**SRS-2023**

SOFTWARE REQUIREMENT SPECIFICATION

Car Rental Application

client:

Muhammad Syafiq Aqilla

Prepared by:

Group 02

Faculty of Informatics

Telkom University

2023

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Informatika  -  Fakultas Informatika | Document Number | | Number of Pages |
| *SRS-2023* | | 20 |
| Revisi | *<nomor revisi>* | *Tgl: 8 November 2023* |

# Revision List

|  |  |
| --- | --- |
| Revision | Descriptioon |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| Date |  |  |  |  |  |  |  |  |
| Created by |  |  |  |  |  |  |  |  |
| Reviewed by |  |  |  |  |  |  |  |  |
| Approved by |  |  |  |  |  |  |  |  |

# List of Revision Pages

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Revision | Page | Revision |
|  |  |  |  |

# Table of Contents

[Revision List 1](#_Toc1804010422)

[List of Revision Pages 2](#_Toc1962683545)

[Table of Contents 3](#_Toc1223177313)

[List of Figures 4](#_Toc1514500219)

[List of Tables 5](#_Toc542683556)

[1. Introduction 6](#_Toc2089340892)

[1.1 Purpose of Writing the Document 6](#_Toc125959184)

[1.2 Scope of Problem 6](#_Toc168689017)

[1.3 Definitions and Terms 6](#_Toc1877339423)

[1.4 References 6](#_Toc1895424953)

[2. General Description of the Software 7](#_Toc231855451)

[2.1 Statement of Software Objective 7](#_Toc6083948)

[2.2 Software Benefits / Functions 7](#_Toc1149280639)

[2.3 User Characteristics 7](#_Toc160020051)

[2.4 Software / System Limitations 8](#_Toc55422488)

[3. Detailed Description of Software 9](#_Toc1107324079)

[3.1 Requirement Descriptions 9](#_Toc605820360)

[3.1.1 Functional Requirements 9](#_Toc581133036)

[3.1.2 Non-Functional Requirements 10](#_Toc795316821)

[3.2 Analysis Modeling 11](#_Toc1006380122)

[3.2.1 Usecase Diagram 11](#_Toc1608788460)

[3.2.1.1 Usecase Scenario #1 < use case Check Car Condition> 11](#_Toc684312152)

[3.2.1.2 Usecase Scenario #2 < use case Rent Car> 12](#_Toc22674920)

[3.2.1.3 Use case Scenario #3 < use case Return car> 12](#_Toc2116918364)

[3.2.1.4 Use case Scenario #4 < use case Payment> 13](#_Toc2007572661)

[3.2.1.5 Use case Scenario #5 < use case Provide Car> 13](#_Toc861878043)

[3.2.2 Class Diagram 14](#_Toc1898701257)

[3.2.2.1 Class Diagram Description 15](#_Toc1810396440)

[3.3 System Environment Requirements 16](#_Toc732182124)

[3.3.1 Operating Environment 16](#_Toc1275533976)

[3.3.2 Hardware Requirements 16](#_Toc298183694)

[3.3.3 Software Dependencies 17](#_Toc703033364)

[4. Appendix 18](#_Toc972448111)

# List of Figures

1. Page 12 Usecase Diagram
2. Page 15 Class Diagram

# List of Tables

1. Page 10 about Functional Requirements
2. Page 11 about Non-Functional Requirements
3. Page 16 Class Diagram Description

# Introduction

## Purpose of Writing the Document

This SRS document is written to detail important information, provide a detailed overview of our software product, its parameters and goals and steps that will be used to help our group in completing our task for the Software Engineering class of the 3rd semester Informatics major at Telkom University.

## Scope of Problem

The Scope problem statement describes the current situation, the requirements (functional and nonfunctional requirements) and the target environment in which the system will be deployed. This section should also outline any constraints, assumptions, and dependencies that may affect the software’s development.

## Definitions and Terms

Software Requirement Specifications (SRS) are documents which explain the various needs that must be met by software. This document was created by the developer (software maker) after digging up information from potential software users. The manufacturer should also follow existing standards and are most recognized by software engineering practitioners in the world.Requirement specifications (requirements) are the attributes required in system, a statement that identifies a capability, characteristic, or quality factors of a system with the aim of obtaining value and utility in customers or users

## References

* <https://www.adesso.de/adesso/adesso-de/news/blog/blog-22-09-16-problem-statement-adesso.pdf>
* <https://www.geeksforgeeks.org/software-requirement-specification-srs-document-checklist/>
* <https://repository.uksw.edu/bitstream/123456789/17830/2/T1_682014075_Full%20text.pdf>
* <https://www.utdallas.edu/~chung/RE/Presentations07S/Team_1_Doc/Documents/SRS4.0.doc#:~:text=In%20short%2C%20the%20purpose%20of,interface%2C%20hardware%20and%20software%20requirements>.

# General Description of the Software

## Statement of Software Objective

The Car Rental Application is a user-friendly digital platform intentionally developed to streamline and enhance the car rental process. Operating independently, users can efficiently browse and reserve vehicles. The application can also connect with external services for added features. Its design allows for future expansions, and a visual diagram illustrates component interactions.

## Software Benefits / Functions

The car rental application aims to offer users a seamless experience in renting vehicles. Key functions include user registration, vehicle booking, secure payment handling, efficient pickup and return processes, user feedback integration, admin management tools, a notification system, and robust search and filter capabilities. These functions are interconnected to provide a user-friendly and comprehensive car rental service.

## User Characteristics

* Owner : Owners create and manage listings for their vehicles on the online car rental platform, Owners are responsible for ensuring that their vehicles are in good condition and meet all safety and regulatory standards.
* Customer : A customer can search and select the car and also could check the availability of the car, At the end of the rental period, customers return the vehicle to the specified drop-off location.
* Customer Service : Customer service remains available for assistance throughout the rental period and as a representative to help users navigate the booking process, guiding them through selecting the right vehicle, rental duration, and any additional options or services they may need.
* Insurance : Insurance manages insurance-related functionalities for users. This includes tasks such as user policy management, claims processing, premium calculation, policy renewal, user communication, and reporting/analytics.
* Police : Police may be involved in various situations, including the recovery of stolen vehicles, responding to traffic violations or accidents involving rented vehicles, investigating criminal activities, and addressing fraud or identity theft related to rental transactions.
* Bank : Bank here might not always refer to a traditional banking institution. In the context of online transactions, it could also include third-party payment processors or digital wallet services that handle the financial aspects of the car rental transactions.

## Software / System Limitations

The software is accessible via both web and app platforms, allowing users to connect through mobile phones or PCs using Wi-Fi or mobile data. The app is available on the Google Play Store for Android and the App Store for iOS, ensuring a user-friendly experience across different devices. This dual-platform approach enhances accessibility and flexibility for users.

# Detailed Description of Software

The car rental application is a software solution designed for effective management of car rental businesses. It serves two primary user classes-customers and administrators. Key features include user registration, vehicle search, reservation, and online payment for customers. Administrators can manage the fleet, provide customer support, and generate reports. The functional hierarchy comprises a user interface, reservation, payment, fleet management, customer support, and reporting modules. Operating modes include online and walk-in booking. Object classes involve user accounts, vehicles, and reservations. Non-functional requirements rioritize security, scalability, reliability, performance, and usability to create a user-friendly and secure platform for optimizing the car rental process.

## Requirement Descriptions

### Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirement ID | Feature | Description |
| 1. | FR-001 | Vehicle Input | This function shall be used by users to input items into the system |
| 2. | FR-002 | Vehicle Booking | This function shall be used by users to book a vehicle for a specified duration and location. |
| 3. | FR-003 | Vehicle Management | Customer shall use the feature to be able to add new vehicles to the system or remove existing ones. |
| 4. | FR-004 | Customer Registration | Customers shall be able to create accounts with personal information. |
| 5. | FR-005 | Customer Authentication | Customer shall authentication is required for secure access to the application. |
| 6. | FR-006 | Rental Calculation | The application should calculate rental prices based on factors like the type of car, duration of rental, and any additional services |
| 7. | FR-007 | Customer Billing | Customers shall receive clear and detailed invoices, including rental charges, taxes, and any applicable fees. |
| 8. | FR-008 | Payment Gateway Integration | The application integrate with a secure payment gateway for processing rental payments. |
| 9. | FR-009 | User Reviews and Ratings | Customers shall be able to leave reviews and ratings for the rented cars and overall service. |
| 10. | FR-010 | Location Services | Integration with mapping services shall be used to assist users in locating rental locations and returning vehicles. |
| 11. | FR-011 | Insurance Information | Function that shall be used to display information about insurance options for rented vehicles. |
| 12. | FR-012 | Feedback | Function shall provide a mechanism for users to submit feedback and contact customer support. |
| 13. | FR-013 | Mobile Compatibility | The application shall be accessible and user-friendly on mobile devices. |

### Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Quality | Requirement ID | Description |
| 1. | Security  Safety | NFR-01 | This function shall users to input items into the system |
| 2. | Performance | NFR-02 | The application shall respond to user requests within 2 seconds under normal load conditions. |
| 3. | Availability | NFR-03 | The application shall be available 24/7, allowing users to make reservations and access information at any time. |
| 4. | Compliance | NFR-04 | The application shall comply with relevant data protection and privacy regulations |
| 5. | Usability | NFR-05 | The customer interface shall be intuitive and easy to navigate. |
| 6. | Reliability | NFR-06 | The application shall have a recovery time of no more than 30 minutes. |
| 7. | Compatibility | NFR-07 | The application shall be compatible with major web browsers (e.g., Chrome, Firefox, Safari) and mobile platforms (iOS, Android). |
| 8. | User Load Handling | NFR-08 | The application shall be capable of handling a minimum of 10,000 simultaneous users during peak hours. |
| 9.. | Backup and Recovery | NFR-09 | Regular backups of the system data shall be performed, and a comprehensive recovery plan shall be in place. |

## Analysis Modeling

### Usecase Diagram

#### 3. Usecase Scenario #1 < use case Check Car Condition>

1. Pre-Condition : The system displays the check car condition
2. Use Case Description
   1. Primary Flow:
      * Customer shall starts the Car Condition Check application
      * Customer shall enters car information including make, model, year, and VIN
      * Customer shall choose the type of condition check (general inspection).
      * Customer shall follows guided steps to perform the check; pplication generates a clear report on the car’s condition.
   2. Alternative Flow:
3. Post-Condition :The customer know their car’s condition and can take appropriate actions based on the insights provided by the application.

#### Usecase Scenario #2 < use case Rent Car>

1. Pre-Condition: The system presents the ‘Rent Car’ option.
2. Use Case Description
   1. Primary Flow:
      * The customer shall select a car from the given options and click on the icon.
      * The application shall display details about the chosen car including its mechanical specifications and its price.
      * The customer shall click next on the chosen car.
      * The application shows the input address page and the available payment options.
      * The customer shall input the needed information and clicks next.
      * The application shall display options on whether the customer wants manually to pick up the car or have it sent to their address.
      * The customer shall select their preferred option and clicks next.
   2. Alternative Flow:
3. Post-Condition: The application saves the customer’s data and sends the order.

#### Use case Scenario #3 < use case Return car>

1. Pre-Condition: The system display return car
2. Use Case Description
3. Primary Flow:
   * Customer shall starts the return process through the rental platform.
   * Customer shall inputs reservation info, reason for return, and selects drop-off location.
   * The customer shall returns the car to the chosen location within the specified time.
   * Rental company shall inspects the car, confirms payment if needed, and issues a confirmation and receipt to the customer.
4. Alternative Flow:
   * Customer Services shall identifies a potential problem with the return, such as location unavailability or unexpected fees.
   * Customers shall receives a notification about the issue and is prompted to address it
   * Customer shall responds by selecting alternatives, providing clarification, or addressing the identified problem
   * Customer Services shall adjusts the return details based on user input.
   * Customers shall review and confirm the updated details, acknowledging any changes, and complete the return process.
5. Post-Condition: After a successful return of the car, the system updates the reservation status, issues a confirmation and detailed receipt to the customer, and makes the vehicle available for future bookings.

#### Use case Scenario #4 < use case Payment>

1. Pre-Condition: The system display payment
2. Use Case Description
3. Primary Flow:
   * Customer shall selects a car and proceeds to the checkout/payment screen.
   * Customer Service shall give total cost of the rental, including any additional fees.
   * Customer shall chooses a payment method (debit card or QRIS).
   * Customer Service shall receives confirmation from the payment
   * Customer Service shall creates a digital receipt and send to customer's email and/or available for download.
   * Customer Service shall updates reservation status
   * Customer Service shall completed transaction.
4. Alternative Flow
5. Post-Condition : payment authorization, customer service updates the reservation status, generates a digital receipt, and provides the user with rental details, confirming the completion of the transaction

#### Use case Scenario #5 < use case Provide Car>

1. Pre-Condition: The system presents the ‘Provide Car’ option.
2. Use Case Description
   1. Primary Flow:

* The owner shall initiates the process by selecting the "Provide Car" option from the system menu or interface.
* The system shall presents a form or wizard for the user to input relevant details about the car they want to provide.
* The owner shall have the option to upload images or videos of the car to enhance the listing.
* The system may ask the user to set a selling price or provide guidance based on market value and some additional details such as payment methods accepted, delivery options, and any terms or conditions should be specified by the owner.
* The system shall validates the provided information, checking for completeness and correctness.
* Once all necessary information is provided and validated, the user shall confirms the listing.
* The car listing shall becomes visible on the platform for potential buyers to browse.
* The owner shall has the option to edit, update, or remove the listing at any time through their account settings.
  1. Alternative Flow

1. Post-Condition: The owner's successful provision of the requested car in good working condition and the customer confirmation and satisfaction with the provided car

### Class Diagram

#### Class Diagram Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Class ID** | **Class Name** | **Attribute (visibility)** | **Method / Operation** |
| 01 | *Customer* | Id\_customer : int  Name : string  Address : string | MakeReservation(car: Car): Reservation  ViewReservationHistory(): List<Reservtion> |
| 02 | *Car* | Id\_car : int  name : string  brand : string  plate : string  car\_type : string | isAvailable(): boolean |
| 03 | *Customer Service* | *car: List<Car>*  *customers: List<Car>* *rentals: List<Rental>* | addCar(car: Car): Void  addCustomers(customer: Customers): void  CreateReservation(customer: Customer, car : Car, startDate:Date, endData: Data): Reservation  returnCar(rental: Rental,returnData: Date) |
| 04 | *Reservation* | ReservationId :int  car : Car  customer : Customer  startDate : Date  endDate : Date  totalCost : Decimal | calculateRentalCost() : Double  isLate() : boolean |
| 05 | *Bank* | id\_bank : int  bank\_type : int  bank\_name : int  accounts : List <BankAccount> | ProcessPayment(customer :Customer, rental : Rental, amount : double) ; boolean  issueRefund(customer : Customer, rental : Rental, amount : double) : boolean |
| 06 | *Bank Account* | accountNumber: String  balance: decimal  accountHolder: Customer | issue\_policy(): void  renew\_policy(): void |
| 07 | *Police* | Id\_police : int | Inspect\_car(car : car) void  checkCar (car : car) boolean |
| 08 | *Insurance* | id\_insurance : int  policy\_number :string  coverage\_amount : decimal | issue\_policy(): void  renew\_policy() : void |

## System Environment Requirements

### Operating Environment

The application shall be compatible with all available computer and smartphone-based operating systems and devices. The aplication shall take up very minimal resources to run and allows customers to use the given services with smooth, uninterrupted performance.

### Hardware Requirements

Logical Characteristics

1. User Interface (UI)
   * Type: Graphical User Interface (GUI) for ease of use.
   * Devices Supported: Desktop/laptop computers, mobile devices (smartphones, tablets).
   * Nature of Interaction: User-friendly forms for car rental requests, returns, and account management.
2. Car Interface

* Type: Machine-to-Machine (M2M) interface.
* Devices Supported: Onboard computers in rental cars.
* Nature of Interaction: Data exchange for tracking car status (location, fuel level, maintenance history).

1. Payment Gateway

* Type: API-based interaction.
* Devices Supported: Desktop and mobile devices for online payments (QRIS and Bank Transfer only).
* Nature of Interaction: Secure data exchange for processing payments.

1. GPS Integration

* Type: GPS module interaction.
* Devices Supported: Mobile devices and cars with GPS.
* Nature of Interaction: Real-time location tracking for navigation and fleet management.

Physical Characteristics

1. Server Infrastructure

* Type: Centralized server for data processing.
* Devices Supported: Rack-mounted servers.
* Nature of Interaction: Store and retrieve data, process rental requests, and manage user accounts.

1. Database System

* Type: Centralized database.
* Devices Supported: Server hosting the database.
* Nature of Interaction: Store and retrieve customer data, reservation details, and car information.

1. Car Hardware

* Type: Onboard computers and sensors.
* Devices Supported: Installed in rental cars.
* Nature of Interaction: Transmit data on car status, location, and condition to the central system.

1. Payment Terminals

* Type: Point of Sale (POS) terminals.
* Devices Supported: QRIS, mobile payment devices or bank transfer.
* Nature of Interaction: Process payment transactions securely.

1. Network Infrastructure

* Type: Internet and/or Intranet.
* Devices Supported: Routers, switches, and communication equipment.
* Nature of Interaction: Enable communication between client devices, servers, and external services

### Software Dependencies

* + Programming Language: C++ Language and for database MySQL
  + Database Management System (DBMS) : for the database system use MySQL
  + Payment Gateway Integration: Integrate a secure payment gateway use QRIS or Bank Transfer
  + Geolocation Services: Utilize geolocation services or APIs (e.g., Google Maps API) for tracking and mapping functionalities.

Programming Language :

C++ is a programming language. It's an extension of the C programming language with additional features such as object-oriented programming. It provides low-level memory manipulation features along with high-level abstractions, making it versatile and suitable for various applications, including system/software development, game development, and more.

Database :

MySQL is an open-source relational database management system that organizes data into tables using Structured Query Language (SQL). It's known for its scalability, cross-platform compatibility, and high performance. MySQL is commonly used in web development for storing and retrieving data, and it comes in two editions – Community (free and open-source) and Enterprise (with additional features and support).

# Appendix

The appendix for the car rental application includes vital supplementary requirements beyond the Software Requirements Specification (SRS). These encompass database specifications, internationalization needs, legal compliance, and goals for code reuse. Addressing these aspects ensures comprehensive development, covering database architecture, accommodating international users, complying with legal regulations, and promoting code efficiency and maintainability throughout the project.

Appendix A: List of Difficult Words

1. Streamline : make (an organization or system) more efficient and effective by employing faster or simpler working methods.

2. Constraints : a limitation or restriction.

3. Internationalization : the action or process of making something international.

4. Compliance : the action or fact of complying with a wish or command.

5. Maintainability : Maintainability is the ease of maintaining or providing maintenance for a functioning product or service.

6. Backend : the part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data.

7. Frontend : relating to or denoting the part of a computer system or application with which the user interacts directly.

8. Compatible : (of two things) able to exist or occur together without conflict.

9. Fleet : refers to the collection of vehicles available for rental

10. Comprises : used to convey the idea that the functional hierarchy of the car rental application is made up of several distinct components or modules.