Software Design Specification

for

Car Rental Application

Version 4.0 approved

Prepared by Efkan Efe Kabakcı, İbrahim Taha Pınar,

Mehmet Tahir Doğu, Yusuf Emre Yüzügüllü

Fırat University – Software Engineering Department

29.05.2025

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Initial Release | 22.05.2025 | Starting | 1.0 |
| Modification in System Architecture | 29.05.2025 | Business Logic part is extended and some filters are added. | 2.0 |
| Creating Appendix C | 29.05.2025 | Inputs and Outputs tables are created. | 2.0 |
| Updating Use Cases | 29.05.2025 | Use Case Part is updated. | 2.0 |
| Modification in ERD Diagram and Database Table Definitions | 03.06.2025 | Fixed errors in tables | 3.0 |
| Sequence Diagrams are Created | 20.06.2025 |  | 4.0 |

# Table of Contents

[Revision History 2](#_Toc199463546)

[Table of Contents 3](#_Toc199463547)

[1. Introduction 4](#_Toc199463548)

[1.1 Purpose of this Document 4](#_Toc199463549)

[1.2 Scope of the Development Project 4](#_Toc199463550)

[1.3 Definitions, acronyms, and abbreviations 4](#_Toc199463551)

[1.4 References 4](#_Toc199463552)

[1.5 Overview of Document 5](#_Toc199463553)

[2. System Architecture Description 5](#_Toc199463554)

[2.1 System Architecture 5](#_Toc199463555)

[2.1.1 High Level Architecture Diagram 5](#_Toc199463556)

[2.1.2 Architecture Narrative 5](#_Toc199463557)

[2.2 Database Components 6](#_Toc199463558)

[2.2.1 Entity Relationship Diagram (ERD) 6](#_Toc199463559)

[2.2.2 Database Table Definitions 6](#_Toc199463560)

[2.3 GUI Components with Layouts and Navigation 8](#_Toc199463561)

[2.3.1 User Interface Issues 8](#_Toc199463562)

[2.3.2 List of User Interface Screens and Reports 8](#_Toc199463563)

[2.3.3 User Interface Screens or Reports (Layouts) 9](#_Toc199463564)

[3. Detailed Description of System Components 10](#_Toc199463565)

[3.1 Class Diagram 10](#_Toc199463566)

[3.2 Use Cases 11](#_Toc199463567)

[Product Perspective 11](#_Toc199463568)

[User Use-Cases 11](#_Toc199463569)

[4. Design Decisions and Tradeoffs 15](#_Toc199463570)

[5. Appendices 16](#_Toc199463571)

[Appendix C: List of Inputs and Outputs (labeled and numbered) 16](#_Toc199463572)

[*Inputs* 16](#_Toc199463573)

[*Outputs* 17](#_Toc199463574)

# Introduction

## Purpose of this Document

This document aims to describe the overall design architecture, database schema, UI elements, and component interactions of the Car Rental Application project.

## Scope of the Development Project

Car Rental Application is a desktop application that allows remote management of car rental related operations. The main purpose of the app is to save time by saving people the hassle of going to a car rental company and automating of procedures related to vehicle rental.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Abbreviation** |
| SDS | Software Design Specification |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| CSS | Cascading Style Sheets |
| QSS | Qt Style Sheets |

## References

* IEEE Standard 1016-2009 for Software Design Descriptions
* Car Rental Application SRS Document v2.0
* Course Lecture (Week 10)

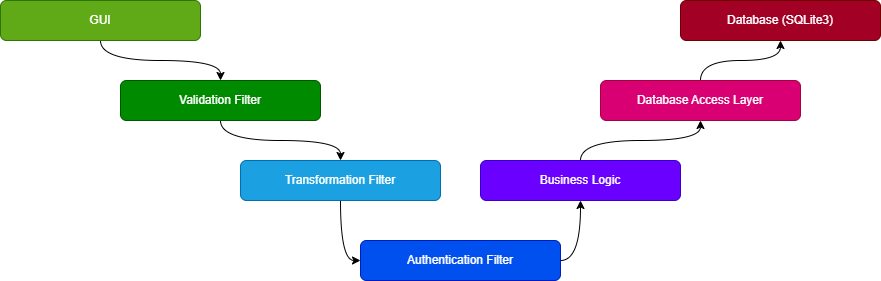
## Overview of Document

The rest of the document describes the system’s architecture, its components in detail, database structure, and all related UI interfaces and diagrams.

# 2. System Architecture Description

## 2.1 System Architecture

2.1.1 High Level Architecture Diagram



2.1.2 Architecture Narrative

The system follows a seven-layered architecture:

**GUI Layer**: Built using PyQt6, this layer provides a desktop graphical interface for normal users.

**Validation Filter**: Checks the format of inputs for correctness and missing data. Returns feedback to the GUI if there is faulty data.

**Transformation Filter**: Converts validated data into usable format. For example, case conversions of names, integer conversions of numbers received as strings from the UI.

**Authentication Filter**: Username/password check verification is performed. If unsuccessful, the data cannot pass to Business Logic.

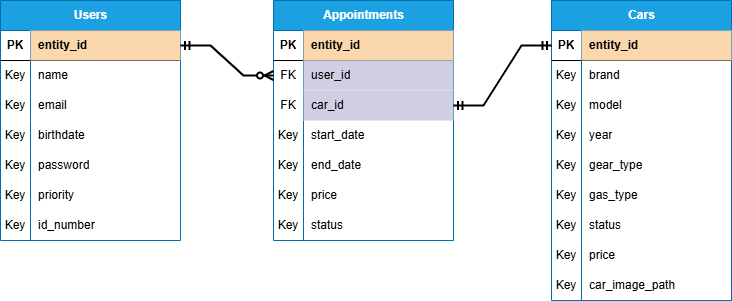
**Business Logic**: This is the layer where the main functions of the application (appointment booking, appointment cancellation, etc.) are performed.

**Database Access Layer**: Abstracts the database connection. Business Logic uses this layer to communicate the SQLite3 Server.

**Database (SQLite3)**: Permanent data storage. Appointments, users, cars are kept here.

## 2.2 Database Components

2.2.1 Entity Relationship Diagram (ERD)



2.2.2 Database Table Definitions

**User Table: Keeps the user data registered in the system.**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Constraints |
| entity\_id | VARCHAR | Unique ID of user | PRIMARY KEY |
| name | VARCHAR | First name | NOT NULL |
| email | VARCHAR | Email address | UNIQUE, NOT NULL |
| birthdate | VARCHAR | User’s birth date | NOT NULL |
| password | VARCHAR | Hashed password | NOT NULL |
| priority | INT | 0 for Admin and 1 for Normal User | NOT NULL |
| id\_number | INT | Identity Number of User | NOT NULL |

**Car Table: Keeps the car data available in the system.**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Constraints |
| entity\_id | VARCHAR | Unique ID of car | PRIMARY KEY |
| brand | VARCHAR | Brand of car | NOT NULL |
| model | VARCHAR | Model of car | NOT NULL |
| year | INT | How old the car is | NOT NULL |
| gear\_type | VARCHAR | Gear type of car (auto - manual) | NOT NULL |
| gas\_type | VARCHAR | Gas type of car (diesel - gasoline) | NOT NULL |
| status | BOOLEAN | Rentability status | NOT NULL |
| price | FLOAT | Daily price of the car | NOT NULL |
| car\_image\_path | VARCHAR | Relative path of the car image | NOT NULL |

**Appointments Table: Keeps rented vehicle data.**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Constraints |
| entity\_id | VARCHAR | Unique ID of appointment | PRIMARY KEY |
| user\_id | VARCHAR | ID of the renter | FOREIGN KEY → User |
| car\_id | VARCHAR | ID of the rented car | FOREIGN KEY → Car |
| start\_date | VARCHAR | The day the rental starts | NOT NULL |
| end\_date | VARCHAR | The day the rental ends | NOT NULL |
| price | FLOAT | Total price | NOT NULL |
| status | BOOLEAN | Appointment status | NOT NULL |

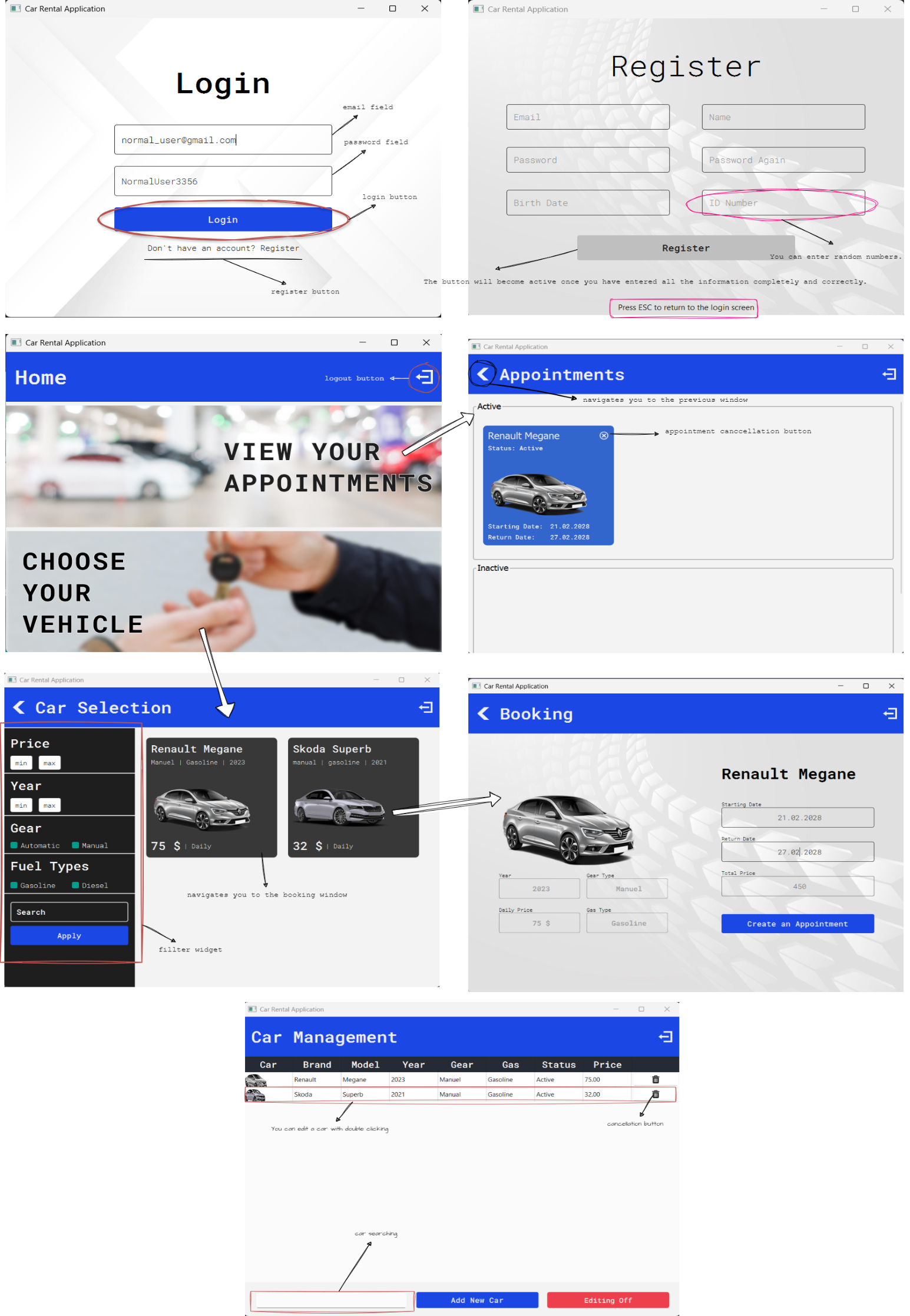
## 2.3 GUI Components with Layouts and Navigation

2.3.1 User Interface Issues

The application is designed for normal users with basic computer skills. Therefore, the user interface must be intuitive, visually simple, and accessible without training. Error messages must be displayed clearly under each form field to prevent confusion.

2.3.2 List of User Interface Screens and Reports

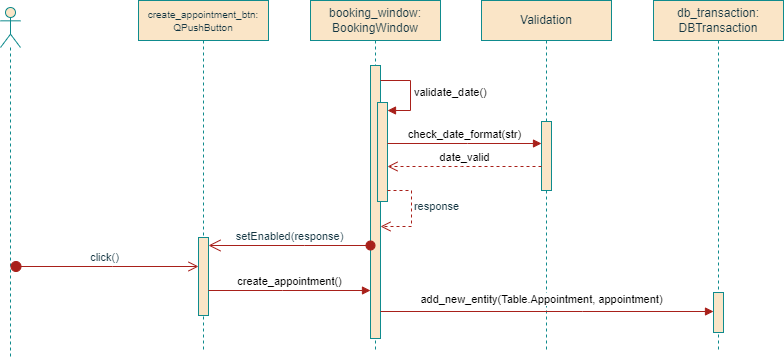
* ***Login Screen***: Allows users to enter their email and password. Contains a "Don’t have an account? Register" link, and a "Forgot Password?" option.
* ***Registration Screen***: Enables users to create a new account. Fields include full name, email, password, phone number, and license document.
* ***Home Screen***: Presents the user "Rent a Car" and "View Rented Cars" buttons.
* ***Cars Screen***: Displays available vehicles with filtering options (brand, model, price, fuel type, gear type).
* ***Booking Screen***: Allows the user to and enter reservation dates approve its selection vehicle. A summary widget shows total cost.

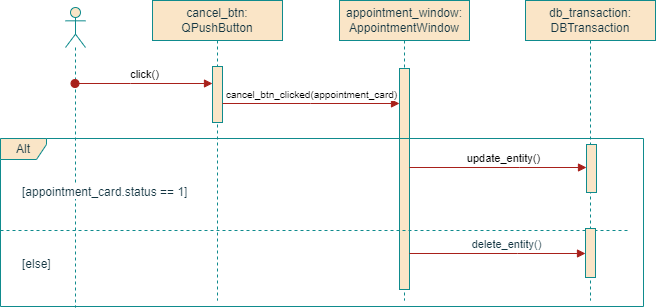


# 3. Detailed Description of System Components

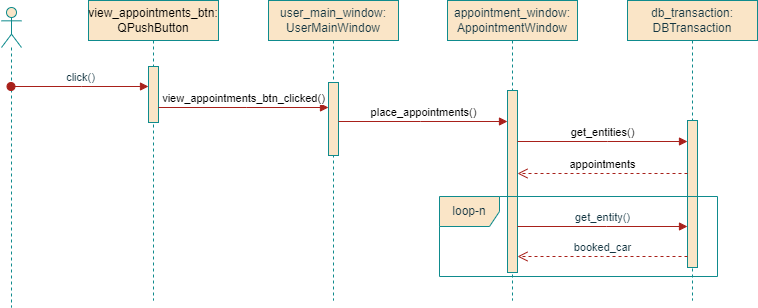
## 3.1 Class Diagram

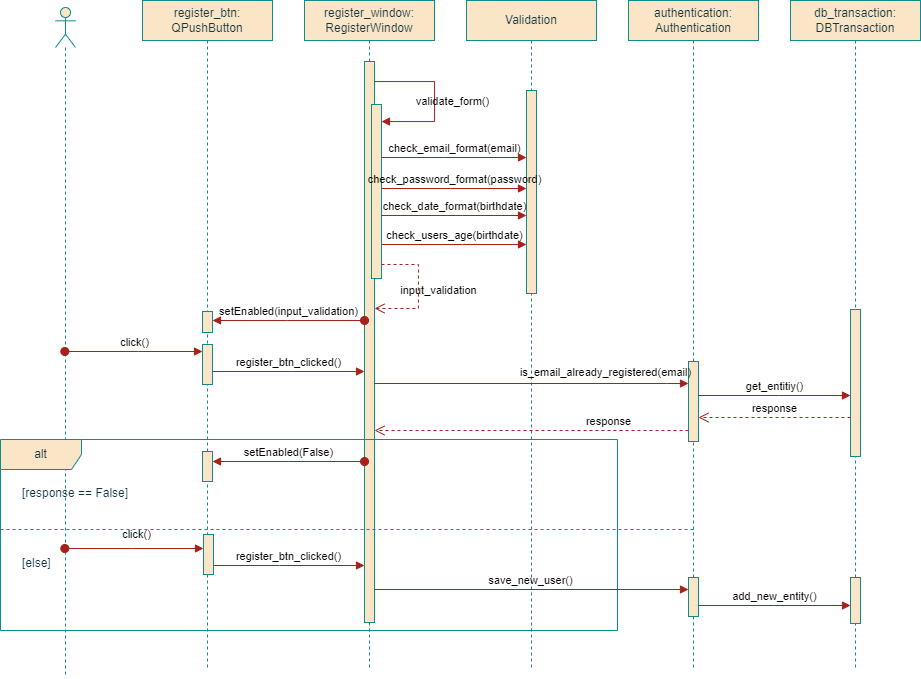
## 3.2 Sequence Diagrams

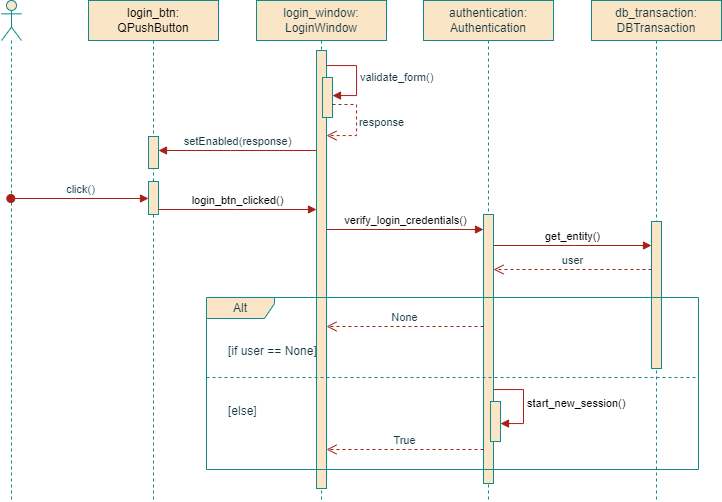
**Create Car Rental Appointment**

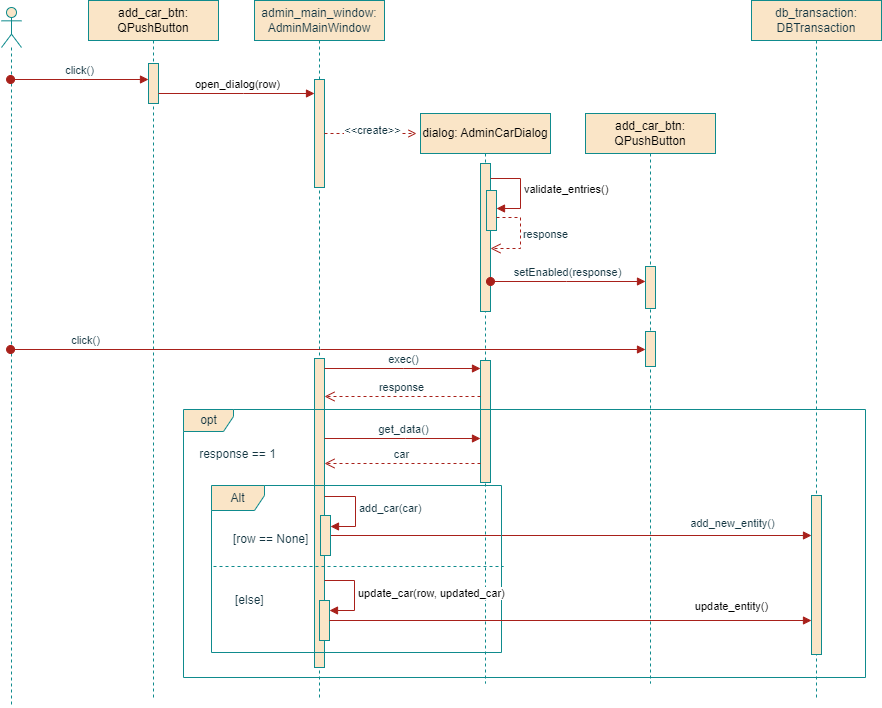
**Cancel an Appointment**

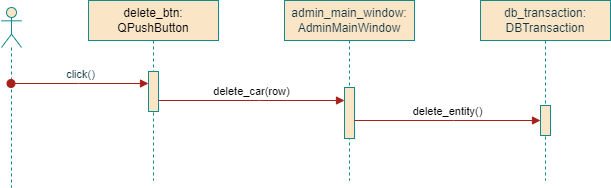
**View Your Appointments**

****

**Register**

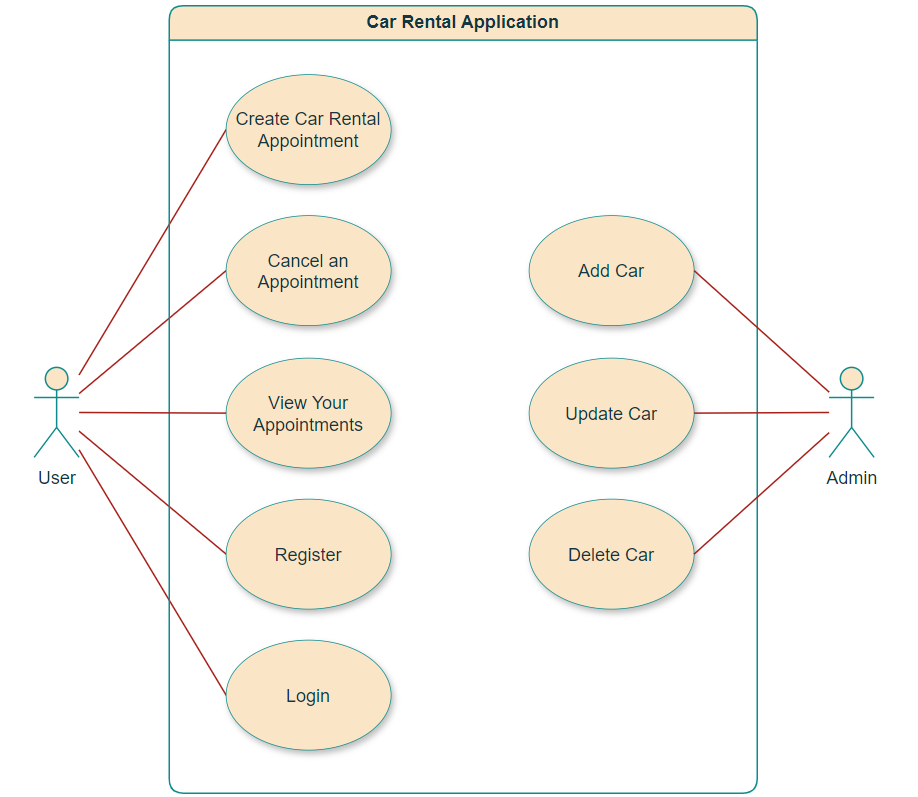
**Login**

**Add Car – Update Car**

**Delete Car**

## 3.3 Use Cases

## Product Perspective



### **User Use-Cases**

Create Car Rental Appointment Use Case

**Pre-Conditions**

* The user must be logged in to the system.

**Main Scenario**

1. The user clicks the “Select a Car” button.
2. The system navigates the user to the “Cars” screen.
3. The user selects a car from the list.
4. The system navigates the user to the “Booking” screen.
5. The user enters the start and end day for the appointment.
6. The system checks the validity of the dates.
7. The user clicks the “Create Appointment” button.
8. The system displays a confirmation dialog to the user.
9. The user confirms the appointment.
10. The system saves the appointment in the database.

**Exceptions**

* 6a. The dates are invalid.
  + 6a.1. The “Create Appointment” button is disabled.
  + 6a.2. The text field enters error state.

View Your Appointments Use Case

**Pre-Conditions**

* The user must be logged in to the system.

**Main Scenario**

1. The user clicks the “View Your Appointments” button.
2. The system navigates the user to the “Appointments” screen.
3. The system fetches rental appointments that match the user's ID.
4. The system displays the list of appointments created in the past or currently active.

Cancel an Appointment

**Pre-Conditions**

* The user must be logged in to the system.
* The user must have at least one active appointment.

**Main Scenario**

1. [Include: View Your Appointments]
2. The user selects the appointment they want to cancel from the list.
3. The user clicks on the “Confirm” button.
4. The system asks the user a confirmation question.
5. The user confirms the cancellation.
6. The system updates the appointment status as “0”.

Register Use Case

**Pre-Conditions**

* The user must be of legal age.

**Main Scenario**

1. The user clicks on the “Don’t have an account? Register” button.
2. The user fills in all the information completely.
3. Each information is validated by the system.
4. The user clicks on the “Register” button.
5. The system registers the new user in the database.
6. The system shows a success dialog.
7. The system redirects the user to the home page.

**Exceptions**

* 3a. Any information may be in an incorrect format or incomplete.
  + 3a.1. The “Register” button is disabled.
  + 3a.2. The text field enters error state.

Login Use Case

Pre-Conditions

* The user must have a registered account.

Main Scenario

1. User enters email and password.
2. System forwards credentials to Auth System.
3. Auth System verifies the credentials.
4. If valid, user is authenticated.
5. System redirects to main dashboard.

**Exceptions**

* 3a. Incorrect credentials.
  + 3a.1. The system shows an error message.

### **Admin Use Cases**

Add Car Use

**Pre-Conditions**

* The admin must be logged in to the system.

**Main Scenario**

1. The admin clicks the “Add New Car“ button.
2. The system shows the “AdminCarDialog“ window.
3. The admin enters the information about the car to be added.
4. Inputs are validated by the system.
5. The admin clicks the “Confirm” button.
6. The system adds a new car to the database.

**Exceptions**

* 4a. Any information may be in an incorrect format or incomplete.
  + 3a.1. The “Confirm” button is disabled.

Update Car Use

**Pre-Conditions**

* The admin must be logged in to the system.

**Main Scenario**

1. The admin clicks a row from the table.
2. The system shows the “AdminCarDialog“ window.
3. The admin enters new information about the car.
4. Inputs are validated by the system.
5. The admin clicks the “Confirm” button.
6. The system updates the car.

**Exceptions**

* 4a. Any information may be in an incorrect format or incomplete.
  + 3a.1. The “Confirm” button is disabled.

Delete Car Use

**Pre-Conditions**

* The admin must be logged in to the system.

**Main Scenario**

1. The admin clicks on the trash can icon in the row containing the car they want to delete.
2. The system shows a confirmation dialog.
3. If the admin confirms, the system deletes the car from the database.

# 5. Appendices

## Appendix C: List of Inputs and Outputs (labeled and numbered)

## *Inputs*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Input Name | Type | Description | *Source* |
| I1 | Email | String | The e-mail used by the user when logging or registering. | *Login Screen,*  *Register Screen* |
| I2 | Password | String | The password used by the user when logging or registering. | *Login Screen,*  *Register Screen* |
| I3 | Name | String | The name | *Register Screen* |
| I4 | Surname | String | The surname | *Register Screen* |
| I5 | Birth Date | Datetime | The birth date | *Register Screen* |
| I6 | ID Number | Integer | The user identification number | *Register Screen* |
| I7 | Driver License | Image | The user’s driver license uploaded by the user | *Register Screen* |
| I8 | Price | Float | Maximum or minimum price for filtering cars | *Cars Screen* |
| I9 | Year | Datetime | Maximum or minimum year for filtering cars | *Cars Screen* |
| I10 | Keyword | String | A special word for filtering cars | *Cars Screen* |
| I11 | Starting Date | Datetime | The day the user will rent the car | *Booking Screen* |
| I12 | Return Date | Datetime | The day the user will return the car | *Booking Screen* |

## *Outputs*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Output Name | Type | Description | Destination |
| O1 | Login Result | Boolean | Indicates whether the login was successful or not. | GUI |
| O2 | Registration Confirmation | Message | Confirmation message after successful registration, with email sent. | Register Screen, Email System |
| O3 | Appointment Confirmation | Message | Displays confirmation of the created car rental appointment. | Booking Screen + Email System |
| O4 | Appointment PDF | File (PDF) | A downloadable PDF document containing appointment details. | System Default PDF Viewer |
| O5 | Cancellation Status | Boolean | Result of the appointment cancellation request. | Appointments Screen + Email System |
| O6 | Error Message | String | Message shown when a user or system error occurs. | GUI |