Analysis & Design Project 2014



Robert Gabriel, Rokas Lukosevicius, Piotr Kawalec

Fall 2014

Table of Contents

Contents

[Analysis & Design Project 2014 1](#_Toc387144619)

[Table of Contents 2](#_Toc387144620)

[Student Declaration 4](#_Toc387144621)

[Glossary of Terms 5](#_Toc387144622)

[Assumptions 6](#_Toc387144623)

[Part A 7](#_Toc387144624)

[1 .Draw up a list of all events that the car rental company encounters in making reservations, renting, accepting returned cars, inspecting, billing and receiving payment. 8](#_Toc387144625)

[2. Create an event table using list of events identified in (i) 9](#_Toc387144626)

[3. Create a Use Case Diagram to model functional requirements 10](#_Toc387144627)

[4. Create a use case specification or an activity diagram for each use case. 11](#_Toc387144628)

[5. Create a System Sequence Diagram for each use Case. 21](#_Toc387144647)

[6. Create a Domain Model. 26](#_Toc387144657)

[7. Create a sequence or a Communication diagram to realize the happy scenario for each use case. 27](#_Toc387144658)

[8. Create a Design Class Diagram to support interaction diagrams developed in (vii) 32](#_Toc387144659)

[9. Extend Class Diagram to allow for the following additional requirements. 33](#_Toc387144660)

[10. Develop a state chart Diagram (state transition diagram) for the different possible states of a vehicle based on information provided above. 34](#_Toc387144661)

[11. Explain how the class diagram would be represented in a relational database 35](#_Toc387144662)

[Part B 36](#_Toc387144663)

[Q1 Explain the meaning of the MVC pattern using the following sample code. 37](#_Toc387144664)

[Q2 Explain the meaning of the Observer Design Pattern using the following code 38](#_Toc387144665)

[Q3 Explain the meaning of the facade design pattern using the following code 39](#_Toc387144666)

Student Declaration

I can confirm the following details:

Names/Student ID

1. Piotr Kawalec - R00103800

2. Rokas Lukosevicius.

3. Robert Gabriel -   R00102430.

Module Name: OO Analysis and Design

Module Lecturer: Mary Davin

I confirm that this is my own work.

Due Date: 3/5/2014

Signed . .

Signed . .

Signed . .

Glossary of Terms

* Branch - a location, other than the main office, where business is conducted.
* Space Number - space in a car park
* Additional item - extra items
* Rental agreement - a lease is a contractual arrangement calling for the lessee to pay the lessor for use of an asset.
* Rate - a fixed price paid or charged for something.
* Customer - a person who buys goods or services from a shop or business.
* Vehicle Status - is it out for repair or is currently outed.
* Catalogue - a complete list of items, typically one in alphabetical or other systematic order, in particular:
* Payment - the action or process of paying someone or something or of being paid.
* Customer - a person who buys goods or services from a shop or business
* Employee - a person employed for wages or salary, especially at non-executive level.
* Manager – a person responsible for controlling or administering an organization or group of staff.
* Rental agreement - a lease is a contractual arrangement calling for the lessee to pay the lessor for use of an asset.
* Reservation - the action of reserving something.
* Database - a structured set of data held in a computer, especially one that is accessible in various ways.
* Bill - the action of receiving something or the fact of its being received.

Assumptions

* Create an event table using list of events identified in (i)
* Manager (assumption: reservation can be done by phone)
* Get a quote for vehicle (assumption: service is available online)
* Create a Use Case Diagram to model functional requirements

1. Sent a vehicle for maintenance (assumption: a vehicle that needs to be repaired must be marked in the system as not operational for the time specified
2. Send vehicle to a lot (assumption: this use case is a result of the vehicle being in rental condition and it is a default, extended use case for returned vehicle which passes inspection

**Other assumptions are stated during the document.**

Part A

# 1 .Draw up a list of all events that the car rental company encounters in making reservations, renting, accepting returned cars, inspecting, billing and receiving payment.

Presented below list describes business events in Davin’s Rent a

Car Company. It is high level view of typical events that trigger use of the system.

1.      New vehicle is catalogued into the system

2.     Vehicle is removed from the system (damaged or too old)

3.     Customer wants to get a quotation for a vehicle

4.     Vehicle is being reserved

5.     Vehicle is being rented

6.    Vehicle is being returned

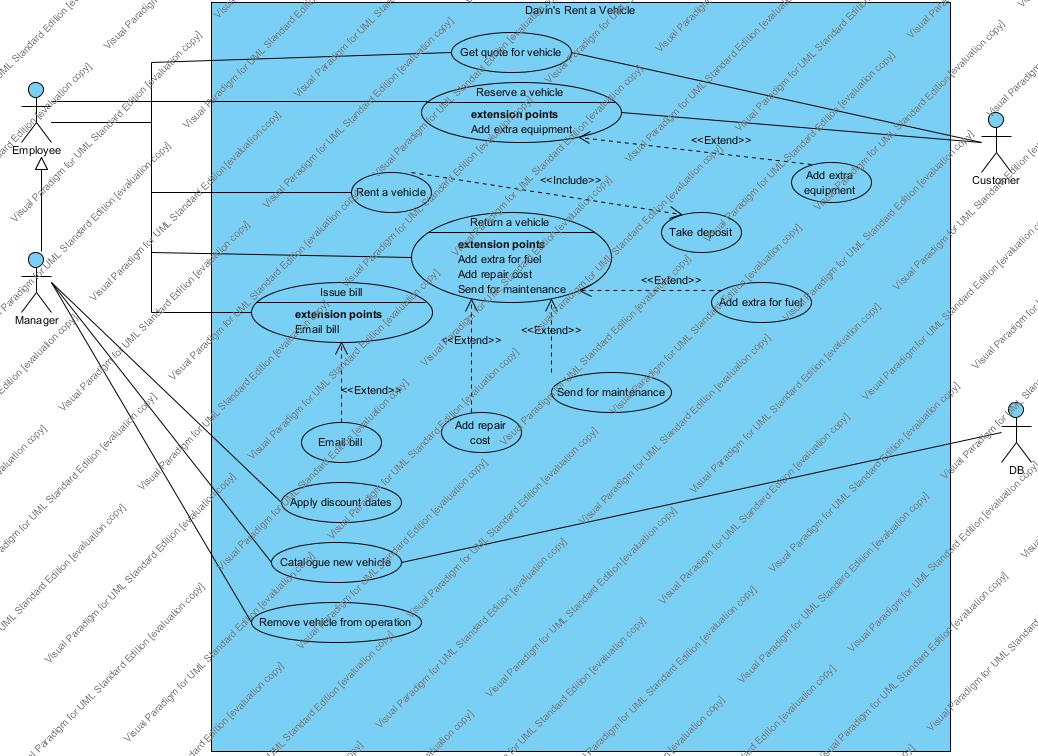
7.     Payment is issued

8.     Discount is applied

# 2. Create an event table using list of events identified in (i)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event No | Event Dec | System input | Actor proving input | System output | Actor output |
| 1 | Catalogue new Vehicle | Input Vehicle Details | Manager | Details sent to database | Database |
| 3 | Remove vehicle from operation | Vehicle registration number | Manager | Vehicle details removed from database | Manager, Database |
| 4 | Get a quote for vehicle | Vehicle details, dates required | Customer  Employee | Vehicle schedule | Customer, Employee |
| 5 | Reserve a vehicle | Vehicle type, time, dates. | Customer, Employee, Manager | Vehicle Reservation | Customer |
| 6 | Rent a vehicle | Details of:  Drivers,  Customer,  Rental agreement,  Third-party. | Employee, Manager. | Billing updated.  Rental agreement | Customer, Database |
| 7 | Return a vehicle | Date, time, mileage returned, Amount for fuel | Employee, Manager | Damages and maintenance recorded.  Bill is issued | Database |
| 8 | Payment | Car License Regulation.  Credit Card number. | Employee, Manager | Receipt given  Billing Updated | Customer, Database |
| 9 | Issue the  Bill | Bill | Employee, Manager | Bill /Mail for Customer | Customer |
| 10 | Apply discount Dates | Start date, finish date | Manager | Discount rate and time span | Database |

# Create a Use Case Diagram to model functional requirements



# Create a use case specification or an activity diagram for each use case.

**Use case:** Catalogue new vehicle

**Actors:** Manager

**Purpose:** To enter details of new vehicle into the system

**Overview :** Company after acquiring new vehicle needs to enter its details into the system in order to make it available for rent. System prompts for details of new vehicle, than data is recorded.

**Type:** Essential

**Preconditions:** New vehicle must be purchased (its details are available)

**Post conditions:** Vehicle is added to specified branch and its details are in the system

**Special Requirements:** Vehicle must be under certain mileage (assumption).

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. New vehicle is purchased and needs to be recorded in the system. | 2. System prompts for category of the vehicle, make, model, registration number, color, mileage, date of acquisition, image and branch it is assigned to. |
| 3. Manager enters details. | 4. System confirms new vehicle record. |

Alternative Flow of Events

Line 4. (assumption: System may validate some information entered e.g. correctness of registration number . Return to line 3)

**Use case:** Remove vehicle

**Actors:** Manager

**Purpose:** To remove vehicle from the system

**Overview:** Vehicle that is damaged or too old and will not be in operation needs to be removed (its details) from the system. User provides vehicle registration number (assumption: reg No is how vehicles are identified), system confirms vehicle removal.

**Type:** Essential

**Preconditions:** Vehicle (its details) must be in the system already

**Post conditions:** Vehicle is removed (its details) from the system, vehicle is no longer available for rent.

**Special Requirements:** None

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. When vehicle is no longer suitable for rental, system needs to be informed. | 2. System prompts for vehicles registration number (assumption: vehicles are identified by registration number) |
| 3. Manager enters registration number | 4. System displays details of the vehicle |
| 5. Manager confirms vehicle removal | 6. Vehicle is removed from rental availability and confirmation is displayed |

Alternative Flow of Events

Line 4. Wrong registration number entered. Display message and return to line 3

**Use case:** Get a quote

**Actors:** Customer

**Purpose:** To get details of a vehicle’s availability

**Overview :** Customer wants to find out what time range and vehicle’s category are available to rent. Customer provides category within specified and the time range. System returns vehicle schedule.

**Type:** Normal

**Preconditions:** Service is available online, customer has an access to internet.

**Post conditions:** Customer is presented with vehicle schedule.

**Special Requirements:** None

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Customer wants to quote for vehicle availability | 2. System prompts for category of the vehicle and time range. |
| 3. Customer enters details | 4. System displays available vehicles |

Alternative Flow of Events

Line 4. No vehicles available. Display nearest available time range, return to line 3.

**Use case:** Reserve a vehicle

**Actors:** Customer, employee

**Purpose:** To get reserve a vehicle

**Overview :** Customer wants to reserve a vehicle in advance before renting. He must provide  first and last name, and credit card details (assumption: credit card details must be verified before reservation is confirmed). Customer may reserve additional items.

**Type:** Normal

**Preconditions:** Service is available online,  customer has credit card, customer has an access to internet (assumption: customer can call company and employee reserves a vehicle for a customer)

**Postconditions:** Vehicle is reserved for a specific customer and removed from being available for rent.

**Special Requirements:** (assumption) Time restrictions may be applied.

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Customer wants to reserve a vehicle | 2. System asks for category of vehicle and dates the customer wants to reserve the vehicle for. |
| 3. Customer or employee enters the details being asked for. | 4. System displays vehicles available with that category. |
| 5. Customer or employee choose a vehicle he wants | 6. System then asks for Customer first name, last name and credit card details |
| 7. Customer or employee enters in the details. | 8. System validates information. Verifies the credit card details. |
| 9. Customer or employee can confirm or cancel | 10. If confirm pressed, reservation details are displayed. |

Sub flow of Events

|  |  |
| --- | --- |
| 6a. Credit card  is authorized by external authority | 6b. Credit card verification message is returned |
| 8a. Customer decides to add extra equipment | 8b. System records additional information |

Alternative Flow of Events

Line 4. No vehicle available. Display nearest available date for vehicle category specified.  Go back to line 3.

Line 8. Credit card is not accepted. Display message, go back to line 7.

**Use case:** Rent a vehicle

**Actors:** Employee

**Purpose:** To allow customer to hire a vehicle.

**Overview :** Customer wants to rent a vehicle. All needed details must be collected. Extra information might be provided, rental agreement must be created.

**Type:** Essential

**Preconditions:** Customer must be able to provide all necessary details, vehicle is available for rent

**Post conditions:** Rental agreement is made.

Special Requirements: (assumption: Customer must be of certain age.)

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Use case starts when employee collects all the details for rental agreement. | 2. System prompts for details of each driver. |
| 3. Employee provides: full name, birth date, driving license number, country of issue for each driver. | 4. System  records driver’s details and prompts for any extra information (collision damage waiver, liability insurance, personal  accident insurance) |
| 5. Employee enters additional details. | 6. Additional information is recorded. System prompts for details of rental agreement. |
| 7. Employee enters details of: customer’s name, type of rate, vehicle identification, category price and date, time, mileage when leave and expected date, time when returned. | 8. System records information and confirms vehicle rental. |

Sub flow of Events

|  |  |
| --- | --- |
|  | 4a. If none of the drivers is a customer then system prompts for third party details |
| 4b. Employee provides all necessary information | 4c. System records third party details. |
|  | 6a If no damage waiver specified prompt for deposit amount. |
| 6c. Employee enters amount of deposit paid. | 6d. System records amount. |

Alternative Flow of Events

**Use case:** Return a vehicle

**Actors:** Employee

**Purpose:** To return a vehicle.

**Overview :** Customer wants to return a vehicle. The date time and mileage are recorded. Fuel level is checked, vehicle is inspected for damages and maintenance.

**Type:** Essential

**Preconditions:** A vehicle must be on rent agreement

**Post conditions:** Payment is issued. Vehicle is available for rent or sent for maintenance.

**Special Requirements:**

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1 Use case starts after when Customer returns a vehicle and Employee does the inspection. | 2 System displays return form. |
| 3 Employee enters details of the vehicle at return time(date, time, mileage) and fuel amount | 4 System records details entered, prompts for damage/maintenance information. ( If fuel amount is specified it is added to the bill) |
| 5 Employee enters damage/maintenance details | 6 System records details provided and display a bill. |
| 7 Employee confirms bill details |  |

Sub flow of Events

|  |  |
| --- | --- |
|  | 4a If fuel amount is specified it is added to the bill |
|  | 6a If damage waiver is not specified system prompts for  cost of repair |
| 6b Employee enters cost of repair | 6c System updates a bill and marks the Vehicle to be sent for maintenance (assumption) |

Alternative Flow of Events

None

**Use case:** Issue a bill

**Actors:** Employee

**Purpose:** To provide customer with a bill.

**Overview :** After returning a vehicle, charge for rental agreement is being issued. It may also consist of additional elements depending from results of inspection (fuel amount, cost of repair,).

**Type:** Essential

**Preconditions:** Vehicle is returned and inspection is carried out.

**Post conditions:** Customer is issued with the bill.

**Special Requirements:** None

Flow of Events

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1 Use case starts after when Employee checks in vehicle and collects all details | 2 System prompts for customer details |
| 3 Employee  provides customer’s  name | 4 System displays  customer details |
| 6 Employee accepts the bill | 5 System issues a bill |

Alternative Flow of Events

Line 4. If details not found, error message is displayed. Go back to line 2.

Line 6. If customer is not present, bill is sent to via email.

**Use case:** Apply discount dates

**Actors:** Manager

**Purpose:** To provide offer to customers.

**Overview :** Manager wants to apply percentage discount for specific dates in the year. He enters details of the dates and the rate of discount.

**Type:** Normal

**Preconditions:** Rate for rent must be present.

**Post conditions:** New rate for vehicle rent is available.

**Special Requirements:**

Flow of Events

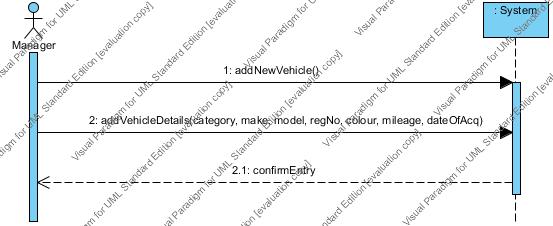
|  |  |
| --- | --- |
| Actor Action | System Response |
| 1 : This use case applies when the Manager wants to apply a discount  2: The manager inputs the dates, the rate of discount. | 3: System receives inputs and applies the discount to the rates. |
|  |  |

Alternative Flow of Events

None

# 5. Create a System Sequence Diagram for each use Case.

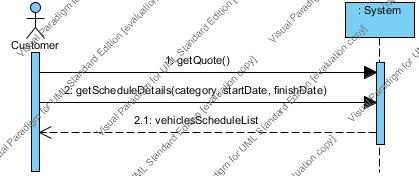
Add New Vehicle



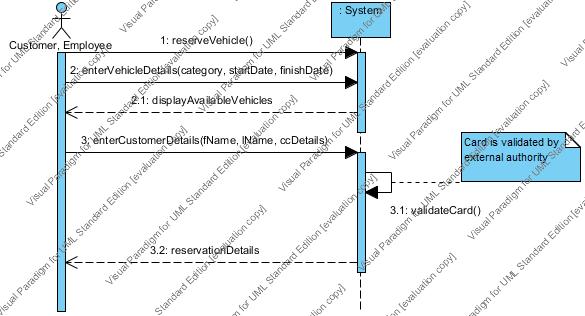
Remove Vehicle



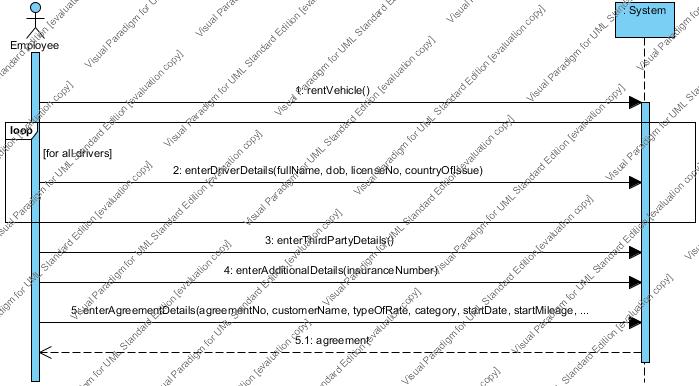
Get a quote



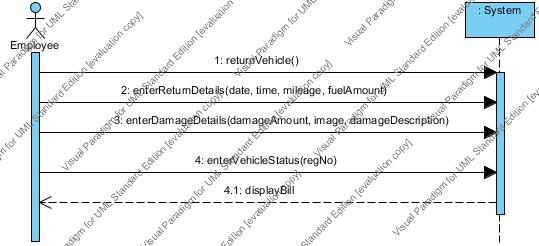
Reserve a Vehicle



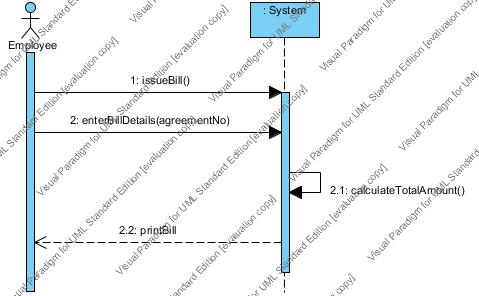
Rent a vehicle



Return a vehicle



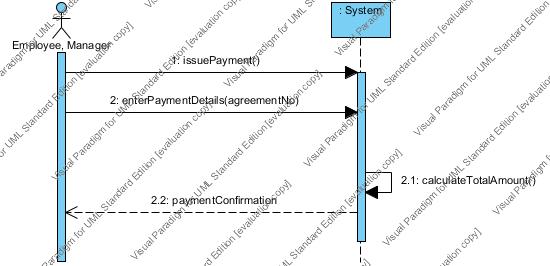
Issue a bill



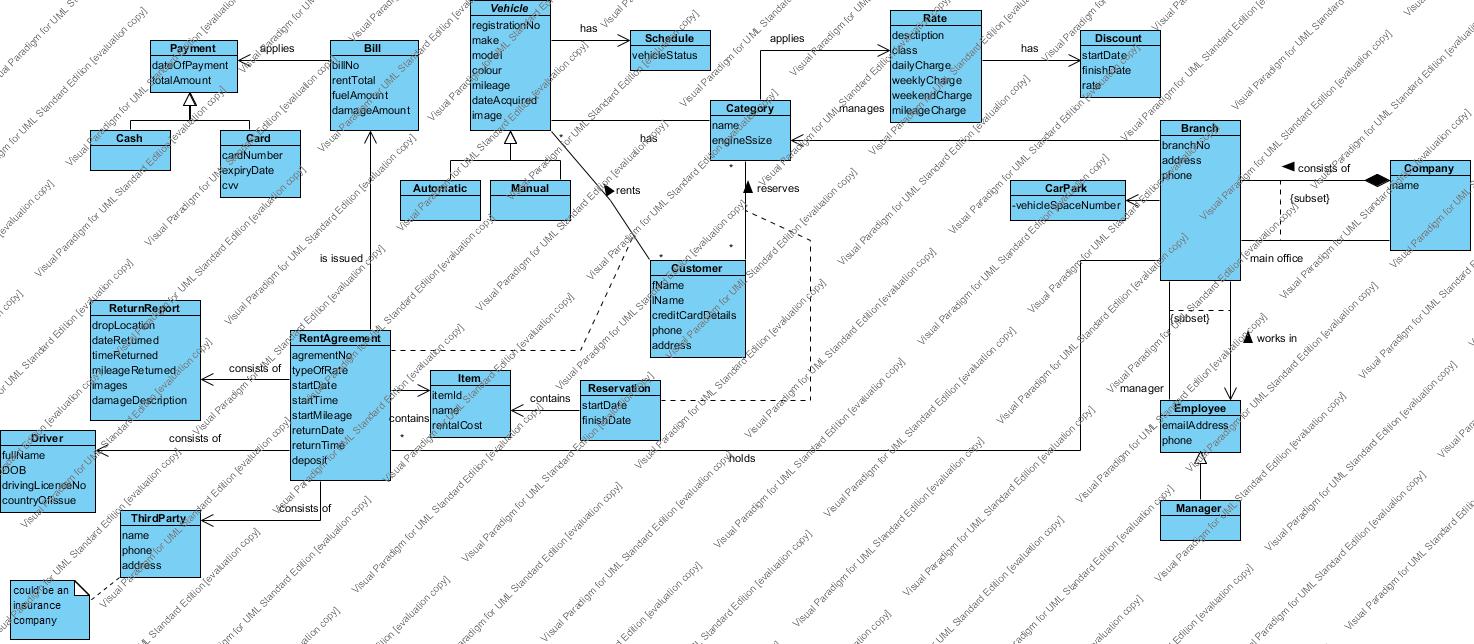
Apply a discount



Issue Payment

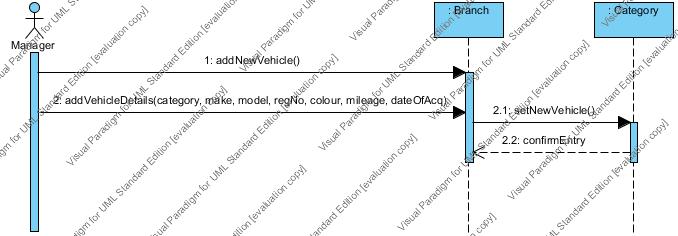


# 6. Create a Domain Model.

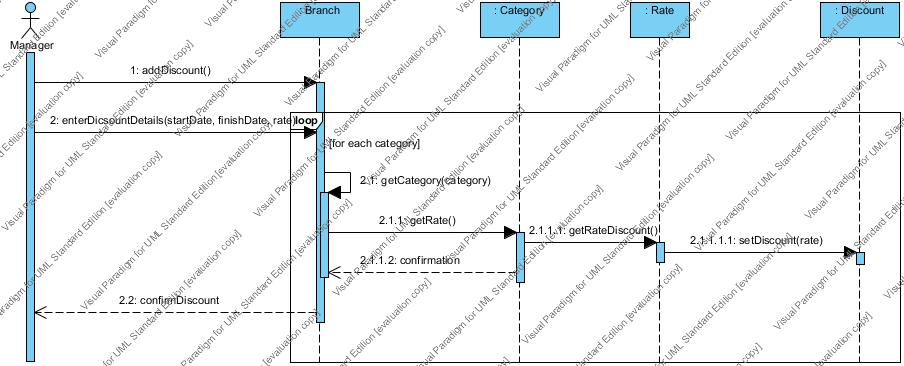


# 7. Create a sequence or a Communication diagram to realize the happy scenario for each use case.

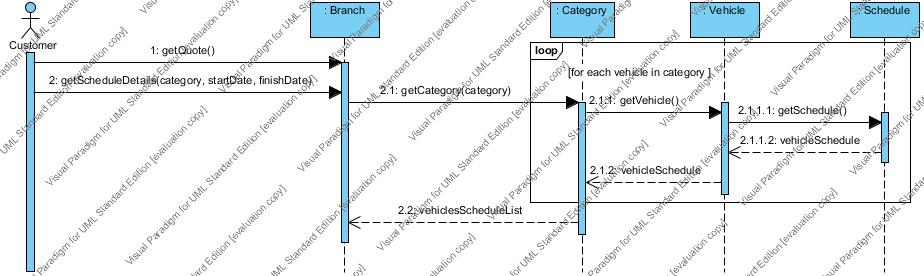
Add New Vehicle - Happy scenario



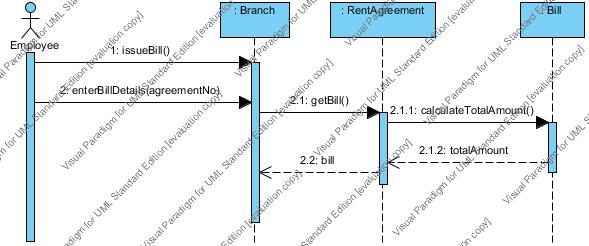
Apply discount - Happy scenario



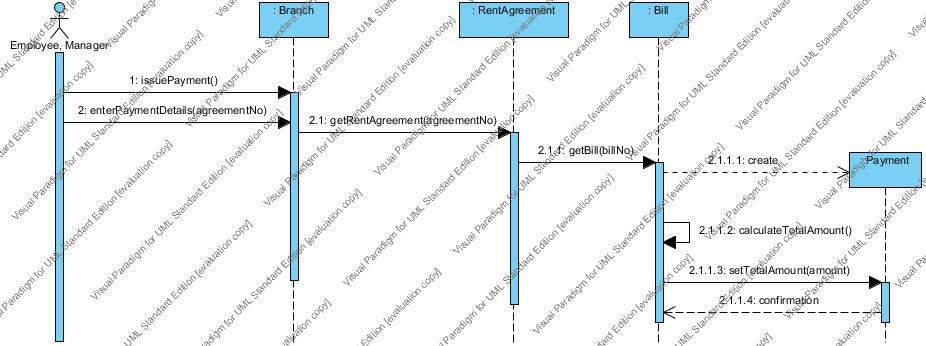
Get a Quote - Happy Scenario



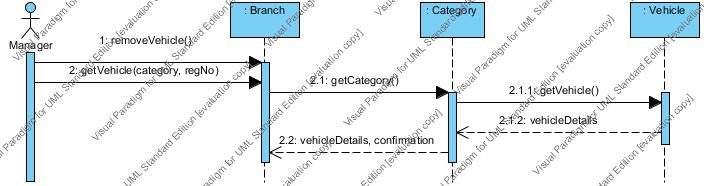
Issue a bill - Happy Scenario



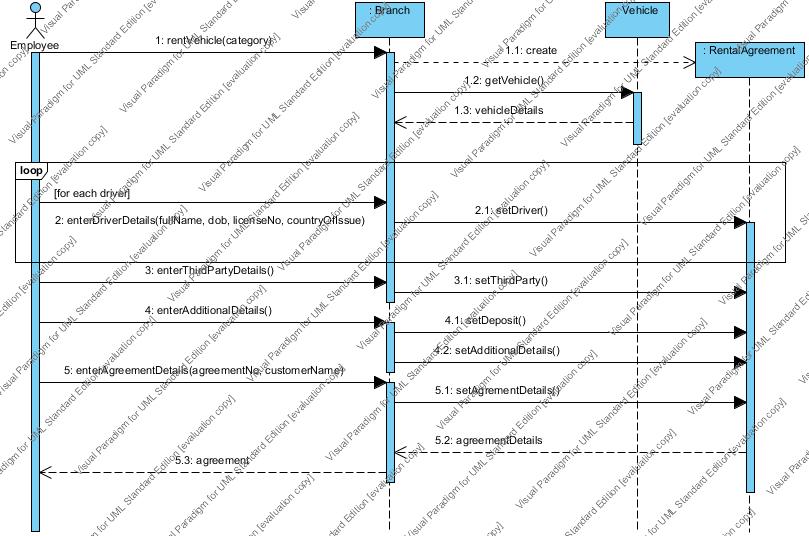
Payment - Happy Scenario



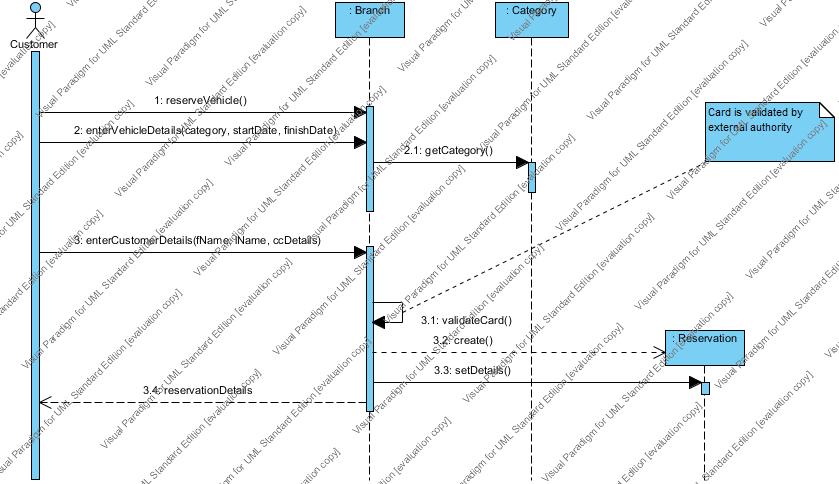
Remove a vehicle - Happy Scenario



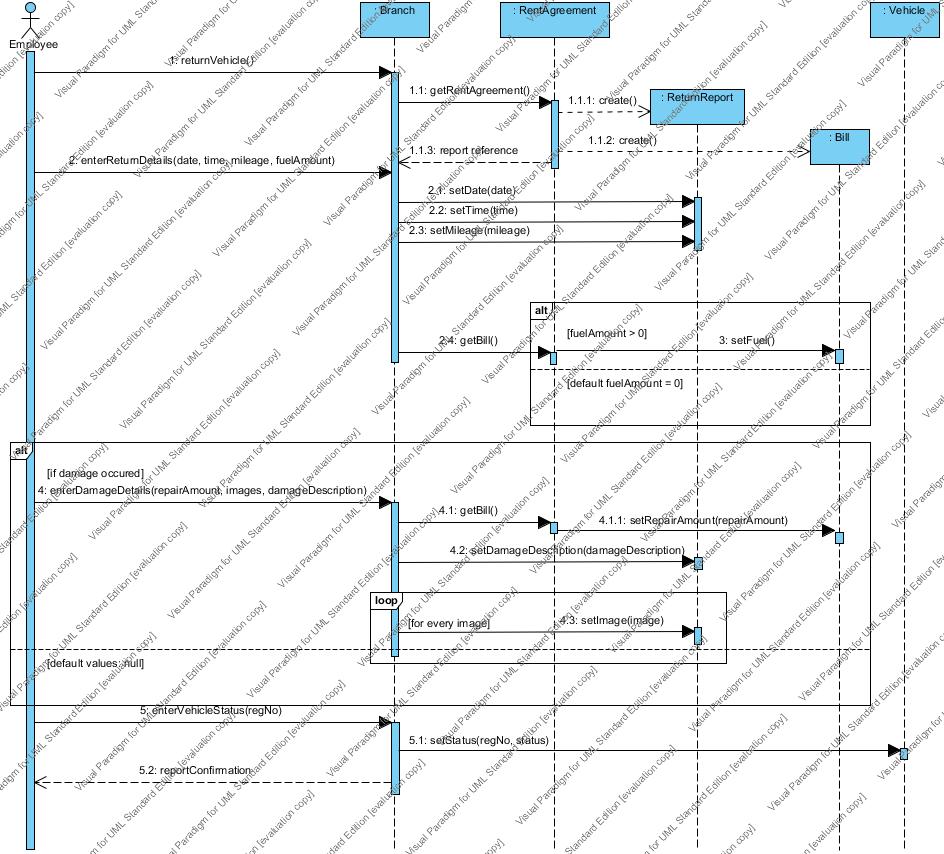
Rent a Vehicle - Happy Scenario



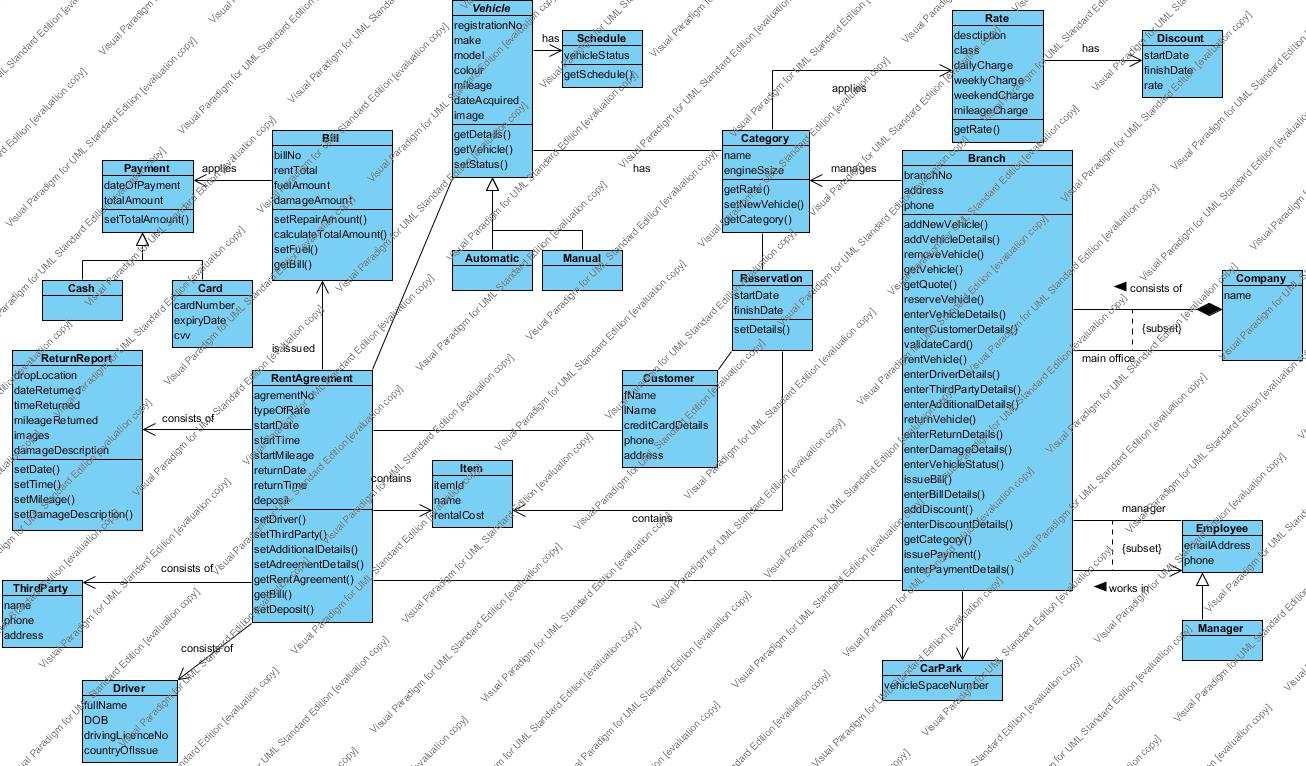
Reserve a vehicle - Happy Scenario



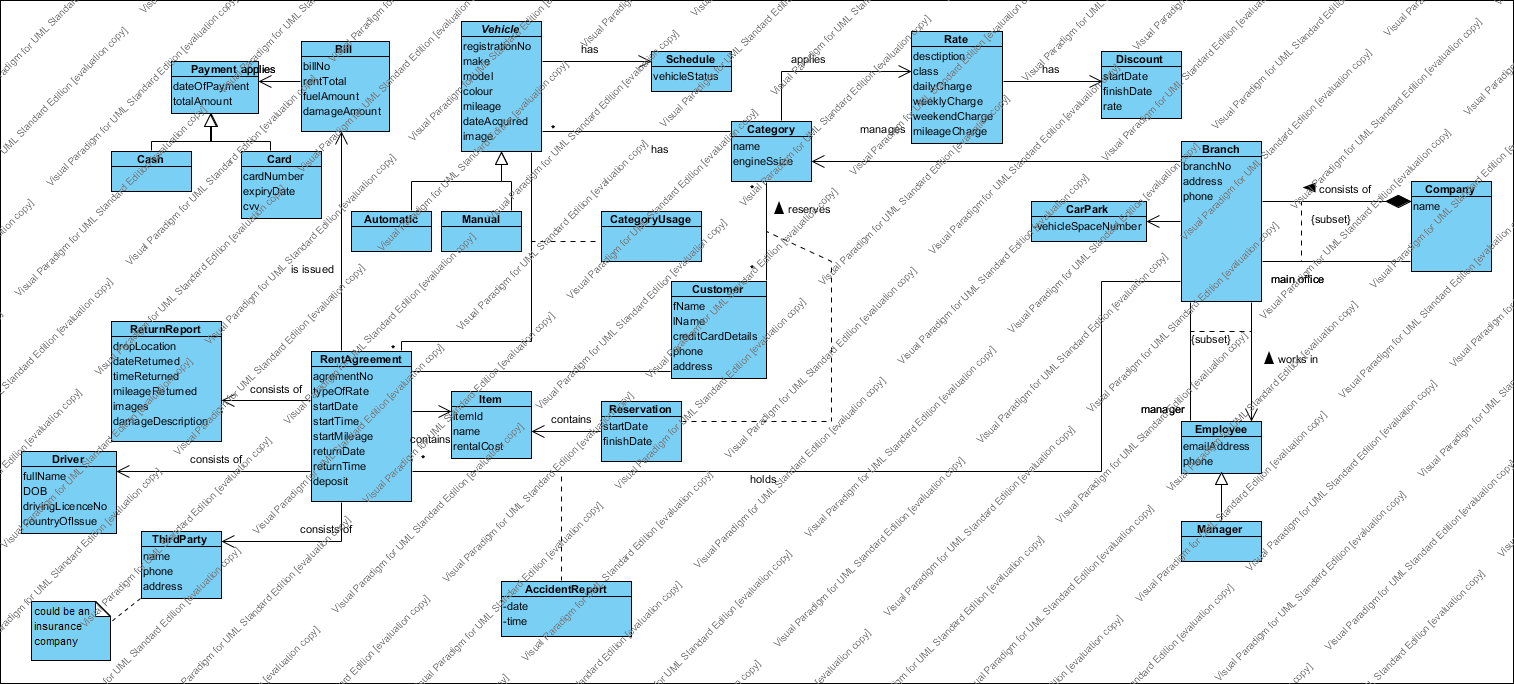
Return a Vehicle - Happy Senario



# 8. Create a Design Class Diagram to support interaction diagrams developed in (vii)



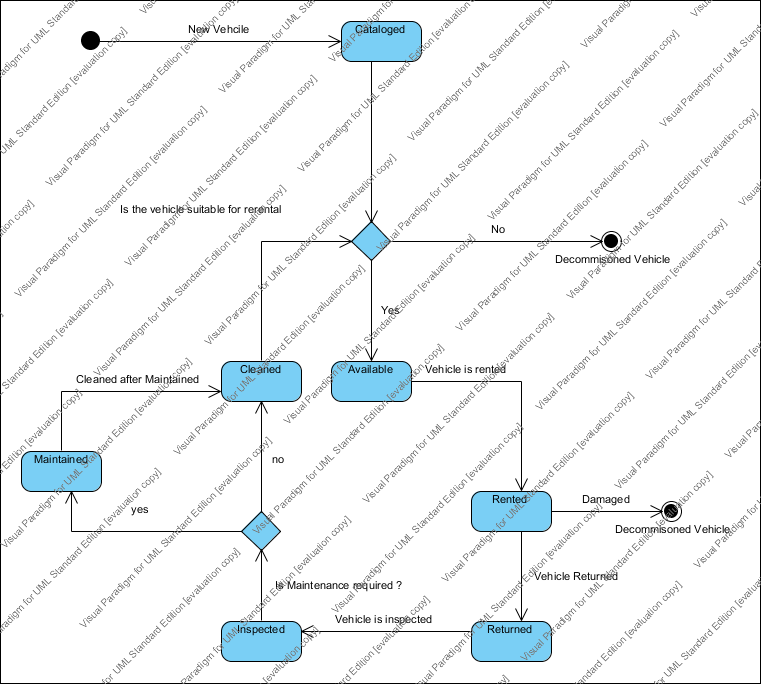
# 9. Extend Class Diagram to allow for the following additional requirements.



The manager would like to determine the license numbers of all drivers for a specific vehicle that has been involved in a motorway accident while it was being rented.

The manager would like to determine which category of vehicle was the most popular during a given time period.

# 10. Develop a state chart Diagram (state transition diagram) for the different possible states of a vehicle based on information provided above.



# 11. Explain how the class diagram would be represented in a relational database

The class diagram can be represented in a relational database by converting each class into a relational table. You identify the tables by putting the classes names into titles and the attributes from the class become columns titles.

Links between classes need to be mapped using foreign keys.

**One to one**

One to one we add a foreign key which references the other table.

**One to many**

To the 'many' table, we add a foreign key referencing the 'one' table.

**Many to many**

We need to add a third table to record the multiple associations between two classes.

An example of this is below

Vehicle table {registrationNo, make, model, colour, mileage, dateAcquired, image}

Every class normally becomes a structure for one table in the database.

Part B

# Q1 Explain the meaning of the MVC pattern using the following sample code.

What it does is put up a black circle that gradually turns red, then the program stops.

The Model does the work. Whenever it does something that another class might be interested in, it "broadcasts" the fact to all other classes that might be needed to know. The Model does not need to know or care anything about which other classes, if any, might be interested, so it remains completely independent of those classes.

The Controller in this program doesn't actually have any controls. It does set up the GUI, creates the Model and the View, and register the View as an observer of the Model.



# Q2 Explain the meaning of the Observer Design Pattern using the following code

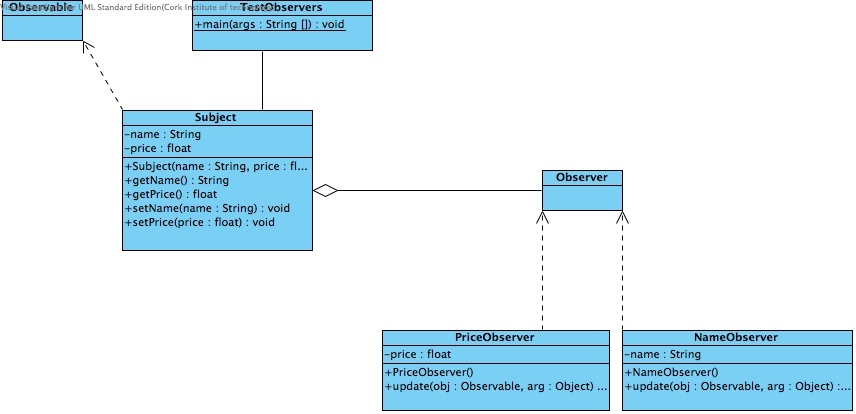
The use of the Observer Design Pattern is the following

Specify a one-to-many dependency between objects so that when one object changes state, all its dependents are given notice and updated automatically.

The "View" part of Model-View-Controller.

The problem:

The problem with the Observer Design Pattern is a large design does not scale easily as new graphing or monitoring requirements are imposed.



# Q3 Explain the meaning of the facade design pattern using the following code

The Facade design pattern simplifies the interface to a complex system. it is usually composed of all the classes which make up the subsytems.

A Facade shields the user from the complex details of the system and provides them with a simplified view of it which is easy to use. It also decouples the code that uses the system from the details of the subsystems, making it easier to modify it later.

What the programme does is it looks over and get the CPU power, memory and hard drive information.

