|  |
| --- |
| bahir dar university  INSTITUTE OF TECHNOLOGY    School of computing and Electrical Engineering  Computer Science Program |
| Industrial Project I |
| Car Rent and Online Reservation System |
| For Budget Car Rent (BRC) Bahir-Dar Branch |
|  |
|  |

Group Members

NAME IDNO

Nesredin Idris 1035/2001

Yonas Engida 1444/2001

Micheal Birara 0048/1999

Yohannes Afework 1437/2001

|  |
| --- |
|  |

Advisor: Worku K.

February 06, 2012

Acknowledgements

We would like to express our gratitude to our advisor for his insightful comments and technical support to accomplish this paper successfully.

Table of Contents

[List of figures 4](#_Toc316365767)

[List of Tables 5](#_Toc316365768)

[Abbreviations 6](#_Toc316365769)

[Chapter 1: Introduction 7](#_Toc316365770)

[*1.1.* *Background* 7](#_Toc316365771)

[*1.1.1.* *Background of the organization* 7](#_Toc316365772)

[*1.1.2.* *Mission and Vision of the organization* 7](#_Toc316365773)

[*1.2.* *Existing System* 8](#_Toc316365774)

[*1.2.1.* *Existing system function* 8](#_Toc316365775)

[*1.2.2.* *Problems in existing system* 8](#_Toc316365776)

[*1.3.* *Proposed System* 9](#_Toc316365777)

[*1.4.* *Project Scope* 9](#_Toc316365778)

[*1.5.* *Objective of the Project* 9](#_Toc316365779)

[*1.5.1.* *General Objective* 9](#_Toc316365780)

[*1.5.2.* *Specific Objective* 9](#_Toc316365781)

[*1.6.* *Methodology* 9](#_Toc316365782)

[*1.6.1.* *Data gathering methods* 9](#_Toc316365783)

[*1.6.2.* *Design Method* 10](#_Toc316365784)

[Chapter 2: System Features 11](#_Toc316365785)

[*2.1.* *User Requirement* 11](#_Toc316365786)

[*2.1.1.* *Functional Requirement* 11](#_Toc316365787)

[*2.1.2.* *Non-Functional Requirements* 13](#_Toc316365788)

[*2.2.* *System Requirement* 14](#_Toc316365789)

[*2.2.1. Use-Case Diagram* 14](#_Toc316365790)

[*2.2.2. Use-case Description* 15](#_Toc316365791)

[*2.3.* *Analysis Model* 22](#_Toc316365792)

[*2.3.1.* *Activity Diagram* 22](#_Toc316365793)

[*2.3.2.* *Sequence Diagram (SD)* 31](#_Toc316365794)

[*3.* *Introduction* 38](#_Toc316365795)

[*3.1.* *Deployment Diagram* 38](#_Toc316365796)

[*3.2.* *Architectural Design* 39](#_Toc316365797)

[*3.2.1.* *Class Diagram* 39](#_Toc316365798)

[*3.3.* *User Interface (UI) Design* 40](#_Toc316365799)

[*3.4.* *Data Structure Design* 45](#_Toc316365800)

[*3.4.1.* *Entity Relationship Diagram (ERD)* 45](#_Toc316365801)

[*3.4.2.* *Entity Description* 46](#_Toc316365802)

[*3.4.3.* *Relational schema* 47](#_Toc316365803)

[*3.4.4.* *Normalization* 48](#_Toc316365804)

[*3.4.5.* *Physical data model(PDM)* 49](#_Toc316365805)

[*3.5 Algorithm Design* 51](#_Toc316365806)

[*References* 54](#_Toc316365807)

[Appendix A 55](#_Toc316365808)

[Questions asked during requirement elicitation using interview 55](#_Toc316365809)

[Appendix B 56](#_Toc316365810)

# List of figures

[Figure 1 Use case diagram 14](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320817)

[Figure 2: Activity Diagram - Login 22](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320818)

[Figure 3: Activity Diagram - Reserve a Vehicle 23](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320819)

[Figure 4: Activity Diagram - Rent Registration 24](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320820)

[Figure 5: Activity Diagram - Vehicle Registration 25](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320821)

[Figure 6: Activity Diagram – Search Vehicle 26](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320822)

[Figure 7: Activity Diagram – Update Vehicle 27](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320823)

[Figure 8: Activity Diagram - View Vehicle 28](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320824)

[Figure 9: Activity Diagram - Update Rent 29](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320825)

[Figure 10: Activity Diagram - Cancel Reservation 30](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320826)

[Figure 11: Sequence Diagram - Login 31](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320827)

[Figure 12; Sequence Diagram - Vehicle Reservation 32](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320828)

[Figure 13: Sequence Diagram - Rent Registration 32](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320829)

[Figure 14: Sequence Diagram - Vehicle Registration 33](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320830)

[Figure 15: Sequence Diagram - Search Vehicle 33](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320831)

[Figure 16: Sequence Diagram - Update Vehicle 34](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320832)

[Figure 17: Sequence Diagram - View Vehicle 35](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320833)

[Figure 18: Sequence Diagram - Cancel Reservation 35](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320834)

[Figure 19: Sequence Diagram - Update Rent 36](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320835)

[Figure 20: Sequence Diagram - View Reservation 37](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320836)

[Figure 21: Sequence Diagram - Generate Report 37](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320837)

[Figure 22: Deployment Diagram 38](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320838)

[Figure 23: Class Diagram 39](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320839)

[Figure 24: UI - Home 40](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320840)

[Figure 25: UI - Reservation 41](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320841)

[Figure 26: UI - View Vehicle 42](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320842)

[Figure 27: Vehicle Registration 43](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320843)

[Figure 28: UI - Rent Registration 44](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320844)

[Figure 29: UI - Search Vehicle 44](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320845)

[Figure 30: Entity Relationship Diagram (ERD) 45](file:///C:\CSD%20DOC\4th%20Year%20Documents\Mini%20Project\final\MinVersion1.docx#_Toc316320846)

[Figure 31: Relational Mapping 47](#_Toc316320847)

# List of Tables

[Table 1: Use Case - Login 15](#_Toc316320848)

[Table 2: Use Case - Reserve Vehicle 15](#_Toc316320849)

[Table 3: Use Case - Rent Registration 16](#_Toc316320850)

[Table 4: Use Case - Vehicle Registration 17](#_Toc316320851)

[Table 5: Use Case - Search Vehicle 18](#_Toc316320852)

[Table 6: Use Case - Update Vehicle 18](#_Toc316320853)

[Table 7: Use Case – View Vehicle 19](#_Toc316320854)

[Table 8: Use Case - Update Rent 19](#_Toc316320855)

[Table 9 Use case - Cancel Reservation 20](#_Toc316320856)

[Table 10: Use Case - View Reservation 20](#_Toc316320857)

[Table 11: Use Case - Generate Report 21](#_Toc316320858)

[Table 12: Use Case – Logout 21](#_Toc316320859)

[Table 13: Entity Description 46](#_Toc316320860)

[Table 14: Normalization Tables 48](#_Toc316320861)

[Table 15: PDM - Employee Table 49](#_Toc316320862)

[Table 16: PDM - Customer Table 49](#_Toc316320863)

[Table 17 PDM - CustomerInfo Table 49](#_Toc316320864)

[Table 18: PDM - Vehicle Table 50](#_Toc316320865)

[Table 19: PDM - Reservation Table 50](#_Toc316320866)

[Table 20: PDM - Rent Table 50](#_Toc316320867)

# Abbreviations

* BRC – Budget Rent Car
* UC – USE CASE
* ERD – Entity Relationship Diagram
* UI – User Interface
* AD- Activity Diagram
* SD – Sequence Diagram
* PDM – Physical Data Model

# Chapter 1: Introduction

1. Introduction

This chapter of the project document which provides a general introduction about the industrial project. The chapter contains and describes about background of the organization, existing system functions and problems.

This chapter also describes about the proposed system, objective including the general and specific objectives of the project, as well as the methodologies the we used for data gathering, analysis and design.

## *Background*

### *Background of the organization*

Budget Rental car (BRC) organization is a private limited organization established under the commercial law of Ethiopia and its head office is located in the capital city of Ethiopia. Owned and Managed by an Ethiopian, with a great experience by working in Ethiopia.

This organization through time has extended its branches. Nowadays, it has four branches in Ethiopia; these branches are located in Harar, Hawassa, Jimma, Bahir-Dar.

The organization has a total number of 52 employees, 28 in Addis Ababa, 8 in Bahir-Dar, 6 in Hawassa, 5 in Harrar and 5 in Jimma. The organization has around 40 cars such as Pick up, Cruiser, Mini Buses and Coasters.

### *Mission and Vision of the organization*

***Vision***

We will be a growth-oriented car rental organization by efficiently serving value-conscious customers for all of their car rental occasions.

***Mission***

We will consistently deliver a quality product, friendly service and great value that make customers confident that Budget is their best car rental choice.

## *Existing System*

### *Existing system function*

Budget Car Rental (BRC) organization gives car rental service for both foreign and local customers. This organization carries out its daily work by providing; their service to the customers using manually system. The organization uses a manual system for reserving, renting, register and to keep record of all the rental activities and customer information. the detailed existing system functions are listed as follows: -

* During vehicle reservation the customers reserve a vehicle by making a phone call to the organization; otherwise he/she is expected to go to the organization to make reservation.
* During renting a vehicle the customer personal information, payments status and rent agreements are filled in the car rent agreement form(Appendix B); in order to hold legal contract between the customer and organization for renting the vehicle.
* The organization normal work time schedule is from 1:30am – 6:00pm; therefore the organization gives services for ten and half hours a day.
* The organization makes a general report about the rented vehicles once at the end of the month and generates a report.

### *Problems in existing system*

The existence system has a number of problems in the working procedure for the organization and customers. These problems are: -

1. During reservation a customer reserve a vehicle on phone or expected to go to the organization for reserving. This has its own problem in the business sector.

* The organization phone may be busy or may not work when customer is calling to make reservation.
* It may be difficult to customers to get the phone number or the location of the organization.
* The customers may be resulted to unnecessary extra expense and waste their time.
* Their may accrue duplicated reservation of the same vehicle.
* The organization may not be able to serve many customers.
* The customers may not get service of the organization 24/7 (twenty four hours a day and seven days a week) and limiting their service to local domain only.

1. During renting a vehicle, the customer information and contract made between them are filled in a form.

* The contract form made between the customer and organization may be difficult to find incase of misplacing or disappearing due to massive collection of data.

1. During generating monthly report.

* When generating report, it takes time and it may not be easy to manage and analysis the monthly work due to massive collection of data and the generated report may not be accurate.

## *Proposed System*

This car rent and online reservation system is developed to provide the following services.

1. Customer can reserve a vehicle online form anywhere in the world.
2. Every work process activity is done by computer means no need of hardcopy.

## *Project Scope*

The scope of this project is developing web based system for BRC car rent organization only for Bahir-Dar branch. The functions which cover in this project are we are focusing on making rent vehicle and online reserve. Customers as well as the organization’s staff will be able to use the system effectively.

## *Objective of the Project*

### *General Objective*

The general objective of this project is to develop interactive web based system for the BRC car rent organization.

### *Specific Objective*

Here are some specific objectives that would together help us achieve the overall the project as follows:

* Study the existing system and find out the problem.
* Find the solution for the problem found in existing system.
* Design and build a particular model of this proposed system.
* Deploy the system and maintain it till it fits to the needs of the organization.

## *Methodology*

### *Data gathering methods*

The method used for achieve the development of the project based on the exact need of the organization and to meet their business procedure; we had applied or used two types of data gathering methodologies. These methodologies are ***introspection*** and ***Interview***.

1. **Introspection**

This method has been the primary base for the project. Therefore Using the current or background knowledge and experience of the team, the team was able be to identify and list out the common functionalities and requirements for the project. These helped the team to proceed to the next level. Furthermore, it had been a bridge or cause for other methodology to conduct them in proper method.

1. **Interview**

This methodology encapsulates two types of methods. These methods are ***closed*** and ***open*** interview. So the team has selected an open interview for interviewing the manager and employees for recognizing the existing working procedure of the organization. So the team was able be to gather more information about the organization and requirements (see appendix A).

### *Design Method*

#### System development methodology

A detailed object-oriented design for the system design is used. UML is used again for the graphical representation and documentation of the design.

#### Tools used for analysis and design

The tools we use in this project are:

* Microsoft Visio 2003

We use Microsoft Visio 2003 to draw class diagram, activity diagram, sequence diagram, use case diagram, and graphical user interface.

# Chapter 2: System Features

1. Introduction

This chapter of the project document which provides a system features of the project. This chapter contains and described about User requirement, Functional requirement, Use case diagram, and use case description, Analysis Model (Activity diagram and Sequence Diagram).

## *User Requirement*

*User requirements* are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate. That describes user goals or tasks that the users must be able to perform with the system. User requirements therefore describe what the user will be able to do with the system. (Sawyer I. a.)

### *Functional Requirement*

*Functional requirements* These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. It specifies the software functionality that the developers must build into the product to enable users to accomplish their tasks. (Sawyer I. a.)

**Reservation**

1. The system must allow the customer to register for reservation.
2. The system shall allow the customer to view detail description of particular vehicle.
3. The system must notify on selection of unavailable vehicles while reservation.
4. The system shall present an option for advanced search to limit the vehicle search to specific categories of vehicles search. E.g. By Brand, Type and Model.
5. The system must allow the customers to select specific vehicle using different search category while reservation.
6. The system must view list of available vehicles during reservation.
7. The system shall allow the customers to cancel reservation using reservation confirmation number.
8. The system shall allow the employee to update reservation information.
9. The system shall allow the employee to view reservations made by customers.
10. The system shall presents information on protection products and their daily costs, and requests the customer to accept or decline regulation terms during reservation.
11. The system must be able to provide a unique reservation conformation number for all successfully committed reservations.
12. The system must be able to display reservation summary for successfully committed reservation.

**Log in**

1. The system should allow manager to login to the system using their username and password.
2. The system should allow staff to login to the system using their username and password.
3. The system shall allow the manager to create new user account.
4. The system shall allow manager to change account password.
5. The system shall allow staff to change account password.
6. The system shall allow staff to logout.
7. The system shall allow manager to logout.

**Vehicle**

1. The system should allow staff to register new vehicles.
2. The system shall allow staff to select vehicles in the list.
3. The system shall allow customer to select vehicles in the list.
4. The system shall allow staff to Search vehicles by specific record.
5. The system shall allow customer staff to Search vehicles by specific record.
6. The system shall allow staff to update information of the vehicle in need of modification.
7. The system shall allow staff to display all lists of vehicle.
8. The system shall allow staff to display all available vehicle.
9. The system shall allow customer to display all available vehicle.
10. The system shall allow staff to display all rented vehicle.
11. The system shall allow staff to display all off duty vehicles.

**Rent**

1. The system shall allow staff to register customers into rental list.
2. The system shall allow staff to update about customer rent record details in the rental list.
3. The system shall be able to save all changes made on the customer rent list.
4. The system shall allow staff to select customer rent record by specific search category.
5. The system shall allow staff to search rent record of customers using specific categories.
6. The system shall allow staff to display customers, who rent vehicles.
7. The system shall allow staff to display all customers rent record
8. The system must provide printable summary for successful committed rent.

### *Non-Functional Requirements*

Introduction

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time, and store occupancy. Alternatively, they may define constraints on the system implementation such as the capabilities of I/O devices or the data representations used in interfaces with other systems. Non-functional requirements, such as performance, security, or availability, usually specify or constrain characteristics of the system as a whole. .

**Usability**

* The system provides a help and support menu in all interfaces for the user to interact with the system.
* The user can use the system by reading help and support.

**Security**

* The system provides username and password to prevent the system from unauthorized access.
* The staffs’ password must be greater than eight characters.

**Performance**

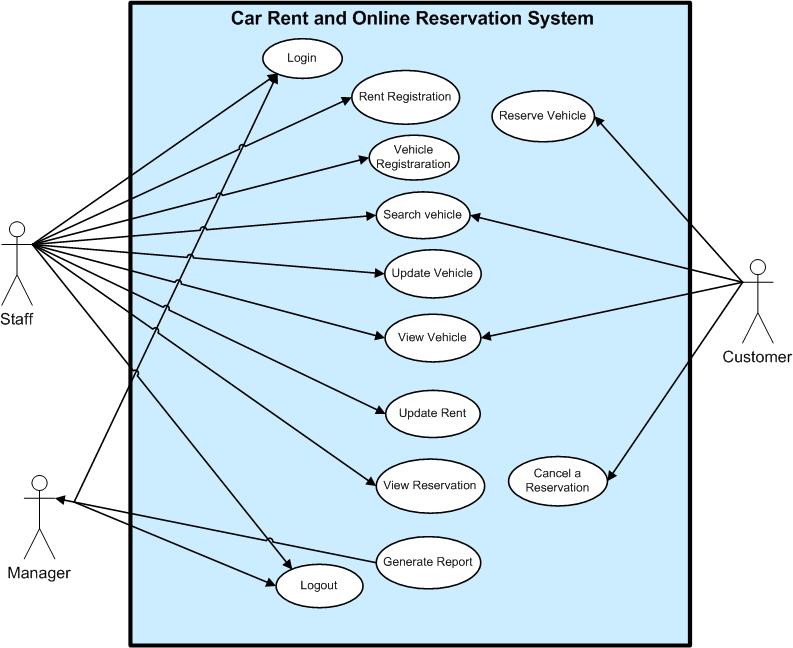
* The system response time for every instruction conducted by the user must not exceed more than a minimum of 10 seconds.
* The system should have high performance rate when executing user’s input and should be able to provide response with in a short time span usually 50 second for highly complicated task and 20 to 25 seconds for less complicated task.

**Availability**

* The system should always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that business process is not severely affected.

## *System Requirement*

### *2.2.1. Use-Case Diagram*



*Figure 1 Use case diagram*

### *2.2.2. Use-case Description*

#### Use-case Login

*Table 1: Use Case - Login*

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-01 | |
| Use-Case Name | Log in | |
| Priority | High | |
| Actor | Staff | |
| Description | This use case describes how Staffs to login into the BRC System. | |
| Precondition | None | |
| Post-condition | If the use case was successful, the actor is now logged into the BRC system. If not, the system state is unchanged. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The staff is on the home page to login to the system.   3. The staff enters username and password, Click on Login Button. | 1. The system promotes the staff to enter Username, Password. 2. The system verifies that all the filled have been filled out and valid. 3. The system successfully logged in the system. 4. Use case Exit |
| Alternate course of Action | 6.1 If all fields are not filled out and not matched to the username and password the system notifies the actor a message Verify Username or Password and then goes back or returns to step 4 of basic course of Action to enter again. | |

#### Use-case Reserve Vehicle

*Table 2: Use Case - Reserve Vehicle*

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-02 | |
| Use-Case Name | Reserve vehicle | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case permits customers to reserve and make schedule for renting vehicle, based on the availability of the vehicle. | |
| Precondition | Customer wants to reserve a vehicle and reservation details about customer have to be entered. | |
| Post-condition | Customers reserve successfully | |
| Basic course of  Action | **User Action** | **System Response** |
| 1. The customer wants to reserve a vehicle.  2. The customer clicks reservation page.  4. The customer enters the following information customer (full name, ID/Passport No, Country, Mobile number and selects vehicle plate number, Pickup date & return date)  5. The customer clicks reserve button to reserve.  8. The customer accepts the reservation and clicks Accept. | 1. The system prompts the customer to fill a reservation form .   6. The system checks all required information had been filled and the date entered dates are valid  7. The system presents information to accept or decline the rental Agreement.  9. The system shows the customer that the reservation has been completed, and presents the customer a reservation confirmation number.  10. Use case ends. |
| Alternate course of Action | 6.1 If the customer enters invalid date and time, the system goes back to step 4 to enter the valid date and time.  6.1 If the customer fills invalid information, the system goes back to step 4 to enter the invalid field again.  7.1 If the customer declines the agreement, the system displays a message that reservation canceled. | |

#### Rent Registration

#### Table 3: Use Case - Rent Registration

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-03 | |
| Use-Case Name | Rent Registration | |
| Priority | High | |
| Actor | Staff | |
| Description | This use case permits to register rental information of the customers and the vehicle that the customer rents. | |
| Precondition | UC-1 | |
| Post-condition | Customer rent information | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The customer wants to take the reserved vehicle.  2. The staff open rent page.  4.The staff enters Full name, Nationality, Country, City, Identification Number, Phone, Plate No, Down Payment, Daily Price, Rent Date, Return Date, Total Rent Day, Total Payment, Refund  5. The staff clicks on rent button. | 3. The system displays a form to be filled out for renting the vehicle.  4. The system prompts to enter the following information.  6. The system verifies that basic fields have been filled out.  7. The system displays successful rent summary  8. Use case Exit. |
| Alternate course of Action | 6.1 If Full name, Nationality, country, City, Id/Passport, Phone, Car Plate No, Down Payment, Price/day, Rent Date, Return date and Total Payment this fields are not filled out system goes back or returns to step 4 of basic course of Action. To fill invalid field. | |

#### Vehicle Registration

*Table 4: Use Case - Vehicle Registration*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-04 | |
| Use-Case Name | Vehicle Registration | |
| Priority | High | |
| Actor | Staff | |
| Description | These use case permits staff to register New Vehicles to the system with detail descriptions about the Vehicle such as condition, Model, Brand, fuel type, Number of sits and amount of price per day. | |
| Precondition | New vehicle Purchased | |
| Post-condition | New Vehicle information stored successfully. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The staff wants to add a new vehicle  2. The staff requests add new vehicle form page.  4. The staff enters the following information in the form.  Vehicle Brand, Vehicle Type, Vehicle Model, Fuel Type, Plate Number, Number of Sits, Condition, Price per day  5. The staff clicks or presses on the save or insert button. | 3. The system response or displays a form to be filled out for vehicle registration.  6. The system verifies that the fields have been filled out correctly.  7. The system displays a successfully stored message to the employee.  8. Use case Exit |
| Alternate course of Action | 6.1 If all fields are not filled out the system goes back or returns to step 4 of basic course of Action. To fill the invalid or the empty field. | |

1. Search Vehicle

*Table 5: Use Case - Search Vehicle*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-05 | |
| Use-Case Name | Search Vehicle | |
| Priority | Medium | |
| Actor | Staff and customer | |
| Description | This use case permits staff and customer to search vehicle from the vehicle list in order to display. | |
| Precondition | UC-3, UC-2 | |
| Post-condition | Display | |
| Basic course of Action | User Action | System Response |
| 1. The staff or Customers clicks on search vehicle link.  3. The staff or customers select one of the following lists from the combo Box, Vehicle Brand. Vehicle Type. Vehicle Model or default is All.  Clicks on search button. | 1. The system displays combo box to select search to a vehicle.   4. Then the system display all information about the vehicle based on selected list.  6. Use case End. |
| Alternate course of Action | 4.1 If any lists are not selected from the combo box system goes back or returns to step 3 of basic course of Action to select from the combo box. | |

#### Update Vehicle

*Table 6: Use Case - Update Vehicle*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-06 | |
| Use-Case Name | Update Vehicle | |
| Priority | High | |
| Actor | Staff | |
| Description | This use case permits staff to update or modify vehicle information. | |
| Precondition | UC-1, UC-5, | |
| Post-condition | updated vehicle information | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The user wants to update vehicle information.  2. Search vehicle by plate number.  4. The staff enters update information of vehicle.  5. The employee click on update button. | 3. The system will display all information about the vehicle.  6. The system successfully updates information in to database.  7. Use case Ends. |
| Alternate course of Action | * 1. If vehicle is not found back to basic course of action 2 | |

#### View Vehicle

*Table 7: Use Case – View Vehicle*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-07 | |
| Use-Case Name | View Vehicle | |
| Priority | Medium | |
| Actor | Staff and customer | |
| Description | This use case allows staff and customer to view or display all vehicles with their detail description about the vehicle. | |
| Precondition | Vehicle Rent, Reserve | |
| Post-condition | Views all vehicles | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The staff or Customer wants view vehicle.  2. The staff or customer click on view vehicles button. | 3. The system retrieves all information about the vehicles.  4. Use case exit. |
| Alternate course of Action | * 1. If in the database no matched vehicle available or empty go to Basic course action of 4. | |

#### Update Rent

*Table 8: Use Case - Update Rent*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-08 | |
| Use-Case Name | Update Rent | |
| Priority | High | |
| Actor | Staff | |
| Description | This use case permits employee to update or modify Rent information incase when there is a need for editing | |
| Precondition | Need to Change information | |
| Post-condition | Successful Update Message | |
| Basic course of Action | **User Action** | **System Response** |
| 1. Staff wants to update rent.  2.Open the rent page  3. Search by unique attribute which is give to customer during rent.  5.The Staff update the information  6. Click on update button. | 4. The system displays the rent information.  7. The system validates updated information and saves updated information in to database.  8. Exit use case. |
| Alternate course of Action | 4.1 If match is not found go back to basic course of action 3.  7.1if the entered information is invalid the system back to basic course of action 5 | |

#### Cancel Reservation

*Table 9 Use case - Cancel Reservation*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-09 | |
| Use-Case Name | Cancel a Reservation | |
| Priority | Medium | |
| Actor | Customer | |
| Description | This use case permits a customer to cancel a reservation. | |
| Precondition | Customer already has reserved and wants to cancel the reservation | |
| Post-condition | Customer successfully cancel a vehicle | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The customer wants to cancel reservation  2. The customer opens reservation page and clicks cancel reservation link  4. The customer enters reservation confirmation number and clicks cancel reservation button.  6. Are you sure you want to cancel, the customer clicks “Yes” button. | 1. The system displays a form   5. The system verifies the field has been filled out correctly and checks validity of confirmation number, then popup a message to verify the canceling.  7. The system cancels the reservation and display a message the reservation is canceled.  8.use case Exit |
| Alternate course of Action | 5.1 If the customer enters invalid number system goes back or returns to step 4 of basic course of Action. To fill invalid or the empty field again.  6.1 If the customer clicks “NO” reservation canceling will be terminated. | |

#### View Reservation

*Table 10: Use Case - View Reservation*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-10 | |
| Use-Case Name | View Reservation | |
| Priority | Medium | |
| Actor | Staff | |
| Description | These use case allow staff to view or display customer reservation. | |
| Precondition | UC-1 | |
| Post Condition | Display all reservations | |
| Basic Course of Action | **User Action** | **System Response** |
| 1. The staff wants to view reservation. 2. The staff requests the reservation Page. 3. Then on reservation page the employee clicks view button. | 1. The system responds the requested page. 2. The system puts on view or displays all reservation information to the employee. 3. Use case ends |
| Alternate course of Action | 5.1 If reservation not found system goes to basic course of action 6. | |

#### Generate Report

*Table 11: Use Case - Generate Report*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-11 | |
| Use-Case Name | Generate Report | |
| Priority | High | |
| Actor | Manager | |
| Description | These use case allow Manager of the organization to generate a report about the renting information of a month. | |
| Precondition | Manager wants to see report | |
| Post Condition | Generate monthly Report Information | |
| Basic Course of Action | **User Action** | **System Response** |
| 1. The Manager wants to generate report. 2. The Manager clicks rent pages. 3. Then on the rent page the Manager specifies the month and then clicks on the generate button. | 1. The system responds the requested page. 2. Use case ends |
| Alternate course of Action | * 1. If the reservation information is empty or not found go to 8. | |

#### Logout

*Table 12: Use Case – Logout*

|  |  |  |
| --- | --- | --- |
| Use-Case Number | UC-12 | |
| Use-Case Name | Log out | |
| Priority | High | |
| Actor | Staff | |
| Description | These use case allow Staff to log out from the system at a time of accomplishing their work. | |
| Precondition | UC-1 | |
| Post Condition | System logs out | |
| Basic Course of Action | **User Action** | **System Response** |
| 1. The Staff or manager wants to log out 2. The Staff or manager clicks the log out button | 1. The system responds to the requested action. 2. The system displays a message that the Staff or manager logged out from the system. 3. Use case Ends |

## *Analysis Model*

### *Activity Diagram*

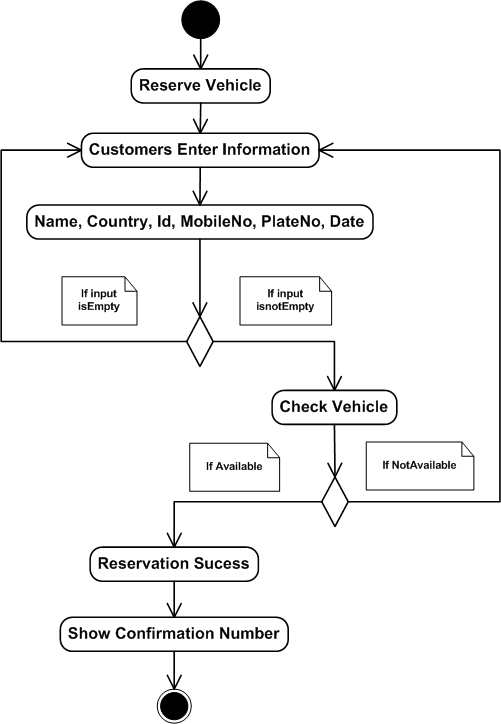
An activity diagram is a variation of a state machine in which the states represent the performance of actions or sub activities and the transitions are triggered by the completion of the actions or sub activities. It represents a state machine of a procedure itself.

#### Activity Diagram - Login

## C:\CSD DOC\4th Year Documents\Mini Project\Diagrams\Activity\AD-01_Login.jpg

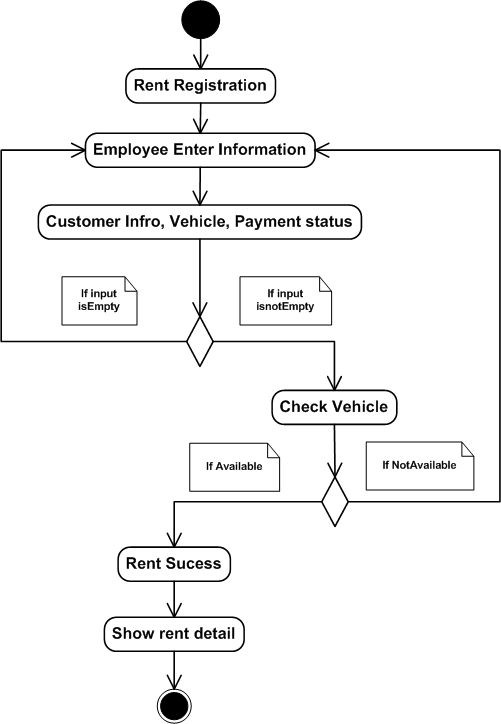
*Figure 2: Activity Diagram - Login*

#### Activity Diagram – Reserve a Vehicle



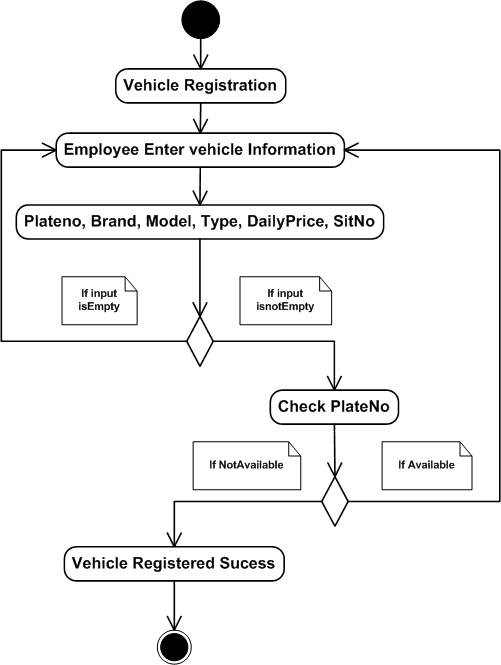
*Figure 3: Activity Diagram - Reserve a Vehicle*

#### Activity Diagram – Rent registration



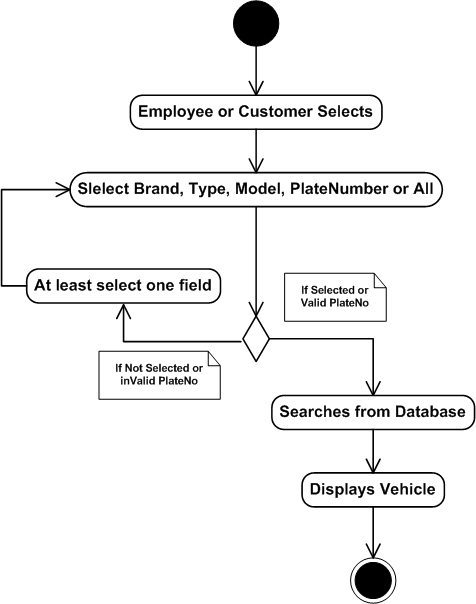
*Figure 4: Activity Diagram - Rent Registration*

#### Activity Diagram – Vehicle registration



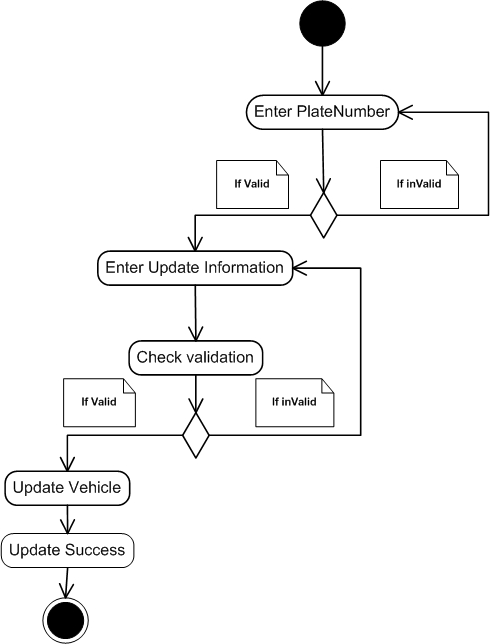
*Figure 5: Activity Diagram - Vehicle Registration*

#### Activity Diagram – Search Vehicle



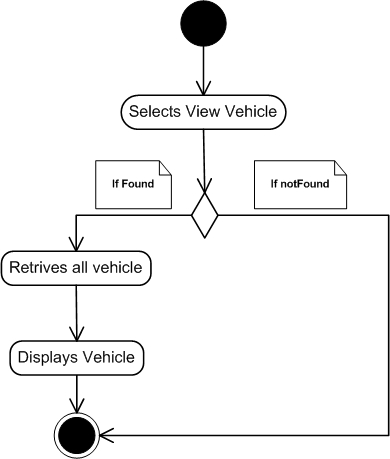
*Figure 6: Activity Diagram – Search Vehicle*

#### Activity Diagram – Update Vehicle



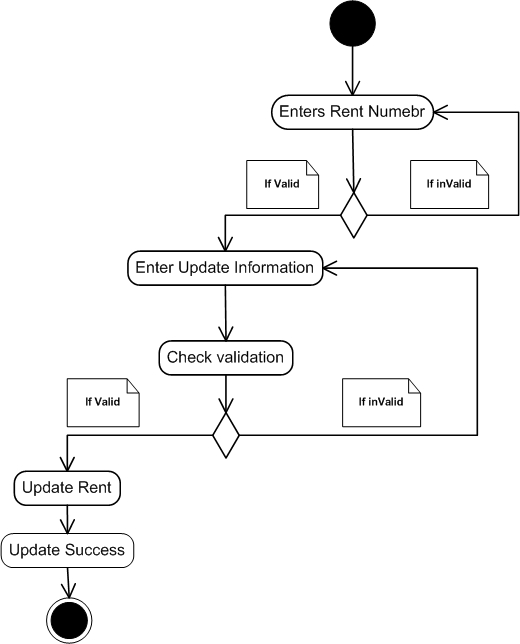
*Figure 7: Activity Diagram – Update Vehicle*

#### Activity Diagram – View Vehicle



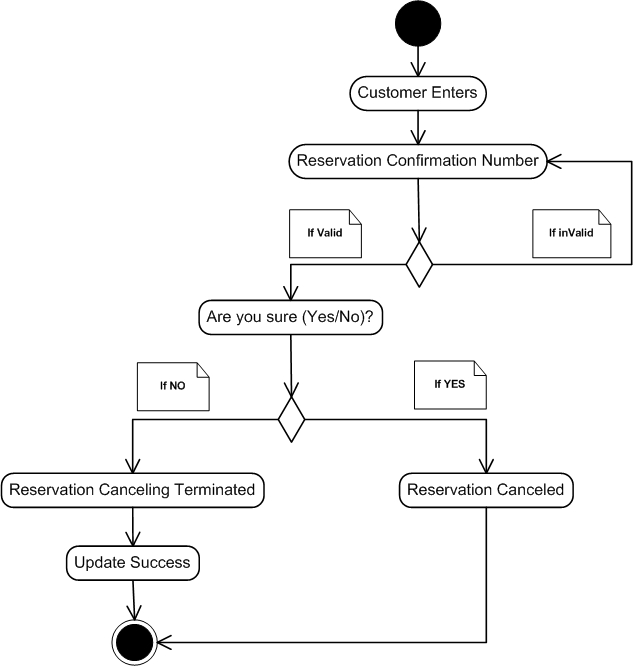
*Figure 8: Activity Diagram - View Vehicle*

#### Activity Diagram – Update Rent



*Figure 9: Activity Diagram - Update Rent*

#### Activity Diagram – Cancel Reservation

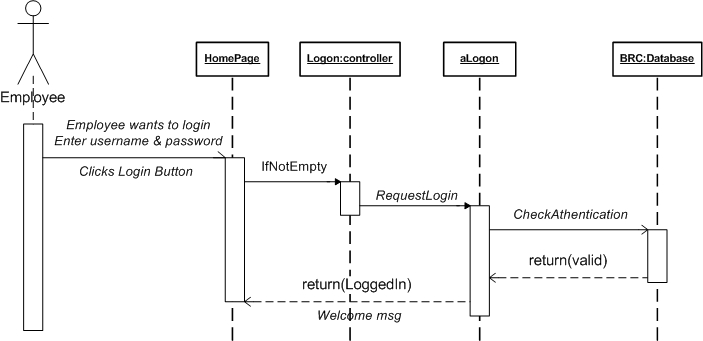


*Figure 10: Activity Diagram - Cancel Reservation*

### *Sequence Diagram (SD)*

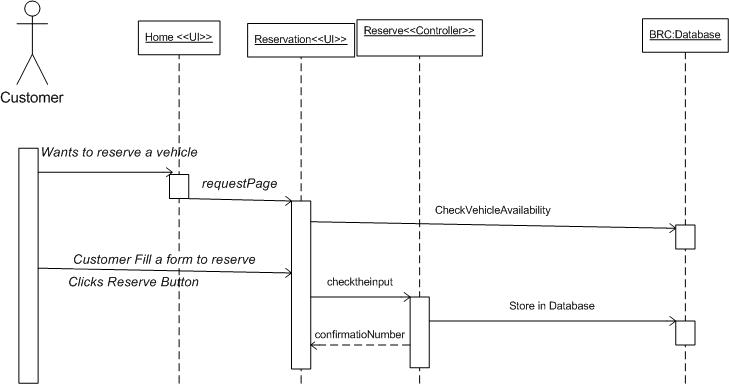
A *sequence diagram* shows an interaction arranged in time sequence. In particular, it shows the instances participating in the interaction by their “lifelines” and the stimuli that they arranged in time sequence. It does not show the associations among the objects. (Sawyer I. a.)

#### Sequence Diagram – Login



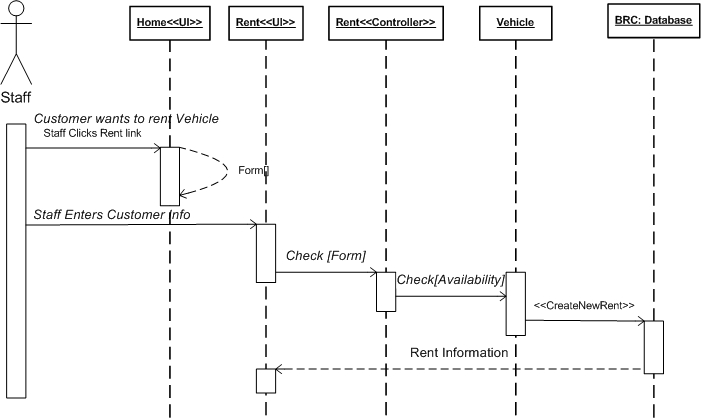
*Figure 11: Sequence Diagram - Login*

#### Sequence Diagram – Vehicle Reservation



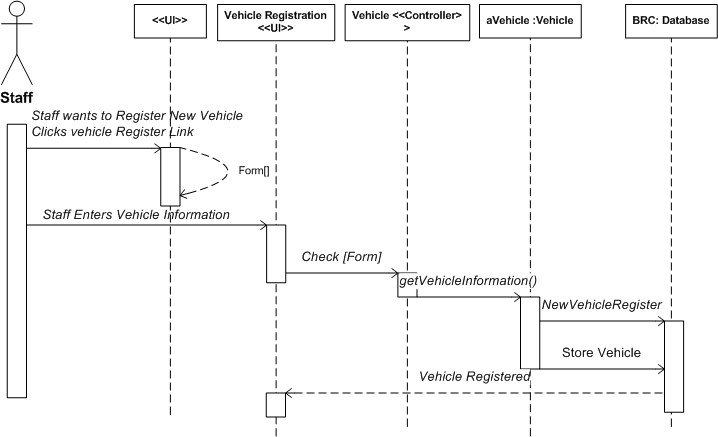
*Figure 12; Sequence Diagram - Vehicle Reservation*

#### Sequence Diagram – Rent Registration



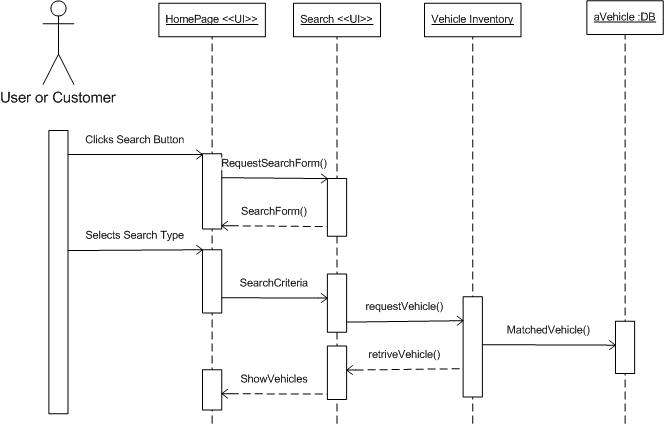
*Figure 13: Sequence Diagram - Rent Registration*

#### Sequence Diagram – Vehicle Registration



*Figure 14: Sequence Diagram - Vehicle Registration*

#### Sequence Diagram – Search Vehicle

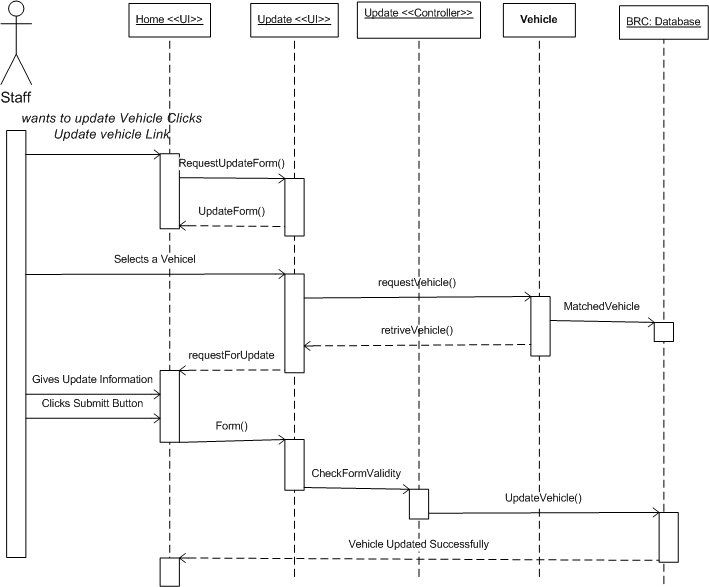


Staff, Customer

**BRC: Database**

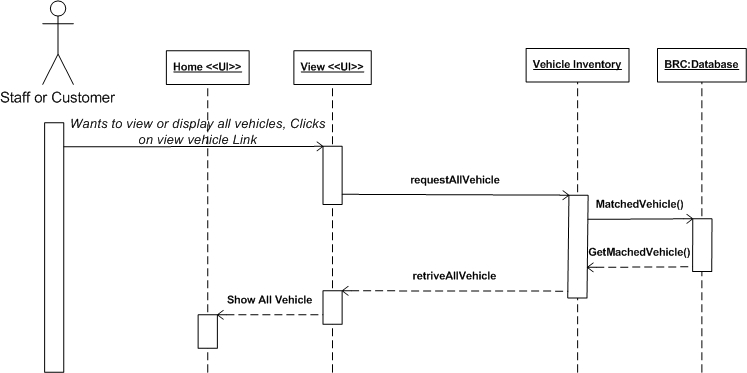
*Figure 15: Sequence Diagram - Search Vehicle*

#### Sequence Diagram – Update Vehicle



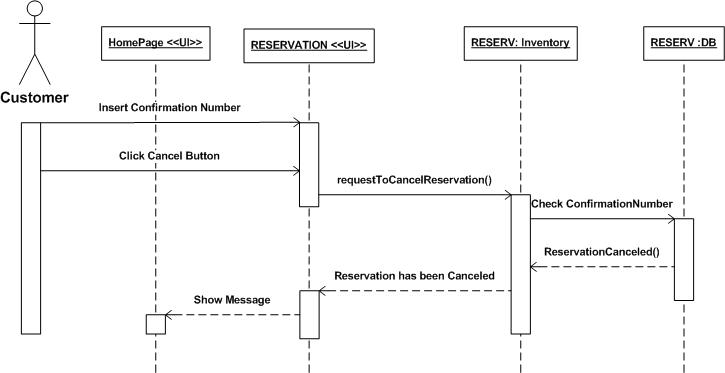
*Figure 16: Sequence Diagram - Update Vehicle*

#### Sequence Diagram – View Vehicle



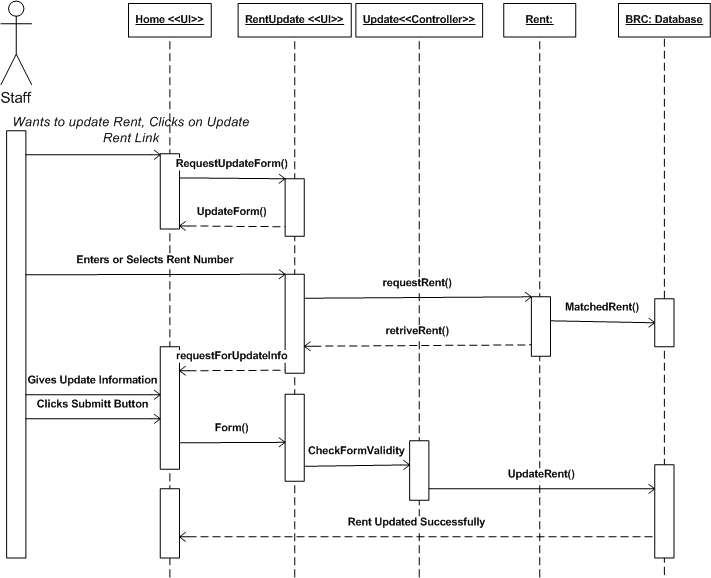
*Figure 17: Sequence Diagram - View Vehicle*

#### Sequence Diagram – Cancel Reservation



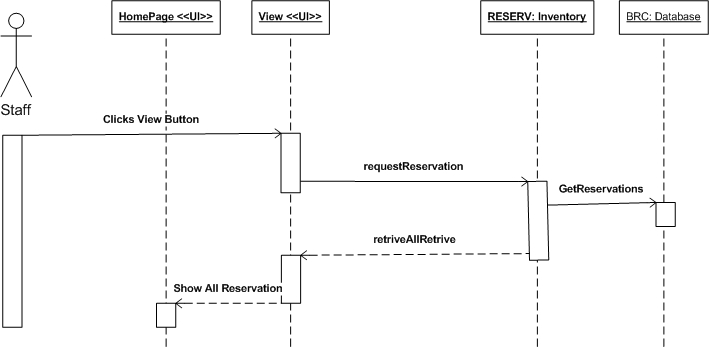
*Figure 18: Sequence Diagram - Cancel Reservation*

#### Sequence Diagram – Update Rent



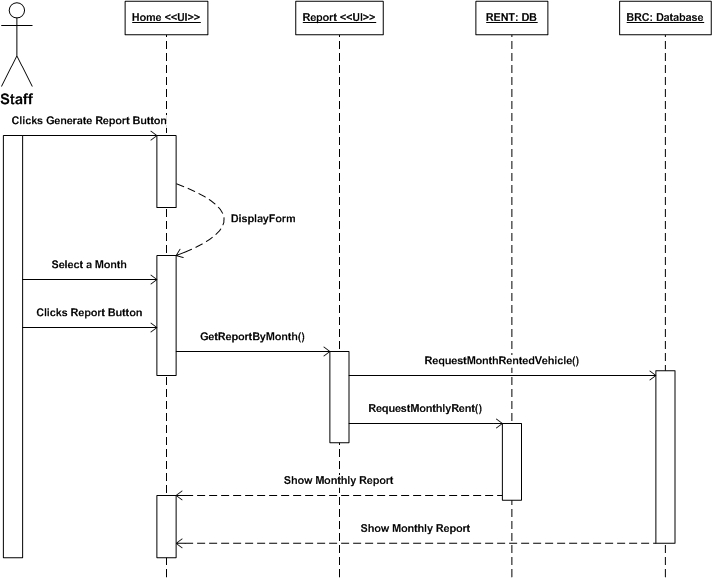
*Figure 19: Sequence Diagram - Update Rent*

#### Sequence Diagram – View Reservation



*Figure 20: Sequence Diagram - View Reservation*

#### Sequence Diagram – Generate Report



*Figure 21: Sequence Diagram - Generate Report*

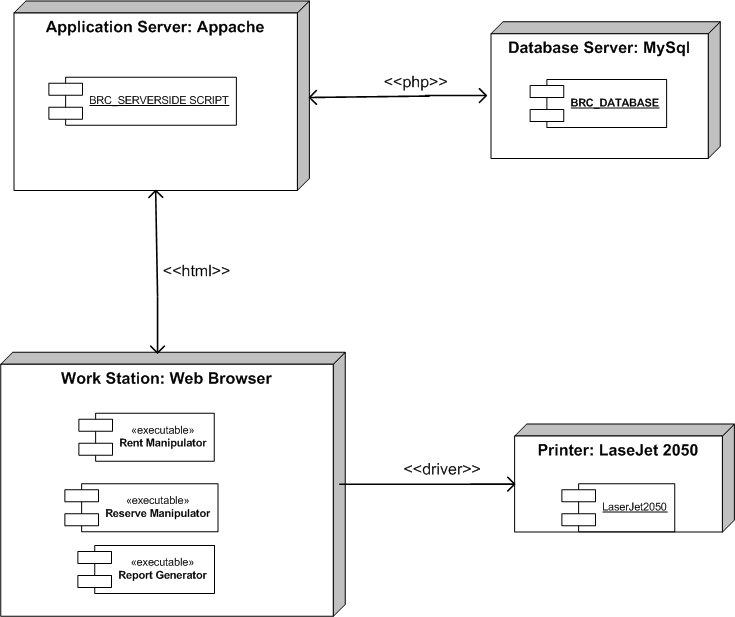
**Chapter Three: SYSTEM DESIGN**

# *Introduction*

This chapter of the project document which provides a system design of this project. This chapter contains and describes about deployment diagram, architectural design, user interface design, data structure design and algorithm design.

## *Deployment Diagram*

Deployment diagrams show the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code units.



*Figure 22: Deployment Diagram*

## *Architectural Design*

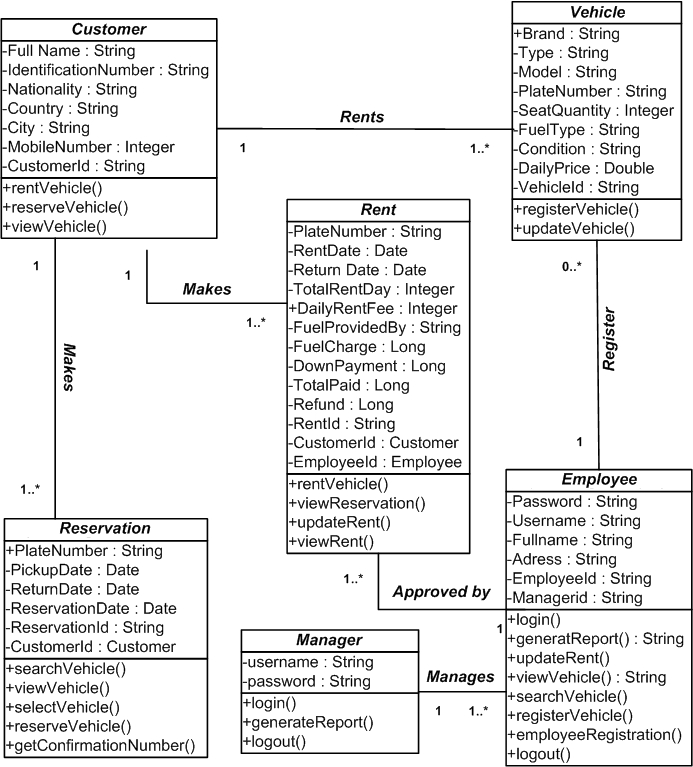
A software system is a set of communicating entities that collaborates to perform task. The architectural Design is a top level design which shows these entities, their relationships.

And we use to describe or show architectural design using class diagram.

## *Class Diagram*

This section discusses classes and their variations, including templates and instantiated classes, and the relationships between classes association and the contents of classes (attributes and operations).

Class diagrams show the static structure of the model, in particular, the things that exist (such as classes and types), their internal structure, and their relationships to other things.



*Figure 23: Class Diagram*

## *User Interface (UI) Design*

User interface is the external part of the system which is used to access and interact with the system easily.

1. **UI- Home Page**



*Figure 24: UI - Home*

1. **UI-Reservation**

These is the external user interface for reserving vehicle, the customers prompt to fill their Full name, Nationality, Mobile Number and after filling personal information they prompt to select the vehicle type, Pick up date and Return date.

**(C) UI- View Vehicle**

*Figure 25: UI - Reservation*

This is the user interface for the user after logged in to the system. The User interface containing links for register vehicle or Rental information and users would be able to update, view and search vehicle or rental information’s.



*Figure 26: UI - View Vehicle*

\

1. **UI-Vehicle Registration**

This is the user interface require username and password to access the system. The User interface containing links for register vehicle or Rental information and users would be able to update, view and search vehicle or rental information’s.

When adding new vehicles you have to fill form.



*Figure 27: Vehicle Registration*

1. **UI-Rent Registration**



*Figure 28: UI - Rent Registration*

1. **UI- Search Vehicle**



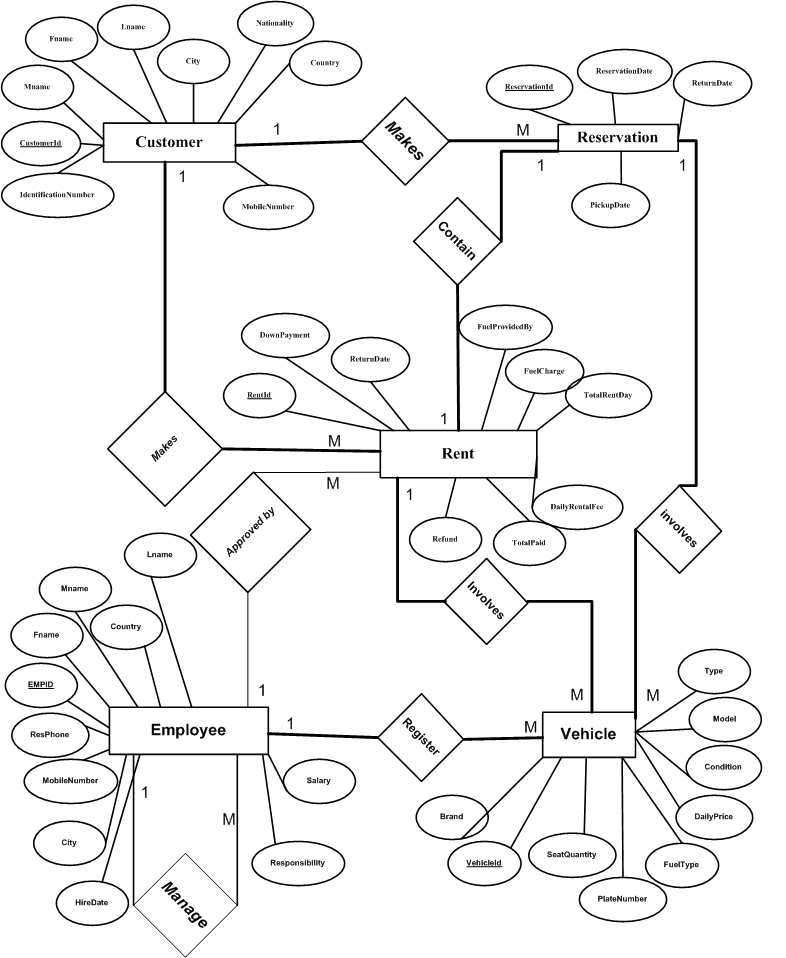
*Figure 29: UI - Search Vehicle*

## *Data Structure Design*

Data structure design describes about data modeling and in this part Entity Relational Diagram, relational mapping, and normalization are described.

### *Entity Relationship Diagram (ERD)*

The entity relationship diagram describes the relation ship between entities, cardinality and their attributes.



*Figure 30: Entity Relationship Diagram (ERD)*

### *Entity Description*

In here we provide a description of entities with all their attributes. Describing entity name, business definition for the entities and there attribute and domain.

*Table 13: Entity Description*

|  |  |
| --- | --- |
| **Entity Name** | **Business definition** |
| Employee | This entity is responsible to store Employee information in the database. |
| Customer | Attribute stores customers’ details information in the database, in order to identify the customer. |
| Vehicle | This entity is stores the information of the vehicle in the database. |
| Reservation | This stores information about the reservations made by a customer. |
| Rent | This stores rental information of the vehicle, payments |

### *Relational schema*

**Employee**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EmpId | Fname | Mname | Lname | Country | City | ResPhone | MobileNumber | Salary | HireDate | Responsiblity | MEMPID |

**Customer**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerId | Fname | Mname | Lname | IdentificationNumber | Nationality | Country | City | MobileNumber |

**Vehicle**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PlateNumber | Brand | Type | Model | SeatQuantity | FuelType | Condition | DailyPrice | EMPID |

**Reservation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ReservationId | ReservationDate | PickupDate | ReturnDate | PN | CID |

**Rent**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RentId | RentDate | ReturnDate | TotalRentDay | DailyRentalFee | FuelProvidedBy | FuelCharge | DownPayment | TotalPaid | Refund | PN | CID | EID | RSID |

*Figure 31: Relational Mapping*

### *Normalization*

Normalization is a process that aims at achieving better designed relational database schemas through the user of semantic information given by Functional dependencies and Primary keys, Normalization process takes a relational schema through a series of tests to certify whether it satisfies conditions. The schemas that satisfy certain condition are said to be in a given “NORMAL FORM’ and unsatisfied schema are decomposed by breaking up their attributes into smaller relations that posses desirable properties. Normalization allows us to organize data that it allows fast access and reduced space. (*Elmasri; FUNDAMENTALS OF DATABASE SYSTEMS, 2003)*.

**Employee**

*Table 14: Normalization Tables*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| EmpId | **Fname** | **Mname** | **Lname** | **Country** | **City** | **ResPhone** | **MobileNumber** | **Salary** | **HireDate** | **Responsiblity** | MEMPID |

**Customer**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerId | **Fname** | **Mname** | **Lname** | **IdentificationNumber** | **Nationality** | **Country** | **City** |

**Customer\_Info**

|  |  |
| --- | --- |
| CustomerId | **Mobile Number** |

**Vehicle**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PlateNumber | **Brand** | **Type** | **Model** | **SeatQuantity** | **FuelType** | **Condition** | **DailyPrice** | EMPID |

**Reservation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ReservationId | **ReservationDate** | **PickupDate** | **ReturnDate** | **PN** | **CID** |

**Rent**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RentId** | **RentDate** | **ReturnDate** | **TotalRentDay** | **DailyRentalFee** | **FuelProvidedBy** | **FuelCharge** | **DownPayment** | **TotalPaid** | **Refund** | **PN** | **CID** | **EID** | **RSID** |

### *Physical data model(PDM)*

*Table 15: PDM - Employee Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | **Employee** | | |
| **Attribute** | **Data type** | **Length** | **Type of attribute** |
| EmpId | Varchar | 50 | Primary Key |
| Fname | Varchar | 50 |  |
| Mname | Varchar | 50 |  |
| Lname | Varchar | 50 |  |
| Country | Varchar | 50 |  |
| City | Varchar | 50 |  |
| ResPhone | int |  |  |
| MobileNumber | int |  |  |
| Salary | float |  |  |
| HireDate | Date |  |  |
| Responsiblity | Varchar | 50 |  |
| MEMPID | Varchar | 50 | Foreign key |

*Table 16: PDM - Customer Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | **Customer** | | |
| **Attribute** | **Data type** | **Length** | **Type of attribute** |
| CustomerId | Varchar | 50 | Primary Key |
| Fname | Varchar | 50 |  |
| Mname | Varchar | 50 |  |
| Lname | Varchar | 50 |  |
| Country | Varchar | 50 |  |
| City | Varchar | 50 |  |
| Nationality | Varchar | 50 |  |
| IdentificationNumber | Varchar | 50 |  |

*Table 17 PDM - CustomerInfo Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | **CustomerInfo** | | |
| **Attribute** | **Data type** | **Length** | **Type of attribute** |
| CustomerId | Varchar | 50 | Foreign Key |
| MobileNumber | int |  | Primary Key |

*Table 18: PDM - Vehicle Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | **Vehicle** | | |
| **Attribute** | **Data type** | **Length** | **Type of attribute** |
| PlateNumber | Varchar | 50 | Primary Key |
| Brand | Varchar | 50 |  |
| Type | Varchar | 50 |  |
| Model | Varchar | 50 |  |
| NumberofSeat | int |  |  |
| FuelType | Varchar | 10 |  |
| Condition | Varchar | 20 |  |
| DailyPrice | float |  |  |
| EMPID | varchar | 50 | Foreign Key |

*Table 19: PDM - Reservation Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | **Reservation** | | |
| **Attribute** | **Data type** | **Length** | **Type of attribute** |
| ReservationId | Varchar | 50 | Primary Key |
| ReservationDate | Date |  |  |
| PickupDate | Date |  |  |
| ReturnDate | Date |  |  |
| PN | varchar | 50 | Foreign Key |
| CID | Varchar | 50 | Foreign Key |

*Table 20: PDM - Rent Table*

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | **Rent** | | |
| **Attribute** | **Data type** | **Length** | **Type of attribute** |
| RentId | varchar | 50 | Primary key |
| RentDate | Date |  |  |
| ReturnDate | Date |  |  |
| TotalRentDay | int | 10 |  |
| DailyRentFee | int | 10 |  |
| FuelProvidedBy | varchar | 30 |  |
| FuelCharge | varchar | 10 |  |
| DownPayment | Float | 10 |  |
| Total paid | Float | 10 |  |
| Refund | Float | 10 |  |
| PN | varchar | 50 | Foreign Key |
| CID | Varchar | 50 | Foreign Key |
| RSID | Varchar | 50 | Foreign Key |

## *3.5 Algorithm Design*

In this part we describe the algorithm of the operations or methods which found in class diagram using Pseoudocode. Pseoudocode is one type of algorithm representation method by using English language.

**Pseoudocode reserving vehicle**

**Steps/procedure**

**Method name=reserve vehicle**

**Begin**

Variables -plate number

-pickup date

-reservation date

-reservation id

-customer id

**If** (\*variables are valid\*)

***Then***

Add to table reserve (plate number, pickup date, reservation date, reservation id

customer id)

***Otherwise***

Display “the inputs are invalid!”

**Endif**

Display “the conformation number”

**End**

**Pseoudocode for rent vehicle**

**Steps/procedure**

**Method name=rent vehicle**

**Begin**

Variables -plate number

-rent date -total paid

-return date -refund

-total rent day -rent id

- daily rent fee -customer id

-fuel provided by -employee id

-down payment

If (\*variables are valid\*)

Then

Add to table rent (rent date, total paid, return date ,refund, total rent day, rent id, daily rent fee, customer id, fuel provided by, employee id, down payment)

Otherwise

Display “the inputs are invalid!”

**Endif**

**End**

**Pseoudocode employee for registration**

**Steps/procedure**

**Method name=employee registration**

**Begin**

Variable - full name

-address

-employee id

-user name

-password

**If** (\*inputs are valid\*)

**Then**

Add to table employee (full name, address, employee id, user name, password)

**Otherwise**

Display “inputs are not valid try again!”

**Endif**

**End**

**Pseoudocode register vehicle**

**Steps/procedure**

**Method name=register vehicle**

**Begin**

Variables-brand

-model

-plate number

-number feat

-fuel type

-condition

-daily price

**If** (\*variables are valid\*)

**Then**

Add to table vehicle (brand, model, plate number, number feat, fuel type, condition,

Daily price)

Otherwise

Display “inputs are invalid”

**Endif**

**End**

**Pseoudocode for login**

**Steps/procedure**

**Method name=login**

**Begin**

Variables:-username

-password

**If** (\*inputs are valid\*)

(\*select the previous username and password from database and compare with entered\*)

**If** username = entered username and

Password = entered password

(\*go to login page\*)

Otherwise

Display “login error!”

**Endif**

Otherwise

Display “inputs are invalid try again!”

**Endif**

**End**

# *References*

Elimasir. (3343). *database.* azxa.

Elmasri, R.

*FUNDAMENTALS OF DATABASE SYSTEMS.* (2003).

jacobson, G. i. (2003). *Applying use cases.*

Sawyer, I. a. *Requirement Engineering a good Practice.*

Sawyer, S. &. *Requirement Enginering.*

Will, T. (2009). *SSD.*

# Appendix A

## Questions asked during requirement elicitation using interview

Q1. What makes your organization different from other organization who rents a car?

Q2. What is the objectives of your organization?

Q3. What is the mission of your organization?

Q4. How many branches does your organization have?

Q5. How many employees do you have?

Q6. How does your current system work?

A. Is it manual?

B. is it computerized?

C. is it semi computerized?

Q7. If your answer for question number 6 is choice “b” or “c” what computer applications do you use?

Q8. How many cars do you have?

Q9. What kinds of car models do you have?

Q10.What is the procedures or steps when a customer rents a car?

Q11.What qualifications are expected from a customer who wants to rent a car?

Q12.Where do you keep customer and rental information’s?

Q13.How do you keep track of which cars are rented and which are not?

Q14.How many cars a client can rent at a time?

Q15. How do you generate customer and rental information’s?

# Appendix B