# Task A

## System Requirement Specification for the GoCheeta Cab Service.

## 1.1 **Purpose**

The Functions and the Nonfunctions are determined by the SRS Document. SRS Document helps people who code and who don’t know to code understand the system very easily. When someone is handling a project, they check the SRS to check the functions in the system.

## 1.2 Requirements

## 1.2.1 Functional Requirements

|  |  |  |
| --- | --- | --- |
|  | **User Requirements** |  |
| **Admin** | **Customer** | **Driver** |
| Login | Register | Login |
| Add Drivers | Login | View Hires |
| View Users | Book a Driver | Accept Hires |
| View Bookings | View the Booked Hire | View Accepted Hires |
| View Vehicles | Logout | Logout |
| Manage User (Edit/Delete) | - | - |
| Manage Bookings (Edit/Delete) | - | - |
| Manage Vehicles (Edit/Delete) | - | - |
| Add Vehicles | - | - |
| Logout | - | - |

## 1.2.2 Non-Functional Requirements

This web Site is made for performance and security and to make day to day life easy for drivers and customer and the users can navigate through the site very easily the useability is very clean and there is no lag or any brakes in the system and it is very securable.

# Task B

## 2.0 UML Diagrams for GoCheeta Online cab service system

## 2.1 Use Case Diagram

### 2.1.1 Customer Use Case

Diagram

Description automatically generated

This UML Diagram shows all the access points of a customer as it shown the customer can see the home page and register as a customer and then login to the system after the login the customer can Add Bookings and see his/her own bookings and log out of the system.

### 2.1.2 Driver Use Case

Diagram

Description automatically generated

This UML Diagram shows all the access points of a driver as it shown the driver can see the home page and the driver can not register admin can add drivers and then drivers can login view hires accept hires and then view accepted hires and lastly logout of the system.

### 2.1.3 Admin Use Case

Diagram

Description automatically generated

This UML Diagram shows all the access points of a admin as it shown the admin can see the home page Admin is hard coded to the system so their will be only one admin witch can login manage users and manage vehicles and manage bookings after before that as I said driver can not register so admin will be adding the driver to the system. The admin will be adding vehicles and lastly logout of the system.

## 2.2 Sequence Diagrams

### 2.2.1 Customer Sequence

### 2.2.2 Driver Sequence

### 2.2.3 Admin Sequence

Graphical user interface, application, table

Description automatically generated

# Task C

## Design Patterns

There are 3 types of design patterns

1. Creational
2. Structural
3. Behavioral

A design pattern offers an all-encompassing, reusable answer to the typical issues that arise in software design. Usually, the pattern demonstrates the connections and interactions between various classes or objects.

### Creational

Creational design patterns in software engineering deal with object creation mechanisms and attempt to generate objects in a situation-appropriate way. Two main concepts make up creational design patterns. One is storing information about the concrete classes the system use. Another is to conceal the process of creating and combining instances of these concrete types.

Apply Creational Patterns when.

* A system should be unconstrained by the processes used to make its constituent parts.
* A group of associated items is intended to be utilized collectively.
* A class library or product's implementations are concealed, leaving only their interfaces visible.
* Building various representations of separate complex items.
* A class expects the object it creates to be implemented by its subclass.
* At run-time, the class instantiations are specified.
* There must be just one instance, and clients must always have access to it.
* The instance should be able to grow without changing. (Gupta, 2022)

Text

Description automatically generated

### Structural

A structural design pattern is a guide for combining various objects and classes to create a larger structure that can accomplish several objectives at once. The patterns in structural designs demonstrate how distinctive system components can be put together in a flexible and extensible way. Therefore, we can target and modify a specific portion of the structure with the use of a structural design pattern without altering the entire structure. By outlining an easy method to construct entity relationships, structural design patterns facilitate the design process. Different kinds of structural design patterns exist. Adaptor, bridge, composite, and proxy are among examples.

Text

Description automatically generated

Types of Structural design patterns

These are only few of the structural design patterns that are use commonly

* Adapter Pattern
* Bridge Pattern
* Composite Pattern
* Decorator Pattern
* Facade Pattern
* Flyweight Pattern
* Proxy Pattern

### Behavioral

Behavioral design patterns are patterns that recognize typical communication patterns between objects and implement them in software engineering. These patterns boost this communication's flexibility by doing so.

Text

Description automatically generated

# Task D

## Git Hub Link:

[emalyaditha/GoCheeta (github.com)](https://github.com/emalyaditha/GoCheeta)

# Task E

## Test Plan - Customers

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario | Test Data | Expected Results | Actual Results |
| Enter your details in register form | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  First Name – Kirito  Last Name – Kazuto  Password – 11111  Phone No – 1111111111 | Customers Login Page | Customers Login Page |
| Enter Correct Email and Password | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  Password - 11111 | Show Customer Home page | After entering correct email and password it shows a bootstrap link after coming back and trying again will send you to the Customer Home page. (Don’t know the reason for the error) Most probably it occurs because of version issues |
| Enter Incorrect Email and Password | Email – [kirito11@gmail.com](mailto:kirito11@gmail.com)  Password - 00000 | Load the Login page again | Load the Login page again |
| Click Customer Add Booking | - | Show Add Booking Page | Show Add Booking Page |
| Customer Add Booking | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  Destination – Delkanda  Starting Location – Wijerama  Vehicle Branch – Nugegoda  Vehicle Name - BMW | Booking Successfully Added | Booking Successfully Added |
| Click My Bookings | - | Show his/her past booking. | Show his/her past booking. |

## Test Plan – Driver

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario | Test Data | Expected Results | Actual Results |
| Enter Correct Email and Password | Email – [mac@gmail.com](mailto:mac@gmail.com)  Password - 11111 | Show Driver Home page | After entering correct email and password it shows a bootstrap link after coming back and trying again will send you to the Driver Home page. (Don’t know the reason for the error) Most probably it occurs because of version issues |
| Click View Hires | - | Show all Hires Page | Show all Hires Page |
| Click Accept a hire | Email – mac@gmail.com | Add the hire to My Hires | Add the hire to My Hires |
| Click View My Hires | - | Show My Hires page | Show My Hires page |

## Test Plan – Admin

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario | Test Data | Expected Results | Actual Results |
| Enter Correct Email and Password | Email – [admin@gmail.com](mailto:admin@gmail.com)  Password - 11111 | Show Admin Home Page | Show Admin Home page (Admin is hardcoded to the system) |
| Click Add Driver | - | Show Add Driver Page | Show Add Driver Page |
| Add a Driver | Email – [asuna@gmail.com](mailto:asuna@gmail.com)  First Name – Asuna  Last Name – Yuuki  Password – 11111  Phone No – 0000000000  Branch – Nugegoda  Car - Saab | Successfully Add Driver | Successfully Add Driver |
| Add a Vehicle | Number Plate -FAF 1111  Model – Eclipse  Name - Mitsubishi | Successfully Add Vehicle | Successfully Add Vehicle |
| Click Admin Manage User | - | View Admin Manage User Page with All User Details | View Admin Manage User Page with All User Details |
| Click the link to Update User | Email – [asuna@gmail.com](mailto:asuna@gmail.com)  First Name – Asuna  Last Name – Yuuki  Password – 00000  Phone No – 0701536969  Branch – Nugegoda  Car – Saab  Role - Driver | Successfully Updated User | Successfully Updated User |
| Click the link to Delete User | Id – U13 | Successfully Deleted User | Successfully Deleted User |
| Click Admin Manage Booking | - | View Admin Manage User Page with All Booking Details | View Admin Manage User Page with All Booking Details |
| Click the link to Update User | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  Destination – Delkanda  Starting Location – Wijerama  Vehicle Branch – Nugegoda  Vehicle Name - BMW | Successfully Updated Booking | Successfully Updated Booking |
| Click the link to Delete Booking | Id - B13 | Successfully Deleted Booking | Successfully Deleted Booking |
| Click Admin Manage Vehicle | - | View Admin Manage Vehicle Page with All Vehicle Details | View Admin Manage Vehicle Page with All Vehicle Details |
| Click the link to Update Vehicle | Number Plate -FAF 0001  Model – Eclipse  Name - Mitsubishi | Successfully Updated Vehicle | Successfully Updated Vehicle |
| Click the link to Delete Vehicle | Id - 13 | Successfully Deleted Vehicle | Successfully Deleted Vehicle |
| Click logout | - | Send back to the index | Send back to the index |