

CarMath Case Study:

CarMath is a franchising company that is being set up to promote car sharing. In many cities, traffic congestion poses a threat to the quality of life as well as causing considerable pollution. This includes the release of carbon dioxide into the atmosphere. Many countries are trying to meet their obligations under international agreements to reduce carbon emissions in an attempt to prevent the worst effects of global warming. CarMath is a response to this situation. In many areas, public transport has declined as car ownership has increased, and the public transport infrastructure is not available to take up the demand from people not using their cars to travel to work. Car sharing schemes offer one short-term way of reducing traffic without the immediate investment in public transport infrastructure that is required in the medium to long term.

CarMath seeks to promote car sharing and to provide a service to potential car sharers by matching up people who both live and work near one another. While many people who work together share transport informally, it is more difficult for people who work near one another to find a suitable person to share transport with, and in some very large organizations, even people who work on the same site may not know one another.

CarMath consists of three layers of structure: the global operation, which is a not-for-profit trust, the central operating company in each country and local franchise operations. Depending on the country in which it is operating, CarMath's central operation will offer its services to local government and large corporations, which have legal obligations to reduce traffic in some countries or states. It will also publicize its services to the general public. People who join up will pay a small membership fee, which will be refunded if the local CarMath franchise is unable to match them up with one or more other people who require or are offering transport. The CarMath franchise will draw up model agreements between the participants, to prevent the money that changes hands to cover fuel costs being treated as taxable income, and advise on the insurance implications of car sharing. It will act as an agent for companies that sell insurance policies that specifically cover car sharing. Research has shown that car sharers are a good insurance risk.

Staff in the local franchises will undergo a comprehensive training course, which covers the consultancy that they must be able to offer to companies and local government, the legal situation in their own country or state, insurance requirements, safety considerations and how to operate CarMath's systems. In some countries, regulation of the insurance industry means that franchise staff must also meet the requirements of regulatory bodies.

CarMath expects to make its money from a combination of membership fees, consultancy income and the commission on insurance sales. A percentage of all income will be taken by the central operation, and the rest kept by the franchise. As road-pricing schemes based on radio communication between vehicles and road-side transponders become more widespread, CarMath franchises will sell and install the necessary equipment and work with toll authorities and road-pricing schemes to negotiate discounts for members on the basis that they are reducing traffic demand.

Use Case Creation/ Analysis:

The first and most essential step towards software design or development process is to analyze the scenario. In our case we will first do the analysis of the given case study. So, to make the Use Case we have to follow the steps given below.

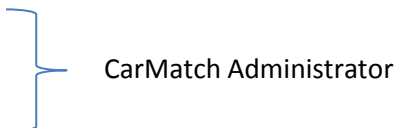
- Identify actors
- Identify goals
- Identify use cases
- Select system boundary

1. Identify/find actors:

As we know actor is any entity that interact with the system or play any role in the system and the system should respond to actor. Remember that the actor can be a person, system or an organization. In the CarMath case study the roles interacting with the system are:

- Clerk
- Receptionist
- Supervisor
- Franchise
- Web server
- Accountant
- Insurance Company System
- Insurance Administrator
- Insurance Clerk
- Credit Card Company
- Credit Card System
- Automated Bank Transfer System

Actors are named according to their goals or functionality in the system. To enter the information about car sharer, the clerk, receptionist or supervisor can also enter the data of the person. As they all are doing the same task, so to make the separate actor for every role is not a good approach. Thus, we name three of them as “*CarMatch Administrator*”.

- Clerk
 - Receptionist
 - Supervisor
- 
- CarMatch Administrator

However, some roles are not always clear from the scenario or case study, you have to do a little brainstorming. For example, a person has to pay membership fee for taking the carsharer facility. He can pay through credit card or through bank transfer. So, credit card company, credit card system and automated bank transfer system can be the actors.

2. Identify goals for actors:

Goals are basically the functional requirements of the system. Goals are the functions performed by the roles/actors of the system. These roles can be a person, organization, external system, clients. An actor can perform more than one activity. Moreover, actors are named according to their activities which they are performing to make UC more understandable.

Actor	Goals
Car Match Administrator	Register car sharer, Manually add car sharer, Match car sharer, Record sharing agreement
Franchise	Register car sharer, Match car sharer,
Web Server	Register car sharer
Accountant	Process payments
Credit Card Company	Process payment

3. Identify use cases and their relationship:

As we studied earlier that use cases are the actions or functions within the system. Generally use cases are defined for each user goal. They are named on the basis of function which describes the behavior of the system being implemented. Use case names are text strings and it is a good approach to keep the name short. In this step of UC analysis the relationship between use case are also identified. In the case study of CarMath following goals i.e. use cases are identified.

Use case 1	Relationship	Use case 2
Register Car Sharer	Generalization	Manually Add car Sharer, Transfer Car sharer from web server, Add car sharer web service
Manually Add Car Sharer	Include	Process Payment
Process Payment	Extend	Process Card Payment, Process Direct Debt
Match Car Sharer		
Record Sharing Agreement		
Produce Performance Report		

Here we have find some functional requirement of the system, these are the requirements that specify all the fundamental actions of the system. In the Software requirement specification document functional requirements are written as FR1 to so on.

FR1: Register a carsharer manually and add car sharer web services.

FR2: If car sharer is registered through web services, then transfer car sharer from the web server.

FR3: Match the car sharer on geographical basis.

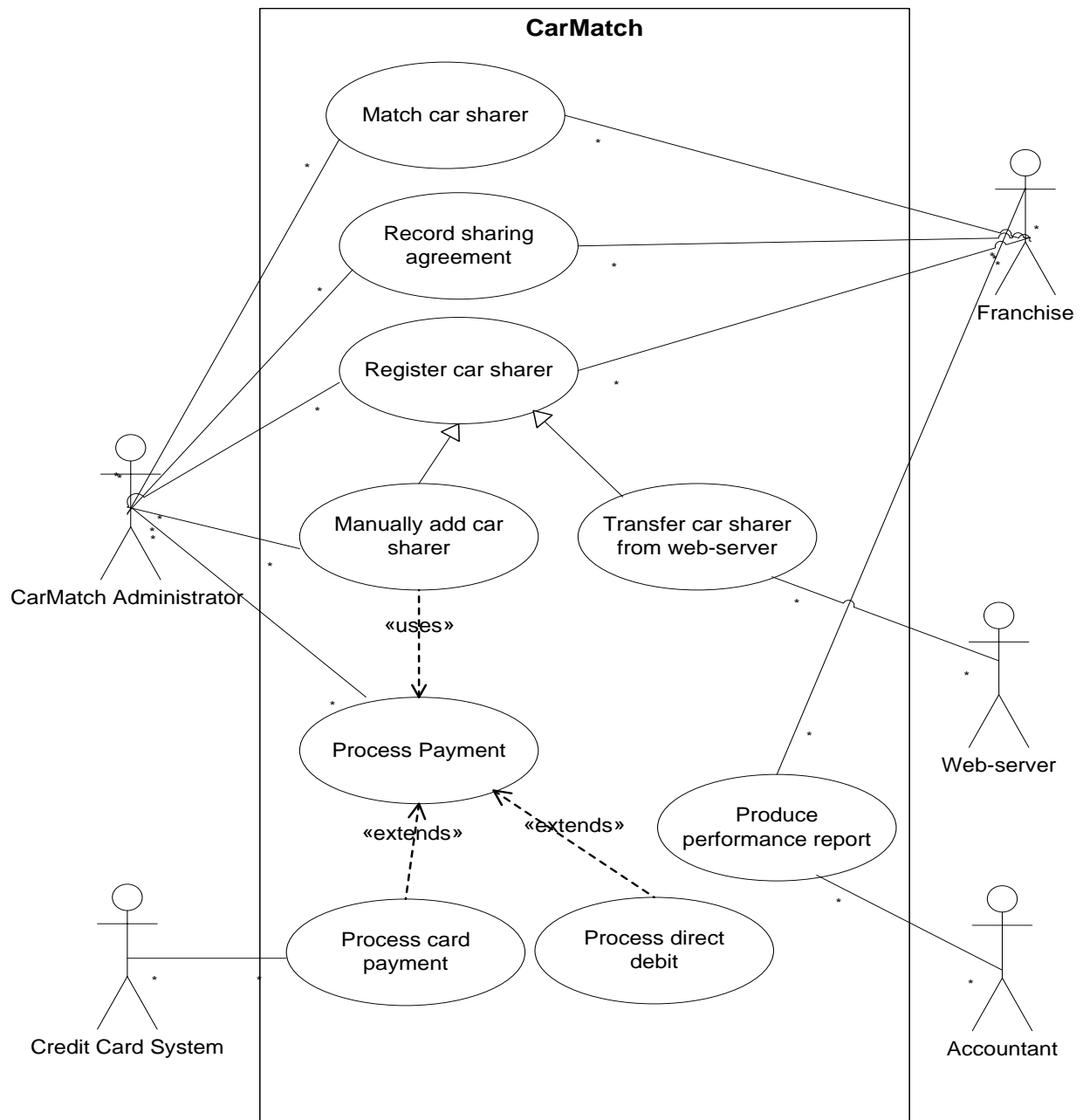
FR4: Pay membership fee for getting car sharer facility if user is added manually

FR5: If the user gets registered through web server then payment will be processed later.

4. Select system boundary:

As described in the earlier document system boundary is used to visualize the large system. It is basically a boundary between actors and the system. After identifying actors and their respective goals, place the system boundary in UC.

Use Case Diagram:



Usage Scenario:

A usage scenario describes that how one or more people or the organization interacts with the system. It defines the action, steps and events occur during interaction with the system, but doesn't indicate that how they will be performed. It may refer to a detailed path by a series of different flows, like use case or a test case. Usage scenario can be detailed or high level, which refers to critical actions only.

Use Case Title	Process Payment	
Use Case Id	CM-PP-06 *	
Requirement Id	FR4**	
Description: This use case is for processing of payment, when a new user gets the facility of car sharing then he/she has to pay some registration charges.		
Pre Conditions:		
<ol style="list-style-type: none"> 1. Administrator must have entered all information about new user in the system. 2. Car sharer must be matched. 		
Task Sequence	Exceptions	
1. Administrator adds a new car sharer account.	System may show authorization exception	
2. System asks for necessary information.		
3. Administrator provides all the required information and selects to complete the operation.	<ul style="list-style-type: none"> • System may show record duplication error. • Systems may show mandatory fields not filled exception. • If no record is entered , system may show NULL exception. 	
4. There is a problem in the data provided; some data needs to be corrected. <ul style="list-style-type: none"> – Administrator checks the available information and corrects the error. – Administrator continues from the step 3. 		
5. System after confirmation register the new car sharer.	System may show user authentication exception	

6. System ask for processing of payment (membership fee), if franchise successfully matched the new car sharer to available transport.	
Post Conditions: <ul style="list-style-type: none">- Sharing agreement must be recorded.- Payment of membership fee is processed.	
Unresolved issues:	
Authority: Administrator	
Modification history: 1.0	
Author: <Project or Group ID>	

- * CM-PP-06: CarMatch- Payment Process– Usecase # 01
- **FR4 : Functional requirement 4