SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT

(SRS DOCUMENT)

FOR

GUEST HOUSE BOOKING SYSTEM WEB APPLICATION

(NALANDA INTERNATIONAL FRIENDSHIP ASSOCIATION)

Version 1.0

By

Group E

Member 1 - S19482

Member 2 - S19576

Member 3 - S19918

Supervisor

Mr. Isuru Madugalla

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to outline the detailed requirements for the development of the Guest House Booking System. This system aims to provide a comprehensive online platform that facilitates the booking of accommodations at a specific guest house, along with additional features such as vehicle selection for traveling to local area attractions.

1.2 Scope

The "Guest House Booking System" is a simplified solution mirroring the core concepts of an online reservation platform. In some areas, Information about available accommodations may be scarce or difficult to obtain. A booking system provides a centralized platform where guests can easily access information about nearby guest houses, their amenities, and availability. In addition to that, we can identify some of problem like shown below.

- Lack of Exposure
- Inefficient Booking Process
- Communication Barriers
- Information cannot exchange the correctly and less time.
- Difficulty in generating reports for analysis

To address these challenges, the Guest House Booking System will provide a centralized service and database for reservation management, guest details, and booking history. It will improve data retrieval efficiency, reduce transaction time and paper usage, enhance data security, and provide tools for efficient reporting and analysis. Implementing the system will revolutionize the Guest House Booking operations, leading to better service quality and a more organized and productive environment. Furthermore, the system will include user friendly portals for the customer to book/cancel appointments times of their own choosing and the system will feature

user-friendly portals allowing customers to book, modify, or cancel taxi services at their convenience. According to the feedback option guests can suggest some additional and flexible operation for the guest house booking system. Developed on a relational database, the system efficiently manages bookings and guest house information. Our database server accommodates guest house across the designated city. Our primary goal is to ensure customers enjoy a seamless experience while securing their bookings at reasonable rates. And also, this project is major effect to the development of the guest house and gains the more benefits. The system facilitates the following functionalities.

Spring boot frame work is used for backend development and combining SQL for data management, and React for frontend user interfaces, we hope to create a robust, scalable, and user-friendly application.

Core Functionality:

- Allow users to browse and search for a guest house based on location, dates, amenities, and pricing.
- Provide detailed property listings with descriptions, photos, availability calendars, and booking options.
- Enable users to make, modify, and cancel bookings directly through the application.
- Support secure online payments for guest house bookings, integrating with payment gateways to process transactions.
- Implement a review and rating system to collect feedback from guests and maintain service quality.

Additional Features:

- Vehicle Selection: Allow users to select vehicles for traveling to local area attractions.
- Offer an immersive 360-degree view of the guest house.
- Local Area Information: Provide real-time information about local attractions, points of interest, and recommended activities.
- Special Offers and Packages: Present users with exclusive deals, discounts, and packages to enhance their booking experience.

- Multi-language Support: Offer support for multiple languages to accommodate users from diverse linguistic backgrounds.

Non-Functional Features:

- Responsive Design: Ensures that the web application is accessible and functional across various devices and screen sizes, enhancing user experience.
- Website Speed: Optimizes website performance to reduce loading times and improve user engagement and satisfaction.

A guest house booking website brings a plethora of benefits to both owners and guests. For owners, it offers heightened visibility, simplifies booking procedures, and grants round-the-clock accessibility, resulting in increased guest bookings. The website's automation minimizes administrative tasks. Integrated payment processing guarantees secure transactions, while the platform's marketing capabilities and data analytics fuel business expansion and elevate guest contentment.

1.3 Product perspective

1.3.1 System interfaces

The System Interface of the guest house booking system encompasses various functionalities tailored to provide users with a seamless experience. This includes User Registration and Login mechanisms, facilitating user account creation and authentication. while the Booking Interface guides them through selecting and confirming reservations with detailed room information and interaction elements. Within the User Dashboard, users can manage their bookings and profile information effortlessly. The Payment Interface ensures secure transactions, offering various payment methods and associated security measures. Cancellation Interface allows users to initiate cancellations, adhering to respective policies. Furthermore, Feedback and Support mechanisms enable users to provide input or seek assistance easily like chat bot. Responsive Design principles ensure adaptability across devices, enhancing accessibility and usability. This application runs on the latest version of Chrome or Firefox browser on Windows.

1.3.2 User interfaces

The user interface of the system consists of web pages accessible via standard web browsers. It provides intuitive navigation and interaction options for users to browse, search, and book guest houses, as well as manage their bookings and accounts.

1.3.3 Hardware interfaces

The outlines the logical properties of each interaction between the software product and the hardware components in terms of the hardware requirements. It outlines the hardware specifications needed for the software to function, such as memory limits, cache size, processor, RAM capacity, etc.

Minimum Hardware Requirements

- Processor intel CORE i3
- Hard disk drive 128 GB
- RAM 4GB
- Preferred Hardware Requirements

Other Hardware

- Mobile Devices, Tablets, Desktop and etc.
- Network bandwidth, server reliability, and security measures.

1.3.4 Software interfaces

The software interface facilitates seamless communication between distinct modules or subsystems within the system. This interface ensures enabling components to exchange information efficiently and securely. Specifications within this interface encompass message formats, communication and error handling mechanisms. The product will be operating in a Windows environment. The Guest House Booking System is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. The only requirement to use this online product would be the internet connection.

1.3.5 Communications interfaces

The system communicates with users via standard web protocols such as HTTP and HTTPS. Email and mobile notifications will be used to provide updates to users.

1.3.6 Memory constraints

Memory constraints for the system are minimal and primarily depend on the hosting environment. The system will be designed to efficiently utilize memory resources to ensure optimal performance.

1.3.7 Operations

The system will support standard operations such as browsing, searching, booking, modifying, and canceling guest house bookings. It also includes administrative operations for managing guest house listings and user accounts.

1.3.8 Site Adaptation Requirements

The system is adaptable to different hosting environments and can be deployed on various web hosting platforms. It is designed to scale based on the volume of users and bookings.

1.3.9 Interfaces with Services

Integration with external services includes payment gateways for processing online payments securely and APIs for accessing real-time local area information and additional functionalities.

1.4 Product functions

Product Function	Description
Search	• Search functionality where users can input their desired check-in and check-out dates, number of guests, and any other preferences relevant to the specific guest house. • Upon submission, display available rooms or accommodations that match the entered criteria for the particular guest house.
Selection	• Once users have searched for availability, present a list of available rooms or accommodations within the chosen guest house. • Provide detailed information about each room, including amenities, pricing, and any special features.
Booking	• A unique booking ID to each reservation made for the guest house. • Capture the check-in and check-out dates and times for the guest's stay. • Calculate the total cost of the booking, including room charges, taxes, and any additional fees specific to the guest house.
Review	• Allow users to review their booking details before finalizing the reservation. • Offer the option to make any necessary adjustments or corrections to the booking details.
Travelers Information	• Request information for each guest who will be staying in the chosen guest house, including their name, phone number, and email address.
Taxi booking	• Integrating taxi booking on the website enhances guest convenience, streamlines travel planning, and boosts revenue.

Payment	• Support various payment methods, ensuring compliance with security standards and regulations.
Cancellation	• Provide a straightforward process for users to initiate cancellations, adhering to any applicable cancellation policies or procedures specific to the guest house.
Vehicle Selection	• Users shall have the option to select additional services such as transportation arrangements to and from the guest house. • Vehicle options shall include types of vehicles available, rental prices, and pick-up/drop-off locations.

1.5 User characteristics

- Education level: The system's users must at the very least be fluent in English.
- Technical proficiency: The user should feel at ease utilizing general- purpose computer software

1.6 Limitations

The system may have limitations in terms of the number of concurrent users it can handle and the availability of real-time data for local area information.

1.7 Assumptions and Dependencies

The system assumes that users have access to a stable internet connection and compatible devices for accessing the web application. It may depend on external APIs for certain functionalities such as payment processing and local area information. In addition to that, we can consider the following assumption also,

- The Booking System is running 24 hours a day.
- The system should be user-friendly so that it is easy to use for the users.
- The information of all users, rooms details, booking details and other details must be stored in a database that is accessible by the website
- The system should have more storage capacity and provide fast access to the database.

- Users may access from any devices such as (Laptop and Desktop Computers, Mobile Devices, Tablet, Smart Phones and etc.) that has Internet browsing capabilities and an Internet connection
- The system should provide a search facility and support quick transactions.
- The coding should be error free
- Users must have their correct usernames and passwords to enter into their online accounts and do actions

Dependencies

- The end users (Admin) should have proper understanding of the product.
- The information of users must be stored in a database.
- Any update regarding the details of the guest house is to be recorded to the database and the data entered should be correct

- The Guest House Booking System leverages Code, Frameworks, and Database to facilitate seamless booking experiences

- Ensuring that users have the correct permissions to access certain resources or perform specific actions.

- APIs and External Services.

- Implementing backup and recovery procedures to safeguard against data loss.

- Implementing security measures to protect against common web vulnerabilities

1.8 Definitions

Guest House Booking System

The web application designed to facilitate the booking of guest houses.

Relational Database Server

The server used to store and manage guest house information and bookings.

1.9 Acronyms and abbreviations

SRS: Software Requirements Specification

API: Application Programming Interface

UI: User Interface

2. Requirements

2.1 External interfaces

External interface	Description
Payment Gateway Integration	The booking platform integrates with the Stripe payment gateway API to enable credit card payments for room reservations.
Mapping Service API	The website utilizes the Google Maps API to show interactive maps of the guest house location and nearby points of interest.
Review Aggregator Integration	Guest reviews and ratings from the TripAdvisor API are displayed on the booking platform to build trust and credibility.
Email Notification Service	The booking system uses the SendGrid API to send automated email notifications for booking confirmations, check-in details, and special offers.

2.2 Functions

Actors	Admin - The owner of the guest house.
	Manager - The one who is added by the admin and participates in management tasks when required.
	Driver - The one who provides taxi service to the guests.
	Guest - The one who can make bookings and get other services.

As an Admin

User Type	Epic	Use case	Expected Outcome
Admin	Admin Registration	Create new account	Success: Account created Failure: Account creation failed (e.g., due to invalid information, existing email, etc.)
		Login	Success: Logged in Failure: Invalid credentials (username/password), account locked, etc.
		Logout	Success: Logged out Failure: N/A (assuming logging out is always successful)
		Forgot Password	Success: Password reset email sent Failure: Email not found, error sending email, etc.
		Edit Account	Success: Account information Updated Failure: Error updating information (e.g., invalid data), account not found, etc.

Admin	Property Listings	Add room	Success: Room added Failure: Error adding room
		Delete room	Success: Room deleted Failure: Error deleting room
		Update room	Success: Room updated Failure: Error updating room
Admin	Booking Management	View detailed information about individual bookings	Success: Booking details displayed Failure: Error retrieving booking details
		Approve booking	Success: Booking approved Failure: Error approving booking
		Cancel booking	Success: Booking canceled Failure: Error canceling booking
Admin	Review System	Manage reviews	Success: Reviews managed Failure: Error managing reviews
Admin	360-Degree View	Modify view	Success: View modified Failure: Error modifying view
Admin	Local Area Information	Add location	Success: Location added Failure: Error adding location
		Delete location	Success: Location deleted Failure: Error deleting location
		Update location	Success: Location updated Failure: Error updating location

Admin	Special Offers	Add offer	Success: Offer added Failure: Error adding offer
		Delete offer	Success: Offer deleted Failure: Error deleting offer
		Update offer	Success: Offer updated Failure: Error updating offer
Admin	Multi-language Support	Add language	Success: Language added Failure: Error adding language
		Delete language	Success: Language deleted Failure: Error deleting language
Admin	Manager Registration	Add new manager	Success: Manager added Failure: Error adding manager
		Delete manager	Success: Manager deleted Failure: Error deleting manager
		Update manager	Success: Manager updated Failure: Error updating manager
Admin	Driver Registration	Add driver	Success: Driver added Failure: Error adding driver
		Delete driver	Success: Driver deleted Failure: Error deleting driver
		Update driver	Success: Driver updated Failure: Error updating driver

Admin	Inquiries and Assistance Option / FAQ section	View inquiries	Success: Inquiries viewed Failure: Error viewing inquiries
		Write Feedback	Success: Feedback written Failure: Error writing feedback
Admin	Analytic Report	View Report	Success: Report viewed Failure: Error viewing report
		Delete Report	Success: Report deleted Failure: Error deleting report
		Modify Report	Success: Report modified Failure: Error modifying report

As a Manager

User Type	Epic	Use case	Expected Outcome
Manager	Manager Registration	Login	Success: Logged in Failure: Invalid credentials, account locked, etc.
		Logout	Success: Logged out Failure: N/A (assuming logging out is always successful)
		Forgot Password	Success: Password reset email sent Failure: Email not found, error sending email, etc.

		Update account	Success: Account updated Failure: Error updating account (e.g., invalid data)
Manager	Property Listings	Update room	Success: Room updated Failure: Error updating room (e.g., room not found)
Manager	Booking Management	Cancel booking	Success: Booking canceled Failure: Error canceling booking (e.g., booking not found)
Manager	Review System	Manage reviews	Success: Reviews managed Failure: Error managing reviews (e.g., review not found)
Manager	Local Area Information	Update location	Success: Location updated Failure: Error updating location (e.g., location not found)
Manager	Special Offers	Add offer	Success: Offer added Failure: Error adding offer (e.g., duplicate offer)
		Delete offer	Success: Offer deleted Failure: Error deleting offer (e.g., offer not found)
		Update offer	Success: Offer updated Failure: Error updating offer (e.g. offer not found)

Manager	Inquiries and Assistance Option/ FAQ section	View inquiries	Success: Inquiries viewed Failure: Error viewing inquiries (e.g., no inquiries found)
		Write Feedback	Success: Feedback written Failure: Error writing feedback (e.g., invalid input)

As a Driver

User Type	Epic	Use case	Expected Outcome
Driver	Driver Registration	Login	Success: Logged in Failure: Invalid credentials, account locked, etc.
		Logout	Success: Logged out Failure: N/A (assuming logging out is always successful)
		Forgot Password	Success: Password reset email sent Failure: Email not found, error sending email, etc.
		Update account	Success: Account updated Failure: Error updating account (e.g., invalid data)
Driver	Vehicle Registration	Add new Vehicle	Success: Vehicle added Failure: Error adding vehicle (e.g. invalid information)

		Delete Vehicle	Success: Vehicle deleted Failure: Error deleting vehicle (e.g., vehicle not found)
		Update Vehicle information	Success: Vehicle information updated Failure: Error updating vehicle information (e.g., vehicle not found)
Driver	Service	View ride information	Success: Ride information viewed Failure: Error viewing ride information (e.g., no rides found)
		Accept Ride	Success: Ride accepted Failure: Error accepting ride (e.g., ride not available)
		Reject Ride	Success: Ride rejected Failure: Error rejecting ride (e.g., ride not available)
Driver	Review System	View ratings	Success: Ratings viewed Failure: Error viewing ratings (e.g., no ratings found)

As a Guest

User Type	Epic	Use case	Expected Outcome
Guest	Guest Registration	Create Account	Success: Account created Failure: Error creating account (e.g., invalid data)
		Update Account	Success: Account updated Failure: Error updating account (e.g., invalid data)
		Delete Account	Success: Account deleted Failure: Error deleting account (e.g., account not found)
Guest	Property Listings	View details of rooms	Success: Room details viewed Failure: Error viewing room details (e.g., no rooms found)
Guest	Check Availability	Add check-in	Success: Check-in added Failure: Error adding check-in (e.g., invalid date)
		Add check-out	Success: Check-out added Failure: Error adding check-out (e.g., invalid date)
Guest	Booking Management	Make Booking	Success: Booking made Failure: Error making a booking (e.g., room unavailable)

		Cancel Booking	Success: Booking canceled Failure: Error canceling booking (e.g., booking not found)
		Update Booking	Success: Booking updated Failure: Error updating booking (e.g., booking not found)
Guest	Payment Processing	Make payment	Success: Payment made Failure: Error making payment (e.g., insufficient funds)
		Request refund	Success: Refund requested Failure: Error requesting refund (e.g., refund not available)
Guest	Special Offers	View special offers	Success: Special offers viewed Failure: Error viewing special offers (e.g., no offers available)
Guest	Review and Rating System	View reviews	Success: Reviews viewed Failure: Error viewing reviews (e.g. no reviews available)
		View ratings	Success: Ratings viewed Failure: Error viewing ratings (e.g. no ratings available)
		Submit new review	Success: Review submitted Failure: Error submitting review (e.g., invalid data)
		Add new rating	Success: Rating added Failure: Error adding rating (e.g., invalid rating value)

Guest	Local Area Information	View local area information	Success: Local area information viewed Failure: Error viewing local area information (e.g., no information available)
Guest	Taxi Service	View available vehicles	Success: Available vehicles viewed Failure: Error viewing available vehicles (e.g., no vehicles available)
		Request ride	Success: Ride requested Failure: Error requesting ride (e.g., no drivers available)
Guest	Multi-language Support	Change language	Success: Language changed Failure: Error changing language (e.g., language not supported)
Guest	Inquiries and Assistance Option / FAQ section	View FAQ section	Success: FAQ section viewed Failure: Error viewing FAQ section (e.g., no questions found)
		Add new question	Success: Question added Failure: Error adding question (e.g. invalid question)

2.3 Usability requirements

The users of the system are foreign and local customers, taxi service drivers and staff, even those with limited technical expertise, and clients. The members are assumed to have basic knowledge of computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and be able to rectify small problems that may arise and maintain the system. The proper user interface, user manual, online help, and guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

User Interface (UI) Design	The UI features easy-to-understand menus, prominent call-to-action buttons, and consistent color schemes for visual coherence. font color, font and font sizes Also, the homepage includes a prominent search bar (search rooms, available rooms, search vehicles), top navigation menu, and visually appealing graphics.
Responsive Design	The website is optimized for desktops, tablets, and smartphones, with a layout that adapts based on screen size. It includes mobile-friendly elements like collapsible menus, touch-friendly buttons, and legible text sizes.
Efficient Search & Filtering	Users can search for rooms by date range, find available rooms efficiently, available taxi services, filter by room type, amenities, and price range, and view results effectively and efficiently. The search includes date pickers, dropdown menus, checkboxes, and sliders for filtering.
User Guidance & Help Features	Contextual tooltips, on-screen instructions, and help buttons guide users through booking, explain terms, and provide troubleshooting tips.
Accessible Design	offering keyboard-friendly navigation, alt text for images, and ARIA landmarks for screen readers. Users can navigate using keyboard shortcuts, and images have descriptive alt text.
Error Handling & Feedback	Displays specific error messages, highlights fields with issues, and offers suggestions for correction during the booking process.

Efficient Checkout Process	One-page checkout with clear steps and multiple payment options (credit card, PayPal, etc.). The checkout page includes a step-by-step progress bar (Review Details, Enter Payment, Confirm Booking) to guide users through the process.
Feedback & User Testing	Conducts regular user testing, collects feedback through surveys/forms, and analyzes user behavior data to enhance usability and satisfaction. All the users of the web site can take clear information using feedback option and review system.

2.4 Performance requirements

Guest house Booking system careful consideration of performance requirements to ensure that the software operates efficiently and effectively. The system should be the handle the number of transaction and system should be fast and accurate. In addition to that the System shall handle expected and unexpected errors in ways that prevent loss in information and long downtime periods. Thus, it should have inbuilt error testing to identify invalid username/password (driver details). We can consider the following things as performance requirement.

Response Time	System should be the responsive and provide quick feedback to users. Setting with the automative answer for the guest's questions (eg. chatbot). Maximum response time of 2 seconds for loading pages, processing requests, and displaying search results it will affect to the users.
Concurrency and Load Handling	Supports concurrent user sessions and handles a load of up to 500 simultaneous users without performance degradation.
Database Performance	Database queries related to room availability, driver details, pricing, and booking operations should execute within 100 milliseconds on average for real-time updates. The system shall store in the database no more than one million Transactions. If the database grows over this limit, old transactions shall be backed up and deleted from the operational database.

Scalability	Scalable to handle a 20% increase in user traffic and booking transactions, ensuring uninterrupted service and performance.
Caching and Optimization	Implements caching mechanisms and optimization techniques to reduce page load times and improve overall performance. Static resources (CSS, JavaScript) should be cached locally on the user's device, and server-side caching should be implemented for frequently accessed data to enhance performance.
Usability	The system's user interface should be intuitive and easy to use, allowing users to perform tasks quickly and without confusion. A well-designed interface can enhance the perception of system performance.
Availability and Uptime	Minimum uptime of 99.9% per month, with scheduled maintenance and downtime communicated to users through notifications.
Network Latency	Network latency for data transmission between the client and server should be less than 100 milliseconds under typical network conditions.
Error Handling and Recovery	Handles errors gracefully with error recovery mechanisms to minimize downtime and data loss.

2.5 Logical Database Requirements

Our comprehensive guest house booking system database seamlessly manages reservations, guest information, integrated taxi services, property details, review system, and user accounts. Core tables include 'Guests' for guest details, 'Rooms' for room information, 'Bookings' linking guests to rooms, 'RoomTypes' defining room categories, and 'TaxiServices' for taxi details. The addition of 'PropertyDetails' table captures essential property information. A 'ReviewSystem' table facilitates guest feedback. User accounts are created for administrators and managers, with separate tables for 'AdminAccounts' and 'ManagerAccounts', ensuring secure access and management privileges. Foreign key constraints maintain data integrity, guaranteeing accurate references throughout the database. This integrated solution offers a seamless guest experience,

from booking to departure, while empowering administrators and managers with efficient control over operations. Furthermore, following things also can be consider,

Data Structure	The database should have tables to store information about rooms (room details, availability), bookings (reservation details, guest information), users (user profiles, authentication), payments (transaction records), and reviews (guest feedback).
Guest bookings	Information such as guest details (name, contact information), booking dates, room type, and special requests.
Taxi service	Details including driver information, vehicle details, availability, and booking information (pickup location, drop-off location, pickup time).
Admin and driver accounts	User credentials, permissions, and profile details.
Reporting and feedback	Data related to guest reviews, feedback, and performance reports.
Data Integrity	Data in the database should maintain referential integrity, ensuring consistency and accuracy of relationships between related tables. Ensure that guests cannot book the same room or taxi service simultaneously. Maintain relationships between bookings, guests, rooms, and taxi services. Validate input data to prevent errors and inconsistencies.
Normalization	The database should be normalized to reduce data redundancy and ensure efficient storage and retrieval.

Indexing	Indexes should be implemented on frequently queried columns (e.g., room types, availability dates) to optimize search and retrieval performance.
Data Security	Access to sensitive data (e.g., payment information, user passwords) in the database should be restricted based on user roles and permissions.
Backup and Recovery	Regular database backups should be performed to prevent data loss and ensure recovery in case of system failures or disasters.
Concurrency Control	The database should use locking mechanisms and transaction isolation levels to prevent data inconsistencies and ensure data integrity during concurrent user operations.
Data Migration and Versioning	The database should have documented procedures for migrating data between versions, applying schema changes without data loss, and maintaining data integrity across upgrades.
Frequency of Use	Guest bookings and taxi service data may have high access frequencies, especially during peak booking periods or busy travel times. Admin and driver account data might have lower access frequencies but still require frequent updates for account management and permissions.
Accessing Capabilities	Users (guests, admins, drivers) need read and write access to their respective data. Admins require additional permissions for account management, reporting, and feedback moderation.

Data Entities and Their Relationships	Entities include guests, rooms, bookings, taxi services, admins, drivers, and feedback/reviews. Relationships involve guests booking rooms, guests booking taxi services, admins managing accounts, drivers providing taxi services, and feedback/reviews associated with bookings.

2.6 Design constraints

Technology Stack	The application must be developed using specific technologies or frameworks (e.g., LAMP stack, MEAN stack) due to compatibility or organizational requirements.
Regulatory Compliance	The system must comply with legal regulations and industry standards (e.g., GDPR for data privacy, PCI DSS for payment security) related to user data handling and security.
Performance Requirements	The system must meet specific performance targets (e.g., response times, scalability) based on business needs and user expectations. The application's design must ensure that page load times are within 3 seconds, and the system can handle a peak user load of 1000 concurrent users without performance degradation.
Budgetary Constraints	The development of the application must stay within the allocated budget, which restricts the use of expensive third-party services or extensive customization.

Compatibility Requirements	The application's design must ensure compatibility with popular web browsers (Chrome, Firefox, Safari) and mobile devices (iOS, Android) to reach a wide audience.
Resource Availability	Limited availability of development resources (skilled personnel, hardware infrastructure) may impact project timelines and feature implementation.
Third-Party Integration Limitations	The design must consider the API rate limits of third-party services for real-time data updates, which may require caching strategies or asynchronous processing.
Scalability and Future Expansion	The application's architecture must support horizontal scalability to add more servers and handle increased user traffic or new features in future versions.
Time constraint	Project dead

2.7 Standards compliance

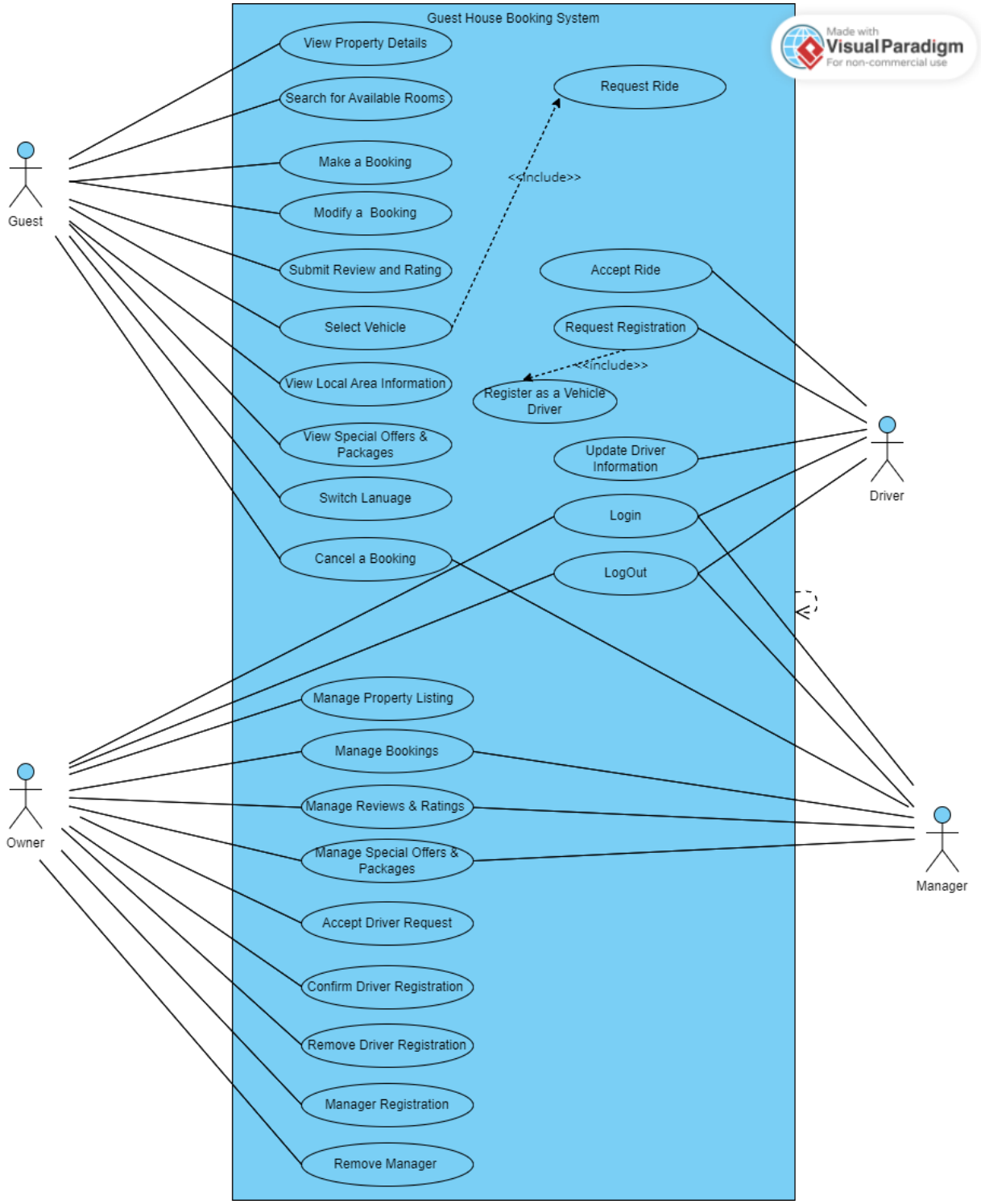
Aspects	Description
Security Standards	Complies with Payment Card Industry Data Security Standard (PCI DSS) for handling credit card information securely during online transactions.
Data Privacy	Adheres to GDPR principles by obtaining user consent, providing data access controls, and implementing data encryption for sensitive information.
Accessibility	Follows WCAG accessibility standards for keyboard-friendly navigation, ARIA labels, and contrast ratios for text readability.
Interoperability	Ensures APIs comply with industry standards for seamless integration with third-party services, applications, and data sources.
Web Browser Compatibility	Compatible with major web browsers (Chrome, Firefox, Safari, Edge) and adheres to World Wide Web Consortium (W3C) standards for HTML, CSS, and JavaScript.
Quality Assurance	Implementing processes and practices to verify that the system functions correctly, meets user needs, and delivers a positive user experience.
Testing	verify that different components work together correctly. Validate the system against the specified requirements. Ensure that the system can handle expected user loads and transactions efficiently.

2.8 Software system attributes

Attributes	Description
Reliability	The system should perform consistently and accurately under various conditions without unexpected failures or errors like Database information, Guest communication, special offers and packages.
Performance	The website should load search results pages in under a few seconds, display room availability in real-time, and complete booking transactions swiftly for a seamless user experience. Not only that, the system should be the responsive and provide quick access to the information and efficiently process the queries. Allows admins to get insights into performance about progress.
Scalability	The system should handle increased user traffic, bookings, and data volumes during peak periods without performance degradation. In addition to that, the system should be use and navigate for customers, Taxi services, admin and managers. The user interface should be intuitive, well-organized, and provide clear instructions.
Security	Protecting sensitive patient data and clinic information is paramount. The system must implement appropriate access controls, encryption, and other security measures to prevent unauthorized access, data breaches, or data loss.
Ease of use (usability)	The guest house booking website should have a user-friendly interface with intuitive search filters, clear room descriptions, and easy-to-follow booking steps.
Maintainability	The development team should use modular code architecture, guests booking maintaining, provide correct way for room booking process, taxi service maintain (handle the drivers) and comprehensive documentation for system maintenance and enhancements.
Availability	The booking platform should maintain high availability to allow users uninterrupted access to room information, bookings, and reservations management.

Portability	The software system should be deployed and run on different platforms, devices, and environments without major modification or compatibility issues, supporting cross-platform compatibility and flexibility.

3. Supporting Information



4. References

https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/

https://online.visual-paradigm.com/