Nile University

CSCI313: Software Engineering – Section 4

RailWise Project

Team members:

Dima Mamdouh Mohamed - 211000081

Mariam Mohamed Attia - 211000533

Mohamed Osama Mahmoud - 211001816

Nesma Mohamed Hegazy - 211001626

Table of Contents

1) Introduction

1.1Purpose:

The purpose of this Software Requirements Specification (SRS) document is to present a clear and thorough explanation of our Railway Tracking and Arrival Time Prediction application. The features, interface, and purpose of the application, as well as its test cases, performance, and constraints, will all be included in this document.

1.2 Project Scope:

The Railway Tracking and Arrival Time Prediction System is a software initiative designed to improve the efficiency and dependability of rail transport. This program will offer live train tracking and precise predictions for arrival times, to enhance the passenger experience, safety, and operational effectiveness of railway networks.

1. Project Objectives:
    - Real-Time Train Tracking: The system will involve creating an extensive, live tracking component that utilizes GPS, sensors, and other technological means to precisely determine the precise position of every train within the network.
    - Arrival Time Prediction: A predictive model will be implemented to estimate the arrival times of trains at various stations. This will consider historical data, current conditions, and potential delays.
    - Passenger Updates: Furnish passengers with up-to-the-minute details about their train's status, any potential delays, and other pertinent information.
    - Enhanced Safety Measures: Bolster railway safety by providing insights into train locations and speeds, thereby reducing the likelihood of accidents.
    - Streamlined Operations: Optimize the allocation of resources and scheduling for railway operators, ultimately enhancing the overall efficiency of the system.
    - Data Integration: The system will integrate with various data sources, including train schedules, weather conditions, and historical performance data to improve prediction accuracy.

1.3 Used Technologies:

- Flutter
- Dart
- Figma for User Interface (UI)
- MySQL for Database
- Google Map API

*1.4 **Intended Audience**:*

- Passengers (users): To know the exact time of the train and if there is a delay in train arrival time.
- Developers: To develop, implement, and maintain the required tasks for the app.
- Testing Team: To ensure that the system meets all the requirements by using test cases.
- Railway Authorities: To build the plan and the system requirements.
- Station Staff: To manage train movements, platform allocations, and other related tasks to ensure smooth train operations within stations.

1.5Overview:

The specifics of our Railway Tracking application are explained in the remaining chapters of the SRS. Chapter Two covers the general descriptions of the application and its requirements, such as how our product functions and its constraints. And in Chapter Three, the functional and non-functional requirements for our application are detailed and explained clearly. The rest of the chapters of the SRS talk about the interfaces, diagrams, and any other specifications concerning our Railway Tracking application.

2) <u>Overall Description</u>

2.1Product Perspective:

1. The “RailWise” mobile application is a self-contained and independent system.
2. “RailWise” has two types of users, one type is the Passenger, and the other type is the Administer, where each type will be able to do different features.
3. The application will use an API to provide the geographical information, and a database that contains all other necessary information for the application.

2.1.1 Product Functions:

<u>Passenger:</u>

- The Passenger can register as a new member using their email address and other personal information.
- The Passenger can login if they have an existing account using their username and password.
- The Passenger will be able to search for the specified train they want, using the train or trip number, arrival/departure time or station, or date.
- The Passenger will be able to select the train they want from the search’s results to view its details.
- The Passenger will have the option to book a train.
- The Passenger will be able to see their train’s real-time location, estimated arrival time, and other train details.
- The Passenger will be able to edit their personal information on their account.
- The Passenger will be able to log out of their account or delete it.
- The Passenger will receive notifications regarding their train.

<u>Administrator:</u>

- The admin can log in to the system using their username and password.
- The admin can add a new train and its details to the database.
- The admin can edit the train’s details such as departure time to help the system predict the train’s arrival time.
- The admin can delete a train or an account from the database.
- The admin can send messages to the Passenger if there is a delay in the train’s arrival time.

2.2 User Characteristics:

- The user should be able to use smartphones or tablets.
- The user should be able to make an account on the application.
- The user should have access to the internet.
- The user should be able to understand and follow simple instructions.

2.3Constraints:

- The application must be able to run on a variety of devices, including smartphones and tablets.
- The application must be able to work with different internet connections, including Wi-Fi and cellular data.
- The application must be able to work in different cities.
- The application must be able to handle a large number of concurrent users.
- The application must be able to process data quickly and efficiently.
- The application must be secured and protect user data.

2.4 Assumptions and Dependencies:

Requirements for the Railway Tracking and Arrival Time Prediction System:

- Availability: The application will be available for download from the Google Play Store.
- GPS: The user's mobile device must have GPS navigation capabilities.
- Internet: The user must have internet access while using the application.
- Location permission: The user must grant the application permission to access their device's location services.
- Software updates: Users should keep the application updated to access new features, bug fixes, and security enhancements.

3) <u>Functional Requirements</u>

3.1 User Class 1: The User

- **<u>Title:</u>** Registration

Description: The system will enable users to register for an account so that they can access their account later.

Required Information:

- Username
- Password (at least 8 characters)
- Email (unregistered email)
- Phone Number
- City

- **<u>Title:</u>** Log in

Description: The system would enable the user to log in for their account if they entered a valid username and password.

Required Information:

- Email
- Password

- **<u>Title:</u>** Edit Profile

Description: The system will allow the user to edit their profile information

- **Title:** Delete Account

Description: The system will allow users to delete their account by entering the email and the password

Required Information:

- Email
- Password

- **Title:** Search for Trip

Description: This feature allows users to search for trips by entering their desired departure and arrival locations, as well as their travel dates. The system will return a list of available trips, along with their prices, estimated travel times, and other relevant information.

Required Information:

- Departure Location
- Arrival Location
- Travel Dates
- Travel Class

- **Title**: Display Train Details

Description: This feature allows users to view detailed information about a particular train, such as its schedule, route, and amenities.

- **Title:** Select the Specified Train

Description: This feature allows the user to select a specific train from a list of available trains.

Required Information:

- Number of tickets

- **Title**: Book Train

Description: This feature allows users to book a ticket on a specific train.

Required Information:

- Passengers' name
- Seat Preference (Window, aisle, table)
- Payment Information

- **Title:** Real-Time Train Location

Description: This feature allows users to view the real-time location of a specific train.

- **Title:** Receive Notification

Description: This feature allows users to receive notifications about their train booking and the train's arrival time.

3.2 User Class 2: The Administrators

- **Title:** Log in

Description: The system would enable the administrators to log in to their accounts if they entered a valid username and password.

Required Information:

- Email
- Password

- **Title:** Add Train

Description: The system will allow the administrators to create and add new train records to the system.

Required Information:

- Train name and identifier.
- Departure and arrival times.
- Train schedule.
- Train capacity.
- Ticket prices.

- **Title:** Edit Train Information

Description: The system would enable the administrators to edit or change the information of trains.

Required Information:

- Train identifier.

- **Title:** Send Messages

Description: The system would enable the administrators to send messages to passengers if there is any important information, updates, or announcements.

Required Information:

- Target passenger's phone number.
- Message content

- **Title:** Delete Train

Description: The system enables administrators to delete specific train versions. This action requires authentication through the input of an email and password associated with the administrator account

Required Information:

- Email
- Password

4) Non-Functional Requirements

4.1 Reliability:

- **Hardware failures:** This could include failures of sensors, tracking devices, or servers.
- **Communication failures:** This could include failures of network connections, satellite signals, or cellular networks.
- **Data quality issues:** This could include inaccurate or incomplete data, or data that is not updated in real time. Human errors: This could include errors made by train operators, dispatchers, or other personnel involved in the operation of the system.
- **Inaccurate arrival time predictions:** This could happen due to a variety of factors, such as unexpected delays, changes in traffic conditions, or errors in the prediction algorithm.
- **Failure to track train locations**: This could happen due to hardware failures, communication failures, or other factors.
- **Failure to detect delays:** This could happen due to data quality issues, software errors, or other factors.
- **Failure to notify passengers of delays:** This could happen due to communication failures, software errors, or other factors.

4.2 Recoverability:

- **Recovery time:** The application should be able to recover after a breakdown in approximately 4 hours.

4.3 Performance:

- **Response time:** The application should only take 2 seconds to load pages.
- **Accuracy of arrival time predictions:** This metric measures how closely the predicted arrival times match the actual arrival times, where the predicted arrival time of the trains should lag the actual arrival times by only 5 minutes.

4.4 Maintainability:

The application allows us to add/update functionalities and features to the system later on.

5) Interface

5.1 System Interface:

When a user first opens the RailWise application they will see the Starting page as shown in Figure 1. Then they will be directed to the Login page as shown in Figure 2. The user will enter their name and password to log in or if they don't have an account, they will choose the sign-up option at the bottom of the page, this will direct them to the Sign-up page as shown in Figure 3. In the Sign-up page they will need to enter their name, email, password, city, and phone number to create an account.

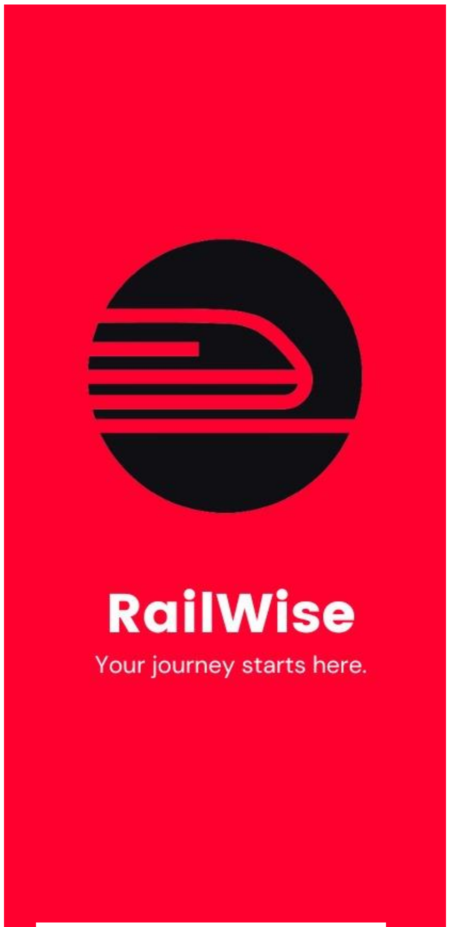


Figure 1. Starting Page

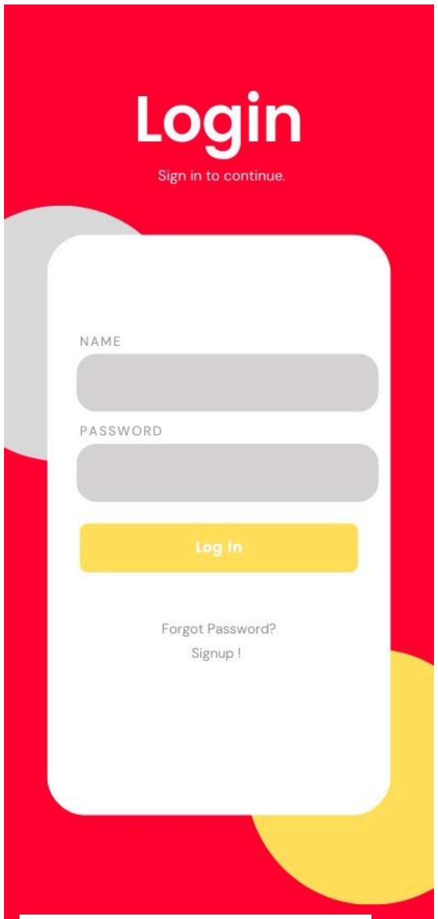


Figure 2. Login Page

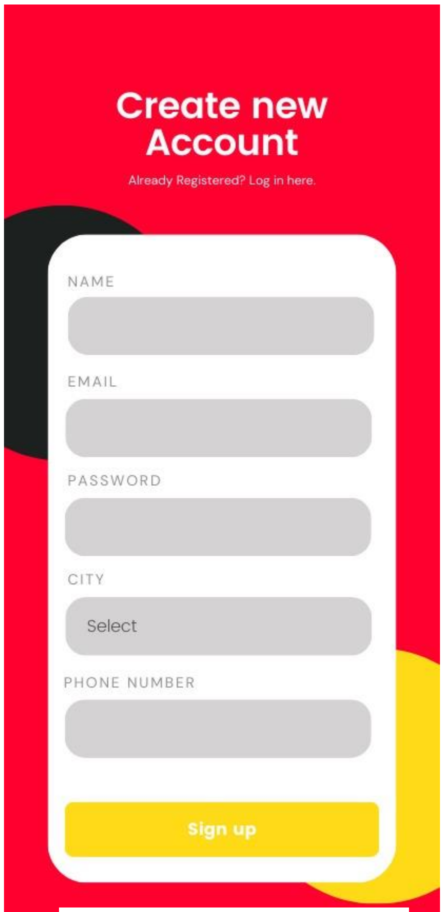


Figure 3. Sign-up Page

After the user finishes signing up, they will be directed to the Instruction page to get familiar with the app and how it works as observed in Figure 4. Afterwards, the user will be directed to the Home page as seen in Figure 5. In the Home page the user will have multiple options like searching for a trip, viewing their current trip's information, and in the menu bar they can click on the discover option to see trips available for this day, or the tickets option to see prices, or the profile option to view their account's information.

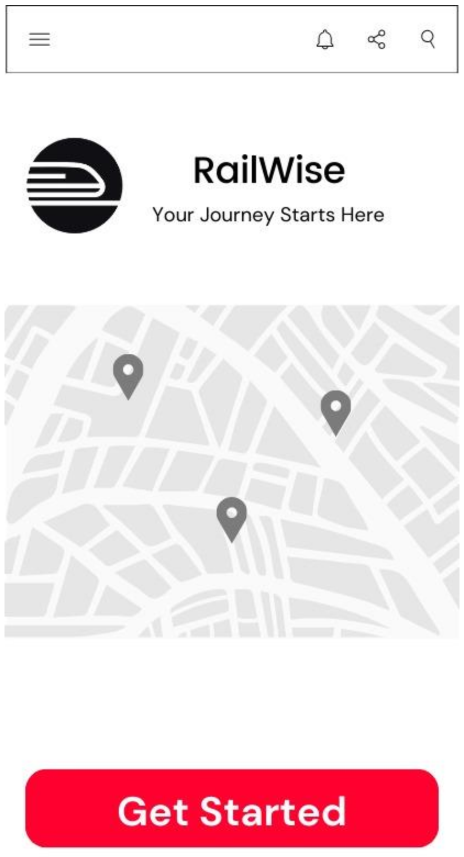


Figure 4. Instruction Page

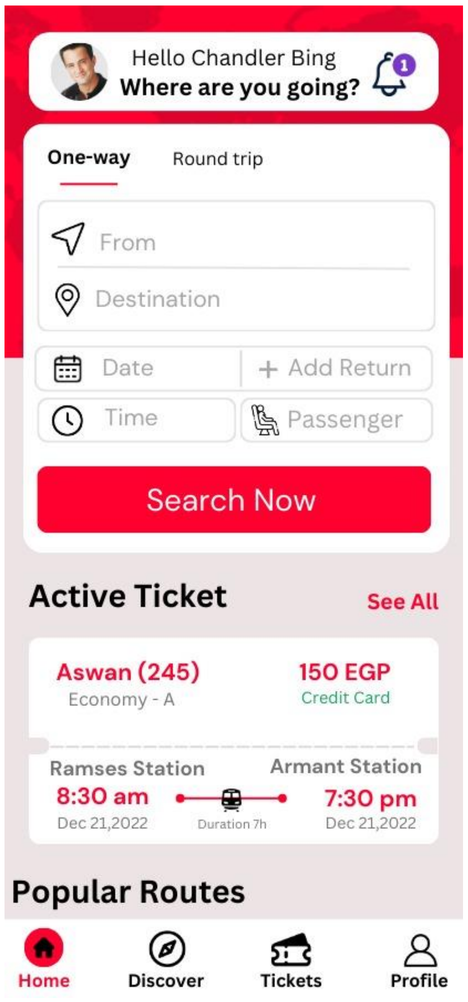


Figure 5. Home Page

If the user chooses to search for a trip, they will have two options to choose from, either they want to book a one-way trip or a round trip. In Figure 6, the user will enter from where they want to leave and their desired destination, then they will pick the date, time, and number of passengers to book their one-way trip. In Figure 7, if the user chooses to book a round trip, they will do the same process as the one-way trip but with a minor difference of having to enter the date of their return.

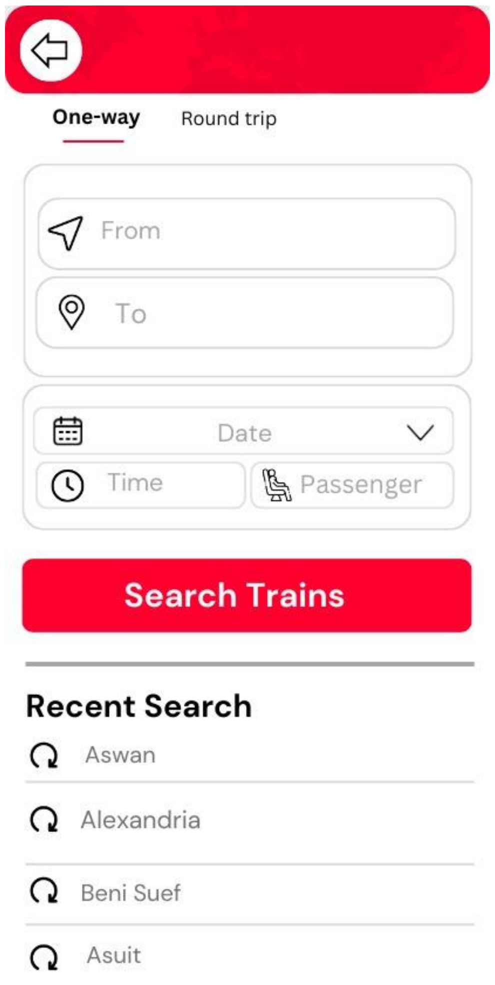


Figure 6. One-way Trip Search

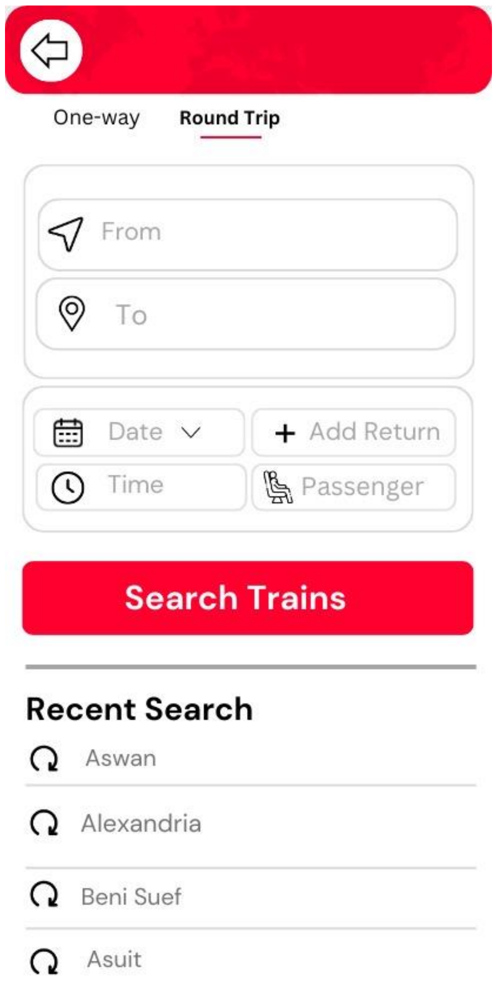


Figure 7. Round Trip Search

The Results page will show up for the user as shown in Figure 8. They will see the options available and their prices; once a user picks a train, they will be directed to the Select Seat page as shown in Figure 9. They will be able to see the available and unavailable seats of their desired train and their prices so they can choose the seats they want. Then they will click confirm to finish booking their seats.

In Figure 9, the Train Details page will show up for the user where they can see the details of their trip and have the option to track the train live. In the Track Train page, the user will be able to see where their train is currently and how long it will take for it to arrive as shown in Figure 11.

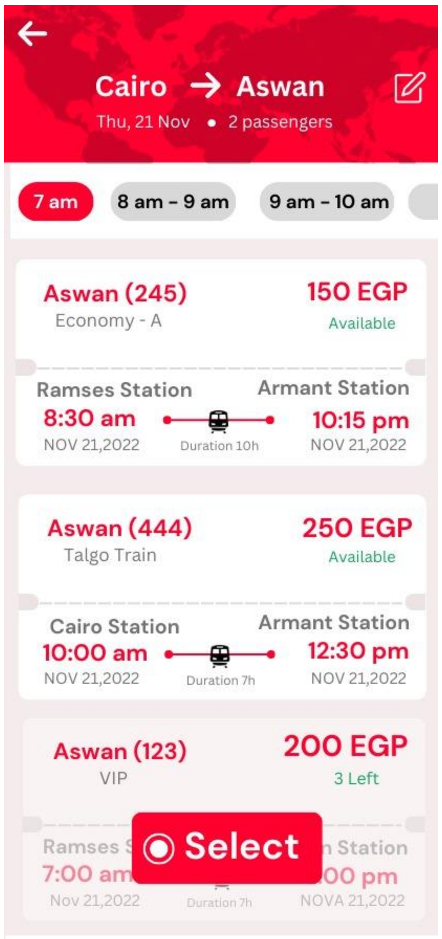


Figure 8. Results Page

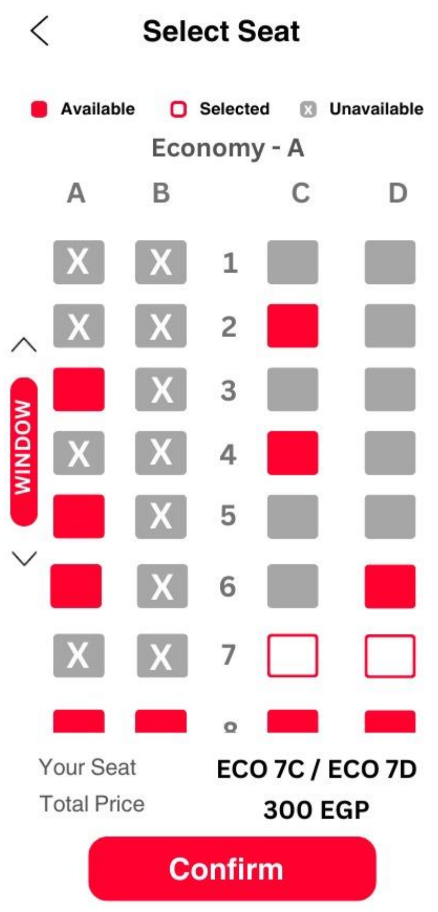


Figure 9. Select Seat Page

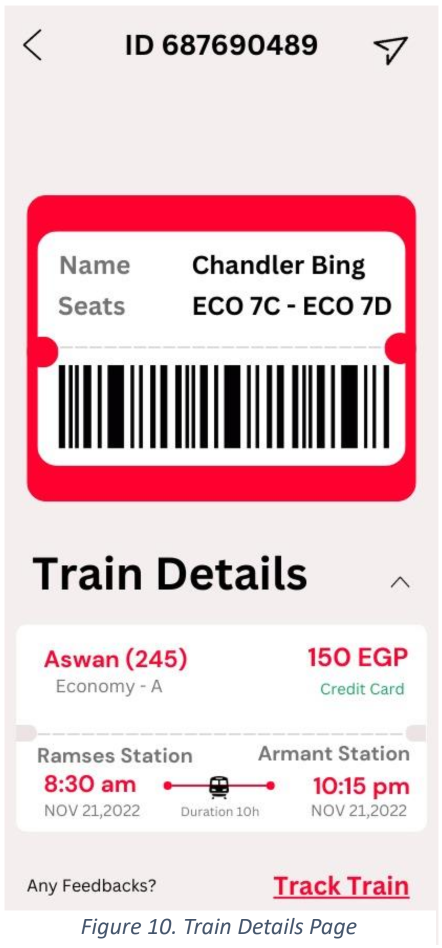
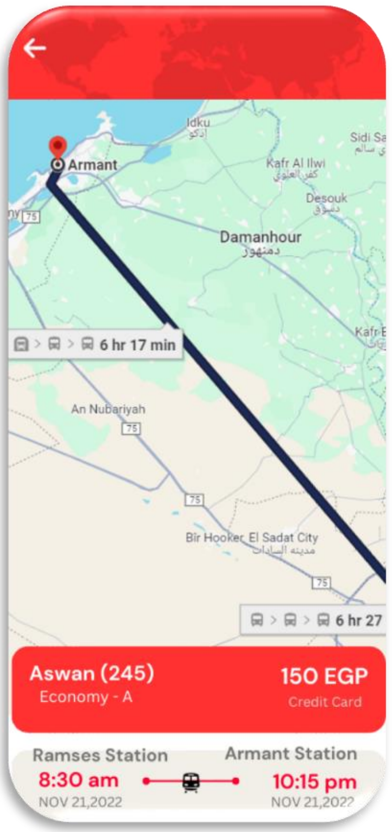
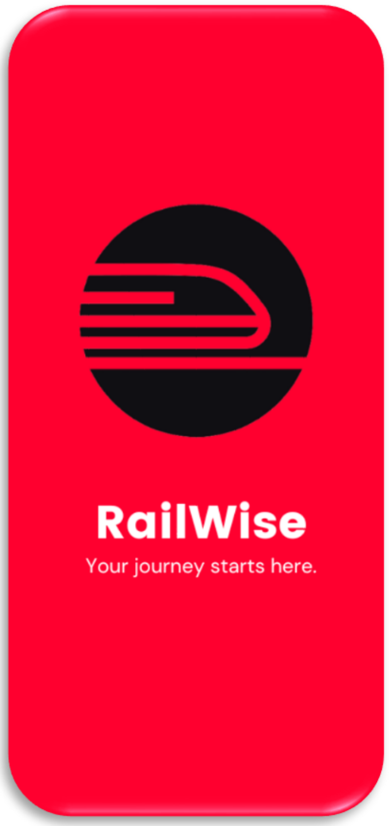


Figure 10. Train Details Page

The Track Train feature enhances the user experience by providing real-time information on the current location and estimated arrival time of the booked train. Upon booking a ticket, users gain the valuable capability to track their reserved train throughout the entire journey. This feature allows passengers to stay informed about the train's progress, offering a sense of reassurance and convenience. Whether the user is eagerly anticipating the arrival of their train or simply curious about the journey's status, the Track Train page serves as a reliable tool to visualize the train's location and anticipate its arrival time. This functionality adds an extra layer of transparency and control for passengers, ensuring a more seamless and enjoyable travel experience.



Aswan (245)
Economy - A
150 EGP
Credit Card
Ramses Station
8:30 am
NOV 21,2022
Armant Station
10:15 pm
NOV 21,2022



5.2 Software Interface

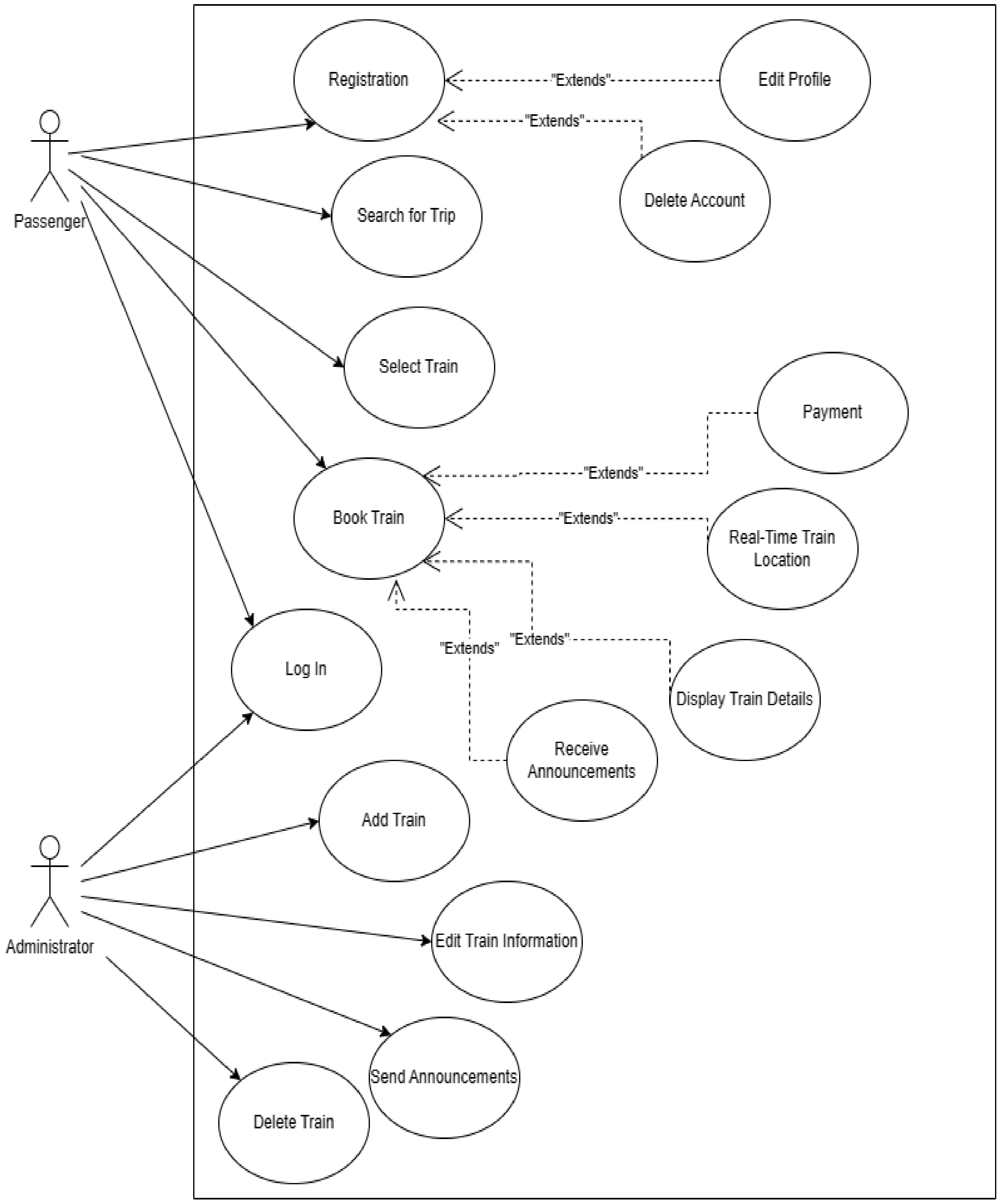
The RailWise mobile application is a sophisticated cross-platform solution developed using Flutter and Dart SDKs. It seamlessly integrates various libraries like Provider for state management, HTTP for server communication, geolocator for precise location tracking, Flutter Map for interactive maps, shared preferences for local data persistence, firebase analytics for analytics, and flutter_local_notifications for timely notifications. This comprehensive approach ensures a smooth and intuitive user experience, empowering users with real-time train tracking and accurate arrival time predictions, thereby contributing to increased efficiency and reliability in the rail transport sector.

*5.3*Hardware Interface:

The successful operation of the Railwise application hinges on specific hardware prerequisites integrated into the train's infrastructure. First and foremost, the train must be equipped with a GPS module to enable real-time location tracking, ensuring precise positioning within the railway network. Additionally, various sensors, including speed and proximity sensors, are essential for monitoring train speed and identifying potential safety hazards. A reliable communication module, such as cellular or Wi-Fi connectivity, is necessary to facilitate seamless data transfer between the train and the central server. An onboard computer serves as the core processing unit, collecting, and analyzing data from sensors and the GPS module. The installation of cameras on the train contributes to security and incident monitoring, requiring compatibility with the train's design. A passenger-friendly display unit in the train's compartment showcases real-time tracking information and arrival time predictions. The entire system relies on a dependable power supply integrated into the train's power system. An emergency stop button, crucial for immediate halting in emergencies, should seamlessly integrate with the train's control system. Lastly, the application's effectiveness depends on a robust server infrastructure capable of storing and processing data from multiple trains, necessitating compatibility with specified communication protocols and a reliable network infrastructure, be it cellular or a dedicated railway network.

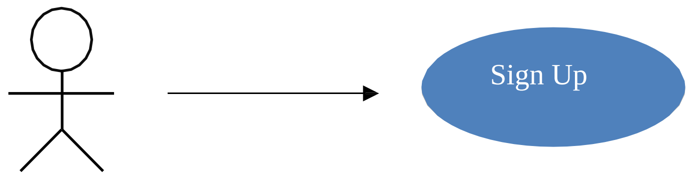
6) Diagrams

6.1 Use case Diagram:



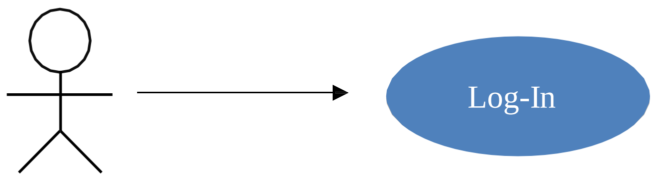
6.1.1 Use Case Scenarios

Use case 1: Registration (Sign-up)



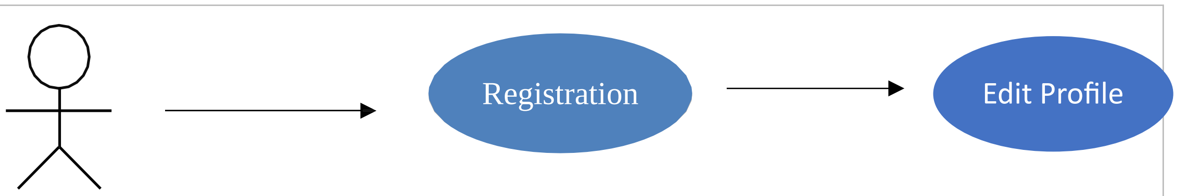
User case Name	Sign-Up
Actors	User (Non-Registered)
Main success scenario	1. The User chooses to create an account. 2. The system requires the user to enter their personal information: • Username / Password (at least 8 characters) • Email Address (unregistered email) /Phone Number/City 3. The user gives the system the required information. 4. The user clicks on the sign-up button. 5. The system checks the email (see if it is registered or not) 6. The system checks the username (check if it is registered or not) 7. The system creates the account
Exceptions	1. The user clicks the sign-up button without filing all the required information. 2. The username is already registered. 3. The email address is already registered. 4. Weak password 5. Invalid phone number
Actions	– **1** The system displays to the user an alert "Please fill in all the required fields." – **1.b** The User fills in the missing information. – **2.a** The system displays to the user "The email you've entered is already registered." – **2.b** The user enters another email address. – **3.a** The system displays to the user "Username already exists." – **3.b** The user enters another username. – **4.a** The system displays "Password must be at least 8 characters long and include a mix of letters, numbers, and symbols." – **4.b** The user enters another password. – **5.a** The system displays "Invalid phone number format. Please enter a valid phone number." – **5.b** The user enters another valid phone number.
Pre-Condition	• The user must download the application.
Post Condition	• A new user is added to the system. • The user is successfully registered. • The user information is stored in the database

Use case 2: Registration (Log-In)

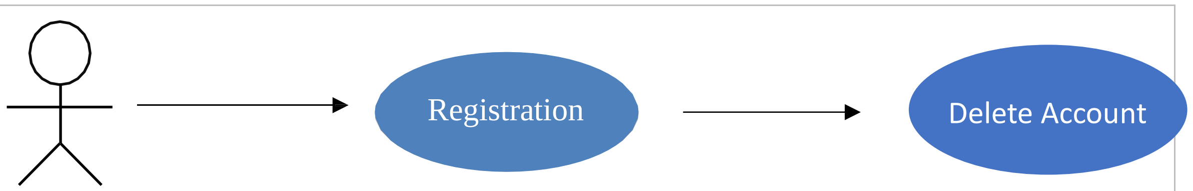


User case Name	Log-In
Actors	User/ Administration (Registered)
Main success scenario	1. The system asks the user/administrator to enter the username and password. 2. The User/administrator enters username and password. 3. The User/administrator clicks on the log-in button. 4. The System checks that the username and password are valid.
Exceptions	1. The user/administrator clicks the login button without filling in all the required information. 2. The username is invalid. 3. The password is invalid. 4. The account is deactivation (the user's account has been deleted)
Actions	– **1a**. The system displays to the user/administrator "Please fill in all required fields." – **1b**. The user fills in the missing information. – **2a.** The system displays to the user/administrator "The username is invalid." – **2b.** The user enters a valid username. – **3a.** The system displays to the user/administrator "Invalid password." – **3b.** The user enters the correct password. – **4a**. The system displays to the user "Your account has been deleted. If you believe this is an error, please contact support for assistance."
Pre-Condition	• The User had to download the application. • The User/administrator must have an account.
Post Condition	• The User/administrator is logged in to the system. • The User/administrator has access to the system functions.

Use case 3: Edit Profile

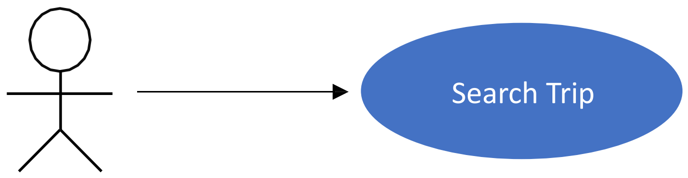
	
User case Name	Edit User Profile
Actors	User (Registered)
Main success scenario	1. The user clicks on the "Edit Profile" option. 2. The system displays the current user profile information. 3. The user clicks on the editing field. 4. The user modified one or more fields such as (Name / Email Address / Phone Number Profile picture) 5. If the validation is successful, the system updates the user's profile information in the database. 6. The system provides a confirmation message indicating that the profile has been successfully updated. 7. The user sees their updated information reflected in the user profile.
Exceptions	1. The user enters invalid data such as (an invalid email format, or incomplete phone number). 2. The user clicks on the change the password field. 3. Cancel Edit
Actions	– **1a.** The system displays error messages next to the relevant fields, informing the user about the specific issue. – **1b.** The user enters/corrects the valid data. – **2a.** The system provides additional fields to enter their current password and the new password. – **3a.** Allow the user to navigate away from the profile editing page without saving changes. – **3b.** The system displays to the user "No changes are applied to the profile."
Pre-Condition	• The user is logged into the application. • The user navigates to the profile editing section within the application.
Post Condition	• A user's profile information is updated in the system's database. • The user can continue using the application with updated profile details.

Use case 4: Delete Account

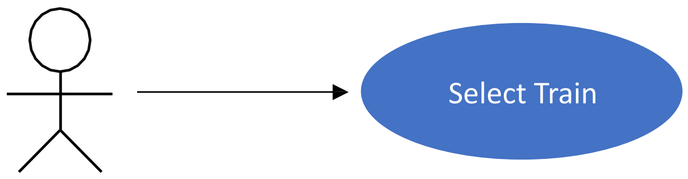


User case Name	Delete User Account
Actors	User (Registered)
Main success scenario	1. The user clicks on the “Delete Account”, located in the profile section. 2. The system prompts the user to confirm their intention to delete the account. 3. The user enters their email address and the password to proceed with the deletion. 4. The system checks the email and the password of the user. 5. The system permanently removes the user’s account and associated data from the system’s database. 6. The system displays a confirmation message “The Account has been successfully deleted.”
Exceptions	1. Invalid Email Address. 2. Invalid Password. 3. Cancel Account deletion
Actions	– **1a.** The system displays an error message to the user “The email address is incorrect.” – **1a.** The User enters the correct email address. – **1b.** The system displays an error message to the user “The password is incorrect.” – **1b.** The user enters the correct password. – **2a.** The User chooses to cancel the deletion process. – **2a.** The system does not proceed with the account deletion and informs the user that their account remains active. “Your account is still active.”
Pre-Condition	• The User is logged into the application. • The User navigates to the account deletion section within the application.
Post Condition	• The User’s account and associated data are permanently deleted from the system’s database. • The user is logged out of the application.

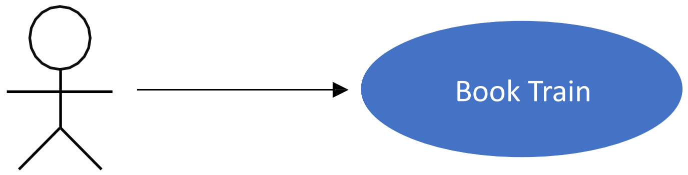
Use case 5: Search Trip

	
User case Name	Search For Trip
Actors	Passenger
Main success scenario	**1.** The passenger chooses from the search options a one-way trip or a round trip. **2.** Fill in the required information. **3.** Clicks on the “Search Trains” button
Exceptions	**1.** The Passenger enters invalid information like a misspelled destination name. **2.** The passenger clicks the “Search Trains” button without filling in the required information.
Actions	– 1.a. The system displays to the passenger an alert that says, “Invalid Information Entered”. – 1.b. The passenger corrects the information entered. – 2. The system displays to the passenger an alert that says, “Please Fill In All The Required Information”. – 2.b Passenger fills in the missing information.
Pre-Condition	• Passenger had to have logged in to their account.
Post Condition	• The results page will be displayed for the passenger.

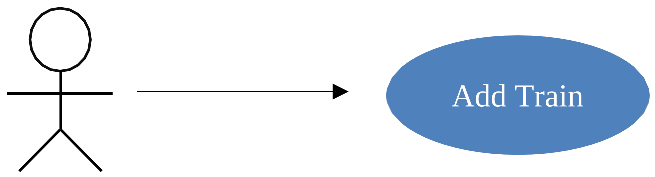
Use case 6: Search Train

	
User case Name	Select Train
Actors	Passenger
Main success scenario	**1.** The passenger browses the displayed results from the search. **2.** The passenger selects the train trip to display its details.
Pre-Condition	• Passenger had to have logged in to their account.

Use case 7: Search Train

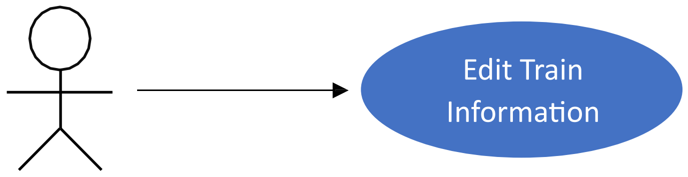
	
User case Name	Book Train
Actors	Passenger
Main success scenario	**1.** After the passenger selects their desired train, they click confirm. **2.** Passenger fills in the payment requirements. **3.** Passenger confirms the transaction. **4.** Train is booked successfully.
Exceptions	**1.** Passenger enters invalid payment information. **2.** Passenger clicks on the ‘Confirm’ button without filling in the required payment information
Actions	– 1.a. The system displays to the passenger an alert that says, “Invalid Payment Information Entered”. – 1.b. The passenger corrects the payment information entered. – A system displays to the passenger an alert that says, “Please Fill In All The Required Information”. – 2.b Passenger fills in the missing payment information.
Pre-Condition	• The Passenger had to have logged in to their account. • The Passenger had to have selected a specific train.
Post Condition	• The passenger will be able to track the train in real-time. • The passenger will be able to display the train’s details. • The passenger will be able to receive announcements concerning their trip

Use case 8: Add Train

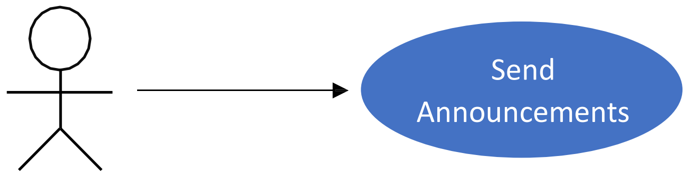


User case Name	Add Train
Actors	Administration
Main success scenario	1. The administrator navigates to the "Add Train" section in the system. 2. The system presents a form with fields to enter information about the train. 3. The administrator fills in the required information: o Train Name o Departure and Arrival Times o Train Schedule o Train Capacity o Tickets Prices 4. The system validates the entered data. 5. If the validation is successful, the system adds the new train record to the system's database. 6. The system displays a confirmation message "New Train has been successfully added." 7. The administrator can view the updated lists of trains, including the newly added train.
Exceptions	1. The administrator clicks "Add Train" without filling in the required information. 2. Duplicated Identifier. 3. Cancel add train process.
Actions	– 1a. The system displays to the administrator "Please fill in all required fields." – 1b. The administrator enters all the required information. – 2a. The system displays an error message to the administrator "The Train you entered already exists." – 2b. The administrator entered another train that did not exist. – 3a. The administrator chooses the cancel button to add a train. – 3b. The system discards any entered data and returns to the previous state.
Pre-Condition	• The administrator is logged into the application. • The administrator has the necessary permission to ass a new train.
Post Condition	• The new train record is added to the system's database. • The administrator can see the updated list of trains, including the newly added train.

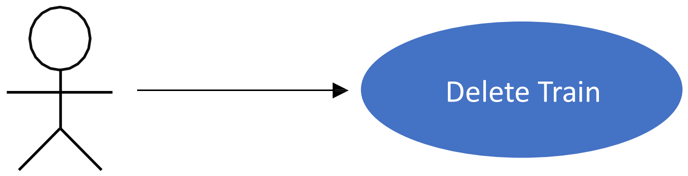
Use case 9: Edit Train

	
User case Name	Edit Train Information
Actors	Administration
Main success scenario	1. The administrator navigates to the “Edit Train” section in the system. 2. The system presents an interface with fields pre-fields with the current information of the selected train. 3. The administrator selects the train they want to edit by entering the train identifier. 4. The system retrieves and displays the current information of the selected train including (Train Name, Departure and Arrival Times, Train Schedule, Train Capacity, and Train Ticket) 5. The administrator updates the train information. 6. The system validates the entered data to ensure it meets any specified requirements. 7. If the validation is successful, the system updates the train information in the system’s database. 8. The system displays a confirmation message “The Train Information has been successfully updated.” 9. The administrator can view the updated information for the edited train.
Exceptions	**1.** Train identifier not found. **2.** Invalid data. **3.** Cancel Edit Train.
– **Actions**	– 1a. The system displays an error message to the administrator “The Train you entered is not found.” – 1b. The administrator entered a valid train identifier. – 2a. The system displays an error message next to the relevant fields, informing the administrator about the specific issues. – 2b. The administrator corrects the errors before proceeding. – 3a. The administrator clicks on the cancel editing train button. – 3b. The system discards any entered changes and returns to the previous state.
Pre-Condition	• The administrator is logged into the system. • The administration navigates to the account deletion section within the application.
Post Condition	• The information of the selected train is updated in the system’s database. • The administrator can view the updated information for the edited train.

Use case 10: Send Announcements

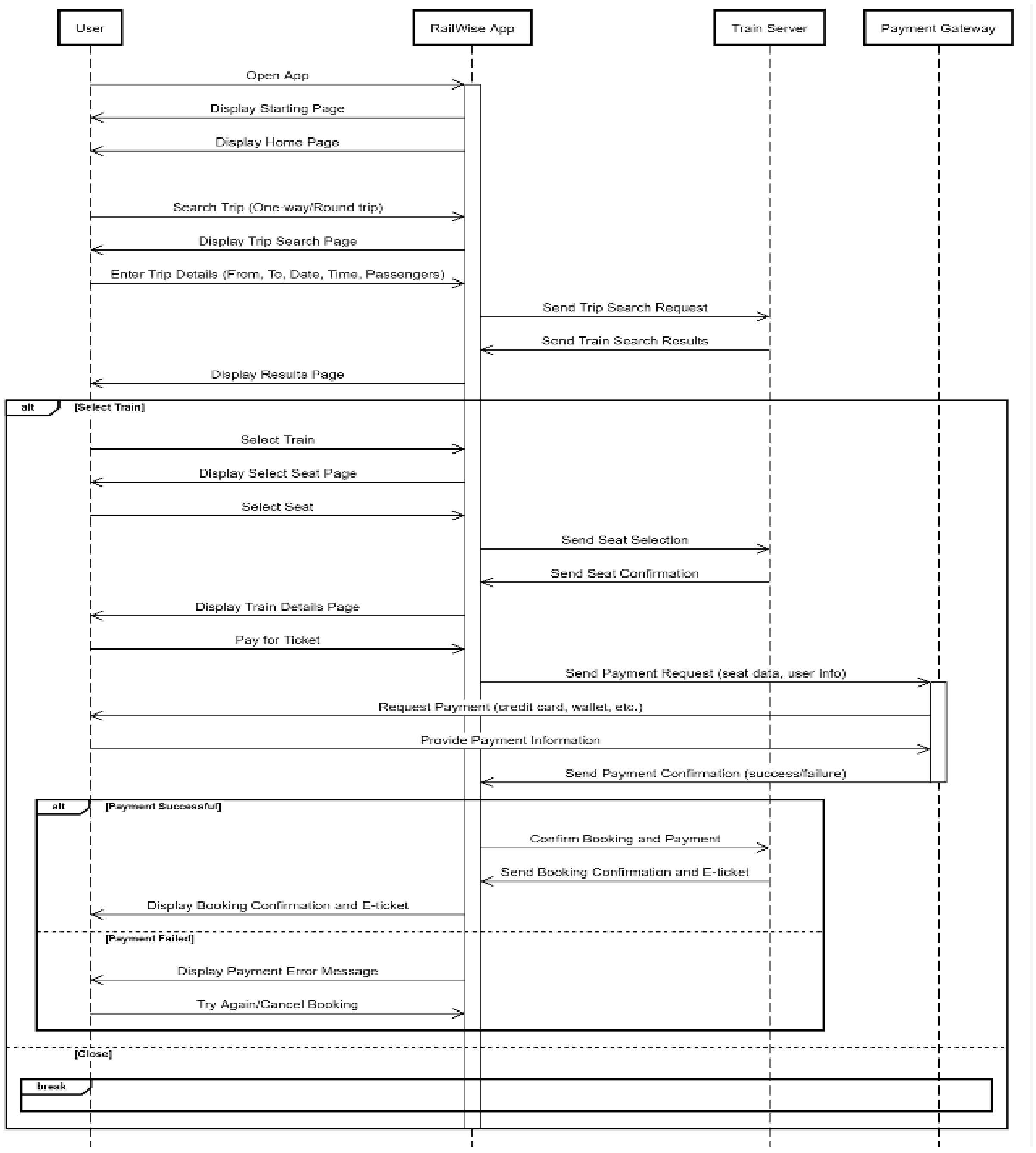
	
User case Name	Send Announcements
Actors	Administration
Main success scenario	1. The administrator navigates to the "Sent Announcements" section. 2. The system presents a form with fields to enter the required information: o Passengers Phone Number o Message Content 3. The administrator enters the target passenger's phone number and the message content. 4. The system validates the entered data to ensure it meets any specified requirements. 5. If the validation is successful, the system sends the announcement to the specified passenger's phone number. 6. The system displays a confirmation message "Announcement has been successfully sent." 7. The administrator can view a log of sent announcements and their status.
Exceptions	**1.** Empty Message content. **2.** Cancel Announcement.
Actions	- 1a. The system displays an error message "Content is Empty." - 1b. The administrator enters a message content. - 2a. The administrator chooses to cancel the announcement. - 2b. The system discards any entered data and returns to the previous state.
Pre-Condition	• The administrator is logged into the system. • Passengers have registered phone numbers in the system
Post Condition	• The announcement is sent to the specified passenger's phone number. • The administrator can view a log of sent announcements and their status.

Use case 10: Delete Train

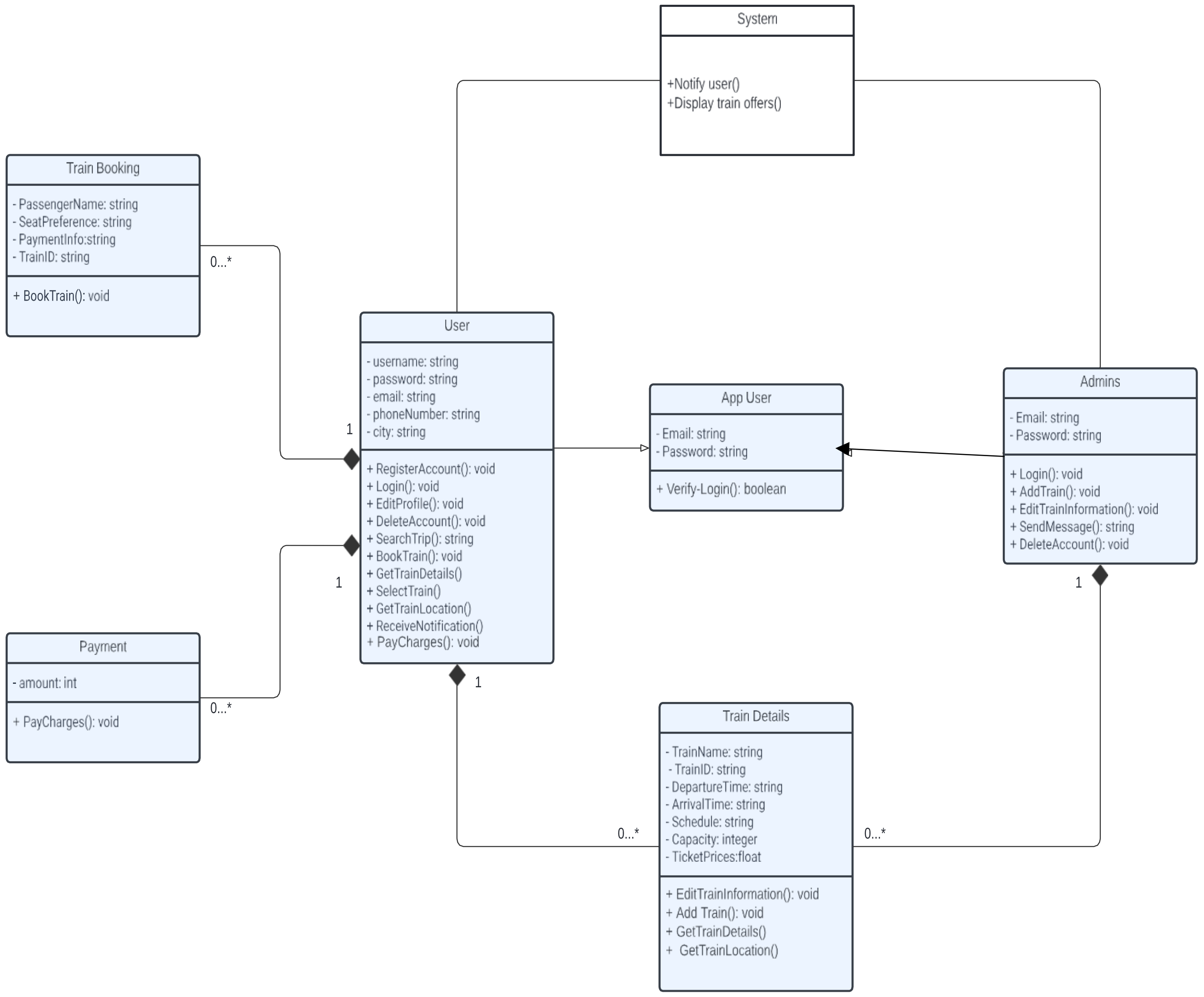


User case Name	Delete an existing train from the system
Actors	Administration
Main success scenario	1. The administrator navigates to the "Delete Train" section in the system. 2. The system presents a list of existing trains with options for deletion. 3. The administrator selects the train they want to delete. 4. The system prompts the administrator to confirm their intention to delete the selected train. 5. The administrator confirms the deletion by clicking a "Delete" or similar button. 6. The system removes the selected train and its associated information from the system's database. 7. The system displays a confirmation message, indicating that the train has been successfully deleted. 8. The administrator can view the updated list of trains, excluding the deleted train.
Exceptions	**1.** Cancel Deletion. **2.** Train not found.
Actions	– 1a. The administrator chooses to cancel the deletion process. – 1b. The system displays a message “The selected train remains in the system.” – 2a. The system displays a message “The selected train is not found”
Pre-Condition	• The administrator is logged into the system. • The administrator has the necessary permissions to delete trains. • There are existing trains in the system.
Post Condition	• The selected train is permanently deleted from the system's database. • The administrator can view the updated list of trains, excluding the deleted train.

6.2 Sequence Diagram:


User
RailWise App
Train Server
Payment Gateway
Open App
Display Starting Page
Display Home Page
Search Trip (One-way/Round trip)
Display Trip Search Page
Enter Trip Details (From, To, Date, Time, Passengers)
Send Trip Search Request
Send Train Search Results
Display Results Page
alt
[Select Train]
Select Train
Display Select Seat Page
Select Seat
Send Seat Selection
Send Seat Confirmation
Display Train Details Page
Pay for Ticket
Send Payment Request (seat data, user info)
Request Payment (credit card, wallet, etc.)
Provide Payment Information
Send Payment Confirmation (success/failure)
alt
[Payment Successful]
Confirm Booking and Payment
Send Booking Confirmation and E-ticket
Display Booking Confirmation and E-ticket
[Payment Failed]
Display Payment Error Message
Try Again/Cancel Booking
[Close]
break

6.3 Class Diagram:



Video Link:
https://nileuniversity.sharepoint.com/sites/SoftwareEngineering559/_layouts/15/stream.aspx?id=%2Fsites%2FSoftwareEngineering559%2FShared%20Documents%2FGeneral%2FRecordings%2FDeliverable%204%20Meeting%2D20231216%5F223010%2DMeeting%20Recording%2Emp4&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview

https://nileuniversity.sharepoint.com/sites/SoftwareEngineering559/_layouts/15/stream.aspx?id=%2Fsites%2FSoftwareEngineering559%2FShared%20Documents%2FGeneral%2FRecordings%2FDeliverable%204%20Meeting%2D20231216%5F224508%2DMeeting%20Recording%2Emp4&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview