Software Requirements Specification

Version 4.2

WEB-BASED CAR RENTAL BROKER MANAGEMENT SYSTEM

Prepared by

Name	Student-id	Email
Kiranmayie	40092284	2809kiran@gmail.com
Rajasekhar	40094479	rajasekhar.grr@gmail.com
Sahana	40092026	Sahana15shankar@gmail.com
Nandini	40105415	nandu.angel555@gmail.com

Instructor:	Dr.Constantinos Constantinides
Course:	Software Design Methodologies (SOEN 6461-SS)
Date:	09/18/2019

Document history

Date	Version	Description	Author
09/15/2019	1.0	Introduction	Sahana Shankar
10/04/2019	1.1	Overall Description Specific requirements Analysis model	Rajasekhar Reddy Guntaka
10/24/2019	2.0	Admin Functionality	Rajasekhar Reddy Guntaka
10/27/2019	3.0	Admin UML	Nandini Bandlamudi
11/16/2018	4.0	Operation Contracts	All Team Members
11/16/2019	4.1	Use Case Scenarios	All Team Members
11/18/2019	4.2	UML diagrams and system operations	All Team Members

Table of Contents

Table of contents

1. Introduction

This Software Requirements Specifications Document provides an overview about the application that is developed- Web Based Car Rental Broker Management. We are following set rules, Provides a clear understanding of Functional and Non - Functional Requirements that helps us designing a sustainable system that supports booking a car for the customer with their requirements.

The document provides the overall description, product features, characteristics, Constraints, Assumptions and Dependencies. The project is divided into four iterations, and described sequentially for every iteration.

Purpose

The purpose of Software Requirements Specifications Document is to understand what are the requirements of Web Based Vehicle Management system. The main aspect is to identify all the requirements from the users and then designing part starts. Functional and non-functional requirements are defined in this document.

Scope

The scope of this document is to explain all the requirements in detail to the users. This works as a communication between developers and the users. It focuses mainly on how the vehicles are booked, how the client records and vehicle records are managed. This document explains how the vehicles are renting, reserved and returned to the users

Definitions, acronyms, and abbreviations

Provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Software Requirements Specifications Document.

Project's Glossary

SRS	Software Requirement Specification
UC	Use Case
OP	Operational Contract
UML	Unified Modeling Language
CRBM	Web Based Vehicle Management System

References

2. Overall description

The overall description of project specifies about detailed view of the application ie. Web based Car Rental Broker Management System that will be accessed by the users. The overall project is made in four iterations as per the stakeholders requirements.

Iteration 1: Environment setup

Making the teams and inviting the markers.

Iteration 2: Clerk functionality

Iteration 3: Admin functionality

Iteration 4: The system supports concurrency and persistence.

Product perspective

Web based car rental Broker Management system is a web application. The main purpose of this application is to provide the vehicle company to rent the vehicles to customers based on some certain requirements. This application will run on device that connected with the internet.

Product Functions

This application has various functions, As there was different functionalities in it they varies from each other. There are 3 primary actors 1. Clerk 2. Client and 3. Admin

1. Registration of customers should be done.
2. Reserve a vehicle to the user as per the start date and time.
3. Clerk should be able to manage the client data, Clerks additionally choose the order by which they view a result set: random order or sort according to some criteria.
4. Clerk can able to view the bookings and availability of vehicle.

User characteristics

Client

Client uses the web based car rental system for vehicle registration, return vehicle. Client need to register with the system.

Clerk

Clerk is the systems employee who can manage a client record(create new, modify, delete), create a rental of a vehicle for a client, create or cancel a reservation of the vehicle for a given client, or handle the return of the given vehicle. Clerks should be able to view the contents of the catalog, either in a random order or by creating a result set through a selection of filtering criteria. Clerks may additionally choose the order by which they view a result set: random order, or sort according to some criteria. The clerk can subsequently proceed to next item in detail view or go back to the initial result set view.

Administrator

Admin is responsible for managing vehicle record (create new, modify, delete). Admin can additionally view the contents of the catalog and perform search history of per client or per vehicle or per due date but cannot perform a booking or return vehicle. Admin should have idea on how application is working and should know the functionalities of the system.

Constraints

User Interface The client can be able to open the application in any browser. Interface to be implemented in java.

Safety The data of the clients to be kept in a safer place so that it cannot be accessed by outsiders.

Security Administrator should monitor all the reservations that have been made and the data should not be shared.

Assumptions and Dependencies

Assumptions	Dependencies
This application to be used in web browser only because it is designed for PC	Almost all the clients should have to use only PC's.
The PC should have an active internet connection	So that clerks can modify the changes if required.
Conform booking of the vehicle for the required date.	It requires the availability of the vehicle for that specific date and time

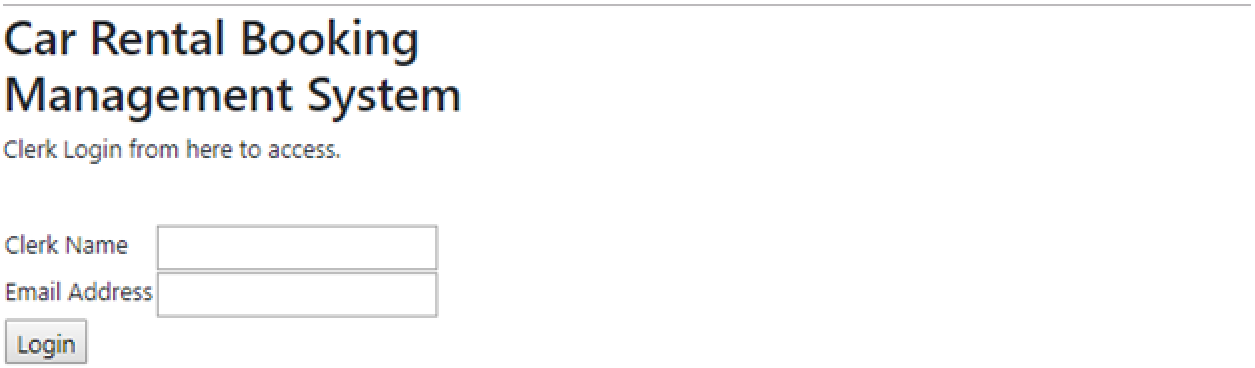
3. Specific requirements

All the requirements are classified to Functional as well as Non - Functional requirements(quality attributes and constraints). The quality attributes are listed according to the ISO/IEC 25010 standard that classifies software quality in a structured set of characteristics and sub-characteristics.

3.1 External Interfaces

Below are the files according to which design should be made and get started by the development.

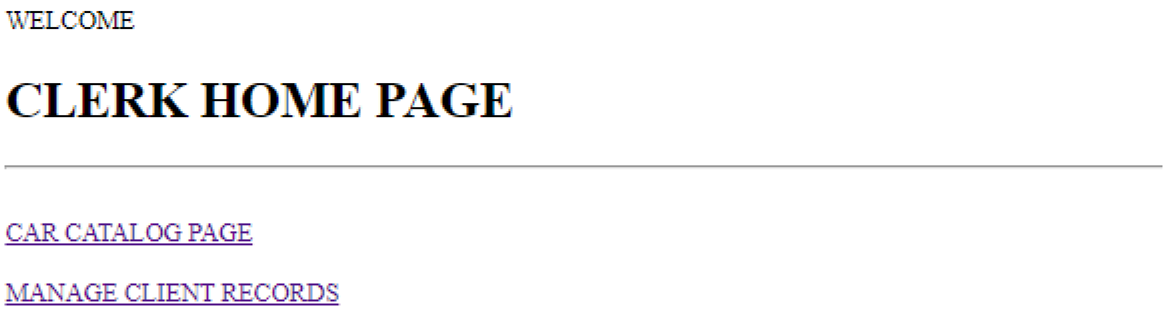
Clerk Login page



Car Rental Booking Management System

Clerk Login from here to access.

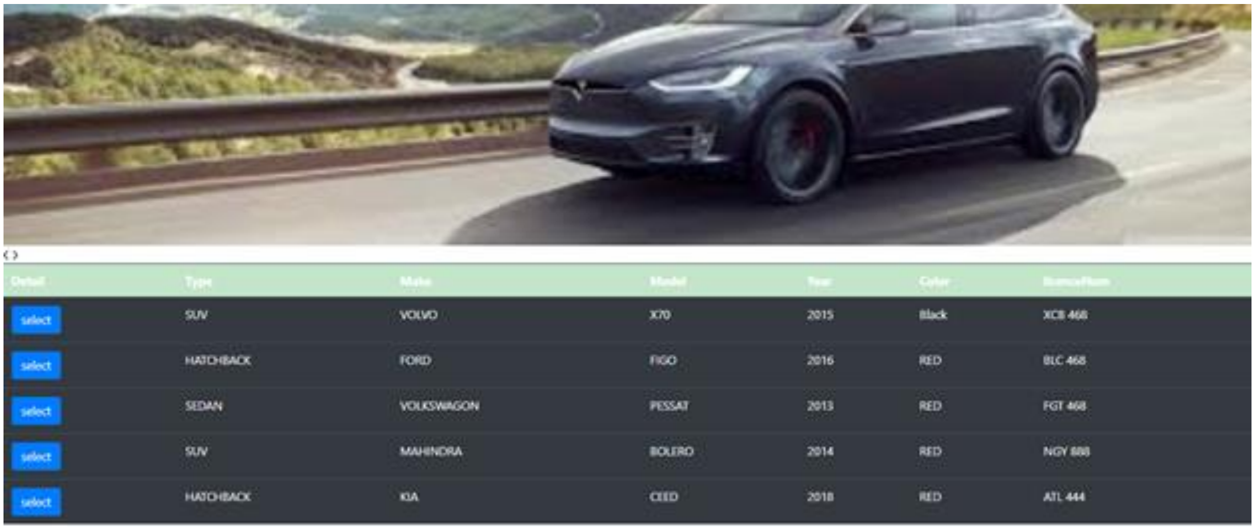
Clerk Home page



WELCOME

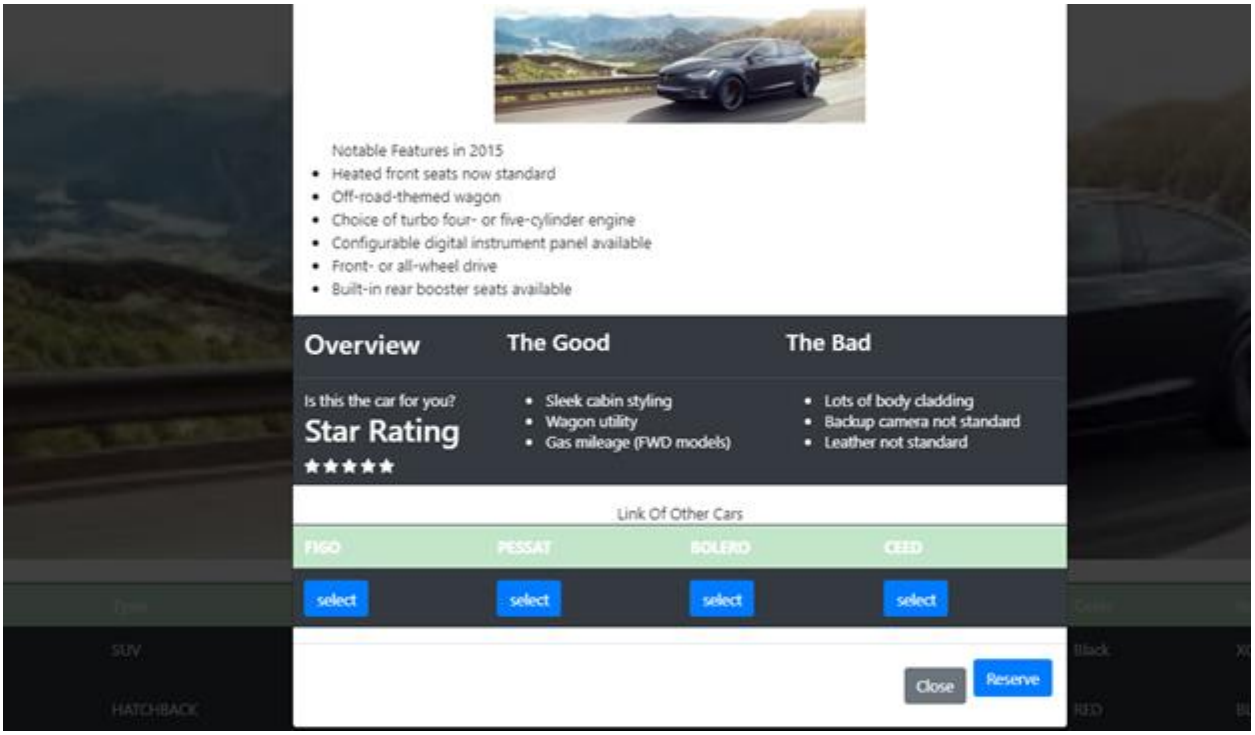
CLERK HOME PAGE

Car Catalog page



Detail	Type	Make	Model	Year	Color	licenceNum
select	SUV	VOLVO	X70	2015	Black	XCB 468
select	HATCHBACK	FORD	FIGO	2016	RED	BLC 468
select	SEDAN	VOLKSWAGON	PESSAT	2013	RED	FGT 468
select	SUV	MAHINDRA	BOLERO	2014	RED	NGY 888
select	HATCHBACK	KIA	CEED	2018	RED	ATL 444

Reservation



Notable Features in 2015

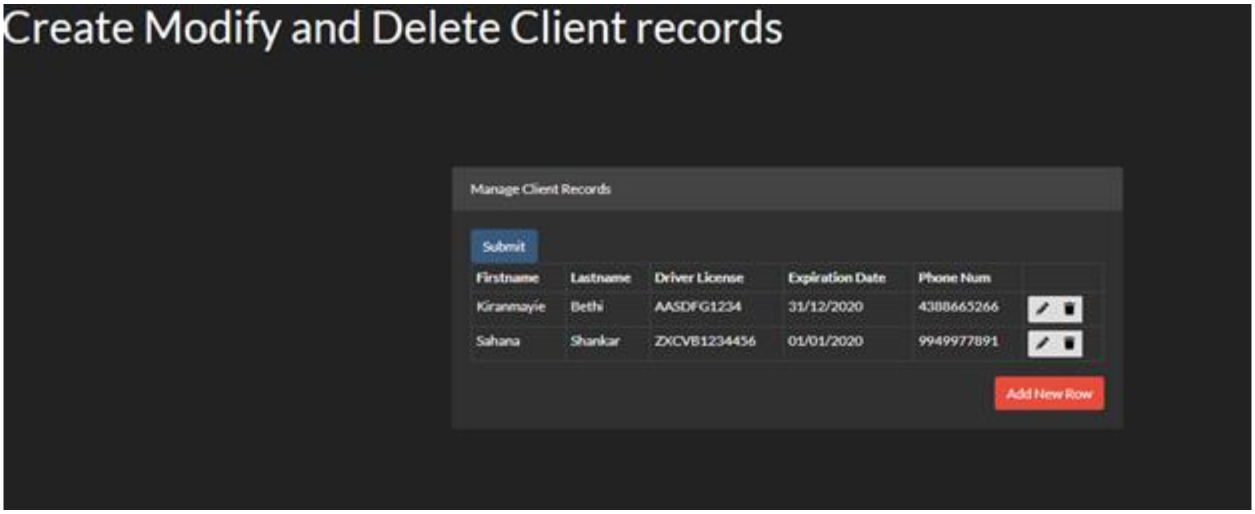
- Heated front seats now standard
- Off-road-themed wagon
- Choice of turbo four- or five-cylinder engine
- Configurable digital instrument panel available
- Front- or all-wheel drive
- Built-in rear booster seats available

Overview	The Good	The Bad
Is this the car for you? Star Rating ★★★★★	• Sleek cabin styling • Wagon utility • Gas mileage (FWD models)	• Lots of body cladding • Backup camera not standard • Leather not standard

Link Of Other Cars

FIGO	PESSAT	BOLERO	CEED
select	select	select	select

Managing client records

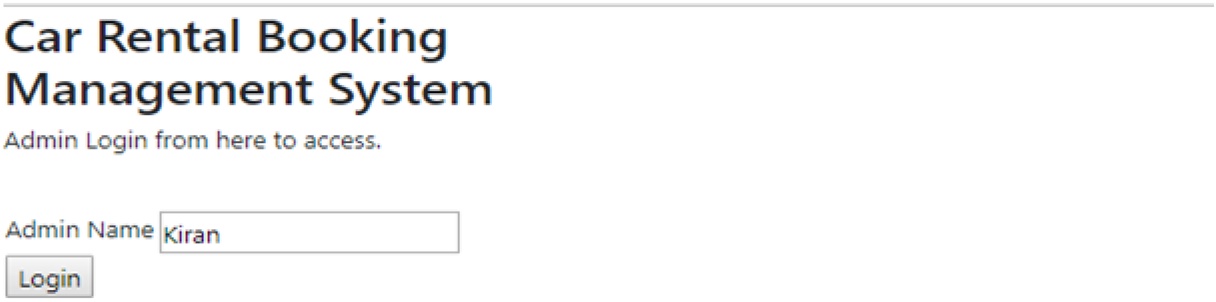


Create Modify and Delete Client records

Manage Client Records

Firstname	Lastname	Driver License	Expiration Date	Phone Num	
Kiranmayie	Bethi	AASDFG1234	31/12/2020	4388665266	✎ 🗑
Sahana	Shankar	ZXCVB1234456	01/01/2020	9949977891	✎ 🗑

Admin Login page

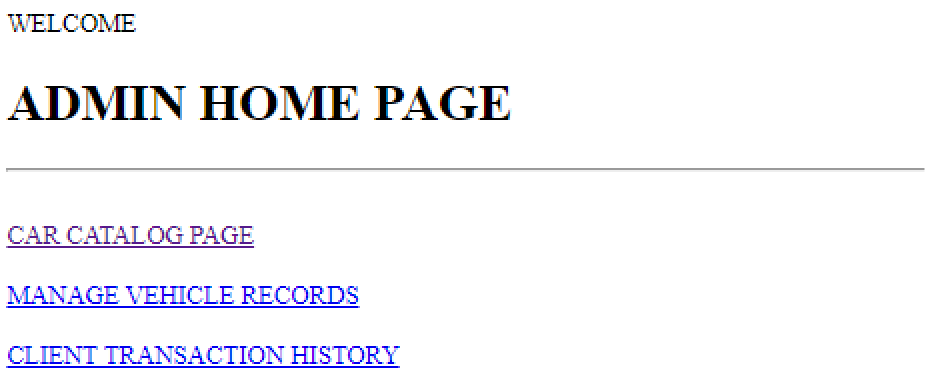


Car Rental Booking Management System

Admin Login from here to access.

Admin Name: Kiran

Admin Home page



WELCOME

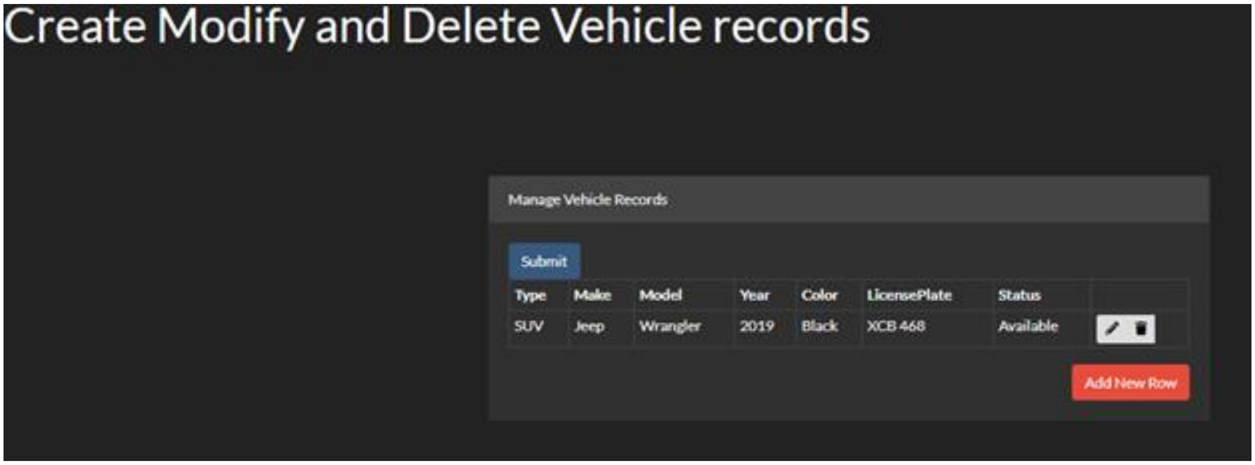
ADMIN HOME PAGE

CAR CATALOG PAGE

MANAGE VEHICLE RECORDS

CLIENT TRANSACTION HISTORY

Vehicle records

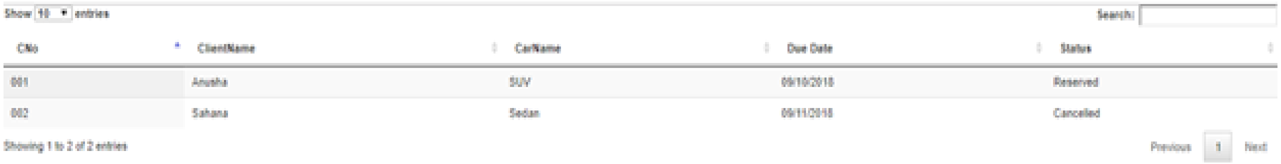


Create Modify and Delete Vehicle records

Manage Vehicle Records

Type	Make	Model	Year	Color	LicensePlate	Status	
SUV	Jeep	Wrangler	2019	Black	XCB 468	Available	✎ 🗑

Client Transaction History



CNo	ClientName	CarName	Due Date	Status
001	Anusha	SUV	09/10/2018	Reserved
002	Sahana	Sedan	09/11/2018	Cancelled

Showing 1 to 2 of 2 entries

3.2 Functional requirements

Functional requirements defines a function of a system or its function.

Viewing and searching the catalogs

Clerks should be able to view the contents of the catalog, either in a random order, or by creating a result set through a selection of filtering criteria, e.g. “View all SUV models not more than 3 years old .” Clerks may additionally choose the order by which they view a result set: random order, or sort according to some criteria, e.g. “View sorted by year.” From a given result set, a clerk may choose an item to view in detail. In this view, the system should provide an indication of whether the vehicle is available or is rented out. The clerk can subsequently proceed to the next item in detail view, or go back to the (possibly filtered) initial result set view.

Renting reserving and returning vehicles

A clerk can manage a client record (create new, modify, delete), create a rental of a vehicle for a given client, create or cancel a reservation of a vehicle for a given client, or handle the return of a given vehicle. The system maintains a record for every transaction (rental, reservation, or return) that includes client and vehicle information, together with a timestamp. In the case of a rent or a reservation, the record will include the due date for each item. Administrators can access, view and search the history of transactions per client, per vehicle or per due date. For example, ‘Show all vehicles currently out’, ‘When is a given vehicle due’, ’Is a given vehicle available on a given date or over certain dates?’, or ‘What vehicles were due yesterday(and they have all been returned)?.’

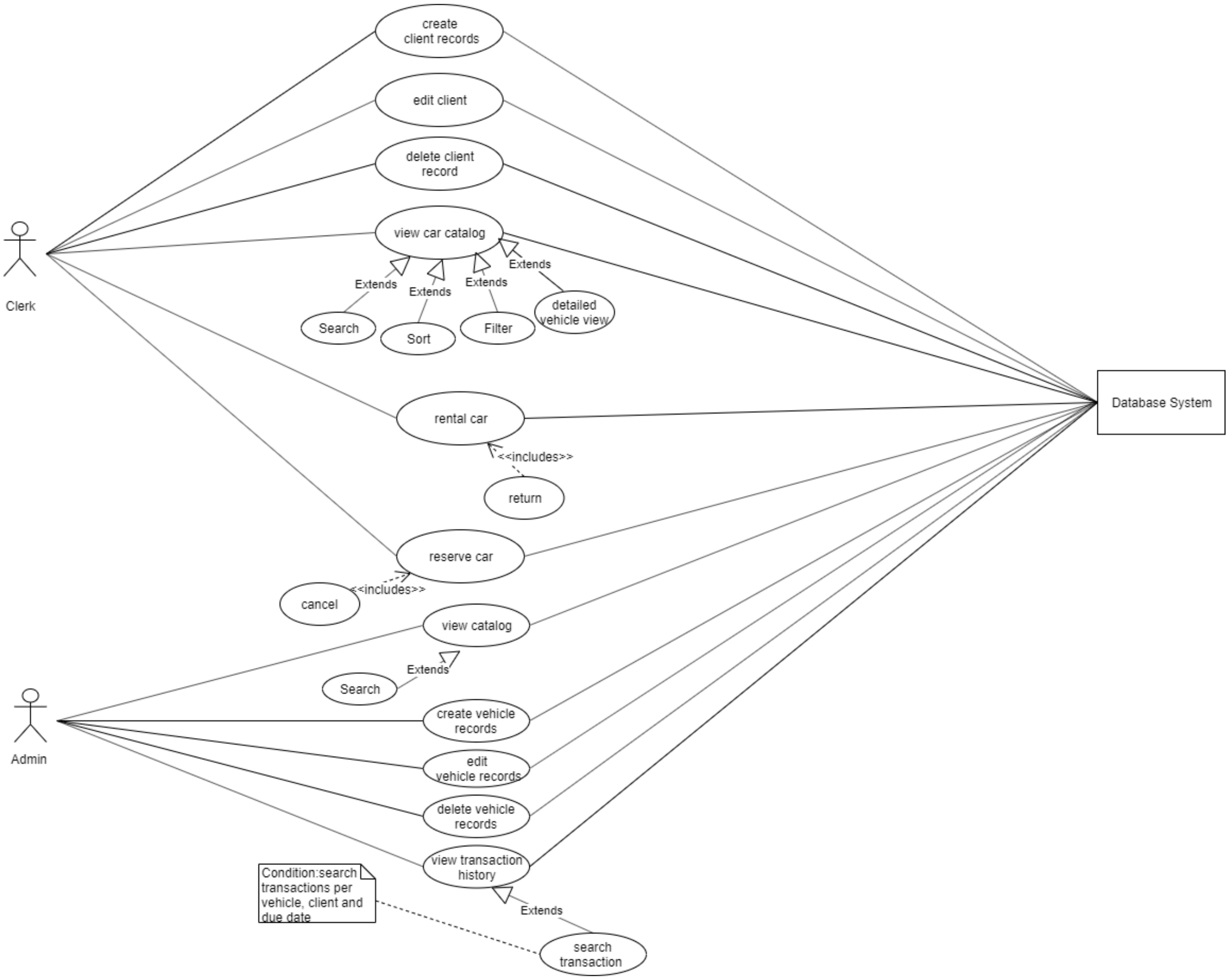
Administering the database

An administrator can manage a vehicle record (create new, modify, delete). An administrator can additionally view the contents of the catalog and perform searches, but cannot perform a rental, a reservation, or a return.

Actor goal list

Actor	Goal
Client	View Client Catalog
Clerk	Create, Modify, Delete Client Records, View Car Catalog, Handle Return and Reservation of the Car
Administrator	should be able to view the content of. Manages vehicles record in the catalog

Use case view



Here are the fully dressed scenarios for all the use cases :

<u>Use case 1:</u> Clerk Login

Primary actor: Clerk.

StakeHolders and Interests: Clerk should be able to login into his/her account using valid login credentials.

Preconditions: Clerk should have an account.

Postconditions: Clerk be able to view car catalog.

Main success Scenario:

1. Clerk should login into his/her account using valid login credentials.
2. Clerk should login into his account to view vehicle catalogue.
3. Additionally clerk can be able to select a vehicle to view in detail from a given set.

Extensions:

1a. Clerk may not have internet.

Use case 2: Client Management

Primary Actor: Clerk

Stakeholders and interest : clerk maintain all the client records.

Preconditions: At least one client should be there

Postconditions: All the client record is stored in client record .

Main success scenarios:

1. clerk should login into his/her account.
2. clerk gathers clients details and makes a client record .
3. All the clients details are maintained in client Management.

Extensions :

1a. If clerk does not have proper internet .

2a.If client provide insufficient data.

Use case 3: Reserving Vehicle

Primary Actor: Clerk

Stakeholders and interests: As per the client requirements clerk should be able to reserve a vehicle to the client.

Preconditions: Vehicle should be available on the specified dates.

Postconditions: Clerk should make a note of the reservation and should keep the vehicle as not available for those specific dates.

Main success scenarios:

1. After logging into his/her account clerk should gather the requirements from the client.
2. Clerk should check whether the vehicle is available or not based on the client requirement.
3. The clerk can view the complete vehicle details, then enters the start and end dates.
4. On clicking Reserve clerk will create a reservation and makes the vehicle was unavailable on those days.

Extensions:

1a. Clerk has trouble while logging into system.

Use case 4: The Return of a vehicle

Primary Actor: Clerk

Stakeholders and interests: Vehicles should be reserved, and the return date should match the current date

Postconditions: Clerk should return the vehicle and should update in the database as vehicle is not available.

Main success scenarios:

1. Clerk should be able to see the reserved vehicles list.

2. Clerk selects either cancelling the reservation or rent the vehicle.

3. After successful rent of the vehicle, the vehicle should be not available from that date.

Extensions:

1a. There may be no vehicle records for particular client to initiate return.

Use case 5: Admin Login

Primary Actor: Admin

Stakeholders and Interests: Admins should have login into his account the system.

Preconditions: Admin should have login credentials.

Postconditions: Admin should be able to see all the transactions.

Main success scenarios:

1. Admin should have his/her own credentials to log into his account.
2. After successful login into his account he/she should be able to view all the transactions that were made previously.

Extensions:

1a. If client does not able to get login credentials.

2a. If there was no transaction history.

Use case 6: Transaction History

Primary Actor: Administrator

Stakeholders and Interests: Admin should be able to view the transaction details as the clerk records.

Preconditions: Admin should login into his/her account.

Postconditions: Admin should verify whether the transactions are valid or not.

Main success scenarios:

1. Admin should login into his/her account.
2. Then he/she should select the view transaction option and then will be able to view all the transactions.

Extensions:

2a. All the transactions should be fetched and displayed correctly from the database.

Use case 7: Transaction history by particular attribute.

Primary Actor: Administrator

Stakeholders and Interests: Admin should be able to view the transaction details as the clerk records.

Preconditions: Admin should login into his/her account.

Postconditions: Admin should verify whether the transactions are valid or not.

Main success scenarios:

1. Admin should login into his/her account.
2. Then he/she should select the transactions based on the specific attribute.
3. Admin should verify whether the details are correct or not.

Extensions:

2a. If the specified attribute details are not fetched properly.

3a. If the details are fetched and are not validated properly.

Use case 8: Catalog with detail view

Primary Actor: Clerk

Stakeholders and interests: Clerk wants to view the vehicle CatLog and view a particular vehicle details.

Preconditions: Clerk should login into account.

Postconditions: Clerk should be able to select the required vehicle in detail from the catalogue.

Main success scenarios:

1. Clerk should login into account.
2. Then clerk should click on view catalogue.
3. The clerk can see the vehicle catalogue, he/she can also view a result set based on some search criteria.
4. Clerk may also be able to see the certain vehicle details in detail or will go back to the original state.

Extensions:

1a. If there was no records.

4a. There was some trouble fetching data from database

Use case 9: Catalog sort and filter

Primary Actor: Clerk

Stakeholders and interests: Clerk can be able to select the vehicles based on sorting order.

Preconditions: Clerk should login into his account.

Postconditions: Clerk can select the vehicle that he wants.

Main success scenario:

1. Clerk should login to his account using his details.
2. He/She can be able to view the car catalogue.
3. Clerk should select vehicle based on the sort order by year or car type.

Extensions:

1a. Clerk may not have internet.

Use case 10: Catalog search by particular attribute

Primary Actor: Clerk

Stakeholders and interests: Clerk should be able to view the catalog by searching a particular attribute.

Preconditions: There should be a vehicle catalog.

Postconditions: View the vehicle in detail.

Main success scenario:

1. Clerk should login into his/her account.
2. Clerk gathers vehicles records and makes a car catalog.
3. All the vehicle details are inserted an individual car details.
4. Clerk can be able to search by particular attribute.

Extensions:

1a.If clerk does not have proper internet.

4a. If the particular attribute was not implemented properly.

Use case 11: Vehicle management system

Primary Actor: Admin

Stakeholders and interests: Admin must be able to insert the new vehicle record.

Preconditions:

- Admin should login into his/her account.
- Vehicle should have a valid license plate and all the details are specified clearly.

Postconditions: Based on client requirement clerk reserve's a vehicle to the client.

Extensions:

1. Admin should insert all details of vehicle in the new vehicle record.
2. License number plate should be valid.
3. A new vehicle record is created and is saved in the database.

Use case 12: Optimistic Lock

Primary Actor: Clerk

Stakeholders and interests: One or more clerks wants to view the client details and car catalog at a time but cannot update at the same time

Postconditions: Clerk will reserve/cancel/return the vehicle.

Main scenario:

1. Initially clerks should login into their accounts.
2. One or more clerks can be able to view catalog or search catalog.
3. But only one clerk can be able to reserve the vehicle at a time.

Extensions:

1a. Clerks does not have proper internet resources.

Use case 13: Pessimistic Lock

Primary Actor: Admin

Stakeholders and interests: Admin want to login and modify the vehicle record only one at a time. Other Admins should not be able to read or write when a vehicle is being modified

Preconditions: Admin should login into his/her account.

Postconditions: View transaction history and Add,Delete,Modify Vehicle records

Main success scenarios:

1. Admin should be able to login into his/her account.
2. if one admin was logged in the other admin cannot be able to login into his/her account.
3. only one admin can be able to login and do the required operations.

Extensions:

1a. Admin does not have login credentials.

3.3 Non-Functional requirements

Performance efficiency

1. The usual load time of each webpage takes approximately 0.5 sec to 2 sec depending on the internet speed
2. The system's core is powered by Tomcat Server

Compatibility

1. As we are using bootstrap 4.5 design in the html web page design, this can damage the design whenever the application is opened in different web browser

Usability

1. This web applications is designed with simple GUI in which clerk functionality is self explanatory.
2. As it has a simple GUI, anyone with a minimum knowledge of web site browsing can access the application

Reliability

Stability: The application won't crashes, they won't be any unhandled exceptions and script errors.

Data integrity: All types of data remain intact throughout the product.

Recoverability: It is possible to recover and continue using the product after a fatal error.

Security

Authentication: Clerk has been given Unique Name and Email Address to login to ensure that an authenticated clerk has been logged in.

Privacy: An authenticated clerk is logged in to ensure that the client records are secured.

Maintainability

The web application has a simple GUI, so it's easy to extend the functionality and give new responsibilities to the use case actors i.e., Client, Clerk and Admin.

Portability

This web application is portable with both ios and windows.

License

All rights are reserved to Team 10 lead by the team leader and guided by professor
Dr. Constaninous Constantinides.

4. Analysis models

Below we have illustrated some critical scenarios

1. ClerkLogin

Operation Contract

Contract OP1: clerkLogin

Operation: validateLogin(id:clerkID, clerkName:String)

PreConditions: There is a valid Clerk created by Admin

PostConditions:

- A Clerk should be able to manage client records.
- A Clerk should be able to reserve cars and manage rentals.
- A Clerk should be able to view and search through Car Catalog.

2. ClientManagement

Operation Contract

Contract OP2: ClientManagement

Operation: AddClient(), UpdateClient(clientID), DeleteClient(clientId), ViewClientDetails()

PreConditions: There is a valid Clerk created by Admin

PostConditions:

- A Clerk should be able to add,update and delete client records.
- A Clerk should be able to view client records

3.. Reserving Vehicle

Contract op3: Reserving vehicle

Operation: checkVehicleAvailability(status : Boolean, StartDate: Date), createRental(status : Boolean, ReturnDate: Date) and viewVehicleRecord(list: list<vehicles>)

PreConditions: There is an instance created by the clerk to a client to make reservation.

PostConditions:

- The vehicle records instance was created
- The instance is associated with the clerk to create rental for a particular client.
- The instance was created based on the vehicle availability status.

4. Return of vehicle

Contract OP4:returnVehicle(returnDate: Date, status: Boolean)

Preconditions: A rental instance was made by clerk for a particular client

PostConditions: The status of the vehicle must be changed to available after return

5. AdminLogin

Contract op5: AdminLogin

Operation: ValidateAdminLogin(String username,String password)

PreConditions: A Valid Admin is created in the database

PostConditions:

- An Error message is thrown on giving wrong credentials of Admin
- A Valid Admin must be able to manage vehicle records
- A Valid Admin must be able to view the transaction history of reservation and rentals created by clerks for particular clients.
- A Valid Admin must be able to view CarCatalog but cannot make a reservation or rental for a particular client

6.TransactionHistory

Contract op6: TransactionHistory

Operation: getTransactionStatus(String status), getAllTransactions(List<Transactions)

PreConditions: Administrator must be able to click on the transaction history page

PostConditions:

- Admin must be able to view transaction history page
- A Valid Admin must be able to get all the transactions made by a clerk against each particular client
- A Valid Admin must be able to view the transaction history of reservation and rentals created by clerks for particular clients.

7.TransactionHistory by particular attribute

Contract op7: TransactionHistory search by particular attribute.

Operation: getTransactionById(int reservationId), getTransactionByName(String clientName)

PreConditions: Administrator must be able to click on the transaction history page

PostConditions:

- Admin must be able to view transaction history page
- A valid Admin must be able to get all the transactions made by a clerk against each particular client by searching particular client or by particular transaction ID
- A Valid Admin must be able to view the transaction history of reservation and rentals created by clerks for particular clients by searching for particular car type type model, make etc.,

8.Catalog with Detail View

Contract op8: Catalog with Detail View

Operation: getCarDetailView(String licenseNum)

PreConditions: Clerk must be able to click on the Catalog Detail View page

PostConditions:

- clerk must be able to view Catalog detail view page
- A valid Clerk must be able to get all the car details on clicking Detail view of the particular car and be able to navigate to next CarDetailView.

9.Catalog Sort and Filter

Contract op9: Catalog with Sort and Filter

Operation: List<CarCatalogue> filterData and List<CarCatalogue> sortData

PreConditions: Clerk must be able to view CarCatalogue page

PostConditions: Clerk must be able to sort and filter data in CatCatalogue page

10.CatalogSearch by particular attribute

Contract op10: Catalog Search with particular attribute.

Operation: getCarDetailsById(int CarId), getCarDetailsByCarName(String)

PreConditions: Clerk must be able to view CarCatakogue page.

PostCondition:

- Clerk must be able to sort and filter data in CarCatalogue page.

11. VehicleManagementSystem

Contract op11: Add,Modify,Delete ,View Vehicles

Operation: getAllVehicles(List<Vehicle>),AddVehicle(),UpdateVehicle(int CarID),DeleteVehicle(int carID)

PreConditions: Admin should be able to login and view CarCatalog.

PostConditions: Admin must be able to add, delete, update and view Car Details.

12.Optimistic Lock

Contract op12: Clerk Optimistic Lock

Operation: OptimisticLock(clerk)

PreConditions: Clerk should be able to login.

PostConditions:

- An instance of clerk is created.
- Under optimistic lock multiple instances of clerk are created and the multiple clerks should be able to login at the same time.
- Multiple clerk should not be able to edit the records at the same time.

13.Pessimistic Lock

Contract op13: Clerk Pessimistic Lock

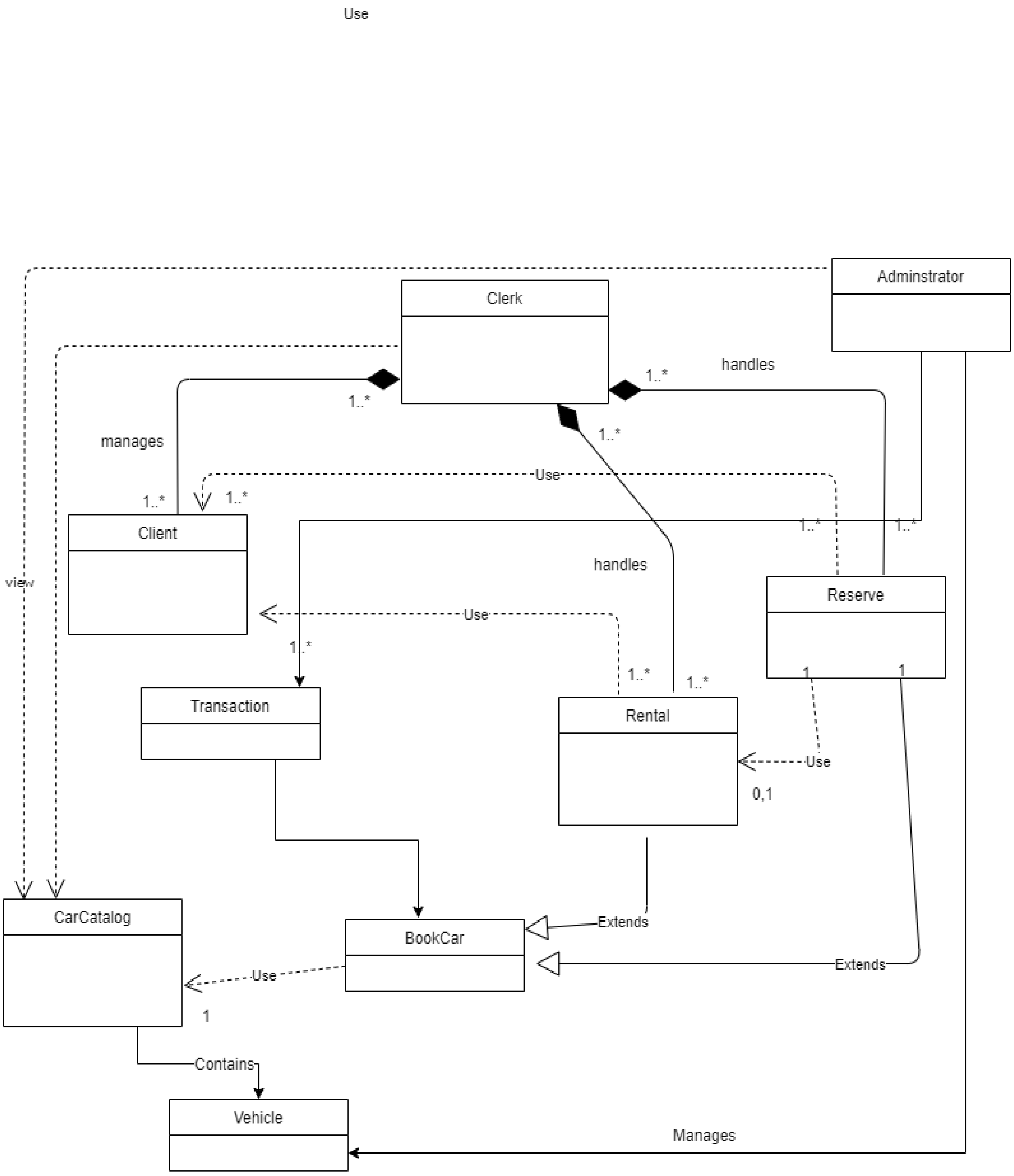
Operations: PessimisticLock(Admin)

Preconditions: Admin should be able to login

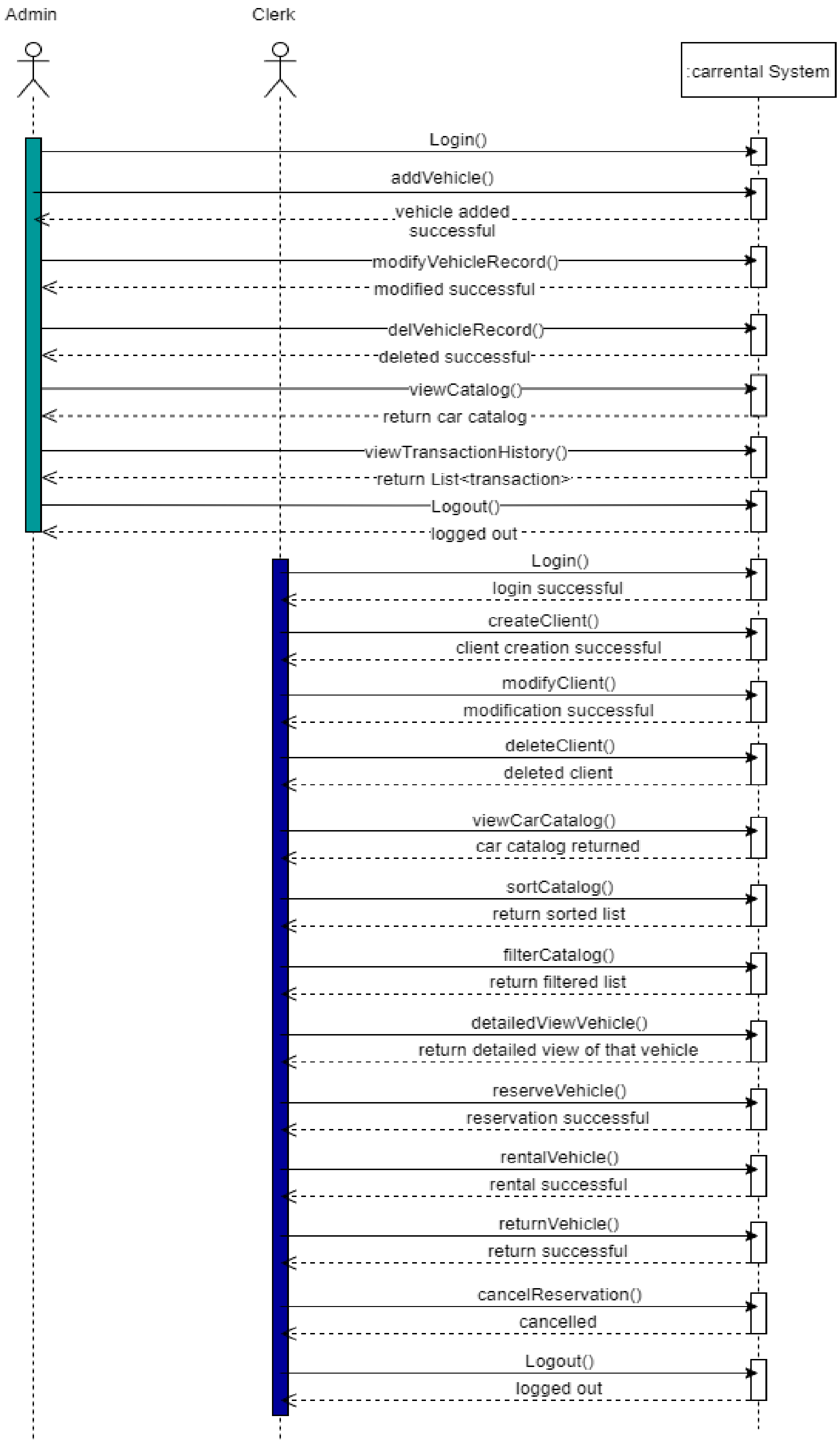
Postconditions:

- An instance of Admin is created.
- Under pessimistic Lock multiple instances of admin are created and the multiple admin should be able to login at the same time.
- Multiple admin should not be able to edit the records at the same time.

Domain Model



System Sequence Diagram



System Diagram

System Operations
login(username,role)
addClient(firstName,lastName,licenseNum,expDate,phoneNumber)
viewClientDetails(clientId)
updateClientDetails(clientId)
deleteClientDetails(clientId)
reserve(carId,availability,licensePlate)
getAllTransactionDetails()
getTransaction(TransId)
addCar(carId,Type,Make,Model,Year,color,LicensePlate)
viewCarDetails()
updateCarDetails(carId)
deleteCarDetails(carId)
carCatalogue()
searchCarCatalogue(licensePlate)
rental(carId,reservationID,availability,licensePlate)