Zoomin Rentals

Thomas Andraos

Zumin Yu

Navya Sri Veerisetty

### 

### Table of Contents

[Table of Contents 2](#_Toc138362237)

[1. Introduction 3](#_Toc138362238)

[1.1. Purpose 3](#_Toc138362239)

[1.2. Project Scope 3](#_Toc138362240)

[1.3. Overview 3](#_Toc138362241)

[2. Target Market 4](#_Toc138362242)

[3. Design Methodology 4](#_Toc138362243)

[4. Development and Deployment Environments 5](#_Toc138362244)

[5. Tools 6](#_Toc138362245)

[6. Requirements Analysis 7](#_Toc138362246)

[6.1. Product Perspective 7](#_Toc138362247)

[6.1.1 System Interfaces 8](#_Toc138362248)

[6.1.2 User Interfaces 8](#_Toc138362249)

[6.1.3 Hardware Interfaces 8](#_Toc138362250)

[6.1.4 Software Interfaces 8](#_Toc138362251)

[6.1.4.1 External System Interfaces 8](#_Toc138362252)

[6.1.4.2 User Interface 8](#_Toc138362253)

[6.1.5 Operations 8](#_Toc138362254)

[6.2. Product Functions 9](#_Toc138362255)

[6.3. User Characteristics 9](#_Toc138362256)

[6.4. Constraints 10](#_Toc138362257)

[6.5. Assumptions and Dependencies 10](#_Toc138362258)

[7. Specific Requirements 11](#_Toc138362259)

[7.1 Functional Requirements 11](#_oq1d0y5wxjgc)

[7.2 Non- Functional Requirements 12](#_oq1d0y5wxjgc)

[7.3 Supplement Requirements 15](#_oq1d0y5wxjgc)

[8. Conclusion 16](#_oq1d0y5wxjgc)

## Introduction

### 1.1. Purpose

The purpose of this document is to determine and establish the specifications of the Zoomin Rental Service. It will detail both the functional and non-functional requirements of our system, design constraints, target market, design methodology, required tools, as well as the development and deployment environments. This document is intended for the design and development teams as well as the client.

### 1.2. Project Scope

The project scope involves developing a user-friendly web-based application for customer access to a car rental agency's services. The application will provide a login page for existing customers and a registration process for new users. Once logged in, customers will be able to search for available cars based on their preferred pick-up and drop-off dates. The search results will display relevant details such as car prices, number of seats, and make/model. Customers can then proceed to book their desired car for the selected dates. The application will facilitate the booking process and generate a unique reservation ID for each booking. Customers will have access to a dashboard where they can manage their bookings. The primary focus of the website is to provide a seamless and convenient experience for customers to search, book, and manage car rentals through an intuitive interface.

### 1.3. Overview

The remainder of this document is divided into the following sections:

• Target Market - Describes the intended audience for the system.

• Design Methodology - Defines the methodology we will be using throughout the project.

• Development and Deployment Environments - Outlines how we plan to develop and deploy the project.

• Tools - Outlines what tools will be required to develop and deploy the project.

• Requirements Analysis - Describes the general factors that affect the system and its requirements.

• Specific Requirements - Contains the software requirements that the system must meet.

### Target Market

The target market for the website that enables customers to access the services of a car rental agency includes individuals who are seeking car rental services for their transportation needs. This may encompass a wide range of potential customers, such as:

1. Travellers: Tourists or business travellers who require temporary transportation during their trips.
2. Local Residents: Individuals who need a car for personal use, such as running errands, attending events, or going on road trips.
3. Commuters: People who rely on rental cars for their daily commute or short-term transportation needs.
4. Event Participants: Attendees of conferences, weddings, or other special occasions where having a rental car is convenient.
5. Replacement Vehicle Seekers: Individuals who need a temporary car while their personal vehicle is undergoing repairs or maintenance.

The website aims to cater to customers of varying demographics, preferences, and purposes for renting a car. It strives to provide a user-friendly platform that meets the needs and expectations of these diverse customer segments, offering a seamless and enhanced car rental experience.

### Design Methodology

For the design and development of the website for customer access to a car rental agency's services, our group has chosen to adopt the Agile methodology. Agile methodology is a people-oriented approach that has gained recognition beyond the software field and is now widely used in various work environments. It emerged as a response to the limitations of the traditional Waterfall methodology and has become one of the most popular methods for software development projects.

The key aspect of Agile methodology is its iterative and evolutionary nature. It allows for continuous improvements with each iteration of the product being released. This methodology promotes a fast and adaptable development process that addresses pressing issues promptly. Its emphasis on constant communication facilitates the quick resolution of any development issues as they arise, fostering a collaborative environment.

The basic outline of the Agile process includes stages such as understanding, research, sketching, design, prototyping, testing, and refinement. To facilitate the Agile methodology, we will be utilising Jira as our primary work management tool. Jira, developed by Atlassian, is a robust tool that supports various use cases, including requirements and test case management, specifically designed for Agile software development.

We have chosen this methodology based on its widespread usage and effectiveness in the design process. According to a study by PwC (PriceWaterhouseCoopers), projects executed using Agile methodology have a nearly 30% higher success rate. By embracing Agile, we aim to efficiently and successfully complete the project, leveraging the strengths of this methodology to deliver the best possible outcome within our group's capabilities.

### Development and Deployment Environments

In the context of developing a website for customer access to a car rental agency's services, the choice of development and deployment environments plays a crucial role in ensuring a smooth and efficient workflow. To facilitate collaboration, version control, and deployment, our project will utilise GitHub as the primary development and deployment platform.

GitHub is a widely recognized and popular code hosting platform that provides powerful version control and collaboration features. It allows multiple developers to work on the project simultaneously, making it easy to manage and track changes to the source code. With GitHub, each team member can create their own branch to work on specific features or fixes, ensuring that their changes do not interfere with others' work.

Using GitHub's repositories, branches, and pull request mechanisms, developers can review each other's code, provide feedback, and merge changes into the main branch. This fosters effective collaboration, reduces conflicts, and maintains a unified and up-to-date codebase.

Furthermore, GitHub provides a seamless deployment workflow through its integration with various deployment services or platforms. Developers can set up automated deployment processes, allowing for efficient and reliable deployment of the car rental website to the desired hosting environment.

By utilising GitHub as our development and deployment environment, we can benefit from its robust version control capabilities, streamlined collaboration features, and simplified deployment processes. It enables our team to work cohesively, track changes effectively, and deploy the car rental website with confidence, ensuring a smooth development and deployment experience for the rental agency.

### Tools

The following tools will be incorporated into our project:

* Visual Studio Code: This integrated development environment (IDE) will be used for coding and developing the system. As we are creating a web application, we have chosen to work with a Blazer web assembly project.
* GitHub: We will leverage GitHub as our development platform, allowing all team members to access, contribute, and modify code. It provides version control and collaboration features, as mentioned in the "Development and Deployment Environments" section.
* Google Docs: As a collaborative tool, Google Docs will enable team members to collaborate on documentation work, making it easy to edit, review, and track changes in real-time.
* Jira: Jira serves as our primary work management tool, aiding us in tracking progress, managing tasks, and identifying what needs to be completed. It provides a comprehensive overview of the project's status and helps streamline project management.
* Figma: Figma will be utilised as our user interface design tool, allowing us to create prototypes and mockups of the website's user interface. This will assist in visualising the design and gathering feedback from stakeholders.
* Discord: Discord serves as our main communication platform, facilitating discussions about the project, coordinating tasks, and arranging Zoom meetings. It provides real-time communication and collaboration features for effective team interaction.
* STS (Spring Tool Suite): We will be utilising STS as our primary integrated development environment (IDE) specifically designed for developing Spring Framework-based applications. STS offers a range of features that support efficient Spring application development. It provides advanced coding assistance, such as code completion, refactoring, and code navigation, tailored for Spring projects. STS also integrates seamlessly with other Spring technologies like Spring Boot, allowing us to easily create, configure, and deploy Spring-based applications. With its comprehensive set of tools and plugins, STS enables efficient development, debugging, and testing of our application, ensuring a smooth development experience for our team.
* Slack: In addition to Discord, we will also utilise STS Slack as a communication channel to stay connected, share updates, and address any project-related queries or discussions.

By leveraging these tools, we aim to enhance collaboration, streamline development processes, and effectively manage the project's progress from start to deployment.

## Requirements Analysis

### 6.1. Product Perspective

The product will be implemented as a web-based application accessible by home computers, tablets, and smartphones.

The car rental tracking website is a new and independent system designed specifically for customer access. It aims to provide a user-friendly web-based platform where customers can conveniently search for available cars, make reservations, and manage their bookings.

The website includes a login page for existing customers and a registration process for new users. Once logged in, customers can specify their preferred pick-up and drop-off dates and browse through the available cars. The website displays essential information about each car, such as its price, number of seats, and make/model, allowing customers to make informed decisions.

Customers can proceed to book their desired car for the selected dates.The website also provides a dashboard where customers can manage their bookings.

The primary focus of the website is to provide a seamless and user-friendly experience for customers, allowing them to efficiently search, select, and book cars according to their preferences. The system aims to enhance the overall car rental process and ensure customer satisfaction through its intuitive interface and convenient booking management features.

### 6.1.1 System Interfaces

The goal is to streamline the car rental tracking process and enhance the overall efficiency and effectiveness of the agency's operations.The car rental website will serve as a web-based application accessible from various devices, including home computers, tablets, and smartphones. The website will provide a responsive interface for customers to access the car rental services. Customers can register for an account and log in through a login page. Upon logging in, they will have the option to choose their desired pick-up and drop-off dates. The website will display a list of available cars along with their respective prices, number of seats, and make/model. Customers can then proceed to book their preferred car based on the provided information. A database will securely store car and agency member information, while a web server will facilitate communication between the website and the database.

### 6.1.2 User Interfaces

The car rental website will be accessible through web browsers on various devices. A database will securely store user information, while a web server will facilitate communication between user devices and the database. The website will provide a responsive interface for agency members to manage car rental information, reservations, and bookings. Integration with external services may be utilised for additional functionalities.

### 6.1.3 Hardware Interfaces

The system must be operated via web browser on a personal computer, tablet and smartphone.

### 6.1.4 Software Interfaces

#### 6.1.4.1 External System Interfaces

The system can import all information from the database. The system can import all information to the database.

#### 6.1.4.2 User Interface

All customers can access all information from the database that their permission level allows. System is web based. System supports ASP.NET Web Development Framework. The focus will be on usability, efficiency, and accessibility.

### 6.1.5 Operations

There will be no requirements for a user to be able to operate the system with the exception of general computer skills to navigate the system. The database is only directly operable by the developers or qualified operators deemed by the developers

### 6.2. Product Functions

The primary functionality of the car rental system is to facilitate the rental process for customers. The system offers the following features:

1. Vehicle Search and Selection: Customers can easily search for available rental cars based on their location, desired dates, and specific preferences such as car type, size, and features. They can view detailed information about each vehicle, including rental rates, mileage, and any additional fees.
2. Online Booking: Customers can conveniently book their chosen rental car through the website. They can select the desired pick-up and drop-off locations, specify the rental duration, and review the total cost, including any applicable taxes or insurance options. The system provides a secure payment gateway for customers to complete their bookings online.
3. Booking Modification and Cancellation: Customers have the flexibility to modify their booking details, such as changing the pick-up or return dates, extending the rental duration, or upgrading to a different car model, subject to availability. They can also cancel their bookings if needed, with the option to receive a refund based on the cancellation policy.
4. Booking Confirmation and Receipts: Once the booking is completed, customers receive a confirmation email or notification with all the relevant details, including the reservation number, pick-up instructions, and contact information. They can also access and download their booking receipts from their account dashboard.
5. Feedback and Reviews: Customers have the opportunity to provide feedback and reviews about their rental experience through the system's website or app. They can rate the rental car and share their thoughts, helping other customers make informed decisions and providing valuable insights for the car rental agency.

By offering a seamless and user-friendly interface, flexible booking options, reliable customer support, and value-added features such as loyalty programs, the car rental system ensures a convenient and satisfying experience for customers.

### 6.3. User Characteristics

Customers: Customers accessing the car rental system are individuals or businesses looking to rent a vehicle. They may have varying levels of familiarity with the system and its features. Their main objective is to search for available cars, make reservations, and view their rental history.

### 6.4. Constraints

The system should enforce constraints to ensure that customers can only select rental cars that are available during their desired dates. This constraint prevents double-bookings and guarantees that customers can only reserve cars that are actually accessible.

1. Rental Duration Constraints: The system can impose constraints on the minimum and maximum rental duration allowed for each car. For example, some cars may have a minimum rental period of 24 hours, while others may have a maximum limit to prevent long-term bookings that exceed a certain threshold.
2. Age and License Requirements: The system can incorporate constraints to validate the customer's age and driving license details. For certain types of vehicles, there may be age restrictions or specific license requirements that customers must meet in order to rent the car. The system should verify and enforce these constraints during the booking process.
3. User Authentication and Access Constraints: The system should enforce constraints to ensure that customers can only access and modify their own bookings and personal information. Robust user authentication mechanisms, such as passwords or multi-factor authentication, should be in place to protect customer accounts from unauthorised access.
4. Data Retention Constraints: The system should comply with data retention policies and enforce constraints regarding the retention period for customer data. This ensures that customer information is not retained longer than necessary and is securely disposed of when no longer needed.
5. Development Framework: The system will be developed using the framework, providing a consistent and efficient development environment.
6. Database Backup and Restoration: The databases used in the system must support easy backup and restoration processes to ensure the safety and integrity of data.
7. Database Integrity: The databases should maintain integrity among different tables and their relationships, ensuring data consistency and preventing the loss or unnecessary storage of information.

These constraints ensure the security, stability, and reliability of the car rental system, providing a robust and efficient environment for users and data management.

### 6.5. Assumptions and Dependencies

The car rental system also relies on certain assumptions and dependencies:

User Access: It is assumed that all potential users of the system have access to a computer, tablet, or smartphone, as these devices are required to interact with the system.

Internet Connection: The system is dependent on an internet connection for users to access its features and functionalities. A stable and reliable internet connection is necessary for smooth operation.

### Specific Requirements

7.1. Functional Requirements

|  |
| --- |
| **FR\_01 Requirement Type: Functional** |
| Description: The system should provide users with a GUI to search and book rental cars. |
| Rationale: Users want to search for available rental cars and make bookings. |
| Source: Users |
| Fit Criterion: Users can search for rental cars and successfully book a car. |
| Dependencies: None |
| Rank of Importance: Critical |

|  |
| --- |
| **FR\_02 Requirement Type: Functional** |
| Description: Users should be able to view their booking history and current reservations. |
| Rationale: Users want to keep track of their past and current car rental bookings. |
| Source: Users |
| Fit Criterion: Users can view their booking history and current reservations. |
| Dependencies: None |
| Rank of Importance: Critical |

|  |
| --- |
| **FR\_03 Requirement Type: Functional** |
| Description: The system should provide a GUI for users to register an account. |
| Rationale: Users need to create an account to make car rental bookings. |
| Source: Users |
| Fit Criterion: Users can register an account successfully. |
| Dependencies: None |
| Rank of Importance: Critical |

|  |
| --- |
| **FR\_04 Requirement Type: Functional** |
| Description: Users should be able to log in to their accounts. |
| Rationale: Registered users need to log in to access their account information and make bookings. |
| Source: Users |
| Fit Criterion: Registered users can log in to their accounts. |
| Dependencies: User account exists |
| Rank of Importance: Critical |

|  |
| --- |
| **FR\_05 Requirement Type: Functional** |
| Description: The system should provide a GUI for users to manage their profile information. |
| Rationale: Users need to update their personal information and preferences. |
| Source: Users |
| Fit Criterion: Users can update their profile information successfully. |
| Dependencies: User account exists |
| Rank of Importance: Critical |

|  |
| --- |
| **FR\_06 Requirement Type: Functional** |
| Description: The system should allow administrators to manage car inventory and availability. |
| Rationale: Administrators need to add, update, and remove cars from the system. |
| Source: Administrators |
| Fit Criterion: Administrators can manage car inventory and availability effectively. |
| Dependencies: Administrator account exists |
| Rank of Importance: Critical |

7.2. Non-Functional Requirements

|  |
| --- |
| **NFR\_01 Requirement Type: Non-Functional** |
| Description: The website should have a user-friendly interface and require minimal training for users to operate. |
| Rationale: The website should be easy to navigate and use, even for users with limited technical knowledge. |
| Source: Users |
| Fit Criterion: Users can easily navigate and use the website without requiring extensive training. |
| Dependencies: None |
| Rank of Importance: Critical |

|  |
| --- |
| **NFR\_02 Requirement Type: Non-Functional** |
| Description: The website should have a fast response time, with pages loading within 10 seconds. |
| Rationale: Users expect a responsive and efficient website for a smooth user experience. |
| Source: Users |
| Fit Criterion: The website should load pages within 10 seconds to provide a seamless user experience. |
| Dependencies: None |
| Rank of Importance: Critical |

|  |
| --- |
| **NFR\_03 Requirement Type: Non-Functional** |
| Description: The database system should be capable of handling a large volume of car inventory and user data. |
| Rationale: The website needs to store and manage a significant amount of car inventory and user information. |
| Source: Users |
| Fit Criterion: The database can handle and store a large volume of car inventory and user data. |
| Dependencies: None |
| Rank of Importance: Critical |

|  |
| --- |
| **NFR\_04 Requirement Type: Non-Functional** |
| Description: The website should include a comprehensive help section or documentation, providing guidance on major operations and how to perform them. |
| Rationale: Users may require assistance or instructions for using specific features or functionalities. |
| Source: Users |
| Fit Criterion: The help section provides clear and detailed instructions on major operations within the website. |
| Dependencies: None |
| Rank of Importance: Critical |

7.3. Supplementary Requirements

|  |
| --- |
| **SR\_01 Requirement Type: Supplementary** |
| Description: The system should provide a GUI for users to create an account using their email address. |
| Rationale: Users need to create an account to access and use the car rental services. |
| Source: Users |
| Fit Criterion: Users can successfully create an account using their email address. |
| Dependencies: Users have a valid email address. |
| Rank of Importance: Critical |

|  |
| --- |
| **SR\_02 Requirement Type: Supplementary** |
| Description: The system should provide a GUI for users to log in to their respective profiles. |
| Rationale: Users need to log in to access their personalized account information and bookings. |
| Source: Users |
| Fit Criterion: Users can log in to their profiles through a login page. |
| Dependencies: Users have a registered account. |
| Rank of Importance: Critical |

8.Conclusion

This car rental website aims to provide users with a convenient and accessible platform to search, book, and manage car rentals. By allowing users to create accounts and log in, the website ensures a personalized experience for each user. These supplementary requirements prioritize user account management, enabling users to access and utilize the car rental services effectively.