Car Rental System

**Students**

Rares Dumitru Bunea: 266983

Anatoli Dunai 269643

Vladiimir Rotaru 266814

**ICT Engineering, VIA University College, Horsens**

**Abstract**

***The purpose of this project is to create a room monitoring system from which a user will be able to rent a car for a period. Each user solicitation will be presented to an administrator that will set a verdict for it.***

***This system will increase efficiency in both in users applying for renting a car and also for administrators to keep track and manage different parts of the system such as handing requests or cars.***

***The benefit of using this system is that the information will be stored on a database which provides scalability and a more fast and efficient manner.***

Table of Contents

[1 Introduction 5](#_Toc16707634)

[2 User stories and requirements 6](#_Toc16707635)

[2.1 User stories 6](#_Toc16707636)

[2.2 Requirements 6](#_Toc16707637)

[2.2.1 Functional requirements 6](#_Toc16707638)

[2.2.2 Nonfunctional requirements 6](#_Toc16707639)

[3 Analysis 7](#_Toc16707640)

[3.1 Use case diagram 7](#_Toc16707641)

[3.2 Use case description 7](#_Toc16707642)

[3.2.1 Choose period 7](#_Toc16707643)

[3.2.2 Visualize cars 8](#_Toc16707644)

[3.2.3 Request car 8](#_Toc16707645)

[3.2.4 Filter cars 9](#_Toc16707646)

[3.2.5 Add car 9](#_Toc16707647)

[3.2.6 View cars 10](#_Toc16707648)

[3.2.7 Delete car 10](#_Toc16707649)

[3.2.8 Search car 11](#_Toc16707650)

[3.2.9 Edit car 11](#_Toc16707651)

[3.2.10 View requests 12](#_Toc16707652)

[3.2.11 Manage Requests 12](#_Toc16707653)

[3.3 Activity diagrams 13](#_Toc16707654)

[3.3.1 Add car 13](#_Toc16707655)

[3.3.2 Delete car 13](#_Toc16707656)

[3.3.3 Edit car 14](#_Toc16707657)

[3.3.4 Manage requests 14](#_Toc16707658)

[3.3.5 Request 15](#_Toc16707659)

[3.4 Domain model diagram 16](#_Toc16707660)

[3.5 System sequence diagram 17](#_Toc16707661)

[3.5.1 Add car 17](#_Toc16707662)

[3.5.2 Delete car 18](#_Toc16707663)

[3.5.3 Edit car 18](#_Toc16707664)

[3.5.4 Manage requests 19](#_Toc16707665)

[3.5.5 Request 19](#_Toc16707666)

[4 Design 20](#_Toc16707667)

[4.1 Conceptual diagram 20](#_Toc16707668)

[5 Implementation 20](#_Toc16707669)

[5.1 IOT Implementation 20](#_Toc16707670)

[5.2 Database Implementation 21](#_Toc16707671)

[5.3 Android Implementation (A. V. and R. D. B.) 21](#_Toc16707672)

[5.3.1 Firebase implementation (A. V. and R. D. B.) 22](#_Toc16707673)

[5.3.2 UI implementation (A. V. and R. D. B.) 25](#_Toc16707674)

[5.3.3 Android application architecture (A. V. and R. D. B.) 26](#_Toc16707675)

[6 Testing 26](#_Toc16707676)

[6.1 Android Testing (A. V. and R. D. B.) 26](#_Toc16707677)

[7 Conclusions 32](#_Toc16707678)

[7 Table of figure 33](#_Toc16707679)

[8 References 33](#_Toc16707680)

[9 Appendices 33](#_Toc16707681)

[9.1 Appendix 1 Project Description 33](#_Toc16707682)

# 1 Introduction

The car rental industry is a worldwide and local business. It is a long-term growing industry, periodic and seasonable. The decisive factor to a prosperous car rental company is efficacious asset management (the most of assets are the rentable cars). .[1]

A car rental agency is a company that leases automobiles, usually for shot periods of time, generally ranging from a few days to a few weeks. Car rental agencies primarily serve people who are in need of a temporary vehicle, people that don’t own a car, owners of damaged or destroyed vehicles of travelers/tourists.[2]

Through this system, users will be able to easily filter through cars and pick the suitable one for ones needs. Employees will overview the entire process form the comfort of their office.

A system that can ease the process of renting a car, both for user and administrator, is a system that will bring value and profit to the car rental company.

# 2 User stories and requirements

## 2.1 User stories

1. As a user, I want to be able to Choose a period so that I can Visualize cars available between the desired dates.
2. As a user, I want to be able to Visualize Cars so that I can request a car.
3. As a user, I want to be able to Visualize Cars so that I can filter the displayed cars.
4. As an administrator, I want to be able to set car info, set deposit, set price so that I can add a car.
5. As an administrator, I want to be able to View cars so that I can delete a car or edit a car.
6. As an administrator, I want to be able to View cars so that I can search for a car.
7. As an administrator, I want to be able to Edit a car so that I can change car info, change the deposit or change the price.
8. As an administrator, I want to be able to View requests so that I can reject or accept requests from the customers.

## 2.2 Requirements

Before the project could begin, some core requirements had to be determined. During this phase of development, every core feature was determined and added to the functional requirement table. In addition, the were a multitude of non-functional requirements that the project had to adhere to. Both functional and non-functional requirements will be covered in the following chapter.

### 2.2.1 Functional requirements

1. User will be able to view available cars for booking.
2. User will be able to send a request for renting a car.
3. User will be able to search for a certain car (by sorting or filtering)
4. User will be able to choose a period of a time when he wishes to book a car (a list of available cars will be pop up)
5. Admin will be able to add, edit, delete a car from the system.
6. Admin will be able to accept or reject requests for car booking from users.
7. Admin will be able to set/change the price of the rented car per day.
8. Admin will be able to set/change the deposit of the car.

### 2.2.2 Nonfunctional requirements

1. Presentation tier should be implemented in Java.
2. Business logic should be implemented in C#.

# 3 Analysis

## 3.1 Use case diagram

As in the diagram displayed below, the system will have to type of actors: User and Administrator.

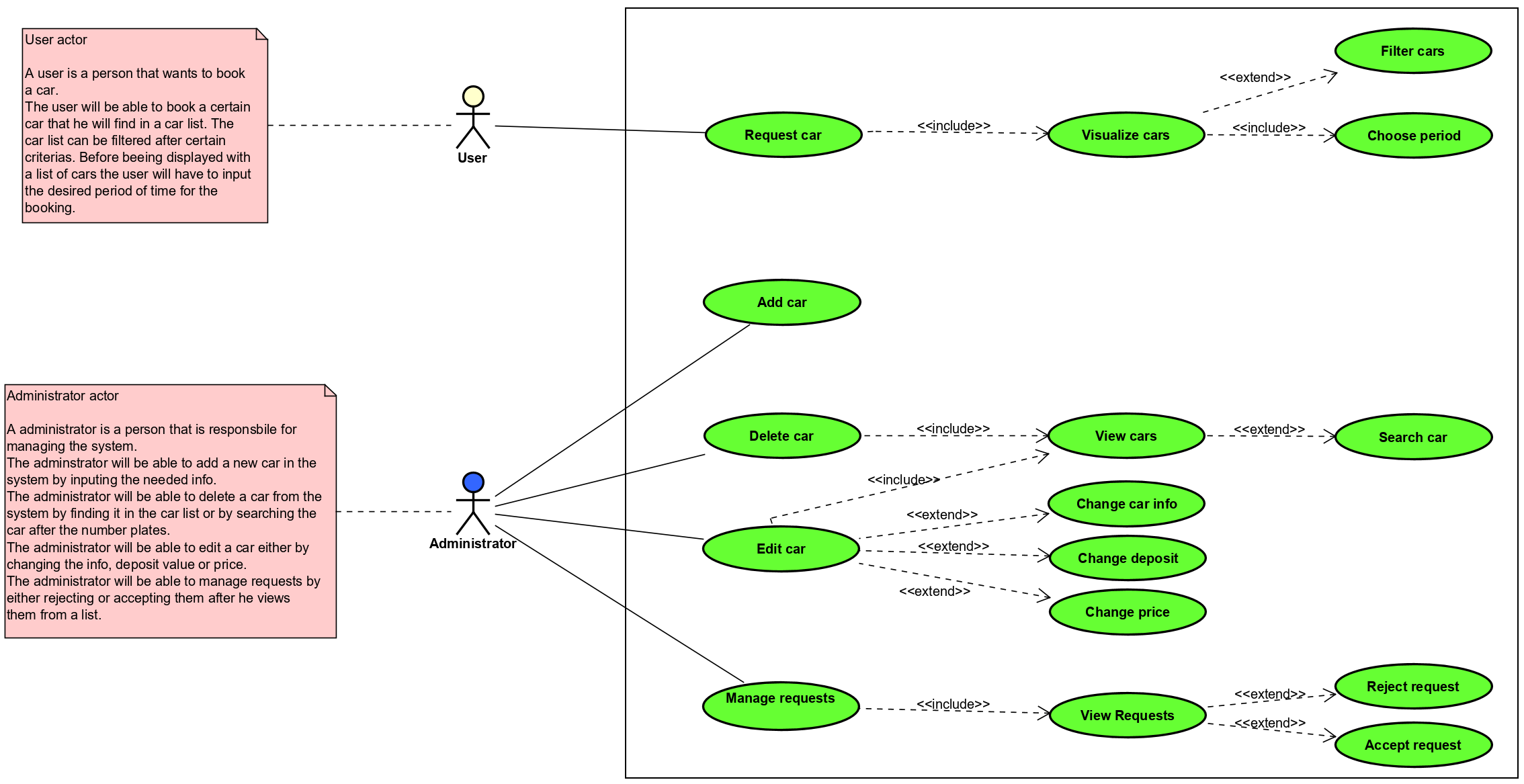


Figure 1-Use case diagram

## 3.2 Use case description

For each use case diagram, a use case description has been created to explain the basic steps and way of thinking for fulfilling different actions across the system. Displayed in logical order order, the use cases are:

### 3.2.1 Choose period

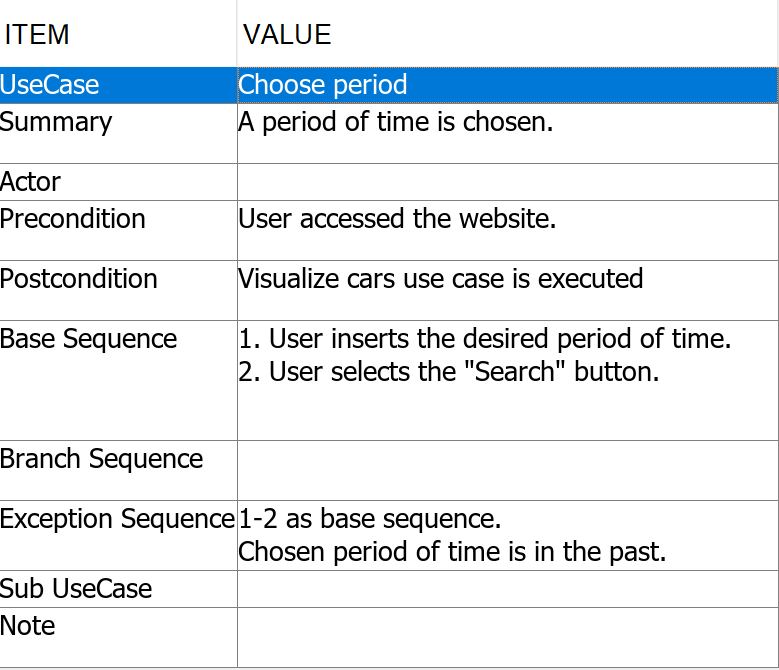


Figure 2- Use case description- Choose period

### 3.2.2 Visualize cars

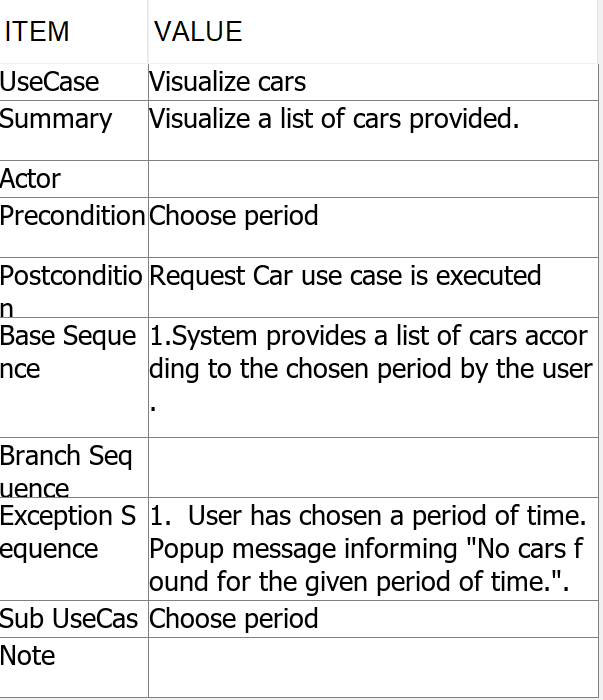


Figure 3- Use case description- Visualize cars

### 3.2.3 Request car

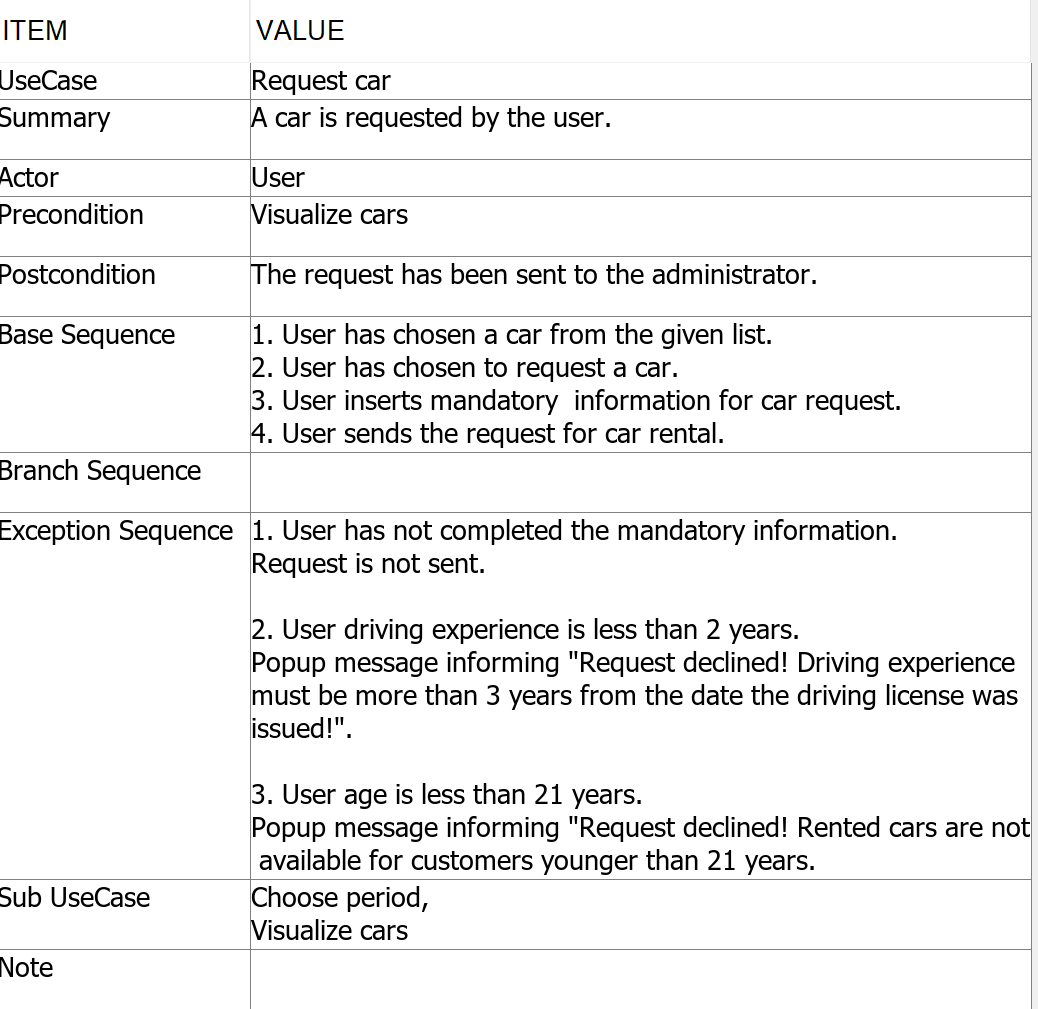


Figure 4- Use case description- Request car

### 3.2.4 Filter cars

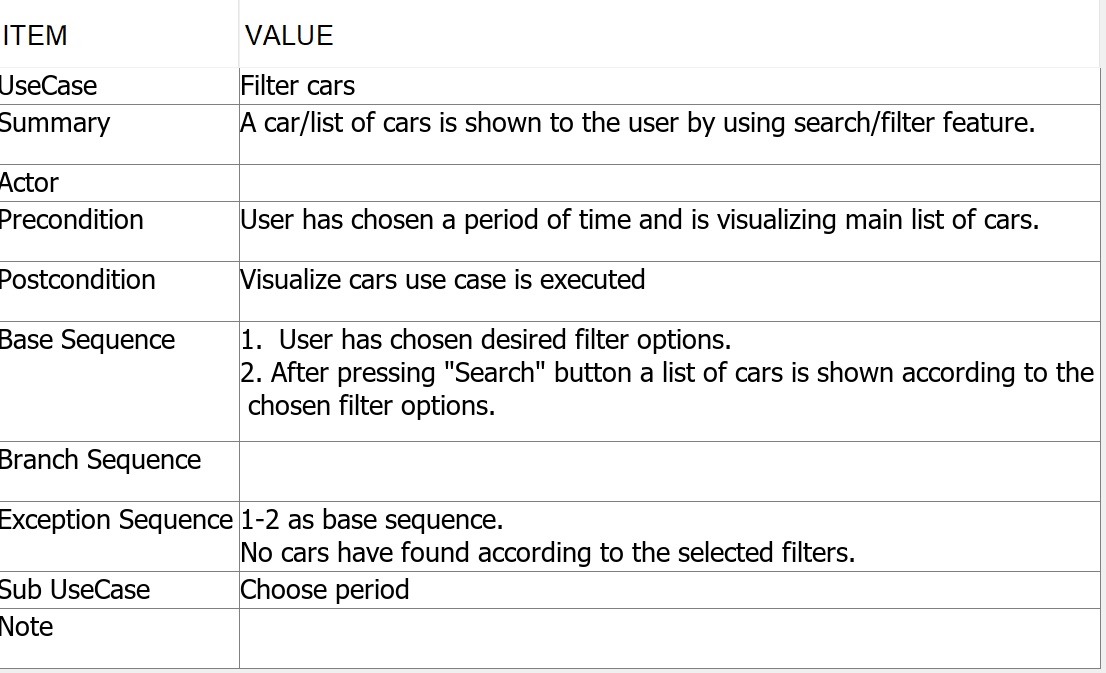


Figure 5- Use case description- Filter cars

### 3.2.5 Add car

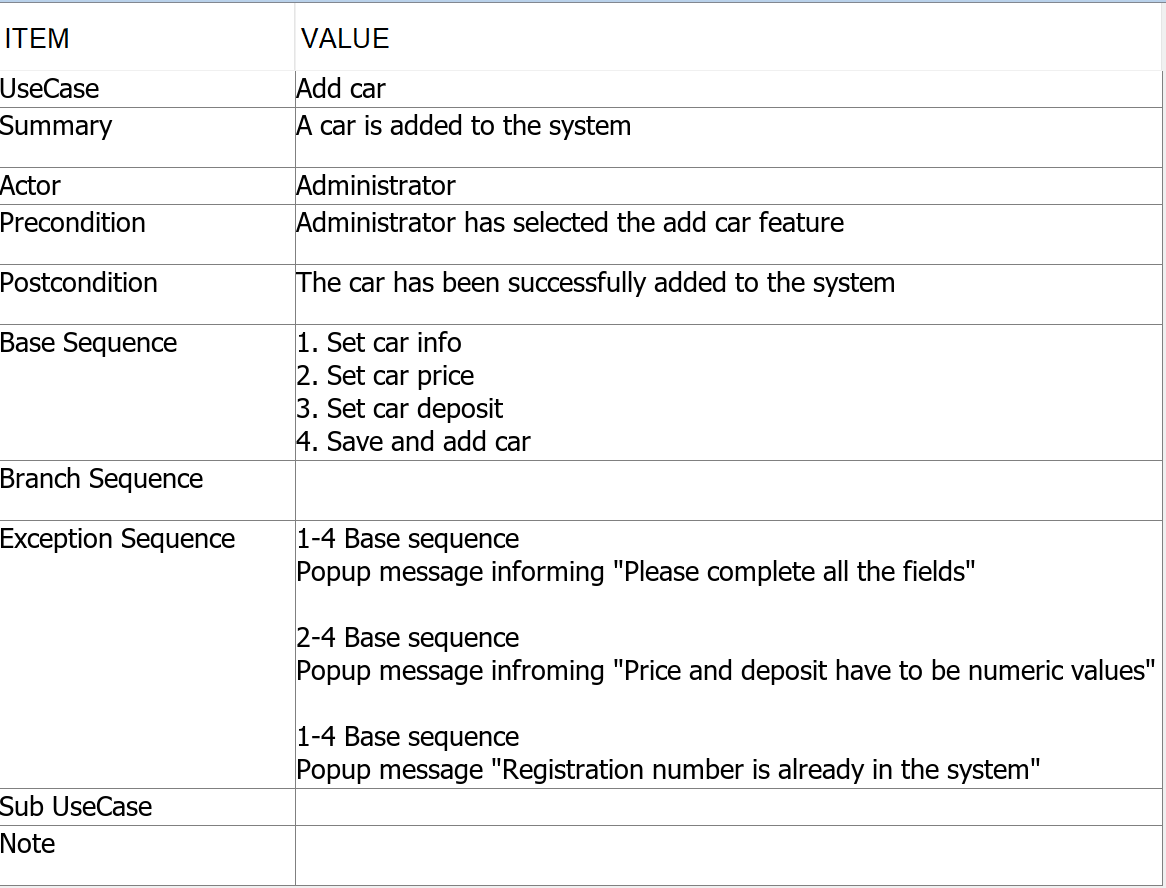


Figure 6- Use case description- Add car

### 3.2.6 View cars

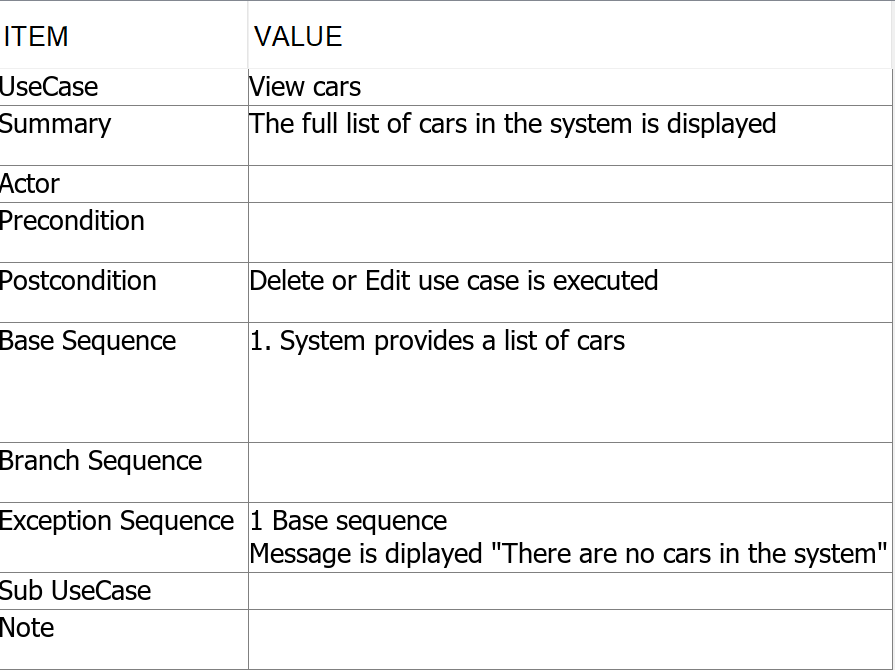


Figure 7- Use case description- View cars

### 3.2.7 Delete car

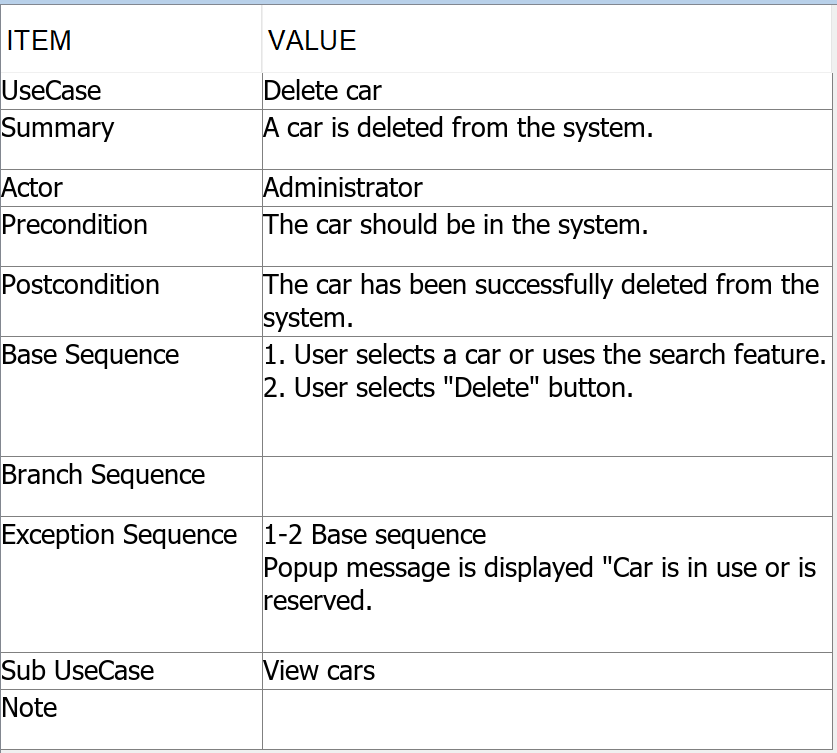


Figure 8- Use case description- Delete car

### 3.2.8 Search car

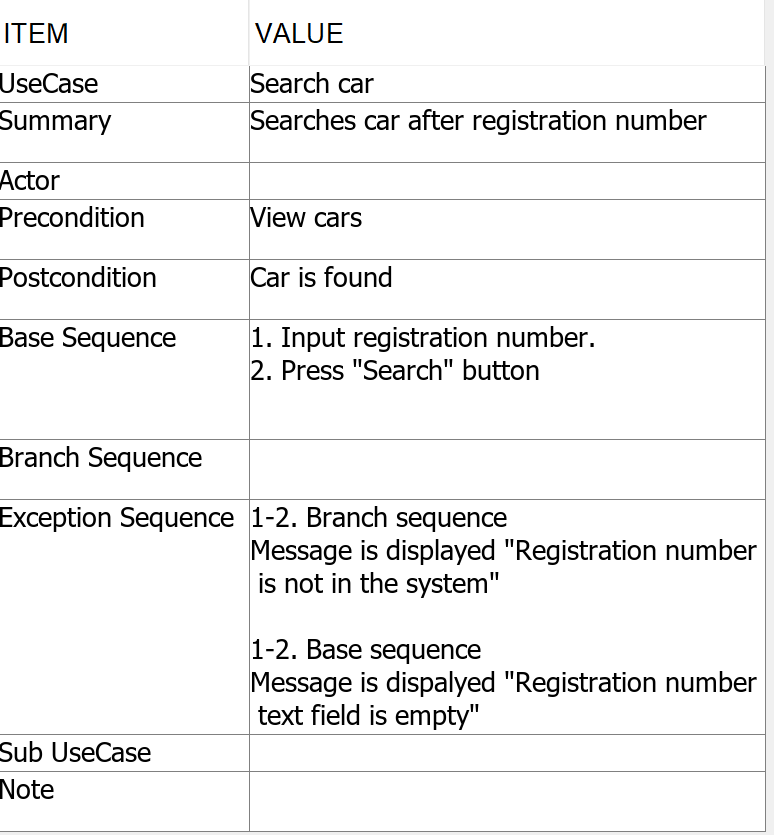


Figure 9- Use case description- Search car

### 3.2.9 Edit car

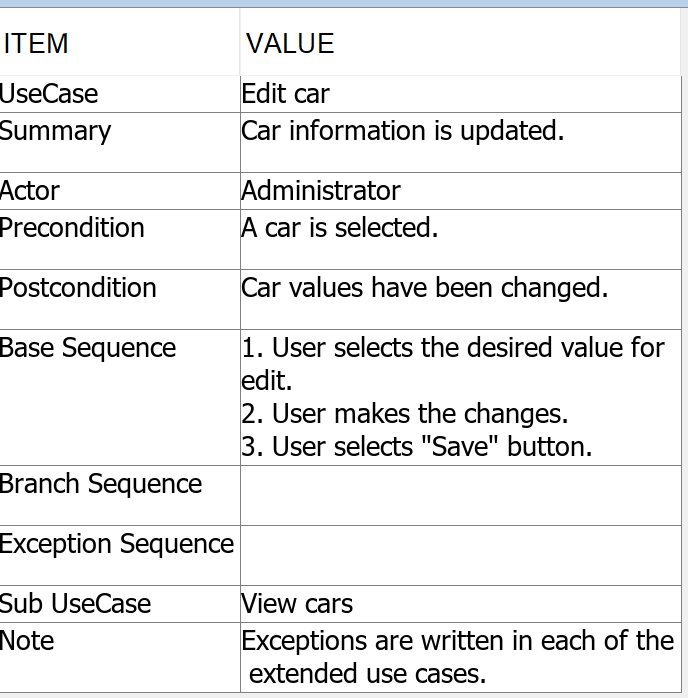


Figure 10- Use case description- Edit car

### 3.2.10 View requests

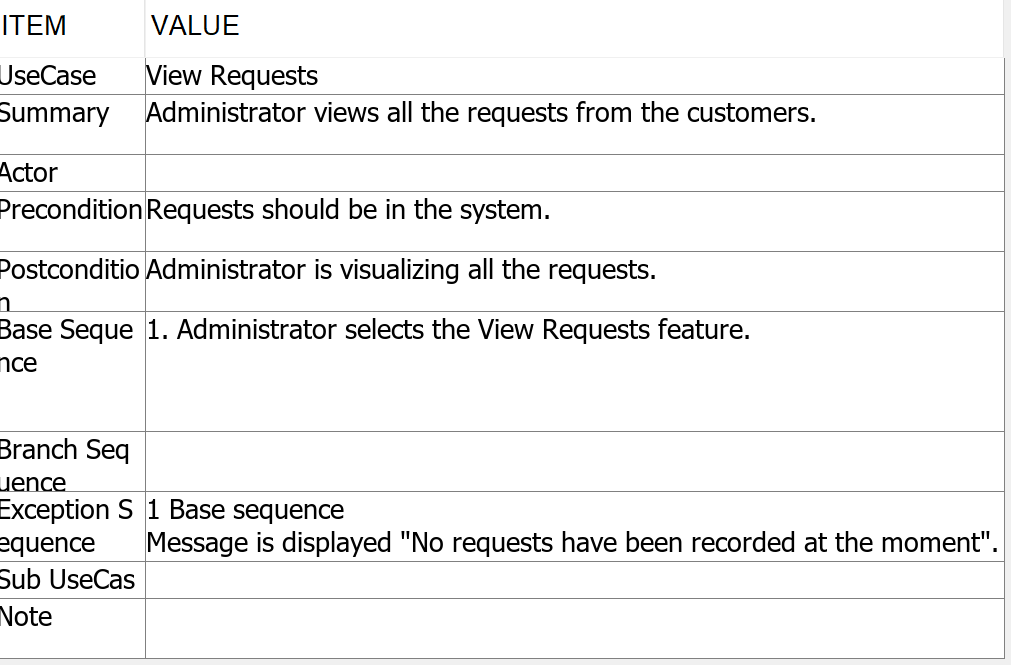


Figure 11- Use case description- View requests

### 3.2.11 Manage Requests

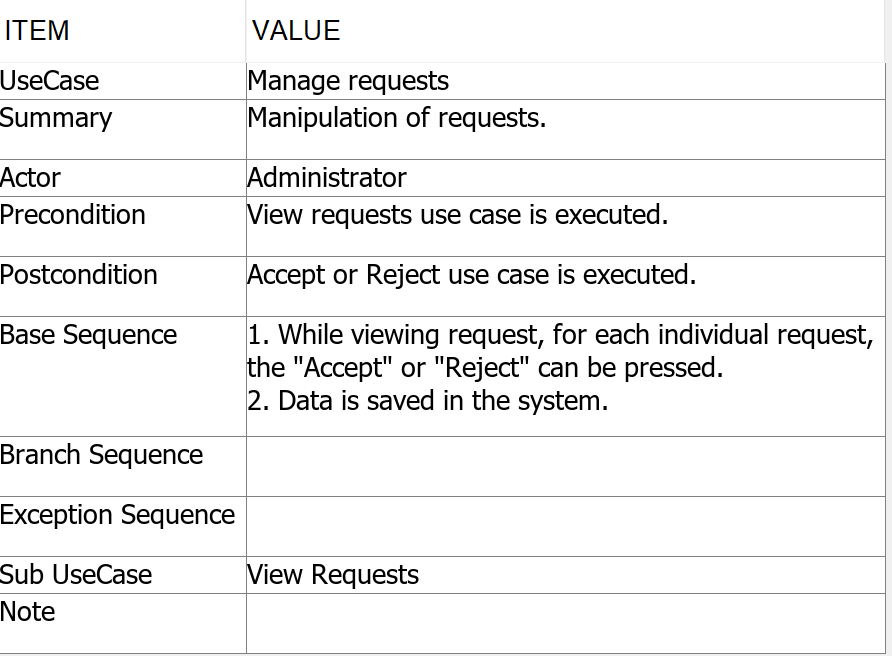


Figure 12- Use case description- Manage Requests

## 3.3 Activity diagrams

From the uses cases, five of them have been identified as being main use cases. For them, activity diagrams have been created.

### 3.3.1 Add car

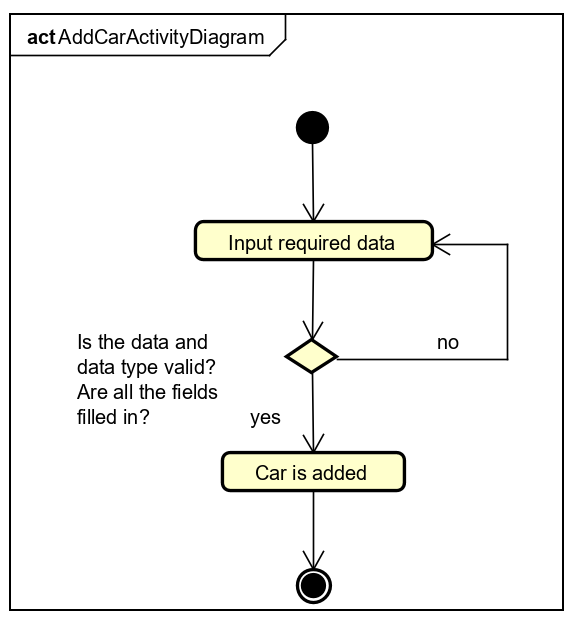


Figure 13- Activity diagram- Add car

### 3.3.2 Delete car

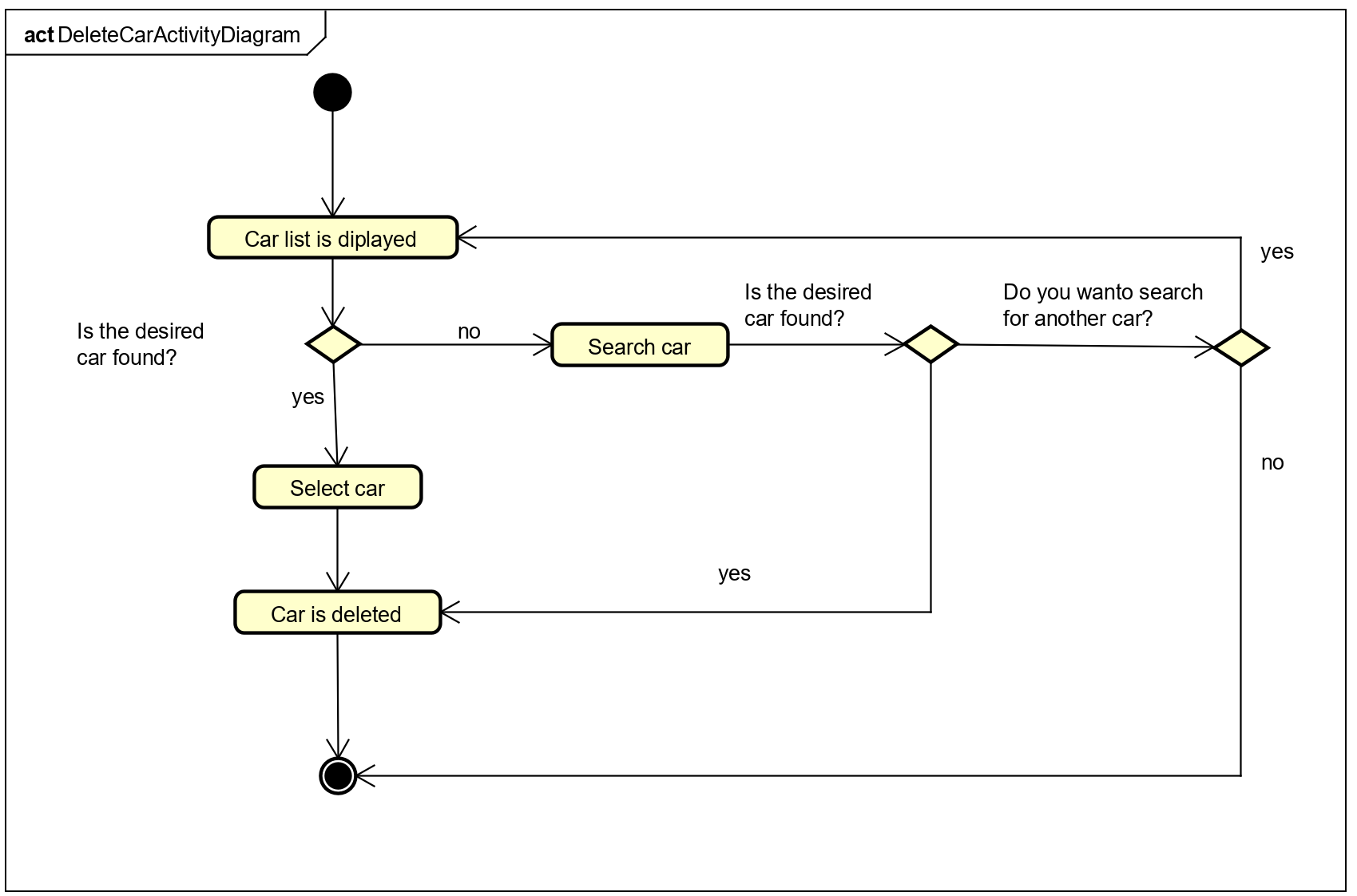


Figure 14- Activity diagram- Delete car

### 3.3.3 Edit car

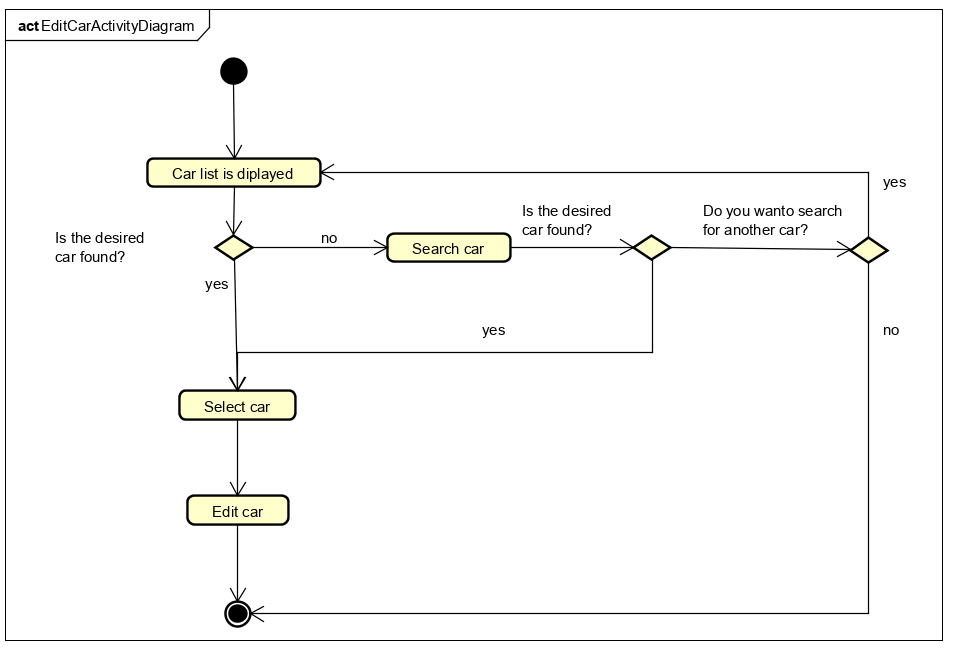


Figure 15- Activity diagram- Edit car

### 3.3.4 Manage requests

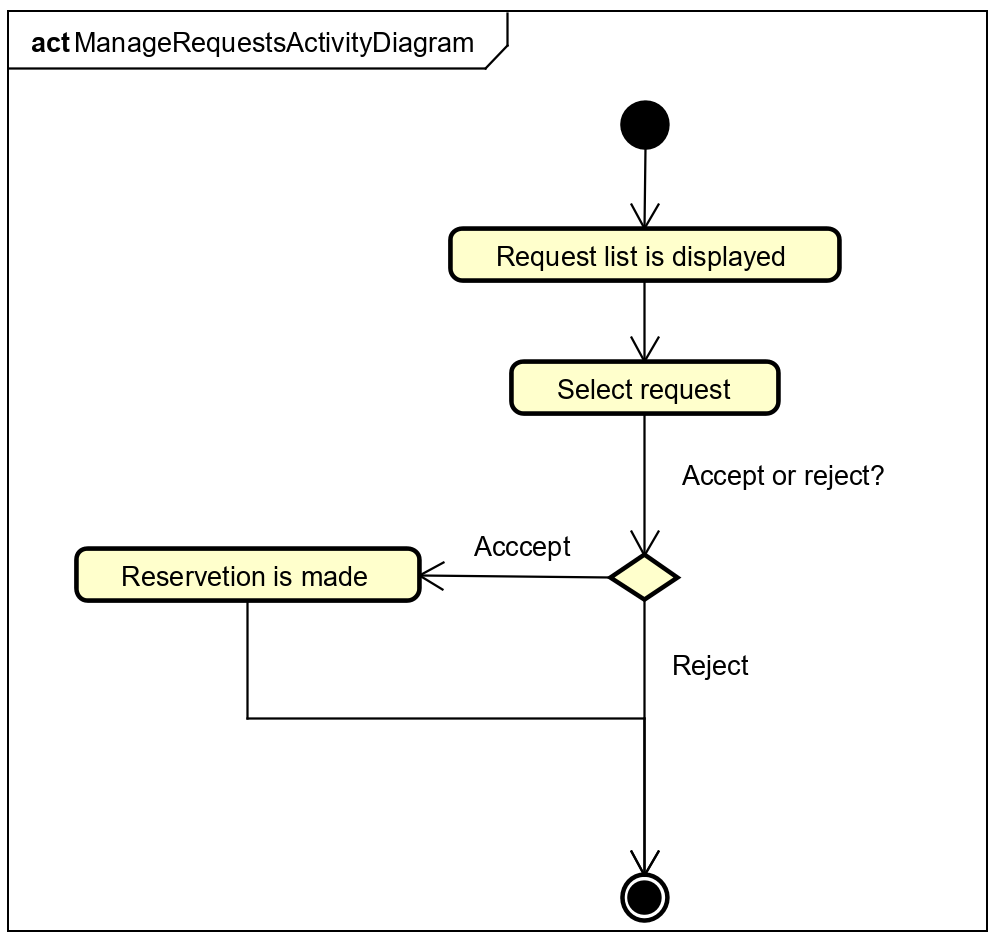


Figure 16- Activity diagram- Manage requests

### 3.3.5 Request

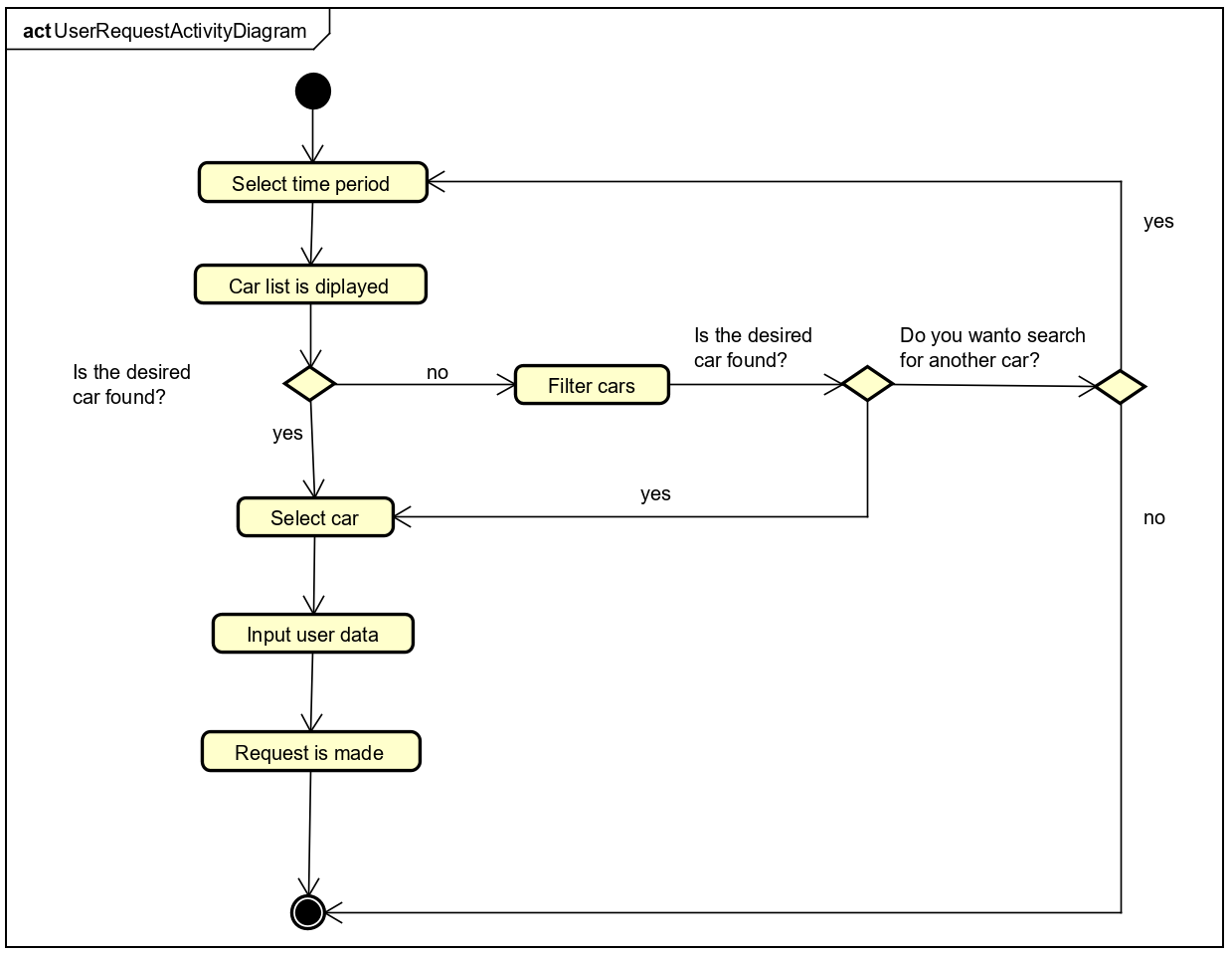


Figure 17- Activity diagram- Request

## 3.4 Domain model diagram

In software engineering, a domain model is a conceptual model of the domain that incorporates both behavior and data. In ontology engineering, a domain model is a formal representation of a knowledge domain with concepts, roles, datatypes, individuals, and rules, typically grounded in a description logic. A domain model generally uses the vocabulary of the domain, thus allowing a representation of the model to be communicated to non-technical stakeholders. It should not refer to any technical implementations such as databases or software components that are being designed.[3]

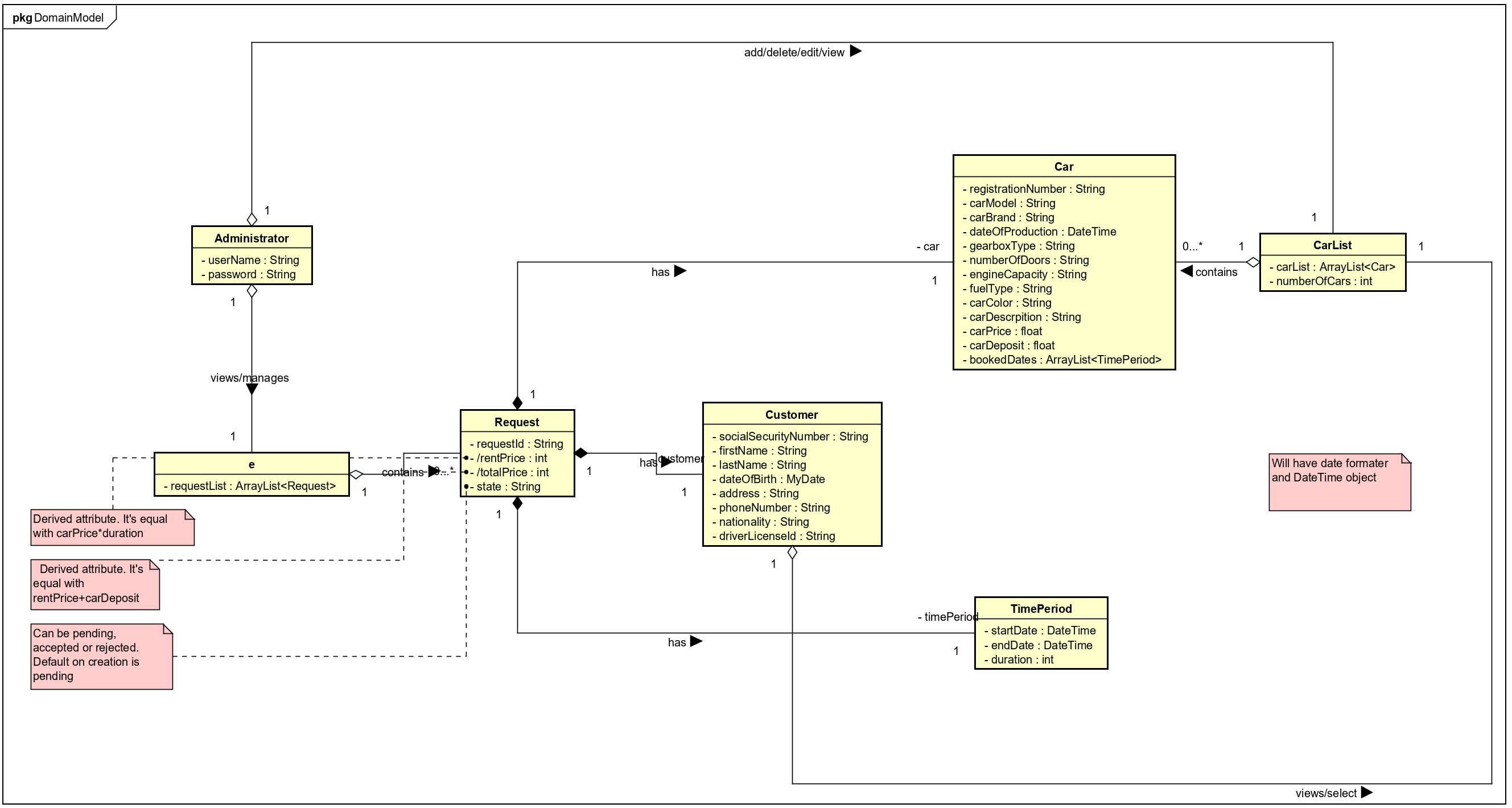


Figure 18- Domain model

## 3.5 System sequence diagram

Combining the logic behind de activity diagrams with the attributes from the domain model, five system sequence diagram were created to better explain the behavior of certain scenarios and the actor’s interaction with the system.

### 3.5.1 Add car

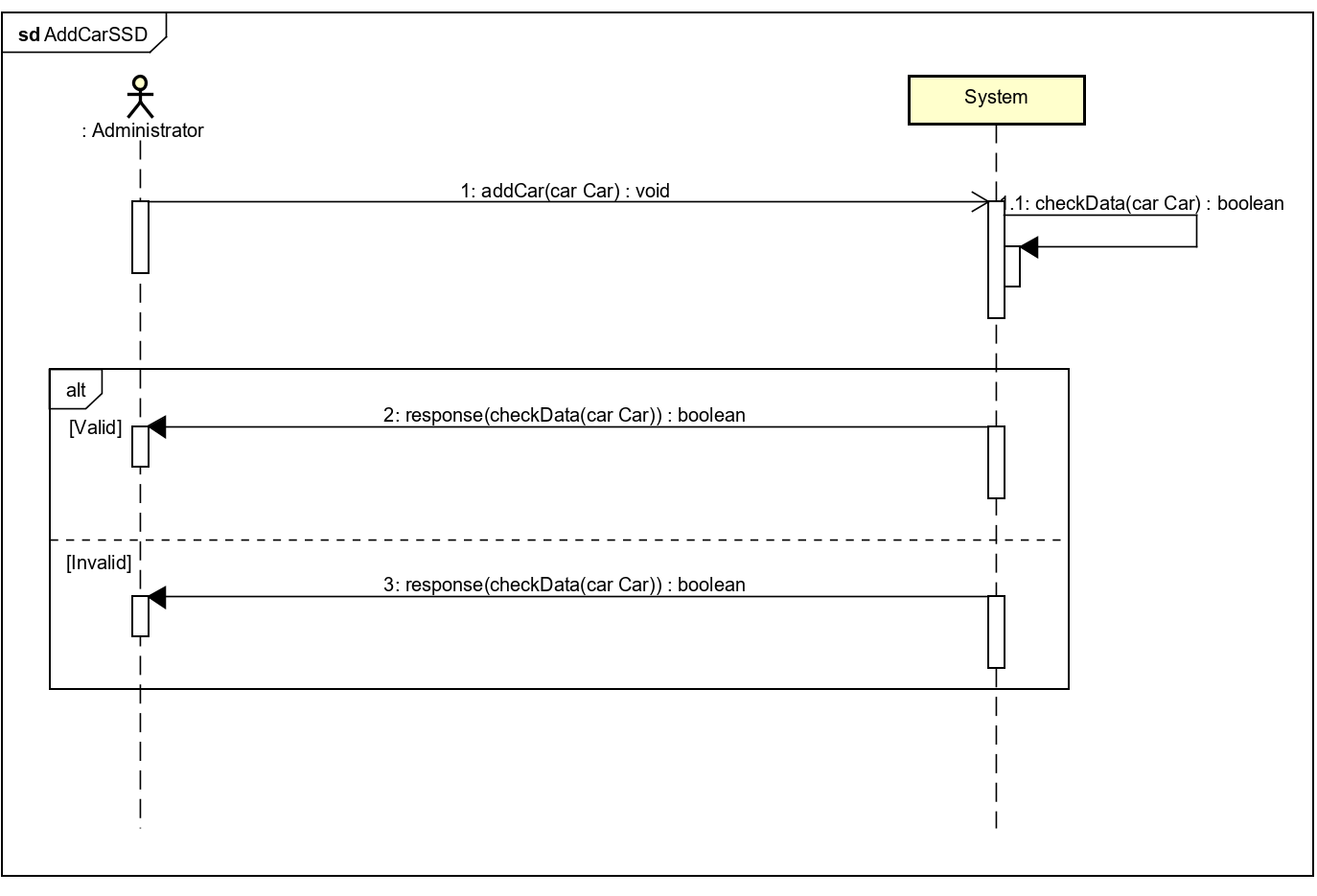


Figure 19- System sequence diagram- Add car

### 3.5.2 Delete car

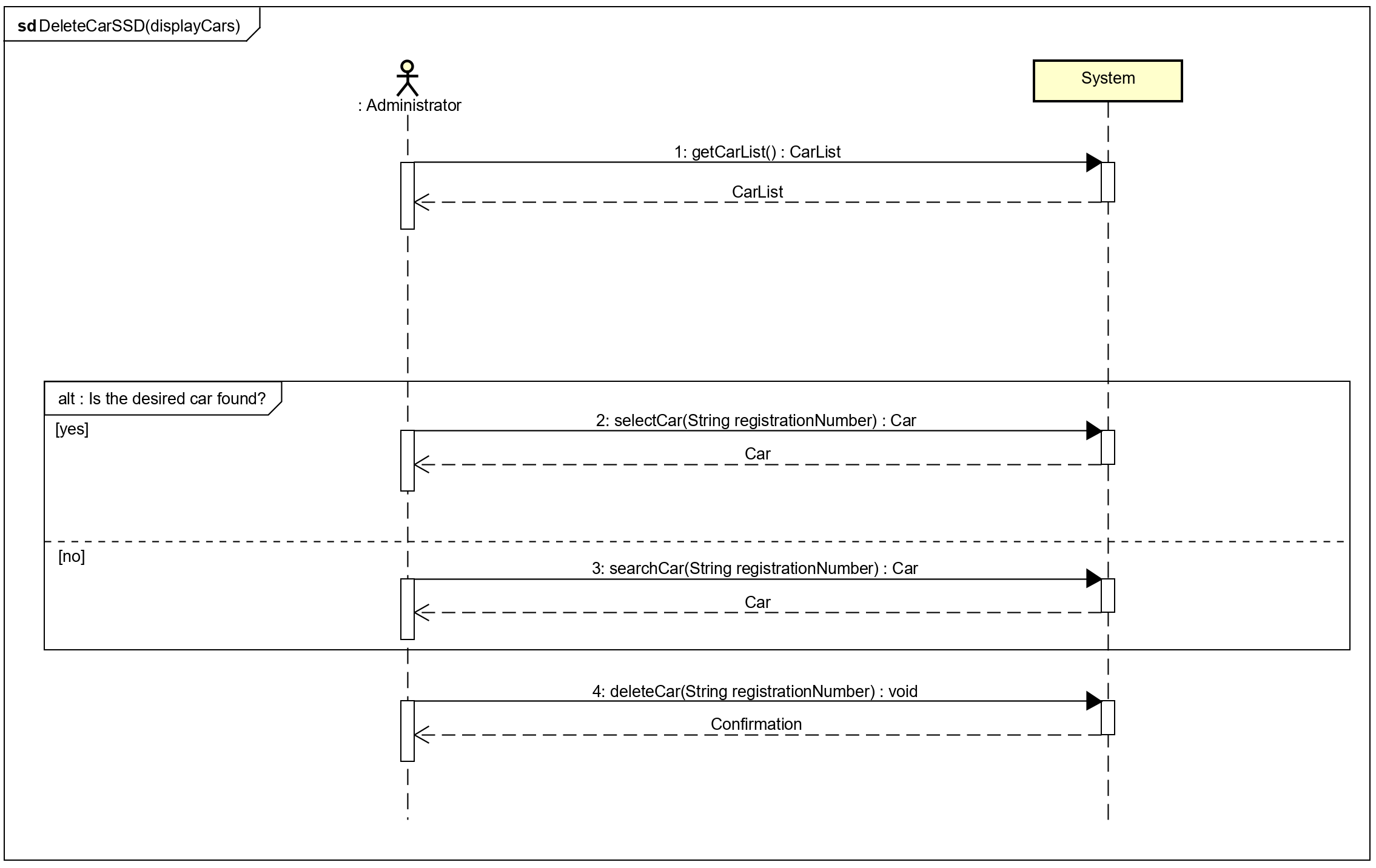


Figure 20- System sequence diagram- Delete car

### 3.5.3 Edit car

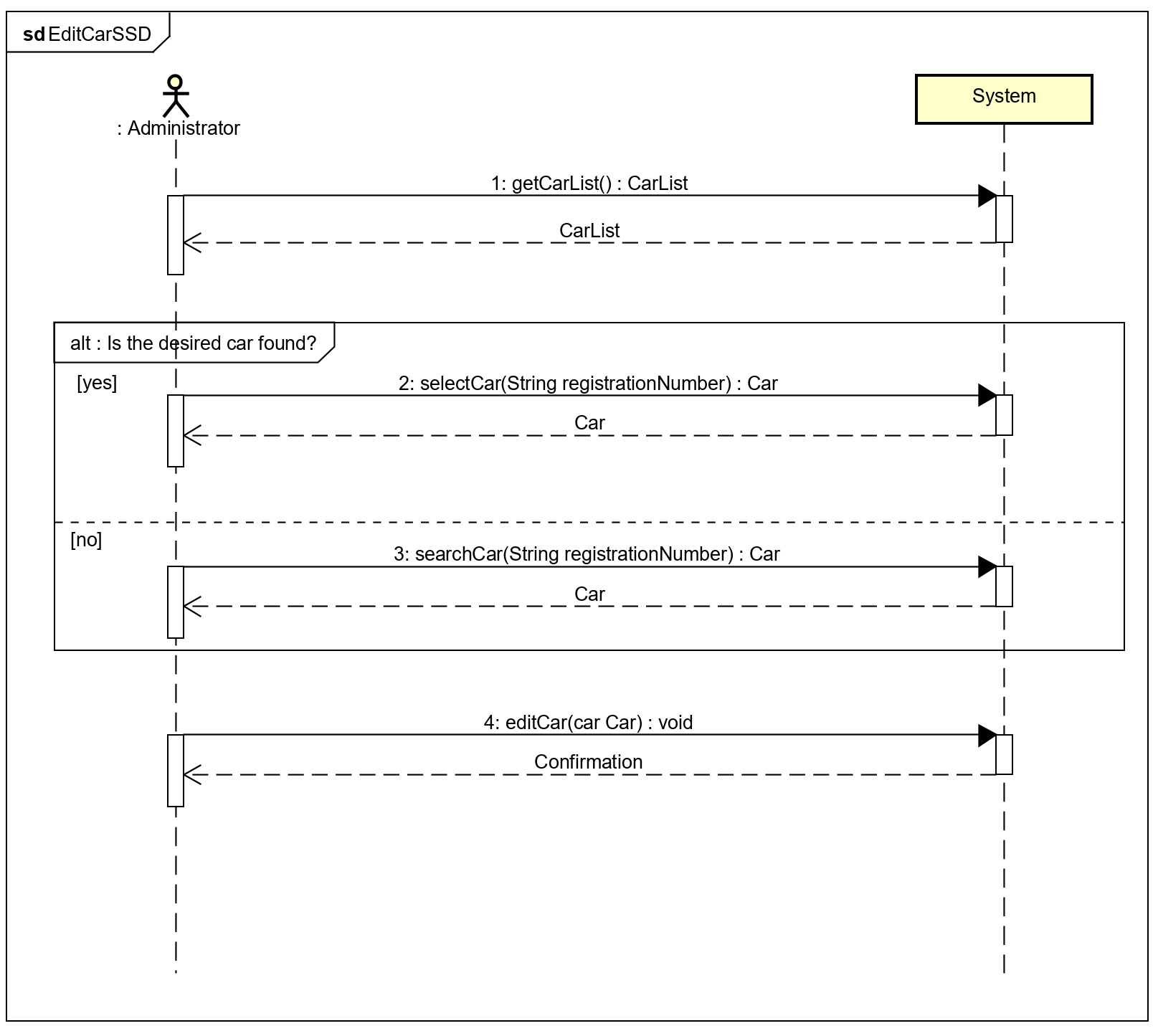


Figure 21- System sequence diagram- Edit car

### 3.5.4 Manage requests

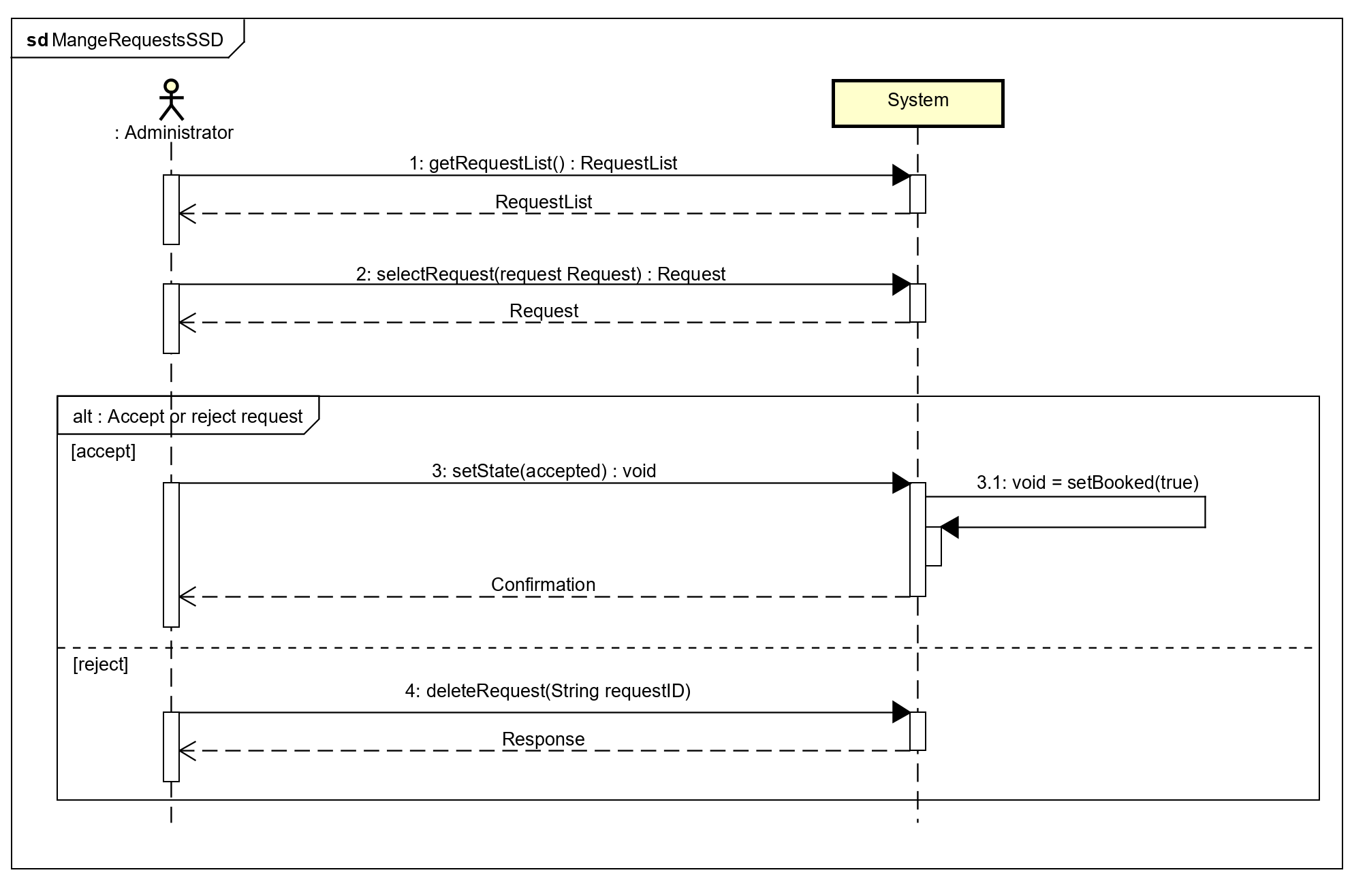


Figure 22- System sequence diagram- Manage requests

### 3.5.5 Request

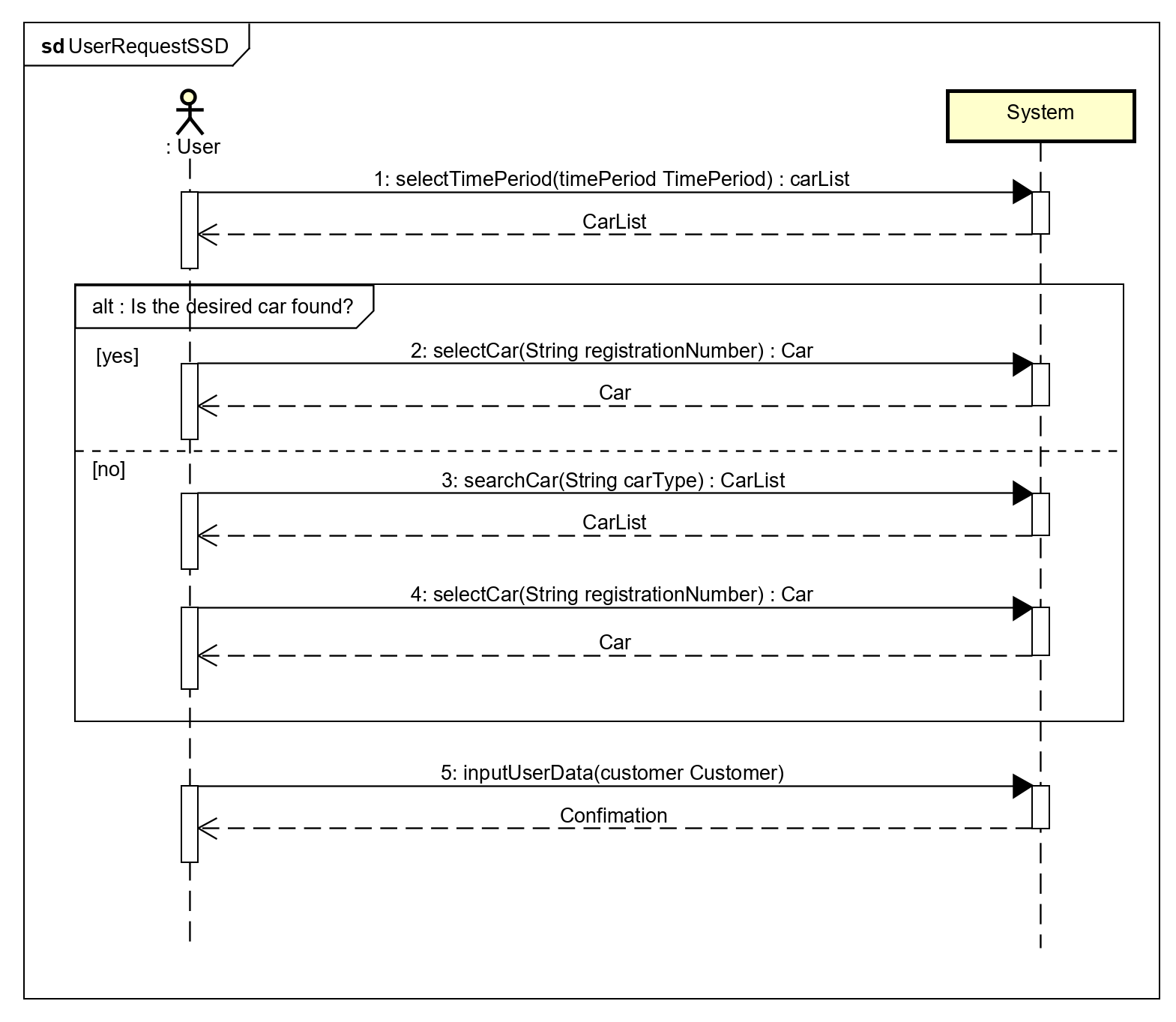
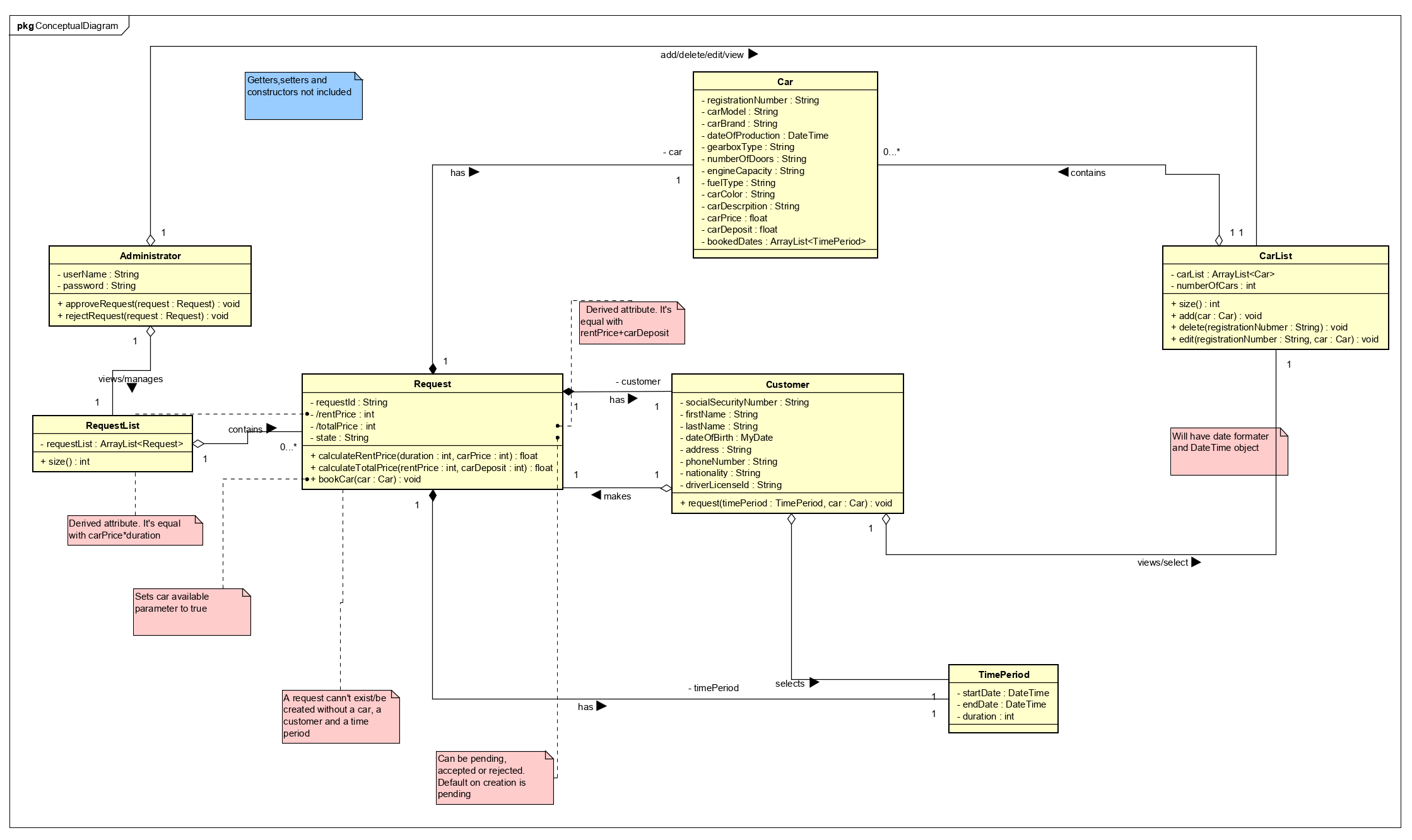


Figure 23- System sequence diagram- Request

# 4 Design

## 4.1 Conceptual diagram

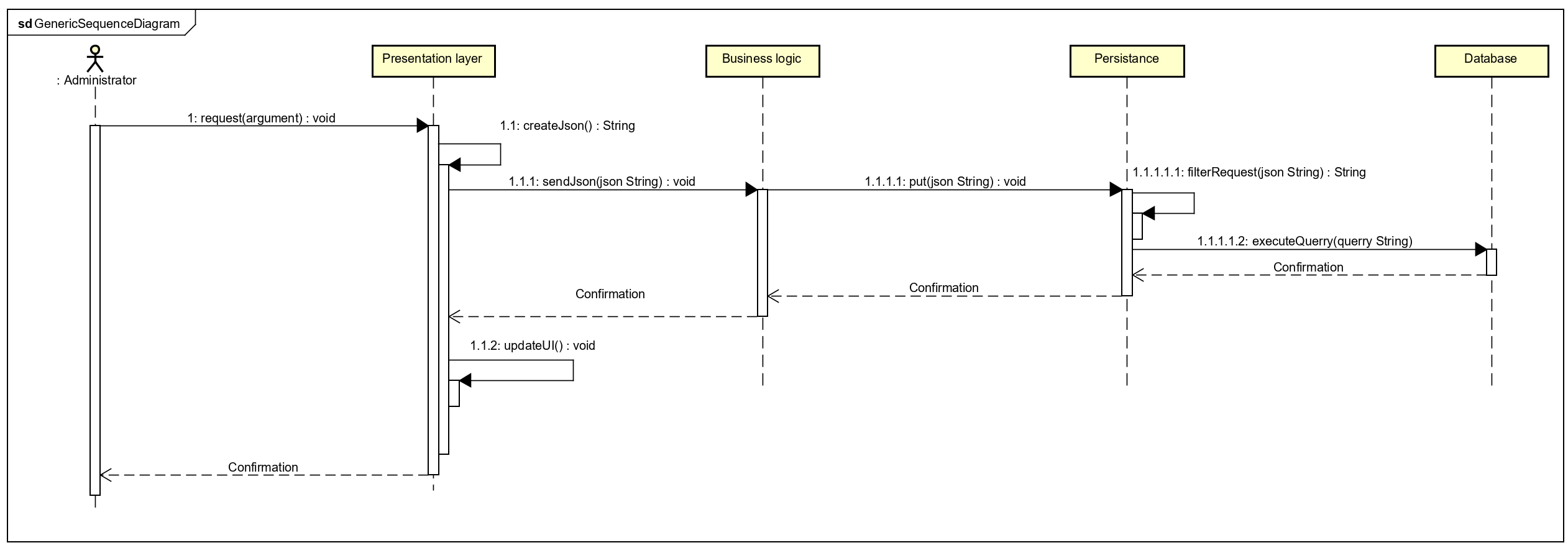
Concluding every answer found in the analysis phase, combining objects from the domain model with methods from the system sequence diagram, the conceptual diagram was formed.



Main point is that it was decided that the object would be connected in such a way that when a Customer will create a request, he will be compelled to previously select a Car and a Time period. So, a request object will not be able to exist without the other three previously mentioned objects as instances in the Request class itself. Administrator will have access to a list of Requests that he will be able to handle. This will also be the relation between the Administrator and the Car list.

## 4.2 Sequence diagram

A generic sequence diagram has been put together, hence the workflow for the requests will be similar.



Different requests can be made in the presentation layer. They will further be communicated in the system as Json objects. Json objects will be later opened so that can create/execute queries in relation to the initial command. User interface will be also updated accordingly to the result.

# 5 Implementation

***Figure 6.1.1 User Login Test Case***

***Pre-condition:*** *Make sure that the user account exists*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Reaction** | **Result** |
| 1 | Verify if a user can login entering a valid username and password | System checks to see if the information entered is of a valid type | System displays a confirmation message.  Access is granted |
| 2 | Verify if a user can login entering a valid username and invalid password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 3 | Verify if a user can login entering an invalid username and valid password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 4 | Verify if a user can login entering an invalid username and invalid password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 5 | Verify if a user can login entering a password not matching the required length | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 6 | Verify if a user can login entering a username and not entering a password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 7 | Verify if a user can login not entering a username but enters a password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 8 | Verify if a user can login by entering no information in the two fields | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 9 | User has internet access | System verifies if information is sent by the user | System displays a confirmation message.  Access is granted |
| 10 | User has no internet access | System verifies if information is sent by the user | System displays an error message.  Access denied. |

**Figure 6.1.2 Create Users Test Case**

***Pre-condition:*** *Make sure that the administrator account can create user accounts*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Reaction** | **Result** |
| 1 | Verify if a user can be created by entering a valid username and password | System checks to see if the information entered is of a valid type | System displays a confirmation message.  Account is created |
| 2 | Verify if a user can be created by entering a valid username and an invalid password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 3 | Verify if a user can be created by entering an invalid username and a valid password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 4 | Verify if a user can be created by entering an invalid username and an invalid password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 5 | Verify if a user can be created by entering a password not matching the required length | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 6 | Verify if a user can be created by entering a username and not entering a password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 7 | Verify if a user can be created by not entering a username but enters a password | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 8 | Verify if a user can be created by entering no information in the username field and password field | System checks to see if the information entered is of a valid type | System displays an error message.  Access denied. |
| 9 | Administrator has internet access | System verifies if information is sent by the administrator | System displays a confirmation message.  Access is granted |
| 10 | User has no internet access | System verifies if information is sent by the administrator | System displays an error message.  Access denied. |

***Figure 6.1.3 View Data Test Case***

**Pre-condition:** *Data must be stored an accessible for the give users*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Reaction** | **Result** |
| 1 | User must be logged in | System checks to see if the information entered is of a valid type | System displays a confirmation message.  Access is granted. |
| 2 | User must be connected to the internet | System checks user credentials | System validates user credentials.  Access is granted. |
| 3 | Establish a secured connection between the data source and the mobile app | System establishes a connection | System displays a confirmation message.  Access is granted. |
| 4 | Connection between data source and the mobile app is not established | System fails to establish a connection | System displays an error message.  Access is denied. |
| 5 | User selects which type of data he/she wants to view | System finds the data and returns it to the app | Data is displayed for the user. |
| 6 | User selects which type of data he/she wants to view | System cannot find the requested data | System displays an error message.  Data is not found |
| 7 | User selects which type of data he/she wants to view | System finds the data but cannot display it | System displays an error message. |
| 8 | User is be able to click and choose the selected data | System checks to see if the function is implemented | System displays the data |
| 9 | User is not able to click and choose the selected data |  |  |

**Figure 6.1.4 Remove User Test Case**

**Pre-Condition:***User account must be created and should be accessible .*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Reaction** | **Result** |
| 1 | User account is available | System checks to see if the account is available | System displays a confirmation message.  Access is granted. |
| 2 | User account is not available | System checks to see if the account is available | System displays an error message.  Access is denied. |
| 3 | Administrator can log in | System checks credentials | System displays a confirmation message.  Access is granted. |
| 4 | Administrator cannot log in | System fails to establish a connection | System displays an error message.  Access is denied. |
| 5 | Administrator can select the user account | System executes the specified task and removes account | System displays a confirmation message. |
| 6 | Administrator cannot select the user account |  |  |
| 7 | Administrator can select the user account | System fails to remote user account |  |
| 8 | Administrator cannot find the user account |  |  |

**Figure 6.1.5 Edit User Test Case**

**Pre-Condition:***User account must be created and should be accessible .*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Reaction** | **Result** |
| 1 | User account is available | System checks to see if the account is available | System displays a confirmation message.  Access is granted. |
| 2 | User account is not available | System checks to see if the account is available | System displays an error message.  Access is denied. |
| 3 | Administrator can log in | System checks credentials | System displays a confirmation message.  Access is granted. |
| 4 | Administrator cannot log in | System fails to establish a connection | System displays an error message.  Access is denied. |
| 5 | Administrator can select the user account | System executes the specified task and edits account | System displays a confirmation message. |
| 6 | Administrator cannot select the user account |  |  |
| 7 | Administrator can select the user account | System fails to save changes for the user account | System displays an error message. |
| 8 | Administrator cannot find the user account |  |  |

# 7 Conclusions

Reflecting on the requirements for the business case established as team work in the analyzing phase, together with the project requirements, it can be concluded that 60-70% of it has been completed.

In final phase the project is able to measure habitat conditions, prepare and upload the data on the LoRiot server from where the Java bridge was able to receive and decrypt the data.

The transmission to a MongoDB database has been successfully done so that the data to be sent further to an SQL Server where a DataWarehouse should have handled it.

Not having available data from the SQL server in due time and being less members than considered in the start, the Android team was able to manage presenting fake data to users who could have access to the data. Creating and managing users accounts in Firebase by an administrator has also been achieved.

Since the communication with the database team got worse with time, their achievements were not clearly identified for the rest of the team in due time.

From the and the ability to do business data processing and analyze for decisions takers (PowerBI).

From the decided business case point of view, the system does not fit the requirements, not being a mean for an end Android user to monitor and interact with real data from a closed environment. Also a technician actor not being able to take decisions and interfere with the system, taking actions in accordance with real data not been achieved. Also, the hardware system is not able to accept interactions and configurations in according to environment state, the thresholds first considered been neglected on the real system.

# 7 Table of figure

[Figure 1-Use case diagram 7](#_Toc16709954)

[Figure 2- Use case description- Choose period 7](#_Toc16709955)

[Figure 3- Use case description- Visualize cars 8](#_Toc16709956)

[Figure 4- Use case description- Request car 8](#_Toc16709957)

[Figure 5- Use case description- Filter cars 9](#_Toc16709958)

[Figure 6- Use case description- Add car 9](#_Toc16709959)

[Figure 7- Use case description- View cars 10](#_Toc16709960)

[Figure 8- Use case description- Delete car 10](#_Toc16709961)

[Figure 9- Use case description- Search car 11](#_Toc16709962)

[Figure 10- Use case description- Edit car 11](#_Toc16709963)

[Figure 11- Use case description- View requests 12](#_Toc16709964)

[Figure 12- Use case description- Manage Requests 12](#_Toc16709965)

[Figure 13- Activity diagram- Add car 13](#_Toc16709966)

[Figure 14- Activity diagram- Delete car 13](#_Toc16709967)

[Figure 15- Activity diagram- Edit car 14](#_Toc16709968)

[Figure 16- Activity diagram- Manage requests 14](#_Toc16709969)

[Figure 17- Activity diagram- Request 15](#_Toc16709970)

[Figure 18- Domain model 16](#_Toc16709971)

[Figure 19- System sequence diagram- Add car 17](#_Toc16709972)

[Figure 20- System sequence diagram- Delete car 18](#_Toc16709973)

[Figure 21- System sequence diagram- Edit car 19](#_Toc16709974)

[Figure 22- System sequence diagram- Manage requests 19](#_Toc16709975)

[Figure 23- System sequence diagram- Request 19](#_Toc16709976)

# 8 References

**[1]**Lazov, I. (2017). *Profit management of car rental companies*. [online] www.sciencedirect.com. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0377221716307032 [Accessed 14 Aug. 2019].

**[2]** (2019). *Car rental*. [online] Available at: https://en.wikipedia.org/wiki/Car\_rental [Accessed 14 Aug. 2019].

**[3]** En.wikipedia.org. (2019). *Car rental*. [online] Available at: https://en.wikipedia.org/wiki/Car\_rental [Accessed 14 Aug. 2019].

# 9 Appendices

## 9.1 Appendix 1 Project Description