.

Software Requirements Specification

Version 6.0

for

Vehicle Renting System

Prepared by

1.Bhavana Bomma - 40071320 - bomma08@gmail.com
2.Divyaprabha Rajendran - 40089282 - divyanajma@gmail.com [Team Leader]
3.Harish Jayasankar - 40105791 - harishjayasankar28@gmail.com
4.Ishpreet Singh - 40093666 - ishpreetsingh923@gmail.com
5.Mahy Salama - 40074737 - mahyhanysalama@gmail.com
6.Swetha Chenna - 40092019 - swethachenna2018@gmail.com

Instructor: Dr. C. Constantinides

Course: SOEN-6461

Date: 06/10/2019

Document history

Date	Version	Description	Author
06/10/2019	**V6**	**Revising, adding missing items and format**	**All the team members**
05/10/2019	**V5**	**Adding sequence diagrams**	**All the team members**
04/10/2019	**V4**	**Adding glossary and references**	**Mahy**
02/10/2019	**V3**	**Adding use cases scenarios and contracts**	**All the team members**
20/09/2019	**V2**	**Use Case Diagram**	**Harish**
18/09/2019	**V1**	**Introduction and purpose and scope**	**Swetha**

Table of contents

List of figures

1. Introduction

Vehicle Rental System is a web application which includes reserving ,renting and returning vehicles. Searching catalogues for vehicle details using clerk functionality.

Purpose

In this software requirement specification document we specify the process of Vehicle Rental System in different aspects starting from identifying requirements from potential users and analysing the requirements to design in broader perspectives using use case modeling,domain model and sequential representations. Implementing the functionality using patterns and to provide good and efficient web application we define non-functional requirements in this document.

Scope

The Software Requirements Specification (SRS) is a communication tool between users and software designers. Scope of this software requirement specification applied mainly to the stakeholders and according to our system we focus on the clerk who is the main actor in the scenario which covers all functionality of clerk.

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

Abbreviation	Definition
SRS	Software Requirements Specification, is a description of a software system to be developed. It lays out functional and non-functional requirements and lists sufficient and necessary requirements for the project development.
UC	Use Case, is a list of actions or event steps typically defining the interactions between a role (an actor in UML) and a system to achieve a goal. The actor can be a human, machine or other external systems.
CO	Contract, is a document that describes system behaviour for system operations that the system (as a black box) offers in its public interface to handle incoming system events.
UML	Unified Modeling Language, is a standardized modeling language enables developers to specify, visualize, construct and document the artifacts of a software system.

References

[1] Cockburn, A. (2000). *Writing effective use cases*. Addison-Wesley Professional.

[2] Ladd, S., Davison, D., Devijver, S., Yates, C., Harrop, R., & Donald, K. (2006). *Expert Spring MVC and Web Flow* (Vol. 1). Berkeley, CA: Apress.

[3] Larman, C. (2012). *Applying UML and patterns: an introduction to object oriented analysis and design and iterative development*. Pearson Education India.

2. Overall description

This section describes the background to the requirements: The general factors that affect the product, such as constraints, assumptions and dependencies.

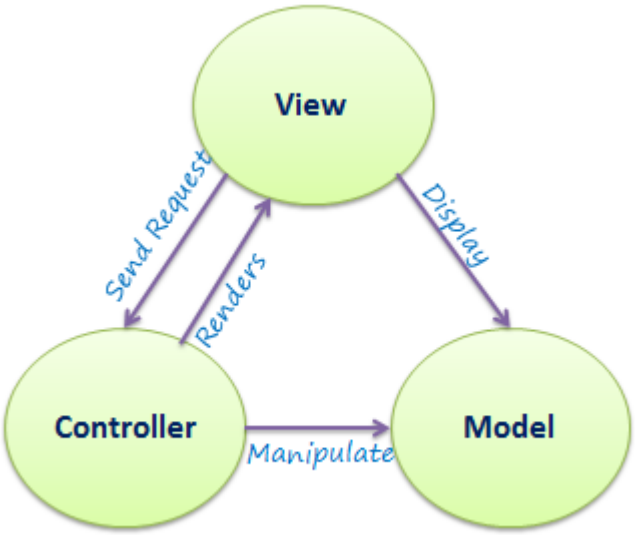


Figure 1. A Model View Controller architecture.

Product perspective

"Vehicle Rental System" web application, which consists of server containing the database using MySQL, the administrator and the clerk can access the application that is connected to the server through the cloud, and the clients that deal with the clerk to get benefit from renting vehicles.

Product functions

In this version of the SRS, the focus is on the Clerk, major functions of the Clerk are:

- Viewing the catalog of vehicles listed or for a detailed item, he can see either the vehicle is available or not (i.e. rented or reserved).
- Updating the catalog, cancel registration or handle a return for a vehicle.
- Viewing the clients records, create new client, modify or delete a client.
- Creating a reservation or renting for a vehicle to a client.

User characteristics

The intended users of the system are anyone that he or she has a valid driving license to drive a vehicle.

Constraints

He/She must have valid driving License.

The License must not be expired.

Assumptions and dependencies

Software used is Java 1.8, Tomact 7, both operating systems Windows and Mac OS are compatible, and an internet connection.

3. Specific requirements

This section contains all requirements in detail: 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

A detailed description of all inputs into the system and all outputs from it (in terms of content and form).

Figure 2. user interface..

View Catalog form:

List of all vehicles details showing their availability.

Inputs: press on vehicle id link or back to previous page or logout the system.

Outputs: transfer to the selected vehicle id to view in details or previous page or logging out.

serialno	vehicleid	type	make	model	year	color	License Plate	availability	cost
1	1006	suv	2015	2011	2011	white	AAA 000	YES	40
2	1001	sedan	2010	2011	2011	black	ABC 123	YES	70
3	1002	sedan	2010	2011	2011	black	BCD 345	NO	70
4	1003	sedan	2010	2011	2011	black	CDE 123	NO	70
5	1004	suv	2011	2011	2011	white	CDE 234	NO	50
6	1005	suv	2013	2011	2011	white	DEF 678	NO	40
7	1008	sedan	2009	2009	2010	red	DIV 280	YES	45
8	1007	suv	2016	2013	2014	black	HAR 281	YES	100
9	1009	suv	2018	2018	2018	white	PRA 274	YES	60

Back Logout

Vehicle details form:

Inputs: press Back or Next.

Outputs: transfer to the previous or next page in the list.

Make :	2010
Model :	2011
Year :	2011
Color :	black
License Plate :	ABC 123
Availability :	YES
Cost :	70

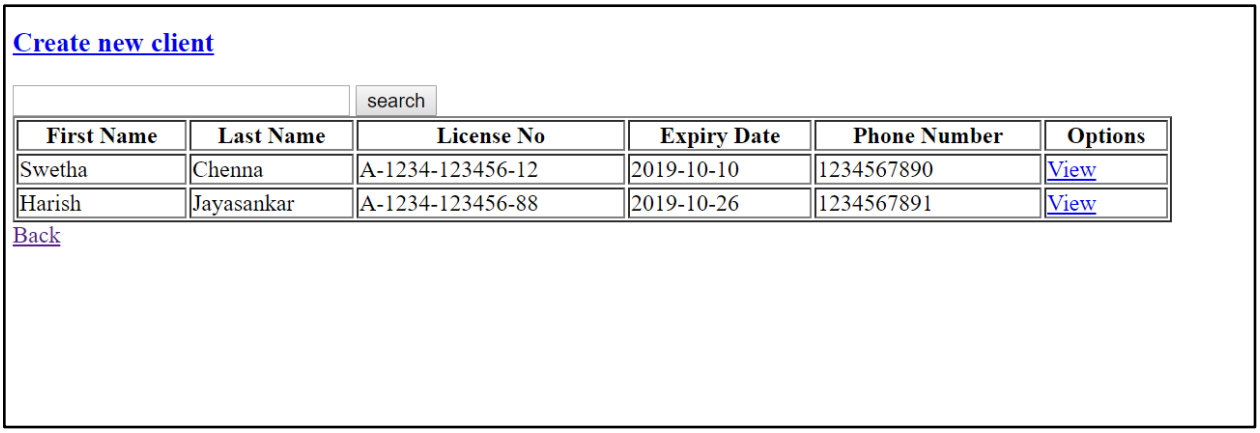
Back Next

Client Management System form:

List of all clients details.

Inputs: press Create new client link or View specific client link or Back

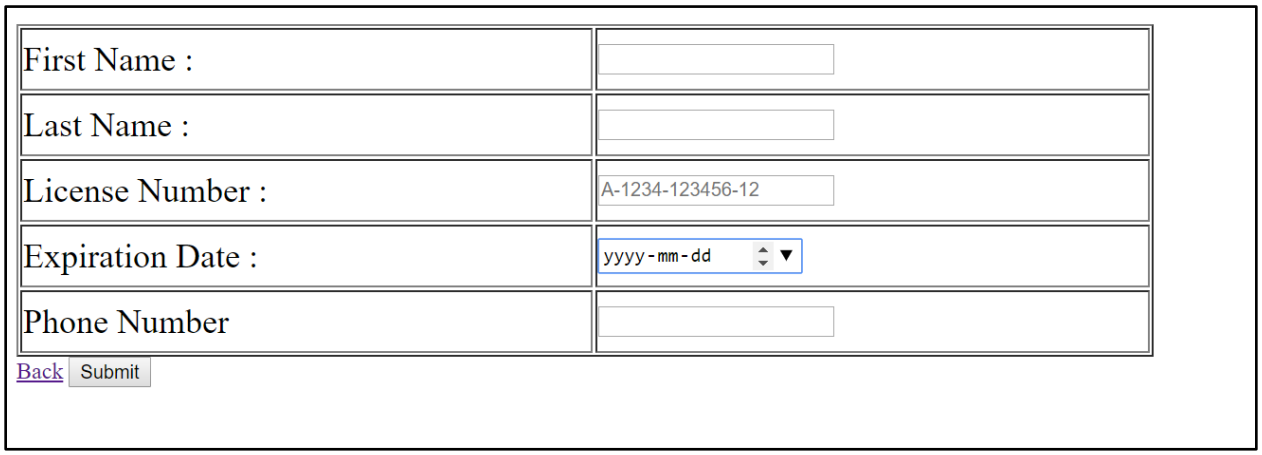
Outputs: transfer to Create new client form or View client form or to the previous page.



Create new client form:

Inputs: Client details (First Name, Last Name, License Number, Expiration Date, Phone Number) then press Submit or press on Back.

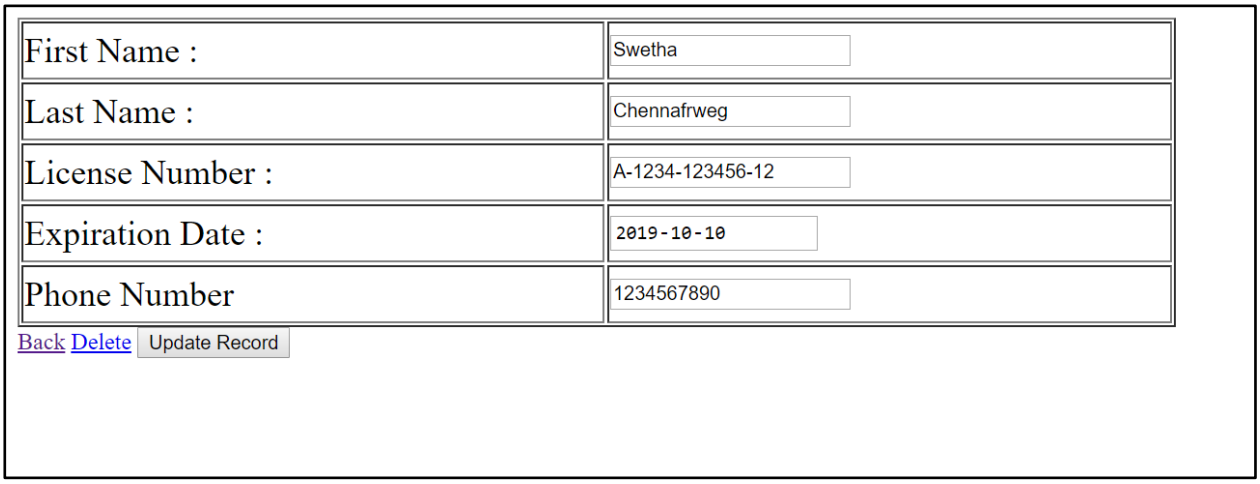
Outputs: new client is added or go to the previous page.



View client form:

Inputs: update to any of the client details (First Name, Last Name, License Number, Expiration Date, Phone Number) then press Update Record or press Delete or press on Back.

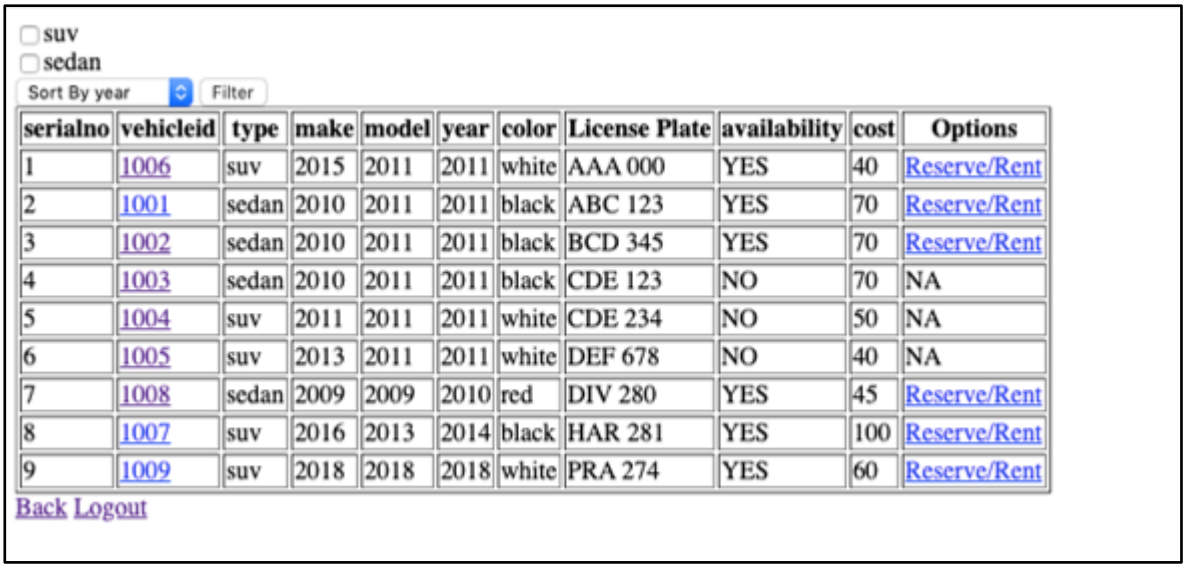
Outputs: client details are updated or client is deleted or go to the previous page.



Reserve or Rent form:

Inputs: press on Reserve/Rent for reserving or renting a car for a client.

Outputs: the Reserve/Rent form will be opened.

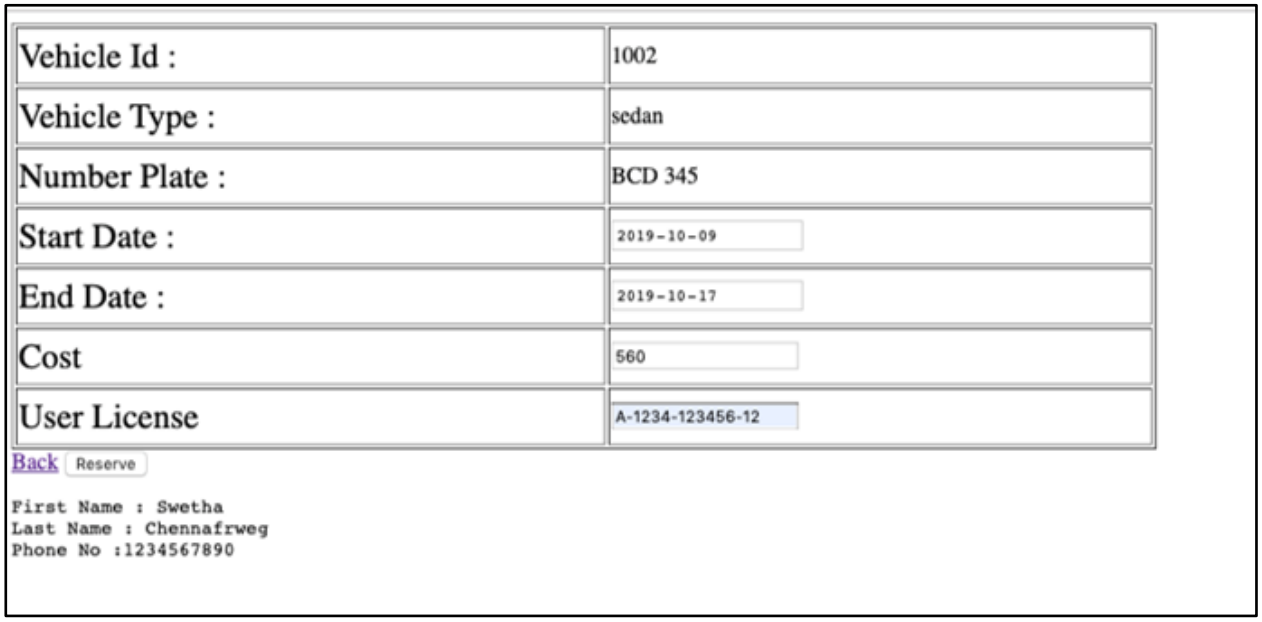


serialno	vehicleid	type	make	model	year	color	License Plate	availability	cost	Options
1	1006	suv	2015	2011	2011	white	AAA 000	YES	40	Reserve/Rent
2	1001	sedan	2010	2011	2011	black	ABC 123	YES	70	Reserve/Rent
3	1002	sedan	2010	2011	2011	black	BCD 345	YES	70	Reserve/Rent
4	1003	sedan	2010	2011	2011	black	CDE 123	NO	70	NA
5	1004	suv	2011	2011	2011	white	CDE 234	NO	50	NA
6	1005	suv	2013	2011	2011	white	DEF 678	NO	40	NA
7	1008	sedan	2009	2009	2010	red	DIV 280	YES	45	Reserve/Rent
8	1007	suv	2016	2013	2014	black	HAR 281	YES	100	Reserve/Rent
9	1009	suv	2018	2018	2018	white	PRA 274	YES	60	Reserve/Rent

Reserve form:

Inputs: press on Reserve to reserve the selected vehicle for the particular client or press Back.

Outputs: the vehicle is reserved or go to the previous page**Sort and Filter form:**



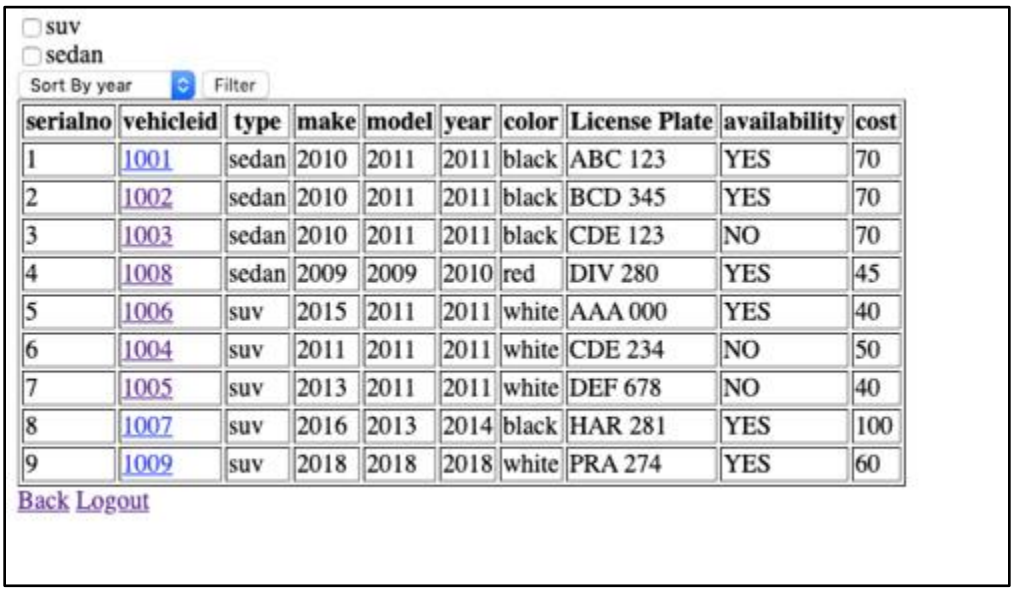
Vehicle Id :	1002
Vehicle Type :	sedan
Number Plate :	BCD 345
Start Date :	2019-10-09
End Date :	2019-10-17
Cost	560
User License	A-1234-123456-12

Back Reserve

First Name : Swetha
Last Name : Chennafrweg
Phone No :1234567890

Inputs: select to sort the vehicles by year, or select the desired type and press on filter.

Outputs: the vehicles are sorted or filtered.



suv
sedan
Sort By year Filter

serialno	vehicleid	type	make	model	year	color	License Plate	availability	cost
1	1001	sedan	2010	2011	2011	black	ABC 123	YES	70
2	1002	sedan	2010	2011	2011	black	BCD 345	YES	70
3	1003	sedan	2010	2011	2011	black	CDE 123	NO	70
4	1008	sedan	2009	2009	2010	red	DIV 280	YES	45
5	1006	suv	2015	2011	2011	white	AAA 000	YES	40
6	1004	suv	2011	2011	2011	white	CDE 234	NO	50
7	1005	suv	2013	2011	2011	white	DEF 678	NO	40
8	1007	suv	2016	2013	2014	black	HAR 281	YES	100
9	1009	suv	2018	2018	2018	white	PRA 274	YES	60

Back Logout

Cancel and Return form:

Inputs:Select Cancel/Return option in the welcome page to display details about vehicles and select option to cancel/return

Outputs:After selecting Cancel/Return option the vehicle should be cancelled if it was reserved or it should be returned if rented.The availability of vehicle should be changed to ‘yes’.

Vehicle Type	License Plate	Availability	Start Date	End Date	License Number	Cost	Options
suv	CDE 234	NO	2019-10-09	2019-10-17	A-1234-123456-12	400	Cancel/Return
suv	DEF 678	NO	2019-10-09	2019-10-17	A-1234-123456-12	320	Cancel/Return

Back

3.2 Functional requirements

Functional requirements capture the intended behaviour of the system. This section contains the *Actor Goal List* and the *Use Case view*.

Actor goal list

Actor	Goal
Clerk	1. To view Catalog. 2. To Filter and view Catalog. 3. To Sort and view Catalog. 4. To create a new client record. 5. To edit the existing client record. 6. To delete the client record. 7. To book a vehicle (rent or reserve a vehicle for a client). 8. To handle a return for a rented vehicle. 9. To cancel reservation of a vehicle.

Use case view

The use case model is shown in Figure 3.

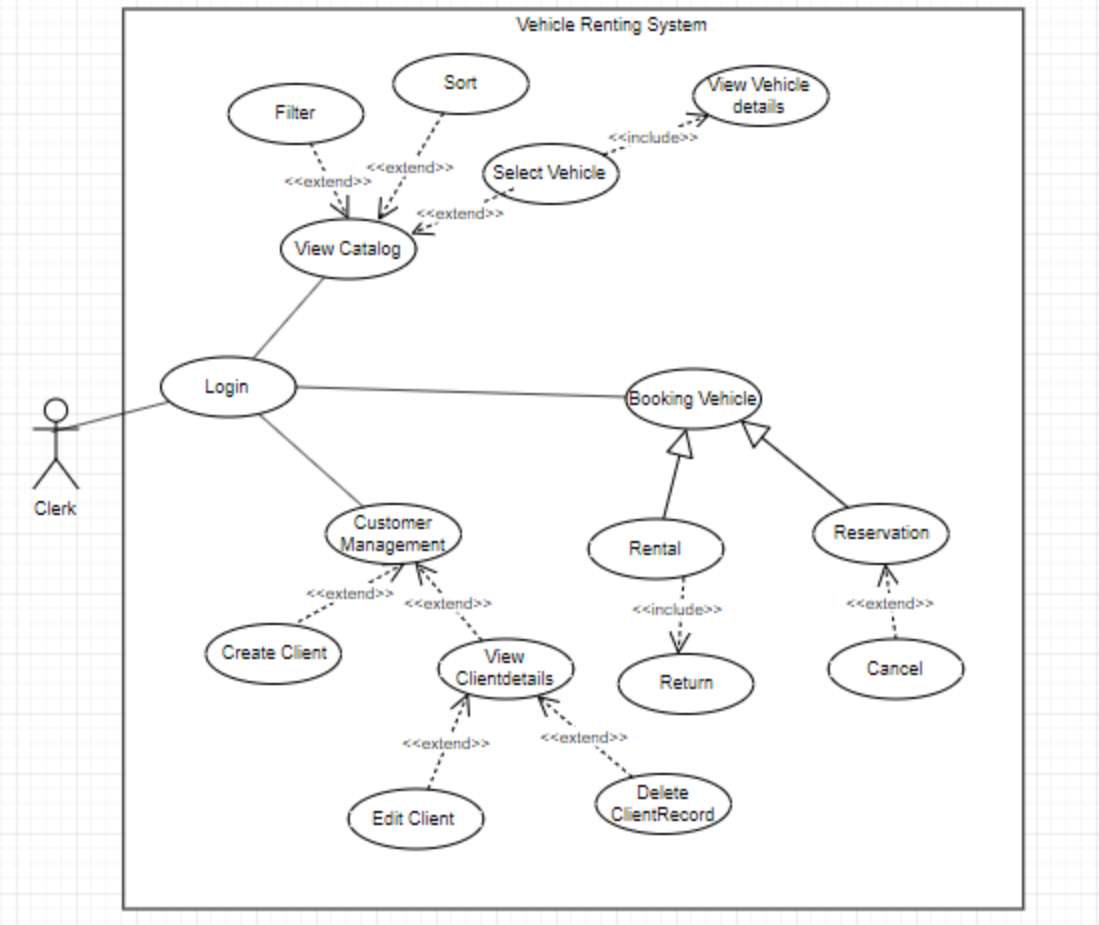


Figure 3. Use case model.

Here is the main description for the major use cases with their inclusions and extensions that are fully addressed as follows:

Use case UC1: View Catalog.

Primary Actor: Clerk.

Stakeholders and Interests: Clerk: Wants to view the vehicle CatLog and view a particular vehicle details.

Preconditions: Clerk is logged in (identified and authenticated).

Success Guarantee (Postconditions): Vehicles can be filtered, sorted and selected.

Main success scenario (or basic flow):

1. The Clerk log in to the system and clicks on View Catalog , He/She is displayed the list of vehicles.

2. The clerk can then create a result set through a selection of filtering criteria, for example the type of car (for ex SUV,Sedan) or on the basis of year.

3. The clerk may additionally choose the order by which he can view a result set: random order, or sort according to some criteria, e.g. View sorted by year.

4. From a given result set. Clerk may select 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.

5. The clerk can subsequently proceed to the next item in detail view or go back to the (possibly filtered) initial result set view.

Extensions (or alternative flows):

1a. The clerk may Not have internet connection to connect to the database.

Use case UC2: View Vehicle Details

Primary Actor: Clerk.

Stakeholders and Interests: Clerk: Wants to view the vehicle CatLog and view a particular vehicle details.

Preconditions: Clerk is logged in (identified and authenticated).

Success Guarantee (Postconditions): he can select a vehicle from catalog to view it in detail.

Main success scenario (or basic flow):

1. The Clerk log in to the system and clicks on View Catalog , He/She is displayed the list of vehicles.

2. The clerk can then create a result set through a selection of filtering criteria, for example the type of car (for ex SUV,Sedan) or on the basis of year.

3. The clerk may additionally choose the order by which he can view a result set: random order, or sort according to some criteria, e.g. View sorted by year.

4. From a given result set. Clerk may select 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.

5. The clerk can subsequently proceed to the next item in detail view or go back to the (possibly filtered) initial result set view.

Extensions (or alternative flows):

1a. The clerk may Not have internet connection to connect to the database.

Use case UC3: Filtering/Sorting

Primary Actor: Clerk

Stakeholders and interests:

clerk: Clerk can select the desired vehicle based on filtering and sorting condition

Preconditions: clerk is identified and authenticated.

Success Guarantee (Postconditions): Clerk will be able to search the vehicles based on filtering and sorting condition.

Main success scenario(or basic flow):

1. Clerk will log onto the system with valid credentials. Upon successful login, system displays the view catalog.
2. Clerk search for the desired vehicle with filtering and sorting criteria by year or vehicle type.

Extensions(or alternative flows):

1a. When clerk enters invalid credentials for logging.

2b. If invalid criteria selected by clerk.

Use case UC4: Create Client.

Primary Actor: Clerk.

Stakeholders and Interests:

Clerk: Create the new Client Record.

Preconditions: Clerk is logged in (Validated and authenticated).

Success Guarantee (Postconditions): Client details will be saved in database.

Main success scenario (or basic flow):

1. Clerk start creating new client record by filling all the mandatory details.

2. If all the details are valid the record will be created in database.

Extensions (or alternative flows):

2a. If the license number and license expiry date is not valid record will not be created and error message will be displayed.

2b. The clerk may fail to connect to the database where all catalog contents are saved.

Use case UC5: Edit Details.

Primary Actor: Clerk.

Stakeholders and Interests:

Clerk: Edit the client record.

Preconditions: Clerk is logged in (Validated and authenticated).

Success Guarantee (Postconditions): Client details will be saved in database.

Main success scenario (or basic flow):

1. Clerk will edit the existing details of the client.

2. If all the details are valid the record will be created in database.

Extensions (or alternative flows):

2a. If the license expiry date is not valid record will not be created and error message will be displayed.

2b. The clerk may fail to connect to the database where all catalog contents are saved.

Use case UC6: Delete Client.

Primary Actor: Clerk.

Stakeholders and Interests:

Clerk: Delete the client record.

Preconditions: Clerk is logged in (Validated and authenticated).

Success Guarantee (Postconditions): Client record will be deleted from the database.

Main success scenario (or basic flow):

1. Clerk will delete the existing client record from the database according to his license number .

2. If the license number exists in the database, the corresponding record will be deleted successfully.

Extensions (or alternative flows):

2a. The clerk may fail to connect to the database where all client records are saved.

Use case UC7: Make reservation.

Primary Actor: Clerk

Stakeholders and Interest: Clerk: Clerk should be able to create a reservation or rental by selecting the desired vehicle and providing appropriate details of the client and the reservation or rental details.

Preconditions: Clerk should be logged in. Vehicle desired should be available for the specified dates. Client should be an existing client.

Success Guarantee (Postconditions): Clerk shall create a reservation or rental record in the database by blocking the desired vehicle for the required period of time.

Main success scenario (or basic flow):

1. After successfully logging in to the system, the clerk can find the make reservation/rental option.
2. once clicked, the clerk is allowed to select one of the existing vehicles(with availability status set to YES) from the list of all vehicles.
3. The clerk can view the complete vehicle details and will be prompted to enter the start and end dates for the reservation and the cost for the reservation is automatically calculated provided the dates are valid. Reservation is made when the starting date is a future date. Rental is made when the start date is the current date.

4. The clerk should enter the license number for the client, provided the client is valid (already existing client), the client details are displayed on the same page.
5. On clicking reserve/rent, the clerk will be creating a reservation (if the start date is a future date) or rental (if the start date is the current date).
6. If successful, the vehicle's availability will be updated to NO, a success message will be printed and the form for making a new reservation is visible.

Extensions (or alternative flows):

1a. The clerk may have issues in logging in or database connectivity issue may occur.

2a. The client may not be an already existing client, which may result in the termination of making reservation. A new client should be created before continuing.

3a. The meantime between selecting the vehicle and making a reservation, some other clerk can make a reservation for the same vehicle during the same period making the vehicle unavailable. Reservation process has to be repeated for another vehicle.

Use Case UC8: Cancel/Return vehicle

Primary Actor: Clerk

Stakeholders and Interests: Clerk -Clerk should be able to cancel the reservation or return the rental by selecting the appropriate vehicle.

Preconditions: Clerk should be logged in .And there should be vehicles that are reserved so that we can cancel the reservation or return the vehicle that are rented.

Success Guarantee(PostConditions): Clerk shall cancel the reserved vehicle or return the rented vehicle and the availability of vehicle should be changed to 'yes'.

Main Success Scenarios(or basic flows):

1. After clerk successfully logging on to the system, clerk can see the cancel/return option.
2. Clerk can view the details of vehicles with option to cancel/return.
3. Select the link to cancel the reservation or return the rented vehicle, and availability will be changed to yes for the appropriate vehicle in the system.

Extensions:

1a. If clerk has invalid login credentials.

1b. There may not be reserved or rented vehicles in the list.

3.3 Non-functional requirements:

Performance efficiency

The system performance in terms of response time has been tested and shows a quick response to any request done from the clerk side as per his functionality.

Compatibility

The web application using JavaSpring as a framework is compatible with all kinds of browsers.

Usability

The Vehicle Rental System has been tested for usability by providing clear messages in case of any invalid input by the user, taking all potential invalid inputs, also introducing the application for non-technical people for using the system and getting their feedback.

Reliability

The web application is ensured to be reliable by by using try and catch mechanism for the code part, all possible error exceptions are handled to check that all inputs are in the correct formats.

Adding some test cases for catalog and client management and run them successfully in test suite to ensure correctness of methods.

Security

As the system has a login form to verify if the clerk has the authorization for using the system or not, it only allows the registered predefined clerks to use the application, so no malicious or theft could occur.

Maintainability

The application is flexible for any critical updates, by making low coupling and high cohesion between classes, methods and attributes, so to reduce dependencies where an update occurs in one class not affect the others.

Portability

The system is designed to be easily moved from one computing environment to another, such as browsers and different operating systems.

Design constraints

Following iteration process model by implementing subsequent functionality (clerk functionality) from the overall functionality of the system, following MVC Java Spring framework in implementing the web system, using UML online tools such as drawio and UML paradigm for designing the system architecture.

4. Analysis Models

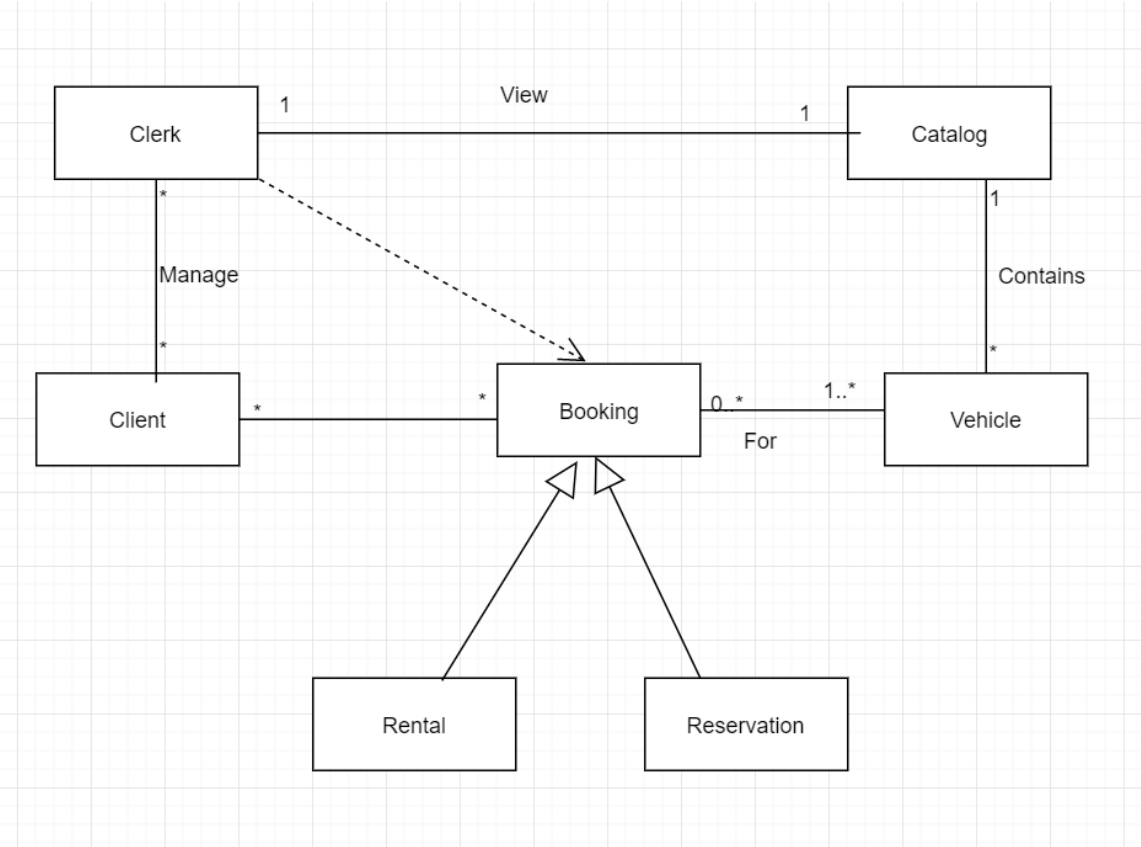
List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

Illustrate (system) ***UML sequence diagrams*** (one for each **critical** scenario), identify system operations and describe operation contracts, **one per critical system operation**.

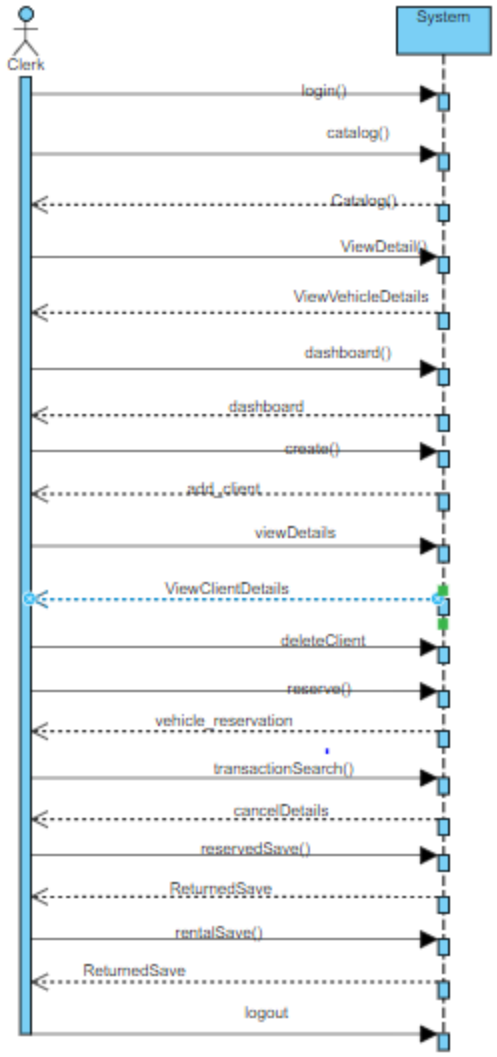
You may also use ***UML state diagrams*** to describe **critical use cases**, one state diagram per use case.

Finally, create a **UML conceptual class diagram ("domain model")** for the system. If the model gets too large, you can use **UML package diagrams** to provide logical groupings for the model.

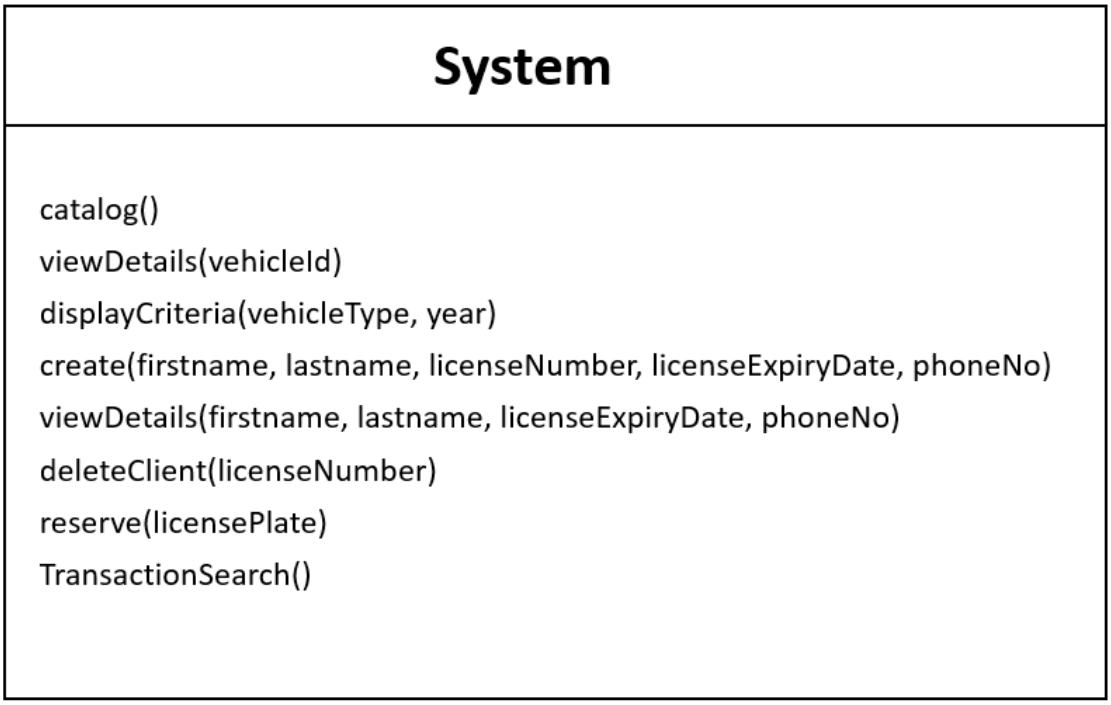
Domain Model : domain model is a conceptual model of the domain that incorporates both behaviour and data



System sequence diagram:



System Diagram



Contract 1:

CO1: catalog.

Operation: catalog().

Cross References: UC1 (View Catalog)

Preconditions: The Clerk must successfully login to the System.

Postcondition: 1. The instance of List<Catalog> cl is created.

2. cl was associated with Vehicle details table.

Contract 2:

CO2: ViewDetails.

Operation: ViewDetails(vehicleId)

Cross References: UC2 (View Vehicle Detail)

Preconditions: The Clerk must successfully login to the System.

Postcondition: 1. The instance of List<Catalog> cl is created.

2. cl was associated with Vehicle details table.

Contract 3:

C03: Sorting/Filtering

Operation: Display criteria(Vehicle type, Year)

Cross References: UC3(Filtering/ Sorting)

Preconditions: 1. The clerk must successfully login to the system

Postcondition: 1. The system displays the desired criteria based on the selected filtering and sorting condition.

Contract 4:

CO4: createclient.

Operation: create(firstname, lastname, licenseNumber, licenseExpiryDate, phoneNo)

Cross References: UC4 (create Client)

Preconditions:1. The Clerk must successfully login to the System.

2. The license number and license expiry date of the client should be valid.

Postcondition: 1. With all the details instance of client will be created.

2. The details in the client instance will be added to the database.

Contract 5:

CO5: editclient

Operation: viewDetails(firstname, lastname, licenseExpiryDate, phoneNo)

Cross References: UC5 (Edit Client)

Preconditions: 1. The Clerk must successfully login to the System.

2. The modified details of the client should be valid.

Postcondition: 1. With all the details instance of client will be created.

2. The details in the client instance will be added in database.

Contract 6:

CO6: Delete Client.

Operation: deleteClient (licenseNumber)

Cross References: UC6 (Delete Client)

Preconditions:The Clerk must successfully login to the System.

Postcondition: 1. The instance of client will be deleted.

2. client record will be deleted from the database.

Contract 7:

CO 7: makeReservation

Operation: reserve(licensePlate)

Cross References: UC7 (makeReservationOrRental)

Preconditions: The Clerk must successfully login to the System.

Preferred vehicle should be available

The client should be an existing client.

Postcondition: 1. The reservation should be created in database

2. The status of the vehicle must be set to “NO”.

Contract 8:

CO8 : Cancel/Return

Operation: transactionsearch()

Cross References: UC8 (Cancel/Return vehicle)

Preconditions: The clerk must be able to login

Vehicles should be available for cancel or return

client should be present in system

Postcondition: 1. Cancel or return should be performed .

2. status of vehicle must be set to “YES”