CruzeIt

Software Requirements Specification *(SRS)*

Submitted by:

Barbosa, Lance Angelo G.

Chavez, Ayen Blaze G.

Cuanan, Gio Paulo A.

Narte, Noel Jr. P.

Valdez, John Mark R.

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the **CruzeIt Car Rental Booking System Website** is to provide a convenient, user-friendly platform for customers to browse available vehicles, make bookings, and track their reservations. It also offers administrators an efficient way to manage car listings, bookings, and related information.

1.2 Scope

The system is a web-based platform accessible via desktop and mobile browsers. The **CruzeIt** system is intended to:

- Provide customers with an online platform for viewing available cars, making reservations, and checking booking status.
- Enable administrators to manage vehicle inventory, update details, and approve or reject reservations.
- Maintain a secure database of all transactions and user data.
- Be responsive and functional on both desktop and mobile devices.

Key Features:

- User account registration and login.
- Car search and filter functionality.
- Booking system with start and end date selection.
- Booking history for customers.
- Administrative tools for managing listings and bookings.

1.3 Users

- **Customers:** General users aged 18 and above with internet access and a valid driver's license.
- **Admins:** Authorized staff responsible for managing cars and handling bookings.

2. Overall Description

2.1 User Interface

- Responsive web design for desktop and mobile.
- Navigation menu for quick access to features.
- Car listing page with search and filter options.
- Booking form with calendar date picker.
- Admin dashboard for car and booking management.

2.2 System Interface

- **Frontend**: Implemented using **React.js** to handle user interface design and client-side logic.
- **Backend**: Developed with **Node.js** and **Express.js** for server-side operations, routing, and API handling.
- **Database**: **MongoDB** is used for storing and managing data related to users, cars, and bookings.
- **Browser Support**: Compatible with major browsers including **Google Chrome**, **Mozilla Firefox**, **Microsoft Edge**, and **Safari**.
- **Payment Mockup**: Includes a simulated payment process to confirm bookings; no actual financial transactions are processed.

2.3 Constraints

- The system will work on desktop and mobile browsers.
- It will be made using HTML, CSS, JavaScript, and either Node.js or Django.
- The design should adjust automatically to fit different screen sizes.
- All data will be stored in a secure SQL database.
- Payment is only a mockup and will not handle real transactions.
- User passwords must be stored securely.

2.4 Assumptions

- Users have basic computer literacy.
- Admin will keep the car inventory updated.
- Database server will be running continuously.

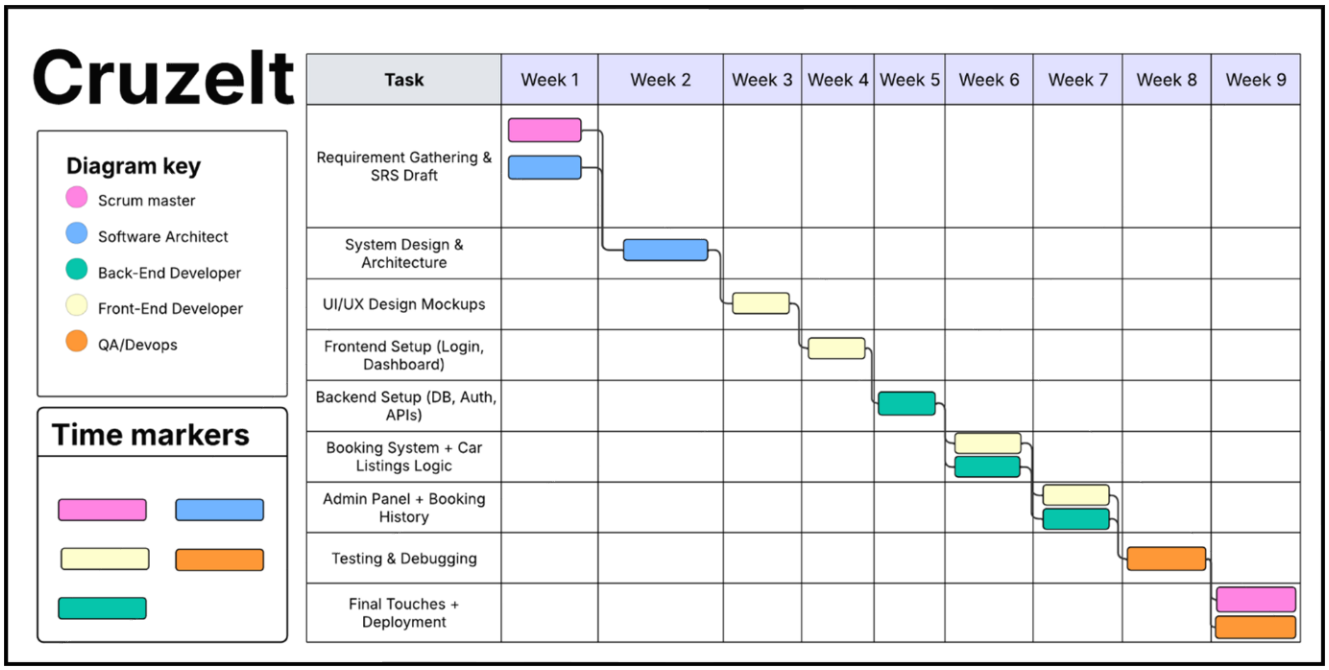
3. Functional Requirements

- User registration and login.
- View available cars with filters.
- Book a car for specific dates.
- Cancel or modify bookings before approval.
- Admin adds/updates/deletes car listings.
- Admin approves/rejects bookings.
- View booking history and status.

4. Non-Functional Requirements

- Page load time ≤ 3 seconds.
- Mobile-friendly and responsive design.
- Secure password storage with encryption.
- Support for at least 100 concurrent users.

5. Gantt Chart



6. Work Breakdown Structure