

Faculty of Computing, Engineering and the Built  
Environment



Undergraduate  
Programme  
Academic  
Year  
2014---2015  
**Coursework:  
Team  
Project**

Module: *CMP2515 Software Design UG2*

School: **Computing, Telecommunication and Networks**

Module Co-ordinator: **Professor Zhiming Liu**

Setup Date: **20/02/2015**

Submission Date: **21/04/2015**

Team Number: T24

Team Leader: Ziaul Islam

Team Members:

Cameron King, Abdirizaq Warsame, Rashid Khan, Yousef Ibrahim & Kamal  
Ducale

Software  
Design

April 16

2015

---

Birmingham City University, Prof Zhiming Liu

Team Project

## Contents

Introduction .....	5
System Functions .....	6
Evident Functions:.....	6
Hidden Functions: .....	6
Frill Functions:.....	6
Essential use cases .....	7
Register a new customer: .....	7
Record a hire:.....	7
Calculate costs: .....	7
Determine if a particular car is due for a specified service: .....	7
Record the details of the mechanics: .....	7
Use Cases .....	8
Register a new customer: .....	8
Record a hire:.....	8
Calculate costs: .....	8
Determine if a particular car is due for a specified service: .....	8
Record the details of the mechanics: .....	8
Use Case Diagrams.....	9
Register a new customer: .....	9
Calculate costs: .....	9
Record a Selecting hire: .....	10
Determine if a particular car is due for a specified service: .....	10
Record the details of the mechanics: .....	11
Application Domain: .....	12
Classes:.....	12
Associations: .....	12
Attributes: .....	13
Conceptual Class Diagram:.....	13
Registering a customer Class diagram .....	14
Calculating costs Class diagram .....	14
Recording details of mechanic Class diagram.....	15
Record a hire Class diagram.....	15
Determine if a particular car is due for a specified service Class diagram .....	16

Systems Operations: .....	16
System Contracts .....	17
Operations: .....	17
Pre-Conditions: .....	17
Post-Conditions:.....	17
Design Class Diagrams .....	18
1) Registering a new customer sequence diagram.....	18
Responsibilities to classes .....	18
2) Record a hire.....	19
Responsibilities to classes .....	19
3. Calculating Costs .....	20
Responsibilities to classes .....	20
4) Service for specific car .....	21
Responsibilities to classes .....	21
5) Record the details of the mechanics: .....	22
Responsibilities to classes .....	22
Attendance record .....	23
Task Responsibilities .....	25
Glossary.....	29

## Introduction

It is more beneficial for the company to use an object-orientated program for their system because users are more likely to be able to pick-up and learn an object-oriented program at a faster rate than a command line program because it is how a vast amount of other systems are created and used throughout the world, this decreases the amount of training time each employee needs decreasing costs for the company and saving money, if the company was to use a command line interface then the training time would be greatly increased as would costs for the company as the average employee does not know how to use a command line to complete functions and may feel intimidated by the software and would not be able to perform to their full potential slowing down the efficiency of the company.

Using their current paper based system the company has to use a lot of their time to update and manage their system, every time a record needs to be created or updated the file needs to be created or found within the system and in some cases multiple files will need to be located and used together in order to produce the desired outcome.

## System Functions

### Evident Functions:

1. Register a new customer onto the system.
2. Record that a particular car has been hired.
3. Record that a particular car has been returned to update the hire log.
4. Calculate the cost based on the daily hire rate.
5. Display the appropriate details, and print out a receipt.
6. Record a service for a particular car, together with the date of the service, the type of the service, and the name of the mechanic responsible.
7. Remove a customer from the system.
8. Add a new car to the fleet on the car database.
9. Delete a car that is no longer in the hire fleet.
10. Add a mechanic who has joined the company.
11. Remove the details of a mechanic who has left the company.
12. List the information (history) about all hires for a specified car.
13. List the information (history) about all services that a specified car has had.
14. Ability to amend the hire rate and other money relates functions.

### Hidden Functions:

1. Login security, unique username and password for users.
2. Input validation and verification, ensure data integrity.
3. Manage payroll of employees.
4. Log a completed hire.
5. Determine if a particular car is due for a particular service.
6. Log all information on customers, staff and current hires.
7. Remove information when record removal is requested from the database.
8. Log all transactions into the history file.

### Frill Functions:

1. Ability to add photos of the cars to the system.
2. Time feature to show remaining times of current hires.
3. Ability to pay for purchases on card/credit payment.

## Essential use cases

### Register a new customer:

This covers the first function in the problem description of being able to add a new customer to the system so that their information is stored for later use within the company.

### Record a hire:

This covers the fourth, fifth and sixth function in the problem description of being able to record the information of a hire including the car that has been hired and information on the customer whom has hired the car.

### Calculate costs:

This covers the 7th function in the problem description where the rates for all the cars need to be known to the system which generates money for the company.

### Determine if a particular car is due for a specified service:

This covers the second and third function in the problem description of being able to have car data stored on the system and then being able to see when the car is up for service.

### Record the details of the mechanics:

This case covers the final function in the problem description where the details of the staff need to be stored on the system to make sure that they can be tracked.

## Use Cases

### Register a new customer:

Registering a new customer starts with the customer wanting to be a part of the system, the member of staff will ask the customer for their data and they will input it into the system, once the information input has been completed then the program will run a validation check on the information entered to make sure it is appropriate for the system, once this has been completed without error the customers data will be recorded onto the database meaning the company has a copy of their information for later use.

### Record a hire:

Recording a hire involves information about the car, the customer and the member of staff involved with the transaction. The customer will be linked to the car they have chosen to hire and the member of staffs details will also be recorded into the hire in the event of any future problems, this information for the hire will be recorded into the database and saved so that the company has a record of this hire.

### Calculate costs:

Calculating costs is essential in every aspect of business as it allows the company to see how much money they are making and can make informed decisions for the business based on this information. Costs need to be calculated when cars are hired, when cars are serviced, if any circumstance involves a discount or a refund, staff wages need to be calculated and all tax involved with transactions needs to be calculated.

### Determine if a particular car is due for a specified service:

It is very important for the business that the cars are regularly serviced as if a car is sent out on loan and is in need of a service then that car is out on the road illegally and if the customer driving the car has a crash, any faults with the car or pulled over by the police for not having M.O.T then the business could be paying out for damages and many other expenses.

### Record the details of the mechanics:

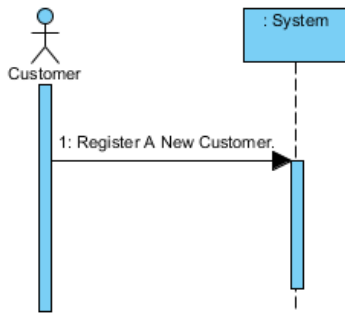
All staff records need to be recorded and regularly updated to ensure the business is as secure as possible and that they do not run into any problems involving their staff, all mechanics that service cars need to be recorded with the service as they are the one at fault if any problems involving the service are to occur.

批注 [ZL1]: Need the expanded use case descriptions

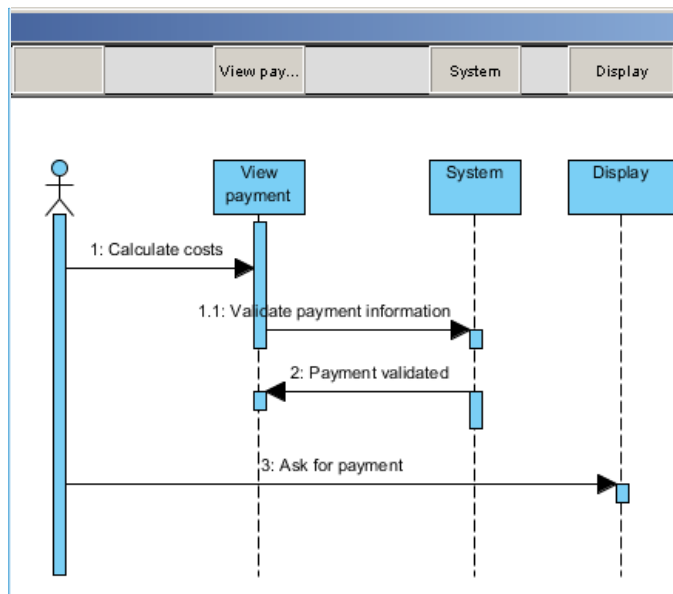


## Use Case Diagrams

### Register a new customer:

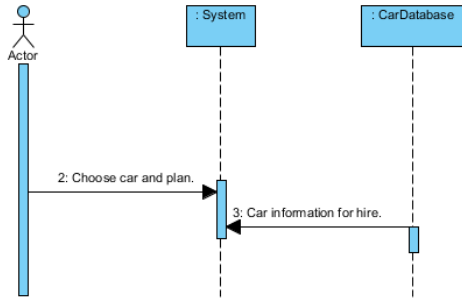


### Calculate costs:

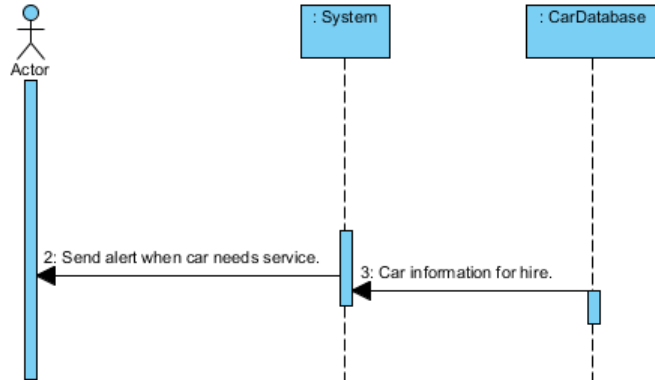


批注 [ZL2]: Not quite a use case sequence diagram

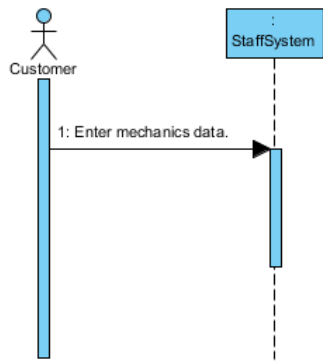
### Record a Selecting hire:



### Determine if a particular car is due for a specified service:



**Record the details of the mechanics:**



## Application Domain:

### Classes:

Customer, this class will contain the fields that will need to be filled in order to add a customer to the customer database.

Register, this class will contain the methods used to add a customer to the customer database, this class will also be used to read items from the database.

Employee, this class will contain the fields that will need to be filled in order to add an employee to the employee database.

Return the car, this class will be used once a hire is a complete and the car is returned to the company.

Remove customer, this class will update the customer database so that their record is no longer in the database.

Car hire, this class will contain the fields that will need to be filled in order to add a hire to the hire database.

Car, this class will contain the fields that will need to be filled in order to add a car to the car database.

CarList, this class will contain the methods used to add a car to the car database, this class will also be used to read items from the database.

Company, this class will contain the main functions, such as: notifying when a car needs a service, entering information into the program and printing out receipt information for hires as well as other functions, this class will also contain the code to create the user interface for the end user.

HireList, this class will contain the methods used to add a hire to the hire database, this class will also be used to read items from the database.

DATE BASE, the database is where all of the records are stored for the system, this helps the efficiency of the business.

### Associations:

The mechanic class will be associated with the remove and add mechanic methods.

The employee class will be associated with the register method which is also associated with the customer class.

The customer class is associated with the remove customer and returned car method.

The car class is associated with the car hire class and the remove car hire and new car hire methods.

All classes are associated with their relevant database.

**Attributes:**

Car – Model (string), Age(int), Class(string), Insurance Group(string), Registration(string), Previous Owners(int), Value(double), Hire Rate(double).

Customer -Name (string), D.O.B (String), Years Driving (int), Email (string), Tell no (String)

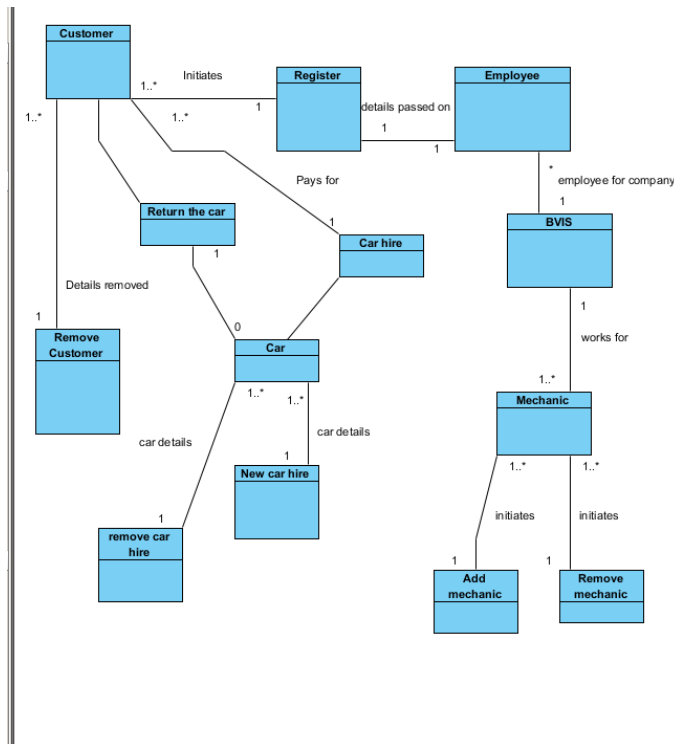
Employee -Name (string), D.O.B (String), Years Driving (int), Email (string), Tell no (String)

CustomerList - Users (array)

Employee - Name(string), password(string), Value(double), login ID(string), Class(string), Age(int), Validation(boolean)

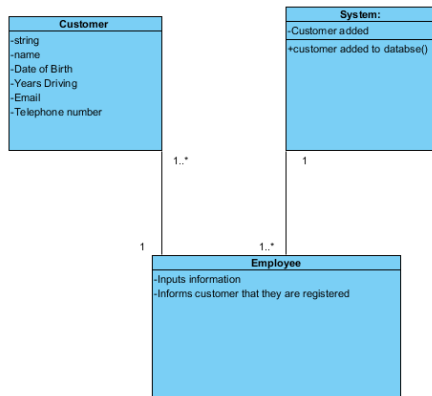
Car Hire - Time (string), Date (string), Car (string), Customer (string), Service Date (string), Return Date (string)

**Conceptual Class Diagram:**

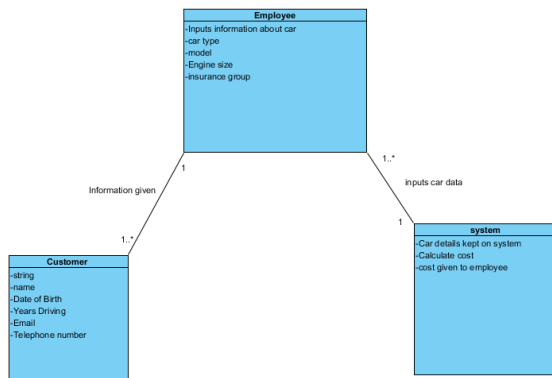


Missing important associations and attributes

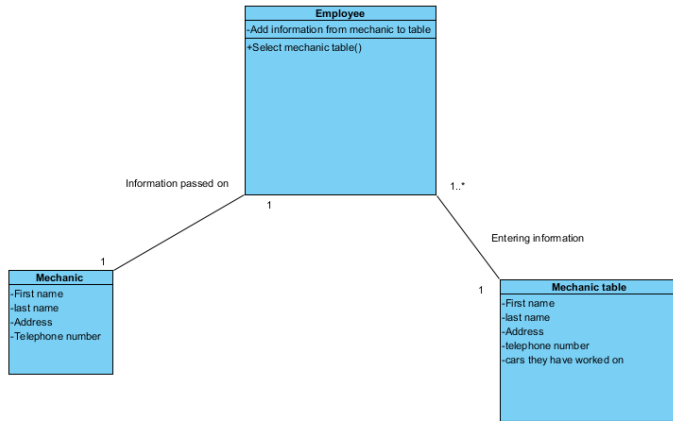
## Registering a customer Class diagram



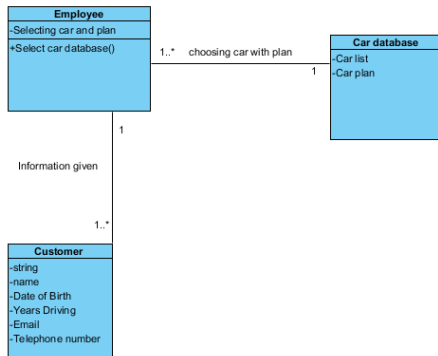
## Calculating costs Class diagram



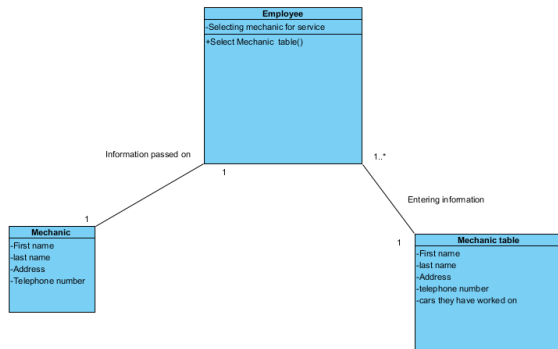
## Recording details of mechanic Class diagram



## Record a hire Class diagram



## Determine if a particular car is due for a specified service Class diagram



### Systems Operations:

Using the logical path needed to be followed in order for the system to carry out the use cases there are some operations that need to be carried out, these included taking information into the system that need to be processed by the system and then operations such as: storing information, displaying outputs, executing calculations and validation methods in order to verify the entered data.

Events that occur in the use cases we have chosen include: Registering a new customer, recording a hire, calculating costs, determining if a car is ready for hire and recording the details of the mechanics.

Below are the sequence diagrams for the typical course of events for the essential use cases we have chosen.



## System Contracts

### Operations:

1. Registering a new customer
2. Recording a hire
3. Calculating costs
4. Determining if a car is due for a service
5. Record the details of a mechanic

### Pre-Conditions:

1. The system is ready to take information  
The details of the customer is known by the member of staff
2. The information required exists within the database  
The system is ready to take information  
A customer has requested the hire with specifics
3. Rates and values known to the system  
Required algorithms to calculate costs are written correctly to give accurate output
4. Car details known to system  
Current date known to the system
5. The system is ready to take information  
Details of the mechanic is known to a member of staff

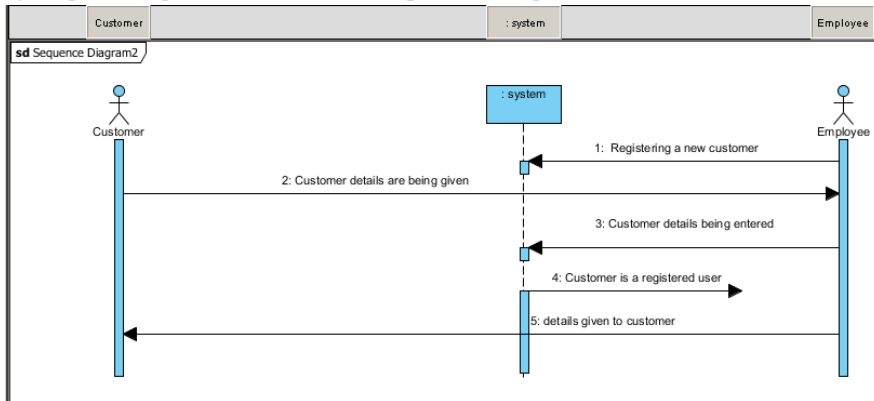
### Post-Conditions:

1. If a new user, a user was created (instance creation)  
A userlist record was created  
Record allows for attribute modification
2. In a new hire, a hire was created (instance creation)  
A hirelist record was created  
Record allows the attribute modification  
If a new hire, the hire is associated with a car and customer
3. Costs is associated with a hire  
Costs are produced on the system  
Values are reset for the next calculation
4. Alert is outputted on the system
5. If a new mechanic, a mechanic was created (instance creation)  
A userlist record was created  
Record allows for attribute modification

批注 [ZL3]: Not very much meaning contracts

## Design Class Diagrams

### 1) Registering a new customer sequence diagram.

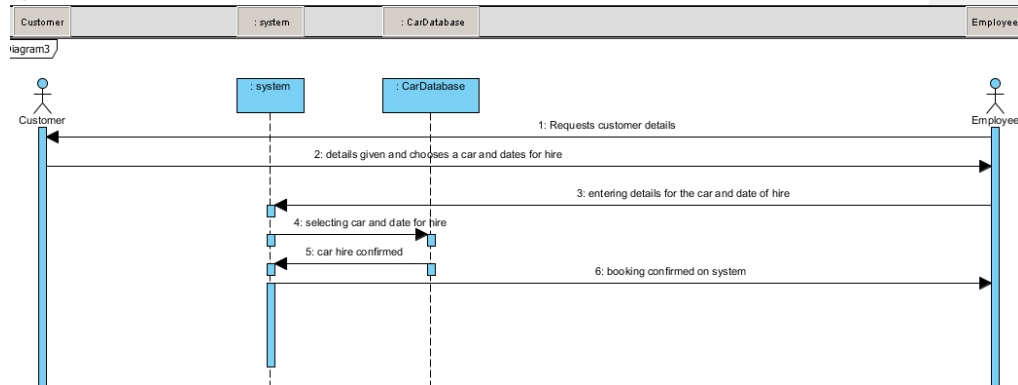


As you can see from the diagram when customer comes to store the employee will ask them to register as a new user so that they can be put into the system, when the customer has to register the customer will provide their personal details to the employee. When the employee gets the personal details of the customer then they will store into the system, when the process of entering the customer details into the system is done the customer will be able will be registered as a new customer. When the customer comes in they will give their name and their personal details will automatically load up.

#### Responsibilities to classes

This function will be dealing with the main company class, the user class and the userlist class. The objects within the company class will be responsible for the obtaining the information which will then go through a validation method to ensure the integrity of the data, once the validation has passed the information will be passed to the user class to create a user record in the correct format, this record will then be passed to the userlist class to be written to the database so that the information can be obtained for later use within the software.

## 2) Record a hire

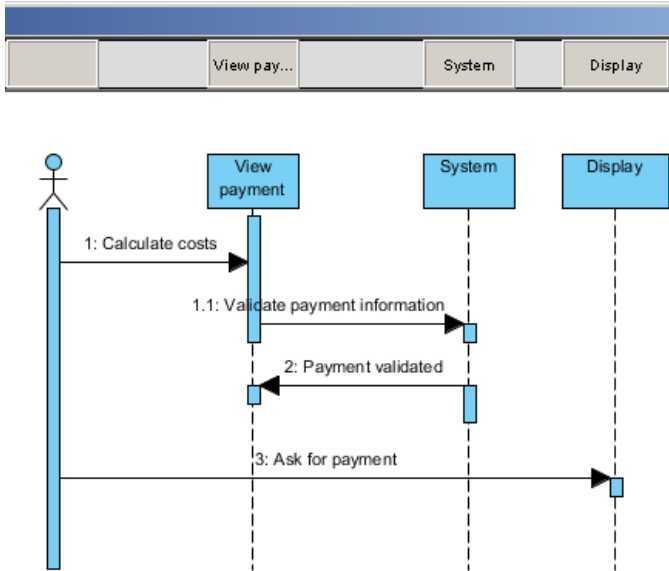


When the customer comes into store to hire a care the employee will ask for their details then the customer provide the details and the car the customer wants to hire and the time and dates the customer wants the car for and the customer details loads for the employee and the employee enters the car details and the date of hire to the system. The system goes into the car hire database to see if that specific car is available for that specific date. The car database confirms to the system that the car is available for specific data, then the booking of the car is confirmed of the system.

### Responsibilities to classes

This function will be dealing the main company class, the car class and the carlist class. The many company class will be responsible for the intake of data from the user can performing the validation checks to ensure that the data is valid, this data will then be passed into the car class to create a car record instance, this record will then be passed into the carlist class so that the record can be written to the database for future use within the company software.

### 3. Calculating Costs

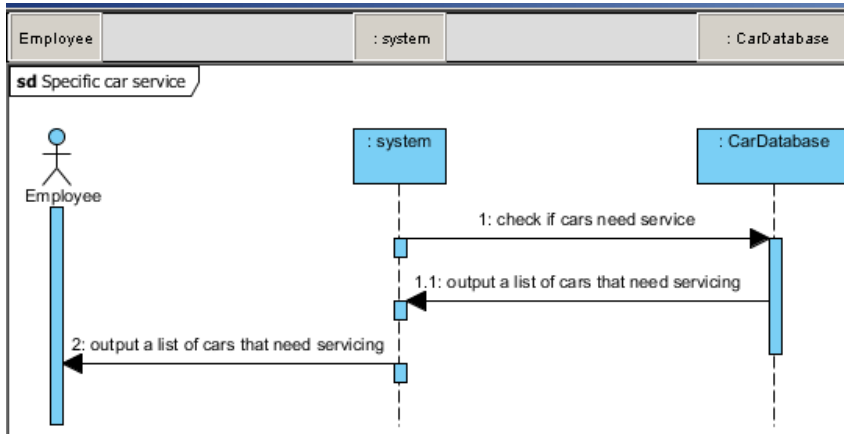


This part of the diagram shows the logic behind the payment and calculating costs function of the software. The customer will communicate to the member of staff entering the data the information about their choice of car, payment plan, and hire plan. Once this information has been entered into the system, the system will retrieve the data from the car database such as: hire rate and service date, this data will be used to calculate the cost of the hire using the information entered regarding the duration of the hire. Once this has been calculated the information will be presented in an understandable format for the staff to inform the customer of the cost of the hire, after this the information will be ready for printing a receipt for the customer and the company.

#### Responsibilities to classes

In this function the classes which will be used are the main company class, the carlist class, the userlist class and the hirelist class. When a customer comes to the company and wants to hire a car there are lots of information needed by the system in order to be able to complete this process. To calculate the costs the process will need to take place within the main company class and acquire information from the related classes, the user from the userclass will be needed to that the costs are related to a customer, then the information about the car used in the hire will be needed to be able to find the rate of the car so that the systems knows how much to charge the customer, this information will be taken from the carlist class, once this has been completed and costs have been calculated then the costs will be used to create a hire instance for the hirelist class.

#### 4) Service for specific car

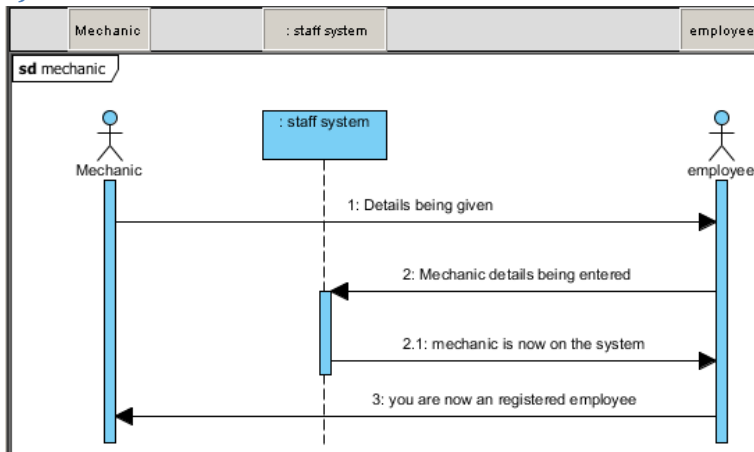


For the service of the specific car the system will check if the specific car will be checked in the car database if the car needs servicing then the database will output the details of the cars that need servicing to the system, then the system will output the cars that need servicing to the employee.

#### Responsibilities to classes

The classes which are going to be used in this process are the main company class and the carlist class. There will be a method within the main class which runs a check through all of the cars to see if any of them are in need of a service. For this the main class will need to use the carlist class to obtain the required information on all of the cars, if an instance occurs where a car is in need of a service then the main class has the information on the specific car and is able to produce a report to the user using the user interface acknowledging them of the instance.

### 5) Record the details of the mechanics:



For this diagram is the mechanic diagram which focus on the fixing part of the vehicle, The mechanic will give his details to the employee which then the employee enters the details that were given by the mechanic, after the details are entered the mechanic will stored in the system and will be on the system. The employee forward mechanic details to the mechanic to show that he’s a registered employee to the car hire company.

#### Responsibilities to classes

This function will be dealing with the main company class, the user class and the userlist class. The objects within the company class will be responsible for the obtaining the information which will then go through a validation method to ensure the integrity of the data, once the validation has passed the information will be passed to the user class to create a user record in the correct format, this record will then be passed to the userlist class to be written to the database so that the information can be obtained for later use within the software.

<b>Attendance record</b>						<b>Weeks</b>	<b>Meetingpoints</b>
Ziaul Islam	Cameron King	Abdirizaq Warsame	Rashid Khan	Yousef Ibrahim	Kamal Ducale		
						02/02/15	Started allocating tasks to give each member of the group for section 1.
						09/02/15	Meeting up to see if the team has any concerns with the tasks they have been allocated with.
						12/02/15	Continued meeting to further discuss section 1.
						16/02/15	Addressing any concerns the team has with moving onto section 2.
						01/03/15	Allocating each task from section 2, to my team members.
						11/03/15	After allocating each task we have held a meeting to address any short comings.
						14/03/15	Continued meetings to further discuss section 2
						21/03/15	Section 2

							coming to an end.
			Holiday			24/03/15	Section 2 has been completed now moving onto section 3.
			Holiday			25/03/15	Discussing points on how to structure section 3.
			Holiday			27/03/15	Started on section 3.
			holiday			04/04/15	Mid way into section 3, discussing any short comings.
			holiday			09/04/15	Continued progress on section 3.
			holiday			17/04/15	Completed section 3.
			Holiday			19/04/15	Team has completed the work, proof read it and will submit it shortly.

Orange= Present.

Purple: Authorized Absence.

Red= Unauthorized absence.



## Task Responsibilities

Task name	Description	Assigned team member(s)	Start date:	Completed by:	End date:
Section 1 Project description	Bvis Car Hire Company: The Problem Description	Ziaul	28 <sup>th</sup> January	Ziaul	17 <sup>th</sup> February
Functions	System functions to be shown from section 4 from coursework booklet.	Cameron	28 <sup>th</sup> January	Cameron	12 <sup>th</sup> February
Essential use cases	Identify the essential use cases so we can then cover them. In addition to this we will need to write out an extended version of the use case.	Abdirizaq & Yousef	28 <sup>th</sup> January	Abdirizaq & Yousef	24 <sup>th</sup> February
Use case diagrams	Create the use case diagram so it can be used to clarify and identify what is happening within the system.	Rashid & Cameron	28 <sup>th</sup> January	Cameron & Rashid	27 <sup>th</sup> February

<p>Identifying classes, associations and attributes.</p>	<p>Using the guidelines, strategies, and notation discussed in the course notes, work through the problem statement and the use cases that you have identified in item 3 to identify classes (concepts), associations, and attributes in the application domain. You should give enough discussion to support your identification.</p> <p>Draw a conceptual class diagram, which includes, the classes, associations, and attributes that you have identified. Again, you only have to consider the functions and the use cases that you considered for item 3. You should give enough discussion to support your identification.</p> <p>Draw a conceptual model, which includes, the classes, associations, and attributes that you have identified. You may find that you need to refine or modify your use cases.</p>	<p>Cameron &amp; Ziaul</p>	<p>28<sup>th</sup> January</p>	<p>Cameron &amp; Ziaul</p>	<p>28<sup>th</sup> march</p>
----------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------	--------------------------------	----------------------------	------------------------------

Task name	Description	Assigned team member(s)	Start date:	Completed by:	End date:
Identifying System Operations	Use the techniques discussed in Chapter 6 of the course notes to identify the system operations from the typical course of events of the use cases that you have produced. Create system sequence diagrams for the typical course of events of the use cases that you think most significant for the development of the system. You may find that you need to refine or modify your use cases and conceptual model that you have produced.	Rashid, Cameron & Kamal	1 <sup>st</sup> march	Cameron, Kamal & Rashid	17 <sup>th</sup> march
System Operation Contracts identified	Based on your use-case model and conceptual model that you have produced write the contracts for the system operations that you have identified. You may find that you need to refine or modify your use-case model and conceptual model while you are working on the contracts.	Abdirizaq, Yousef & Ziaul	1 <sup>st</sup> march	Abdirizaq, Yousef & Ziaul	22 <sup>nd</sup> march

Task name	Description	Assigned team member(s)	Start date:	Completed By:	End date:
Collaboration Diagrams/ Object Sequence Diagrams	The collaboration diagrams or object sequence diagrams (not both) which show the assignment of responsibilities to classes of objects.	Cameron & Ziaul	27 <sup>th</sup> march	Cameron & Ziaul	10 <sup>th</sup> April
Use of patterns regarding the responsibilities of classes of objects	Enough discussion about the use of the patterns in your assignment of responsibilities to classes of objects.	Abdirizaq, Yousef and Kamal	27 <sup>th</sup> march	Abdirizaq, Yousef and Kamal	14 <sup>th</sup> April
The design class diagrams	The design class diagrams, which shows the methods/operations of classes.	Cameron	27 <sup>th</sup> march	Cameron & Ziaul	17 <sup>th</sup> April

## Glossary

Validation - A check to ensure that the data is valid and understandable by the system.

Database - An array of records which contain information stored as a single object for processing.

M.O.T - A Serviced carried out on all cars once per annum.

Attribute - An attribute describes the data held within a field in a record.

Function - A method that performs a specific task and returns a value.

Transaction - A process where money is exchanged for an object or service.

Logical - A task which is performed and follows strict principles of validity.

Operation - A task which is carried out to produce an outcome.

Sequence - When tasks are performed in a specified order.

Algorithms - A problem solving operation.

Method - A process that is carried out to produce an outcome.