Addis Ababa Institute of Technology

School of Information Technology and
Engineering Department of SiTE

Vroomy

Software Requirements Specification (SRS)

Team Members

Name	ID
BEKA BIRHANU ATOMSAU	UGR/3402/14
BETHEL DEREJE TEFERI	UGR/1397/14
DAWIT BELAY YIBELTAL	UGR/8622/14
LEMI DINKU GILO	UGR/3860/14
SAMRAWIT DAWIT MENGSTU	UGR/0221/14
SIMON DEREJE WOLDEAREGAY	UGR/0952/14

Advisor: Nuniyat Kifle
Date 11/15/2023

Table Of Content

Acronyms

SRS - System Requirements Specification
UI - User Interface
API - Application Program Interface

DECLARATION

This document outlines the System Requirements Specification (SRS) for a state-of-the-art web-based car rental system, designed to function as an intermediary or broker in the car rental industry. The primary goal of this system is to provide a seamless and efficient platform for users to search, compare, and book rental cars from various providers. The system will prioritize user experience through a user-friendly interface, comprehensive car listings, and robust authentication mechanisms. Additionally, features such as user registration, mutual review submissions, and interactive notifications will enhance engagement and trust within the platform. The system will incorporate geo-location services for location-based searches, enabling users to find suitable rental options conveniently. For car owners, dedicated interfaces for managing inventory and customer interactions will be provided. This SRS declaration serves as a foundational document for the design and development of the web-based car rental system, ensuring it meets the specified requirements and objectives.

1. Introduction

The Web-Based Rental System, acting as a broker, is a cutting-edge online platform designed to revolutionize the rental industry. In today's dynamic market, the need for a seamless, trustworthy, and efficient rental service is paramount. This system serves as an intermediary, connecting renters and rental service providers in a secure, transparent, and user-friendly environment. By harnessing the power of technology, this platform simplifies the process of renting accommodations, vehicles, equipment, and various other assets, ensuring a hassle-free experience for both parties involved.

1.1 Purpose

This Software Requirements Specification (SRS) serves to detail the essential requirements, functionality, and limitations of the software solution under development. This document is created to offer a clear understanding of the project's scope, objectives, and system functionality to distinct audience groups involved in the project life cycle.

Intended Audiences:

1. **Development Team:** This document assists the development team by providing detailed technical and functional specifications required for **system design**, **implementation**, and **testing**. It ensures a clear understanding of the system's expected behavior and features.

2. **Course Teacher:** The SRS provides the course teacher with a comprehensive overview of the project's scope, functionality, and requirements. It aids in evaluating the project's alignment with learning objectives and grading criteria.

3. **Interested Parties:** including potential users, investors, or external parties interested in the project, can use the SRS to comprehend the system's intended functionality, potential use cases, and expected outcomes. It can also be used as a reference for future related projects.

Thus, the SRS intends to provide a coherent vision of the project's goals and functions to all involved parties by defining its audiences and purpose. This aims to ensure a shared and comprehensive understanding of the system's requirements and features.

1.2 Scope

Our project aims to develop a platform that fosters connectivity between car owners and renters, ensuring reliability and trust for both parties involved in vehicle rentals. The final product of this project will be a web app appropriate for multiple devices. The system will incorporate the following key features:

a. Establishing sustainable connectivity between car owners and renters.

b. Offering renters an extensive array of choices for vehicle selection.

c. Presenting car owners with access to a broad and diverse demand.

d. Providing a reliable and trustworthy platform for both parties involved.

e. Incorporating a review system enabling users to share feedback and ratings based on their rental experiences.

f. Implementing robust verification measures to ensure the reliability of interactions within the platform.

This project will not cover the following services:

1. **Legal Issues, Contracts, and Agreements:** Legal aspects, contracts, and agreements between car owners and renters will not be managed by the platform. Users are responsible for arranging legal matters independently.

2. **Transaction Handling:** The platform will not manage or intervene in financial transactions between car owners and renters. Users will conduct financial transactions independently outside the platform.

2. General description

Customer Integration: The system needs to integrate with a network of car rental companies or individual car owners to display their available vehicles and facilitate bookings. The system should support seamless integration, allowing customer to upload vehicle details, availability, and pricing information.

Multi-Vendor Management: The system should be capable of managing multiple vendors or car rental customers. This includes features such as on boarding new customers, maintaining customer profiles, and ensuring consistent service quality across all customers.

Reviews and Ratings: Implementing a review and rating system for both customers and renter can help build trust and improve the overall quality of the service. Users should be able to leave reviews and ratings for the rental experience and the customer.

User Management: The system should support user account management, allowing customers to create and manage their profiles. This includes features like registration, login, password management, and account settings.

Data Security and Privacy: Ensuring the security and privacy of customers data is paramount. The system should adhere to data protection regulations, implement encryption and secure protocols, and have robust measures in place to protect sensitive information.

2.1 Product Perspective

The car rental system serves as an intermediary between customers and multiple car rental companies or individual car owners. The system facilitates the process of renting vehicles by connecting customers with available rental options from various customers. Here is the product perspective for such a system:

System Context: The rental system interacts with multiple external systems and entities, including:

Car Rental Companies: The system integrates with various car rental companies, allowing them to list their available vehicles, pricing, and availability.

Customers: The system provides a platform for customers to search, compare, and book rental vehicles from different customers.

Interfaces: The system provides various interfaces to support seamless interactions, including:

User Interface: The system offers a user-friendly web interface, allowing customers to search, compare, and book rental vehicles, as well as enabling customers to manage their vehicle listings and reservations.

customer Integration Interface: The system provides a platform for customers to upload and manage their vehicle inventory, pricing, and availability.

Vehicle Information: The system collects and displays vehicle details, such as make, model, year, features, and pricing, obtained from customer systems.

Compatibility: The system is designed to be compatible with various web browsers, operating systems, and devices to ensure a seamless user experience for both customers and customers.

Standards and Regulations: The system adheres to industry standards and regulations for data security, privacy, and compliance. It ensures the protection of customer and customer information and complies with relevant regulations, such as GDPR, to safeguard user data.

2.2 Product functions

Summary of Functions:

User Registration and Authentication:

- Allow users to register and create accounts.
- Provide options for user authentication, such as username/password or social media login.

Vehicle Search and Filtering:

- Enable users to search for available rental vehicles based on criteria like location, date, and vehicle type.
- Provide filtering options to refine search results based on specific preferences, such as price range, vehicle features, or transmission type.

Vehicle Booking and Reservation Management:

- Allow users to book rental vehicles by selecting desired dates, locations, and vehicle options.
- Provide confirmation and reservation details to users and customers.
- Enable users to manage their reservations, including modifying or canceling bookings.

customer Management:

- Provide a platform for car rental companies or individual owners to list their vehicles and manage their inventory.
- Allow customers to update vehicle details, availability, and pricing information.
- notifications between customers and users.

User Reviews and Ratings:

- Enable users to rate and provide feedback on rental experiences.
- Display reviews and ratings to help users make informed decisions.

Admin Dashboard:

- Provide an administrative dashboard for system administrators to manage users, customers, and system configurations.
- Enable administrators to monitor system performance, handle exceptions, and manage system-wide settings.

System Security and Privacy:

- Implement appropriate security measures to protect user data, prevent unauthorized access, and ensure data privacy.

2.3 User Characteristics

User Expertise:

- The software is designed for both novice and experienced users in the car rental domain.
- Novice users may have limited knowledge of the rental process and may require intuitive interfaces and clear instructions.
- Experienced users may have specific preferences and may value advanced search and filtering options.

Technical Proficiency:

- Users are expected to possess basic computer literacy skills and be comfortable navigating web-based applications.
- The software should have a user-friendly interface with intuitive controls and clear instructions to accommodate users with varying technical proficiency.

System Compatibility:

- The software should be compatible with a range of devices, including desktop computers, laptops, tablets, and smart phones, to accommodate different user preferences and usage scenarios.
- It should support common web browsers and operating systems, ensuring a consistent user experience across platforms.

Security and Privacy:

- Users expect their personal and financial information to be handled securely and with privacy in mind.
- The software should incorporate appropriate security measures, such as encryption, secure authentication, and protection against data breaches.

Performance Expectations:

- Users expect the software to be responsive, with quick loading times and minimal latency.
- The system should handle concurrent user requests efficiently and provide a smooth user experience, even during peak usage periods.

Customer Support:

- Users may require access to customer support channels, including online help, FAQs, or direct communication with support representatives.
- The software should provide clear avenues for users to seek assistance or report any issues encountered.

2.4 General Constraints

This subsection of the SRS outlines the limitations and external factors that will impact the design and development of the Vroomy constraints encompass various aspects that developers need to consider, ensuring the system operates effectively within defined boundaries. The general constraints are as follows:

- **Regulatory Compliance:** The platform shall comply with all relevant legal and regulatory requirements, such as data protection, privacy, and intellectual property laws, in the regions where it operates. The platform shall also adhere to any industry-specific standards or best practices for web platforms.

- **Hardware Independence**: The platform shall be designed to operate independently of specific hardware requirements, such as memory, storage, or processing power. The platform shall utilize free database systems and hosting services to ensure optimal performance and reliability. The platform shall also support responsive web design to ensure usability across various devices, such as desktops, tablets, or smartphones.

- **Interface Compatibility:** The platform shall be compatible with various web browsers, such as Microsoft Edge, Google Chrome, or Mozilla Firefox. The platform shall also support integration with third-party applications or services, such as social media messaging systems, through APIs. The platform shall ensure seamless and reliable communication and data exchange with external systems.

- **Performance Standards:** The platform shall meet defined performance standards, such as response time, throughput, or availability. The platform shall be able to handle concurrent requests and transactions from multiple users without compromising its functionality or quality. The platform shall also implement appropriate testing and monitoring mechanisms to ensure its performance and reliability.

- **Security and Privacy:** The platform shall implement robust security and privacy measures to protect user data and transactions. This includes secure data transmission, encryption, and access control. The platform shall also respect user preferences and consent regarding data collection and usage. The platform shall provide a safe and secure environment for users to interact and transact with each other.

2.5 Assumptions and Dependencies

This subsection of the SRS lists factors that can affect the requirements stated in the document. These factors are not design constraints on the software but rather assumptions and dependencies that may impact the project. The Vroomy platform makes the following assumptions and dependencies:

Platform Independence: It is assumed that the platform will be developed to maximize platform independence, allowing it to run on various web browsers, operating systems, and devices.

Hardware Availability: The platform depends on the availability of compatible hardware to ensure optimal performance, though efforts will be made to minimize hardware-specific requirements.

Third-Party Services: The platform assumes the availability and reliability of third-party services and APIs for functionality such as optional messaging integration.

Regulatory Compliance: Assumptions are made that regulatory policies will have minimal impact on the platform, and it will be developed in compliance with relevant legal and regulatory requirements.

User-Provided Content: The platform depends on users to provide accurate and lawful content for car listings, profiles, and other user-generated data.

Data Backup and Recovery Services: It is assumed that data backup and recovery services will be accessible to mitigate data loss in the event of system failures.

Messaging Integration (Optional Feature): The integration of a known social media platform's messaging system depends on the availability and functionality of the chosen third-party service.

These assumptions and dependencies serve as important considerations for the development of the Vroomy platform. Changes or challenges related to these factors may necessitate adjustments to the requirements outlined in this SRS.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interfaces for the Vroomy have been designed with a formal and professional tone, adhering to a third-person style. They will be developed with a focus on user-friendliness and accessibility to ensure a seamless experience for both car renters and car owners. The following interfaces will be included:

Registration and Login Interface:

- Users shall have the option to select their role during the registration process, with choices for "Car Renter" and "Car Owner."
- Both renters and owners will register using their email address or social media profiles.
- The login page will provide a secure means for users to access their accounts.

Dashboard:

- After logging in, users will be directed to a personalized dashboard tailored to their selected role.
- Renters will view their current reservations, search for available cars, and manage their bookings.
- Car owners can manage their car listings, including details about car availability.

Search and Listings Interface:

- Renters will be able to search for available cars based on location, date, and other filters.

Profile Management:

- Users can edit their profiles, update contact information, and select their user role.

Document Sharing:

- Renters can access necessary documents for rental agreements. The platform will provide download able templates for renters to utilize when conducting transactions with car owners.

Cancellation:

- Users shall have the capability to cancel their registration at their discretion.

Messaging System (Optional):

- An optional feature allows users to integrate a known social media platform's messaging system for secure and convenient communication between renters and owners. This feature is provided for users who prefer to communicate through their existing social media accounts.

The user interfaces shall be designed with responsive web design principles to ensure usability on various devices, including desktops, tablets, and smartphones. The platform's primary objective is to provide a user-friendly and intuitive experience for connecting renters and car owners while offering the option to communicate through a familiar social media messaging system.

3.1.2 Hardware Interfaces

- No extra hardware interfaces are needed.
- The system will use the standard hardware and data communication resources.
- This covers general network connections at the server/hosting location, network servers, and network management tools, among others.

3.1.3 Software Interfaces

- OS: Windows 7, Linux
- Web Browser: The system is a web based application; clients need a modern web browser such as Mozilla Firebox, Internet Explorer, Opera, and Chrome. The computer must have an Internet connection in order to be able to access the system.

3.1.4 Communications Interfaces

- This system uses TCP/IP network protocol in conjunction with HTTP protocol, as well as HTTP protocol for communication with web servers and web browsers.
- All booking-related data will be exchanged between this application and the database. Through a feature known as HTTP Service, users can communicate with the server side of the HTTP protocol. This feature enables the program to use the information returned by the server to satisfy the user's request.

3.2 Functional Requirements

3.2.1 User Registration and Authentication

Introduction
User registration and authentication are essential for ensuring that the platform is secure and personalized.

Inputs

- User-provided registration information, including name, email, password, and contact details.
- User authentication information, such as username and password.

Processing

- Verify user-provided registration information for accuracy and uniqueness.
- Authenticate users during login to ensure data security.

Outputs

- Create user accounts upon successful registration.
- Allow authenticated users to access their profiles and the platform's features.

Error Handling

- Notify users when registration information is incomplete or invalid.
- Implement security measures to protect user data and prevent unauthorized access.

3.2.2 Notification System

Introduction

A notification system should inform users and car providers of important events and updates related to their rentals and interactions on the platform.

Inputs

- Rental booking confirmations, updates, and notifications from car rental providers.
- User-generated messages and requests.

Processing

- Process and deliver real-time notifications to users via email, in-app messages, or push notifications.
- Manage communication between users and car rental providers.

Outputs

- Notify users of booking confirmations, updates, messages, and other relevant information.
- Ensure timely communication between users and car rental providers.

Error Handling

- Handle notification delivery failures and provide a mechanism for users to reattempt or troubleshoot any issues.

3.2.4 Comprehensive Listings

Introduction

The comprehensive listings feature should aggregate a wide range of car rental options from various providers to offer customers ample choices.

Inputs

- Information and rental listings from multiple car rental providers.

Processing

- Collect and organize rental listings from different providers into a centralized database.

- Ensure that the listings are regularly updated to reflect real-time availability and pricing.

Outputs

- Present a diverse and comprehensive list of rental cars from different providers, with various vehicle types, locations, and pricing options.
- Allow users to browse through the listings conveniently.

Error Handling

- Handle data inconsistencies or errors in the rental listings, providing users with accurate and up-to-date information.
- Notify users when certain listings are no longer available or have changed.

3.2.5 Customer Interaction Management for Car Owners

Introduction

The system shall provide functionality for car owners to manage customer interactions effectively, including responding to inquiries, accepting or declining rental requests, and maintaining transparent communication.

Inputs

- Car owner's interaction with customer inquiries, requests, and messages.
- Rental requests from customers.

Processing

- Process customer inquiries and rental requests submitted to car owners.
- Enable car owners to accept or decline rental requests in a timely manner.
- Facilitate transparent communication channels between car owners and customers.

Outputs

- Notify customers of car owner responses to inquiries and rental requests.
- Maintain a record of customer interactions and communication for transparency and dispute resolution.

Error Handling

- Ensure that car owners respond to inquiries and rental requests within a reasonable time frame.
- Provide a mechanism for addressing and resolving disputes related to interactions and communication.

3.2.6 Car Management for Car Owners

Introduction

The system shall provide functionality for car owners to manage their car inventory, including adding new cars, updating car descriptions, images, and pricing.

Inputs

- Car owner's input to add, update, or modify car listings.
- Details related to the car, such as description, images, and pricing.

Processing

- Process car owner requests to add, update, or modify car listings in their inventory.
- Validate and verify the accuracy of the information provided.

Outputs

- Reflect changes in the car owner's inventory, including newly added cars and updated information.
- Ensure that the platform's listings are up-to-date and accurate.

Error Handling

- Validate and verify the information provided by car owners for accuracy and completeness.
- Handle any errors or issues related to car management, such as data inconsistencies.

3.2.7 Mutual Review System

Introduction

The review system should allow both customers and car owners to provide feedback and ratings on their rental experiences, fostering a sense of trust and enabling informed decisions for both parties.

Inputs

- User-submitted reviews and ratings from both customers and car owners.
- Rental transaction data, including details of the car rental experience.

Processing

- Collect and store user reviews and ratings associated with specific rental transactions, considering feedback from both customers and car owners.
- Calculate and display overall ratings for customers and car owners based on the received feedback.

Outputs

- Display user reviews and ratings for each customer and car owner involved in a rental transaction.
- Enable both customers and car owners to make informed decisions by considering the feedback and ratings of each other.

Error Handling

- Implement mechanisms to prevent fraudulent or malicious reviews from both customers and car owners.
- Allow users to report inappropriate content, and ensure timely moderation and resolution of reported issues for both parties involved in the transaction.

3.2.8 Geo-Location Services (optional feature)

Introduction

Geo-location services should enable users to find rental cars based on their current location and provide location-based search options.

Inputs

- User location data (if allowed by the user).
- Location-based search criteria and preferences.

Processing

- Utilize geo-location data to provide location-specific rental listings.
- Enable users to search for rental cars based on their desired location.

Outputs

- Display rental listings relevant to the user's location or search criteria.
- Provide users with accurate location information for the rental cars.

Error Handling

- Ensure user privacy by obtaining appropriate permissions for accessing location data.
- Handle location-based search errors and provide alternative search options.

3.2.9 Providing Contact Information for the Booking Process

Introduction

The facilitated booking process should streamline the steps involved in booking a rental car and connect customers with car rental providers.

Inputs

- User's selection of a specific rental option.
- Contact details and terms provided by the car rental provider.

Processing

- Display essential information about the selected rental option, including car details, pricing, and availability.
- Facilitate communication between the user and the car rental provider by providing their contact details.

Outputs

- Enable users to directly communicate with the car rental provider to arrange payment, rental agreements, and address any specific requirements.
- Provide a seamless transition from selecting a rental option to initiating contact with the provider.

Error Handling

- Ensure that user queries or messages are reliably delivered to the car rental provider.
- Offer a support system to handle issues related to booking, communication, or disputes.

3.2.10 User Profiles

Introduction

User profiles should allow users to manage their information, preferences, and rental history.

Inputs

- User updates personal information, including contact details and preferences.
- Rental history and transaction data.

Processing

- Store and update user profile information securely.
- Track and display users' rental history and transaction details.

Outputs

- Display user profiles with accurate information and preferences.
- Provide access to rental history, allowing users to track past transactions.

Error Handling

- Ensure data consistency and integrity in user profiles.
- Handle and notify users of any errors or issues with profile updates.

3.3 Use Cases

3.3.1 Use Case 1: User Registration and Authentication

Goal: Users intend to create accounts, register on the platform, and authenticate themselves to access personalized services.

Primary Actor: User

Scope: Car Rental Platform

Level: User

Precondition: User is accessing the platform for the first time or needs to log in.

Success End: User successfully registers or logs in to access the platform's features.

Failure End Condition: User encounters issues with registration or authentication, leading to unsuccessful login.

Trigger: User selects the "Register" or "Log In" option.

Main Success Scenario:

1. User selects the **"Register"** or **"Log In"** option on the platform.

If registering:

2. The system presents a registration form and prompts the user to provide information, including name, email, password, and contact details.
3. The user enters the required information.
4. The system verifies the accuracy and uniqueness of the user-provided registration information.
5. Upon successful verification, the system creates a user account and informs the user of successful registration.

If logging in:

2. The system prompts the user to enter their authentication information, typically a username and password.
3. The system authenticates the user by verifying the provided login information.
4. The authenticated user gains access to their profile and the platform's features.

Extensions (Error Scenarios):

Registration

1. The user encounters difficulties with the registration form.
2. The system provides guidance and instructions for completing the registration form.
3. The user successfully completes the registration form.

Login

I. **Invalid information**

1. The system encounters registration information that is incomplete or invalid.
2. The system notifies the user of the issue and advises them to correct the information.
3. The user revises and completes the registration information successfully.

II. **Forgotten credential**

1. The user faces difficulties in logging in due to incorrect or forgotten login credentials.
2. The system provides a password reset or recovery option.
3. The user successfully resets their password and logs in.

3.3.2 Use Case 2: Managing User Profiles

Goal: Users intend to manage their user profiles, update personal information, preferences, and access their rental history.

Primary Actor: User

Scope: Car Rental Platform

Level: User

Precondition: User is logged into the platform.

Success End: User successfully manages their user profile, updates information and preferences, and accesses their rental history.

Failure End Condition: User encounters issues with profile management, information updates, or accessing rental history.

Trigger: User selects the "Manage User Profile" option.

Main Success Scenario:

1. User selects the "Manage User Profile" option on the platform.
2. The system displays the user's profile, showing their personal information and preferences.
3. User updates their personal information, such as contact details and preferences, and saves the changes.
4. The system securely stores and updates the user's profile information.
5. The system provides access to the user's rental history and transaction details.

Extensions (Error Scenarios):

I. **From user**

1. The user encounters difficulties while updating their personal information.
2. The system offers guidance and assistance for completing the update.
3. The user successfully updates their personal information.

II. **From system**

1. The system encounters issues in retrieving or displaying the user's rental history.
2. The system displays an error message and advises the user to try again later.
3. The user revisits the rental history section at a later time.
4. The user seeks assistance from the platform's customer support.

3.3.3 Use Case 3: Receiving and Managing Notifications

Goal: Users and car rental providers aim to receive and manage notifications related to their rentals, interactions, and other important events on the platform.

Primary Actor: User or Car Rental Provider

Scope: Car Rental Platform

Level: User

Precondition: User or car rental provider is logged into the platform and has enabled notifications.

Success End: User or car rental provider successfully receives, views, and manages notifications.

Failure End Condition: User or car rental provider encounters issues with notification delivery or management.

Trigger: An event or update on the platform necessitates sending a notification to the user or car rental provider.

Main Success Scenario:

1. An event or update on the platform triggers the need for a notification (e.g., booking confirmation, message from a user, rental updates).
2. The system processes the notification and sends it to the user or car rental provider via the selected notification method (e.g., email, in-app message, push notification).
3. The user or car rental provider receives and views the notification.
4. The user or car rental provider takes necessary actions based on the notification (e.g., responds to a message or acknowledges a booking confirmation).

Extensions (Error Scenarios):

1. The system encounters difficulties in delivering a notification.
2. The system provides an error message and advises the user or car rental provider to check their notification settings and retry the delivery.
3. The user or car rental provider revisits the notification and delivery settings.

Variations (Alternative Scenarios):

Notifications can cover various events, including booking confirmations and reminders. Users and car rental providers may have varying preferences for which events trigger notifications.

3.3.4 Use Case 4: Browsing Comprehensive Car Rental Listings

Goal: Customer aims to browse comprehensive car rental listings, which aggregate options from various providers, to find the desired rental car.

Primary Actor: Customer

Scope: Car Rental Platform

Level: User

Precondition: Customer is logged into the platform.

Success End: Customer successfully finds and selects a rental car that matches their requirements.

Failure End Condition: Customer does not find a suitable rental car or encounters issues with the listings.

Trigger: Customer selects the "Browse Car Rental Listings" option.

Main Success Scenario:

1. Customer selects the "Browse Car Rental Listings" option on the platform.
2. The system retrieves rental listings from multiple car rental providers and presents them in a centralized database.
3. The customer navigates through the comprehensive listings, viewing options from various providers, with details on vehicle types, locations, and pricing.
4. The customer applies filters or sorting options, such as vehicle type or price range, to narrow down their search.
5. The customer finds a rental car that matches their preferences and selects it.
6. The customer proceeds to book the selected rental car or make an inquiry.

Extensions (Error Scenarios):

I. From system

1. The system encounters an error while retrieving rental listings from providers.
2. The system displays an error message and advises the customer to try again later.
3. The customer reattempts to access the listings.

II. From user

1. The customer does not find a suitable rental car in the available listings.

2. The customer adjusts the filter criteria or revises their preferences and performs a new search.

Variations (Alternative Scenarios):

- The customer may choose to explore the comprehensive listings without applying specific filters, allowing for a broader search and discovery of rental cars based on preferences.

3.3.5 Use Case 5: Searching for Rental Cars

Goal: Customer wishes to search for available rental cars using the user-friendly interface.

Primary Actor: Customer

Scope: Car Rental Platform

Level: User

Precondition: Customer is logged into the platform.

Success End: Customer successfully finds a suitable rental car.

Failure End Condition: Customer does not find a suitable rental car.

Trigger: Customer selects the "Search for Rental Cars" option.

Main Success Scenario:

1. Customer selects the "Search for Rental Cars" option on the user-friendly interface.
2. The system presents a search screen with fields for entering preferences, including location, dates, vehicle type, and other relevant criteria.
4. Customer enters their preferences, including location, dates, and vehicle type.
5. The system processes the customer's preferences and search criteria to fetch relevant rental listings from the database.
6. The system displays a list of rental cars that match the customer's preferences, with car details, pricing, location, and availability.
7. The customer reviews the search results, navigates through the listings, and filters options based on their preferences.
8. The customer finds a suitable rental car, selects it, and proceeds to book it or make an inquiry.

Extensions (Error Scenarios):

I. **From user**

1. Customer encounters an issue with the search screen.
2. The system provides clear instructions for completing the search criteria.
3. The customer successfully completes the search criteria.

II. **From system**

1. The system cannot find any rental cars matching the customer's preferences.
2. The system displays a message stating that no results were found.
3. The customer revises their search criteria and repeats the search.

Variations (Alternative Scenarios):

In addition to entering preferences, the customer may choose to browse or explore the website to find cars to their liking.

3.3.6 Use Case 6: Utilizing Geo-Location Services for Location-Based Searches (optional feature)

Goal: Users aim to find rental cars based on their current location and perform location-based searches using geo-location services.

Primary Actor: User

Scope: Car Rental Platform

Level: User

Precondition: User has enabled location services or provided location data.

Success End: User successfully finds rental cars based on their current location or specified search criteria.

Failure End Condition: User encounters issues with location services, search errors, or privacy concerns.

Trigger: User selects the "Search Based on Location" option or enables location services.

Main Success Scenario:

1. User selects the "Search Based on Location" option or enables location services on the platform.
2. The system retrieves the user's current location or location-based search criteria.
3. The system utilizes geo-location data to provide rental listings that are relevant to the user's current location or specified search criteria.
4. The user views rental listings that match their location or search preferences.
5. The user selects a rental option based on the displayed location or search criteria.
6. The user proceeds to book the selected rental car or make an inquiry.

Extensions (Error Scenarios):

I. **From user**

1. The user does not grant permission for the platform to access their location data.
2. The system informs the user that location-based search is not available and provides alternative search options.
3. The user revisits the platform's settings to enable location services.

II. From system

1. The system encounters difficulties in providing accurate location-based rental listings.
2. The system displays an error message and advises the user to try again later.
3. The user revisits the search or location-based search option.

Variations (Alternative Scenarios):

1. Users may choose to search for rental cars based on their current location or input specific location-based search criteria, such as city, neighborhood, or destination.
2. Users may enable or disable location services based on their preferences, privacy concerns, or the need for location-specific searches.

3.3.7 Use Case 7: Car Owner Managing Customer Interactions

Goal: Car owners aim to manage customer interactions effectively, including responding to inquiries, accepting or declining rental requests, and maintaining transparent communication on the platform.

Primary Actor: Car Owner

Scope: Car Rental Platform
Level: User

Precondition: Car owner is logged into the platform and has received inquiries or rental requests from customers.

Success End: Car owner effectively manages customer interactions, responds to inquiries and rental requests, and maintains transparent communication.

Failure End Condition: Car owner faces difficulties in managing customer interactions or encounters errors related to responses and communication.

Trigger: Car owner receives inquiries or rental requests from customers.

Main Success Scenario:

1. Car owner receives inquiries or rental requests from customers on the platform.

2. The system displays the inquiries and rental requests, allowing car owner to review and respond.

3. Car owner responds to customer inquiries, providing relevant information and assistance.

4. For rental requests, car owner can choose to either accept or decline the request within a reasonable time frame.

5. The system processes the response, notifying the customer of the car owner's decision.

6. Car owner maintains transparent and timely communication with customers throughout the rental process.

7. The system records customer interactions and communication for transparency and potential dispute resolution.

Extensions (Error Scenarios):

I. **owner encounters difficulties**
   1. Car owner encounters difficulties in responding to customer inquiries or providing relevant information.
   2. The system provides guidance and assistance for crafting a response.
   3. Car owner successfully responds to customer inquiries.

II. **Car owner faces challenges**
   1. Car owner faces challenges in accepting or declining rental requests within a reasonable time frame.
   2. The system provides reminders and notifications to ensure timely responses.
   3. Car owner successfully accepts or declines rental requests within the required time.

Variations (Alternative Scenarios):

- Car owners may respond to customer inquiries, accept or decline rental requests, and maintain communication based on their preferences and the specific details of the rental process, promoting effective interactions and customer satisfaction.

3.3.8 Use Case 8: Initiating Contact with car Rental Provider

Goal: Customer intends to initiate contact with the car rental provider to proceed with the booking process.

Primary Actor: Customer

Scope: Car Rental Platform

Level: User

Precondition: Customer has selected a specific rental option and is prepared to book a rental car.

Success End: Customer successfully initiates contact with the car rental provider.

Failure End Condition: Customer encounters difficulties in initiating contact or has concerns related to the rental process.

Trigger: Customer selects a specific rental option and chooses to proceed with booking.

Main Success Scenario:

1. Customer selects a specific rental option from the listings.

2. The system displays essential information about the selected rental option, including car details, pricing, and availability.

3. The system facilitates communication between the customer and the car rental provider by providing contact details (e.g., phone number, email, or in-app messaging).

4. The customer contacts the car rental provider to discuss payment, rental agreements, and any specific requirements.

5. The customer and the car rental provider establish clear communication and proceed with the booking process.

Extensions (Error Scenarios):

1. The system experiences technical issues while providing contact details.
2. The system displays an error message and advises the customer to retry contacting the provider later.

Variations (Alternative Scenarios):
- Some customers may choose to initiate contact using the phone number provided, while others may prefer email for communication with the car rental provider. The system should accommodate the customer's preferred method of contact.

3.3.9 Use Case 9: Car Owner Managing Car Inventory

Goal: Car owners aim to manage their car inventory, including adding new cars, updating car descriptions, images, and pricing on the platform.

Primary Actor: Car Owner

Scope: Car Rental Platform

Level: User

Precondition: Car owner is logged into the platform and wants to make changes to their car inventory.

Success End: Car owner successfully manages their car inventory, adds new cars, and updates existing listings.

Failure End Condition: Car owner encounters difficulties in managing their car inventory or faces errors in adding new cars or updating existing listings.

Trigger: Car owner selects the "Manage Car Inventory" option.

Main Success Scenario:

1. Car owner selects the "Manage Car Inventory" option on the platform.
2. The system displays the car owner's inventory, including existing car listings.
3. Car owner chooses to add a new car to their inventory.
4. Car owner provides the required information for the new car, including description, images, and pricing.
5. The system processes the request to add the new car and updates the car owner's inventory.
6. Car owner decides to update an existing car listing by modifying details like description, images, or pricing.
7. Car owner provides the updated information.
8. The system processes the request to update the existing car listing and reflects the changes in the car owner's inventory.

Extensions (Error Scenarios):

a. Car owner encounters difficulties in providing the required information for the new car.
a.1 The system provides guidance and assistance for completing the information.
a.2 The car owner successfully provides the required information.
b. The system faces difficulties in processing the request to add the new car.
b.1 The system notifies the car owner of the issue and ensures that the new car is added promptly.
c. Car owner faces challenges in providing updated information for an existing car listing.
c.1 The system provides guidance and assistance for completing the update.
c.2 The car owner successfully provides the updated information
d. The system encounters issues in processing the request to update the existing car listing.
d.1 The system notifies the car owner of the issue and ensures that the update is applied promptly.

3.3.10 Use Case 10: Submitting and Viewing Mutual Rental Reviews

Goal: Customers and car owners aim to submit their feedback and view the reviews and ratings of each other to make informed rental decisions.

Primary Actors: Customer, Car Owner

Scope: Car Rental Platform

Level: User

Precondition: A rental transaction has been completed between a customer and a car owner, or a customer is considering renting a car.

Success End: Both customers and car owners can successfully submit reviews or view reviews and ratings for each other.

Failure End Condition: Users encounter difficulties submitting a review or accessing reviews and ratings.

Trigger: User (either customer or car owner) selects the "Submit a Review" or "View Reviews and Ratings" option.

Main Success Scenario:

1. User (either customer or car owner) selects the "Submit a Review" or "View Reviews and Ratings" option.

2. If submitting a review:
    - The system prompts the user to select the specific rental transaction for which they wish to provide feedback.
    - The user writes their review, provides a rating, and submits it.

3. If viewing reviews and ratings:
    - The system displays reviews and ratings for the specific rental transaction, showing feedback from both the customer and the car owner.
    - The user views the feedback and ratings to evaluate both parties.

Extensions (Error Scenarios):

2a. The user faces difficulties in submitting their review.

2a.1 The system offers assistance and guidance for the review submission process.

2a.2 The user successfully submits their review.

3a. The system encounters issues in retrieving or displaying reviews and ratings for both parties.

3a.1 The system displays an error message and advises the user to try again later.

3a.2 The user revisits the reviews and ratings section at a later time.

3a.3 The user seeks assistance from the platform's customer support.

Variations (Alternative Scenarios):

- Users may submit reviews and ratings for multiple rental transactions they have completed with different parties.
- Users may choose to browse reviews and ratings for various car rental providers to assess their reputation and the quality of service, considering feedback from both customers and car owners.

3.4 Non-Functional Requirements

3.4.1 Performance

- **Response time:** the website should have an average response time of 200 milliseconds to 1 second.
- **Processing time:** Most operations should be processed in the time range of 10 milliseconds to 500 milliseconds.
- **Query time:** the query time for most common database queries should range from 5 milliseconds to 200 milliseconds.

3.4.2 Reliability

- The car rental platform must provide accurate and up-to-date information about the available car rental options. Any changes in the availability or details of the rental cars must be reflected in real time.
- The system must provide clear and transparent information about the rental cars and the booking process.
- Clear feedback must be provided to the customers in case of any errors or issues and alternative solutions or contact options should be offered.
- The system must ensure the integrity of the data. Any communication between the customer and the car rental provider must be securely stored and easily retrievable for future reference.
- It must also be able to recover quickly from any failures or errors

3.4.3 Availability

- As long as internet connection is secured, the car rental platform should be available 24 hours a day, 7 days a week.
- The system should aim for the highest possible uptime

3.4.4 Security

- The car rental platform should have a robust user authentication mechanism to prevent unauthorized access.
- Protection of sensitive data such as personal information must be ensured
- It should be ensured that users can only access and perform actions that are appropriate to their role (customer or car rental provider).

3.4.5 Maintainability

- The car rental platform must be implemented to be easily maintained and updated over time, following standard coding conventions and the best practices of software engineering and web development.
- Regular unit testing and final system testing should be conducted to ensure the system functions as expected.
- The system should be designed to be scalable to meet high user traffic (in the future) without requiring significant code changes.

3.4.5 Portability

- The car rental platform must be accessible and fully functional on various operating systems
- The website must be compatible with all major web browsers.
- It must also be designed to be responsive so as to adapt to different screen sizes.
- Easy data migration must be allowed, enabling users to transfer their data if they switch devices or platforms

3.5 Inverse Requirements

- The car rental platform must not complicate the rental process.
- The system should not provide an unfriendly user interface; the goal is not to provide a confusing user experience
- The system should not bring a narrow range of choices of car rental options to customers
- The system should not complicate the booking process.

3.6 Design Constraints

Currently we are only planning on launching the web app specifically in Ethiopia and thus, the following design constraints may be imposed based on this context:

1. Internet Connectivity: Ethiopia may have varying levels of internet connectivity, especially in rural areas. Design constraints may include optimizing the platform for slower internet connections and ensuring that essential features are accessible with limited bandwidth.

2. Localization: The platform should be designed to accommodate date formats, currency symbols, and cultural norms specific to the Ethiopian market.

3. Mobile-First Design: Given the prevalence of mobile phone usage in Ethiopia, the platform should have a mobile-responsive design and consider the limitations of mobile devices and networks.

4. Geographical Challenges: Design constraints should consider the reliability of location-based services because the platform provides closer rental options based on current location. And since the platform currently works only in Addis Ababa, the platform will show you rental options closer to your location filter.

5. Economic Factors: Economic factors and the cost of mobile data may impact user behavior. The platform should be optimized for cost-effective use, including data-saving features.

6. Cultural Sensitivity: The platform's design and content should be culturally sensitive and respectful of Ethiopian customs and norms.

7. Local Competition: The presence of local competitors and established car rental services may require the platform to adapt its features, and marketing strategies.

8. Infrastructure Challenges: Considerations for power outages, limited availability of certain hardware or technologies, and other infrastructure challenges may impact the platform's design and features.

9. Customer Support and Training: Given the potential variance in digital literacy and access to customer support, the platform may need to provide user-friendly interfaces and accessible customer support channels.

10. Budget and Resource Constraints: As university students undertaking this project, it's essential to acknowledge the significant constraints in terms of budget and resources. These limitations will substantially influence the choice of technologies, development time frames, and the overall scope of the project. Resource availability, including the number of student developers, their skill levels, and access to hardware and software, will be particularly restricted.
In addition, efficiency constraints may necessitate resource-efficient coding practices, minimizing system resource usage to reduce operational costs.

11. Scalability and Performance: Hardware limitations, including server capacity and network bandwidth, can constrain the system's scalability and performance. These limitations should be considered during the architectural design phase.

12. Third-Party Integrations: Integrations with third-party services (e.g., geo-location services) may have their own constraints and requirements, influencing the system's design and functionality.

3.7 Logical Database Requirements

The car rental platform will employ a database to manage and organize various aspects of the system, including user data, vehicle listings, past rentals, reviews, and authentication information. Specifically, a relational (SQL) database will be utilized to keep track of relationships and dependencies.

Logical Requirements

1. **Data Formats:**
    - User data will include personal information such as names, contact details, and authentication credentials.
    - Vehicle listings will contain detailed information about available cars, including make, model, year, pricing, availability, and related images.
    - Review data will comprise user ratings related to their rental experiences.

2. **Storage Capabilities:**
    - The database should have the capacity to store and manage growing volumes of user and vehicle data effectively.
    - It should support the storage of various media formats for images associated with vehicle listings and user profiles.

3. **Data Retention:**
    - User information and rental data should be retained as required for active user accounts and to maintain a history of past rental history.
    - Old, outdated, or inactive records might be subject to archival or periodic cleanup processes.

4. **Data Integrity:**
    - Ensuring data accuracy and consistency is vital. Robust constraints, validation rules, and foreign key relationships will be established to maintain the integrity of the stored information.
    - Regular data backups and a reliable disaster recovery plan will be in place to safeguard against data loss.

5. **Security Measures:** The database will adhere to security standards to protect sensitive user information. Secure authentication, and access control mechanisms will be employed to prevent unauthorized access or breaches.

3.8 Other Requirements

Since the system is a website, it will simply be deployed on a web server; and the end user would simply access the website through their web browser by entering the website's URL. (No packaging requirements.)

The end user accessing the website doesn't need any training, doesn't need to worry about the code-base, README file, or any scripts. The server hosting the website handles all the behind-the-scenes work, such as running the necessary scripts to deliver the website to the user's browser.

The legal requirement is also not applicable since the project is intended for educational purposes only at this point.