## CAR RENTAL SYSTEM

By

## YASIR ADO HASSAN BU/18A.IT/3041

DEPARTMENT OF COMPUTER SCIENCE BAZE UNIVERSITY ABUJA

## DESIGN AND DEVELOPMENT OF CAR RENTAL SYSTEM

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

## B.Sc. IN SOFTWARE ENGINEERING

BY

## YASIR ADO HASSAN BU/18A/IT/3041

TO

## THE DEPARTMENT OF COMPUTER SCIENCE FACULTY OF COMPUTING AND APPLIED SCIENCE BAZE UNIVERSITY, ABUJA

September, 2020

# DECLARATION

This is to certify that this Report entitled “DESIGN AND DEVELOPMENT OF CAR RENTAL MANAGEMENT SYSTEM ”, which is submitted by YASIR ADO HASSAN in partial fulfilment of the requirement for the award of degree for B.Sc. in Software Engineering to the Department of Computer Science, Baze University Abuja, Nigeria, comprises of only my original work and due acknowledgement has been made in the text to all other materials used.

Date: September, 2020. Name of Student: YASIR ADO HASSAN

**APPROVED BY** ……………………

#### Dean

Faculty of Computing & Applied Science

# CERTIFICATION

This is to certify that this Report entitled DESIGN AND DEVELOPMENT OF CAR RENTAL MANAGEMENT SYSTEM, which is submitted by YASIR ADO HASSAN in partial fulfilment of the requirement for the award of degree for B.Sc. in Software Engineering to the Department of Computer Science, Baze University Abuja, Nigeria is a record of the candidate’s own work carried out by the candidate under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: Supervisor: Prof. Sylvanus Ehikioya

Date: Supervisor: Dr. Samuel Ubaru

# APPROVAL

This is to certify that the research work, CAR RENTAL MANAGEMENT SYSTEM and the subsequent preparation by YASIR ADO HASSAN with BU/18A/IT/3041 has been approved by the Department of Computer Science, Faculty of Computing and Applied Science, Baze University, Abuja, Nigeria.

By

Prof. Sylvanus Ehikioya Date/Sign

1st Supervisor

Dr. Samuel Ubaru Date/Sign

2nf Supervisor

Dr. Chandrashekhar Uppin Date/Sign

Head, Department of Computer Siences

Professor Mohammed Hammawa Baba Date/Sign Dean, Faculty of Computing and Applied Science

Prof. Ahmad Baita Garko Date/Sign

External Examiner

# DEDICATION

This work is dedicated to God Almighty, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this program. This work is also dedicated to my parents and my lecturers.

Thank you, my love for you all can never be quantified. God bless you.

# ACKNOWLEDGEMENTS

All form of praise and thanks is due to Allah, the creator of mankind, the most merciful and gracious for his blessings, protections, courage and guidance.

I would like to acknowledge and thank Prof. Sylvanus Ehikioya, who stood by me and made sure I did everything regarding this project efficiently and effectively, my gratitude to him is unparalleled. I also thank my second supervisor Dr. Samuel Ubaru and my parents for their immense support, guidance and encouragement to make sure that I complete this program successfully. I would also like to acknowledge my lecturers and all who have contributed to this point in the success of my academic pursuit.

## ABSTRACT

This application presents a data management system for a car rental company. This enables the administrator to keep track of all the customers information. This system increases customer retention and simplify vehicle and customers management in efficient way. The car rental management system has a very user-friendly interface. Thus, the users will feel very easy to work on it. By using this system admin can manage their rental, bookings, customer issues and vehicle issues etc. The car rental information can be added to the system, or existing information can be edited or deleted by the administrator. The transaction reports of the car rental system can be retrieved by the administrator, when its required. Thus, there is no delay in the availability of any car information, when ever needed the car rental information can be captured very quickly and easily.

Table of Contents

[ABSTRACT](#_bookmark0) [VI](#_bookmark0)

[CHAPTER 1: INTRODUCTION](#_bookmark1) [1](#_bookmark1)

[OVERVIEW](#_bookmark2) [1](#_bookmark2)

[BACKGROUND AND MOTIVATION](#_bookmark3) [1](#_bookmark3)

[STATEMENT OF THE PROBLEM](#_bookmark4) [2](#_bookmark4)

[AIM AND OBJECTIVES](#_bookmark5) [3](#_bookmark5)

[SIGNIFICANCE OF THE PROJECT](#_bookmark6) [3](#_bookmark6)

[PROJECT RISKS ASSESSMENT](#_bookmark7) [3](#_bookmark7)

[SCOPE/PROJECT ORGANIZATION](#_bookmark9) [4](#_bookmark9)

[CHAPTER 2: LITERATURE REVIEW](#_bookmark10) [6](#_bookmark10)

[INTRODUCTION](#_bookmark11) [6](#_bookmark11)

[HISTORICAL OVERVIEW](#_bookmark12) [6](#_bookmark12)

[RELATED WORK](#_bookmark13) [7](#_bookmark13)

[SUMMARY](#_bookmark28) [21](#_bookmark28)

[CHAPTER 3: REQUIREMENTS ANALYSIS AND DESIGN](#_bookmark29) [22](#_bookmark29)

[OVERVIEW](#_bookmark30) [22](#_bookmark30)

[PROPOSED METHODOLOGY](#_bookmark31) [22](#_bookmark31)

[APPROACH TO CHOSEN METHODOLOGY/METHODS](#_bookmark32) [23](#_bookmark32)

[TOOLS AND TECHNIQUES](#_bookmark33) [23](#_bookmark33)

[REQUIREMENT ANALYSIS](#_bookmark34) [24](#_bookmark34)

[REQUIREMENT SPECIFICATIONS](#_bookmark35) [24](#_bookmark35)

[Functional Requirement Specifications](#_bookmark36) [24](#_bookmark36)

[Non-Functional Requirement Specifications](#_bookmark38) [25](#_bookmark38)

[SYSTEM DESIGN](#_bookmark40) [25](#_bookmark40)

[Use case Diagram](#_bookmark41) [25](#_bookmark41)

[Class Diagram](#_bookmark45) [29](#_bookmark45)

[Database Design](#_bookmark47) [30](#_bookmark47)

[Dataflow Diagram (DFD)](#_bookmark49) [31](#_bookmark49)

[Activity Diagram](#_bookmark52) [33](#_bookmark52)

[Application Architecture](#_bookmark58) [38](#_bookmark58)

[SUMMARY](#_bookmark62) [40](#_bookmark62)

[CHAPTER 4: IMPLEMENTATION AND TESTING](#_bookmark63) [41](#_bookmark63)

[OVERVIEW](#_bookmark64) [41](#_bookmark64)

[DEVELOPMENT TOOLS AND TECHNOLOGIES](#_bookmark65) [41](#_bookmark65)

[IMPLEMENTATION PROBLEMS](#_bookmark66) [41](#_bookmark66)

[TESTING](#_bookmark75) [46](#_bookmark75)

[Test Plans](#_bookmark76) [46](#_bookmark76)

[Test Traceability Matrix](#_bookmark81) [48](#_bookmark81)

[Test Report Summary](#_bookmark83) [48](#_bookmark83)

[USE GUIDE](#_bookmark85) [49](#_bookmark85)

[SUMMARY](#_bookmark86) [49](#_bookmark86)

[CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS](#_bookmark87) [50](#_bookmark87)

[OVERVIEW](#_bookmark88) [50](#_bookmark88)

[OBJECTIVE ASSESSMENT](#_bookmark89) [50](#_bookmark89)

[LIMITATIONS AND CHALLENGES](#_bookmark90) [51](#_bookmark90)

[FUTURE ENHANCEMENTS](#_bookmark91) [51](#_bookmark91)

[RECOMMENDATIONS](#_bookmark92) [52](#_bookmark92)

[REFERENCES](#_bookmark93) [53](#_bookmark93)

[APPENDICES](#_bookmark94) [55](#_bookmark94)

[Appendix A 55](#_bookmark95)

[Appendix B 56](#_bookmark96)

[Appendix C 57](#_bookmark97)

[Appendix D](#_bookmark98) [58](#_bookmark98)

[Appendix E 61](#_bookmark99)

[Appendix F 63](#_bookmark100)

[Appendix G 65](#_bookmark101)

# LIST OF TABLES

[TABLE 1. 1 RISK AND PREVENTION 3](#_bookmark8)

[TABLE 2. 1 FUNCTIONAL REQUIREMENTS 24](#_bookmark37)

[TABLE 2. 2 NON-FUNCTIONAL REQUIREMENTS 25](#_bookmark39)

[TABLE 3. 1 TEST PLANS 46](#_bookmark77)

[TABLE 3. 2 47](#_bookmark78)

[TABLE 3. 3 47](#_bookmark79)

[TABLE 3. 4 47](#_bookmark80)

[TABLE 3. 5 TEST TRACEABILITY MATRIX 48](#_bookmark82)

[TABLE 3. 6 TEST REPORT SUMMARY 48](#_bookmark84)

**LIST OF FIGURES**

[FIGURE 1. 1 **ENTITY RELATIONSHIP DIAGRAM** 8](#_bookmark14)

[FIGURE 1. 2 **SOFTWARE DEVELOPMENT LIFE CYCLE** 9](#_bookmark15)

[FIGURE 1. 3 **SYSTEM COMPARISON ANALYSIS FOR CAR RENTAL MANAGEMENT INFORMATION SYSTEM OF AVIS**](#_bookmark16)[**INDONESIA** 10](#_bookmark16)

[FIGURE 1. 4 **FLOW CHAT FOR CAR RENTAL MANAGEMENT INFORMATION SYSTEM OF AVIS INDONESIA** 11](#_bookmark17)

[FIGURE 1. 5 **DFD LEVEL 0** 11](#_bookmark18)

[FIGURE 1. 6 **ENTITY RELATIONSHIP DIAGRAM CAR RENTAL SYSTEM** 12](#_bookmark19)

[FIGURE 1. 7 **STATE TRANSITION DIAGRAM** 13](#_bookmark20)

[FIGURE 1. 8 **MAIN WEB INTERFACE DESIGN** 14](#_bookmark21)

[FIGURE 1. 9 **CAR RENTAL SYSTEM TEST RESULTS** 15](#_bookmark22)

[FIGURE 2. 1 **THE HOME PAGE OF SCR & TOUR WEBSITE** 17](#_bookmark23)

[FIGURE 2. 2 **THE BOOKING FORM OF SCR & TOUR** 18](#_bookmark24)

[FIGURE 2. 3 **THE HOME PAGE OF WS-RENT-A-CAR** 19](#_bookmark25)

[FIGURE 2. 4 **THE BOOKING FORM OF WS-RENT-A-CAR** 19](#_bookmark26)

[FIGURE 2. 5 **THE BOOKING FORM OF NOVA CAR HIRE** 20](#_bookmark27)

[FIGURE 2. 6 **ADMIN USE CASE DIAGRAM** 26](#_bookmark42)

[FIGURE 2. 8 **CUSTOMER USE CASE DIAGRAM** 27](#_bookmark43)

[FIGURE 2. 9 **CAR RENTAL SYSTEM USE CASE DIAGRAM** 28](#_bookmark44)

[FIGURE 3. 1 **CLASS DIAGRAM** 29](#_bookmark46)

[FIGURE 3. 2 **ENTITY RELATIONSHIP DIAGRAM (ERD)** 30](#_bookmark48)

[FIGURE 3. 3 **CONTEXT LEVEL DFD** 31](#_bookmark50)

[FIGURE 3. 4 **LEVEL 0 DFD** 32](#_bookmark51)

[FIGURE 3. 5 **REGISTRATION ACTIVITY DIAGRAM** 33](#_bookmark53)

FIGURE 3. 6 **PROFILE MODIFICATION ACTIVITY DIAGRAM ERROR! BOOKMARK NOT DEFINED.**

[FIGURE 3. 7 **CAR RESERVATION ACTIVITY DIAGRAM** 34](#_bookmark54)

[FIGURE 3. 8 **PLACING ORDER ACTIVITY DIAGRAM** 35](#_bookmark55)

[FIGURE 3. 9 **ADDING NEW CAR ACTIVITY DIAGRAM** 36](#_bookmark56)

[FIGURE 4. 1 **VIEW REPORT ACTIVITY DIAGRAM** 37](#_bookmark57)

[FIGURE 4. 3 **CAR RENTAL SYSTEM APPLICATION ARCHITECTURE** 38](#_bookmark59)

[FIGURE 4. 4 **ADMIN APPLICATION ARCHITECTURE** 39](#_bookmark60)

[FIGURE 4. 6 **CUSTOMER APPLICATION ARCHITECTURE** 40](#_bookmark61)

[FIGURE 4. 7 **FIRST IMPLEMENTATION PROBLEM** 42](#_bookmark67)

[FIGURE 4. 8 **SECOND IMPLEMENTATION PROBLEM** 42](#_bookmark68)

[FIGURE 4. 9 **THIRD IMPLEMENTATION PROBLEM** 43](#_bookmark69)

[FIGURE 5. 1 **FOURTH IMPLEMENTATION PROBLEM** 43](#_bookmark70)

[FIGURE 5. 2 **FIRST PROBLEM ENCOUNTERED SOLUTION** 44](#_bookmark71)

[FIGURE 5. 3 **SECOND PROBLEM ENCOUNTERED SOLUTION** 45](#_bookmark72)

[FIGURE 5. 4 **THIRD PROBLEM ENCOUNTERED SOLUTION** 45](#_bookmark73)

[FIGURE 5. 5 **FOURTH PROBLEM ENCOUNTERED SOLUTION** 46](#_bookmark74)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| CPU | Central Processing Unit |
| ERD | Entity Relationship Diagram |
| IT | Information Technology |
| ISP | Internet Service Provider |
| IIS | Internet Information Service |
| MIS | Management Information System |
| SDLC | Software Development Life Cycle |
| DFD | Data Flow Diagram |
| ERD | Entity Relationship Diagram |
| PHP | Hypertext Processor |
| SMS | Short Message Service |
| UML | Unified Modelling Language |
| RAM | Random Access Memory |
| IDE | Integrated Development Environment |
| SQL | Structure Query Language |
| CSS | Cascading Style Sheet |
| SRS | Software Requirement Specification |
| HDD | Hard Drive |
| MVC | Model View Controller |
| HTML | Hypertext Mark-up Language |

## CHAPTER 1: INTRODUCTION

### Overview

As the internet improves the life of people, it also gives access to things that were inaccessible before. The internet is one of the most important tools of communication. The world has become a place where there is a lot of technological development which brings the result of almost every single thing has been transformed into computerized form. These days, individual activities have been changed into work done by information systems. One of which is the primary objective of this project which is about car rental management system. Renting car system exists in the previous years where people rent cars for their own reasons. Car rentals is basic to numerous individuals’ arrangement to travel or move from one place to another for business purposes, tour, and visit occasions. Thus, car rental is extremely useful.

My car rental management system is a web-based system for an organization that rents out cars. This system empowers the organization to make their services accessible to the public through the web and furthermore keep records about their services.

### Background and Motivation

#### Background

Car rental has been around for over a century! When cars were introduced to people, clever entrepreneurs identified a growing demand for renting. However, before cars are invented, people would still rent out horses and carriages. Even in ancient Rome, chariots were rented. The earliest records of car hire can be traced as far back as 1904, when a bike shop in Minneapolis began offering vehicles for rent. A couple of years later, the German company, named “Sixt Car Hire” was built up in 1912. It previously began with three vehicles for rent and immediately extended. In 1915, in Omaha, Nebraska, Joe Saunders had a moment and understood that no one in Omaha offered vehicles for renting. He employed only one salesperson, Frank Arndt, from Germany, and began renting Arndt's Model T. This endeavor ended up successfully that by 1917, he was renting

120 of these. He named the organization 'Saunders Drive-It-Yourself System' and put an advertisement in the local paper. Joe Saunders' three siblings and father began participating in the business and by 1927, the Saunders System had branches in more than 85 urban areas.

The information technology nowadays is growing from time-to-time, consequently, the requests of utilizing Internet are expanding year-by-year. Furthermore, most of the traditional companies are moving their traditional business model into the modern business model. With the emergence

of globalization and modernization supported by information and communication technology today one could easily tell that its story has changed and transformed. These gave birth to the numerous problems we have today in the car rental companies. Unlike before where all the processes when renting a car are done manually. However, this is neither effective nor efficient; customers are constantly not satisfied with the manual car rental processes.

#### Motivation

In consideration with historical and rapid development of car rental companies, the way processes in the companies are taking place today which is quite problematic, this project is planned to ease those processes through developing an effective and efficient car rental system, just like other developed countries are using technologies towards facilitating their customers processes through projects like Sixt Car Hire, Germany.

### Statement of the Problem

The manual car rental system provides services only during office hours. So, customers have limited time to make any transactions or reservation of the cars. The problem with some of the current system is that some small companies already have a car rental system which is not a web- based application. This is a limitation that gives them capability to store customer’s details, but at the same time they cannot make their services more available to the public through the internet, they rather make use of posters to advertise their services to the public. They also make use of phone call reservations which is also limited to few features as compare to a web base system. For example, a customer might make a phone call reservation for a particular car, but when he/she comes to pick the car, he/she might turn not to like the car, this could be because the customer could not see a sample picture of the car he/she wants to rent. Below are some problems with running car rental company manually:

* To rent a car, a prospective renter must first go to the nearest office to register as a client, what of if the customer doesn’t have enough time to do that?
* Cars that provide difficulties to rent out are normally advertised in local or national newspaper. It involves a lot of paper work and consumes time.
* The process of managing customer’s data is slow if the company is using manual system and there might be thousands of clients.
* It is very hard to keep record of all rental cars and so on.

### Aim and Objectives

From the above-mentioned problems, the aim is to develop a computer-based information system that will help to address the ongoing issues from the manual information system and help to facilitate some tasks that seems to be difficult for both the car rental company’s staff and those who are renting the cars (customers). The main objective is to design and implement a car rental management system for an organization. Specific objectives are:

* To develop a simple and secure system that protects client information and confidential information of the organization
* To design a user-friendly system that enables client check for availability of vehicle and book or reserve a vehicle online.
* To design a system that enables clients pay their car rent online
* To develop a system that stores bookings and reservations information as well as payment history to help the organization keep track of transactions.
* To implement geofencing and remote car deactivation upon expiration of rental period.

### Significance of the Project

Projects provide a flexible framework for engaging students in exploring curricular topics and developing important skills, such as communication, teamwork, and technology skills.

The car rental management system will help to solve numerous problems associated with the manual way of doing things. Errors, waste of precious time and energy will be eliminated with the system. This will in turn enhance productivity and efficiency in an organization.

Also, it would help students and researchers that are working actively towards enhancing the car rental management system, this work would serve as a reference to them as they strive to develop the car rental company technologically. Additionally, issues of insecurity, trustfulness could be resolved, because payment system would be linked to the system to eliminate these issues between people and the scammers as well as customers and the company’s staffs.

### Project Risks Assessment

Table 1. 1 Risk and Prevention

|  |  |
| --- | --- |
| **RISK** | **PREVENTION** |

|  |  |
| --- | --- |
| Loss of power | There is likelihood of loss of power when working on the project. To avoid that, the laptop used last for nearly five hours which is good enough  between the times taken to bring back the light. |
| Inability to carry out research due to loss of hardware or software resources | Required hardware will be bought instantly while relevant software that may likely be lost will be kept in the computer for easy repair or  reinstallation. |
| Loss of work due to equipment failure /loss | Weekly data backup to portable hard drive |
| Lack of Internet access | To prevent lack of internet access, two means of internet access was provided purposely for this project, which is not from the same ISP and one is  wired whereas the other one is wireless. |

### Scope/Project Organization

The aim of this project is to develop a web-based car rental management system. At this stage of development, the project would be launched using IIS Express web server which already comes along with Visual Studio.

The remaining part of this project is organized as follows:

Chapter Two contains literature review; historical overview of the system and some related works and incredible achievements made by other researchers and entrepreneurs towards enhancing car rental company and the current implemented functionalities which will help in deciding what to add, remove or which part of the system to enhance to reflect the modern way of carrying out those processes.

Chapter Three primarily covers the analysis and design of the system, which are noted after intensively and carefully chosen the development methodology and how the methodology would be approached. In the analysis part, combination of data gathering techniques such as interview and observation were used to gather the data, followed by the modelling that depicts the logical model of the system and the subsequent design that shows the physical design of the system.

Chapter Four comprises the implementation of the system, the programming language used, test cases i.e. the functionalities tested and some errors encountered during the development, those that are resolved and those that are not.

Finally, Chapter Five defines the project evaluation, conclusion, further studies as well as recommendation. Evaluation are literally the assessment of the project with respect to its objectives and subjective, followed the expected further studies then at the end the conclusion.

## CHAPTER 2: LITERATURE REVIEW

### Introduction

So many car rental management systems that have been developed, and all these car rental systems are aiming at offering reliable services which can be accessed by customers at any time regardless of the location. This chapter contains the literature review of the car rental management system to be developed. This chapter will also review the existing systems that are similar to the car rental management system. References are made to sources from the internet.

### Historical Overview

A car rental is a company that rents automobiles for short periods of time, generally ranging from a few hours to a few weeks. Car rental agencies primarily serve people who require a temporary vehicle, for example, those who do not own their own car, travelers who are out of town, or owners of damaged or destroyed vehicles who are awaiting repair or insurance compensation.

The earliest known example of cars being offered for rent dates back to 1906 [1]. The German company Sixt was established in 1912 under the name Sixt Autofahrten und Selbstfahrer (Sixt Car Cruises and Self Drivers) [2]. Joe Saunders of Omaha, Nebraska first started with only one borrowed Model T Ford in 1916, but by 1917, his Ford Livery Company was renting out 18 Model Ts at 10 cents per mile. The company name became *Saunders Drive-It-Yourself System* and then *Saunders System*. By 1926, Saunders had expanded to 56 cities. Saunders' company was bought by Avis in 1955 [3]. An early competitor to Saunders was Walter L. Jacobs, whose Chicago-based *Rent-a-Car* opened in 1918 with twelve Ford Model T [4]. The company was bought in 1923 by John Hertz. In Britain, car rental started with Godfrey Davis, established in 1920, and bought by Europcar in 1981. (Walter L. Jacobs, 88; Rent-a-Car Pioneer, 2015) The sector expanded rapidly in the US; in 1926, the American Driveurself Association assembled over 1200 delegates in Chicago [1]. The growth in travel after World War II led to the establishment of several well-known international companies, including National Car Rental (1947), Europcar (1949), Enterprise Rent-A-Car (1957), Thrifty Rent A Car (1958), and Budget Rent a Car (1958). (Walter L. Jacobs, 88; Rent-a-Car Pioneer, 2015)

### Related Work

The existence of the internet and the World Wide Web in the mid-1980s and late-1980s respectively resulted in an exceptional revolution that inspire provision of car rental related services effectively and efficiently. Below are some related systems.

#### Car Rental System

The project is designed to help people utilize transport effectively [10]. In recent times cars have become most convenient modes of transportation. Sheth, Jay [10] said “our car rental system helps in making this an easier, hassle-free and enjoyable experience to acquire and use a car as per ones needs”. A person can book a car specifically for his travel time, co-travelers and the nature of travel. The rental system traverses from designing a database to understanding business concept and above all to make this easy to adapt system for various travelling needs [10].

Sheth, Jay [10] used the following entities which made up the car rental system database:

* + - * People
      * Customers
      * Agents
      * Employees
      * Car\_details
      * Car\_bookings
      * Extra\_driver
      * Location
      * Insurance
      * Bill\_generate
      * Payment

Shath, Jay [10] described the following database entities as:

* The **people** table stores information about all people associated with the travels system which includes employees, agents and customers. In this entity people id is the primary key. The other attributes are first name, last name, street address, state, zip code, home, cell and email.
* The **customers** table consists of all the information of travelers using Rental service. The customer id(CID) is a primary key. People ID is reference key here which references to people table. The other attributes of this entity are identification type and itinerary id which is the foreign key references to Itineraries table.
* The **Agents** table stores all the data about the agents who setup cars for the renters and their commission percentage. Also notes the location where they work as a foreign key.
* In the **Employees** table, we have all of the employees working in the office, not including on- field agents, car mechanics, etc.
* The **Car\_details** entity keeps the records for all of the available cars. It has car id as primary key. This entity also includes car name, Vin, seating capacity of car, production year. It also stores the price to rent the car.
* The **Car\_bookings** table keeps the records of the cars booked by the travelers at the specific locations. The booked car is added to this table with its car id and itinerary id of the person who booked the car.
* **Extra\_driver**; This is a weak entity derived from the customers table which has customer id as a foreign key which references to the customers table. This entity keeps the record of the person who might drive the rented car with the customer. Driving license number will also store of that person.
* **Locations** are either a car rental distribution center which holds rental cars waiting to be rented, or an office where agents work. This is distinguished by attribute Type.
* **Insurance** table has Insurance Type as primary key which keeps record of insurance type. This table has other entities like collision coverage which stores amount of Collison, Body coverage will cover amount of body damage of car and medical coverage will cover medical issue with customer in accidents.
* **Bill** table holds the bill for the customers. This table has bill number as primary key. Rent per day and rented days are attributes for counting bill amount which shows amount customer needs to pay.
* **Payment** entity has payment id as primary key and bill number is foreign key to get bill details from bill generate. And some other attributes like card number, expiry date, cvv are for the customer who wants to pay by credit card.

Below is an Entity Relationship Diagram for the car rental system.

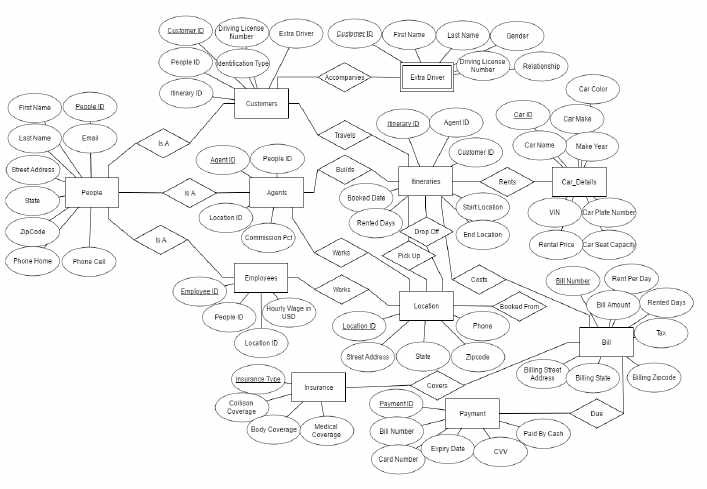


Figure 1. 1 **Entity Relationship Diagram**

* + 1. **Development of Car Rental Management Information System for Avis Indonesia** The information system is designed to more closely manager’s needs and the system is set up as major computer application area [9]. The Management Information System (MIS) as a computer- based system makes information available to users with similar needs [11]. Manager used the output information. The earlier studies shown that MIS could use to manage car rental, expected to accelerate as well as archiving services to customers better and safer, making it easier when required at any time [12] [13]. The online implementation of management information system provides and supports the customers for reservations, assist management in knowing rental car inventory at a specified time, to process transactions between branches car rental, transportation processing, which supports satisfactory service to customers and support the company’s operational process [14]. Web-based car rental information system increases the customers, and help promotion [15]. The aim of the research is solving the problems that occur in Avis Indonesia. The method that was used during the development of system was Software Development Life Cycle (SDLC) which is a series of processes or phases that is used in designing, developing and testing high quality software.

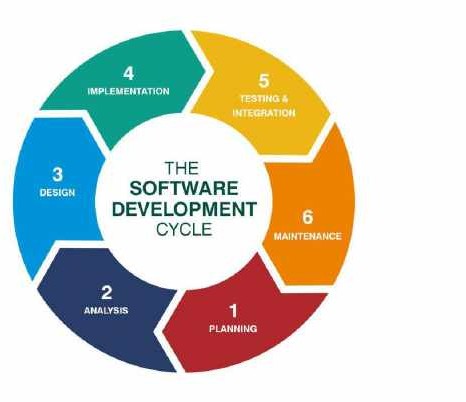
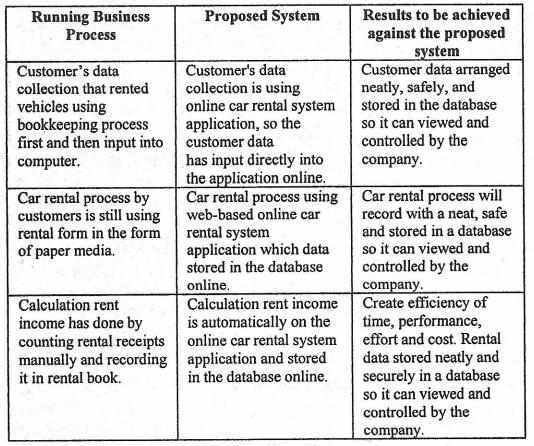


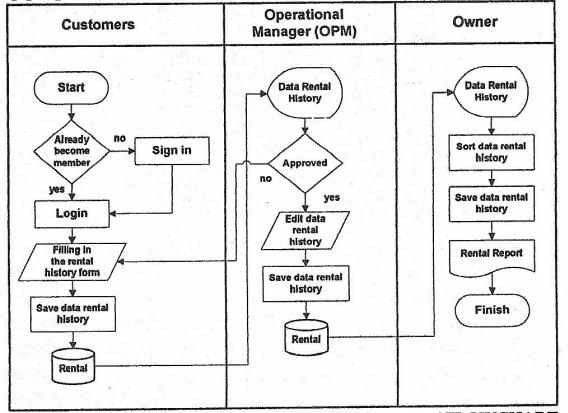
Figure 1. 2 **Software Development Life Cycle**

1. Planning Phase: This is the first phase for the researchers when developing an information system which is to find the core problems and constraints that occur when running the system, and to formulate goals and system development that focuses on online car rental system. In prior planning stage, researchers observe and collect data in Avis Indonesia [14].
2. Analysis Phase: The researchers analyze the company’s management, workflow, look for problems that occur within the company, car rental procedures and car rental data processing. System analysis aim is to find the ideal form of application a researcher will develop by taking into account of various issues and needs that exist on the system as specified in system planning [9].



#### Figure 1. 3 System Comparison Analysis for Car Rental Management Information System of Avis Indonesia

1. Design Phase: The researchers used several tools to create system design, i.e. process design, flowchart for owner and area manager, and Data Flow Diagram (DFD) for the car rental system.[18] [19] [20].



#### Figure 1. 4 Flow Chat for Car Rental Management Information System of Avis Indonesia

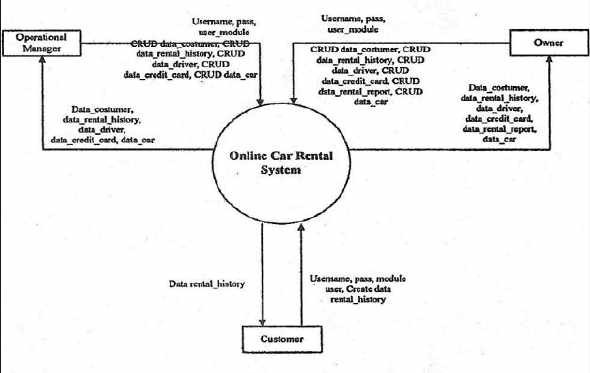


Figure 1. 5 **DFD LEVEL 0**

Database Design: after designing the system, researchers design the database using the tool of Entity Relationship Diagram (ERD) that describes the relationship between entities that exist in DFD [21] [22] [23].

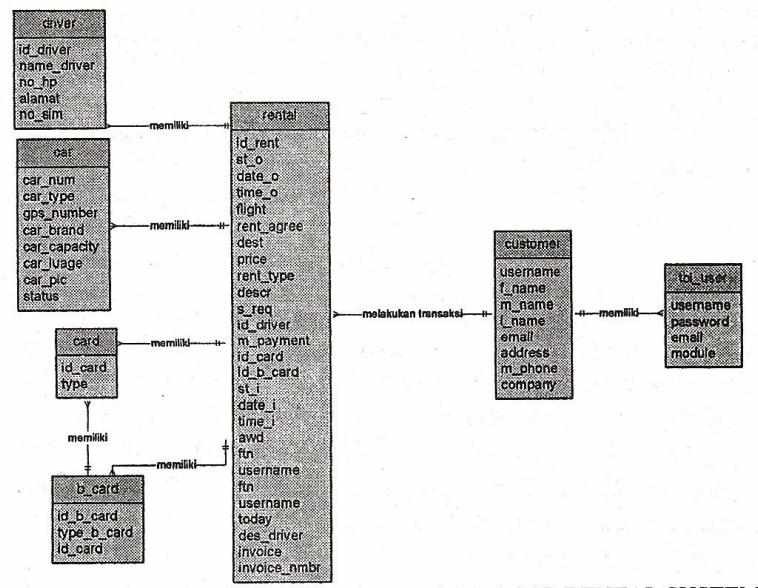


Figure 1. 6 **Entity Relationship Diagram Car Rental System**

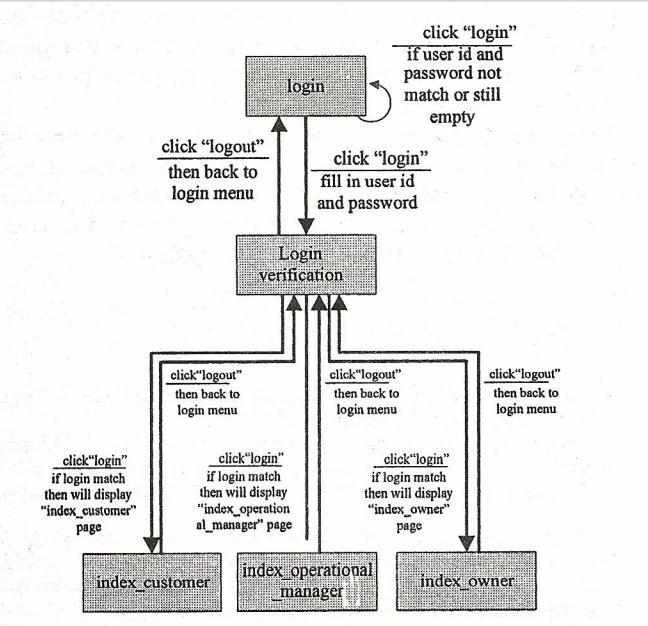


Figure 1. 7 **State Transition Diagram**

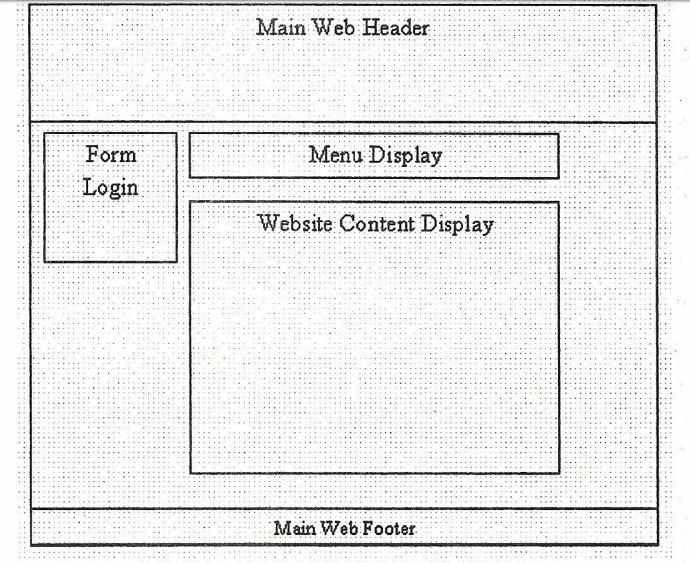


Figure 1. 8 **Main Web Interface Design**

1. Implementation Phase: The next phase is implementation. Researchers will plan to interpret or translate the system application design into programming language that can be understood by computer system so application can run and be used [9]. Researchers used software and programming language like PHP, MySQL for database and framework in visual web implementation displayed [24] [25] [26] [27] [28] [29].
2. Testing and Integration Phase: Before the program is tested, the program must be free from error, free from errors is a necessary testing to find errors that may occur as in the language errors, logic errors and error analysis program [9]. Tests that are performed on the system Car Rental System application include the entire menu of programs, processes applications, and preparing reports [9].



Figure 1. 9 **Car Rental System Test Results**

#### Online Car Rental System with Web-Based and SMS Technology

Car rental or car hire agencies are private companies that provide short time leasing vehicles for a specified time with a fee to their customers [17]. Osman, M., Muhd, Z., Nurzaid, Paidi, Z., & Sedek, K. [17] show that In Malaysia, car rental service increasingly becomes the preferred option for most people, especially among students in campuses and universities. This occurs because not all students can afford having their own vehicle and perhaps the university bus service doesn’t always help. Besides, the raising taxi fares and inconsistent bus arrivals in Malaysia continue to discourage people from taking up the public transport. Therefore, car rental service continues to grow in Malaysia, hence it required an improvement and good monitoring system.

The web-based car rental system integrated with SMS technology has a very user-friendly interface. By using this system, employees can manage bookings, payment, vehicle issues and

SMS notification to the customers within a few clicks only [17]. The new data can be added or an existed data can be edited or deleted too by administrators. Thus, there is no delay in the availability of any information, whether needed, can be captured very quickly and easily. For security purposes, all customers need to create a new account before logging in or he/she can log into the system with his/her created account before they can make a reservation for a car. Then, the customer will be notified the availability of the car reserved through SMS. This system becomes very helpful for employees, administrator and customers [17].

The Software Development Life Cycle (SDLC) was used to develop the web-based system and SMS notification. SDLC is a framework that describes all activities and processes in a software development project. The process is associated with the waterfall model which consists of five phases such as planning, analysis, design, development, and evaluation.

#### Fasim Rentals

Fasim rentals is a web-based system that was developed by Mike Aric who was a student of Ahmadu Bello University Zaria [5]. According to Mike [5], the system was developed for a company named “Fasim rentals”. The methodology used in developing the system was incremental methodology which helps in developing software quickly during software life cycle. In addition to the methodology used, java programming language was the language used to develop the system. Fasim rentals were running all their organization activities manually before they began using the system. Some features of the system include managing customer vouchers, transaction history, reports, reservation, online payment and more [5].

#### Suria Car Rental & Tour

Suria Car Rental & Tour Sdn TOUR Bhd (SCR &) has been committed in offering exceptionally great Budget Car Rental service and comfortable Malaysia Car Rental Services for both domestic and international travellers [6]. SCR focus on delivering customer-oriented services. In other word, they focus on customers’ needs and rent a vehicle tailored to their needs. Their philosophy is reflected in the services they offer, whereby they provide Budget Car Rental and Cheap Car Rental in Malaysia with excellent car condition and high-quality service which not only meeting customers transportation needs yet save their pocket. SCR have an extensive choice of Budget Car Rental solution and services in Malaysia including single journey to/from the airport within Klang

Valley, daily chauffeur rate, in Kuala Lumpur or out of the town, economical, family or luxury vehicles, and more![6]

In cooperation with reliable Strategic Partners, SCR also provide customized individual and corporate services including incentive tour, MICE and group outing both in Malaysia and overseas. Their customized service in the travel industry has benefited their clients from various industries covering government, large corporation, small and medium enterprises and individual clients (Who we are, 2020). In order to achieve the successfulness of theirr services, SCR and its Strategic Partners in Malaysia and oversea always ensure its daily operation being carried out professionally and reliable, which in turn, allowing them to maintain the quality services standard that is beyond industry benchmark over the years.

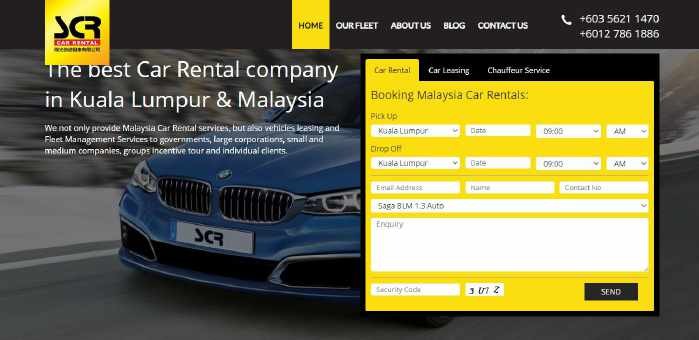


Figure 2. 1 **The home page of SCR & Tour website**

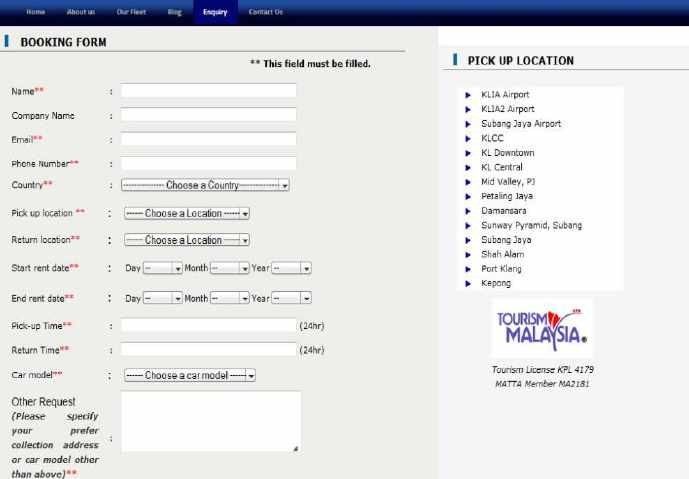


Figure 2. 2 **The booking form of SCR & Tour**

#### WS RENT-A-Car

WSRAC is licensed by the Ministry of Culture, Arts, and Tourism to carry out a car rental business in Malaysia.Their head office is located at Kuala Lumpur [7]. They offer car rental and car leasing services in major cities and airports throughout Malaysia.They offer wide choice of vehicles, from economical to luxury cars, vans, MPV and SUV for their customers to choose the one and most appropriate for their trip and travelling comfort.

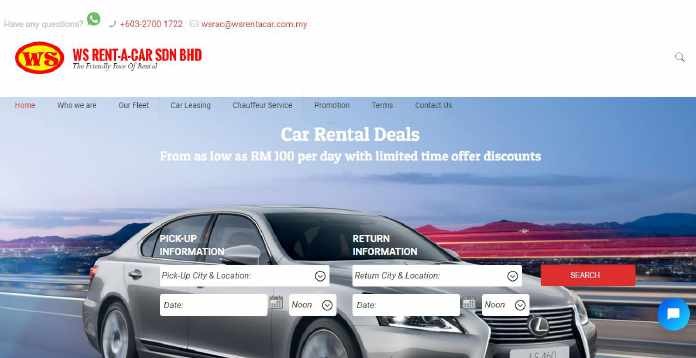


Figure 2. 3 **The home page of WS-RENT-A-Car**

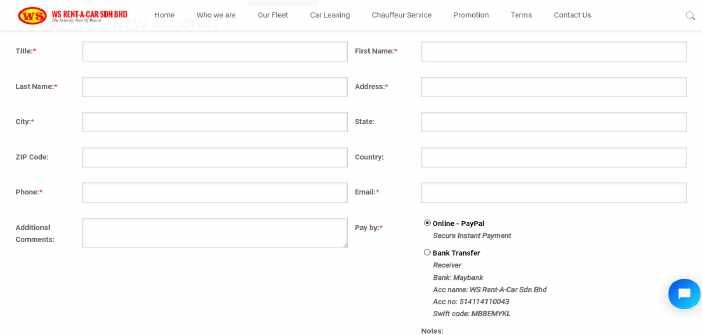
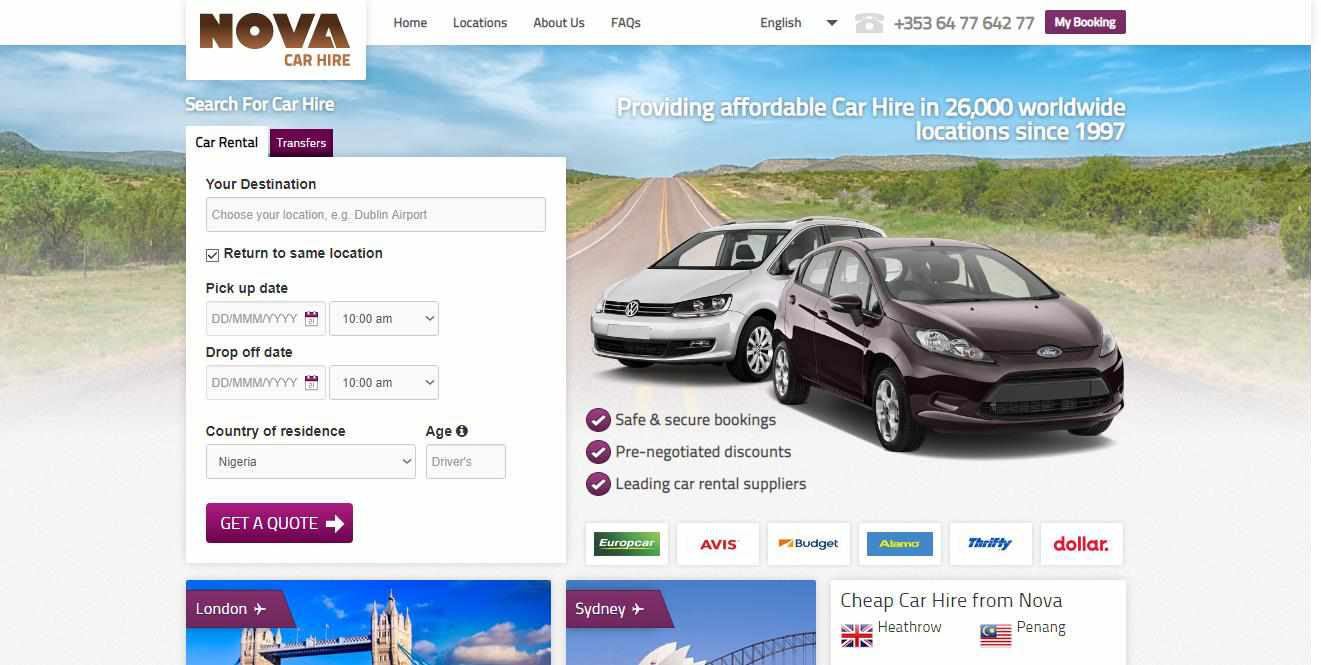


Figure 2. 4 **The booking form of WS-RENT-A-Car**

#### Nova Car Hire

The website novacarhire.com was rebuilt and relaunched in 2012 and now offers the best car rental consumer experience in the market [8]. They have grown from strength to strength over some years due to their focus on their customers. Their continual process of change and innovation ensures

they offer their customers the best booking and car hire experience possible. Nova has had over 1.5million satisfied customers and sold over 12million rental days across the world.



#### Figure 3.1: The home page of Nova Car Hire



Figure 2. 5 **The booking form of Nova Car Hire**

### Summary

This chapter shows that there is increase in the development of car rental industry, many companies are established and digital innovation has extremely changed the way car rental companies are running their businesses. In addition, customers can now rent a car without suffering from manual registrations, booking and payment. Car rental companies also can keep track of all their data without encountering any problem. Customer can also drop-off a car at any location, provided there is a branch of a particular car rental company in that location.

## CHAPTER 3: REQUIREMENTS ANALYSIS AND DESIGN

### Overview

This chapter describes the analysis and design of car rental management system and the type of the methodology used in the system development. The requirements of the car rental management system are identified in this chapter, which includes functional and non-function requirements. The tools that are used when developing the system are identified, which consists of hardware and software tools as well as the technologies used.

In addition, this chapter describes the model of the system which is a simplified, complete, and consistent abstraction of the system, that is created for better understanding of the system using DFDSs and UML diagrams. Finally, this chapter displays the system design; the system architecture, the user interfaces, and the database design which are the physical view of the system.

### Proposed Methodology

Methodology in information system development is the process of dividing software development work into distinct phases to improve design, product management, and project management. Some of the methodologies used in system development include agile development methodology, water fall development methodology, rapid application development methodology, prototyping methodology and spiral development methodology. Each of these methodologies are chosen based on the nature of the system to be developed or the information technology project to be carried out and the stakeholders of the system.

However, in this project, prototyping methodology is chosen, because prototyping methodology is one of the most important software developments models, it’s the process of building prototype (which is the first design of the system). Prototyping methodology allows you to build the first design of your system which will be shown to the system stakeholders in order to go through the prototype and find their likes and dislikes based on how the final system will look like. Stakeholders will constantly be going through the prototype until an acceptable prototype is achieved, which is going to be used to develop the final application.

More so, prototyping methodology is chosen because stakeholders are actively involved in the system development. Using the system prototype, the stakeholders will also have a full knowledge and understanding of the system to be developed, how the system works and also errors will be detected as early as possible.

### Approach to Chosen Methodology/Methods

Prototyping methodology has six software development life cycle (SDLC) phases, at the first stage, all the system requirements are gathered from the stakeholders by interviewing them, then the design of the system is made, which shows a preview of the system to the stakeholders. The followed approach after design is building the prototype. The prototype is built and its presented to the stakeholders for initial evaluation in order to help find the strength and weakness of the prototype.

The prototype is refined according to the stakeholder’s feedback and suggestion, if the initial prototype meet their needs then next prototype will be built and presented to them otherwise requirements will be clearly gathered and reviewed, then another prototype will be built for them to evaluate. Repeatedly, that is how the chosen methodology has been approached till the end of the project.

### Tools and Techniques

The following tools are used throughout the project

* A computer with 2.5Ghz of processor, 8GB of RAM, and 500GB of Hard drive
* UML Diagrams for the analysis and modeling of the system
* Microsoft office programs for necessary documentations and visualizations
* MySQL Workbench
* Erwin Data Modeler
* Microsoft Visio
* Visual Studio as IDE
* IIS Web Sever
* SQL Server Express
* Visual Paradigm

Technologies used in this project are as follows

* Asp.Net Core with C# programming language
* Bootstrap CSS
* JavaScript

### Requirement Analysis

At the end of the requirement analysis process, the expected deliverable is System Requirement Specification (SRS), which describes the behaviors and features of the system. It also contains the list of requirements of the system as well as the diagrams.

Requirements list are the list of functions that a system must possess. Requirements are of two categories, which are functional and non-functional. Functional requirements are the requirement that the stakeholders need from the system, how is the system going to be operated and what the system should have. Non-functional requirements are the requirements that specifies the criteria that can be used to judge how the system operates.

The diagrams, which are also known as system model are the abstraction of the system. Each model is presenting a particular view of the system. The diagrams also show the relationship between the system and the external entities.

### Requirement Specifications

The two tables below contain the functional and non-functional requirements

Functional Requirement Specifications Table 2. 1 Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Req. No.** | **Description** | **Type** |
| R-1 | A customer should be able to register with google  account | Functional |
| R-2 | A customer should be able to view the details of a  particular car | Functional |
| R-3 |  | Functional |
| R-4 | The system should show available cars to the customer  during reservation | Functional |
| R-5 | The system should allow customer to delete reservation | Functional |
| R-6 | The system should display the summary of successful  reservations | Functional |
| R-7 | Admin should be able to login to the system using  his/her username and password | Functional |
| R-8 | The system should allow new user account to be created  by the Admin | Functional |

|  |  |  |
| --- | --- | --- |
| R-9 | The system should allow users to logout | Functional |
| R-10 | Customer can select a car from the car list | Functional |
| R-11 | The system should allow users to search a specific car | Functional |
| R-12 | The system should allow admin to modify or update any  car information when there is need for that | Functional |
| R-13 | Customer should be able to display all the lists of cars | Functional |
| R-14 | The system should be able to save all changes made by  admin | Functional |
| R-15 | The system should be able to print the summary of all  successful rents | Functional |

Non-Functional Requirement Specifications Table 2. 2 Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Req. No.** | **Description** | **Type** |
| R-1 | Transactions should be securely made and users data  should be protected from attacks | Security |
| R-2 | The system should be trusted and relied by the users | Reliability |
| R-3 | The system should be easy for the customers to use it | Usability |
| R-4 | The system should run on any hardware with any kind of browser. It should not conflict with other processes  within these environments | Portability and  compatibility |
| R-5 | System should keep running when its launched unless  there is an intentional shutdown of the system | Performance |

### System Design

Use case Diagram

Below use case diagrams show the interaction between the external entities and the system

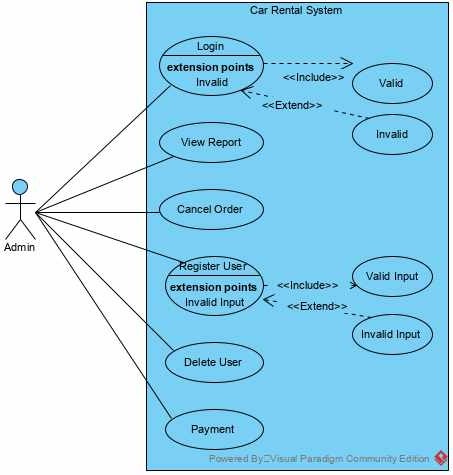


Figure 2. 6 **Admin use case diagram**

This figure models the interaction between the admin and the system. In the diagram, from the left is an admin actor. Within the rectangle which represents the system are what are called use cases. The use cases are basically actions that an actor can perform in the system. From top left within the system is login use case, before login in to the system, the details for logging in need to be valid, which is represented by the valid use case at the top-left corner.

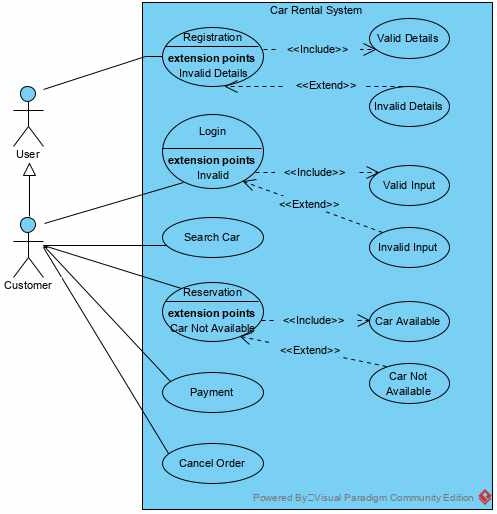


Figure 2. 7 **Customer use case diagram**

This figure models the interaction between the customer and the system. Within the rectangle, all the use cases are the actions that the customer can perform with system.

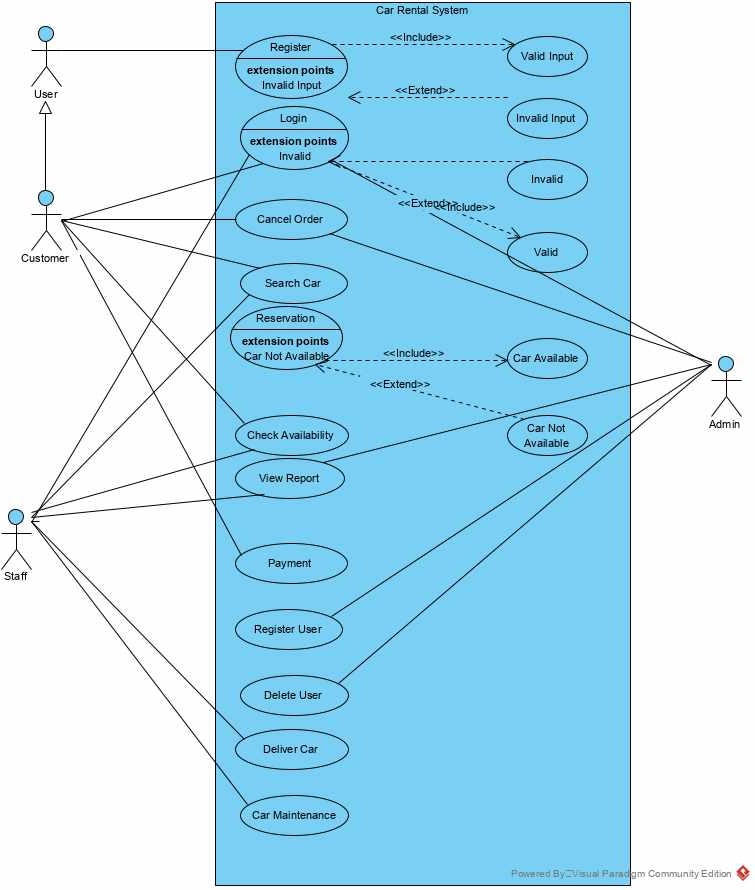


Figure 2. 8 **Car rental system use case diagram**

This figure shows the overall actions that can be performed by the customer, staff, and the admin, which are represented by what are called use cases. From left-side of the system are customer and staff actors, while from the right side of the system is an admin actor.

Class Diagram

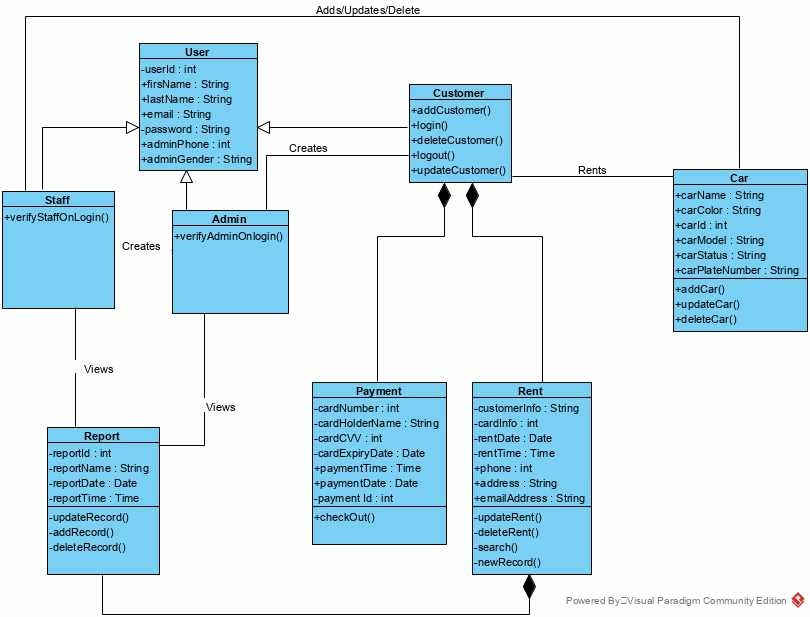


Figure 3. 1 **Class Diagram**

This figure represents the class diagram of the system which shows the classes, including their attributes and operations. It also shows the relationship between the entities. From the top-left is a user class which has three subclasses Admin, Staff and Customer that inherit the attributes of the user class. The empty arrows that mapped from the three subclasses to the superclass show the

inheritance. Every line that is drawn from a single class to another class in the diagram is called Association. The shaded arrows from the payment and rent class to Customer class show that payment and pent cannot exist with Customer, likewise rent and report.

Database Design



Figure 3. 2 **Entity Relationship Diagram (ERD)**

The Car Rental System ERD above shows the entities and the relationship between them. Each and every entity has its own attributes. From the left, there is a many-to-many relationship between Customer entity and Car entity. In this case, a Customer\_has\_Car entity is introduced in-between the two entities which is called a Bridge Entity to solve the many-to-many relationship. Each and every entity in the ERD above has a primary key; which is an attribute or set of attributes that uniquely identify a specific instance of an entity

Dataflow Diagram (DFD)

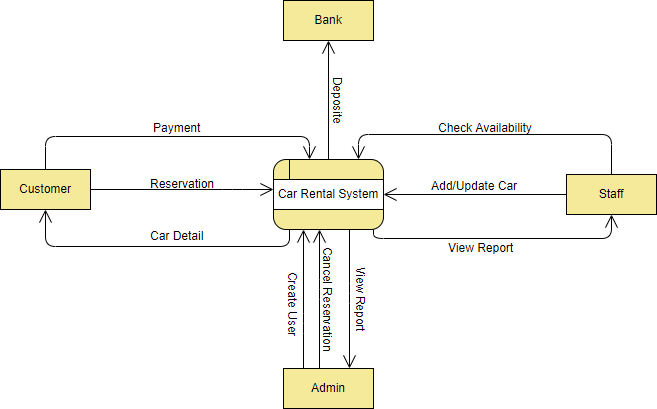


Figure 3. 3 **Context Level DFD**

The Context Diagram above shows the overall view of the system, the data that goes in, and how it flows to other entities. From the left, the square symbol represents a customer entity which can help make reservation from the system. The arrow that is labeled Car Detail is the processed data that comes out of the system depending on the logic executed. The square with rounded corners at the center of the diagram represents the system.

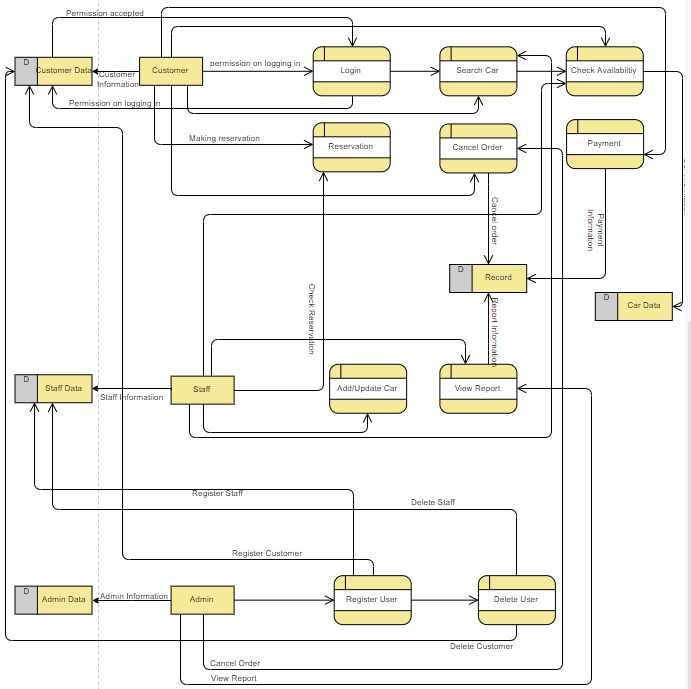


Figure 3. 4 **Level 0 DFD**

The figure above shows the datastores, entities and processes of the system. The squares with rounded corners are the processes, the rectangles shaded by the left are the datastores, and the rectangles with no shade are entities. The arrows that goes to the datastores from the entities shows that their details of logging in to the system are saved in the datastores.

Activity Diagram

Below activity diagrams model the workflow within a system with the use of activity, action, decision, etc.

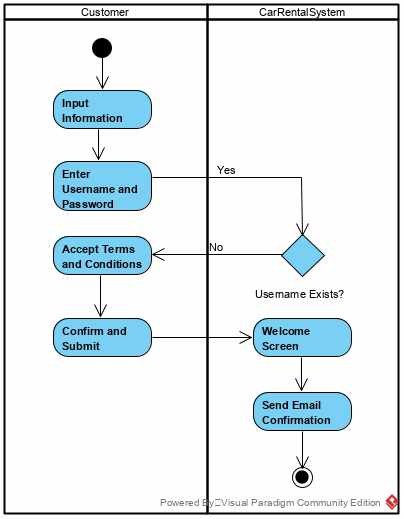


Figure 3. 5 **Registration Activity Diagram**

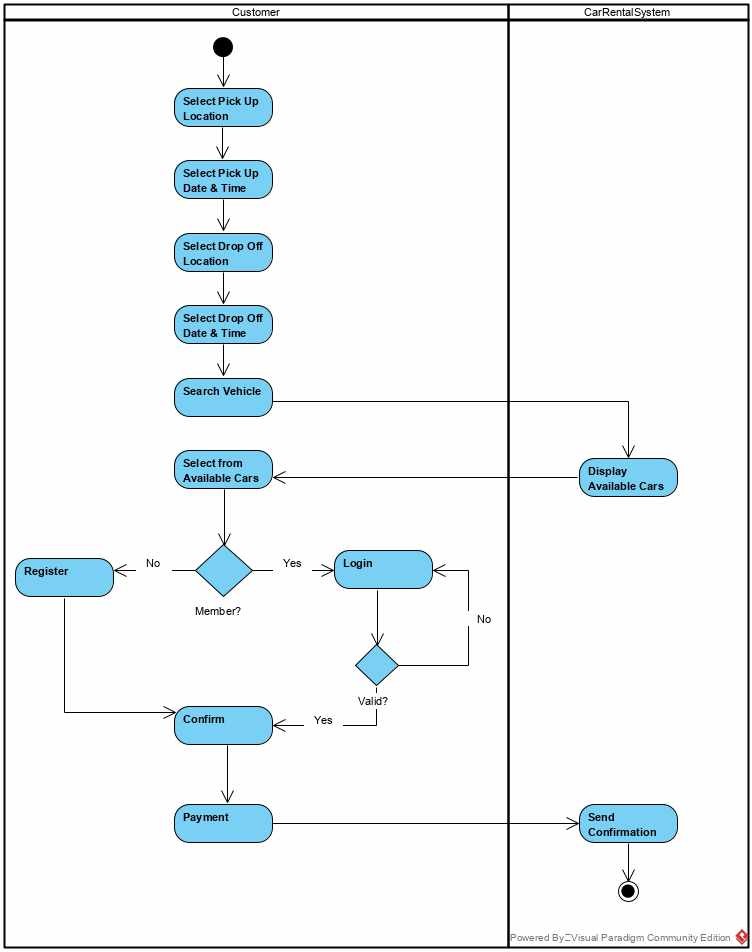


Figure 3. 6 **Car Reservation Activity Diagram**

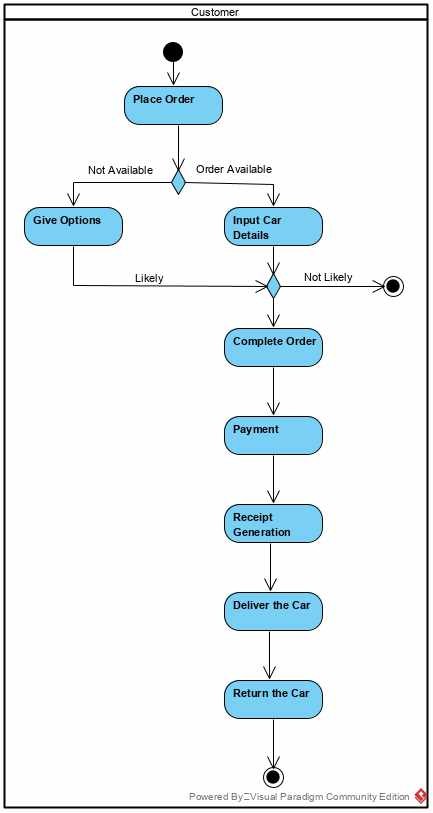


Figure 3. 7 **Placing Order Activity Diagram**

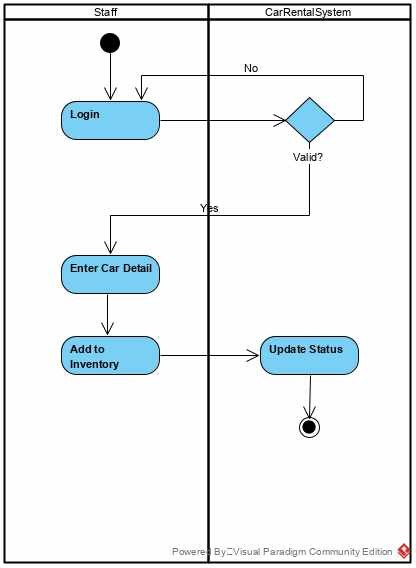


Figure 3. 8 **Adding New Car Activity Diagram**

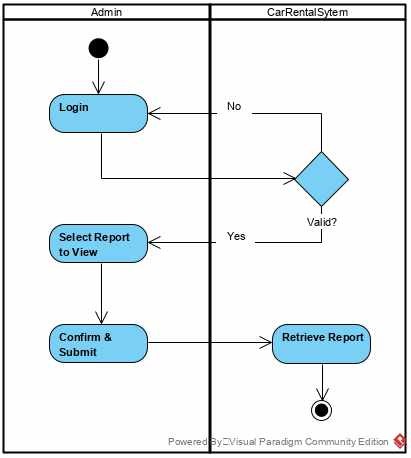


Figure 4. 1 **View Report Activity Diagram**

Application Architecture

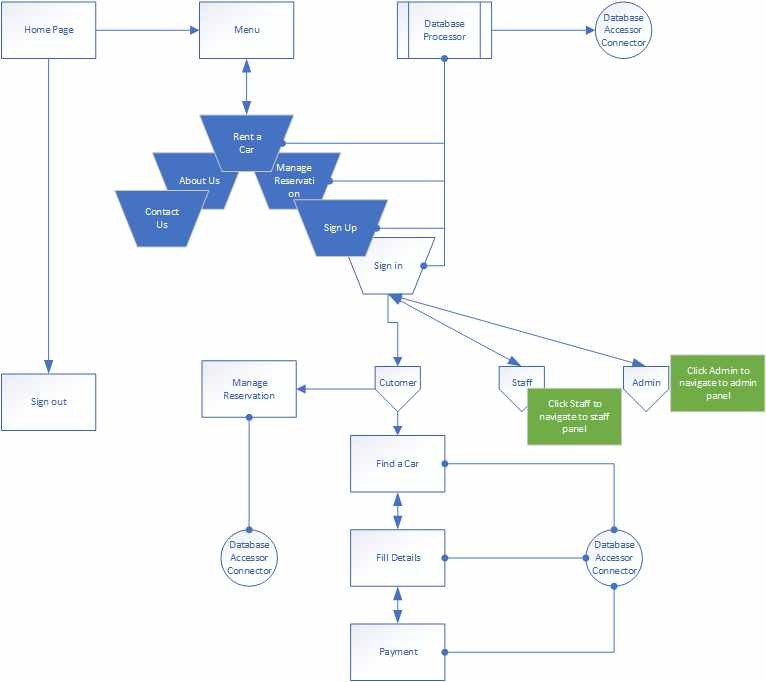


Figure 4. 2 **Car Rental System Application Architecture**

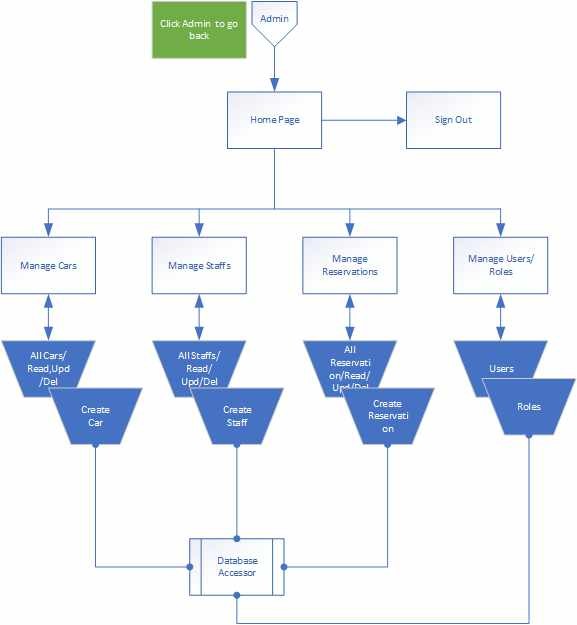


Figure 4. 3 **Admin Application Architecture**

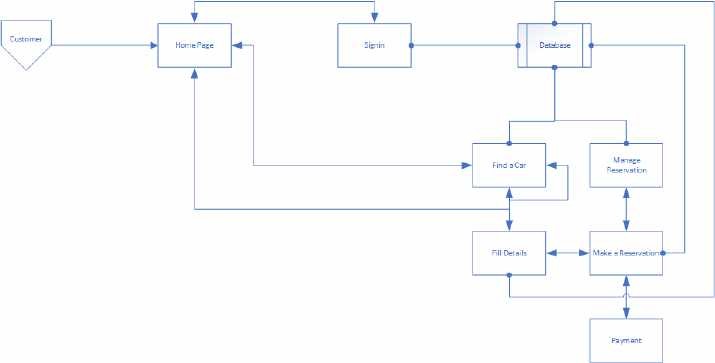


Figure 4. 4 **Customer Application Architecture**

The application architectures above give the high-level representation of the system. It also represents the sub-process, component, and navigation to other parts of the system.

### Summary

This chapter shows the analysis and design of the system requirements which are functional and non-functional requirements, the models and designs of the system that were developed using different kinds UML diagrams and DFDs. The application would also be developed using prototyping methodology.

## CHAPTER 4: IMPLEMENTATION AND TESTING

### Overview

This chapter talks about the implementation and testing of the car rental system, the technologies and tools that were used when implementing the system, the problems and errors encountered during the implementation, and how the problems were solved. This chapter also describes the implemented and functionalities of the system.

### Development tools and technologies

Listed below are the tools that ware used during the implementation of the project

* + A laptop with 2.5GHz clock speed, minimum of 4GB RAM and minimum of 500GB HDD.
  + Visual Studio 2019 as IDE
  + Microsoft SQL Server Express
  + IIS web sever Technologies used are
  + Asp.net core MVC
  + JavaScript
  + jQuery
  + Bootstrap CSS

### Implementation Problems

During the implementation of this web application, problems were encountered, but were nicely solved. The problems encountered has to do with both the font-end and back-end implementation.

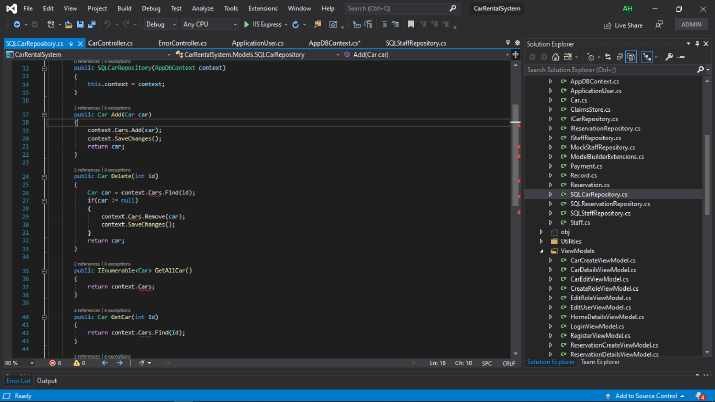


Figure 4. 5 **First Implementation problem**

The first problem was encountered when implementing the back-end, where the property to query and save the instances of car class was missing.

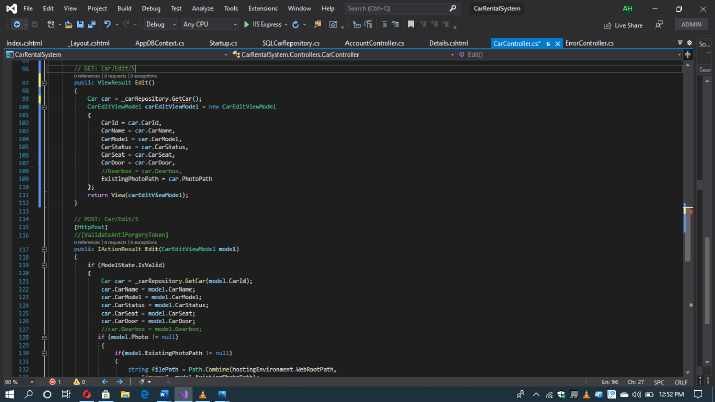


Figure 4. 6 **Second Implementation problem**

The second problem arises when there is no argument given that corresponds to the required formal

parameter “Id” of get car method in ICarRepository.



Figure 4. 7 **Third Implementation problem**

The third problem encountered was in the startup class where the required MVC services to the dependency injection container in the configure services method was not added.

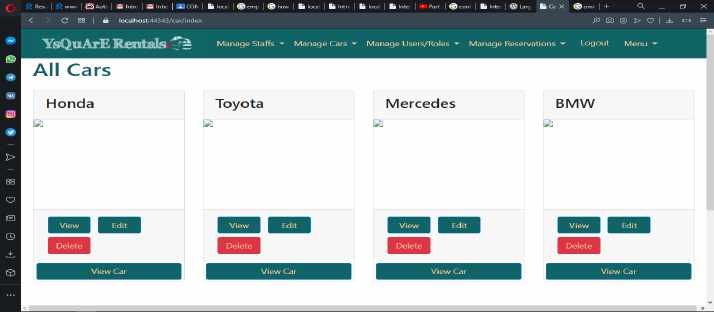


Figure 5. 1 **Fourth Implementation problem**

The fourth problem encountered was in the index view of the cars which happens as the result of omitting forward slash of the images folder.

* 1. **Overcoming Implementation Problems**

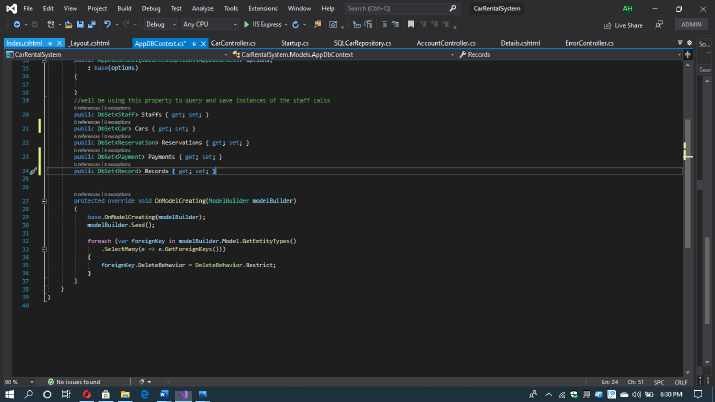


Figure 5. 2 **First problem encountered solution**

The above figure shows the solution to the first problem encountered

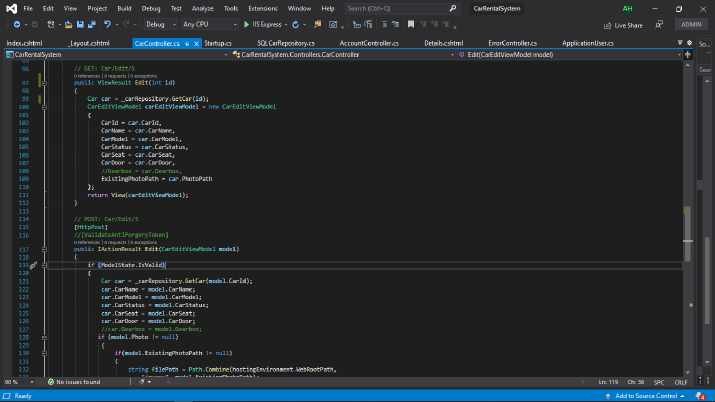


Figure 5. 3 **Second problem encountered solution**

Solution to the second problem encountered

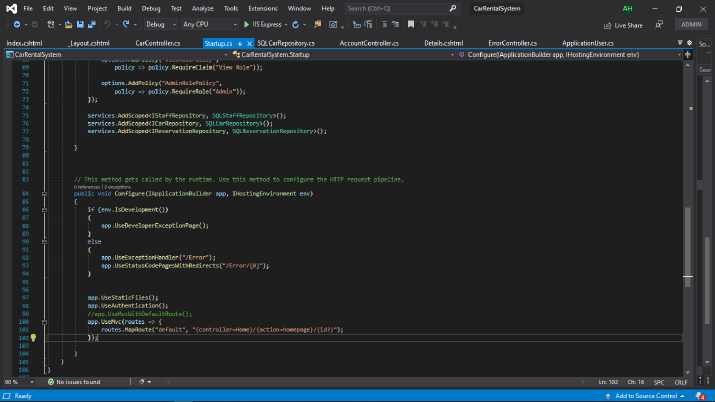


Figure 5. 4 **Third problem encountered solution**

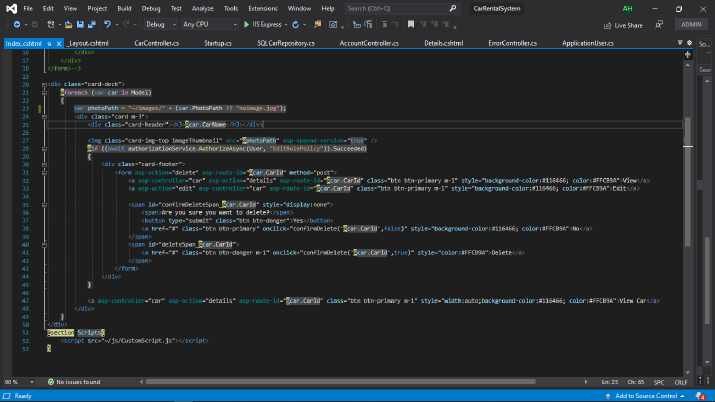
This figure shows the solution to the third problem encountered

Figure 5. 5 **Fourth problem encountered solution**

Solution to the fourth problem encountered

### Testing

This section shows the test carried out in order to check whether the application is error free or not. Test plan was developed and used which contains all the system functionalities that are going to be tested in case of any error. System testing also helps to check whether the system meets its requirements or not. MSUnit framework is used to test all the functionalities to confirm if there is error.

Test Plans

Table 3. 1 Test Plans

|  |  |
| --- | --- |
| Test Suite ID | Car Rental System release 1.0 (MTP) 1.0 |
| Test Case ID | #1 |
| Test Case Summary | The system should allow user to sign up with or without google  account |
| Expected Result | After signing up, the application should redirect to home page |
| Actual Result | After signing up, the application should redirect to home page |

|  |  |
| --- | --- |
| Status | Pass |
| Executed by | Yasir Ado Hassan |
| Date of Execution | 4/09/2020 |
| Test Environment | Opera |

Table 3. 2

|  |  |
| --- | --- |
| Test Suite ID | Car Rental System release 1.0 (MTP) 1.0 |
| Test Case ID | #2 |
| Test Case Summary | Does the system allow user to make reservation |
| Expected Result | After making reservation, the application should navigate to  reservation details |
| Actual Result | After making reservation, the application should navigate to  reservation details |
| Status | Pass |
| Executed by | Yasir Ado Hassan |
| Date of Execution | 4/09/2020 |
| Test Environment | Opera |

Table 3. 3

|  |  |
| --- | --- |
| Test Suite ID | Car Rental System release 1.0 (MTP) 1.0 |
| Test Case ID | #3 |
| Test Case Summary | Admin should be able to assign roles to users |
| Expected Result | Users should be able to access some functionalities based on  their roles |
| Actual Result | Users should be able to access some functionalities based on  their roles |
| Status | Pass |
| Executed by | Yasir Ado Hassan |
| Date of Execution | 4/09/2020 |
| Test Environment | Opera |

Table 3. 4

|  |  |
| --- | --- |
| Test Suite ID | Car Rental System release 1.0 (MTP) 1.0 |
| Test Case ID | #4 |
| Test Case Summary | Does the system allow administrator to manage users, roles,  cars, and reservations |
| Expected Result | Admin should be able to manage roles, users, cars and  reservations |

|  |  |
| --- | --- |
| Actual Result | Admin should be able to manage roles, users, cars and  reservations |
| Status | Pass |
| Executed by | Yasir Ado Hassan |
| Date of Execution | 4/09/2020 |
| Test Environment | Opera |

Test Traceability Matrix

Table 3. 5 Test Traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req. No | Description | Priority | Test  Case | Test Date | Test Result |
| R-1 | The system should  allow user to register with google account | HIGH | 1 | 4/09/2020 | PASS |
| R-2 | The system should allow user to make  reservation | HIGH | 2 | 4/09/2020 | PASS |
| R-3 | The system should allow admin to assign  roles | HIGH | 3 | 4/09/2020 | PASS |
| R-4 | The system should allow administrator to manage users, roles,  cars, and reservations | HIGH | 4 | 4/09/2020 | PASS |

Test Report Summary

Table 3. 6 Test Report Summary

|  |  |
| --- | --- |
| Summary of Test Carried | Results |
| Tests carried out | 4 |
| Tests not carried out | 0 |
| Tests passed | 4 |
| Test failed | 0 |
| Percentage of tests carried out | 100% |
| Percentage of tests not carried out | 0% |
| Percentage of tests passed | 100% |

|  |  |
| --- | --- |
| Percentage of tests failed | 0% |

### Use Guide

The user guide describes how user is going to use the system with ease to accomplish his/her goal, including all the processes involved in operating the system. The user guide also serves as a short reference to some particular aspects of the system. The user guide for the application can be found in Appendix E.

### Summary

This chapter shows the implementation of the system, the tools and technologies used during the implementation, which are asp.net core MVC, bootstrap, jQuery, JavaScript, HTML and CSS. This chapter also covers the problems that were encountered during the implementation and how the problems were solved, test plans, test traceability matrix and test report summary as well.

## CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS

### Overview

This chapter covers the evaluation of the project, the project conclusion and the project recommendations. Firstly, it presents the objective assessment which is related with the aim of the project and also the context of the project which give the general conclusion of the project. This chapter also discusses the limitations of the project and the challenges encountered. Then, some of the future enhancements are going to be made as the project got online and started to be used. This chapter lastly contains the recommendations that are made.

As discussed in chapter 3, the methodology used in developing the software is prototyping methodology, because it is one of the most important software developments models, it’s the process of building prototype (which is the first design of the system). Prototyping methodology allows you to build the first design of your system which will be shown to the system stakeholders in order to go through the prototype and find their likes and dislikes based on how the final system will look like. Stakeholders will constantly be going through the prototype until an acceptable prototype is achieved, which is going to be used to develop the final application.

### Objective Assessment

The car rental system has accomplished some its aims, up to this point, the application: Allows users to register with the system

Allows users to login with the system

Enables users to register and login with their google account Allows the administrator to assign roles and claims for the users Enables users to reserve a car

Allows users to delete or modify their reservations

Can allow the administrator to add and manage users, cars, and records. Allows users to view the cars that are added onto the system

Allows user to print a particular page.

Enables users to search for a specific car in the system. Administrator can also search for a user and car.

Allows to search for a record in the system.

Yet, there are some functionalities that should be evoked, need to be designed, elicited and also implemented.

### Limitations and Challenges

This project is been developed for ysquare rentals which is a Nigeria car rental company based in Kano, therefore the project is currently limited within Nigeria, and most of the users will are Nigerians. Majority of the processes that will be carried out manually in the company are analyzed and implemented. So, it would be easy for them to use the system than any other person outside Nigeria. For a user to rent a car, he/she must sign in to their own account, and guest must create their own account.

During the analysis and implementation of this project, there are challenges that were encountered. When analyzing the requirements of the system, there was difficulty in understanding what exactly the users want, because they might not know what and how the system will look like. The challenges were addressed by the application of some skills which like; problem solving skills, communication skills as well as critical thinking skills to think, solve problems, and communicate with people. Challenges were also faced during the prototype development which did not fit the users; therefore, it has to be analyzed and designed again.

### Future Enhancements

Some enhancements are supposed to be made as this project is launched, it includes;

* Application should run on mobile phones

Nowadays, the number of mobile phone users is very high than computer users most especially in Nigeria. The application is going to be in android and iOS platforms and also developed in order to increase accessibility and inclusiveness.

* Staff Interface
* Geofencing and remote car deactivation

When geofencing is integrated in the application, it will use GPS or WI-FI to trigger a pre- programmed action when the mobile device owned by the customer enters or leaves a virtual limit set up around a geographical area.

* Car sales

Car sales functionality will be implemented to the system when this project got online; therefore, user can register with the system as a salesman and add the car he/she wants to sale. So, when this functionality is in place, customers can not only rent car, but buy cars from the system as well.

### Recommendations

The car rental company need to open branches in some other states within Nigeria. The project should have a cloud storage for backing up user’s data because the cloud storage can save costs, protects user’s data from ransomware or malware, regulatory compliance, and data tiering for cost savings. Also, the web-based car rental system should also be developed in form of mobile application, to help improve usability for mobile phone users.

## REFERENCES

* + 1. Car Renting… Its Development… And Future (2020). Retrieved from

<https://www.automotive-fleet.com/147063/car-renting-its-development-and-future>

* + 1. “History – The beginning of a moving tale”, Sixt (2020). Retrieved from

[https://about.sixt.com/websites/sixt\_cc/English/0/about-us.html#HISTORY](https://about.sixt.com/websites/sixt_cc/English/0/about-us.html%23HISTORY)

* + 1. “Joe Saunders, Omaha Innovators” (2020). Retrieved from

<http://www.douglascohistory.org/>

* + 1. “Walter L. Jacobs, 88; Rent-a-Car-Pioneer” (2020). Retrieved from

<https://www.latimes.com/archives/la-xpm-1985-02-08-me-4716-story.html>

* + 1. Mike, A. (2018). Design and development of online car rental system. Ahmadu Bello University Zaria.
    2. About Us (2020,) <https://www.scr.com.my/about-us/>
    3. Few words about us (2020). Retrieved from [https://www.wsrentacar.com.my/kl-car-rental-](https://www.wsrentacar.com.my/kl-car-rental-klia-klia2/) [klia-klia2/](https://www.wsrentacar.com.my/kl-car-rental-klia-klia2/)
    4. About us (2020). Retrieved from <https://www.novacarhire.com/about/about-us.htm>
    5. Waspodo, Bayu, Aini, Qurrotul and Nur, Syamsuri. (2011). Development of Car Rental Management Information Systems (Case Study: Avis Indonesia).

10. Sheth, J.. (2016). Car Rental System. 10.13140/RG.2.2.12716.87680.

1. Mcleod, R. and Schell, Jr. G. P.(2007). Management Information System, (10th Edition). India: Pearson Education, Inc.
2. Damayanti, Y. (2005). “Car Rental Information System Design using Microsoft visual basic version 6.0”. Gunadarma University.
3. Cahyanto, M. R.T. et al. (2010). “Car Rental Management Information System CV transport with Java SE and MySQL based OOP”
4. Yodiyanto. (2006). “Analysis and Design of Web-based Car Rental and Travel

Transportation Information Systems at PT”. Bina Nusantara University.

1. Sar, R. D. (2011). “Building Application System Car Rental Reservation and Payment

Online Web-based (Case Study in Th Rental Daras Corporation)”.

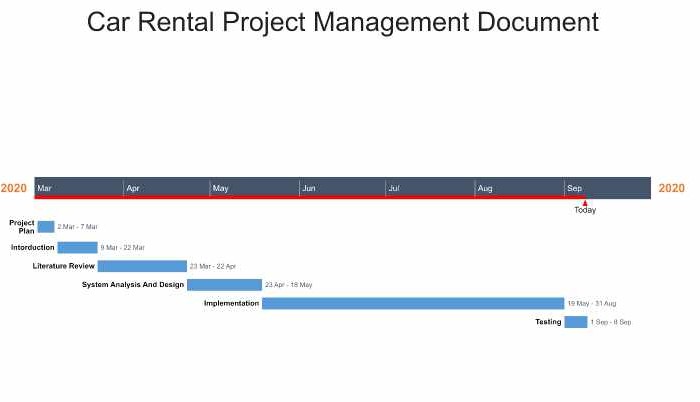
1. Avis Indonesia. (2009). “Journal Avis Indonesia”. Jakarta: Avis Indonesia.
2. Osman, M., Muhd, Z., Nurzaid, Paidi, Z., and Sedek, K. (2017). Online Car Rental System using Web-Based and SMS Technology.
3. Jogiyanto. (2001). Information System Analysis and Design Structured Approach, Theory and Practical Business Application (2nd Edition). Yogyakarta: Andi Yogyakarta.
4. Andri, K. (2004) Software Device Manipulation. Yogyakarta: Gava Media.
5. Kendall, K. (2006). System Analysis and Planning. Jakarta: Penerbit Indeks.
6. Connolly, T. and Begg, C. (2002) Database System: A practical Approach to Design, Implementation, and Management, (3rd Edition). England: Addison Wesley.
7. Fathansyah. (2000). Basis Data: CV. Informatics.
8. Ladjamudin, A. (2005). Analysis and Design of Information Systems. Yogyakarta: Science House.
9. Hakim, L. (2004). 150 Secretes and Tricks to Master PHP. Jakarta: Elex Media Komputindo.
10. Sidik, B. (2001). Web Programming with PHP. Bandung: Informatics.
11. Sidik, B. (2005). MySQL for Administrator Users and Web Application Development. Bandung: Informatics.
12. Siswoutomo, W. (2008). PHP Enterprise. Jakarta: Elek Media Komputindo.
13. Sudirman. (2003) Build Web Applications with PHP and MySQL. Yogyakarta: Science House.
14. Suprianto, D. (2008). Smart Book PHP programming. Jakarta: Oase Media.

## APPENDICES

Appendix A **– Project Management Document**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task No | Task Name | Duration | Start | Finish | Deliverable |
| 1 | Project Plan | 6 days | 2nd March, 2020 | 7th March, 2020 | Work breakdown  structure and Gantt Chart |
|  |  |  |  |  |  |
| 2 | Introduction | 14 days | 9th March, 2020 | 22nd  March,2020 | Introductory Chapter |
|  |  |  |  |  |  |
| 3 | Literature Review | 30 days | 23rd  March, 2020 | 22nd April, 2020 | Reviewed Literature |
|  |  |  |  |  |  |
| 4 | System Analysis and Design | 27 days | 23nd April, 2020 | 18th May, 2020 | System requirement specification which includes functional requirements, non- functional requirements and their models which are Activity Diagram, Class Diagram, Use Case, Data flow diagram, Entity Relationship  Diagram, etc. |
|  |  |  |  |  |  |
| 5 | Implementation | 104 days | 19th My, 2020 | 31st August, 2020 | Coding |
|  |  |  |  |  |  |
| 6 | Testing | 8 days | 1st  September, 2020 | 8th  September, 2020 | System Testing,  Unit Testing, Integration Testing |
|  |  |  |  |  |  |

Appendix B **– Gantt Chat**



Appendix C **– Proceedings of interview**

Q1. What makes your organization different from other organization who rents a car? Q2. What are the objectives of your organization?

Q3. What is the mission of your organization?

Q4. How many branches does your organization have? Q5. How many staffs do you have?

Q6. How does your current system work? Q8. How many cars do you have?

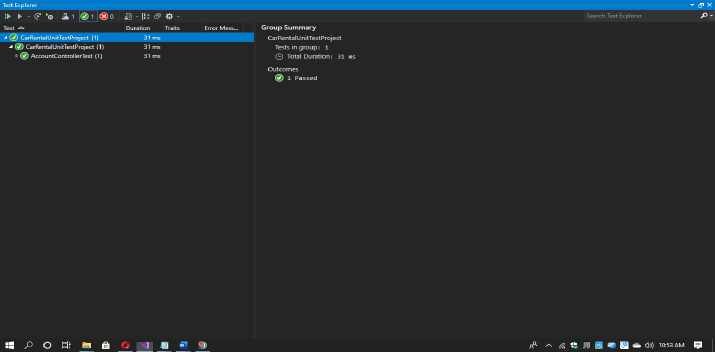
Q9. What kinds of car models do you have?

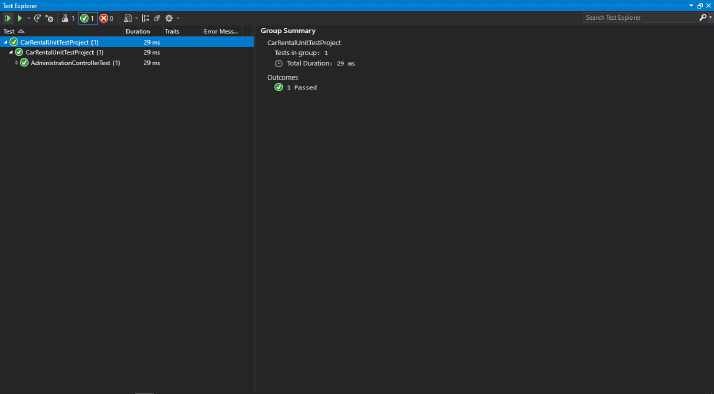
Q10. What are the procedures or steps when a customer rents a car? Q11. Where do you keep customer and rental information?

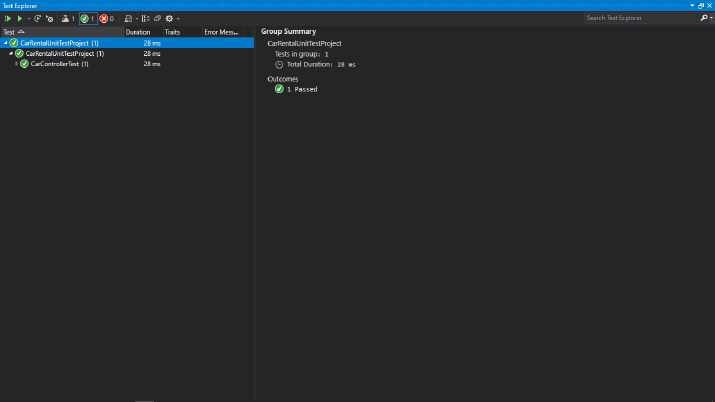
Q12. How do you keep track of which cars are rented and which are not? Q13.How many cars can a customer rent at a time?

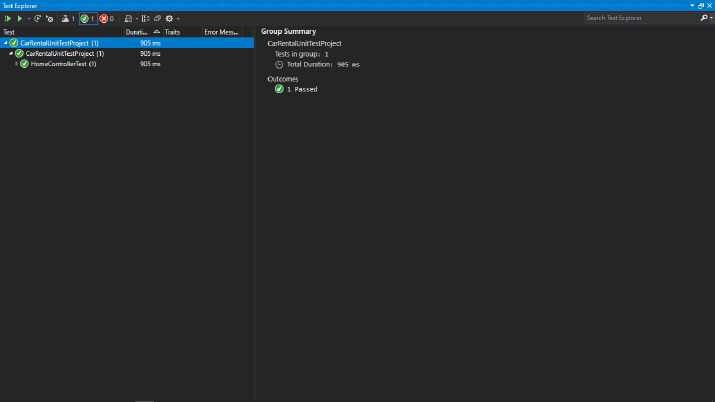
Q14. How do you generate customer and rental information?

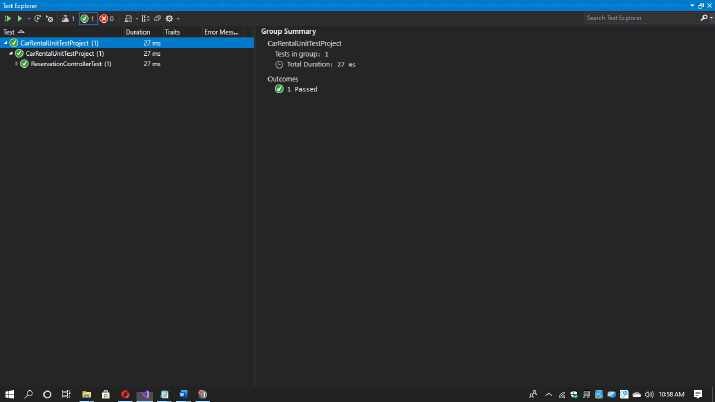
Appendix D **– Test Cases**











Appendix E **- User Guide**

#### System Overview

The application is a web-based application which means it only run on web browsers like google chrome, safari, and opera. The car rental system would function in such a way that user would register with the system, followed by logging in. After registration, the second step is to search for a car using the search box by inputting pick up and drop off date and time. A customer would select the car of his/her choice, then make a reservation.

#### organization of the guide

This user guide consists some sections which are; system summary, getting started, and using the system.

System Summary section provides a general overview of the system. The summary shows the uses of the system’s hardware and software requirements, system’s configuration, user access levels and system’s behavior.

Getting started section explains how to search for the system on the internet and start using it. The section presents briefly system menu.

#### SYSTEM SUMMARY

This section shows the general overview of the system. It outlines the uses of the system’s hardware and software requirements, system’s configuration, user access levels and system’s behavior.

#### System Configuration

YsQuArE Rentals system operates on a web browser installed on any operating system. It is compatible with any version of browser. The system requires internet connection to operate, save data, and load the pages that are contained inside the system.

#### User Access Levels

A customer must register with the system before he/she reserves a car.

#### Contingencies

If there is no internet connection, the system pages will not be loaded and system will not connect to the web server so data will be saved to the database.

#### GETTING STARTED

Getting started section describes how to get the system on the internet and use it.

#### Searching for The System and Logging In

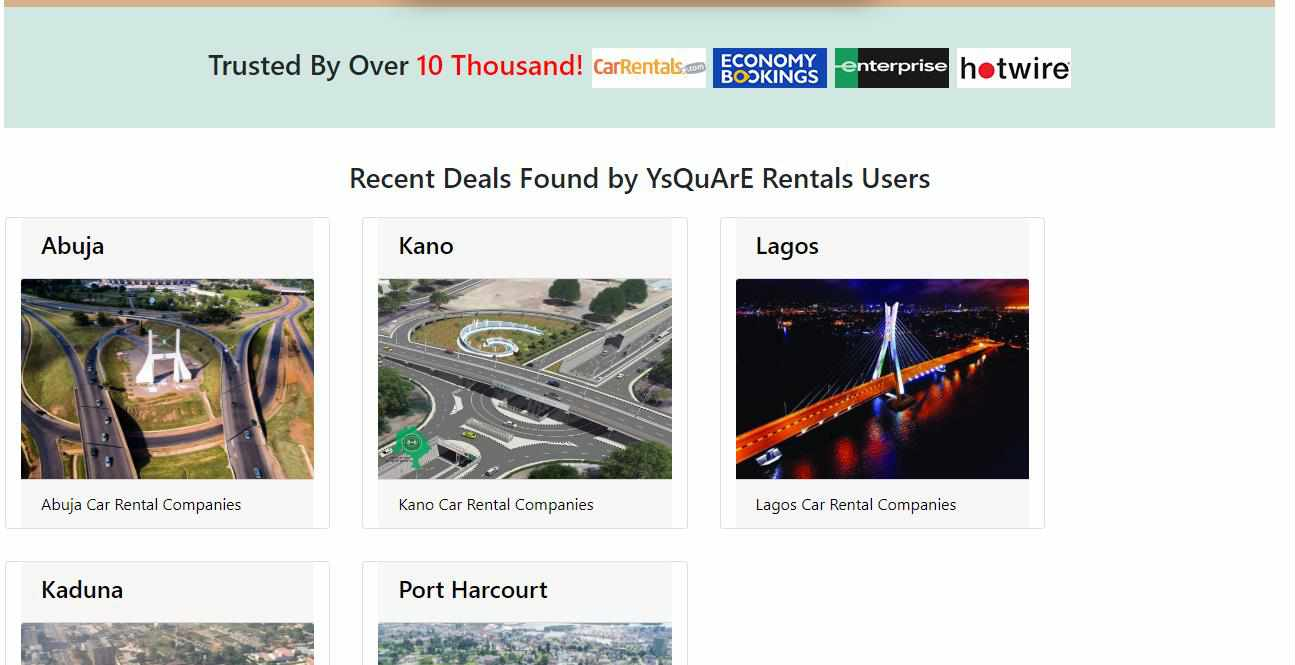
When the system is launched, it can be found in any search engine by typing [www.ysquarerentals.com](http://www.ysquarerentals.com/) or ysquare rentals and hit the link that appears on the browser. The user as to register first before logging in to system.

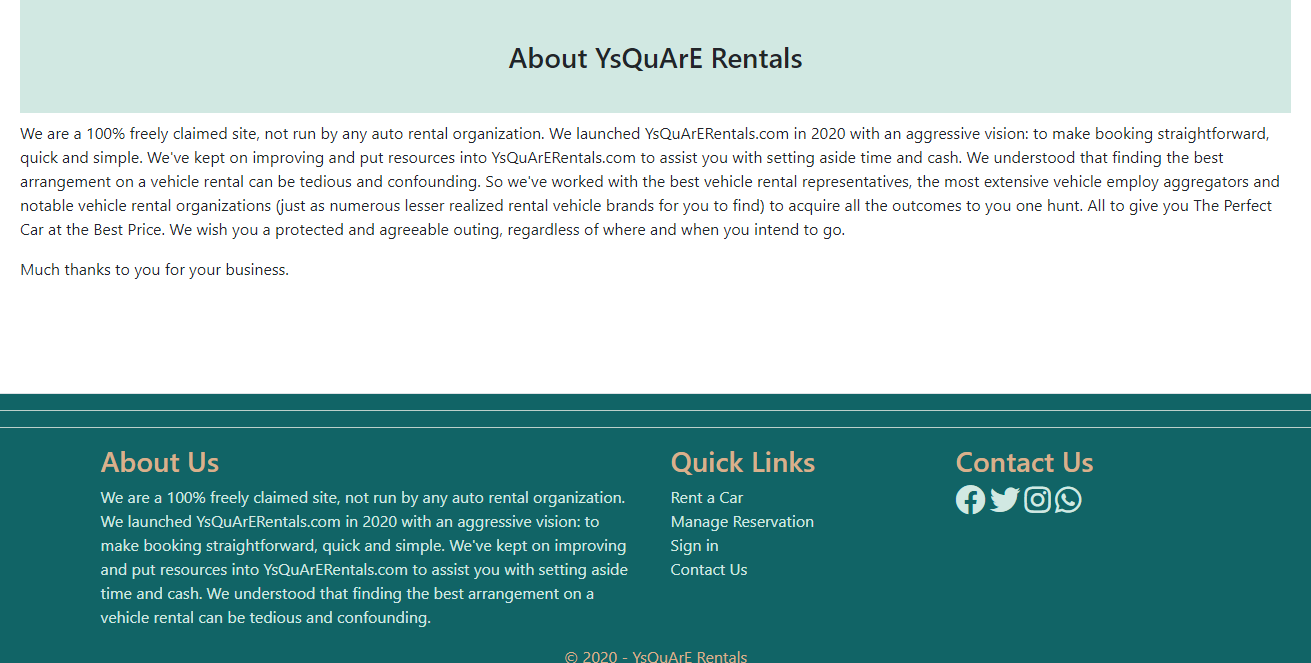
#### System Menu

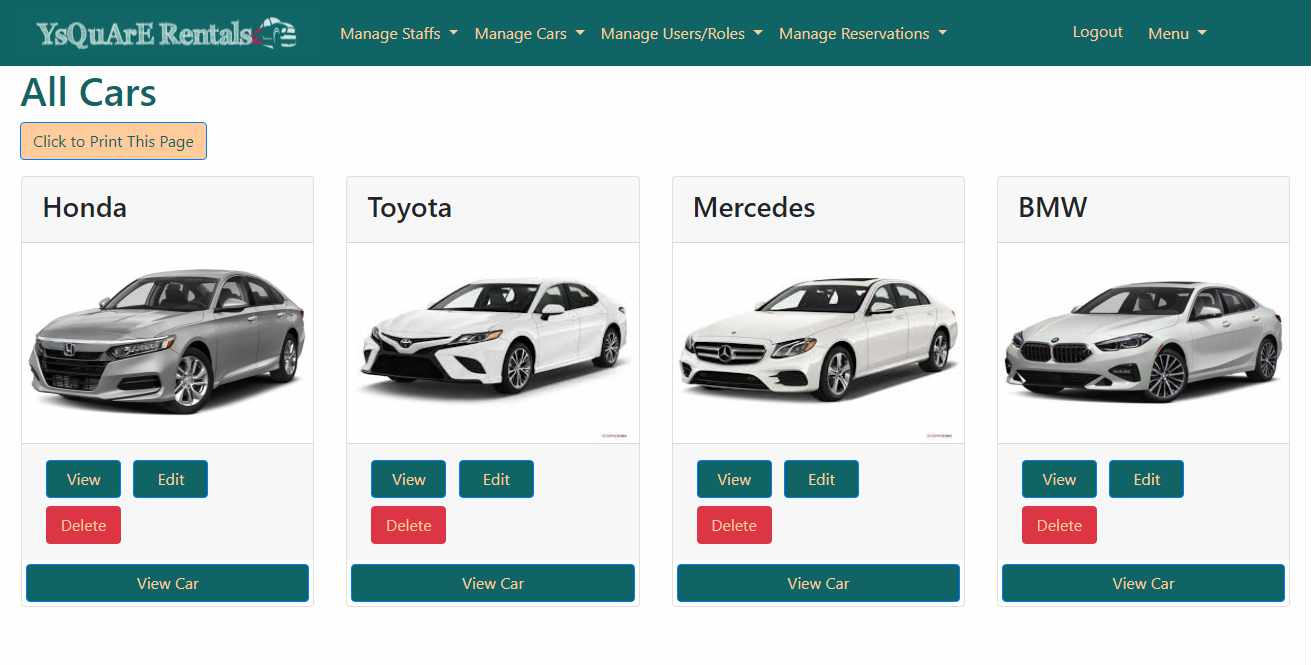
The car rental system uses a dropdown menu which contains the navigation links to the remaining part of the system.

Appendix F **– User Interfaces**









Appendix G **– Source Codes Home Controller**

namespace CarRentalSystem.Controllers

{

[Authorize(Roles = "Admin")]

//[Authorize(Roles = "User")]

public class CarController : Controller

{

private readonly ICarRepository \_carRepository;

private readonly IHostingEnvironment hostingEnvironment;

public CarController(ICarRepository carRepository,

IHostingEnvironment hostingEnvironment)

{

\_carRepository = carRepository; this.hostingEnvironment = hostingEnvironment;

}

// GET: Car [AllowAnonymous]

public ActionResult Index()

{

var model = \_carRepository.GetAllCar(); return View(model);

}

// GET: Car/Details/5 [AllowAnonymous]

public ViewResult Details(int? id)

{

Car car = \_carRepository.GetCar(id.Value);

if(car == null)

{

Response.StatusCode = 404;

return View("CarNotFound", id.Value);

}

CarDetailsViewModel carDetailsViewModel = new CarDetailsViewModel

{

Car = car,

PageTitle = "Car Details"

};

//car = homeDetailsViewModel

return View(carDetailsViewModel);

}

// GET: Car/Create

public ViewResult Create()

{

return View();

}

// POST: Car/Create

[HttpPost]

//[ValidateAntiForgeryToken]

public IActionResult Create(CarCreateViewModel model)

{

if (ModelState.IsValid)

{

string uniqueFileName = null; if(model.Photo != null)

{

"images");

string uploadsFolder = Path.Combine(hostingEnvironment.WebRootPath,

uniqueFileName = Guid.NewGuid().ToString() + "\_" +

model.Photo.FileName;

string filePath = Path.Combine(uploadsFolder, uniqueFileName); model.Photo.CopyTo(new FileStream(filePath, FileMode.Create));

}

Car newCar = new Car

{

CarName = model.CarName, CarModel = model.CarModel, CarStatus = model.CarStatus, CarSeat = model.CarSeat, CarDoor = model.CarDoor,

//Gearbox = model.Gearbox, PhotoPath = uniqueFileName

};

\_carRepository.Add(newCar);

return RedirectToAction("details", new { id = newCar.CarId });

}

return View();

}

// GET: Car/Edit/5

public ViewResult Edit(int id)

{

Car car = \_carRepository.GetCar(id);

CarEditViewModel carEditViewModel = new CarEditViewModel

{

CarId = car.CarId, CarName = car.CarName, CarModel = car.CarModel,

CarStatus = car.CarStatus, CarSeat = car.CarSeat, CarDoor = car.CarDoor,

//Gearbox = car.Gearbox, ExistingPhotoPath = car.PhotoPath

};

return View(carEditViewModel);

}

// POST: Car/Edit/5 [HttpPost]

//[ValidateAntiForgeryToken]

public IActionResult Edit(CarEditViewModel model)

{

if (ModelState.IsValid)

{

Car car = \_carRepository.GetCar(model.CarId);

car.CarName = model.CarName; car.CarModel = model.CarModel; car.CarStatus = model.CarStatus; car.CarSeat = model.CarSeat; car.CarDoor = model.CarDoor;

//car.Gearbox = model.Gearbox; if (model.Photo != null)

{

if(model.ExistingPhotoPath != null)

{

string filePath = Path.Combine(hostingEnvironment.WebRootPath, "images", model.ExistingPhotoPath);

System.IO.File.Delete(filePath);

}

car.PhotoPath = ProcessUploadedFile(model);

}

\_carRepository.Update(car); return RedirectToAction("index");

}

return View();

}

private string ProcessUploadedFile(CarCreateViewModel model)

{

string uniqueFileName = null; if (model.Photo != null)

{

"images");

string uploadFolder = Path.Combine(hostingEnvironment.WebRootPath,

uniqueFileName = Guid.NewGuid().ToString() + "\_" + model.Photo.FileName; string filePath = Path.Combine(uploadFolder, uniqueFileName);

using (var fileStream = new FileStream(filePath, FileMode.Create))

{

model.Photo.CopyTo(fileStream);

}

}

return uniqueFileName;

}

public ActionResult Delete(int id)

{

var model = \_carRepository.Delete(id); return RedirectToAction("index");

}

}

**Account Controller**

namespace CarRentalSystem.Controllers

{

public class AccountController : Controller

{

private readonly UserManager<ApplicationUser> userManager; private readonly SignInManager<ApplicationUser> signInManager;

public AccountController(UserManager<ApplicationUser> userManager,

SignInManager<ApplicationUser> signInManager)

{

this.userManager = userManager; this.signInManager = signInManager;

}

[HttpPost]

public async Task<ActionResult> Logout()

{

await signInManager.SignOutAsync();

return RedirectToAction("homepage", "home");

}

// GET: /<controller>/ [HttpGet] [AllowAnonymous]

public IActionResult Register()

{

return View();

}

[HttpGet][HttpPost] [AllowAnonymous]

public async Task<IActionResult> IsEmailInUse(string email)

{

var user = await userManager.FindByEmailAsync(email); if (user == null)

{

return Json(true);

}

else

{

return Json($"Email{email} is already in use");

}

}

//recieves register view model as an input parameter [HttpPost]

[AllowAnonymous]

public async Task<IActionResult> Register(RegisterViewModel model)

{

if (ModelState.IsValid)

{

//var username = new IdentityUser { }

var user = new ApplicationUser { UserName = model.Email, Email = model.Email, city = model.City };

var result = await userManager.CreateAsync(user, model.Password);

if (result.Succeeded)

{

if (signInManager.IsSignedIn(User) && User.IsInRole("Admin"))

{

return RedirectToAction("ListUsers","Administration");

}

await signInManager.SignInAsync(user, isPersistent: false); return RedirectToAction("index", "home");

}

foreach (var error in result.Errors)

{

ModelState.AddModelError("", error.Description);

}

}

return View(model);

}

[HttpGet] [AllowAnonymous]

public async Task<IActionResult> Login(string returnUrl)

{

LoginViewModel model = new LoginViewModel

{

providers

ReturnUrl = returnUrl,

//getexternal...returns us the list of all configuredd external login

ExternalLogins = (await

signInManager.GetExternalAuthenticationSchemesAsync()).ToList()

};

return View(model);

}

//recieves Login view model as an input parameter [HttpPost]

[AllowAnonymous]

public async Task<IActionResult> Login(LoginViewModel model, string returnUrl)

{

if (ModelState.IsValid)

{

var result = await signInManager.PasswordSignInAsync(model.Email, model.Password, model.RememberMe, false);

if (result.Succeeded)

{

if (!string.IsNullOrEmpty(returnUrl) && Url.IsLocalUrl(returnUrl))

{

return LocalRedirect(returnUrl);

}

else

{

return RedirectToAction("homepage", "home");

}

}

ModelState.AddModelError(string.Empty, "Invalid Login Attempt");

}

return View(model);

}

[AllowAnonymous] [HttpPost]

public IActionResult ExternalLogin(string provider, string returnUrl)

{

var redirectUrl = Url.Action("ExternalLoginCallback", "Account",

new { ReturnUrl = returnUrl });

var properties = signInManager.ConfigureExternalAuthenticationProperties(provider, redirectUrl);

return new ChallengeResult(provider, properties);

}

[AllowAnonymous]

public async Task<IActionResult> ExternalLoginCallback(string returnUrl = null, string remoteError = null)

{

returnUrl = returnUrl ?? Url.Content("~/");

LoginViewModel loginViewModel = new LoginViewModel

{

ReturnUrl = returnUrl,

ExternalLogins = (await signInManager.GetExternalAuthenticationSchemesAsync()).ToList()

};

if(remoteError != null)

{

ModelState.AddModelError(string.Empty, $"Error fromexternal provider:

{remoteError}");

return View("Login", loginViewModel);

}

var info = await signInManager.GetExternalLoginInfoAsync(); if (info == null)

{

information");

ModelState.AddModelError(string.Empty, $"Error loading external login

return View("Login", loginViewModel);

}

var signInResult = await signInManager.ExternalLoginSignInAsync(info.LoginProvider,

info.ProviderKey, isPersistent: false,

bypassTwoFactor: true);

if (signInResult.Succeeded)

{

return LocalRedirect(returnUrl);

}

else

{

var email = info.Principal.FindFirstValue(ClaimTypes.Email);

if (email != null)

{

var user = await userManager.FindByEmailAsync(email); if ( user == null)

{

user = new ApplicationUser

{

UserName = info.Principal.FindFirstValue(ClaimTypes.Email), Email = info.Principal.FindFirstValue(ClaimTypes.Email)

};

await userManager.CreateAsync(user);

}

await userManager.AddLoginAsync(user, info);

await signInManager.SignInAsync(user, isPersistent: false);

return LocalRedirect(returnUrl);

}

ViewBag.ErrorTitle = $"Email claim not received from:{info.LoginProvider}";

ViewBag.ErrorMessage = "Please contant support on carrental.com"; return View("Error");

}

//return View("Login", loginViewModel);

}