

Ahsanullah University of Science and Technology
Department of Computer Science and Engineering



Car Rent and Sell

Information System Design
and
Software Engineering Lab(CSE 3224)

Submitted to

Mr. Emam Hossain
Assistant Professor, AUST

Dr. Nusrat Sharmin,
Lecturer, AUST

Submitted By:

Md. Rubel Mia
ID: 15.02.04.007
Asadullahhil Galib
ID: 15.02.04.022
Narayan Das Nitol
ID: 15.02.04.025

Contents

1	Introduction	1
1.1	Project Goal	1
1.2	Features	2
1.3	Feasibility Analysis	2
1.3.1	Technical Feasibility	3
1.3.2	Economical Feasibility	3
1.3.3	Operational Feasibility	4
1.4	Cost and Benefit Analysis	4
1.4.1	Cash Flow Analysis	5
1.4.2	Present Value Analysis	5
1.5	Project Scheduling	6
1.5.1	Gantt Chart	7
1.6	Conclusion	7
2	Information Gathering	8
2.1	Objectives of Information Gathering & Interview	8
2.2	Questionnaire & Interview Pattern	8
2.3	Reason To Select Interview Personnel	9
2.3.1	Interview Questions & Answers	10
2.4	Revised Requirement Analysis	11
2.5	The list of Activities	11
2.6	Conclusion	12
3	Data Flow Diagram and Use Case Diagram	13
3.1	Activities of the project	13
3.2	Main Process	14
3.3	Sub Processes	14
3.4	Entity Names	15
3.5	Database Names	15
3.6	Data Flow Diagram Levels	15
3.7	Context Level Diagram	15
3.8	Level 1 Diagram	17
3.9	Level 2 Diagram	18

3.10	Use Case Diagram	23
3.10.1	Actors	23
3.10.2	Details of Use Case Models	23
3.10.3	Association	24
3.10.4	Include	24
3.10.5	Extend	24
3.10.6	Generalization/ Inheritance	25
3.11	Use Case Models	25
3.12	Conclusion	25
4	Relationship Diagram and Class Diagram	26
4.1	Entity Relationship Diagram (ERD)	26
4.1.1	ER Diagram	27
4.1.2	Entities	27
4.1.3	Relationships	28
4.1.4	Cardinality Constraints of Relationship	28
4.1.5	Attributes	28
4.2	Class Diagram	30
4.2.1	Classification	30
4.2.2	Relationship	30
4.2.3	Visibility	31
4.2.4	Classes of our project	31
4.2.5	Class Diagram	31
4.3	Conclusion	33
5	Discussion	34
5.1	Limitation	34
5.2	Future Work	35
6	Project Source Code	36

List of Figures

1.1	Gantt Chart for Car Rent and Sell Website	7
3.1	Level 1 diagram of Car Rent and Sell Website	17
3.2	Select Car Process of Car Rent and Sell Website	18
3.3	Registration for Car Process of Car Rent and Sell Website	19
3.4	Cancel Order Process of Car Rent and Sell Website	20
3.5	Update Order Process of Car Rent and Sell Website	21
3.6	Review Process of Car Rent and Sell Website	22
3.7	Use Case Diagram for Car Rent and Sell Website	25
4.1	ER Diagram for Car Rent and Sell Website	27
4.2	Class Diagram Of Car Rent and Sell Website	32

List of Tables

1.1	Cost Benefit Analysis for Car Rent and Sell Website	4
1.2	Cash Flow Analysis for Car Rent and Sell Website	5
1.3	Project Scheduling for Car Rent and Sell Website	6
2.1	Interview Questionnaire Pattern of Car Rent & Sell Website . . .	9

Chapter 1

Introduction

We're living in a world of click-and-consume. All our car customers can buy anything they want online. So, often in the auto industry, we forget that long before they hit the lot, consumers are looking online for their next car. To do that, we have to promote our inventory. As, we start to build our online inventory, think of it more as a front door to our dealership. We are welcoming people . We are showing them around. We are making them feel comfortable with our inventory. Our online inventory is a way for people to collect leads which is just one of many ways to rent and sell cars online.

1.1 Project Goal

1. It will optimize the time spend in the activities performed during the sale and rent purposes.
2. It will help customers to choose their dream car sitting at their home.
3. It will gain maximum business region.
4. Providing better accessibility and up to date information for customer.
5. Managing important data and information of car and customer and place.
6. Providing a friendly interface for the customer/tourist.

1.2 Features

1. The main purpose of Car Rent and Sell is buy and rent cars online. Using this project anyone can rent/buy cars online including by budget, cities, model, company etc.
2. Our project is meant to give people a better and trustworthy platform where they can buy and rent cars of their own choice and obviously on their own terms and condition.
3. With the help of internet and computer systems a man from remote area can buy his car from anywhere the country/world.
4. This project will provide facility to all buyer, seller and dealers for rent/sell cars. If anyone want to buy cars online, then first he will register in this website and then he can see all the details about the cars including prices, model and features.
5. Any person can register in the website and after that he can buy and rent cars, compare the cars regarding budget, model, company, city. Un-registered person may only see common pages with all buyer and seller's information.

1.3 Feasibility Analysis

Feasibility Analysis lets the developer foresee the future of the project and the usefulness. Feasibility Analysis of a system or website is according to its workability, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study. Here we are going to provide the feasibility of the project that is being designed and lists various areas that will be considered very carefully during the feasibility study of our website such as Technical, Economical and Operational feasibilities.

1.3.1 Technical Feasibility

The website must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. This website will be done with following elements given below :

1. **Front-end Language:** HTML, CSS
2. **Back-end Language:** ASP.NET

1.3.2 Economical Feasibility

Economical feasibility is the second part of resource determination. This includes the time of the system analysis team, the cost of doing a full systems study (including the time of employees) the cost of the business employee time, the estimated cost of hardware/software or software development. Developing this application is highly economically feasible. We don't need to spend much money for the development of the system. The only thing to development with an effective supervision. The costs for the software developers and servers are reasonable and the company for which the project will be developed will be highly benefited by serving customers as well as selling auto parts as every details of the servicing and selling of parts will be listed in the system. So, we can say that our project is economically feasible.

1.3.3 Operational Feasibility

The system analyst must consider the operational feasibility of the requested project even if the technical and economic resources are both judged adequate. Operational feasibility is dependent on the human resources available for the project and involves projecting whether the system will operate and be used once it is installed. If users are virtually used to the present system, see no problems with it and generally are not involved in requesting a new system, resistance to implementing the new system will be strong. Chances for it ever becoming operational are low. Alternatively, if users themselves have expressed a need for a system that is operational more of the time, in a more efficient and accessible manner, chances are better that the requested system will eventually be used.

1.4 Cost and Benefit Analysis

Cost Benefit Analysis estimates and totals up the equivalent money value of the benefits and costs to the community of projects to establish whether they are worthwhile. Tangible and Intangible are the types of Cost Benefit Analysis. The Cost Benefit Analysis of our project is given below:

	Cost	Benefit
Tangible	<ul style="list-style-type: none">• <i>Direct project cost</i>• <i>Software cost</i>• <i>Server cost</i>• <i>Maintenance cost of project</i>• <i>Network cost</i>• <i>Developer cost</i>	<ul style="list-style-type: none">• <i>Improve the speed of productivity</i>• <i>Lowering the inventory and production cost.</i>• <i>Well-arranged resources in a single system</i>
Intangible	<ul style="list-style-type: none">• <i>The lack of proper promotion</i>• <i>Project might fail to live up to expectation</i>• <i>International market could go high thus lowering the sale</i>	<ul style="list-style-type: none">• <i>Faster access of current data</i>• <i>Improvement of life of vehicles</i>• <i>The reputation of the company will increase</i>

Table 1.1: Cost Benefit Analysis for Car Rent and Sell Website

1.4.1 Cash Flow Analysis

An examination of a company's cash inflows and outflows during a specific period. The analysis begins with a starting balance and generates an ending balance after accounting for all cash receipts and paid expenses during the period. The cash flow analysis is often used for financial reporting purposes. See also cash flow projection, cash flow forecast:

	January	February	March	April
REVENUE	92,50,000	1,23,48,500	2,01,66,200	99,22,100
SERVER COST	9000	9000	9000	9000
SOFTWARE DEVELOPMENT	20,000	25000	15000	12000
PROJECT MAINTENANCE	10000	12800	22500	12750
Office Supplies	12,000	15,100	17,000	20,200
<u>TOTAL COST</u>	51,000	1,77,100	63500	53750
<u>CASH FLOW</u>	91,99,000	1,21,71,400	2,01,02,700	98,68,350
<u>CUMULATIVE CASH FLOW</u>	91,99,000	2,13,70,400	4,14,73,100	5,13,41,450

Table 1.2: Cash Flow Analysis for Car Rent and Sell Website

1.4.2 Present Value Analysis

Present value (PV) is the current value of a future sum of money or stream of cash flows given a specified rate of return. Future cash flows are discounted at the discount rate, and the higher the discount rate, the lower the present value of the future cash flows. Determining the appropriate discount rate is the key to properly valuing future cash flows, whether they be earnings or obligations.

We know that,

$$P.V = \frac{F.V}{(1+i)^n}$$

Where,

P.V (Present Value) = 88,50,200 (let us assume)

F.V (Future Value) = ?

i (Interest Rate) = 8% = 0.08

n (Time Period) = 3 years

We can write,

$$\begin{aligned} F.V &= P.V * (1 + i)^n \\ &= 88,50,200 * (1 + 0.08)^3 \\ &= 1,11,48,703 \end{aligned}$$

1.5 Project Scheduling

Project scheduling is a mechanism to communicate what tasks need to get done and which organizational resources will be allocated to complete those tasks in what timeframe. A project schedule is a document collecting all the work needed to deliver the project on time. The table is given below:

Activities	Description	Predecessors	Time(Day)
A	Project Discussion	-	10
B	Report Submission	-	2
C	Interview	A, B	6
D	Data Analysis	C	7
E	Coding	C, D	20
F	Documentation	E	3
G	Testing	E	5
H	Bug Fixing	G	8
I	Deployment	H	8

Table 1.3: Project Scheduling for Car Rent and Sell Website

1.5.1 Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. The gantt chart of our project is given below:

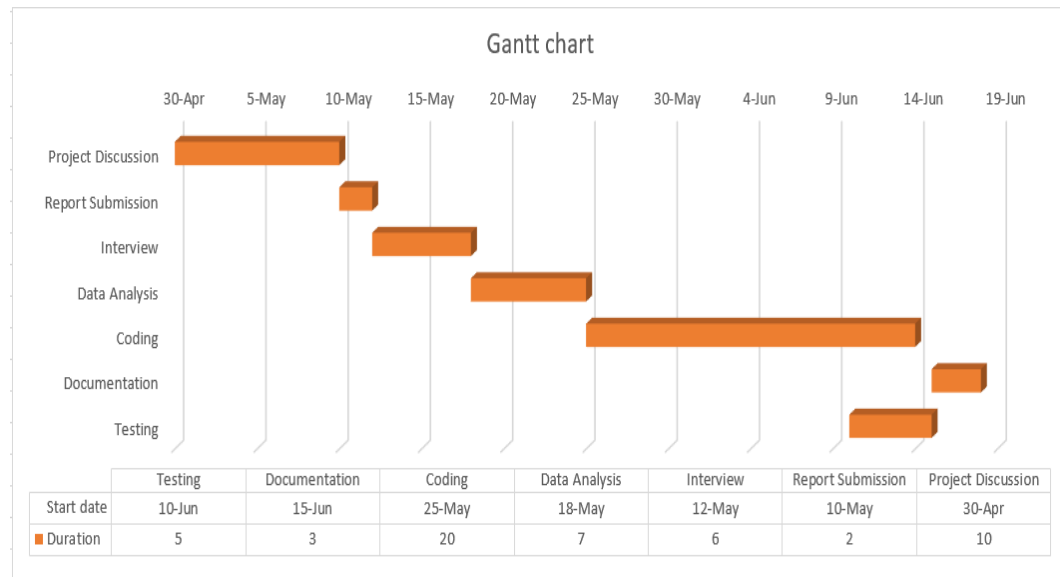


Figure 1.1: Gantt Chart for Car Rent and Sell Website

1.6 Conclusion

We know about that people in our country, who want to buy a new car, must have to go first in the car showroom., then choose his/her cars, then purchase it. This process is very time consuming. So, our main goal is to reduce the time-consuming factors.

Chapter 2

Information Gathering

Introduction

We know about that people in our country, who want to buy a new car, must have to go first in the car showroom., then choose his/her cars, then purchase it. This process is very time consuming. So, our main goal is to reduce the time-consuming factors. For making our project successful, we need to follow some steps gradually for the entire development of this project. Regarding this, information gathering is the first and foremost step.

2.1 Objectives of Information Gathering & Interview

The name of our project is CAR Rent & Sell. To make the project user-friendly, by useful data and information, it is important to interview authority of the car showroom. The information obtained from the relevant people working there would help us for the project to be perfect and precise work. Here we have constructed interview questions to gather information requirements and structured them in a meaningful way. All this information is gathered from the authority who invoked us to work for this project. According to the gathered information and collection of data, we have numerous scopes to improve the features and add new functionalities in future.

2.2 Questionnaire & Interview Pattern

Interviewing is a specific form of meeting or conference, and is usually limited to two persons, the interviewer and the interviewee. We have assembled information using the following 1 methods:

Car Rent and Sell Website			
Authors: Md. Rubel Mia Asadullahhil Galib Narayan Das Nitol	Date: 23-05-2018	Time: 02:30 pm	Duration: 60mins
Participants: Mr. Tanvir Ahmed Iftekhar Sales Executive – Navana Limited		Comments: We have got a clear idea of the website that authority requires and the current existing systems	

Table 2.1: Interview Questionnaire Pattern of Car Rent & Sell Website

1. **Interview of the authority :**We interviewed Mr. Tanvir Ahmed Iftekhar who is the Sales Executive of the Navana Limited.

For both methods of question and answers, we have chosen **Diamond shaped Structure**. The diamond-shaped structure for interviewing combines the pyramid and funnel structure. This structure entails beginning in a very specific way, then examining general issues, and finally coming to a very specific conclusion. So, we decided to work with this structure. In the interview process, we have included:

1. 4 Close Ended questions at first.
2. 4 Open Ended questions in the middle
3. 2 Close Ended questions at last

The table pattern for interview & questionnaire is given below:

2.3 Reason To Select Interview Personnel

As Mr. Tanvir Ahmed Iftekhar is the Sales Executive of Navana Limited, he has the utmost knowledge of which is required for his company to flourish further. He knows better about entire sector of this company. That's why we have selected for the interview.

2.3.1 Interview Questions & Answers

1. **How many people are working in this showroom?** Answer:65.
2. **How many people are working in this showroom?**
Answer:Around 500.
3. **Do you have any existing website for your company?**
Answer:Yes. It is www.amartoyota.com.
4. **Do you want any online customer support?**
Answer:Yes.
5. **How many cars you are selling in a month?**
Answer:Well, its very tough to say. Because it is not constant. It varies from month to month. As an example, sometimes we sell around 5 cars in a month. And sometimes we sell around 100 cars in a month. In the month of May-July, we sell maximum cars as it is the month of Government Purchase. Government alone buys maximum car for their employee and higher authority.
6. **How many categories you want to include in this website?**
Answer: There should be 5 categories in the website :
 - (a) Choice
 - (b) Test Drive
 - (c) Verification
 - (d) Rent a Car
 - (e) Services
7. **Do you want to include any virtual reality system in this website?**
Answer: Obviously. I want a 3D view of our showroom where people can see like the real one. Our customer can get the real feel of our showroom seating at their house. Customers can open the door of a car virtually, can see the windshield virtually, also can drive the card virtually. This will help our customers to select their cars.
8. **How you will control the Car Rental System of this website?**
Answer:Those cars which we want to rent, there will be a GPS system included with every cars. We can track any cars as we want. So, our cars will remain safe. We will provide 2 packages for our clients for rental purpose :
 - (a) Car with Driver
 - (b) Car without Driver

Car with a driver - in this package, our official driver will drive the cars
Car without driver - in this package, our clients will drive the cars.

9. **Which things is the must you want to include in the website?**

Answer:A realistic picture of the car with the price tag.

10. **Do you want any online review system for every car in your showroom?**

Answer:Yes. We will upload every video of every single cars of your showroom so that our customer will get a clear idea.

2.4 Revised Requirement Analysis

Through all the information collection, we revised our work and concluded that along with our features. For this constant update of information, we will need to create an Virtual Reality System. This will remain in our future goal for this application.

1. User friendly interface.
2. Well-interactive interface between customer and employee.
3. Adding a searchbox for initial search.
4. Admin panel is a must for handling the dealer, product and employee.

2.5 The list of Activities

We have interviewed people of a company before designing the project. We have collected the following information:

1. How the services are provided to the customers.
2. How the services are provided to the customers.
3. How they will manage the entire system.
4. The security management of the rental cars.
5. Virtual Reality of the showroom.

We also wanted to know the problems they are encountering for not having any dynamic website. We gathered the critical information from them regarding the types of features they want in the website. All these data and information would help us to design the website.

2.6 Conclusion

The interview and survey process gave us the opportunity to visit and interview different people and helped us a lot to know what users actually want from a website which contains information of such things related to buy rent a car company. These information and data will help us to design a website according to user choice and make our work easier

Chapter 3

Data Flow Diagram and Use Case Diagram

Introduction

Data Flow Diagram (DFD) graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

1. Logical information flow of the system.
2. Determination of physical system construction requirements.
3. Simplicity of notation.
4. Establishment of manual and automated systems requirements.

3.1 Activities of the project

The activities of our project are given below:

1. Ensure user-friendly interface.
2. Creating search field.
3. Creating compare field.
4. Add review.

5. Choose cars.
6. Creating an advance database for rental cars.
7. Test-drive scheduling.
8. Rental car scheduling.

3.2 Main Process

There are 5 main process in our system:

1. Select Car.
2. Registration for order
3. Cancel order.
4. Update order.
5. Review.

3.3 Sub Processes

There are 3 sub processes under **Select Car**

1. Display Car
2. Check Availability
3. Search Car.

There are 2 sub processes under **Registration for order**

1. Register for Test drive
2. Register for Rent

There are 1 sub processes under **Cancel Order**

1. Cancel Order

There are 1 sub processes under **Update Product**

1. Update Product

There are 1 sub processes under **Product Review**

1. Product Review

3.4 Entity Names

There are total 2 entity in our system:

1. Customer
2. Admin

3.5 Database Names

There are total 5 table in our system Database:

1. Car info
2. Order info
3. TestDrive info
4. Rent info
5. Review

3.6 Data Flow Diagram Levels

A data flow diagram can dive into progressively more detail by using levels and layers. The first level of DFD shows the main process within the system. This main process can be divided into further processes until we reach the final level. DFD levels are numbered 0, 1 or 2, and occasionally can go to even Level 3 or beyond. All these levels define the scope of what we are trying to accomplish. For our project we have constructed 3 levels. The levels are given below:

1. Context Level/ Level 0
2. Level 1
3. Level 2

3.7 Context Level Diagram

Context Diagram is a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It only contains one process node that generalizes the function of the entire system in relationship to external entities.

The diagram of the Context Level of our project is given below:

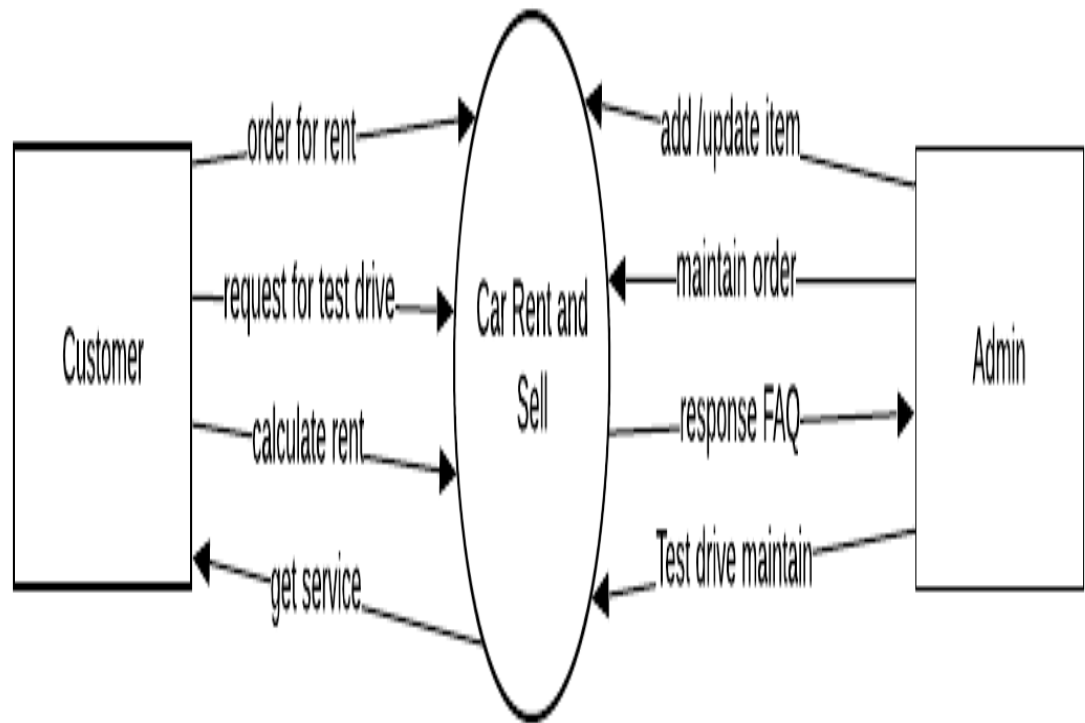


Figure 3.1: Context Level diagram of Car Rent and Sell Website

3.8 Level 1 Diagram

A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into subprocesses that can be analyzed and improved on a more intimate level.

The diagram of Level 1 is given below:

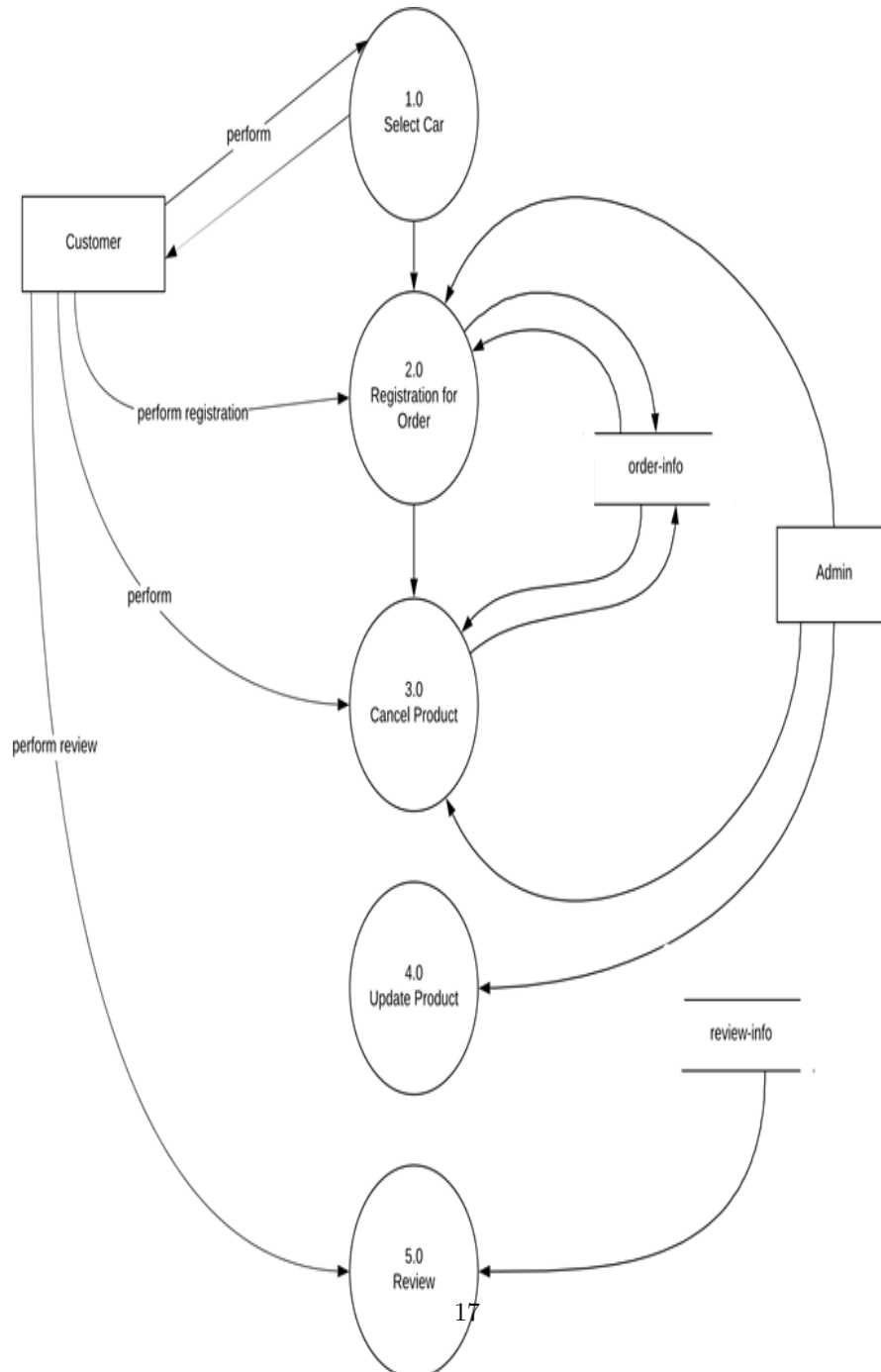


Figure 3.1: Level 1 diagram of Car Rent and Sell Website

3.9 Level 2 Diagram

A level 2 data flow diagram (DFD) offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system.

1.Select Car Process:

The diagram of Select Car Process is given below:

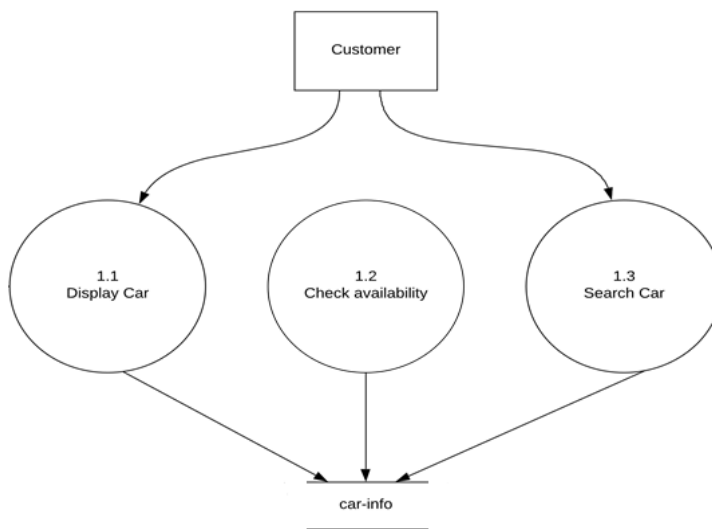


Figure 3.2: Select Car Process of Car Rent and Sell Website

2.Registration for Order Process:

The diagram of Registration for Order Process of Car Rent and Sell Website is given below:

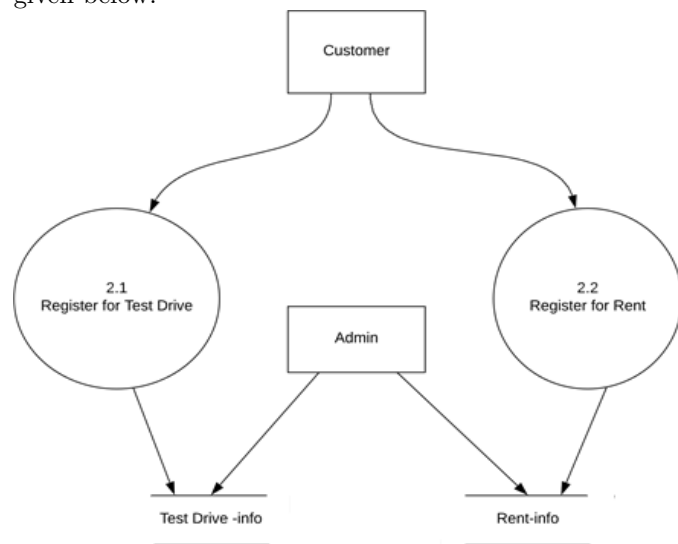


Figure 3.3: Registration for Car Process of Car Rent and Sell Website

3.Cancel Order Process:

The diagram of Cancel Order Process of Car Rent and Sell Website is given below:

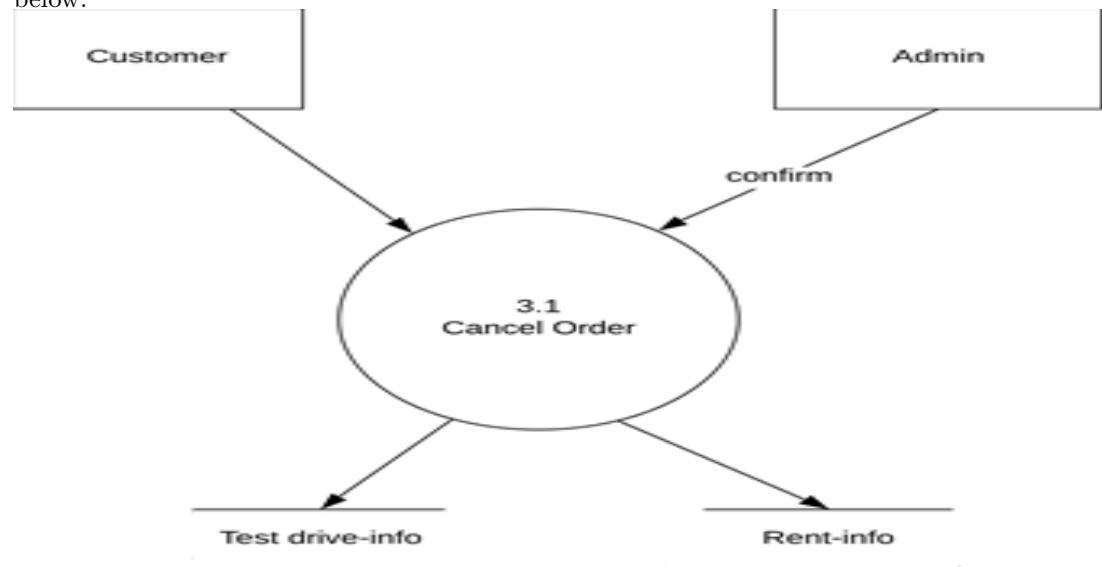


Figure 3.4: Cancel Order Process of Car Rent and Sell Website

4.Update Order Process

The diagram of Update Order Process of Car Rent and Sell Website is given below:

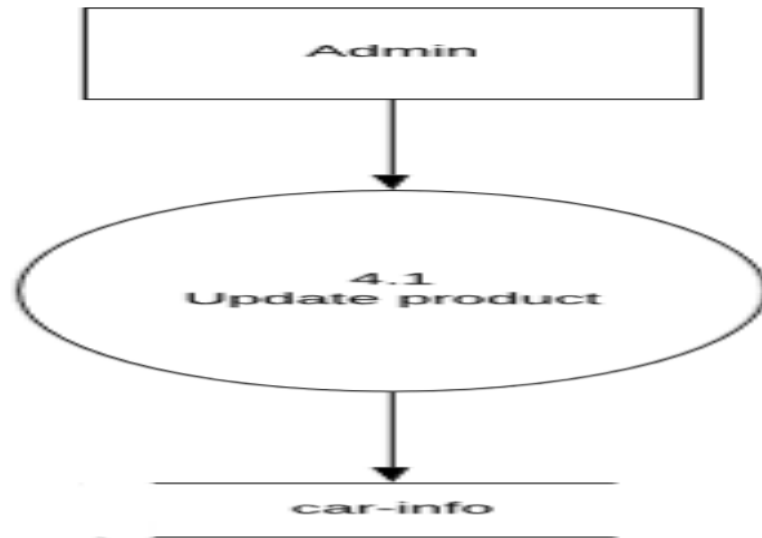


Figure 3.5: Update Order Process of Car Rent and Sell Website

5.Review

The diagram of Review Process of Car Rent and Sell Website is given below:



Figure 3.6: Review Process of Car Rent and Sell Website

3.10 Use Case Diagram

A Use Case model defines what a system does without describing how the system does it. This logical model of the system replicates the understanding of a system from the viewpoint of a user outside the system, which is the system's requirement. It also splits the mode of the system that works into actions, services and reactions that are major to the users of the system.

3.10.1 Actors

Customer: Customers can select car, ask for test drive, rent car, cancel orders.

Admin: Admin can manage registration process, update products and make orders decisions.

3.10.2 Details of Use Case Models

In Car Rent and Sell Website, we have the following use cases for these two actors:

Select Car

Actor: Customer Primary Path:

1. Check availability.
2. Select required car.

Registration for Order

Actor: Customer Primary Path:

1. Collect customer's info.
2. Verify customer.

Alternative Path:

1. In case of failure display registration error.

Cancel Order

Actor: Customer Primary Path:

1. Confirm Cancellation

Update Product

Actor: Admin Primary Path:

1. Add new car.
2. Remove unavailable car.
3. Update rent price and selling price.

Alternative Path:

1. Can't update due to some unexpected reason.

Review

Actor: Customer Primary Path:

1. Can review rent service.
2. Can review sell service.

Use Case Relationships

The final element in Use Case diagrams are Relationships. There are 4 types of relationships:

3.10.3 Association

An association between an actor and a use case specifies that the actor and the use case in some way interact or communicate with each other. In this relationship, an actor is connected to a use case using line with no arrowheads.

3.10.4 Include

Between 2 use cases include is a directed relationship, used to show that performance of the included use case is inserted into the behavior of the including (base) use case. Usually a use case comprises a behavior that is common to more than one other use case. Here, the arrow directs to the common use case.

3.10.5 Extend

Extend identifies how & when the behavior cleared in usually optional extending use case can be introduced into the behavior established in the extended use case. From the base use case, different use cases handle exceptions. The arrow plugs from the extended to the base use case.

3.10.6 Generalization/ Inheritance

Generalization states the relationship that can happen between two use cases and shows that one use case (child case) inherits the structure, behavior and relationships of the another actor (parent). Here, the arrow points to the general use case (parent case).

Extension Points

Use cases can include conditional steps as well as extensions or alternative scenarios. Extension point is a property of a use case that categorizes a point where the actions of a use case can be amplified with elements of another (extending) use case.

3.11 Use Case Models

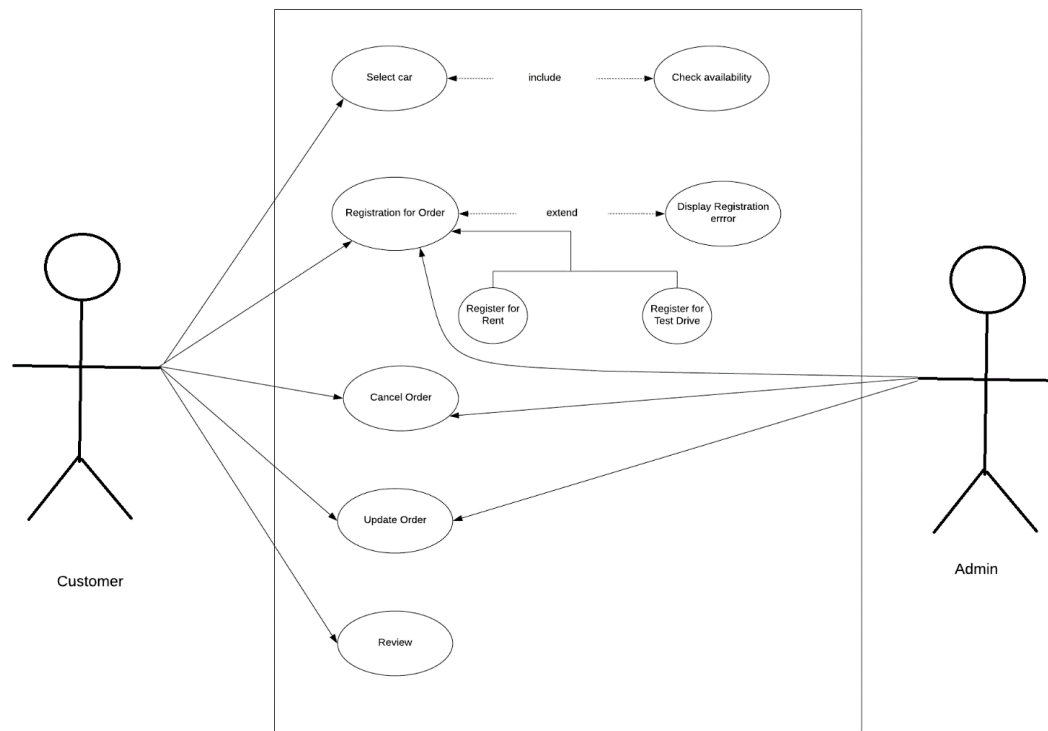


Figure 3.7: Use Case Diagram for Car Rent and Sell Website

3.12 Conclusion

As our project is Car Rent and Sell, the main intention behind this project is to provide such type of facilities which can prevent the wastage of time while buying car in the showroom and book a car for rental purpose. With the help of data flow diagram, we have achieved the clear idea about process on the system and relations among them. In use case diagram we have found the relationship among actors and entire system which will help to develop our project very clearly and smoothly.

Chapter 4

Relationship Diagram and Class Diagram

Introduction **Class diagram** and **Entity Relationship Diagram(ERD)** both model the structure of a system. **Class diagram** represents the dynamic aspects of a system, both the structural and behavioral features. **ERD** represents only structural features which provide a static view of the system.

4.1 Entity Relationship Diagram (ERD)

An entity relationship model, also called an **Entity Relationship Diagram (ERD)** is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An **Entity Relationship Diagram (ERD)** shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

A basic ER model is composed of entity types and specifies relationships that can exist between instances of those entity types. The 3 key components of ERD are:

1. Entity
2. Relationship
3. Attribute

4.1.1 ER Diagram

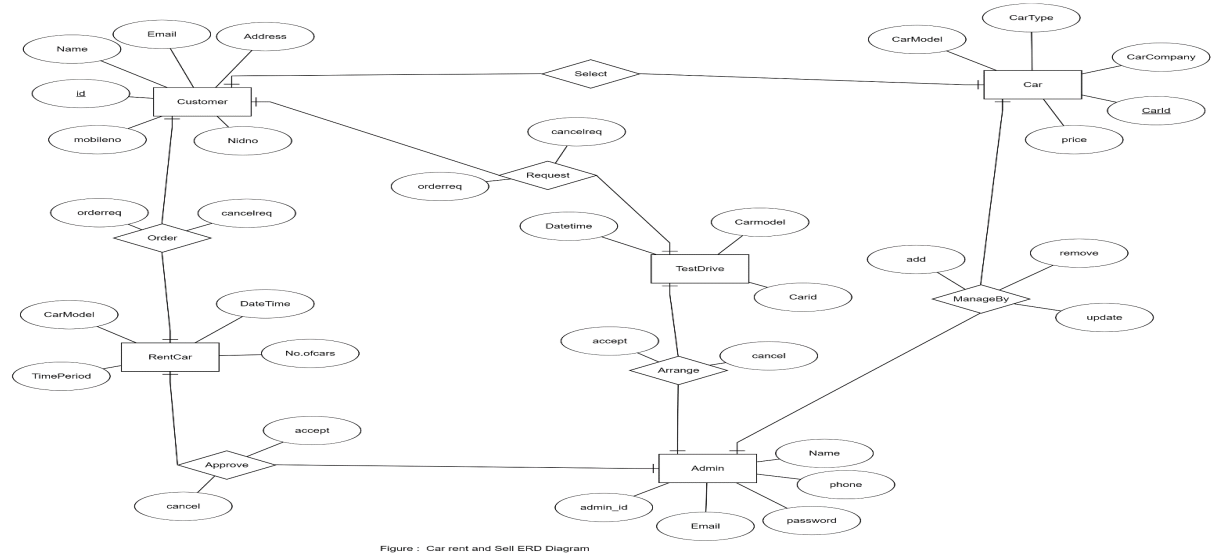


Figure 4.1: ER Diagram for Car Rent and Sell Website

4.1.2 Entities

Entity will be real world object from problem domain and data will be stored in the database on entity. It is noun all time and distinguishable from other objects.

In our project, we have the following entities in the ERD:

1. Customer
2. Admin
3. Rent_Car
4. Test_Drive
5. Gallery

4.1.3 Relationships

The interaction between the entity set are called relationship. It is always a verb and creates association among two or more attributes.

In our project, the relationships between the entity sets are:

Entity Set 1: Customer, Rent_Car

Relationship: Order

Entity Set 2: Admin, Rent_Car

Relationship: Approve

Entity Set 3: Customer, Text_Drive

Relationship: Request

Entity Set 4: Admin, Text_Drive

Relationship: Arrange

Entity Set 5: Admin, Gallery

Relationship: ManagedBy

4.1.4 Cardinality Constraints of Relationship

The maximum number of entity that can be associated with another entity via a relationship, are defined as cardinality ratio. It is most useful in describing binary sets of relationship sets. There are 4 types of cardinality constraints:

1. Many to many
2. Many to one
3. One to many
4. One to one

4.1.5 Attributes

The properties of entity or relationship are called attributes. An entity is basically described using a set of attributes. In our project, we have the following attributes of each entity in the ERD:

Entity Name: Customer

Attributes:

1. user_id
2. name
3. email
4. password
5. address
6. nid

Entity Name: Rent_Car

Attributes:

1. rent_car_id
2. car_model
3. time_period
4. date_time
5. no_of_cars

Entity Name: Admin

Attributes:

1. admin_id
2. password
3. email
4. address
5. phone
6. mobile

Entity Name: Test_Drive

Attributes:

1. car_id
2. user_id
3. date_time
4. car_model

Entity Name: Gallery

Attributes:

1. car_id
2. car_model
3. car_type
4. car_company
5. price

4.2 Class Diagram

Class diagrams are fundamental to the object modeling process and model the static structure of a system. Class diagrams is a snapshot that describes exactly how the system works, the relationships between system components at many levels, and how to implement those components.

During the analysis and design phases of the development cycle, create class diagrams to perform the following functions:

1. Define the structure of classes and other classifiers.
2. Define relationships between classes and classifiers.
3. Illustrate the structure of a model by using attributes, operations, and signals.
4. Show the common classifier roles and responsibilities that define the behavior of the system.
5. Show the implementation classes in a package.
6. Show the structure and behavior of one or more classes.

4.2.1 Classification

Class Diagram has three parts:

1. Class
2. Attribute
3. Method

1. Class Classes represent an abstraction of entities with common characteristics. **2. Attribute** An attribute is a named property of a class that describes the object being model. In class diagram, attributes appear in the second compartment just below the name compartment. **3. Method** Method describe the class behavior and appear in the third compartment.

4.2.2 Relationship

There are 4 types of relationships in a class diagram:

Inheritance

Inheritance refers to the ability of one class (child class) to inherit the identical functionality of another class (parent class), and then add new functionality of its own. The advantage of inheritance is if we want to add or change attributes of the child classes, we don't have to change the subclasses, we can only change the super class and it applies across all sub classes.

Association

The simplest type of relationship is association, whose presence or absence does not affect in the whole scenario of the class diagram. Associations are shown as a simple line on a class diagram.

Aggregation

Aggregation offers a means of showing that the whole object is composed of the sum of its parts. It is often described as a “has” relationship. The diamond in the end of the relationship line is not filled in here. It is a special type of association that specifies the whole and its parts.

Composition

Composition is a whole/ part relationship in which the whole has a responsibility for the part, is a stronger relationship. It is usually shown with a filled-in diamond. The classes which are related through composition relationship are dependent on each other.

4.2.3 Visibility

We use visibility markers to signify who can access the information contained within a class. Private visibility, denoted with a - sign, hides information from anything outside the class partition. Public visibility, denoted with a + sign, allows all other classes to view the marked information. Protected visibility, denoted with a # sign, allows child classes to access information they inherited from a parent class.

4.2.4 Classes of our project

The class names are listed below:

1. Customer
2. Rent_Car
3. Admin
4. Gallery
5. Test_Drive
6. Review

4.2.5 Class Diagram

The class diagram of Car Rent and Sell Website is given below:

Car Rent & Sell

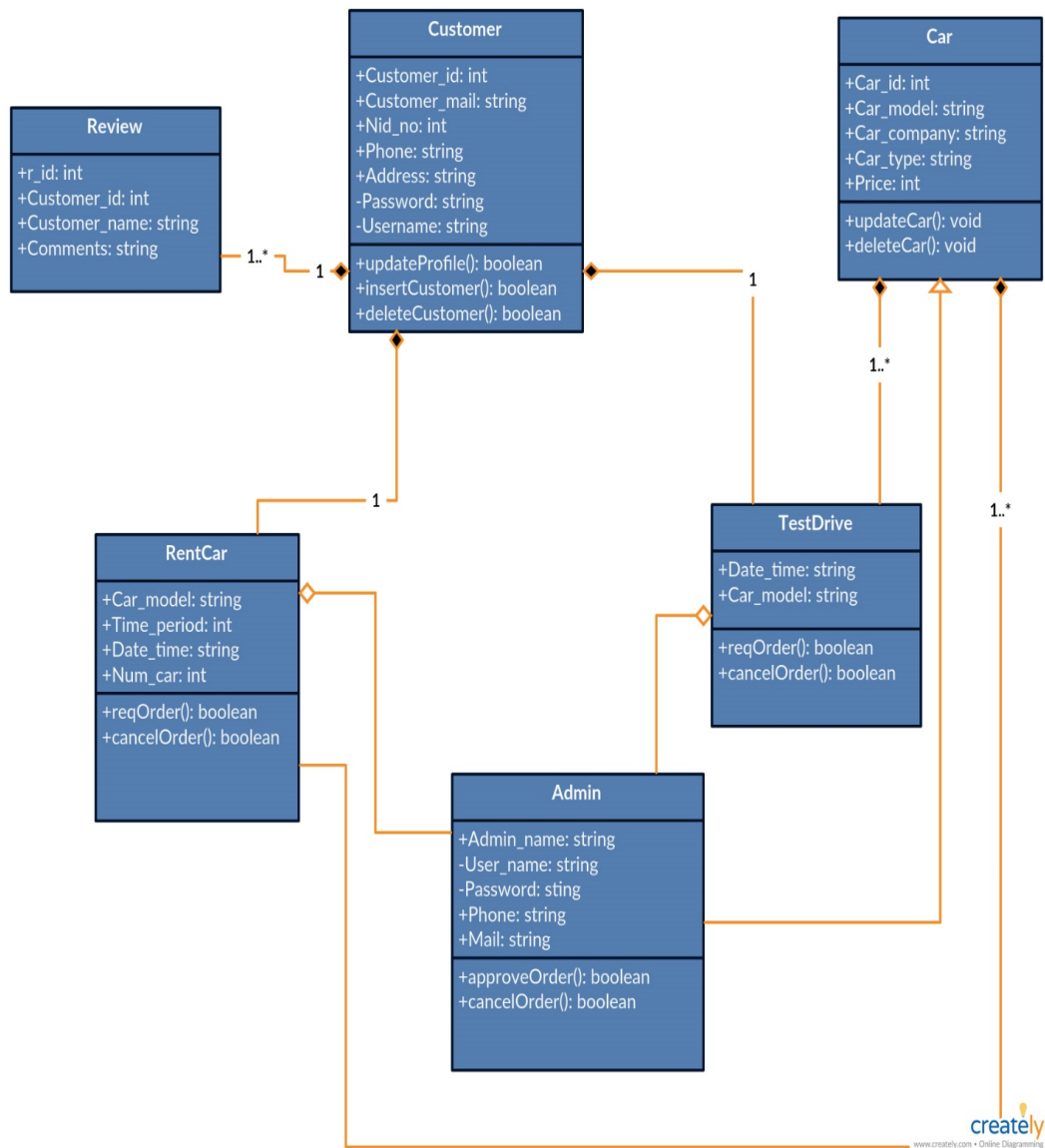


Figure 4.2: Class Diagram Of Car Rent and Sell Website

4.3 Conclusion

A use case diagram provides project planning skeleton. We have thought through every user's needs and designed our class diagram. The entity relationship diagram (ERD) is the logical structure of database and shows relationships of entity sets stored in a database. We have illustrated all the entities, attributes, data types and relationships of our entity relationship diagram. Above mentioned diagrams will help us to visualize the whole system more evidently

Chapter 5

Discussion

Conclusion

The main focus of Car Rent and Sell website is to consume time of customers to choice cars and also to rent a car for their different purpose. Customers do not need any physical contact of the showroom , rather than they can choose car from anywhere of the country. We have initially started our project by our imagination and the requirements of the company. After that we have gone through all the feasibility analysis, cost benefit analysis, cash flow analysis and so on. We have also designed our Gantt chart and started working by following the project scheduling of our project. Then we interviewed the authority. We also gathered information to know the requirements of the authority.

After the interview we have designed our Data Flow Diagram (DFD) by which we got the specific vision of our project and got a clear graphical representation of our system's requirements. After designing the DFD, we designed our Use Case Diagram. Then we designed the entity relationship diagram (ERD) which is the logical structure of database and shows relationships of entity sets stored in a database. We have also designed our class diagram which contains all relevant relations and data types.

5.1 Limitation

We tried our best to impliment all the features of our project which the authority wants. We have collected every important information about this company which is related to our project. But after all, we have some limitations in our project which we could not impliment at all but have plan to do it in future. The limitations of our project are given below:

1. We have designed our website only for 1 showroom.
2. Rental management system is not satisfactory.

3. Virtual Reality System is not implemented.
4. Online payment is not included in our website.
5. Online Customer Support feature is not working.

5.2 Future Work

In future we will fulfill the limitations of our website which we mentioned before. Then we have also some idea related to Car Rental System which will give customer more pleasure and satisfaction. The authority also wants some unique features for their Car selling and Rental system by which their customer will get 100

The future work of our project is given below:

1. Automatic Car rental management system
2. Online Car rental system like UBER where user can select the destination point of their travel.
3. We will implement Virtual Reality System by which the customer get the real feel of the car showroom. They can touch the cars virtually . Also see all the specifications of the car as like real.
4. Online payment system with MasterCard/Visa/CreditCard/Bkash.
5. 24/7 online customer support.
6. Well-interactive interface between customer and company.

Chapter 6

Project Source Code

```
// Add_Product.cs

protected void ButtonAdd_Click(object sender, EventArgs e)
{

    string CS = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;

    if (FileImage.HasFile)
    {
        SavePath = Server.MapPath("~/Images/ProductImage/") + PID;
        if (!Directory.Exists(SavePath))
        {
            Directory.CreateDirectory(SavePath);
        }
        Extention = Path.GetExtension(FileImage.PostedFile.FileName);
        FileImage.SaveAs(SavePath + "\\\" + TextBoxCarName.Text.ToString().Trim()
+ TextBoxCarModel.Text + Extention);

    }

    using (SqlConnection con = new SqlConnection(CS))
    {
        SqlCommand command = new SqlCommand("insert into ProductDetails
values('\" + TextBoxCarName.Text + '\", '\" + TextBoxCarModel.Text + '\",
 '\" + TextBoxCarPrice.Text + '\", '\" + TextBoxCarName.Text + TextBoxCarModel.Text.ToString() .Trim()
+ '\", '\" + Extention + '\", '\" + TextBoxDescription.Text + '\", '\" +
```

```

TextBoxReviewLink.Text + "')", con);

        con.Open();
        command.ExecuteNonQuery();

        LabelAdd.Text = "Successfully Added ! ";
        TextBoxCarName.Text = "";
        TextBoxCarPrice.Text = "";
        TextBoxCarModel.Text = "";

    }

}

// Admin.cs

public partial class Admin : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btn_add_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/AddProduct.aspx");
    }

    protected void btn_update_Click(object sender, EventArgs e)
    {

    }

    protected void btn_rent_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/ViewCarRentOrder.aspx");
    }

    protected void btn_sell_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/ViewTestDriveOrder.aspx");
    }
}

```

```

// Comment_Box.cs

public partial class CommentBox : System.Web.UI.Page
{
    string cs = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            fillData();
        }
    }
    //FillData method for filling Repeater Control with Data
    private void fillData()
    {
        SqlConnection con = new SqlConnection(cs);
        con.Open();
        DataTable dt = new DataTable();
        SqlDataAdapter adapt = new SqlDataAdapter("Select * from Review
Order by Id Desc", con);
        adapt.Fill(dt);
        con.Close();
        PagedDataSource pds = new PagedDataSource();
        DataView dv = new DataView(dt);
        pds.DataSource = dv;
        pds.AllowPaging = true;
        pds.PageSize = 4;
        pds.CurrentPageIndex = PageNumber;
        if (pds.PageCount > 1)
        {
            rptPaging.Visible = true;
            ArrayList arraylist = new ArrayList();
            for (int i = 0; i < pds.PageCount; i++)
                arraylist.Add((i + 1).ToString());
            rptPaging.DataSource = arraylist;
            rptPaging.DataBind();
        }
        else
        {
            rptPaging.Visible = false;
        }
        Repeater1.DataSource = pds;
        Repeater1.DataBind();
    }
}

```

```

// Car_Details.cs

public partial class DetailsCar : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string myid = Request.QueryString["id"];

        string CS = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;

        using (SqlConnection con = new SqlConnection(CS))
        {
            using (SqlCommand cmd = new SqlCommand("select * from ProductDetails
where id='"+myid+"' ", con))
            {
                using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                {
                    DataTable dtBrands = new DataTable();
                    sda.Fill(dtBrands);
                    RepeaterProduct.DataSource = dtBrands;
                    RepeaterProduct.DataBind();
                }
            }
        }
    }
}

```

```

// Gallery.cs

public partial class Gallery : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindProductRepeater();
        }
    }

    private void BindProductRepeater()
    {
        string CS = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;

        using (SqlConnection con = new SqlConnection(CS))
        {
            using (SqlCommand cmd = new SqlCommand("select * from ProductDetails",
con))
            {
                using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                {
                    DataTable dtBrands = new DataTable();
                    sda.Fill(dtBrands);
                    RepeaterProduct.DataSource = dtBrands;
                    RepeaterProduct.DataBind();
                }
            }
        }
    }

    protected void TestDrive_Click(object sender, EventArgs e)
    {
        Button btn = (sender as Button);
        string id = btn.CommandArgument;
        // Session["id"] = id;
        Response.Redirect("Sell.aspx?id=" + id);
    }
}

```

```

//Car_Rent.cs

public partial class Rent : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindProductRepeater();
        }
    }

    private void BindProductRepeater()
    {
        string CS = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;

        using (SqlConnection con = new SqlConnection(CS))
        {
            using (SqlCommand cmd = new SqlCommand("select * from CarRent",
con))
            {
                using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                {
                    DataTable dtBrands = new DataTable();
                    sda.Fill(dtBrands);
                    RepeaterProduct.DataSource = dtBrands;
                    RepeaterProduct.DataBind();
                }
            }
        }
    }

    protected void ButtonOrderRent_Click(object sender, EventArgs e)
    {
        Button btn = (sender as Button);
        string id = btn.CommandArgument;
        // Session["id"] = id;
        Response.Redirect("Signup.aspx?id=" + id);
    }
}

```

```

// Search_Car.cs

public partial class Search : System.Web.UI.Page
{
    string val;
    protected void Page_Load(object sender, EventArgs e)
    {
        if((string)(Session["key"]) != null)
        {
            val = (string)(Session["key"]);

            string CS = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;

            string s = "";
            s += "'";
            s += val;
            s += "%";
            s += "'";
            using (SqlConnection con = new SqlConnection(CS))
            {
                using (SqlCommand cmd = new SqlCommand("select * from
ProductDetails where CarName like " + s, con))
                {
                    using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                    {
                        DataTable dtBrands = new DataTable();
                        sda.Fill(dtBrands);
                        RepeaterProduct.DataSource = dtBrands;
                        RepeaterProduct.DataBind();
                    }
                }
            }
        }

        Session["key"] = null;
    }
}

```

```

// Sell_Car.cs

public partial class Sell : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnSignup_Click(object sender, EventArgs e)
    {
        if (tbFullName.Text != "" && tbEmail.Text != "" && tbAddress.Text
        != "" && tbContactNumber.Text != "" && tbNid.Text != "" && tbDate.Text
        != "" )
        {
            string myid = Request.QueryString["id"];
            string orderRequest = "Pending";

            string CS = ConfigurationManager.ConnectionStrings
["CustomerInfoConnectionString"].ConnectionString;

            using (SqlConnection con = new SqlConnection(CS))
            {
                SqlCommand command = new SqlCommand("insert into TestdriveInfo
values('" + tbFullName.Text + "', '" + tbEmail.Text + "', '" + tbNid.Text
+ "', '" + tbContactNumber.Text + "', '" + tbAddress.Text + "', '"
+ tbDate.Text + "', '"+myid+"', '"+orderRequest+"')", con);
                con.Open();
                command.ExecuteNonQuery();
                Label2.Visible = true;
                Label2.Text = "Your request has been sent!";
                Label1.Visible = false;

                tbFullName.Text = "";
                tbAddress.Text = "";
                tbNid.Text = "";
                tbDate.Text = "";
            }
        }
        else
        {
            Label1.Visible = true;
            Label1.Text = "All fields are required!";
            Label2.Visible = false;
        }
    }
}

```



```

// Signup.cs

protected void btnSignup_Click(object sender, EventArgs e)
{
    if (tbFullName.Text != "" && tbEmail.Text != "" && tbAddress.Text
    != "" && tbContactNumber.Text != "" && tbNid.Text != "" && tbCars.Text
    != "" && tbDate.Text != "" && tbHour.Text != "")
    {
        string CS = ConfigurationManager.ConnectionStrings["CustomerInfoConnectionSt

        string orderRequest = "Pending";

        string myid = Request.QueryString["id"];

        using (SqlConnection con = new SqlConnection(CS))
        {
            SqlCommand command = new SqlCommand("insert into
RegistrationInfo values('" + tbFullName.Text + "', '" + tbEmail.Text
+ "', '" + tbNid.Text + "', '" + tbContactNumber.Text + "', '" + tbAddress.Text
+ "', '" + tbCars.Text + "', '" + tbDate.Text + "', '" + tbHour.Text +
"', '" + myid + "', '" + orderRequest + "')" , con);

            con.Open();
            command.ExecuteNonQuery();
            Label2.Visible = true;
            Label2.Text = "Your request has been sent!";
            Label1.Visible = false;

            tbFullName.Text = "";
            tbEmail.Text = "";
            tbContactNumber.Text = "";
            tbAddress.Text = "";
            tbNid.Text = "";
            tbCars.Text = "";
            tbDate.Text = "";
            tbHour.Text = "";

        }
    }
    else
    {
        Label1.Visible = true;
        Label1.Text = "All fields are required!";
        Label2.Visible = false;
    }
}

```

```

//ViewCarRentOrder.cs

protected void btnApprove_Click2(object sender, EventArgs e)
{
    string CS = ConfigurationManager.ConnectionStrings["CustomerInfoConnectionString"].ConnectionString;

    string a = "Approved";

    using (SqlConnection con = new SqlConnection(CS))
    {
        SqlCommand command = new SqlCommand("update RegistrationInfo
set OrderRequest = '" + a + "'where Id = '" + tbOrderId.Text + "'",
con);

        LabelApprove.Visible = true;
        LabelApprove.Text = "Order Approved!";

        con.Open();
        command.ExecuteNonQuery();

        //Fetching Settings from WEB.CONFIG file.
        string emailSender = "carrentsell2018@gmail.com";
        string emailSenderPassword = "12345678Nn";
        string emailSenderHost = "smtp.gmail.com";
        int emailSenderPort = 587;
        Boolean emailIsSSL = true;

        //Fetching Email Body Text from EmailTemplate File.
        string FilePath = "C:\\Users\\Nitol Das Neel\\Documents\\Visual
Studio 2017\\Projects\\CarRentSell2018\\CarRentSell2018\\EmailTemplates\\SignUp3.html";
        StreamReader str = new StreamReader(FilePath);
        string MailText = str.ReadToEnd();
        str.Close();

        //Repalce [newusername] = signup user name
        MailText = MailText.Replace("[newusername]", tbEmail.Text.Trim());

        string subject = "Welcome to Car Rent & Sell";

        //Base class for sending email
        MailMessage _mailmsg = new MailMessage();

```

```

//Make TRUE because our body text is html
_mailmsg.IsBodyHtml = true;

//Set From Email ID
_mailmsg.From = new MailAddress(emailSender);

//Set To Email ID
_mailmsg.To.Add(tbEmail.Text.ToString());

//Set Subject
_mailmsg.Subject = subject;

//Set Body Text of Email
_mailmsg.Body = MailText;

//Now set your SMTP
SmtpClient _smtp = new SmtpClient();

//Set HOST server SMTP detail
_smtp.Host = emailSenderHost;

//Set PORT number of SMTP
_smtp.Port = emailSenderPort;

//Set SSL --> True / False
_smtp.EnableSsl = emailIsSSL;

//Set Sender userEmailID, Password
NetworkCredential _network = new NetworkCredential(emailSender,
emailSenderPassword);
_smtp.Credentials = _network;

//Send Method will send your MailMessage create above.
_smtp.Send(_mailmsg);
    }
}

```

```

//ViewTestDriveOrder.cs

protected void btnApprove_Click(object sender, EventArgs e)
{
    string CS = ConfigurationManager.ConnectionStrings["CustomerInfoConnectionString"].ConnectionString;

    string a = "Approved";

    using (SqlConnection con = new SqlConnection(CS))
    {
        SqlCommand command = new SqlCommand("update TestdriveInfo
set OrderRequest = '" + a + "'where Id = '" + tbOrderId.Text + "'",
con);

        LabelApprove.Visible = true;
        LabelApprove.Text = "Order Approved!";

        con.Open();
        command.ExecuteNonQuery();

        //Fetching Settings from WEB.CONFIG file.
        string emailSender = "carrentsell2018@gmail.com";
        string emailSenderPassword = "12345678Nn";
        string emailSenderHost = "smtp.gmail.com";
        int emailSenderPort = 587;
        Boolean emailIsSSL = true;

        //Fetching Email Body Text from EmailTemplate File.
        string FilePath = "C:\\Users\\Nitol Das Neel\\Documents\\Visual
Studio 2017\\Projects\\CarRentSell2018\\CarRentSell2018\\EmailTemplates\\SignUp.html";
        StreamReader str = new StreamReader(FilePath);
        string MailText = str.ReadToEnd();
        str.Close();

        //Repalce [newusername] = signup user name
        MailText = MailText.Replace("[newusername]", tbEmail.Text.Trim());

        string subject = "Welcome to Car Rent & Sell";

        //Base class for sending email
        MailMessage _mailmsg = new MailMessage();

```

```

//Make TRUE because our body text is html
_mailmsg.IsBodyHtml = true;

//Set From Email ID
_mailmsg.From = new MailAddress(emailSender);

//Set To Email ID
_mailmsg.To.Add(tbEmail.Text.ToString());

//Set Subject
_mailmsg.Subject = subject;

//Set Body Text of Email
_mailmsg.Body = MailText;

//Now set your SMTP
SmtpClient _smtp = new SmtpClient();

//Set HOST server SMTP detail
_smtp.Host = emailSenderHost;

//Set PORT number of SMTP
_smtp.Port = emailSenderPort;

//Set SSL --> True / False
_smtp.EnableSsl = emailIsSSL;

//Set Sender userEmailID, Password
NetworkCredential _network = new NetworkCredential(emailSender,
emailSenderPassword);
_smtp.Credentials = _network;

//Send Method will send your MailMessage create above.
_smtp.Send(_mailmsg);

}

}

```