duration and vehicle for their proposed trip. The system creates and stores the reservation and sends a confirmation to the member.	
Actors:	Member	
Related Use Cases:	Includes: <i>Member requests unavailable vehicle</i>	
Stakeholders:	Management	
Preconditions:	Member must be signed up for the Car Sharing project. Requested vehicle must exist.	
Postconditions:	A reservation must be created. The reservation must be associated with a member. The reservation must be associated with a vehicle.	
Flow of Activities:	Actor	System
	1 Member connects to the Car Sharing website and logs in.	1.1 Validate member account.
	2 Member goes to the new reservation page and enters date, time, duration and vehicle.	2.1 Create new reservation. 2.2 Display reservation information with date, time, duration and vehicle.

	3 Member confirms request. 4 Member confirms reservation.	3.1 Finalize reservation. 3.2 Display confirmation screen with reservation information. 4.1 Finalize reservation and send email. 4.2 Return customer to main page.
Exception Conditions:	2.1 If member requests unavailable vehicle, then invoke <i>Member requests unavailable vehicle</i> . 3.1 If member does not confirm request details, delete reservation and return member to new reservation page.	

Intermediate use cases:

Flow of activities for scenario of *Member makes phone reservation*

Main Flow:

- 1** Member calls the Car Sharing phone number and gets phone clerk.
- 2** Phone clerk verifies member information.
- 3** Phone clerk initiates creation of a new reservation.
- 4** Member requests a date, time, duration and vehicle for reservation.
- 5** Phone clerk submits reservation details; system creates new reservation and displays confirmation.
- 6** Phone clerk confirms reservation details with member.
- 7** Phone clerk confirms reservation in system.
- 8** System finalizes reservation and sends confirmation email.

Exception Conditions:

- 1.1** If member calls during business hours, the call is forwarded to the main office. Otherwise it is forwarded to the contracted call center.
- 5.1** If phone clerk enters information for unavailable vehicle, the system invokes *Member requests unavailable vehicle*.
- 6.1** If member does not confirm details, clerk cancels order and member can:
 - a.** begin a new reservation with different details
 - b.** end phone call without making reservation

Flow of activities for scenario of *Member requests unavailable vehicle*

Main Flow:

- 1** Member inputs date, time, duration and vehicle for reservation, but requested vehicle is unavailable at that time.

- 2 System displays list of alternate vehicles based on closest locations.
- 3 Member selects alternate vehicle for reservation from list.
- 4 System updates Reservation with alternate information selected by user.
- 5 System displays confirmation screen with reservation details.
- 6 Member confirms alternate reservation details.
- 7 System finalizes reservation and sends confirmation email.
- 8 System displays final reservation information.
- 9 Member reviews confirmation information.
- 10 System returns user to home screen.

Exception Conditions:

- 3.1 Member not satisfied with alternate options, chooses to cancel reservation.
- 3.2 System deletes reservation and returns member to initial reservation screen.
- 6.1 Member chooses not to confirm reservation, instead cancels reservation.
- 6.2 System deletes reservation and returns member to initial reservation screen.

Flow of activities for scenario of *Member changes or cancels reservation*

Main Flow:

- 1 Member logs on to system.
- 2 System confirms member identity.
- 3 Member navigates to previously booked reservation page for future reservations.
- 4 System displays list of member reservations for future.
- 5 Member selects reservation from list to edit.
- 6 System accesses and displays reservation, displays reservation details.
- 7 Member inputs new date, time, duration and vehicle information.
- 8 System updates reservation with new information and displays confirmation screen.
- 9 Member confirms new reservation.
- 10 System finalizes reservation and sends confirmation email.
- 11 System displays final reservation information.
- 12 Member reviews confirmation.
- 13 System returns member to home page.

Exception Conditions:

- 6.1 Member chooses to cancel reservation.
- 6.2 System displays confirmation screen to cancel reservation.
- 6.3 Member confirms cancellation.
- 6.4 System deletes reservation and returns user to reservation page.
- 7.1 Member inputs reservation information for unavailable vehicle.
- 7.1 System invokes *Member requests unavailable vehicle*.
- 9.1 Member chooses not to confirm reservation, cancels instead.

9.2 System deletes reservation and returns user to reservation page.

Brief use cases:

Potential member needs recruitment description

When the project is ready to expand, potential members stored in the system are sent recruitment packages to join the program if they are in the expansion area.

Member joins car sharing program description

When the member returns an application and submits the membership fee, they are added to the system as full members and provided with details to log on to the car sharing website. A SmartCard is sent to the member.

Member is terminated from program description

When member fails to uphold the terms of the car sharing program agreement, the manager accesses the system to remove them. The system will remove the member from full membership and store them as a terminated member, and the member will return their SmartCard and no longer be able to access the system.

Member wants to change preferences or info description

The member logs on to the system and accesses their membership information. They update their preferences or other information. The system updates the member information/preferences.

Member or Management wants to check account or other information description

The member or manager logs on to system and accesses the information they desire to see. The system displays the necessary information on screen.

Member tries to open vehicle door description

When the member arrives at the vehicle for their reservation time, they scan their SmartCard on the vehicle system. The vehicle system sends the member information to the system. The system confirms that the member has a valid reservation for the current date, time and vehicle and sends confirmation if valid. When the vehicle system receives confirmation, it unlocks the door for the member.

Member tries to get vehicle key description

When the member enters the vehicle, they access the key storage container by entering their PIN number. The vehicle system confirms the PIN number with the main system. Then the main system confirms the PIN, the key storage container is opened and the user retrieves the key.

Member starts car description

The member uses the vehicle key to start the car. The vehicle system creates a new trip and begins tracking the car.

Member drives vehicle description

As the member drives the vehicle, the vehicle system constantly tracks and updates the trip information.

Reservation time begins without member access description

If the requested reservation begins and the member has not accessed the vehicle, the vehicle system creates a new trip and begins tracking the vehicle for the duration of the reservation.

Member refuels vehicle description

When the fuel level of the vehicle is low, the member uses the fuel card located in the vehicle to make a payment at a service station. The system stores the transaction information.

Member returns vehicle description

When the member has finished their trip, they turn off the car and return the key to the storage container and lock it. The system finalizes the trip information and displays the results in the vehicle system.

Member locks car with SmartCard description

The user exits the vehicle and locks the door using their SmartCard. The vehicle system confirms the member identity with the main system and locks the door. The vehicle becomes ready for the next reservation.

Member files expense claim description

The member fills out an expense claim with the required information and sends it to management. Management confirms the details of the claim and invokes *Management adjusts member charges* if the claim is valid. Otherwise the user is notified of why their claim has been rejected.

Management adjusts member charges description

When management has determined that a member account needs to be adjusted, they access the member information and credit or debits the account with the necessary funds.

Debit/Credit payment made description

When a member decides to make a payment, the system stores the payment information and gets the funds from the required bank or credit company and updates the user account once funds are confirmed.

Time to produce monthly bill description

When the end of the month is reached, the system invokes *Time to send newsletter* and *Time to send seasonal info*. It prints and sends a bill to the user with a list of transactions for the month

Time to send newsletter description

When the end of the month is reached a newsletter is printed with the necessary information for the month and included with the monthly bill.

Time to send seasonal notification description

When the end of a season is reached, the system prints out a seasonal notification with the necessary information for the upcoming season and includes it with the monthly bill.

Time to produce membership report description

When the end of the month is reached, the system produces a report with details of all members in the system and emails it to the manager.

Time to produce preference trends report description

When the end of the month is reached, the system compiles the information from members, reservations, and trips into a preference report which is emailed to the manager.

Time to produce vehicle usage and availability report description

When the end of the month is reached, the system compiles information from vehicles, trips and reservations for the month into a usage and availability report which is emailed to the manager.

Time to produce reservation patterns report description

When the end of the month is reached, the system takes the reservation information for the month and puts it into the required report, which is then emailed to the manager.

Time to produce trip statistics report description

When the end of the month is reached, the system takes the trip information for the month and puts it in the required report, which is then emailed to the manager.

Management adds new service contractor description

When the manager makes a contract with a new service contractor, they enter the details into the system. The system creates a new service contractor and stores the details. The service contractor is provided with information to access the system to reserve vehicles for servicing.

Management removes service contractor description

When the manager decides to terminate a contract with a service contractor, they access the system to remove the service contractor. The system removes the contractor from the system and they are no longer allowed access.

Time to service vehicle description

When the time or distance requirements of the vehicle are met, the system sends a

notification to the service contractor to inform them they need to make a servicing reservation of the necessary vehicle.

Time to check tidiness of vehicle description

When the required amount of time since the last tidiness check has passed, the system sends the vehicle information to the manager and informs them of the need to reserve the vehicle for another tidiness check.

Management removes vehicle from service description

When a vehicle has met the end of its service the manager accesses the system and deletes the vehicle. The system removes the vehicle from the system and it is no longer able to be accessed for reservations.

Part C: Essay Question

Models and diagrams are an important part of systems analysis and design. They are usually created to help all those involved with the project understand the project better. Everything from the project scope, to fine detail can be captured using models and diagrams. They can also be easily stored and retrieved, allowing easy access to required information for the project. Without models and diagrams, we would be limited to long searches through text to find what is needed. Additionally, the visualization of certain aspects of the project allow for easier understanding and interpretation, rather than dry written descriptions.

One of the most important of these is the domain model class diagram. This diagram shows the relationships between different types of data over the entire system. This diagram is especially useful in larger projects, so that a complete overview of the system is easily seen. On very small projects, it may not be necessary to spend the time modeling the system. However, this diagram is only used in the object-oriented approach.

A use case diagram is also a critical part of most projects. The use case shows how actors actually use the system. For simpler projects, all of the use cases can be modeled in a single diagram, but for more complex projects, a set of use case diagrams can be constructed. Either way, the use case diagram provides an easily understood method of identifying use cases.

Another way of showing these interactions is with a system sequence diagram (SSD). However, the SSD focuses on the messages passed between the system and the user during the course of a user interaction with the system. These messages form the basis for actual system programming, so the importance of this diagram should not be overlooked.

Part D: Reflection

I have included my reflection upon this assignment here, and I will be posting the same text on my blog shortly after I submit the assignment.

This assignment has been much easier for me to get a handle on compared to the first assignment. The previous assignment had me wading into an unknown ocean. I wasn't sure if I was on the right track at all, and the scope of the required answers seemed very daunting. However, after receiving good results on the first assignment, my confidence in my own understanding of the material has been bolstered. This has allowed me to tackle this assignment with more assurance. Additionally, the scope of this assignment is large, but most of it has already been worked out from the previous assignment, so this familiarity also made answering the questions easier. More specifically, it was easier to create most of the descriptions and the use case diagrams because these are based off the event table in the previous assignment.

However, this assignment was not without its own difficulties. The sheer number of descriptions needed was very large, because there were so many events in the event table. Thank goodness most of them were only brief descriptions. Yet the most difficult part for me was keeping track of the domain model class diagram. I went through all the use cases and made the diagram with the corresponding entities and relationships. However, as I began writing the use case descriptions, I discovered that the relationships were not quite the way I had modeled them or that other entities and attributes needed to be added. As I added and shifted things around, the model became more confusing, and I would often switch back and forth between ideas on how to represent what I wanted.

For those attempting this assignment for the first time, I would warn them that the domain model class diagram you make on your first attempt should be a quick preliminary run, which you confirm and or update as you work through each use case description in detail. You should only attempt to understand the diagram fully when you have completed the entire assignment, so you can be sure that your diagram is as complete and accurate as possible.

With the use case descriptions themselves, I found a further difficulty. I wanted to make sure that the intermediate and fully developed cases were complete, and covered all aspects of the use case. I found that I did not have as much familiarity with the material as I had assumed at times, and I would go through the use cases several times to make sure I had what I thought to be a complete description. I wasn't sure when to include a two small processes in a single step, or to break out each one into its own step. In the end, I just did what made the most sense, or seemed the most elegant to me. I would recommend that others do the same as they work through the assignment so that they can easily understand what they have done as they review their work. What makes sense to you will probably be the same for most others, if you put care and effort into the process.

Overall, I am pleased with how the assignment turned out. In general, I would advise others to carefully review all that they have done throughout the assignment as soon as

they are finished. This is because the diagrams and the descriptions are based on the same event table, and each one can inform the others as you work through them. If you describe something in a use case that is not modeled in the diagram, then you should update it. If you list an attribute in your diagram that is not mentioned in any of your use cases, then you should look into the discrepancy. In the end, this will allow for the most complete description of the system.

Again, good luck to everyone on Assignment 2!