System Requirement Documentation

INFO-C450

Joshua Willis

2025

Table of Contents

[Customer Problem Statements and System Requirements 2](#_Toc197064667)

[Functional and Nonfunctional Requirements 2](#_Toc197064668)

[System Sequence Diagram 1 4](#_Toc197064669)

[System Sequence Diagram 2 5](#_Toc197064670)

[Activity Diagram 6](#_Toc197064671)

[User Interface Specification 7](#_Toc197064672)

[Traceability Matrix 9](#_Toc197064673)

[System Architecture and System Design 10](#_Toc197064674)

[User Interface Design 10](#_Toc197064675)

[Project Plan 10](#_Toc197064676)

# Customer Problem Statements and System Requirements

***Problem Statement:***

Customers of car rental services often face difficulties when attempting to book a rental based around the in-store experience. By the time the customer steps into the store the selection of available vehicles may be limited. The rental provider may lack transparency in their pricing model. The time spent in the store can be drawn out and arduous. An online system that allows customers to browse available vehicles and lock in their selection and price ahead of time could alleviate these difficulties.

***Glossary of Terms:***

Customer: User of the system

Car: Vehicle that can be selected for rental

Rental Length: Number of days the rental is being booked for

# Functional and Nonfunctional Requirements

***Functional Requirements:***

|  |  |  |
| --- | --- | --- |
| **No.** | **Priority Weight** | **Description** |
| Availability | High | Check to see what cars are available for rental in real-time |
| New Customer | High | Ability to create new customer registration in system |
| Price | Medium | Ability to see/set prices based on vehicle |

***Nonfunctional Requirements:***

|  |  |  |
| --- | --- | --- |
| **No.** | **Priority Weight** | **Description** |
| Downtime | High | Keep downtime to a minimum and system availability high |
| Scalability | Medium | Make system scalable to meet customer demand and car inventory |
| Security | High | Keep customer data safe and secure |

***User Interface Requirements:***

|  |  |  |
| --- | --- | --- |
| **No.** | **Priority Weight** | **Description** |
| Create Customer | High | Button to add Customer registration info to database |
| Check Availability | Medium | Button to see cars available for rental |
| Checkout | High | Button to calculate cost with length of rental and secure booking |

# System Sequence Diagram 1

***A diagram of a car rental system

AI-generated content may be incorrect.***

# System Sequence Diagram 2

***A diagram of a car rental system

AI-generated content may be incorrect.***

# Activity Diagram

***A diagram of a customer service

AI-generated content may be incorrect.***

# User Interface Specification

Use Case - Create New Customer

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Click Create Customer

Enter Phone #

Enter License #

Enter Name

Use Case - Select Customer

Use Case - Display Available Cars

Click Display Customers

Enter Customer ID

Select Customer

Click Display Available Cars

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a car rental window

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Use Case - Display Daily Cost

Use Case - Checkout

Click Display Available Cars

Enter Car ID #

Click Display Cost

Set Rental Length

Enter Car ID #

Click Checkout

After Display Daily Cost

# Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Type | Code Element | Test Case |
| FRQ1 | System will display a welcome screen with a login option to access the main interface | Functional | WelcomeGUI class, actionPerformed for login button | Verify that clicking the "Login" button on the welcome screen opens CarRentalGUI and closes WelcomeGUI |
| FRQ2 | System will allow users to view a list of available cars from the database | Functional | CarRentalGUI.displayAvailableCars, DatabaseManager.getAvailableCars | Verify that clicking jButton1 displays all cars with Available = true in jTextArea1. Check that list matches the database |
| FRQ3 | System will allow users to add a customer with name, license, and phone to the database | Functional | CarRentalGUI.jButton5ActionPerformed, DatabaseManager.addCustomer | Enter customer data, click jButton5, verify success message with the correct CustomerID, and check the Customers table for the new record |
| FRQ4 | System will allow users to rent a car by specifying a car ID and rental days, marking the car as unavailable | Functional | CarRentalGUI.CheckoutButtonActionPerformed, RentalSystem.rentCar, DatabaseManager.rentCar | Select a customer, enter a valid CarID and days, click "Checkout", verify success message, check Rentals table for new record, and confirm car’s IsAvailable = false in Cars table |
| NFRQ1 | System will provide a user-friendly interface with clear feedback for actions | Non-Functional | WelcomeGUI, CarRentalGUI, AdminGUI (use of JOptionPane for feedback) | Perform actions (add customer, rent car, etc.), verify clear success/error messages via dialogs. Check that fields are intuitive and labeled |
| NFRQ2 | The system shall store data persistently in a Microsoft Access database | Non-Functional | DatabaseManager (all methods) | Add customers and cars, close the application, reopen, and verify data persists in CarRental.accdb |

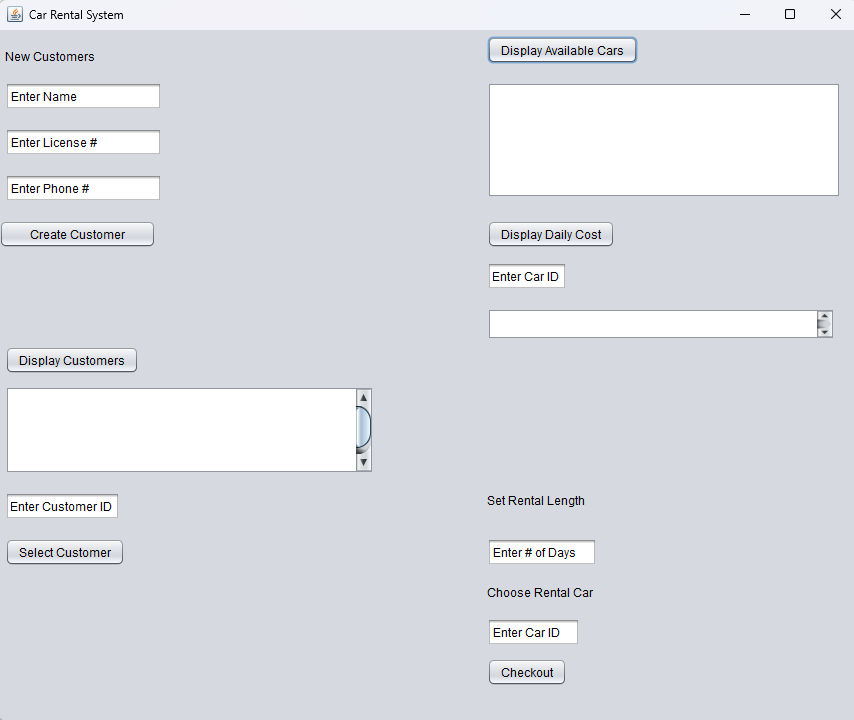
# System Architecture and System Design

The Car Rental System is a client-side desktop application developed on JAVA IDE Netbeans and connected to a local Microsoft Access database through JDBC and the drive UCanAccess for data storage. The system employs a layered architecture that separates core functions into a presentation, business logic, and data access layer. This makes the system easier to maintain and scale for future needs.

# User Interface Design

A screenshot of a computer

AI-generated content may be incorrect.



# Project Plan

Development Plan

W1 – 2: determine the framework and establish the structure of the system *Complete*

W3 – 4: build the customer interface *Complete*

W5 – 7: build the provider interface *Complete*

W8: test features, record demo for mid-term *Complete*

W9 – 11: improve features based on feedback *Complete*

W12 – 14: writing test cases for implemented features and bug testing *Complete*

W15: record demo for final presentation *In Progress*