Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements for the development of a travel website with advanced booking features. This includes pre-filled booking forms, spontaneous ticket generation, order management in user profiles, PDF download capabilities, and advanced booking options.

1.2 Scope

This SRS covers the functional and non-functional requirements for the travel website, which allows users to book cars, manage their orders, and download booking confirmations in PDF format. It also includes admin functionalities to manage bookings and enable rebooking.

1.3 Definitions, Acronyms, and Abbreviations

• User: A registered member of the travel website.

• Admin: A user with administrative privileges.

• Booking: The process of reserving a car for a specified time.

• PDF: Portable Document Format.

• XAMPP: A free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

1.4 References

• UML Class Diagram for the database schema.

• Draw.io for visual representations.

2. Overall Description in Software Requirements:

Operating System : Windows

Technology : PHP

Web Technologies : Html, JavaScript, CSS

External web design: bootstrap, jquery

IDE : vs studio, intelliJ

Web Server : Apache

Database : MySql(MariaDB)

2.1 Product Perspective

The travel website is a standalone system that interacts with users and admins. Users can book cars, manage their orders, and download booking confirmations, while admins can manage bookings and enable rebooking. The system will be hosted locally using XAMPP.

2.2 Product Functions

• User registration and login.

• Car search and listing.

• Advanced booking options (initial booking, quick book, rebook).

• Pre-filled booking forms for returning users.

• Spontaneous ticket generation upon booking.

• Order management in the user profile.

• PDF download of booking confirmations.

• Admin management of bookings.

2.3 User Classes and Characteristics

• Users: Individuals who book cars through the website.

• Admins: Users who manage car bookings and user orders.

2.4 Operating Environment

The system will be web-based and accessible through modern web browsers. The backend will use PHP and MySQL, hosted locally on XAMPP.

2.5 Design and Implementation Constraints

• The system must ensure data security and privacy.

• The booking system should be robust and handle concurrent users efficiently.

• The website should be responsive and accessible on various devices.

• The system should be compatible with the XAMPP environment.

2.6 Assumptions and Dependencies

• Users have access to the internet and a web browser.

• The system will be hosted on a local server using XAMPP.

3. Functional Requirements

3.1 User Registration and Login

• FR1.1: Users shall be able to register with a username, email, password, and other personal details.

• FR1.2: Users shall receive an OTP for email verification.

• FR1.3: Users shall be able to log in using their email and password.

3.2 Car Search and Listing

• FR2.1: Users shall be able to search for cars based on location, category, and availability.

• FR2.2: The system shall display car details, including name, seating capacity, options (AC/non-AC), and price.

3.3 Booking Functionality

• FR3.1: Users shall be able to make an initial booking by filling out a booking form.

• FR3.2: Users shall be able to use a quick book option for faster bookings.

• FR3.3: The booking form shall be pre-filled with user information for returning users.

• FR3.4: Upon booking, a ticket shall be generated spontaneously and visible in the user's profile.

3.4 Order Management

• FR4.1: Users shall be able to view their bookings in the orders tab of their profile.

• FR4.2: Users shall be able to download booking confirmations as PDFs.

• FR4.3: Users shall be able to rebook a car if the admin marks an order as complete.

3.5 Admin Management

• FR5.1: Admins shall be able to view and manage all bookings.

• FR5.2: Admins shall be able to mark bookings as complete.

• FR5.3: Admins shall be able to enable rebooking for completed orders.

4. Non-Functional Requirements

4.1 Performance Requirements

• NFR1.1: The system shall handle up to 1000 concurrent users.

• NFR1.2: Booking confirmation generation shall take no more than 2 seconds.

4.2 Security Requirements

• NFR2.1: User passwords shall be stored securely using encryption.

• NFR2.2: All user data transmissions shall be encrypted using HTTPS.

4.3 Usability Requirements

• NFR3.1: The user interface shall be intuitive and easy to navigate.

• NFR3.2: The website shall be responsive and accessible on mobile devices.

4.4 Maintainability Requirements

• NFR4.1: The system shall be modular to facilitate easy updates and maintenance.

• NFR4.2: The codebase shall be documented for future developers.

5. System Features

5.1 Pre-filled Booking Form

• Description: Returning users will see their booking forms pre-filled with their saved information, expediting the booking process.

5.2 Spontaneous Ticket Generation

• Description: Upon booking, a ticket will be generated and displayed in the user's profile, available for immediate download.

5.3 Advanced Booking Options

• Description: Users can choose between initial booking, quick book, and rebook options for a streamlined booking experience.

5.4 Order Management in User Profile

• Description: Users can manage their bookings from their profile, view order status, and download confirmations as PDFs.

5.5 Admin Booking Management

• Description: Admins can manage all bookings, mark them as complete, and enable rebooking for users.

6. UML Diagrams

6.1 Use Case Diagram

(Figure: Use Case Diagram)

[Use Case Diagram](https://example.com/use\_case\_diagram.png)

6.2 Class Diagram

(Figure: Class Diagram)

[Class Diagram](https://example.com/class\_diagram.png)

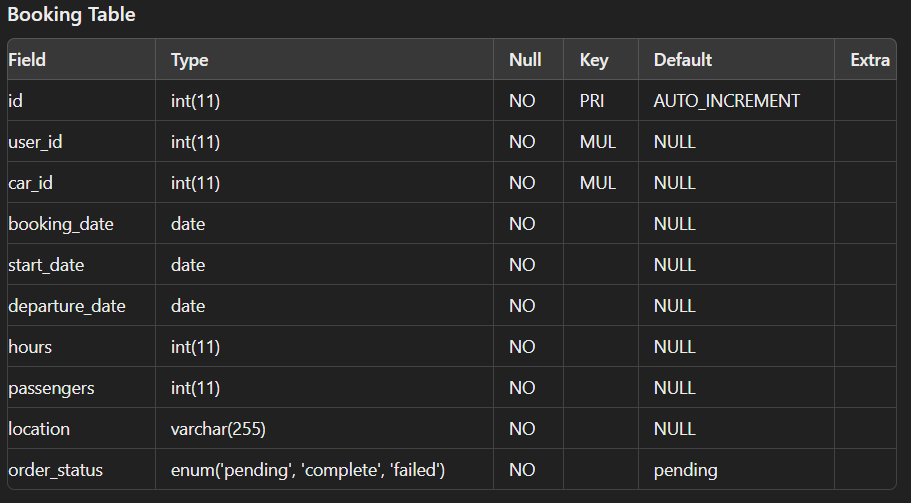
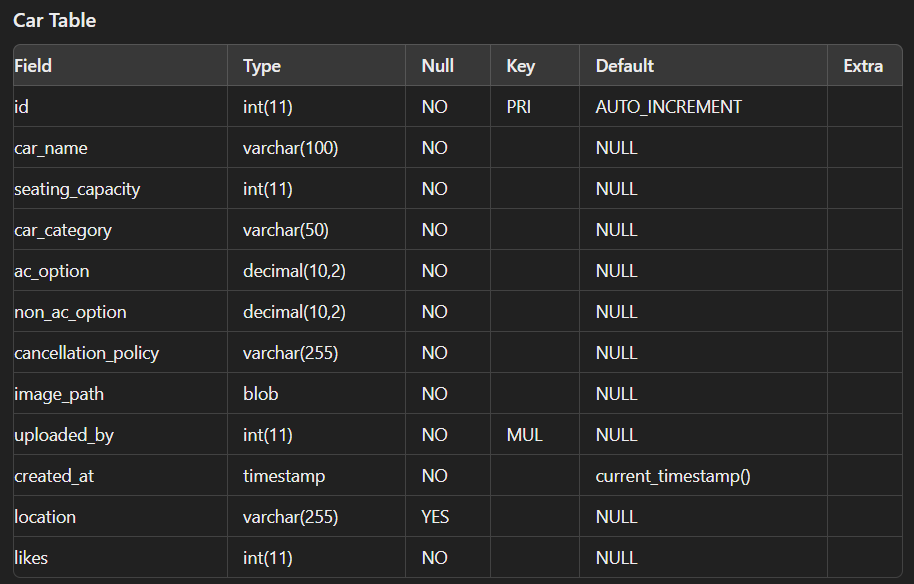
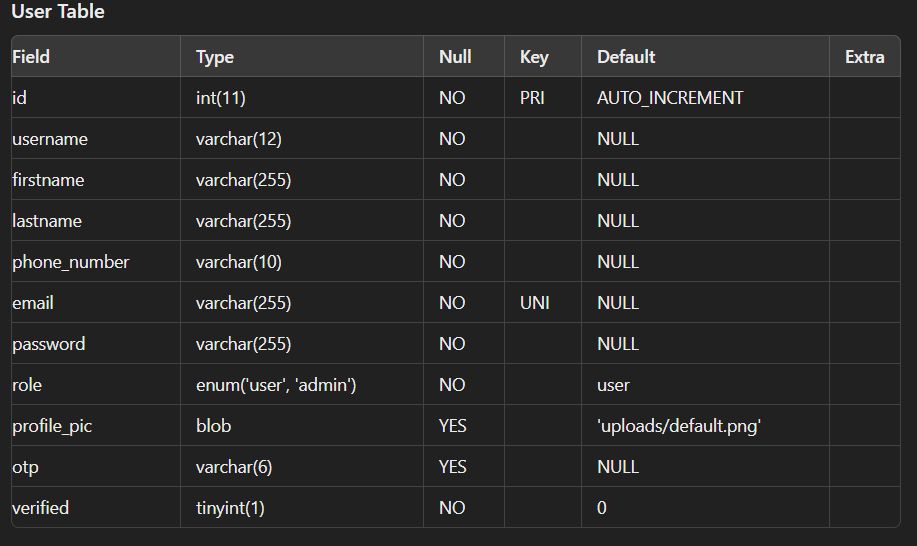
6.3 Sequence Diagram

(Figure: Sequence Diagram)

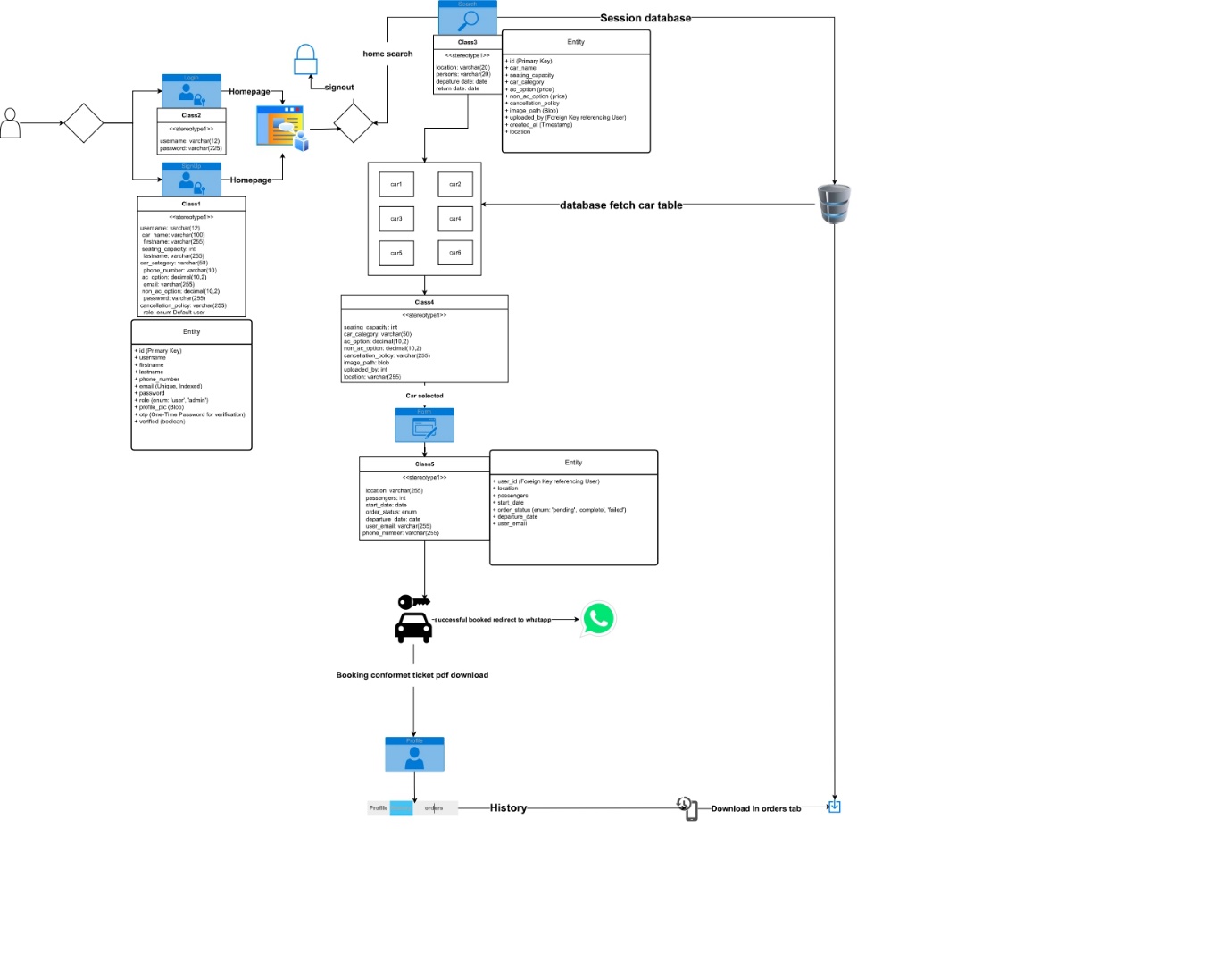
[Sequence Diagram](https://example.com/sequence\_diagram.png)

7. Appendix

7.1 Database Schema



User Usecase:



Admin usecase:

