

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

Software Requirements Specification

Version 3.6

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:	Nov 2019

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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 Dat
26th Oct 2019	2.3	Review (Iteration 3 release)	Gurjot/Ravneet
13th Nov 2019	3.0	Updated section 3.3	Gurjot Singh
14th Nov 2019	3.1	Update for “version” attribute	Adrien Dat
15th Nov 2019	3.2	Added Non Functional requirements (3.3.3, 3.3.4, 3.3.5, 3.3.6)	Ravneet Singh Brar
15th Nov 2019	3.3	Updated UC9 and UC10 to show concurrency.	Ravneet Singh Brar
16th Nov 2019	3.4	Updates and first review	Adrien/Richard
17th Nov 2019	3.5	Updated section 2(High level view)	Gurjot Singh
17th Nov 2019	3.6	Review	Adrien Dat

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

Introduction	5
1.1 Scope	5
1.2 Definitions, acronyms, and abbreviations	5
1.3 References	5
Overall description	6
2.1 Product perspective	6
2.2 Major Product functions	6
2.3 User characteristics	6
2.4 Constraints	7
2.5 Assumptions and dependencies	7
Specific requirements	8
External interfaces	8
3.2 Functional requirements	8
3.2.1 Actor goal list	8
3.2.2 Use case view	9
3.3 Non-functional requirements	19
4. Analysis Models	20
4.1 System Sequence Diagram	20
4.1.1 Clerk - View and Search Catalog	21
4.1.2 Clerk - Make Rental	22
4.1.3 Clerk - Make Reservation	23
4.1.4 Clerk - Make Return	24
4.1.5 Clerk - Manage Client Records	25
4.1.6 Administrator - Manage Vehicle Records	27
4.2 System Operations and Operation Contract	27

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.1 Login	27
4.2.2 View Catalog	28
4.2.3 View Vehicle Details	28
4.2.4 View Next Vehicle Details	28
4.2.5 Select Renting Dates	28
4.2.6 Initiate Reservation	29
4.2.7 Reserve Vehicle	29
4.2.8 Initiate Cancellation	29
4.2.9 Confirm Reservation Cancellation	30
4.2.10 Make Return	30
4.2.11 Create New Client	30
4.2.12 Modify Client	30
4.2.13 Client Informations	31
4.2.14 Delete Client	31
4.3 Conceptual Class Diagram (Domain Model)	31

Figures

- Figure 1. A client-server architecture.6
 Figure 2. An example user interface..8
 Figure 3. Use case model.10

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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:

- <https://dzone.com/articles/introduction-to-web-applications-performance>

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

2. Overall description

The product we are designing is a web application. Illustrated below is the high level view of the system.

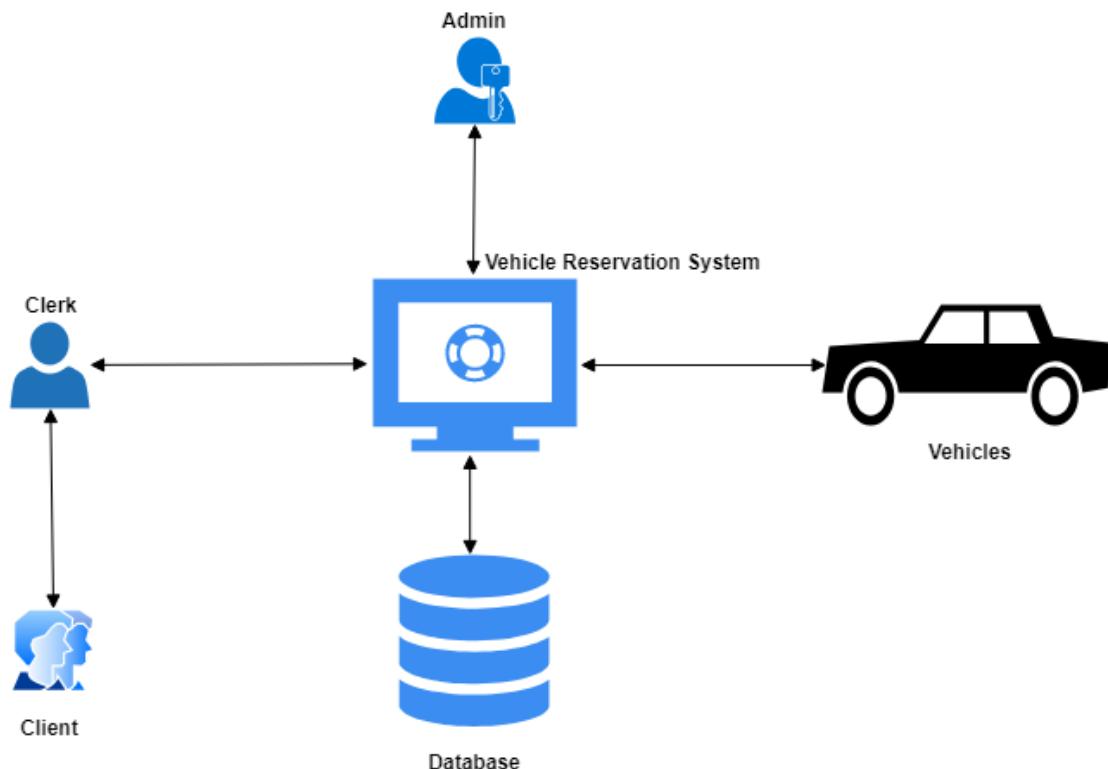


Figure 1. High level view of the system

2.1 Product perspective

Clients interact with the Clerks in order to rent/reserve/return/cancel a vehicle. The clients are able to perform rentals, reservations, returns, cancellations on the vehicles. In addition, they are also able to add a new client or update any kind of information for a client in the database. At one time, multiple clerks are able to perform all types of transactions concurrently. Clerks cannot perform any modifications on the vehicle records.

Admins can view the data about all the transactions carried out in the system. In addition, admins can add or modify vehicles in the database, but cannot perform all the other vehicle related transactions.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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

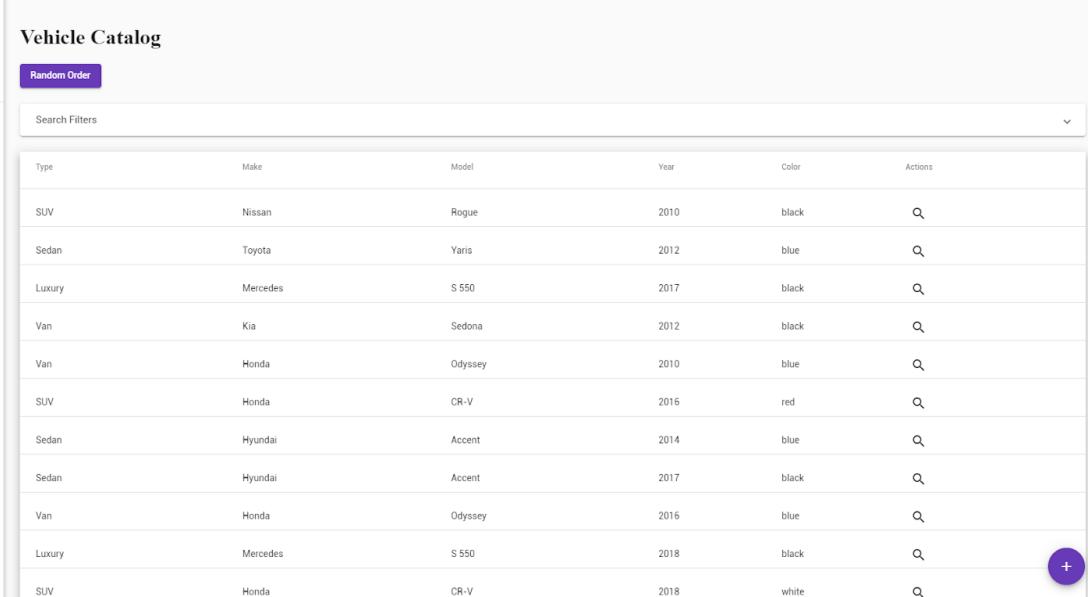
Since we are designing a web application, as long as the computer is equipped with an internet connection and a browser, the user should be able to access the application.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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



The screenshot shows a user interface for a 'Car Rental' system. On the left, a sidebar menu lists 'Vehicle Catalog', 'Client Records', 'Rentals', 'Reservations', 'Returns', and 'Logout'. The main area is titled 'Vehicle Catalog' and contains a table of vehicle data. A 'Random Order' button is visible above the table. The table has columns for Type, Make, Model, Year, Color, and Actions. The data includes:

Type	Make	Model	Year	Color	Actions
SUV	Nissan	Rogue	2010	black	
Sedan	Toyota	Yaris	2012	blue	
Luxury	Mercedes	S 550	2017	black	
Van	Kia	Sedona	2012	black	
Van	Honda	Odyssey	2010	blue	
SUV	Honda	CR-V	2016	red	
Sedan	Hyundai	Accent	2014	blue	
Sedan	Hyundai	Accent	2017	black	
Van	Honda	Odyssey	2016	blue	
Luxury	Mercedes	S 550	2018	black	
SUV	Honda	CR-V	2018	white	

Figures 2. An example user interface.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

3.2 Functional requirements

3.2.1 Actor goal list

Actor	Goal
Clerk	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

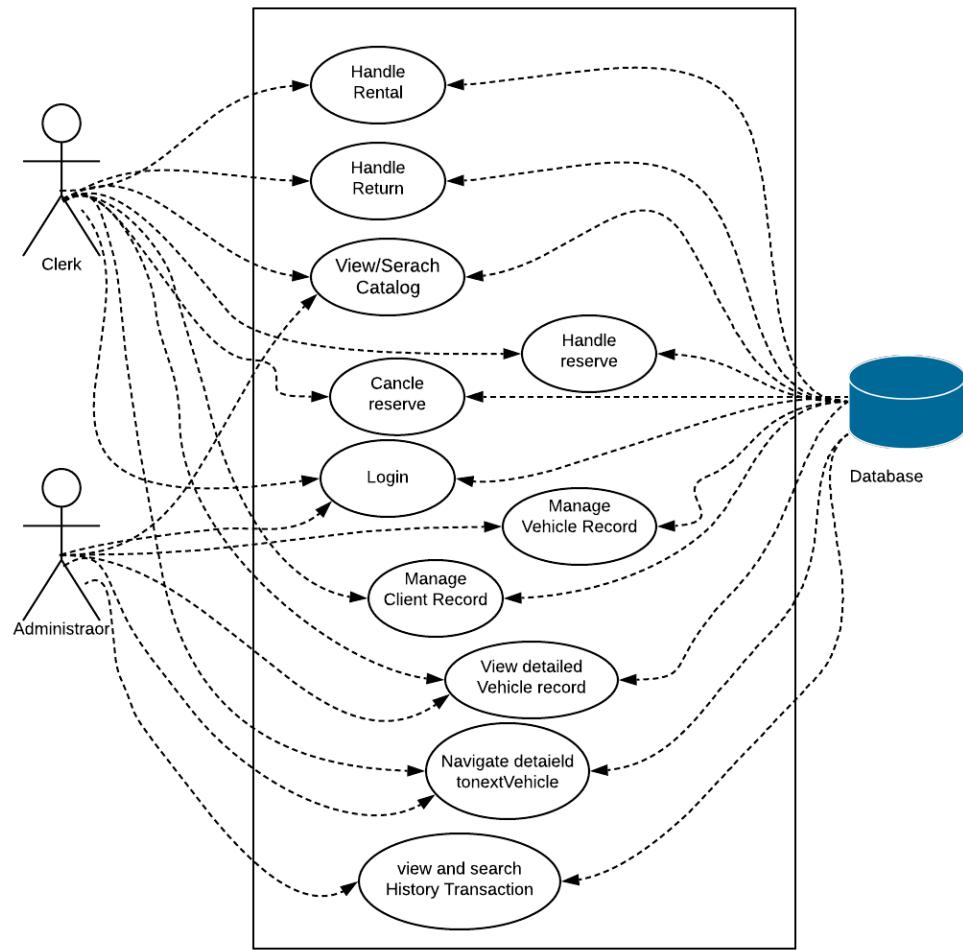


Figure 3. Use case model.

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

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

UC9:

Use Case Name:	Clerk Login
Actors:	Clerk
Description:	Clerk needs its credentials to login to the system in order to access the system.

UC10:

Use Case Name:	Administrator Login
Actors:	Administrator
Description:	Administrator needs its credentials to login to the system in order to access the system. Only one admin can login to the system at any point of time

UC11:

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.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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:	<ol style="list-style-type: none"> 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 <i>Vehicle Catalog</i>. 4. The admin or clerk can select to view all the vehicles in a random order, or by selecting filtering criteria. 5. Clicks on search button. 6. Then the system displays all information about the vehicle based on selected list.
PRE CONDITION	<ul style="list-style-type: none"> ● Clerk or Administrator is authenticated.
POST CONDITION	<ul style="list-style-type: none"> ● The list is displayed according to the order and filters selected
EXCEPTION / Alternatives	2a. A clerk do not pass the authenticated.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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:	<ol style="list-style-type: none"> 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	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● System send confirmation to the Clark ● Database get updated with all new information
EXCEPTION / Alternatives	<p>2a. A clerk do not pass the authenticated.</p> <p>3a. A desired car is not available.</p> <p>6a. The system don't sent the confirmation.</p>

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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:	<ol style="list-style-type: none"> 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	<ul style="list-style-type: none"> ● Clerk is authenticated. ● Car is available on that dates
POST CONDITION	<ul style="list-style-type: none"> ● Display the confirmation ● Database updated with new information
EXCEPTION / Alternatives	<p>2a. A clerk do not pass the authenticated.</p> <p>5a. A car is not available on that particular dates</p>

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

NAME	Cancel Reservation
ID	UC3
DESCRIPTION	Based on some info, the clerk can cancel an already existing reservation.
ACTORS	Clerk
Main Success Scenario:	<p>A clerk logged in to the system successfully.</p> <p>The system authenticated the username and password.</p> <p>The a clerk clicks on search reservation list.</p> <p>The system displays list of reservation and allow us to have filter based on the type of car or dates.</p> <p>A clerk pick the reservation which the customer will cancel.</p> <p>Clicks on cancellation button.</p> <p>The system displays a confirmation of cancellation.</p>
PRE CONDITION	<ul style="list-style-type: none"> • Clerk is authenticated. • Car display on reservation list.
POST CONDITION	<ul style="list-style-type: none"> • Display the confirmation • Database updated with new information
EXCEPTION / Alternatives	<p>2a. A clerk do not pass the authenticated.</p> <p>5a. A car is not available on the list.</p>

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ul style="list-style-type: none"> • 1. Catalog is updated with the new/modified information.
EXCEPTIONS/ Alternatives	<ol style="list-style-type: none"> 2a. Administrator fails to log-in. 5a. Information admin wishes to modify does not exist in the catalog.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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	<ol style="list-style-type: none"> 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - System is updated with the new/modified information.
EXCEPTIONS/Alternatives	<p>2a. Clerk fails to log-in.</p> <p>5a. Client Record that clerk wishes to modify does not exist in the system.</p>

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

NAME	Administrator Log in
ID	UC10
DESCRIPTION	Administrators can access the system after entering its login credentials
ACTORS	Administrator
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin logs in to the system 2. System authenticates the Username and password of the admin. 3. Administrator is logged in to the system and can perform any operation it wants.
PRE-CONDITIONS	No other Admin is logged in.
POST-CONDITION	Administrator is logged into the system and can perform its tasks.
EXCEPTIONS/Alternatives	<ul style="list-style-type: none"> - Invalid login credentials. - Another admin is already logged in

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

NAME	View and Search History of Transactions
ID	UC11
DESCRIPTION	Administrators can access,view and search the history of transactions per client, per vehicle, or per due date.
ACTORS	Admin
Main Success Scenario	<ol style="list-style-type: none"> 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 <i>transactionSearchResult</i> 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 allows at most one administrator being logged in at a time. The system allows 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.'

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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 is provided by a relational database(MySQL), the object relational mappings have been implemented to map objects to the database tables, as shown in the SAD document.

3.3.3 Performance

Performance of a web based system is calculated on the basis of the following metrics:

- Response time : It measures how much time the application takes to respond to the users request. It is the most common metric used to evaluate the efficiency of the system.
- Resource Consumption: It measures the amount of resources(CPU, Memory and Disk Space) that are being consumed by the application while it is running.
- Latency- It is the time in which the user gets any kind of response from the system even if the amount of work to be done by the system is minimum.
- System Availability - Fraction of time when the system is available as opposed to when it is not, is System availability. Used by many companies as a system uptime in Service Legal Agreements

Web Rental System will be able to respond to the user requests immediately without any visible latency and will be available to the user as long as the system is connected to the Internet..

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

3.3.4 Compatibility

Compatibility of a system is measured in terms of the fact that how efficiently the system is able to perform its tasks while sharing the resources with the other applications.

Rental System will be able to work efficiently while sharing resources of the system with the other applications running on the system without affecting its own performance and performance of the other applications.

3.3.5 Usability

Usability of a system can be defined in terms of how easily and accurately a system is able to perform all user tasks. In the Application, Administrator and Clerk will be able to perform its tasks accurately and efficiently.

3.3.6 Security

It involves the ability of the system to protect sensitive and private data of the users and to avoid any unauthorised access to the system. Rental System application will be accessed only by the authorized users with the correct login credentials

3.3.7 Other

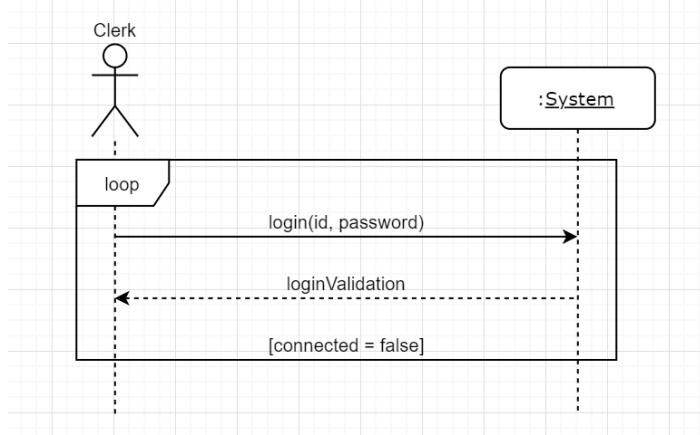
You must design and implement the application in an object-oriented environment with an appropriate architectural style that supports quality characteristics such as separation of concerns, modularity, reusability and adaptability

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

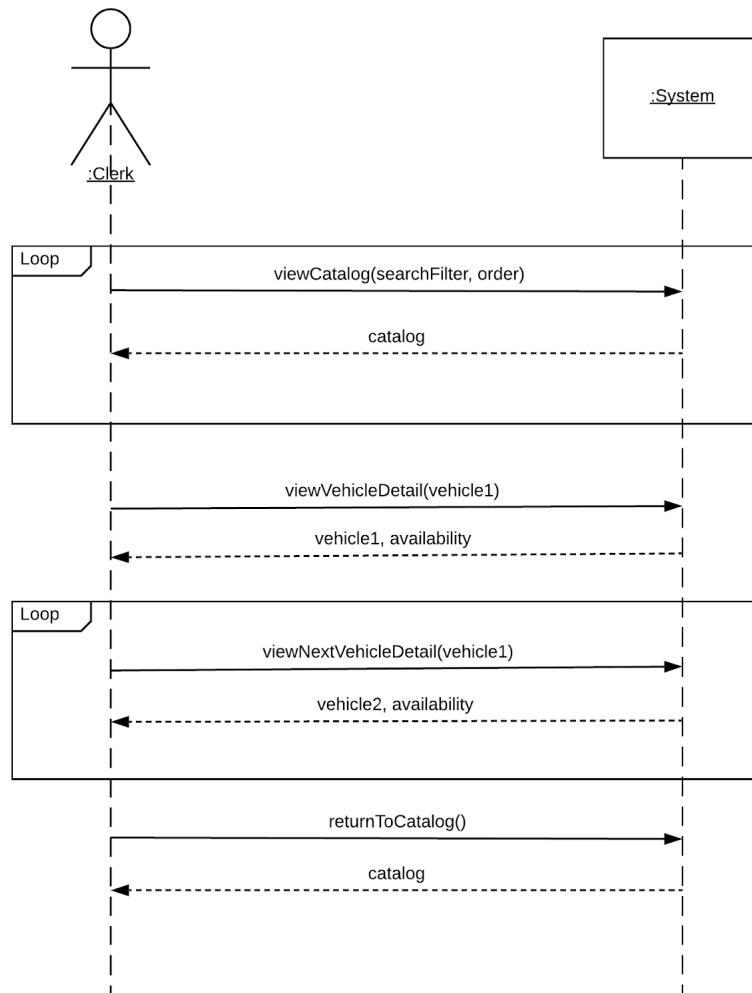


Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.1.1 Clerk - View and Search Catalog

Related to UC1 - Search Catalog

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



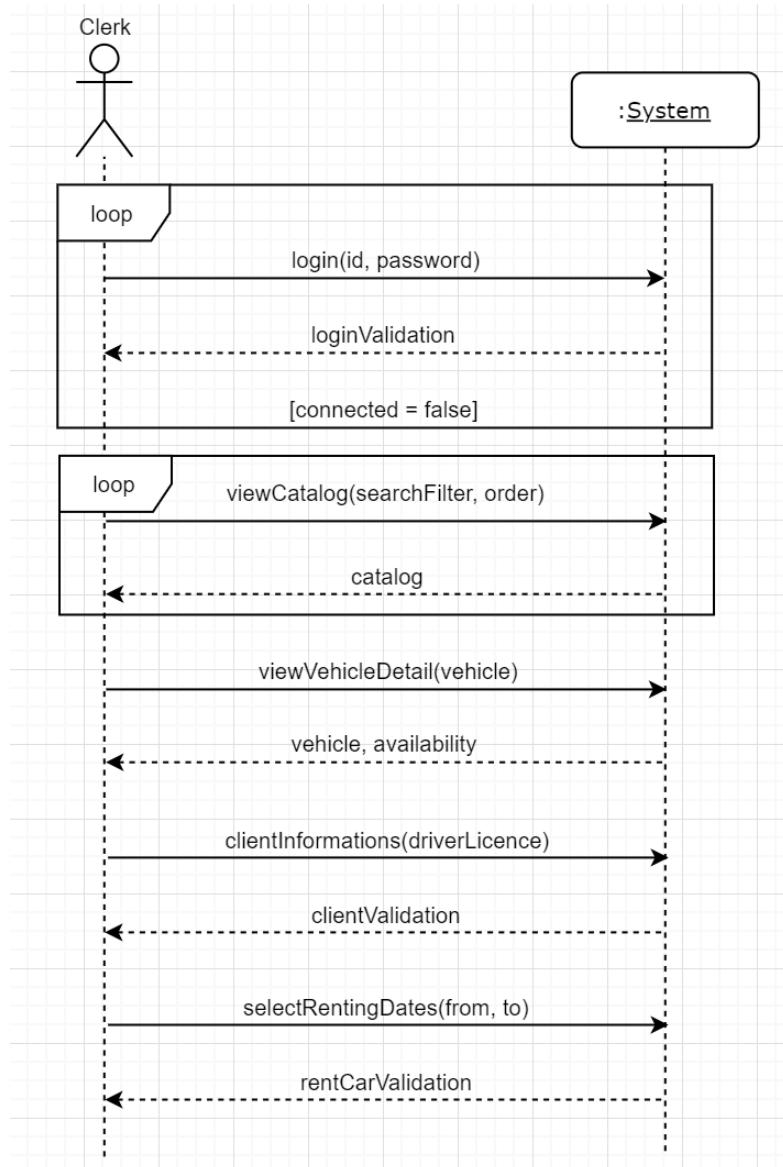
Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

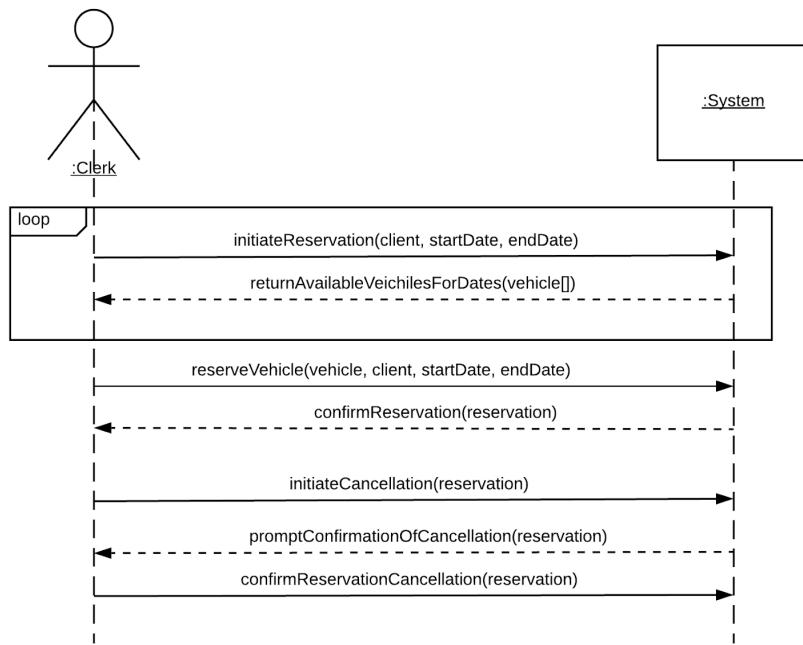


Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

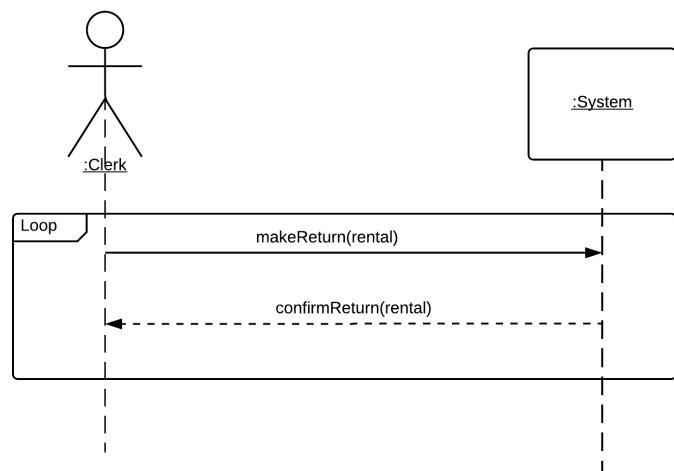


Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

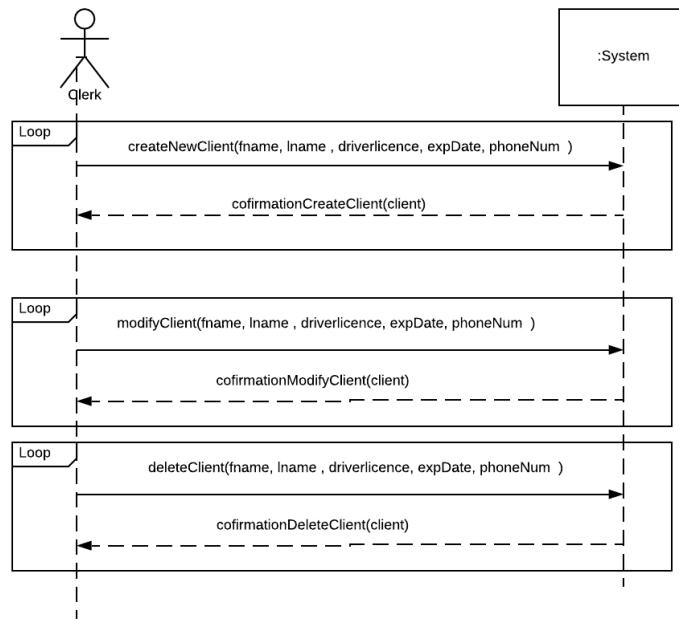


Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

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.

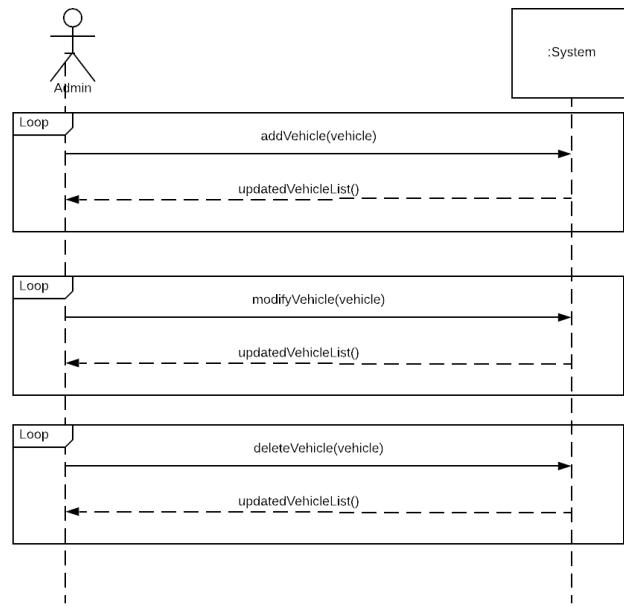


Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.1.6 Administrator - Manage Vehicle Records

Related to UC6 - Manage Vehicle Records

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

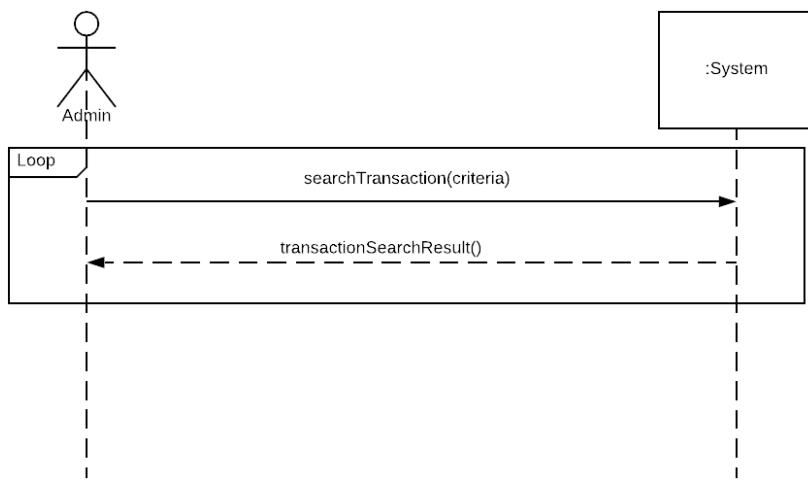


Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.1.7 Administrator - View And Search History Transactions

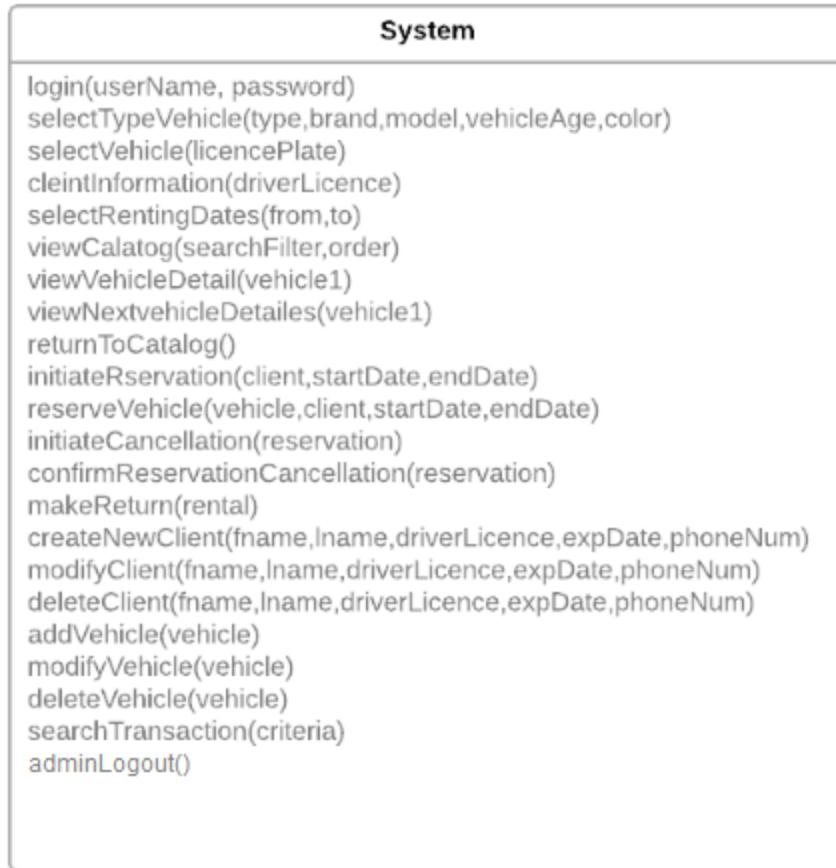
Related to UC10 - View And Search History Transactions

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



Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2 System Operations and Operation Contract



Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.1 Clerk Login

Contract C01	Clerk Login
Operation	login(username, password)
Use Case	Use Case: Clerk Login
Preconditions	
Postconditions	<p>LogInResponse <i>logInResponse</i> is created <i>logInResponse.isSuccess</i> = true <i>logInResponse.username</i> = username <i>logInResponse.role</i> = "clerk"</p>

4.2.2 Administrator Login

Contract C01	Administrator Login
Operation	login(username, password)
Use Case	Use Case: Administrator Login
Preconditions	<i>UserController.hasAdmin</i> = false
Postconditions	<p>LogInResponse <i>logInResponse</i> is created <i>logInResponse.isSuccess</i> = true <i>logInResponse.username</i> = username <i>logInResponse.role</i> = "admin" <i>UserController.hasAdmin</i> = true</p>

4.2.3 View Catalog

Contract C02	viewCatalog
Operation	viewCatalog(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	<p>If none existed before, an VehicleIdentityMap <i>vehicleIdentityMap</i> is created, then all of the active Vehicles found in the database are mapped to Vehicle instances and added to <i>vehicleIdentityMap</i>.</p> <p>If a <i>vehicleIdentityMap</i> existed, there is no new database access.</p> <p>The instances in <i>vehicleIdentityMap</i> are returned to the frontend and ordered and filtered according to the request.</p>

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.4 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

4.2.5 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 <i>vehicle</i>
Postconditions	A VehicleDetail instance <i>vehicleDetails</i> was created A Vehicle instance <i>vehicle2</i> is created <i>vehicleDetail.vehicle</i> is set to <i>vehicle2</i>

4.2.6 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	The selected vehicle will be rented for the selected time by the selected client

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.7 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

4.2.8 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

4.2.9 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

4.2.10 Confirm Reservation Cancellation

Contract C09	confirmReservationCancellation
Operation	confirmReservationCancellation(reservation)
Use Case	Use Case: Cancel Reservation
Preconditions	A Clerk is logged in The Clerk has started the cancelling process of the registration
Postconditions	The registration is cancelled

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.11 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 vehicle is returned

4.2.12 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

4.2.13 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

4.2.14 Client Informations

Contract C13	clientInformations
Operation	clientInformations(driverLicence)
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

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.15 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 Instance c of Client exists
Postconditions	The <i>active</i> property of c is set to 0.

4.2.16 Search Transaction

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

4.2.17 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.

4.2.18 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.

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.2.19 Delete vehicle

Contract C15	deleteVehicle
Operation	deleteVehicle(vehicle)
Use Case	Manage vehicle record
Preconditions	Administrator is logged in. Vehicle instance <i>v</i> exists
Postconditions	The <i>active</i> property of <i>v</i> is set to 0.

4.2.20 Search Transaction

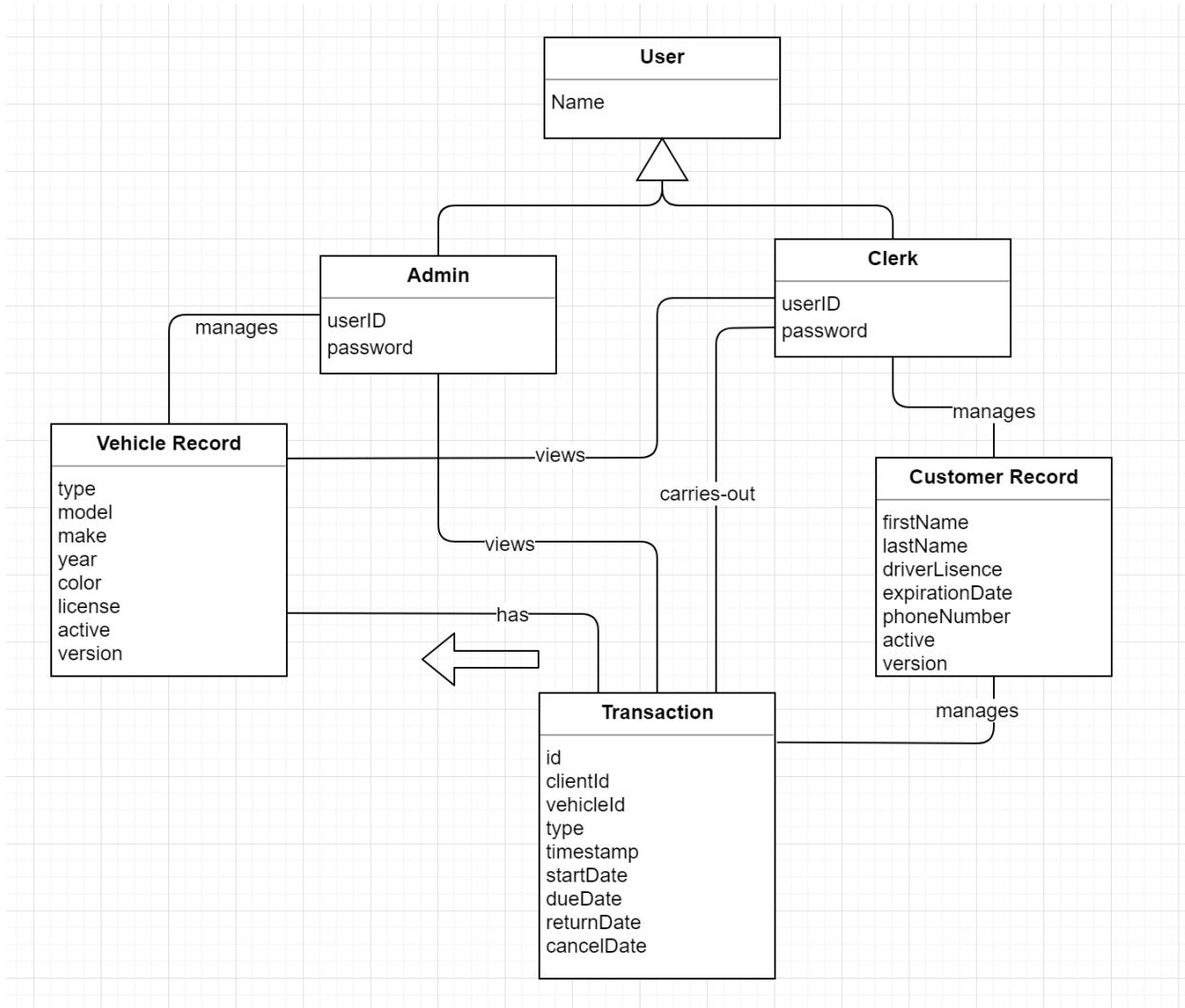
Contract C15	searchTransaction
Operation	searchTransaction(vehicle)
Use Case	Manage vehicle record
Preconditions	Administrator is logged in
Postconditions	Admin can search for transaction regards to criteria.

4.2.2 Administrator Logout

Contract C01	Administrator Logout
Operation	adminLogout()
Use Case	Use Case: Administrator Logout
Preconditions	<i>UserController.hasAdmin</i> = false
Postconditions	LogInResponse <i>logInResponse</i> is created <i>logInResponse.isSuccess</i> = true <i>logInResponse.username</i> = username <i>logInResponse.role</i> = "admin" <i>UserController.hasAdmin</i> = true

Soen6461 Project	Version: 3.6
Software Requirements Specification	Date: 17/11

4.3 Conceptual Class Diagram (Domain Model)



Note: As you can see, there is no return object in this diagram, because in our implementation, it is not needed. Indeed, you can see a returnDate attribute in the Transaction table, which allows us to know if vehicle has been returned and when, or not yet. Furthermore, this implementation allows us to conserve information about the transaction like its type (rental or reservation), without having multiple tables that will repeat the same parameters. It also allows us to know if a vehicle has been returned before or after its due date.