# Software Requirement and Design Specifications

***[MightyRides Online Car trading platform]***

***Version: [1.0]***

|  |  |
| --- | --- |
| *Course Code* | CS 3007 |
| *Instructor* | Miss Noureen Fatima |
| *Project Team* | Syed Muhammad Bilal(20K-0209)  Sohaib Akhter (20K-0292)  Syed Nofel Talha (20K-0151)  Abdul Wasay(20K-0300) |
| *Submission Date* | 5/May/2023 |

## [Instructions]

###### No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.

* *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.*

###### This’ Instruction’ section should also be removed in final document.

*Table of Contents*

1. [INTRODUCTION 5](#_bookmark0)
   1. [Purpose of Document 5](#_bookmark1)
   2. [Intended Audience 5](#_bookmark2)
2. [OVERALL SYSTEM DESCRIPTION 6](#_bookmark3)
   1. [Project Background 6](#_bookmark4)
   2. [Project Scope 6](#_bookmark5)
   3. [Not In Scope 6](#_bookmark6)
   4. [Project Objectives 6](#_bookmark7)
   5. [Stakeholders 6](#_bookmark8)
   6. [Operating Environment 6](#_bookmark9)
   7. [System Constraints 6](#_bookmark10)
   8. [Assumptions & Dependencies 6](#_bookmark11)
3. [EXTERNAL INTERFACE REQUIREMENTS 7](#_bookmark12)
   1. [Hardware Interfaces 7](#_bookmark13)
   2. [Software Interfaces 7](#_bookmark14)
   3. [Communications Interfaces 7](#_bookmark15)
4. [FUNCTIONAL REQUIREMENTS 8](#_bookmark16)
   1. [FUNCTIONAL HIERARCHY 8](#_bookmark17)
   2. [Use Cases 8](#_bookmark18)
      1. [[Title of use case] 8](#_bookmark19)
5. [NON-FUNCTIONAL REQUIREMENTS 9](#_bookmark20)
   1. [Performance Requirements 9](#_bookmark21)
   2. [Safety Requirements 9](#_bookmark22)
   3. [Security Requirements 9](#_bookmark23)
   4. [User Documentation 9](#_bookmark24)

[SDS 10](#_bookmark25)

1. [SYSTEM ARCHITECTURE 11](#_bookmark26)
   1. [SYSTEM LEVEL ARCHITECTURE 11](#_bookmark27)
   2. [SOFTWARE ARCHITECTURE 11](#_bookmark28)
2. [DESIGN STRATEGY 12](#_bookmark29)
3. [DETAILED SYSTEM DESIGN 13](#_bookmark30)
   1. [DATABASE DESIGN 13](#_bookmark31)
4. [APPLICATION DESIGN 15](#_bookmark32)
5. [REFERENCES 15](#_bookmark33)
6. [APPENDICES 17](#_bookmark34)

## Introduction

##### Purpose of Document

##### The purpose of this document is to provide a detailed view of the project we have been working on.

##### Intended Audience

##### The Course Instructors or any other person who is connected to this field or finds this project relevant to his needs.

[**Definition of Terms, Acronyms and Abbreviations**

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]

|  |  |
| --- | --- |
| ***Term*** | ***Description*** |
| *ASP* | *Active Server Pages* |
| *DD* | *Design Specification* |
| UCD | USE CASE DIAGRAM |
| URL | UNIFORM RESOURCE LOCATOR |
|  |  |
|  |  |
|  |  |
|  |  |

##### Document Convention

Font style: Arial.

Font size: 10

Font decoration: none.

For headings: (As given)

Font style: Arial.

Font size: 12

Font decoration: Italic, and Bold.

## Overall System Description

##### Project Background

##### The idea behind this project is to allow people to purchase cars of their choice online, , as well as it will allow customers to pre-book their orders for future (future scheduling). In addition to benefitting the customers, it makes job of car dealers easier, as they can list cars for sale within seconds.

##### Project Scope

##### The system has multiple User interfaces:

##### First interface is for the Dealer. The Dealer can sign up and create a personal account on the platform which he can use to list cars/car parts for sale and contact customers if they need help regarding any matter.

##### Second interface is the main interface the project focuses on: the customer’s interface. Here, a customer is allowed to create a customer account where he can explore all the cars on the platform or search cars and filter them, he can view their details and then proceed to buy it through a secure online transaction system.

##### Not In Scope

##### Not Applicable.

##### Project Objectives

##### This project will finally give customers to view listings from a diversity of dealers and get all purchases handled by secure transactions. In addition, customers can select how to customize all their products, which has not been a feature earlier.

##### It will also help people to work with us easily. It’s a free platform for car dealers to use to boost their sales.

##### These are the major objectives that were aimed and achieved by this project. Apart from that, the project helped us learn various strategies for solving the same problem.

##### Stakeholders

##### The customer who can login and use the system anytime once he has signed up on the website. The user can choose any category of car, any car dealer, and any cars/spare parts from the given list. In addition, he / she can also avail extended discounts if the car dealership is offering any. The user can also track their order/past orders.

##### Developer who developed the project, for further releases and new versions (Both front end, and back end).

##### Internal database engineer who manages the database working.

##### Software quality assurance engineer, who is responsible to test the environment for bugs, and errors.

##### Operating Environment

***Hardware platform:*** It needs basic hardware requirements but an operating system that supports a modern browser as this is a modern web application and the front end will be compiled by a compatible modern browser. Any average hardware will do the work. The minimum hardware we used to run this is mentioned below:

* Core 2 duo E75 / AMD Athlon 64 X2 5600+,
* 256 MB ram (hardly uses 128 MB of ram)

***Operating System:*** Windows 11/10/7/XP/Linux/Android/IOS any variant (Recommended)

Network Environment: Should have a decent internet connection.

***Applications:***

* Visual Studio Code
* XAMPP
* Php Myadmin

##### System Constraints

##### Software constraints:

It needs a browser that can compile html 5 and JS ES6.

##### Hardware constraints:

It needs basic hardware requirements. Any average system will do the work. The hardware we used to run this is mentioned below:

* Ryzen 7 5800h 16gb ram (hardly uses 256 mb of ram in a browser.

##### Cultural constraints:

##### The person should know how to read and understand basic English language.

##### Legal constraints:

##### Not applicable.

##### Environmental constraints:

* There are no environmental constraints. The system can be used anywhere anytime.
* The project is developed for people of all ages and ethnicities

##### User constraints:

##### No specific age constraints. Users of all ages can use the webpage.

##### Assumptions & Dependencies

##### Assumptions:

##### Customer and Dealers reside in Pakistan.

##### A car that has been sold will continue to show but will be labelled as sold

## External Interface Requirements

[This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.]

##### Hardware Interfaces

##### The system is perfectly supported by desktop computers. It cannot be supported on mobiles currently as it hasn’t been deployed to a domain.

##### The type of data we are taking from the user are login credentials, feedback, and we’re saving the information for the future.

Processor: Intel

Installed Memory: 2 GB or Higher

Speed: 1.40 GHz or Higher

##### Operating System: 32/64-bit Operating System

##### Software Interfaces

**Operating System:** Windows 10

**Database:** MySQL Workbench

**Webserver**: XAMPP

**Web Technologies**: HTMLL/ CSS/ JS/ PHP

**Ide and tools:** Visual Studio Code

##### Communications Interfaces

##### User email for signup should be valid.

##### Credit / debit card credentials input must be valid.

##### No encryption usage, or data transfer or synchronization issues.

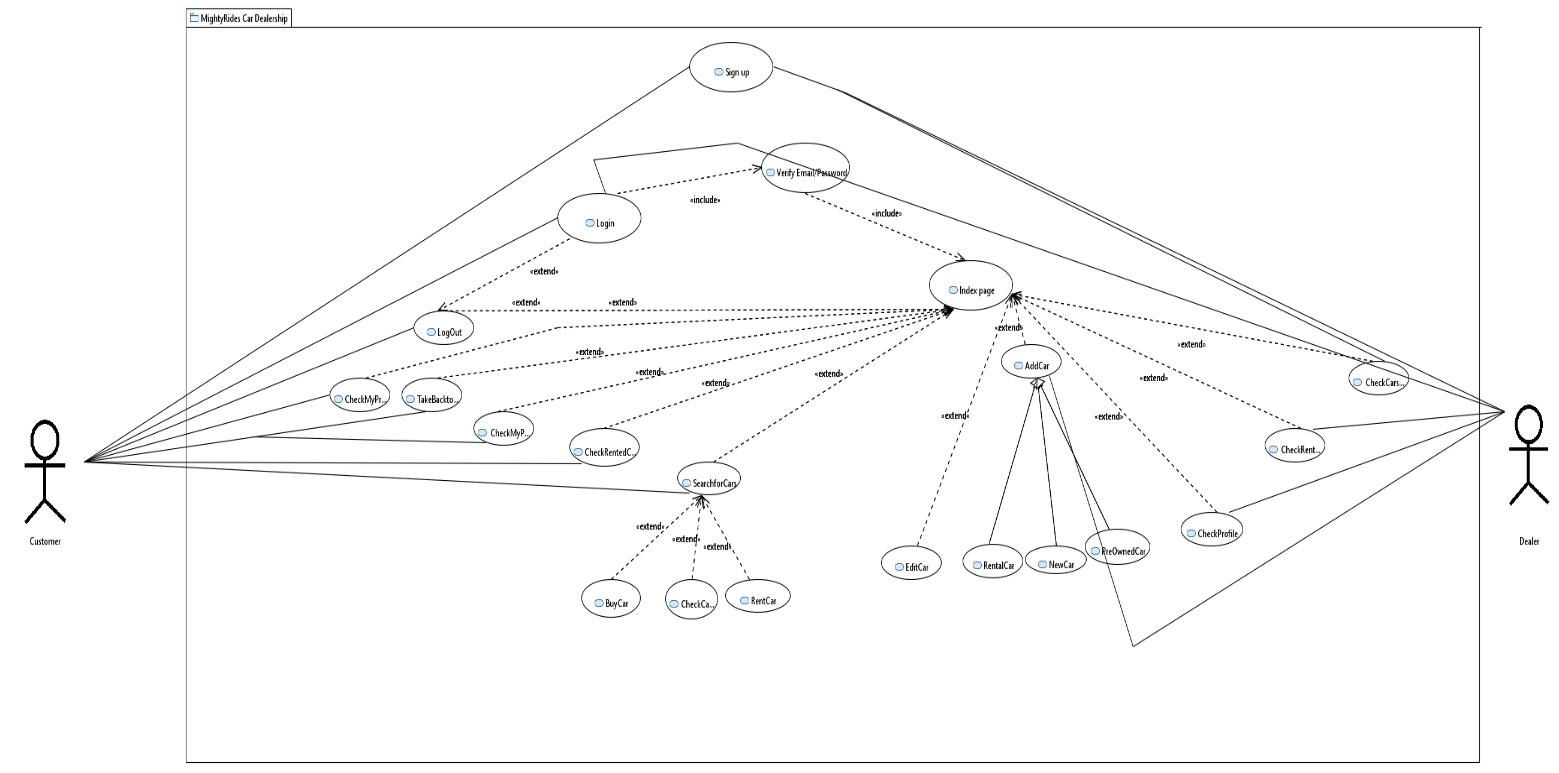
## Functional Requirements

##### Functional Hierarchy

[This section will give a big picture of overall system functionality. The main modules/features of system and their sub-functions will be described here in the form of a functional hierarchy so that, before getting into the use case, audience could grab the idea of overall system functions.]

##### Use Cases

***4.2.1. [MightyRides Online Car trading platform]***



|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Online Car trading platform |
| ***Use Case Description:*** | |
| ***Primary actor:*** customer | ***Other actors:*** Dealer |
| ***Stakeholders:*** customer, dealer, software developer, software designer, DBA |  |
| ***Relationships***   * ***Includes:***   view\_info includes login  store\_customer\_details includes signup  customize includes add\_items\_to\_cart  checkout includes add\_items\_to\_cart  calculate\_bill includes checkout   * ***Extends:***   Signup extends from login | |
| ***Pre-conditions:***  customer must have an adequate internet connection, and a desktop computer to run the webpage, and he should signup before moving ahead. | |
| ***Flow of Events for Customer:***   1. user signs up. 2. user logins. 3. user searches for/ explores car. 4. System displays categories. 5. User selects one. 6. User Views Car details/Dealer info. 7. User selects one. 8. User clicks buy. 9. System asks for confirmation and proceeds with transaction. 10. User selects payment method. 11. User enters info. 12. System verifies information. 13. transaction proceeds and an advance fee is paid, and the car is booked. 14. ***Customer receives date to go and inspect/pick up the car.***   ***Flow of Events for Dealer:***   1. user signs up. 2. user logins. 3. user lists a car on the website. 4. If that particular is bought than the dealer is temporarily placed in a open chat with the user. | |
| ***Alternative and exceptional flows:***  1. If login user doesn’t match, the information is taken again and again.  2. if the customer doesn’t checkout, and he logs out, his cart items are lost.  3. If order is scheduled for future, it cant be tracked. | |
| ***Post-conditions:***  -user registered.  - car manufacturer must exist in database  - | |

## Non-functional Requirements

##### Performance Requirements

1. Performance wise, our project is built to be very responsive and fast. The transitions between the interfaces take no time.
2. Capacity wise, our system is very storage friendly, hardly requires some MBs of data.
3. Safety wise, the data of users, interacting with our system stays safe and can only be accessed by the system owner.
4. The software is reliable in a sense that it fulfills all the needs that it is promised to fulfill. It was tested for any sort of bugs/ issues and was fixed by the developers eventually.

##### Safety Requirements

1. We took extra care that our system must not cause any damage on the machine on which the user is running our system.
2. The only thing that the user should take care of is the entry of dummy (fake) results in the database. Dummy data must be deleted from the database.

##### Security Requirements

1. External users such as someone out of the organization must not be given access to the system’s Admin panel. Login ensures this.
2. Only the Stakeholders should have access to the system.
3. The data of the user stays safe and untouchable. So, privacy is maintained.

##### User Documentation

User manuals, will be provided alongside and they will be explained how to use the system, first they are asked to sign-in if not already and then login to continue using the system, they can select their order from menu and can customize the order.

# SDS

## System Architecture

## The architecture embodies the major static and dynamic aspects of a system. It is a view of the whole system highlighting the important characteristics and ignoring unnecessary details. In the context of our approach, architecture is primarily specified in terms of views of tier architecture which is a client-server architecture in which the presentation, the application processing and data management are logically separate processes.

### *System Level Architecture*

### *COMPONENT DIAGRAM*

### 

### *DEPLOYMENT DIAGRAM*

### 

### *Software Architecture*

### *User Interface Layer:* not applicable

### *Middle tier:* not applicable

### *Data access Layer:* not applicable

## Design Strategy

## System Reuse: the webpage may be deployed on a world wide web, allowing users from all around the world to access the website.

## Future system extension or enhancement: we aim to add additional features to the webpage, such as adding chat bot using Machine learning, and deep learning, as well as AI generated user preferences.

## User interface paradigms: the program prompts the user to enter their information at multiple stages: while signing up, logging in, as well as during checkout. Moreover, the UI also displays the user information at multiple pages, such as displaying the order summary, or menu items.

## Concurrency and synchronization: only one user is allowed to use the system at a time.

## Detailed System Design

## Class Diagram:

## 

## Functions:

## Customer:

## Signup – takes customer information, to register customer as a user.

## Login - takes customer information and verify the information from the database. If the information matches, customer information is loaded, which is used when the customer places order. It prevents asking for repeated information from the user.

## Add\_to\_cart – the customer adds car parts to his cart, this option is only available for car parts and not for cars themselves. product id is stored in the customer’s suborder.

## AddCustomization – allows the customer to customize his order.

## Checkout – allows the customer to finalize his / her cart, displays order summary, and places the order.

## Search – allow customer to search cars on various filters and requirements.

## Order:

## Display – shows order summary.

## Calculate\_amount – calculates the final amount to be payed, after the discounts If offered by dealer.

## Checkout – order finalized, and placed.

## Card\_info:

## Add – add credit / debit card information.

## Update - Update credit / debit card information.

## Verify – verifies card credentials.

## Billing:

## Generate\_bill – generates bill with payable amount, and order summary.

## Get\_cust\_info – fetches customer data from customer id to use the information in printing the bill.

## Feedback:

## Get\_review – allows customer/dealer to add a review regarding the working of the system so the developers can further enhance the experience.

## Dealer:

## Register – allows Dealer to register inputting some basic credentials.

## Add\_car – allows Dealer to list rental/new/preowned cars.

## History:

## View– allows user to check their bought/sold/rented history.

## Time track – keeps track of rental times and informs customers on when they have to return cars..

## Category:

## Cartype : allows customer to select which type of car he is looking for.

## Car parts:

## Add – add car parts to cart.

## Update – update car parts.

## Car Compare:

## Compare–user can compare cars side by side to get a better comparison of them , helps in making better decisions..

## Suborder:

## Update\_order – add items to order, as cart is updated.

## Admin:

## Login – takes login credentials from the admin, verifies it, and then allows access to the functions.

## Check\_info – checking reviews.

### *Database Design*

#### ER Diagram

#### Diagram Description automatically generated

#### Data Dictionary

Not Applicable

### *Application Design*

#### Sequence Diagram

Diagram

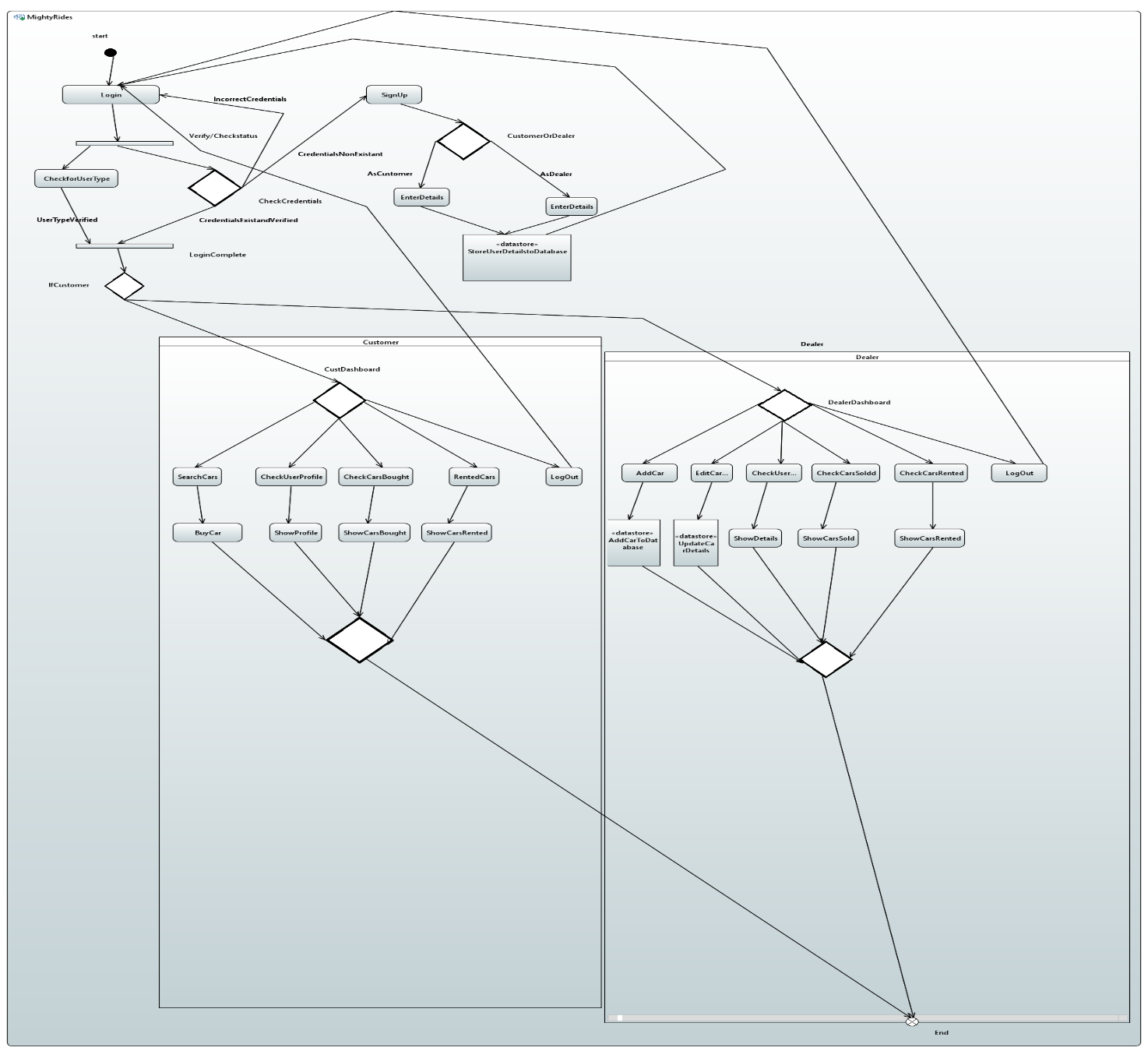
Description automatically generated

**Explanation:**

Explanation: this sequence diagram is for our online car trading platform. The webserver displays the user a form, in which the user enters his details for signup: his username, contact number, address, city, zip code, payment method, and then the user proceeds log in. All the inputted details are stored in the server, and this information is then used later on when either the customer proceeds to purchase a car or the dealer sells a car.

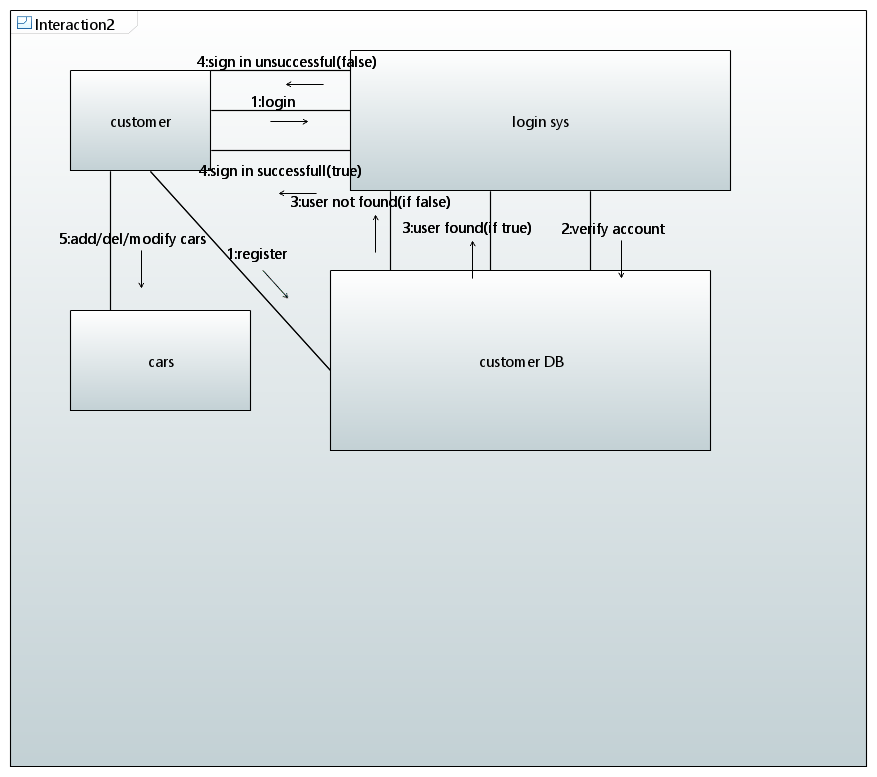
.

#### Activity Diagram

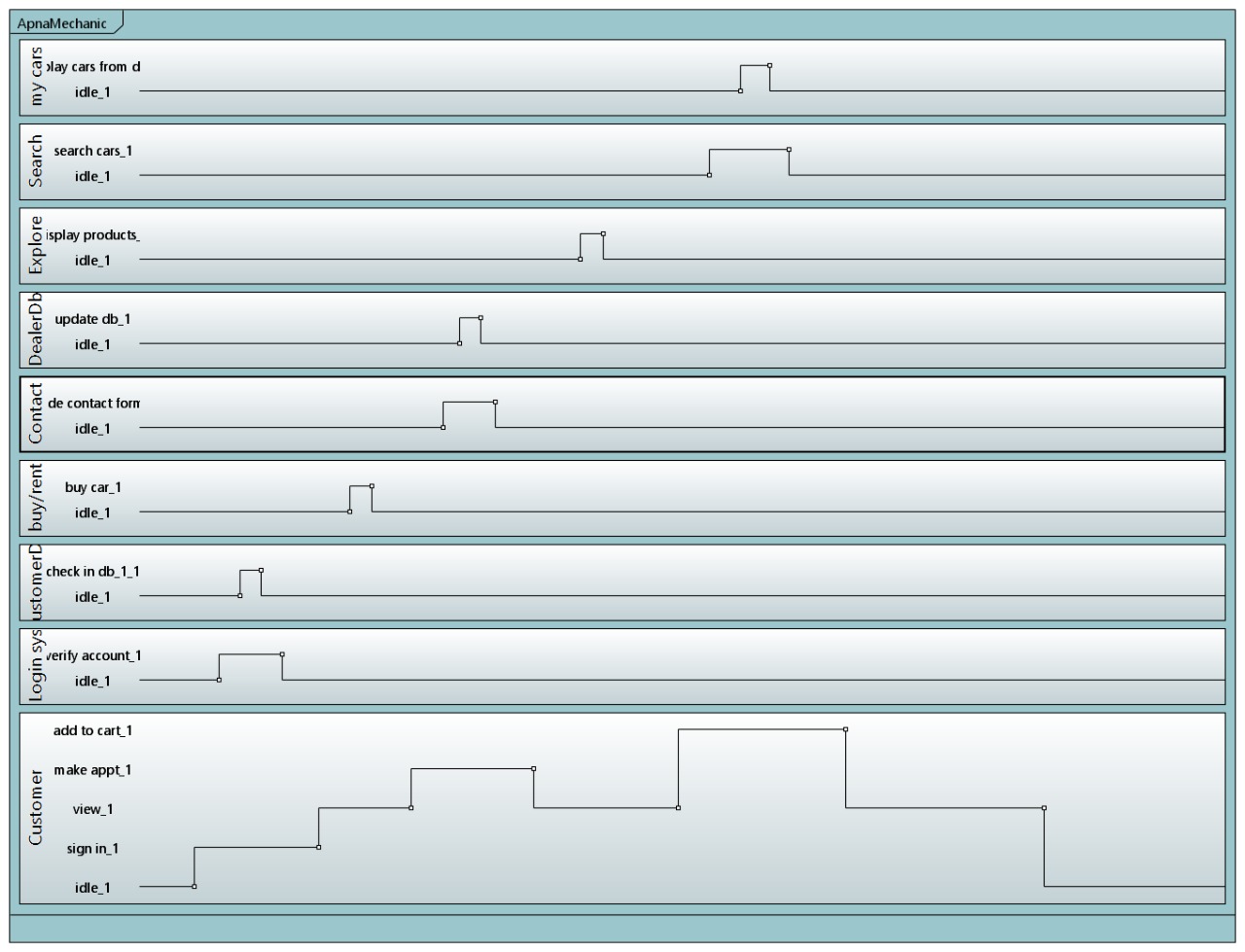


Explanation: This activity diagram summarizes the working of the whole system, starts with the sign up and login features, if an account exists it will log the user in after verifying that the details are correct by searching them up from the customers and dealers’ tables in the database and then matching them. If the account does not exist, then it’s obvious that you must sign up first. The swim lanes show how the functionality changes between customer and dealers. Upon reaching the dashboard all both type of users has their own respective optional activities to execute.

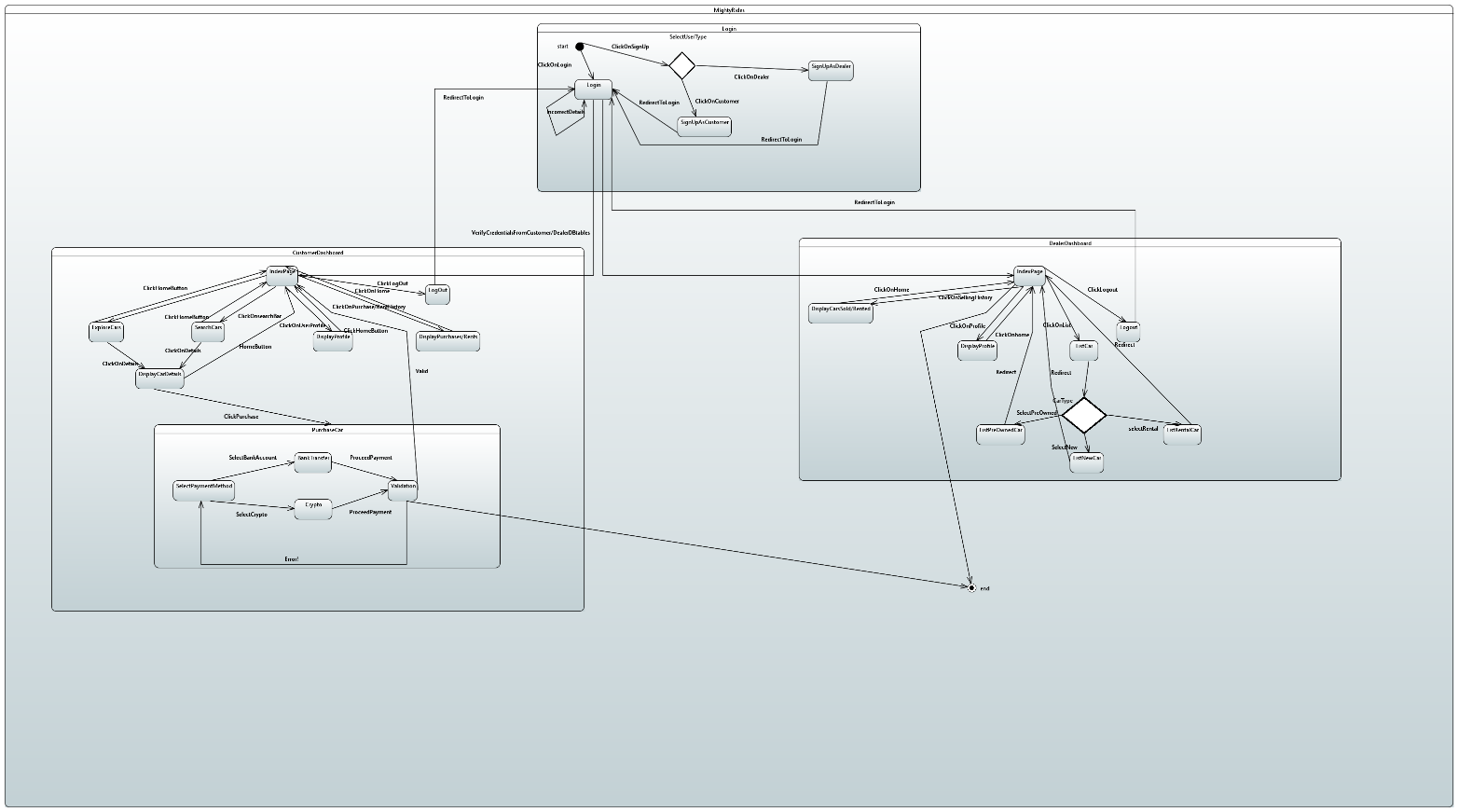
**Collaboration Diagram:**



**Timing Diagram:**



**State Machine Diagram:**

****

## References

https://www.php.net/docs.php

https://getbootstrap.com/docs/3.3/getting-started/

https://www.apachefriends.org/docs/

## Appendices

Not Applicable.