Software Requirements Specification

Version 2.3

for

Soen6461

Prepared by

Pargol Poshtareh	26428126	Pargol.Poshtareh@gmail.com
Richard Grand'Maison	26145965	Rich.Grandmaison@gmail.com
Ravneet Singh Brar	40078628	rippybrar1994@gmail.com
Gurjot Singh	40091208	gurjot5076@gmail.com
Adrien Dat	40072965	ad_dat@encs.concordia.ca

Instructor:	Dr. Constantinos Constantinides
Course:	Soen 6461
Date:	Oct 2019

Document history

Date	Version	Description	Author
18th Sep 2019	0.0		Pargol / Richard
25th Sep 2019	1.0		Pargol/Richard
29th Sep 2019	1.2		pargol/richard
30th Sep 2019	1.3		Ravneet Singh Brar
30th Sep 2019	1.4		Adrien Dat
30th Sep 2019	1.5	Conceptual Class diagram	Gurjot Singh
1st Oct 2019	1.6	More Contracts and reorganisation	Adrien Dat
1st Oct 2019	1.7	Fixes on Domain Model	Gurjot Singh
2nd Oct 2019	1.8	Review	Adrien Dat
4th Oct 2019	1.9	Review	Pargol/ Richard
13th Oct 2019	2.0	Add Admin stories and features	Pargol/Richard
14th Oct 2019	2.1	Add missing for administration	Pargol
23rd Oct 2019	2.2	Add some Admin informations	Adrien
26th Oct 2019	2.3	Review (Iteration 3 release)	Gurjot/Ravneet

Figures

1. Introduction

We have been asked to create a web application for a renting car system. In this document you will find a Software Requirement specifications analysis, completed iteration through iteration. The first iteration deals with the clerk functionality. The second iteration deals with the administrator functionality. The third and final iteration deals with concurrency and persistence.

1.1 Scope

This document will help us in the building process of our application. Indeed, we will be able to identify more easily the different utilisation of our application since we will have to think to many different use cases, how to handle them and how to avoid failures along the use cases ways. Finally, from it, we will start building our design, which you will find in the SAD document in the same folder as this requirement analysis.

1.2 Definitions, acronyms, and abbreviations

UC - Use Case: a series of related and consecutive success and/or failure scenarios that describe actor(s) using a system to achieve a goal (reformulated from the lectures)

In. - Include

DB - Database

SRS - Software Requirements Specification

1.3 References

List of all documents referenced in the SRS:

- None

2. Overall description

The product we are designing is a web application.

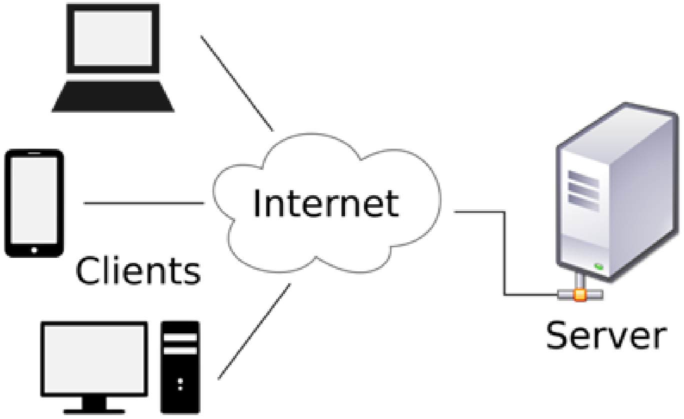


Figure 1. A client-server architecture.

2.1 Product perspective

The product is composed of a database, a front and a backend. Backend and database or located in the server (see figure 1).

2.2 Major Product functions

- Rent, reserve or cancel a reservation on a vehicle
- Add or remove client from the Database
- Add or remove a vehicle from the Database
- See list of vehicle or client

2.3 User characteristics

There are two types of User, Clerk and Admin. In terms of education, they are expected to know what they are able to do on the platforme (see 3.2), and to know how to use a computer in order to connect to the platform, nothing more.

2.4 Constraints

For the moment, there is no limitation identified.

2.5 Assumptions and dependencies

Since we are designing a web application, as long as the computer is equipped with an internet connexion and a browser, the user should be able to access the application.

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

yet to come

Figure 2. An example user interface.

3.2 Functional requirements

3.2.1 Actor goal list

Actor	*Goal*
Clerk	- view the contents of a catalog, either in a random order, or by creating a result set through a selection of filtering criteria - may additionally choose the order by which they view a result set - from a given result set, may choose an item to view in detail - proceed to the next item in detail view, or go back to the initial result set view - 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 - handle the return of a given vehicle
Admin	- Can access, view and search the history of transactions per client, per vehicle, or per due date - Can manage a vehicle record (create new, modify, delete).

	- Can additionally view the contents of the catalog and perform searches - /!\ Cannot perform a rental, a reservation, or a return

3.2.2 Use case view

The use case model is shown in Figure 3.

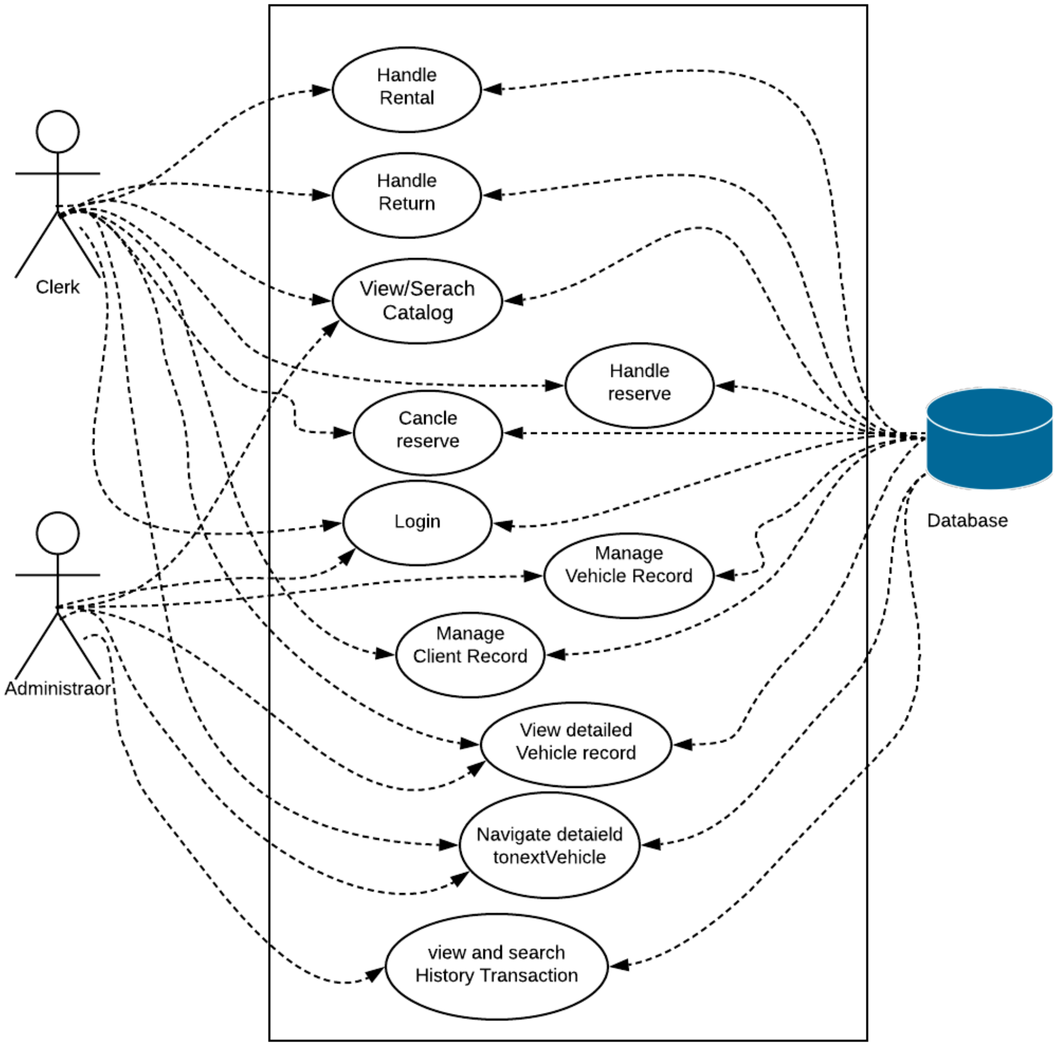


Figure 3. Use case model.

3.2.3 Brief Use cases

UC1:

Use Case Name:	Search catalog
Actors:	Clerk / Administrator
Description:	Clerk or administrator will use the system to input information and find the right vehicle or see all the vehicles

UC2:

Use Case Name:	Handle Rental
Actors:	Clerk
Description:	The clerk uses the system to book a car for a customer, based on information from the customer. The system save all the car/customer information.

UC3:

Use Case Name:	Cancel Reservation
Actors:	Clerk
Description:	Based on some info, the clerk can cancel an already existing reservation.

UC4:

Use Case Name:	Handle Return

Actors:	Clerk
Description:	Clerk use the system to record each returned car based on info.

UC5:

Use Case Name:	Handle Reservation
Actors:	Clerk
Description:	The clerk inputs a customers Information and the car which he/she wish to reserve.

UC6:

Use Case Name:	Manage Vehicle Record
Actors:	Administrator
Description:	Administrator have access to change the info about the vehicle.

UC7:

Use Case Name:	Manage Client Records
Actors:	Clerk
Description:	Clerk can create new client or delete or change info of each.

UC8:

Use Case Name:	View/Navigate detailed Vehicle record
Actors:	Clerk and Administrator

Description:	Clerk and Administrator have access to detailed of vehicles and they can Navigate to the next or previous vehicle.

UC9:

Use Case Name:	Login
Actors:	Clerk / Administration
Description:	Clerk or Admin need to login to the system to access to system.

UC10:

Use Case Name:	View and Search History of Transactions
Actors:	Administration
Description:	Administrators can access, view and search the history of transactions per client, per vehicle or per due date.

3.2.4 Fully dressed Use Cases

NAME	Search Catalog
ID	UC1
DESCRIPTION	This use case permits clerk or admin to search vehicle from the vehicle list in order to display.
ACTORS	Clerk , Administrator
Main Success Scenario:	1. A clerk or Admin logged in to the system successfully. 2. The system authenticated the username and password. 3. The admin or clerk clicks on search vehicle link. 4. The system displays combo box to select search to a vehicle. The admin or clerk select one of the following lists from the combo Box, Vehicle Brand. Vehicle Type. Vehicle Model or default is All. 5. Clicks on search button. 6. Then the system displays all information about the vehicle based on selected list.
PRE CONDITION	● Clerk or Administrator is authenticated.
POST CONDITION	● Display
EXCEPTION / Alternatives	2a. A clerk do not pass the authenticated.

NAME	Handle Car Rental
ID	UC2
DESCRIPTION	The clerk uses the system to book a car for a customer, based on information from the customer. The system save all the car/customer information.
ACTORS	Clerk
Main Success Scenario:	1. A clerk logged in to the system successfully. 2. The system authenticated the username and password. 3. A clerk retrieve a list of vehicles according to the client's preferences 4. A clerk pick the car which the customer desired. 5. The system check for availability of chosen car for desire date. 6. A clerk click “Submit” button. 7. The system sends the confirmation.
PRE CONDITION	● Clerk is authenticated. ● Selected vehicle is not rented. ● Selected vehicle is not reserved for the desired duration. ● Client renting the vehicle exists in the system.
POST CONDITION	● System send confirmation to the Clark ● Database get updated with all new information
EXCEPTION / Alternatives	2a. A clerk do not pass the authenticated. 3a. A desired car is not available. 6a. The system don't sent the confirmation.

NAME	Handle Reservation
ID	UC5
DESCRIPTION	The clerk inputs a customers Information and the car which he/she wish to reserve
ACTORS	Clerk
Main Success Scenario:	1. A clerk logged in to the system successfully. 2. The system authenticated the username and password. 3. The a clerk clicks on search vehicle link. 4. The system displays combo box to select search to a vehicle. The clerk select one of the following lists from the combo Box, Vehicle Brand. Vehicle Type. Vehicle Model or default is All. 5. Aclerk pick the car and date desired by customer 6. Clicks on reservation button. 7. Then the system displays confirmation of reservation.
PRE CONDITION	● Clerk is authenticated. ● Car is available on that dates
POST CONDITION	● Display the confirmation ● Database updated with new information
EXCEPTION / Alternatives	2a. A clerk do not pass the authenticated. 5a. A car is not available on that particular dates

NAME	Cancel Reservation
ID	UC3
DESCRIPTION	Based on some info, the clerk can cancel an already existing reservation.
ACTORS	Clerk
Main Success Scenario:	A clerk logged in to the system successfully. The system authenticated the username and password. The a clerk clicks on search reservation list. The system displays list of reservation and allow us to have filter based on the type of car or dates. A clerk pick the reservation which the customer will cancel. Clicks on cancellation button. The system displays a confirmation of cancellation.
PRE CONDITION	● Clerk is authenticated. ● Car display on reservation list.
POST CONDITION	● Display the confirmation ● Database updated with new information
EXCEPTION / Alternatives	2a. A clerk do not pass the authenticated. 5a. A car is not available on the list.

NAME	Manage Vehicle Record
ID	UC6
DESCRIPTION	Administrator can add new vehicle to the database, Modify/Delete existing vehicle information.
ACTORS	Administrator
Main Success Scenario	1. Admin logs in to the system 2. System authenticates the Username and password of the admin. 3. Admin selects Add new Vehicle information or Manage Existing information. 4.Depending on the previous Choice, admin is able to add the new vehicle information to the Vehicle catalog or is able to modify the existing information.
PRE-CONDITIONS	1.Administrator is authenticated with the system 2. If Admin wishes to manage some already existing vehicle info, it is already available in the Catalog.
POST-CONDITION	● 1.Catalog is updated with the new/modified information.
EXCEPTIONS/ Alternatives	2a. Administrator fails to log-in. 5a. Information admin wishes to modify does not exist in the catalog.

NAME	Manage Client Record
ID	UC7
DESCRIPTION	Client can add new customer record or make changes to the existing client records
ACTORS	Clerk
Main Success Scenario	1. Clerk logs in to the system 2. System authenticates the Username and password of the clerk. 3. Clerk selects Client Record 4. Clerk can select existing record and modify it or add a new Client Record 5. Depending on the previous Choice, Clerk is able to add the new client record or is able to modify the existing Client records.
PRE-CONDITIONS	· Clerk is authenticated with the system · If Clerk wishes to manage some already existing Client Record, it is already available in the System.
POST-CONDITION	· System is updated with the new/modified information.
EXCEPTIONS/Alternatives	2a. Clerk fails to log-in. 5a. Client Record that clerk wishes to modify does not exist in the system.

NAME	View and Search History of Transactions
ID	UC10
DESCRIPTION	Administrators can access,view and search the history of transactions per client, per vehicle, or per due date.
ACTORS	Admin
MainSuccess Scenario	1. Admin logs in to the system 2. System authenticates the Username and password of the admin. 3. Admin navigate to transaction's history section 4. Admin select search criteria. 5. System return search results.
PRE-CONDITIONS	Admin logged in.
POST-CONDITION	A TransactionSearchResult object *transactionSearchResult* is created. Transaction objects matching the search criteria are created and associated to the TransactionSearchResult.
EXCEPTIONS/Alternatives	2a. Admin fails to log-in.

3.3 Non-functional requirements

3.3.1 Concurrency

The system would allow at most one administrator being logged in at any time. The system should allow possibly multiple clerks being logged in performing their tasks. We can divide system operations into two categories:

Write operation A write operation accesses in order to modify the contents of a shared resource. A client who wishes to perform a write operation is referred to as a 'writer.'

Read operation A read operation accesses in order to view (but does not modify) the contents of a shared resource. A client who wishes to perform a read operation is referred to as a 'reader.'

The following properties are relevant to any concurrent system:

- **Safety**

A shared resource must not be simultaneously accessed by a writer and reader. Furthermore, a shared resource must not be simultaneously accessed by more than one writer.

- **Liveness**

A client who wishes to obtain access to a shared resource will eventually be allowed access.

- **Fairness**

Requests for access must succeed infinitely often. No client will wait for ever to be serviced. Furthermore, in this system we want writers to have priority over readers.

3.3.2 Persistence

Persistence must be provided by a relational database and you must design and implement object-relational mapping from scratch.

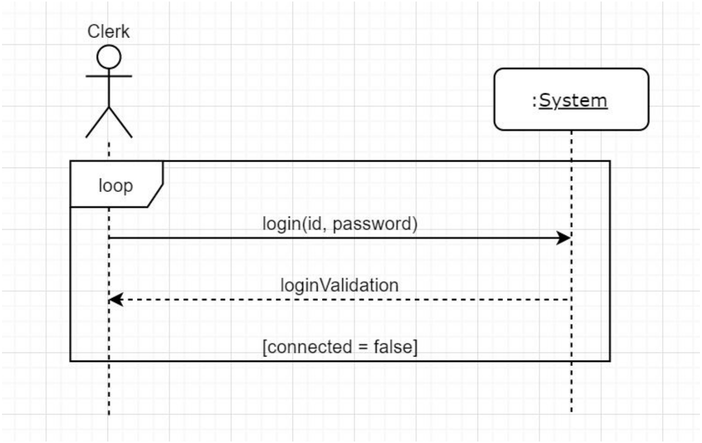
3.3.3 Other

You must design and implement the application in an object-oriented environment with an appropriate architectural style that supports quality characteristics such as separation of concerns, modularity, reusability and adaptability

4. Analysis Models

4.1 System Sequence Diagram

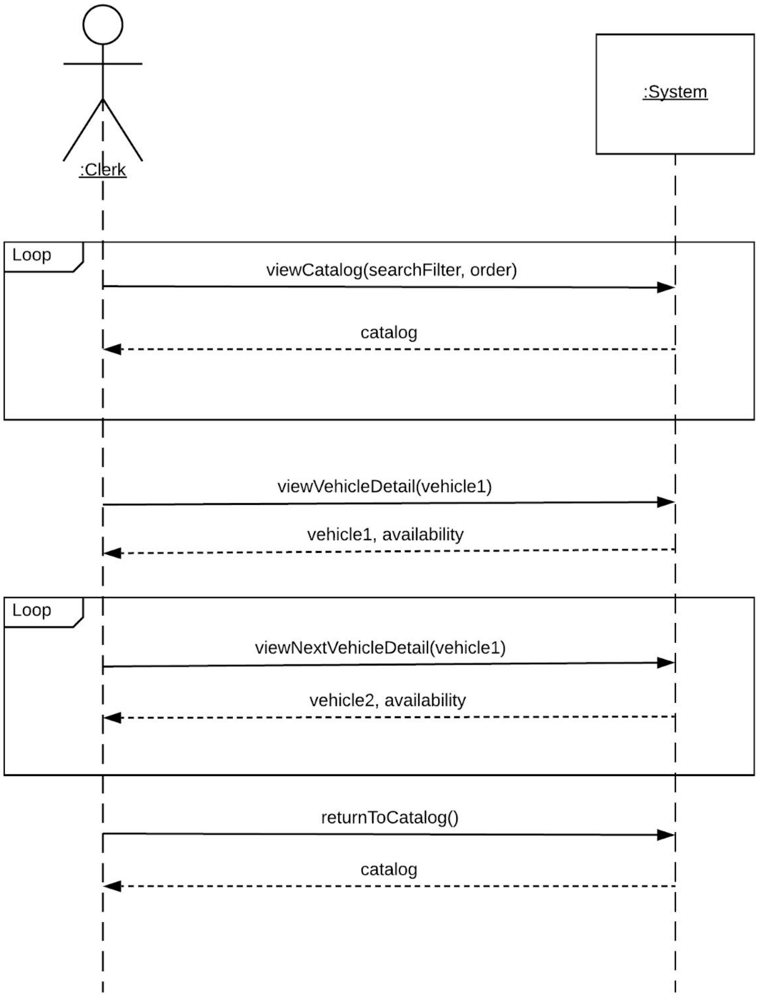
Here is the login System Sequence Diagram, common to every System Sequence Diagram, an example of it placed is available on the System Sequence Diagram of UC2.



4.1.1 Clerk - View and Search Catalog

<u>Related to UC1 - Search Catalog</u>

Assumption: The actor is a Clerk which has successfully logged-in as a Clerk, and has navigated to the “Vehicle Catalog” section.

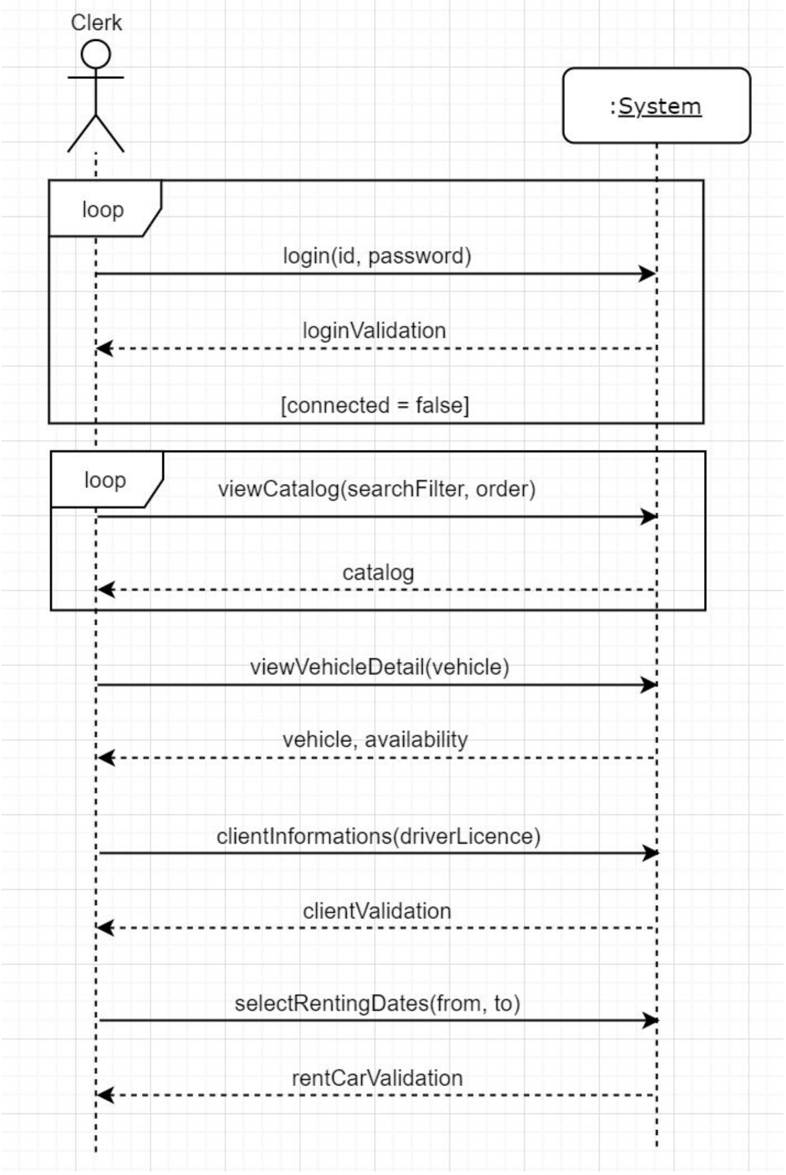


4.1.2 Clerk - Make Rental

Related to UC2 - Handel Rental

Assumption: After login, the clerk has navigated to the “Returns” section.

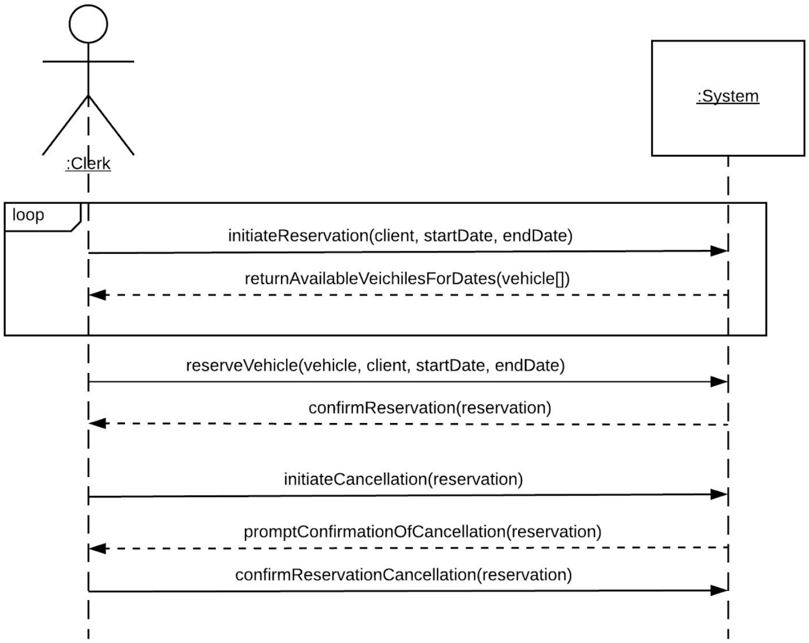
NOTE: The login process is demonstrated in this diagram by the loop. It is assumed to be identical in other system sequence diagrams which have a Clerk as the actor.



4.1.3 Clerk - Make Reservation

Related to UC5 - Handle Reservation and UC3 Cancel Reservation

Assumption: The actor is a Clerk which has successfully logged-in as a Clerk, and has navigated to the “Reservations” section.

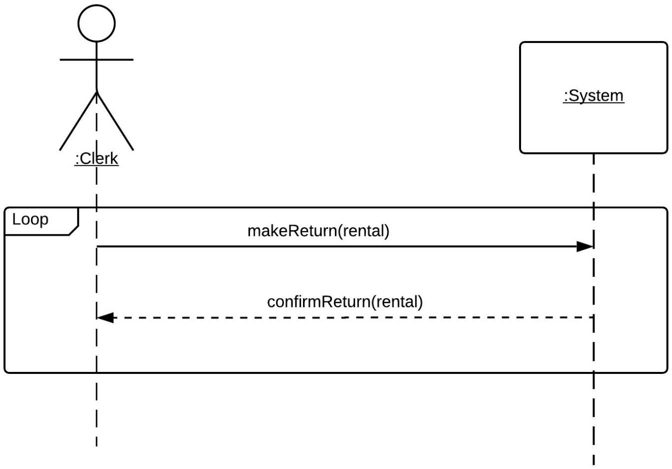


:Clerk
:System
loop
initiateReservation(client, startDate, endDate)
returnAvailableVeichilesForDates(vehicle[])
reserveVehicle(vehicle, client, startDate, endDate)
confirmReservation(reservation)
initiateCancellation(reservation)
promptConfirmationOfCancellation(reservation)
confirmReservationCancellation(reservation)

4.1.4 Clerk - Make Return

Related to UC4 - Handle Return

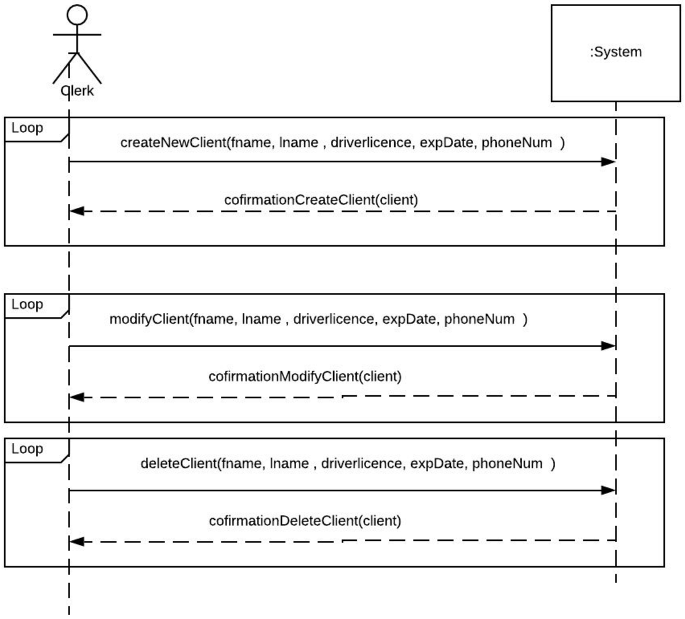
Assumption: The actor is a Clerk which has successfully logged-in as a Clerk, and has navigated to the “Returns” section.



4.1.5 Clerk - Manage Client Records

Related to UC7 - Manage Client Records

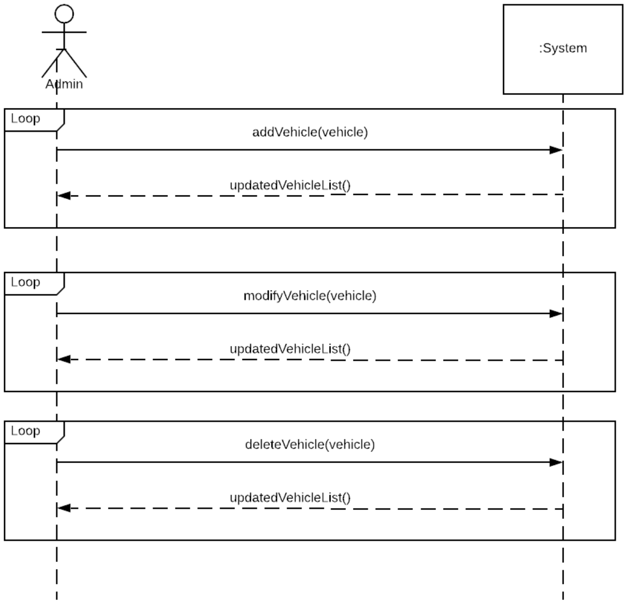
Assumption: The actor is a Clerk which has successfully logged-in as a Clerk, and has navigated to the “Clients” section.


Clerk
:System
Loop
createNewClient(fname, lname , driverlicence, expDate, phoneNum)
cofirmationCreateClient(client)
Loop
modifyClient(fname, lname , driverlicence, expDate, phoneNum)
cofirmationModifyClient(client)
Loop
deleteClient(fname, lname , driverlicence, expDate, phoneNum)
cofirmationDeleteClient(client)

4.1.6 Administrator - Manage Vehicle Records

<u>Related to UC6 - Manage Vehicle Records</u>

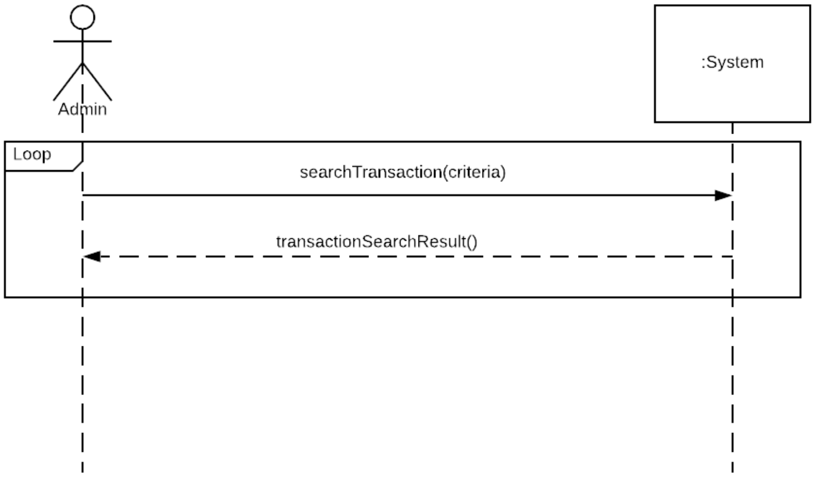
Assumption: The actor is an Administrator which has successfully logged-in as an Administrator, and has navigated to the “Vehicles” section.



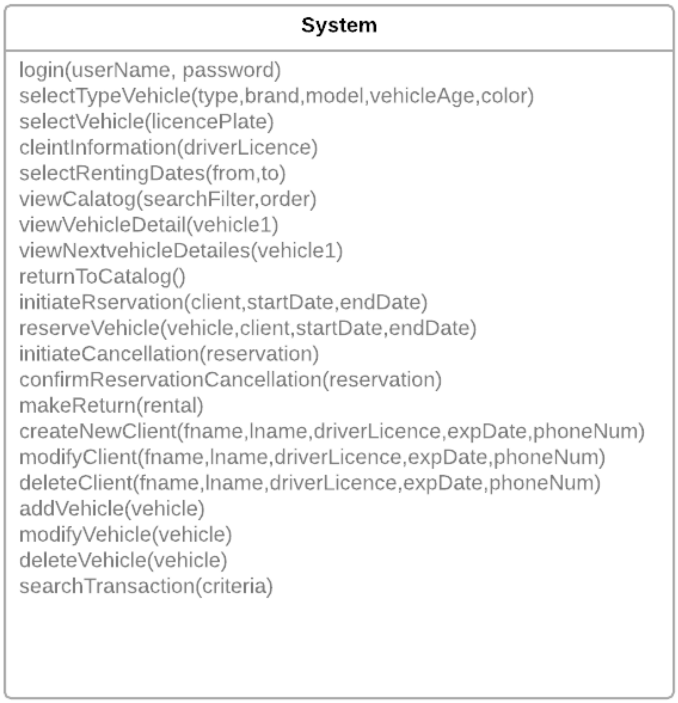
4.1.7 Administrator - View And Search History Transactions

<u>Related to UC10 - View And Search History Transactions</u>

Assumption: The actor is an Administrator which has successfully logged-in as an Administrator, and has navigated to the “Transactions” section.


Admin
:System
Loop
searchTransaction(criteria)
transactionSearchResult()

4.2 System Operations and Operation Contract



System

login(userName, password)
selectTypeVehicle(type,brand,model,vehicleAge,color)
selectVehicle(licencePlate)
cleintInformation(driverLicence)
selectRentingDates(from,to)
viewCalatog(searchFilter,order)
viewVehicleDetail(vehicle1)
viewNextvehicleDetailes(vehicle1)
returnToCatalog()
initiateRservation(client,startDate,endDate)
reserveVehicle(vehicle,client,startDate,endDate)
initiateCancellation(reservation)
confirmReservationCancellation(reservation)
makeReturn(rental)
createNewClient(fname,lname,driverLicence,expDate,phoneNum)
modifyClient(fname,lname,driverLicence,expDate,phoneNum)
deleteClient(fname,lname,driverLicence,expDate,phoneNum)
addVehicle(vehicle)
modifyVehicle(vehicle)
deleteVehicle(vehicle)
searchTransaction(criteria)

4.2.1 Login

Contract C01	login
Operation	login(username, password)
Use Case	Use Case: Login
Preconditions	
Postconditions	TBD

4.2.2 View Catalog

Contract C02	viewCatalog
Operation	viewCatalog(searchFilter, order)
Use Case	Use Cases: View and Search Catalog - Make Rental

Preconditions	A Clerk or and Administrator is logged in The Clerk or the Administrator is on the View Catalog page
Postconditions	A Catalog instance *catalog* was created *catalog.order* was set to order *catalog.vehicles* was set to the list of Vehicles matching the searchFilter criteria

4.2.3 View Vehicle Details

Contract C03	viewVehicleDetails
Operation	viewVehicleDetails(vehicle) - Make Rental
Use Case	Use Case: View and Search Catalog
Preconditions	A Clerk or an Administrator is logged in The Clerk or the Administrator is on the View Catalog page There exists at least one Vehicle instance in the Catalog
Postconditions	A VehicleDetail instance is created

4.2.4 View Next Vehicle Details

Contract C04	viewNextVehicleDetails
Operation	viewNextVehicleDetails(vehicle)
Use Case	Use Case: View and Search Catalog
Preconditions	A Clerk or an Administrator is logged in The Clerk or the Administrator is viewing a VehicleDetail There exists at least one more Vehicle instance in the Catalog, other than *vehicle*
Postconditions	A VehicleDetail instance vehicleDetails was created A Vehicle instance *vehicle2* is created *vehicleDetail.vehicle* is set to vehicle2

4.2.5 Select Renting Dates

Contract C05	selectRentingDates
Operation	selectRentingDates(from, to)
Use Case	Use Case: Make Rental
Preconditions	A Clerk is logged in The Clerk is viewing a VehicleDetail A vehicle is selected Time table given is available for renting

	A valid client has already been selected
Postconditions	Th selected vehicle will be rented for the selected time by the selected client

4.2.6 Initiate Reservation

Contract C06	initiateReservation
Operation	initiateReservation(client, startDate, endDate)
Use Case	Use Case: Make Reservation
Preconditions	A Clerk is logged in
Postconditions	A list of vehicles available at the given timeline is given

4.2.7 Reserve Vehicle

Contract C07	reserveVehicle
Operation	reserveVehicle(vehicle, startDate, endDate)
Use Case	Use Case: Make Reservation
Preconditions	A Clerk is logged in A list of vehicles available at the given timeline exist and the clerk is looking at it The clerk select one vehicle
Postconditions	A Vehicle is reserved

4.2.8 Initiate Cancellation

Contract C08	initiateCancellation
Operation	initiateCancellation(reservation)
Use Case	Use Case: Cancel Reservation
Preconditions	A Clerk is logged in The Clerk is viewing a reservation
Postconditions	The reservation enter the cancellation process

4.2.9 Confirm Reservation Cancellation

Contract C09	confirmReservationCancellation
Operation	confirmReservationCancellation(reservation)
Use Case	Use Case: Cancel Reservation

Preconditions	A Clerk is logged in The Clerk as started the cancelling process of the registration
Postconditions	The registration is cancelled

4.2.10 Make Return

Contract C10	makeReturn
Operation	makeReturn(rental)
Use Case	Use Case: Make return
Preconditions	A Clerk is logged in The Clerk is viewing a rented VehicleDetail
Postconditions	The vehicule is returned

4.2.11 Create New Client

Contract C11	createNewClient
Operation	createNewClient(fname ,lname, driverLicence, expDate, phoneNumber)
Use Case	Use Case: Manage Client Records
Preconditions	A Clerk is logged in The Clerk is viewing at the clients control panel
Postconditions	New client added

4.2.12 Modify Client

Contract C12	modifyClient
Operation	modifyClient(fname ,lname, driverLicence, expDate, phoneNumber)
Use Case	Use Case: Manage Client Records
Preconditions	A Clerk is logged in The Clerk is viewing at the clients control panel The Clerk select a Client
Postconditions	Client’s information have been updated

4.2.13 Client Informations

Contract C13	clientInformations
Operation	clientInformations(driverLisence)
Use Case	Use Case: Make Rental
Preconditions	A Clerk or an Administrator is logged in The Clerk or the Administrator is viewing a VehicleDetail The Clerk enter the renting phase and check if the client exist in the DB
Postconditions	Client existence in the DB is confirmed or ask to be created

4.2.14 Delete Client

Contract C14	deleteClient
Operation	deleteClient(fname ,lname, driverLicence, expDate, phoneNumber)
Use Case	Use Case: Manage Client Records
Preconditions	A Clerk is logged in The Clerk is viewing at the clients control panel The Clerk select a Client
Postconditions	The client is removed

4.2.15 Search Transaction

Contract C15	searchTransations
Operation	searchTransations(criteria)
Use Case	Use Cases: View and Search History of Transactions
Preconditions	Administrator is logged in
Postconditions	A TransactionSearchResult object *transactionSearchResult* is created. Transaction objects matching the search criteria are created and associated to the TransactionSearchResult.

4.2.16 Add vehicle

Contract C15	addVehicle
Operation	addVehicle(vehicle)
Use Case	Manage vehicle record
Preconditions	Administrator is logged in
Postconditions	The new vehicle added to the list.

4.2.17 Modify vehicle

Contract C15	modifyVehicle
Operation	modifyVehicle(vehicle)
Use Case	Manage vehicle record
Preconditions	Administrator is logged in
Postconditions	The new info for vehicle get modified to the list.

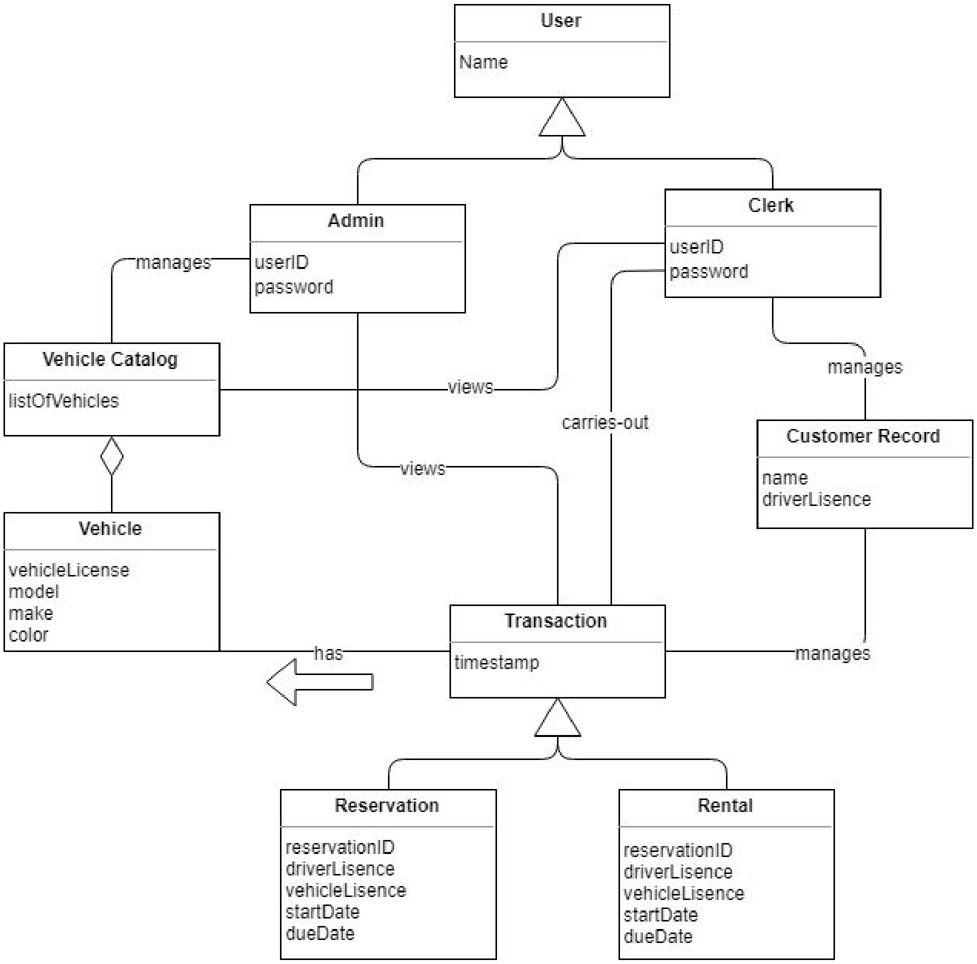
4.2.18 Delete vehicle

Contract C15	deleteVehicle
Operation	deleteVehicle(vehicle)
Use Case	Manage vehicle record
Preconditions	Administrator is logged in
Postconditions	The vehicle get deleted from the list.

4.2.18 Search Transaction

Contract C15	searchTransaction
Operation	searchTransaction(vehicle)
Use Case	Manage vehicle record
Preconditions	Administrator is logged in
Postconditions	Admin can search for transaction regards to creteria.

4.3 Conceptual Class Diagram (Domain Model)


User
Name
Admin
userID
password
Clerk
userID
password
manages
Vehicle Catalog
listOfVehicles
views
views
carries-out
manages
Customer Record
name
driverLisence
Vehicle
vehicleLicense
model
make
color
has
Transaction
timestamp
manages
Reservation
reservationID
driverLisence
vehicleLisence
startDate
dueDate
Rental
reservationID
driverLisence
vehicleLisence
startDate
dueDate