KING SALMAN INTERNATIONAL UNIVERSITY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

BUS RESERV. SYSTEM

RAROUTE

Prepared For :

Dr/ Wael Fawaz
Eng/ Baher Adel

Team :

Yousef Hamed 221101233 CE
Mohamed Younis 222101421 CE
Hamsa Hany 221101230 CE
Ebrahim Hany 221101202 CE

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Project Proposal 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 2

1.5 References 2

2. Overall Description 3

2.1 System Environment 3

2.2 Functional Requirements Specification...4

2.2.1 Register users Use Cases... 4

2.2.1 User Use Case 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation.. 6

2.7 Assumptions and Dependencies 6

3. External Interface Requirements 7

3.1 User Interfaces 7

4. System Features 12

4.1 System Feature 1 12

5. Other Nonfunctional Requirements 13

5.1 Performance Requirements 13

5.2 Safety Requirements 13

5.3 Security Requirements 13

5.4 Software Quality Attributes 14

5.5 Business Rules 14

Appendix: Analysis Models 14

Sequence Diagram 15

Activity Diagram... 16

References 17

1. Introduction

1.1. Purpose

The Bus Reservation System RAROUTE is to streamline and enhance the process of booking and managing bus reservations for both passengers and operators. The system is designed to offer a comprehensive and user-friendly platform that addresses the various needs of the users.

1.2 Document Conventions

The document follows standard typographical conventions for clarity and readability. Requirements are presented in a structured format with headings, bullet points, and numbering for easy navigation. Functional requirements are prioritized based on their importance to the overall system functionality, with higher-level requirements assumed to influence the priority of detailed requirements. Each requirement statement is accompanied by its own priority to ensure clear understanding of its significance.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) is intended for various stakeholders involved in the development, implementation, and maintenance of the Bus Reservation System RAROUTE. These include developers, project managers, marketing staff, users, testers, and documentation writers.

The document is organized to provide a comprehensive understanding of the system's requirements, functionalities, and constraints. Readers are encouraged to begin with the overview sections to gain a high-level understanding of the project scope, objectives, and goals. Subsequently, readers can explore specific sections relevant to their roles and interests, such as functional requirements for developers, operational procedures for project managers, usability guidelines for marketing staff, and so forth.

1.4 Product Scope

The Bus Reservation System RAROUTE is a software solution designed to facilitate the booking and management of bus reservations for passengers and operators. It aims to modernize and streamline the bus travel experience by providing an intuitive online platform for booking tickets, managing schedules, and analyzing operational data.

Key objectives of the system include enhancing efficiency, convenience, and reliability for both passengers and operators. By offering real-time booking capabilities, flexible payment options, and user-friendly interfaces, RAROUTE seeks to improve the overall travel experience while promoting sustainable transportation practices.

1.5 References

- Vision and Scope Document for Bus Reservation System RAROUTE (Version 1.0, Date: [22/5/2024], Source: [ELtor , South Sinia])
- User Interface Style Guide for RAROUTE(Version 1.0, Date: [22/5/2024], Source: [ELtor , South Sinia])
- System Requirements Specifications for RAROUTE(Version 1.0, Date: [22/5/2024], Source: [ELtor , South Sinia])
- Use Case Documents for RAROUTE(Version 1.0, Date: [22/5/2024], Source: [ELtor , South Sinia])
- Any relevant contracts, standards, or regulatory documents applicable to the development and deployment of RAROUTE.

2. Overall Description

2.1 System Environment

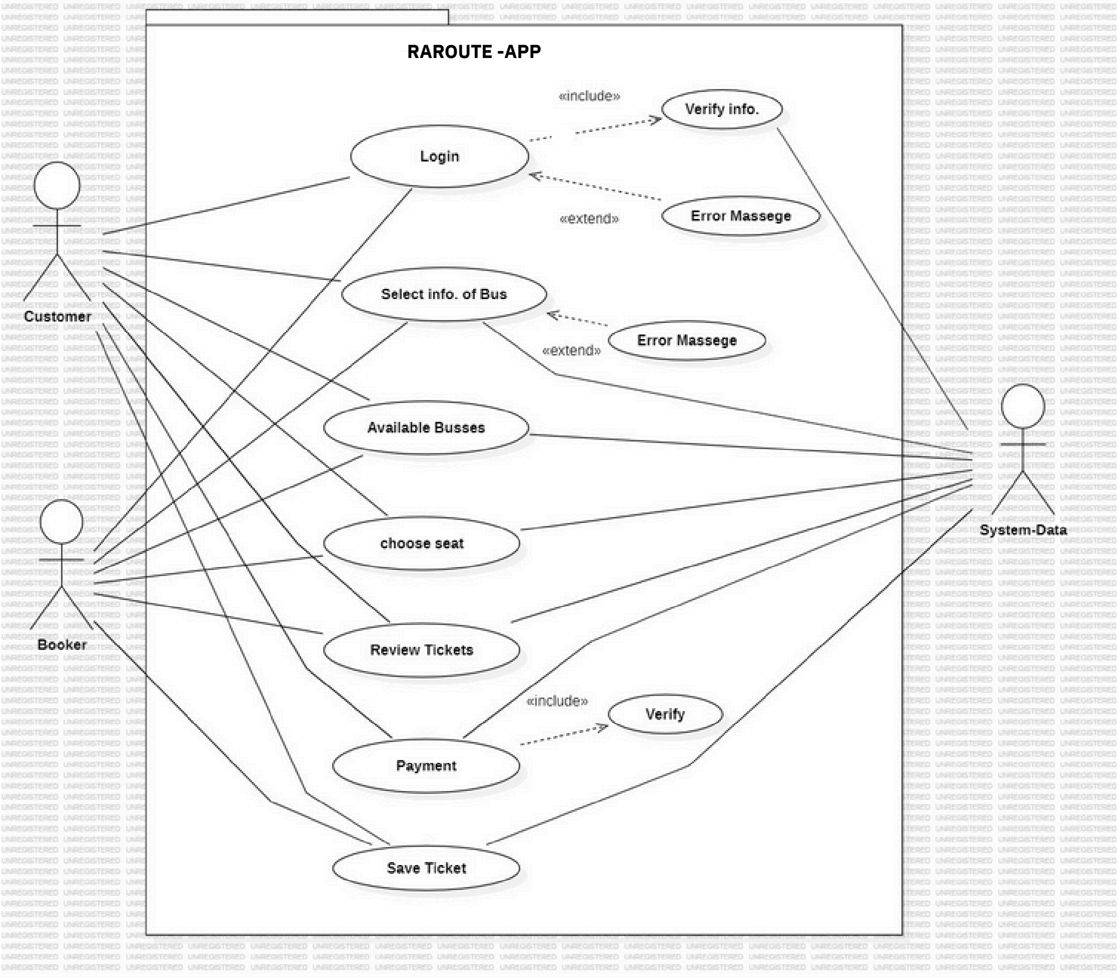


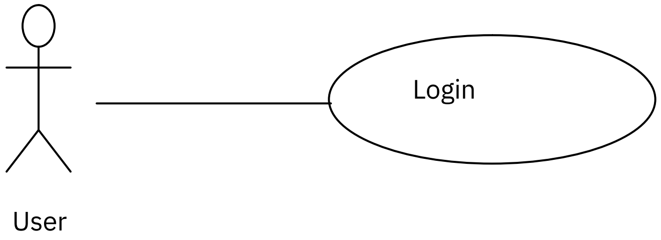
Figure 1 - System Environment

2.2 Functional Requirements Specification

This section outlines the use cases for each of the activities in the program.

2.2.1 Register users Use Case

Use case: Account Create



Brief Description

If the user wants to use the program, he must login within the program to be able to benefit from it.

Initial Step-By-Step Description

Before this use case can be initiated, the user has already the program.

1. The user will login

2. The system requires Username and password
3.The System do verify
4. User access the system.

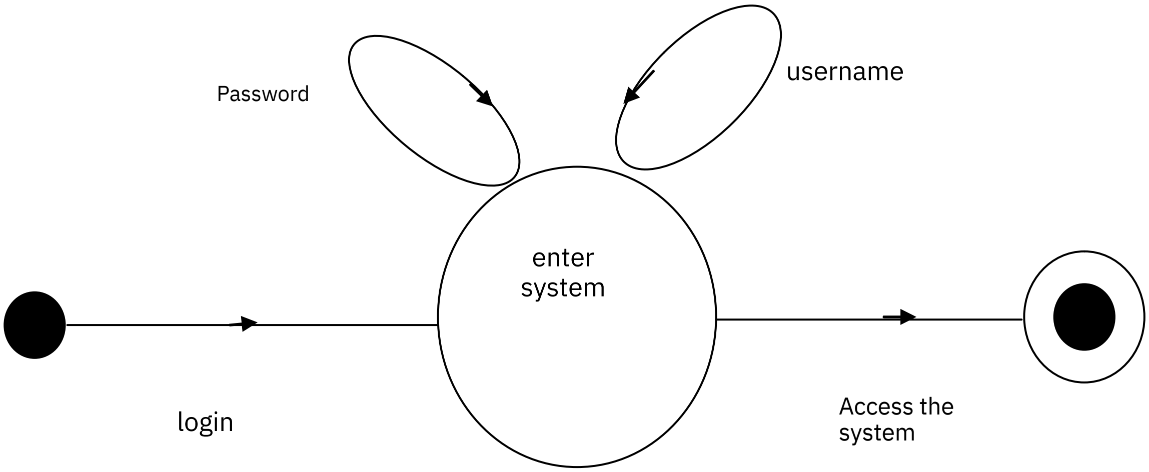


Figure 2 - Create account process

2.2.2 User Use Case

The User after register has the following:

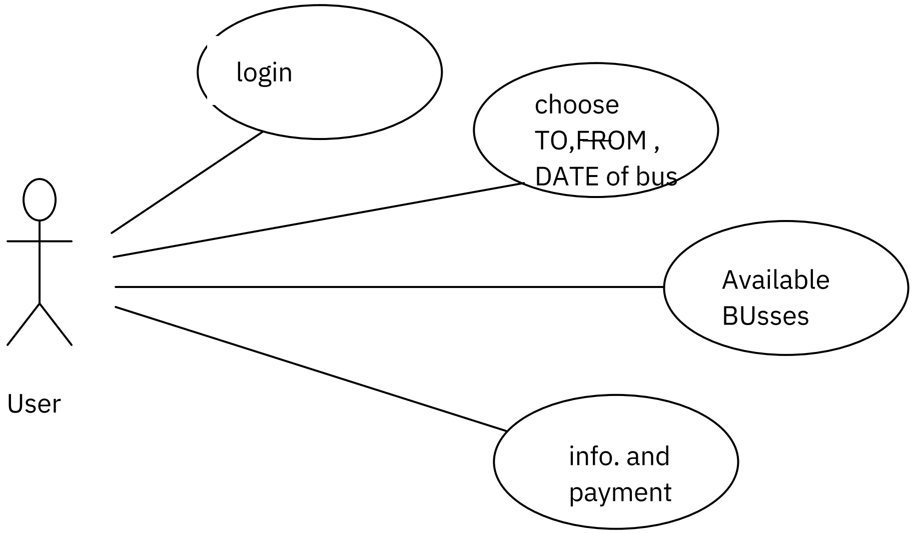


Figure 3 - User Use Cases

2.5 Design and Implementation Constraints

- **Regulatory Policies**: Compliance with data protection regulations (e.g., GDPR) and industry standards governing financial transactions.
- **Hardware Limitations**: System must accommodate varying hardware specifications of end-user devices, ensuring compatibility and performance across different platforms.
- **Interfaces to Other Applications**: Integration with payment gateways, mapping services, and third-party APIs for route optimization and data analytics.
- **Technology and Tools**: Utilization of specific programming languages (Java), frameworks as per project requirements.

- **Security Considerations**: Implementation of encryption protocols, access controls, and secure authentication mechanisms to safeguard user data and transactions.
- **Design Conventions**: Adherence to coding standards, design patterns, and architectural principles established by the development team or client organization.

2.6 User Documentation

User documentation for the Bus Reservation System RAROUTE will include:

- User Manuals: Comprehensive guides explaining system functionalities, booking procedures, and account management.
- Online Help: Context-sensitive help available within the application to assist users with specific tasks or inquiries.
- Tutorials: Interactive tutorials or walkthroughs to onboard new users and familiarize them with the system's features and navigation.
- Delivery Format: User documentation will be delivered in digital formats, accessible online through the system's interface and downloadable as PDF files for offline reference.

2.7 Assumptions and Dependencies

Assumptions:

- Availability of stable internet connectivity for real-time booking and transaction processing.
- Continuous support and maintenance of third-party APIs and services used for payment processing, mapping, and analytics.

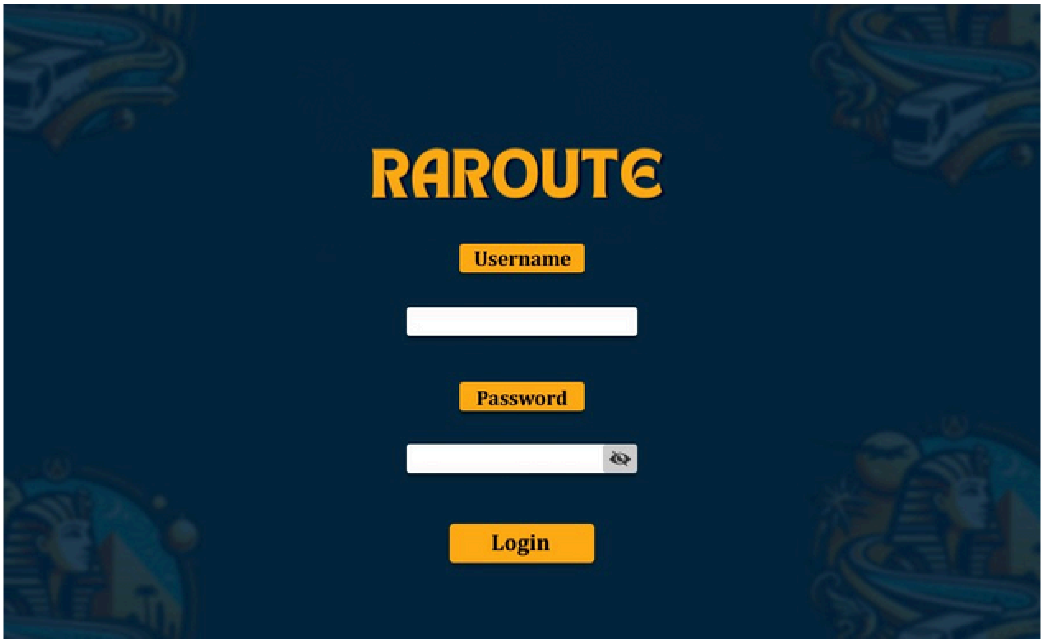
Users have basic familiarity with online booking systems and are comfortable navigating web interfaces.

Dependencies:

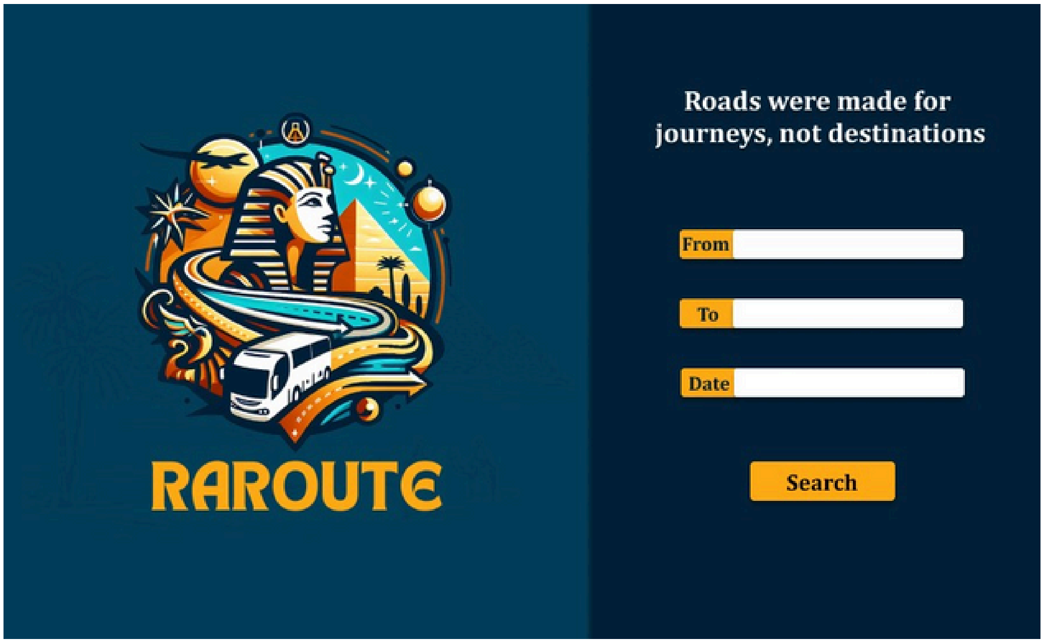
- Integration with external payment gateways for secure transaction processing.
- Access to mapping services for route planning and navigation features.
- Availability of development resources and expertise in chosen technologies and frameworks.
- Collaboration with regulatory bodies to ensure compliance with relevant data protection and privacy regulations.

3. External Interface Requirements

3.1 User Interface :

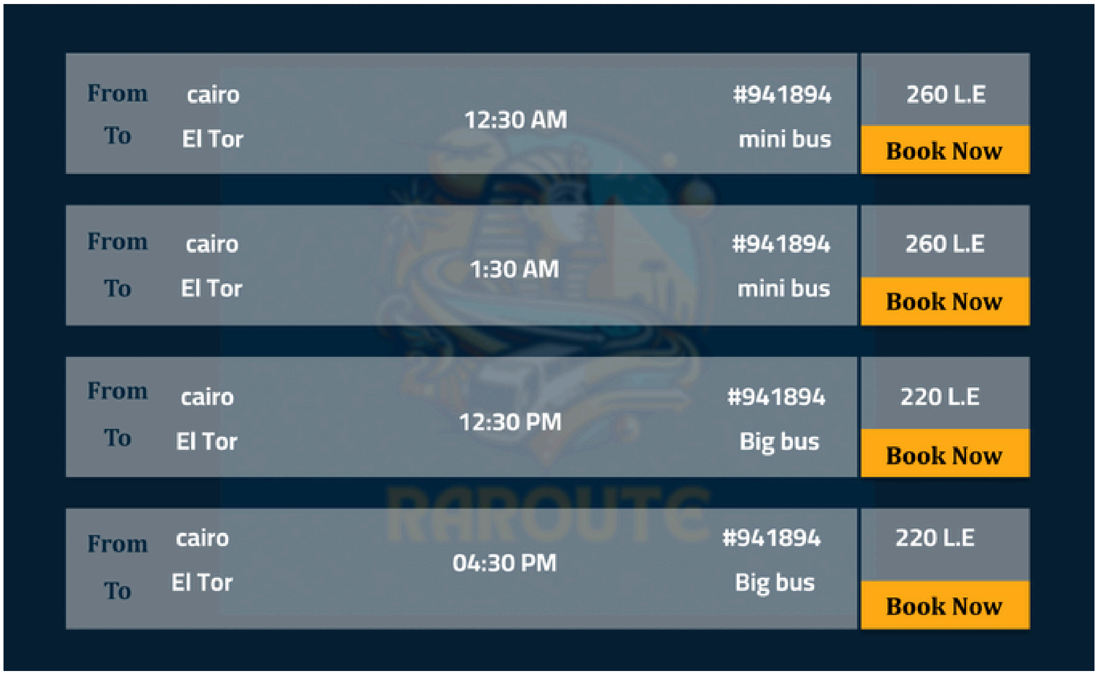


login Page



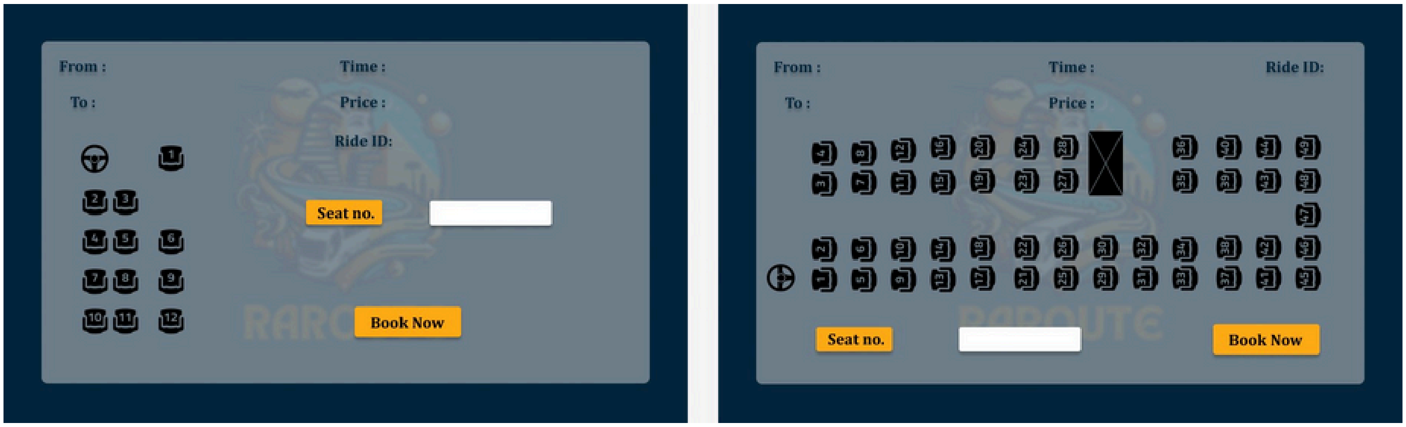
Roads were made for journeys, not destinations

Destintion Page



From To	cairo El Tor	12:30 AM	#941894 mini bus	260 L.E Book Now
From To	cairo El Tor	1:30 AM	#941894 mini bus	260 L.E Book Now
From To	cairo El Tor	12:30 PM	#941894 Big bus	220 L.E Book Now
From To	cairo El Tor	04:30 PM	#941894 Big bus	220 L.E Book Now

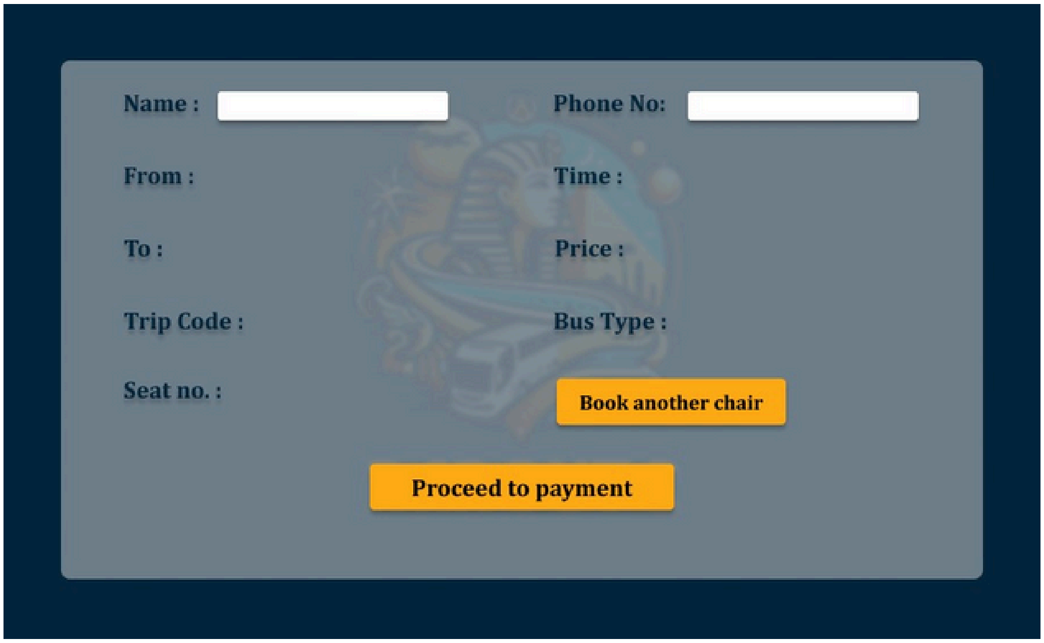
Available Busses Page



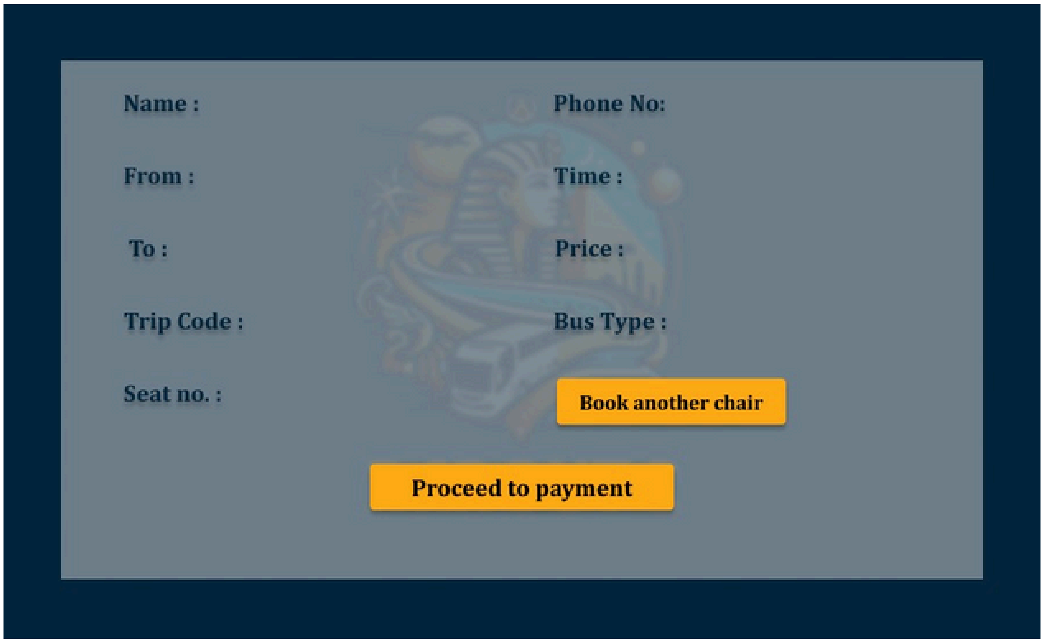
From :
To :
Time :
Price :
Ride ID:
Seat no.
Book Now

From :
To :
Time :
Ride ID:
Price :
Seat no.
Book Now

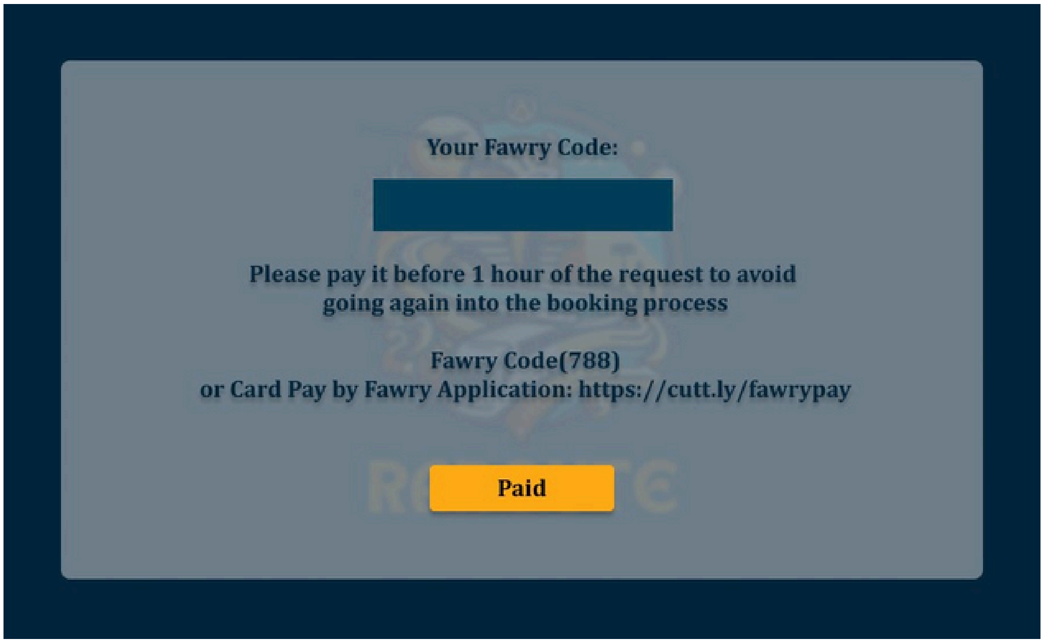
Big - Mini Bus pages



Booker Confirm Chair



User Confirm Chair

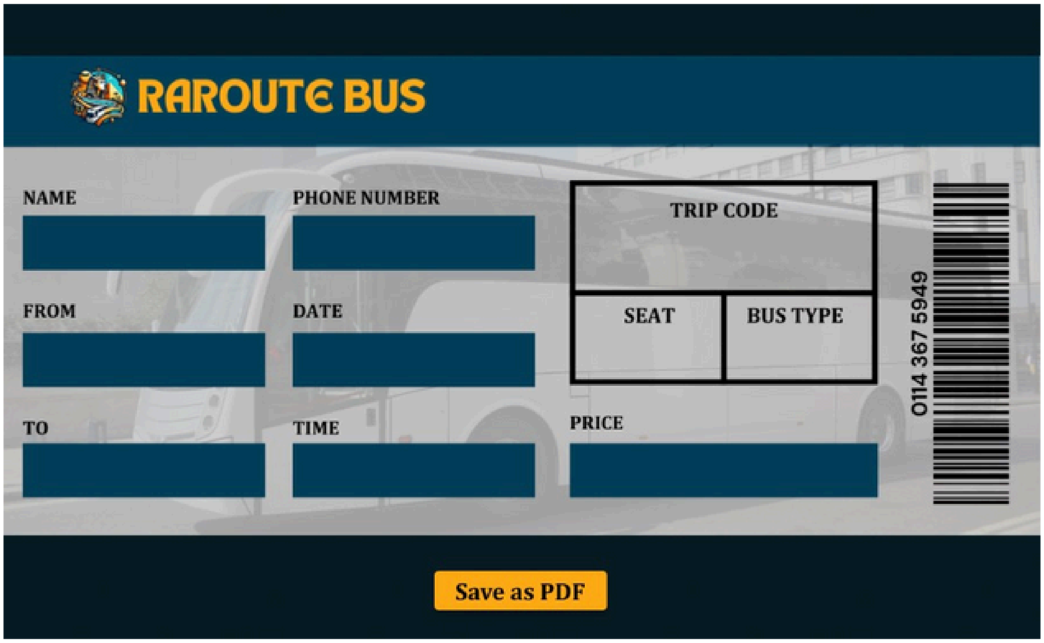


Your Fawry Code:

Please pay it before 1 hour of the request to avoid going again into the booking process

Fawry Code(788)
or Card Pay by Fawry Application: https://cutt.ly/fawrypay

Payment for User



Final Ticket Page

4. System Features

4.1 Bus Reservation

4.1.1 Description and Priority Users should be able to search for available buses, select seats, and make reservations. This feature is of High priority as it forms the core functionality of the system.

4.1.2 Stimulus/Response Sequences

- User initiates a bus search by entering the desired origin, destination, and travel date.
- System retrieves available bus options matching the search criteria.
- User selects a preferred bus and chooses seats.
- System confirms the reservation and provides booking details.

4.1.3 Functional Requirements

- **REQ-1:** Users can search for available buses by specifying the origin, destination, and travel date.
- **REQ-2:** The system displays a list of available buses along with departure times, journey duration, and ticket prices.
- **REQ-3:** Users can select seats from an interactive seat map, with real-time availability updates.
- **REQ-4:** The system calculates and displays the total fare based on the selected seats and any additional services (e.g., luggage storage).
- **REQ-5:** Users can review and confirm their reservation before finalizing the booking.
- **REQ-6:** The system generates a unique booking ID and e-ticket for each reservation, which is sent to the user via email or SMS.
- **REQ-7:** The system updates bus availability in real-time to prevent overbooking and conflicts.
- **REQ-8:** Users receive notifications for booking confirmations, trip reminders, and any changes to their reservation status.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Response Time**: The system should respond to user actions (e.g., searching for routes, booking tickets) within 2 seconds under normal load conditions.
- **Scalability**: The system should handle a minimum of 1000 concurrent users without significant performance degradation.
- **Peak Load Handling:** During peak hours, the system should maintain an average response time of 5 seconds or less for all user interactions.
- **Database Performance**: Database queries should execute within 1 second, ensuring efficient retrieval of booking and scheduling data.
- **Offline Capability**: The system should be capable of functioning offline for basic tasks such as viewing existing bookings and offline ticket generation, with data synchronization occurring upon reconnection.

5.2 Safety Requirements

- **Data Security**: User data, including personal and payment information, must be encrypted and securely stored to prevent unauthorized access or breaches.
- **Transaction Integrity**: Ensure that all financial transactions are conducted securely and accurately to prevent fraud or loss of funds.
- **Emergency Procedures**: In the event of system failures or critical errors, provide clear instructions for users and operators on backup procedures and emergency contact information.

5.3 Security Requirements

- **User Authentication**: Implement secure user authentication mechanisms, such as password hashing and multi-factor authentication, to prevent unauthorized access to user accounts.

- **Data Encryption**: Encrypt sensitive data during transmission and storage to protect against interception and unauthorized access.
- **Regulatory Compliance**: Ensure compliance with relevant data protection regulations, such as GDPR, to safeguard user privacy and data security.
- **Regular Security Audits**: Conduct regular security audits and penetration testing to identify and address potential vulnerabilities in the system.

5.4 Software Quality Attributes

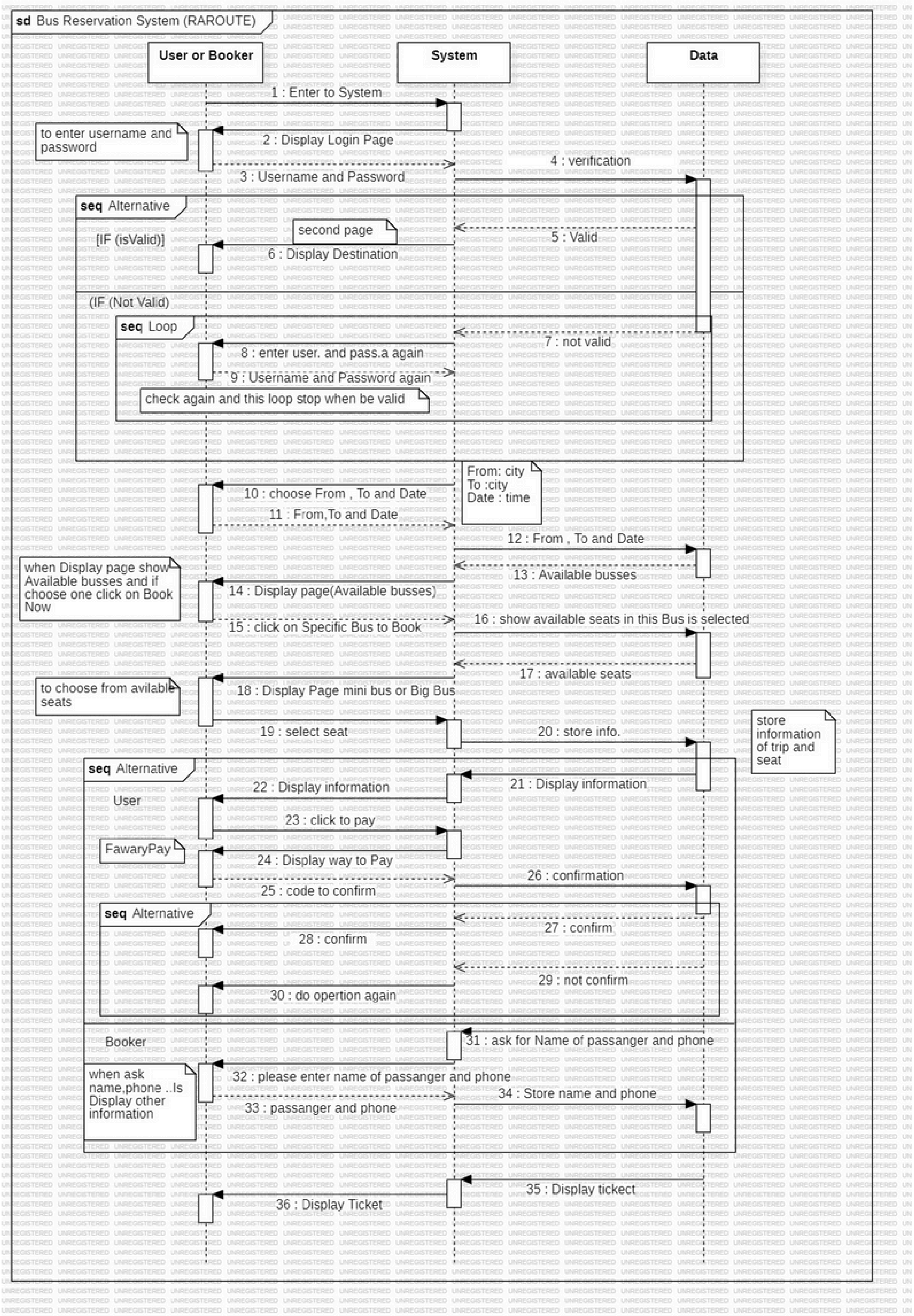
- **Reliability**: Ensure high system uptime and availability to minimize disruptions to user and operator activities.
- **Maintainability**: Design the system with modular components and clear code documentation to facilitate easy maintenance and updates.
- **Usability**: Prioritize user-friendly interfaces and intuitive workflows to enhance the overall user experience for both passengers and operators.
- **Interoperability**: Ensure compatibility with external systems and APIs to facilitate integration with third-party services and tools.

5.5 Business Rules

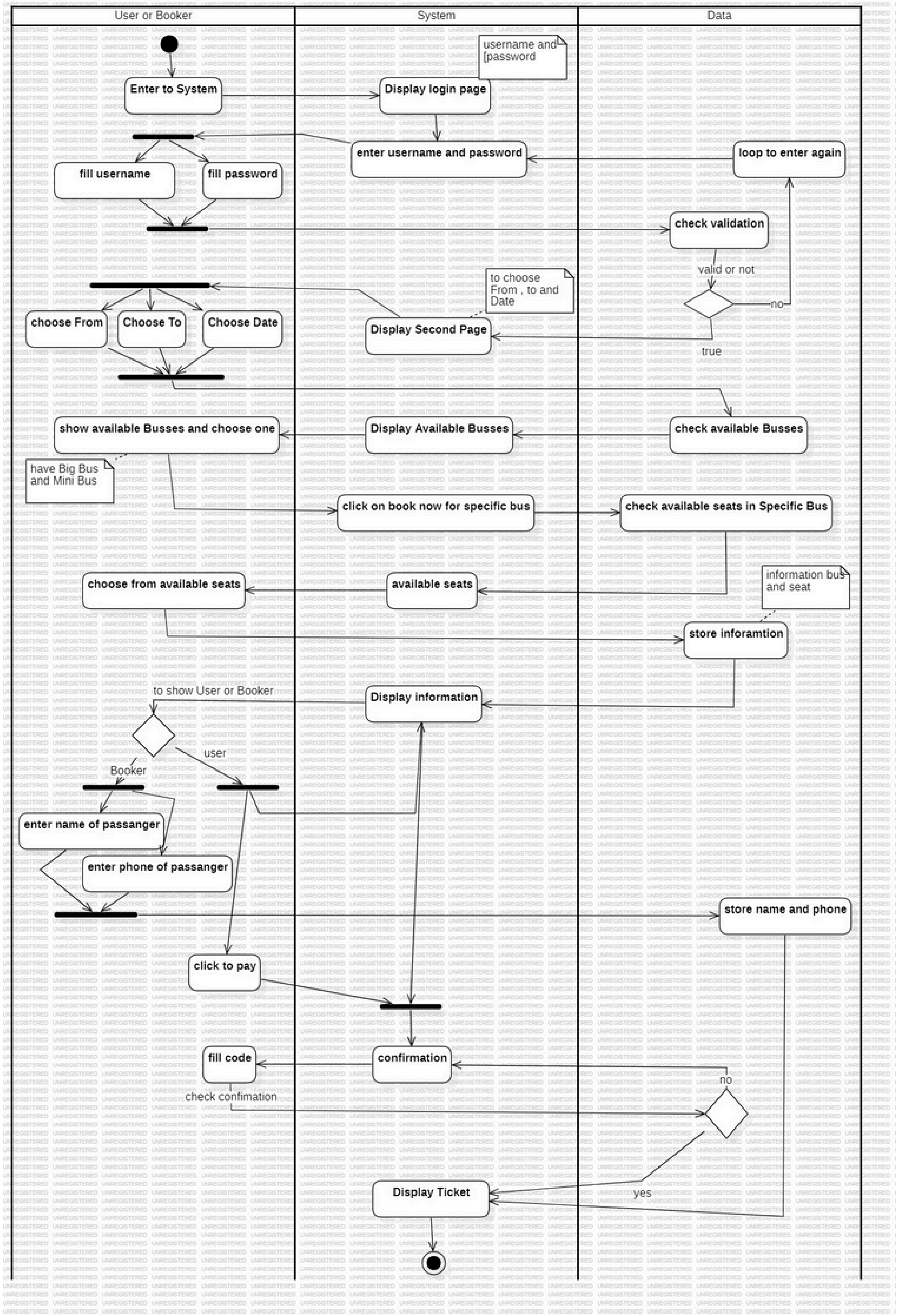
- **Booking Restrictions**: Define rules for booking limitations, such as maximum ticket purchases per user and minimum booking lead times.
- **Cancellation Policies**: Establish guidelines for cancellation fees, refund processes, and timelines to ensure fair treatment of passengers and minimize revenue loss for operators.
- **Operator Privileges**: Specify roles and permissions for bus operators, including access levels for managing schedules, pricing, and reporting functionalities.

Appendix: Analysis Models

Sequence Diagram


sd Bus Reservation System (RAROUTE)
User or Booker
System
Data
1 : Enter to System
to enter username and password
2 : Display Login Page
3 : Username and Password
4 : verification
seq Alternative
[IF (isValid)]
second page
5 : Valid
6 : Display Destination
(IF (Not Valid)
seq Loop
7 : not valid
8 : enter user. and pass.a again
9 : Username and Password again
check again and this loop stop when be valid
From: city
To :city
Date : time
10 : choose From , To and Date
11 : From,To and Date
12 : From , To and Date
when Display page show Available busses and if choose one click on Book Now
13 : Available busses
14 : Display page(Available busses)
15 : click on Specific Bus to Book
16 : show available seats in this Bus is selected
17 : available seats
to choose from avilable seats
18 : Display Page mini bus or Big Bus
19 : select seat
20 : store info.
store information of trip and seat
seq Alternative
User
21 : Display information
22 : Display information
23 : click to pay
FawaryPay
24 : Display way to Pay
25 : code to confirm
26 : confirmation
seq Alternative
27 : confirm
28 : confirm
29 : not confirm
30 : do opertion again
Booker
31 : ask for Name of passanger and phone
when ask name,phone ..Is Display other information
32 : please enter name of passanger and phone
33 : passanger and phone
34 : Store name and phone
35 : Display tickect
36 : Display Ticket

Activity Diagram



References

1 https://en.wikipedia.org/wiki/Software_requirements_specification

2 https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/

3 https://creately.com/app/#

4 https://www.edsd.com/portfolio/mobile-applications

5 https://stackoverflow.com/tags

6 https://github.com/topics