

**An-Najah National University**



**Faculty of Engineering and Information Technology  
Management Information Systems Department**

**Ride-Sharing Mobile Application  
Wasselni**

**Supervised by:**

Dr. Ahmad Shraideh

Prepared by:

Abdullrahman Ramadan

Odai Awartani

Hamza Qawareq

## Contents

.1 Project overview .....	5
1.1 introduction: .....	5
1.2 Objectives:.....	6
1.3 Characteristics:.....	7
1.3.1 User-friendly Interface.....	7
1.3.2 Ride Matching.....	7
1.3.3 Booking System .....	7
1.3.4 Cost Efficiency .....	7
1.3.5 Rating and Reviews: .....	7
1.3.6 Real-time Location Tracking.....	7
1.3.7 Safety Features.....	7
1.3.8 Real-Time Chat: .....	7
1.3.9 Barrier Feedback System:.....	7
2. App Architecture:.....	8
3. Target audience: .....	9
4. Use Case .....	10
5. 3. User Journey Through the App .....	11
5.1 The logo: .....	11
5.2 App description: .....	12
5.3 Authentication: .....	14
5.3.1 Sign-Up Success.....	15
5.3.2 Sign-Up Fail Scenarios .....	16
5.3.3 Sign-In .....	23
5.3.4 Forgot Password .....	26
5.3.5 Reset Password .....	28
5.4 Header .....	31
5.5 Notification Page .....	32
5.6 Profile Page .....	33

5.7 SideMenu .....	34
5.8 Profile Page Edit .....	35
5.8.1 Changing phone number .....	36
5.8.2 Changing Password.....	37
5.8.3 Change Car Type and Seats Number .....	38
5.9 Language Button .....	39
5.10 Location Management .....	40
5.11 Share Button .....	42
5.12 Applicaton Rate .....	43
5.13 Privacy Policy .....	44
5.14 Help & Support .....	45
5.15 Home Page.....	47
5.15.1 Become A Driver .....	48
5.15.2 Track Page .....	53
5.15.3 Suggested Rides .....	56
5.15.4 View All Button.....	58
5.15.5 Create Ride.....	59
5.15.6 Book Ride .....	62
5.15.7 Driver Accept or Reject the Passenger .....	64
5.15.8 Driver Starts a Ride.....	67
5.15.9 Driver Communication with Passenger .....	68
5.15.10 Passenger Check-In .....	69
5.15.11 Passenger Check-Out and Rate the Driver.....	70
5.15.12 Driver Receives the Rating and Ends the Ride .....	71
5.16 Barriers Page .....	72
5.16.1 Viewing Barrier Status .....	73
5.16.2 Updating Barrier Status: .....	74
5.17 Search Page .....	75
5.18 Rides Page .....	76

5.19 Chat Page.....	77
5.20 Driver Profile from Passenger Perspective .....	78
5.21 Passenger Profile from Driver Perspective .....	79
6. Admin Dashboard.....	80
6.1 Driver Applications and .....	80
6.2 User Management .....	81
6.3 Rides Management.....	82
6.4 Reports.....	83
7. Technology used.....	84
7.1 Frontend (User Interface): .....	84
7.2 Backend (Server-Side):.....	84
7.3 Database: .....	84
7.4 APIs and Integrations: .....	85
7.5 Additional Tools and Services: .....	85
8. Frequently Asked Questions (FAQ): .....	85
9. Future Plans: .....	87
10. Conclusion.....	89

# **Wasselni: A Ride-Sharing App for the Palestinian Community**

## **1. Project overview**

### **1.1 introduction:**

In northern Palestine and across the country, daily transportation has become a growing concern, especially for students and employees who struggle to get to universities, workplaces, or essential services on time. The situation has worsened due to the ongoing war, frequent road closures, and heavily guarded military checkpoints that cause unpredictable delays. Public transportation is often unreliable, scarce, or located far from residential areas, making commuting a daily burden. Having personally faced these difficulties, we developed **Wassleni**, a mobile application designed to connect people heading in the same direction by offering or joining rides. The app allows car owners to create trips, and nearby users can view, book, and join these rides by paying a reasonable fee—creating a system that is mutually beneficial, practical, and community-driven. Wassleni mainly targets students, employees, and the general public who lack access to reliable transportation, aiming to make travel easier, faster, and more organized. Each user has a personalized profile with a photo, preferences, and details that help build trust, while drivers can customize their ride settings—such as allowing or disallowing smoking, music, or children, and selecting the preferred gender of passengers. Real-time tracking enables users to monitor their location throughout the journey, view start and end points on the map, and stay informed about nearby trips. In addition, drivers can update the status of military checkpoints they pass through—whether open, closed, or partially blocked—helping others avoid delays. To bring Wassleni to life efficiently across both Android and iOS platforms, we used **React Native**, a powerful cross-platform framework known for its performance, flexibility, and developer-friendly features. Through Wassleni, we aim to provide a transportation solution that not only meets real needs in Palestine but also fosters safety, convenience, and connection among users.

## **1.2 Objectives:**

Wassleni aims to provide a practical and reliable transportation solution for students, employees, and the general public who face daily commuting difficulties in Palestine. The app connects people traveling in the same direction, allowing drivers to offer rides and passengers to join easily, saving time and reducing travel costs. It improves safety and comfort by enabling drivers to set personal ride preferences, such as passenger gender, music, and smoking rules, while passengers can build detailed profiles to increase trust and ensure compatibility. Users can track their trips in real-time, knowing exactly where the ride starts and ends, and receive live updates on military checkpoint statuses to avoid delays. Additionally, drivers can update the app with checkpoint conditions as they pass through, helping others plan their journeys better. Wassleni seeks to build a strong community based on cooperation and social interaction, promote environmentally friendly travel through car sharing, and empower drivers to earn extra income by providing a valuable service. The app also offers features to enhance user convenience, such as notifications, easy payment methods, and clear communication channels between drivers and passengers. Developed with React Native, the app ensures a smooth experience on both Android and iOS devices. By continuously gathering user feedback and improving features, Wassleni strives to reduce reliance on unreliable public transportation and transform commuting in Palestine into a safer, more convenient, and user-centered experience.

## **1.3 Characteristics:**

### **1.3.1 User-friendly Interface:**

Wassleni offers a simple and easy-to-use interface, making it easy for users to find and book rides or create their own.

### **1.3.2 Ride Matching:**

The app connects passengers with drivers heading in the same direction, making it easier to share rides and save on transportation costs.

### **1.3.3 Booking System:**

Users can reserve a seat in advance for a ride, ensuring a smooth and organized travel experience.

### **1.3.4 Cost Efficiency:**

Passengers can save money by sharing rides, and drivers earn money to cover their travel expenses, making the system beneficial for both sides.

### **1.3.5 Rating and Reviews:**

After every ride, passengers can rate their drivers, providing feedback and helping build a sense of trust within the community.

### **1.3.6 Real-time Location Tracking:**

The app allows users to track the real-time location of their ride, ensuring timely pickups and arrivals.

### **1.3.7 Safety Features:**

The app ensures safety by verifying users' profiles and implementing secure ride-sharing guidelines.

### **1.3.8 Real-Time Chat:**

Wasselni offers a real-time chat feature that allows passengers and drivers to communicate instantly before and during the ride—making coordination easier and trips smoother.

### **1.3.9 Barrier Feedback System:**

Users can report and view real-time updates on road barriers and checkpoints. This feature helps passengers check the status of potential roadblocks before starting their trip, improving planning and safety.

## **2. App Architecture:**

This section outlines the overall structure of the application, explaining how the main components interact with each other to deliver the core functionality. The **user interface (UI)** is meticulously designed using **Figma** to ensure a **clean, intuitive, and user-friendly experience**, incorporating essential screens such as **login, ride search, booking, map tracking, and reviews**. It is developed with **React Native**, enabling **cross-platform support** for both **Android and iOS devices**, which maximizes accessibility and performance. The **backend servers**, currently under consideration, will be responsible for managing critical operations including **ride creation, ride reservations, real-time notifications, and secure payment processing**. These servers will later be integrated with the **database** and **APIs** to facilitate **efficient data processing and seamless communication** between the app's components. For **user authentication and secure session management**, the app employs **Clerk**, ensuring users' accounts and data remain protected. **Firebase** is used for handling **image uploads, cloud storage**, and potentially **real-time updates**, enhancing the app's responsiveness and media management capabilities. The **database technology**, still being evaluated, will securely store important information such as **user profiles, ride histories, user reviews, and payment records**, guaranteeing **fast and reliable access** to data critical for the app's functionality and user satisfaction.

### **3. Target audience:**

Wasselni is designed for people across Palestine who seek more efficient, affordable, and socially connected ways to commute. Whether its university students traveling to campus, workers heading to their jobs, or anyone needing a ride in areas with limited public transportation, Wasselni offers a practical alternative to traditional commuting methods.

The app is particularly relevant for younger users, such as students from An-Najah National University and other universities, who are often quick to adopt new technologies and are open to modern, user-friendly solutions. At the same time, Wasselni serves the broader public—drivers with empty seats and passengers in need of transport—helping to reduce travel costs, ease road congestion, and foster a stronger sense of community through shared rides.

By targeting the Palestinian population as a whole, Wasselni aims to build a culture of smart transportation that benefits everyone—making daily travel across the country more organized, economical, and connected.

## 4. Use Case

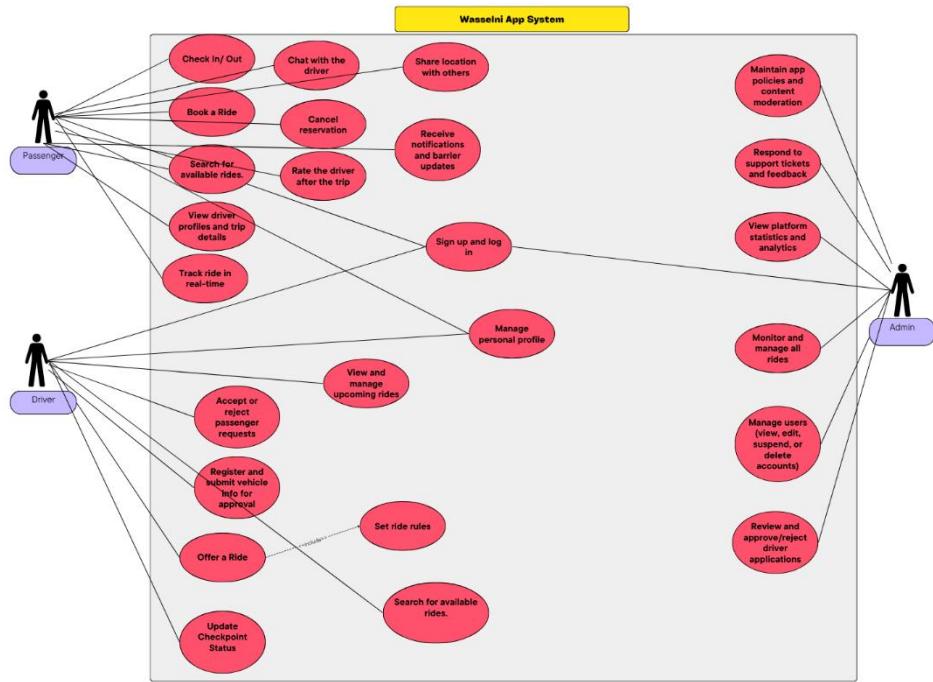


Figure 1 Wasselni Use Case.

## **5. 3. User Journey Through the App**

### **5.1 The logo:**

Starting with the logo, we designed it to reflect the core concept behind Wassleny people sharing rides while traveling the same route. The visual elements of the logo are meant to symbolize streets, movement, and the idea of connection between individuals heading in the same direction. It represents inclusivity, as the app is open to both males and females, and emphasizes the shared journey aspect that lies at the heart of our service. The logo was carefully chosen to communicate trust, community, and the smart use of everyday transportation in a modern and appealing way.



*Figure 2 Splash Screen.*

## 5.2 App description:

This section provides a visual and descriptive overview of the Wasselni app, helping users understand how to navigate and use it effectively. The following images highlight the core features and the intuitive user interface.

When users first launch the app, they are welcomed with a brief tutorial consisting of three interactive slides that introduce the app's key features. These slides cover:

- **What Wasselni is:** an overview of how the app works and how it helps save time through efficient ride-sharing.
- **The tracking feature:** explaining how users can track their rides in real time for better coordination and safety.
- **Barrier updates:** showing how users can view and report road barriers to avoid delays and improve route planning.



Figure 3 Onboarding 1.



Figure 4 Onboarding 2.



Figure 5 Onboarding 3.

To ensure a smooth experience, users can choose to skip the tutorial at any time by tapping the "Skip" button, allowing returning users to jump directly into the app.

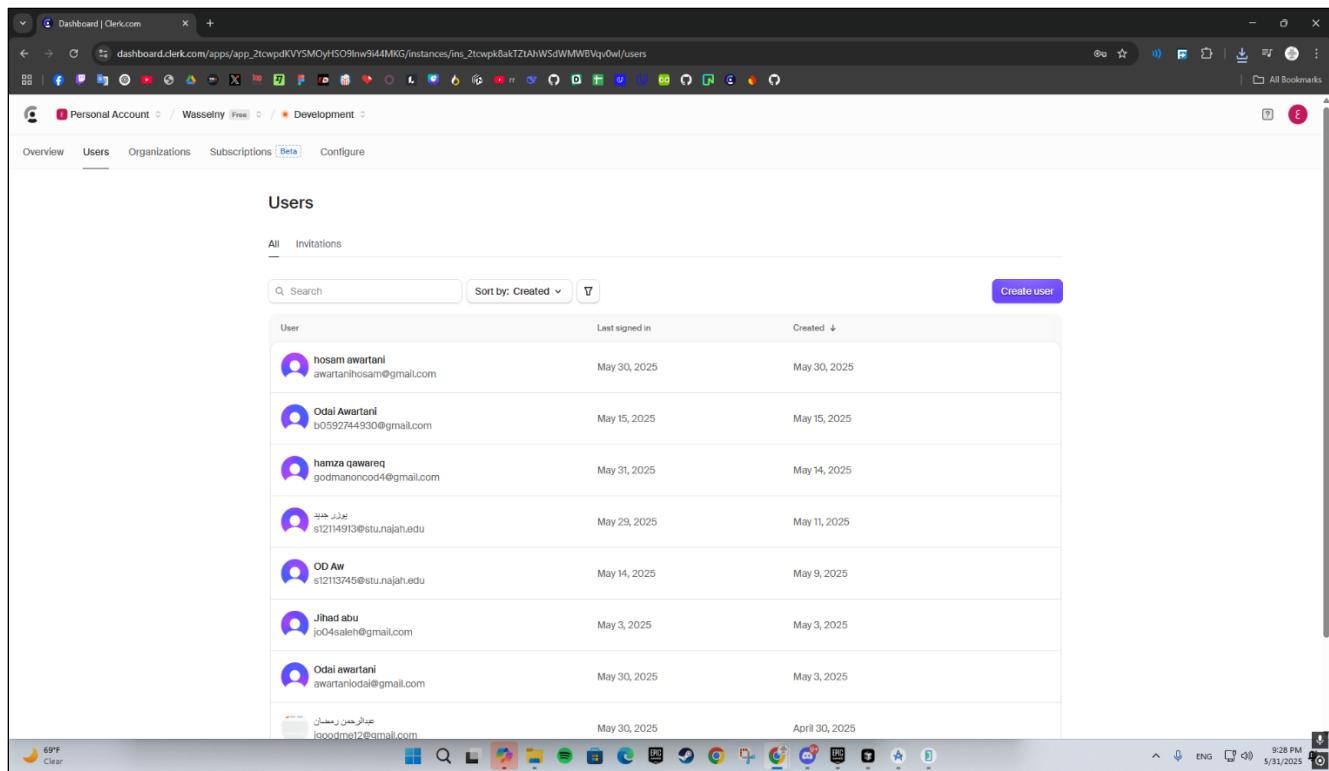
At the top of the screen, users will also find a language toggle represented by flags. Tapping the Palestinian flag switches the app to Arabic, while the American flag changes the interface to English. This easy language selection ensures that users can interact with the app in their preferred language from the start.

By combining an optional tutorial with accessible language switching and a clean interface, Wasselni provides a seamless and user-friendly onboarding experience for everyone.

### 5.3 Authentication:

Privacy and security are crucial for Wasselni users, especially when it comes to protecting personal data and ensuring safe ride-sharing. To maintain this, each user must create a personal account before using the app. This helps prevent unauthorized access and ensures that features like real-time location tracking function securely. Users can log in using their email and password if they already have an account. For new users, the sign-up process is simple and secure they are required to provide their name, phone number, email, password, gender, and work industry. If a user enters an email that is not yet registered, a verification code is sent to their inbox. Once they enter the code, their identity is confirmed, and they gain full access to the app's features.

Once your information is confirmed and the verification code is entered, your data is securely stored using Clerk, an API integrated with our app to manage user authentication safely. Clerk ensures that login credentials and identity verification are handled according to high security standards. However, while Clerk manages authentication, the actual user profile data including name, phone number, gender, and work industry is securely stored in Firebase. This separation ensures both secure account access and flexible, scalable storage of user information. The structure of how accounts are managed and stored is shown in the following image.



The screenshot shows the Clerk dashboard interface. At the top, there is a navigation bar with links for Overview, Users, Organizations, Subscriptions, and Configure. The main area is titled 'Users' and displays a table of user data. The columns in the table are 'User', 'Last signed in', and 'Created'. The table lists eight users with the following details:

User	Last signed in	Created
hosam awartani awartanihosam@gmail.com	May 30, 2025	May 30, 2025
Odal Awartani b0592744930@gmail.com	May 15, 2025	May 15, 2025
hamza qawareq godmanoncod4@gmail.com	May 31, 2025	May 14, 2025
عمر جعفر s12114913@stu.najah.edu	May 29, 2025	May 11, 2025
OD Aw s12113745@stu.najah.edu	May 14, 2025	May 9, 2025
Jihad abu jo04saleh@gmail.com	May 3, 2025	May 3, 2025
Odal awartani awartaniodal@gmail.com	May 30, 2025	May 3, 2025
عبد الرحمن رمضان igoodme12@gmail.com	May 30, 2025	April 30, 2025

Figure 6 Clerk Overview.

### 5.3.1 Sign-Up Success

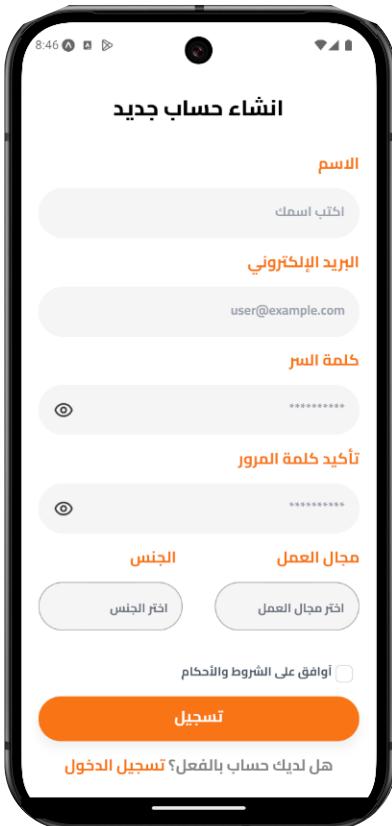


Figure 7 Sign up.

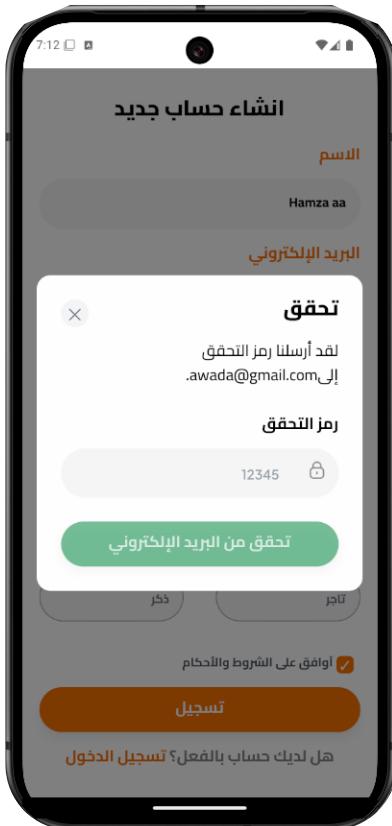


Figure 8 Sign-up Verification Code.

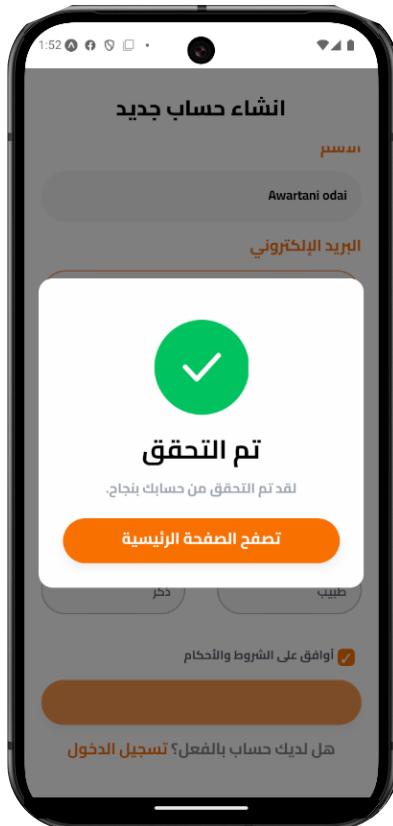


Figure 9 Sign Up Success.

### 5.3.2 Sign-Up Fail Scenarios

During the sign-up process, several validation rules are enforced to ensure users enter correct and secure information. A sign-up attempt will fail in the following cases:

#### 1. Invalid Phone Number:

- The phone number must be exactly **9 digits**.
- If the user enters more or fewer digits, or includes non-numeric characters, an error message is shown.



Figure 10 Sign-up Phone number alert.

## 2. Email Already Used:

- If the entered email address is already associated with an existing account, the system will not allow registration.
- A message like "*This email is already in use*" appears.

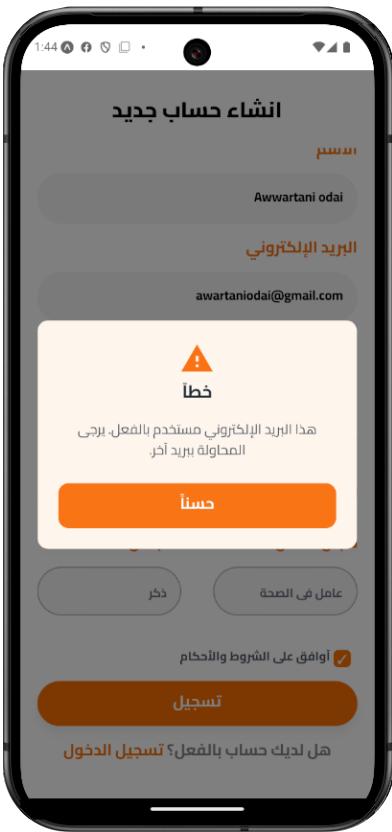


Figure 11 Sign-up Email Already Used Alert.

### 3. Invalid Email Format:

- The email must contain a valid domain (e.g., [@gmail.com](mailto:@gmail.com), [@outlook.com](mailto:@outlook.com)).
- If the email doesn't follow this format, such as missing "[@example.com](mailto:@example.com)", it will be rejected.



Figure 12 Sign-Up Email Wrong Format.

#### 4. Password Mismatch:

- The **confirm password** must exactly match the **original password**.
- If they are different, the user is prompted to correct them.



Figure 13 Password Mismatch.

## 5. Empty Fields:

- All required fields (name, phone number, email, password, gender, work industry) must be filled.
- If any field is left empty, the form submission will be blocked.



Figure 14 Sign-up Fill All Field Alert.

## 6. Weak Password:

- The password must be at least 8 characters long.
- It must contain both letters and numbers.
- If this condition is not met, an alert will appear prompting the user to enter a stronger password, as shown in the image.

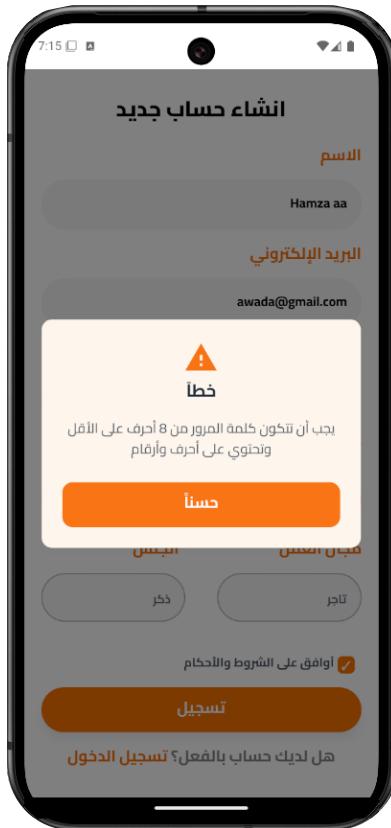


Figure 15 Sign-up Strong Password Alert.

## 7. Terms and Conditions Not Accepted:

- Users must check the box to **accept the terms and conditions**.
- If they skip this step, sign-up will not proceed.

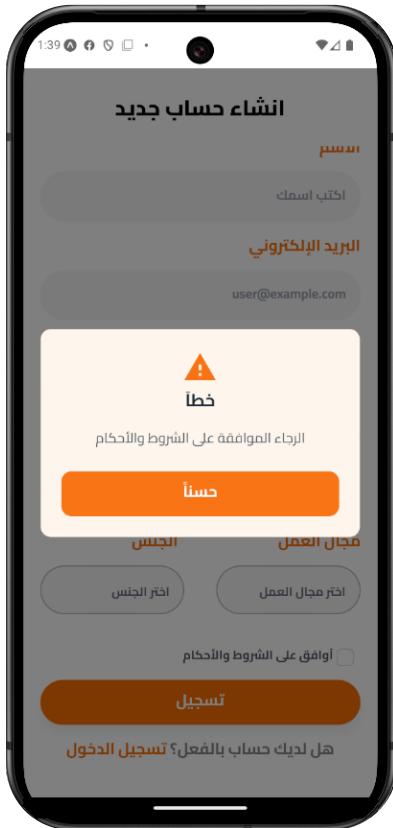


Figure 16 Sign-up Accept Terms Alert.

### 5.3.3 Sign-In

The sign-in functionality allows users to access the application by entering their email and password. The interface is designed to provide a seamless login experience while ensuring proper error handling for incorrect credentials. Below are key aspects of the sign-in process, illustrated with relevant screenshots.

The sign-in screen provides fields for users to input their email and password. It includes a 'Log In' button to submit the credentials and a visibility toggle icon next to the password field to show or hide the input. The design features a background image of a car wheel, aligning with the application's theme.

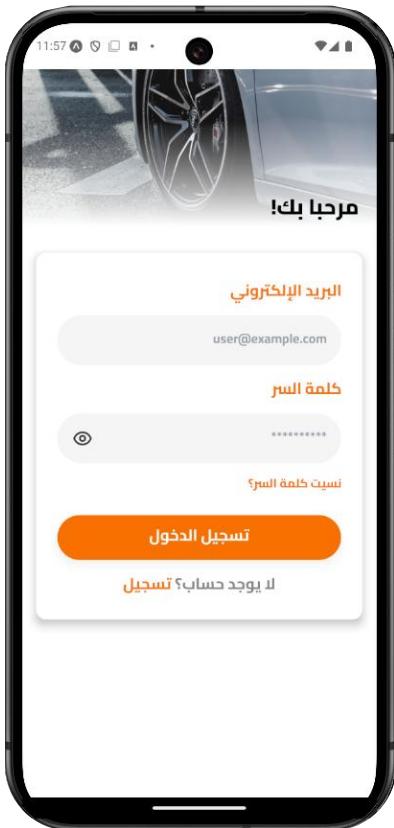


Figure 17 Sign-in.

### 1. Email Not Found:

- If the user enters an email that **does not exist in the system**, the app will display an error such as:  
“*No account found with this email.*”



Figure 18 Sign-in Couldn't find Email Alert.

## 2. Incorrect Password:

- If the entered password does **not match the one linked to the email**, the app will reject the login attempt.
- An error like “*Incorrect password. Please try again.*” is shown.
- This helps prevent unauthorized access and prompts the user to try again or reset their password



Figure 19 Sign-in Incorrect Password Alert.

### 5.3.4 Forgot Password

If a user forgets their password, Wasselny provides a simple and secure password recovery process to help them regain access:

- The user taps on the "**Forgot Password?**" link on the sign-in screen.
- They are prompted to enter their **registered email address**.
- If the email exists in the system, a **verification code or reset link** is sent to that email.
- The user enters the verification code or clicks the link and is redirected to a **password reset screen**.
- They create a **new password** and confirm it.
- Once successfully submitted, they are redirected to the login screen and can sign in with the new password.



Figure 20 Forgot Password Button.

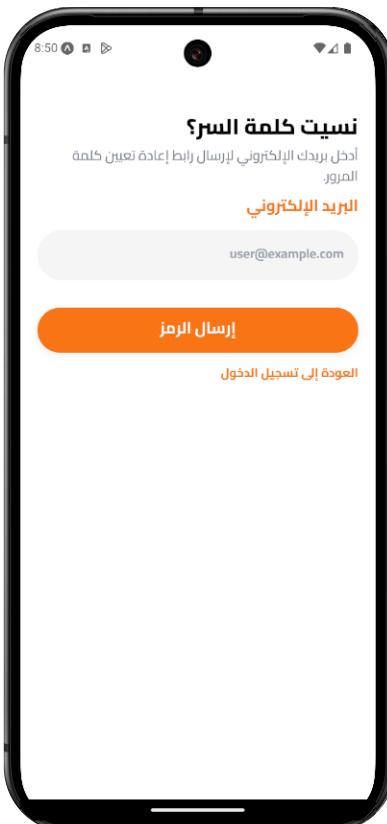


Figure 21 Forgot Password.

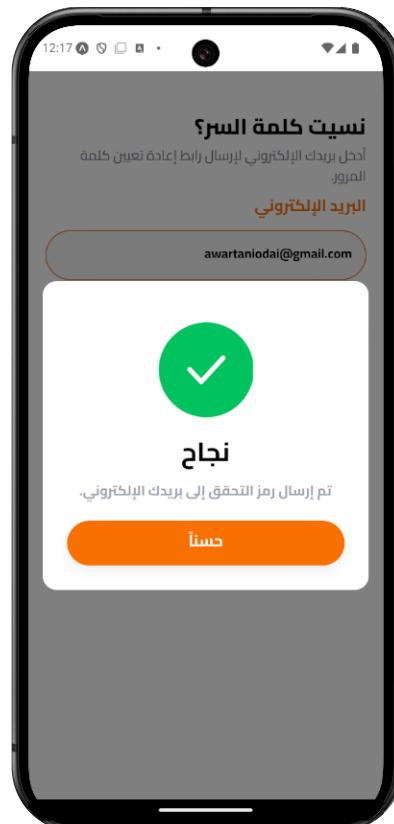


Figure 22 Forgot Password Success.

If the user enters an email address that does not exist in the system, an alert appears informing them that the email is not recognized. This helps prevent confusion and guides the user to either check for typos or proceed to create a new account if they haven't registered yet.



Figure 23 Wrong Email Error.

### 5.3.5 Reset Password

After verifying the reset code sent to their email, the user is directed to the **Reset Password** screen. Here, they are required to enter a **new password** and then confirm it in the second field. Both passwords must match to proceed. Once submitted, the password is updated successfully, and the user is redirected to the **Sign In** page to log in with their new credentials.



Figure 24 Reset Password Page.



Figure 25 Reset Password Success.

## Alert: Fill All Fields

If the user tries to reset the password without filling in both the new password and confirmation fields, an alert appears prompting them to complete all required fields before proceeding.



Figure 26 Reset Password Fill All Field Error.

## Alert: Incorrect Verification Code

If the user enters a verification code that doesn't match the one sent to their email, an error alert is shown to indicate that the code is invalid.



Figure 27 Reset Password Verification Code Expired.

## 5.4 Header

At the top of the screen, a universal header is displayed consistently across all tabs of the app. This header includes a side menu button (to be detailed later), the user's profile picture — which redirects to the Profile Page when tapped — and a notification icon that opens the Notification Center, where users can view alerts and reminders such as trip updates or system messages.



Figure 28 Application Header.

## 5.5 Notification Page

Notifications and communication tools are essential for keeping you informed and in control of your ride-sharing experience. You'll receive real-time updates for booking requests, cancellations, trip changes, passenger arrivals, new ratings, and more via in-app alerts, push notifications, email, or SMS, based on your preferences. A built-in chat system allows direct, secure communication with passengers for coordinating pickups or sharing special instructions, and you're notified instantly of new messages. Even minor updates like seat changes or trip status edits are reported. You can fully customize which notifications you receive, ensuring a smooth, efficient, and safe experience every time.



Figure 29 Notification Page.

## 5.6 Profile Page

The Profile Page provides users with a centralized and detailed overview of their personal information and activity within the app. It features a clean interface with essential elements such as the user's profile picture, name, email, overall rating, and total number of reviews. Users can easily manage their profile through quick-access buttons for editing details, viewing trip history, accessing the dashboard, or logging out. Expandable sections reveal comprehensive information, including detailed ratings from passengers, driver and vehicle information, car image, and account data such as phone number and registration date. This layout ensures a user-friendly experience that promotes transparency, trust, and ease of access to important data.



Figure 30 Profile Page 1.



Figure 31 Profile Page 2.

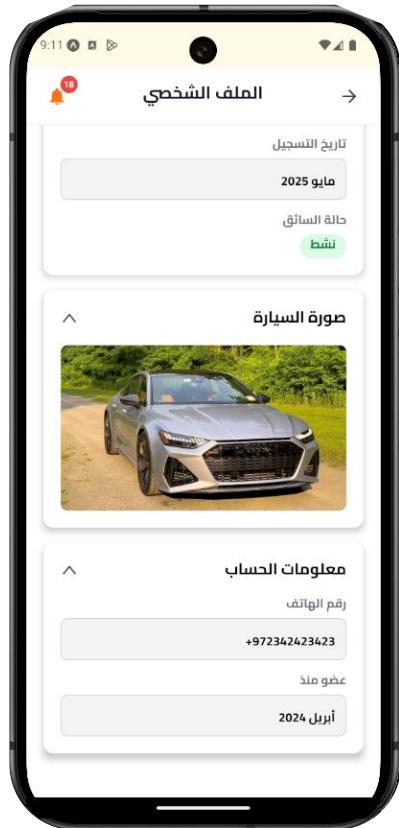


Figure 32 Profile Page 3.

## 5.7 SideMenu

The settings page in the application is a central place where users can personalize their experience and manage various important preferences. It includes features like editing your profile, changing the app language to English, and adjusting your location settings. You can also track your rides, share your live location through a QR code, and start a ride directly from the settings. The page allows you to set up road checkpoints, share the app with other platforms, view and give ratings, and access privacy policies. Additionally, it offers help and support options, and at the bottom, you'll find the ability to sign out of your account. This well-organized page ensures users have full control, a smooth experience, and access to everything they need in one place.

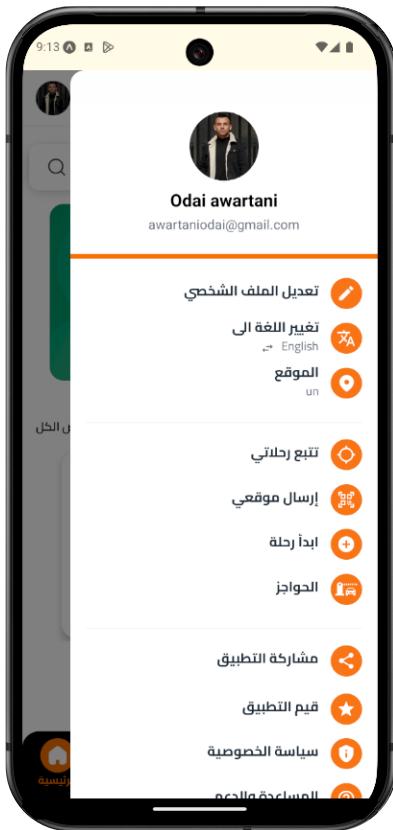


Figure 33 side Menu 1.

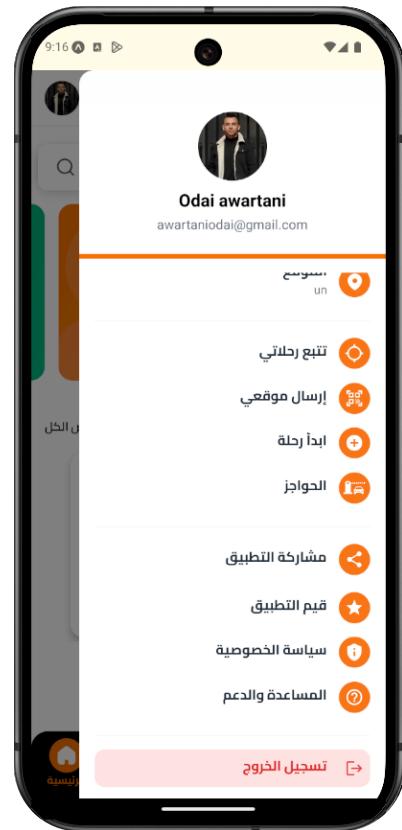


Figure 34 Side Menu 2.

## 5.8 Profile Page Edit

The Profile Page allows users to manage their account details with flexibility, accessible via the profile picture in the app's universal header. Users can update their profile picture, name, email, password, and phone number, the latter being essential for others to contact them. The page supports two perspectives: the standard user view and the driver view. For drivers, additional information is displayed, including the car picture and the number of car seats, ensuring they can provide relevant details for passengers. This dual-perspective design ensures both users and drivers can efficiently customize their profiles to suit their roles within the ride-sharing app.

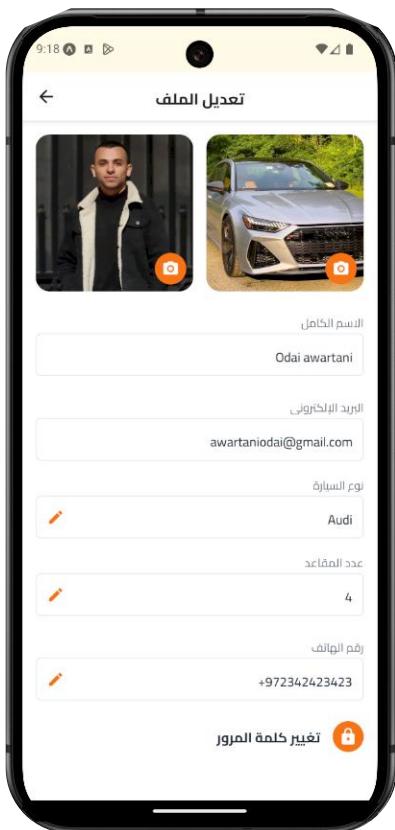


Figure 35 Profile Page Edit Driver Perspective.

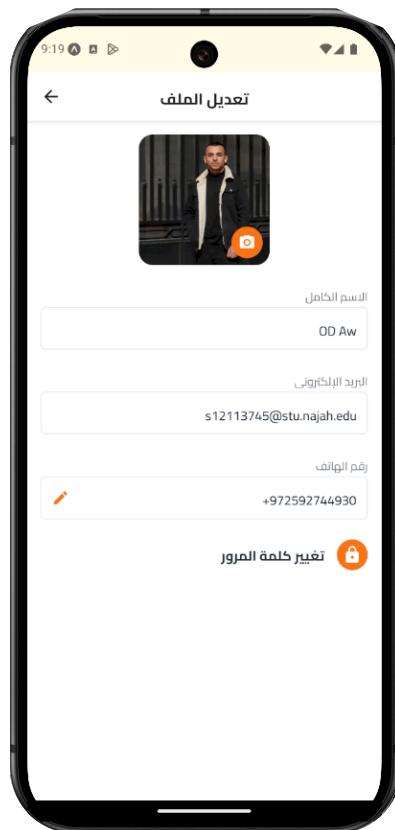


Figure 36 Profile Page Edit User Perspective.

### 5.8.1 Changing phone number

To change my phone number, I simply need to tap the pencil icon, enter the new number, and then I'll receive a success message upon saving. Note that the "Save" button remains disabled unless exactly 9 digits are entered, ensuring the phone number is valid.

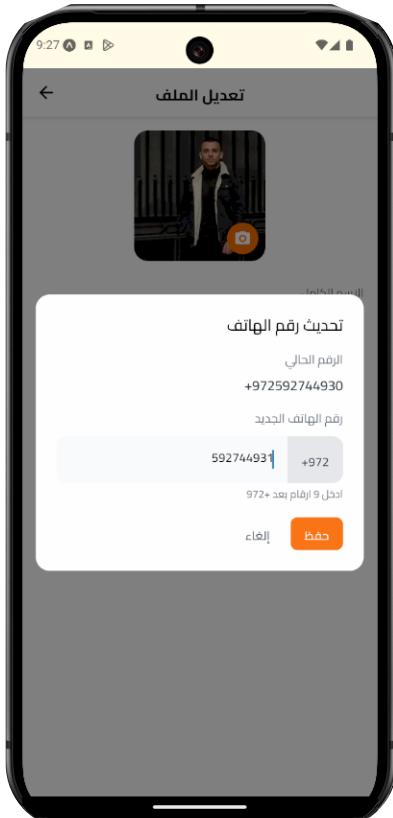


Figure 37 Phone Number model.



Figure 38 Changing Phone Number Success.

### 5.8.2 Changing Password

When a user wants to change their password, they simply need to press the "**Change Password**" button. Three fields will appear: the user must first enter their current password to verify their identity. Then, they need to enter the new password twice to confirm it and ensure they typed it correctly.



Figure 39 Password Changing Model.



Figure 40 Password Mismatch Error.



Figure 41 Password Changed Successfully.

### 5.8.3 Change Car Type and Seats Number

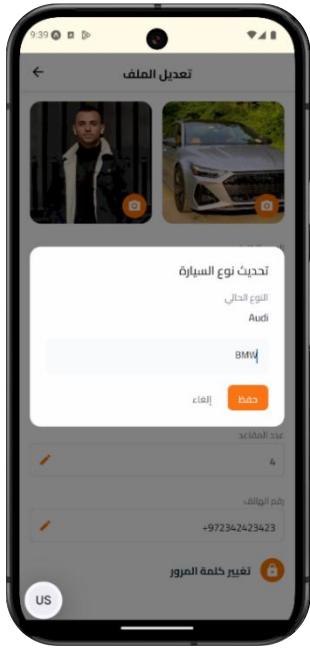


Figure 42 Car Type Change Model.



Figure 43 Car Type Changed Successfully.

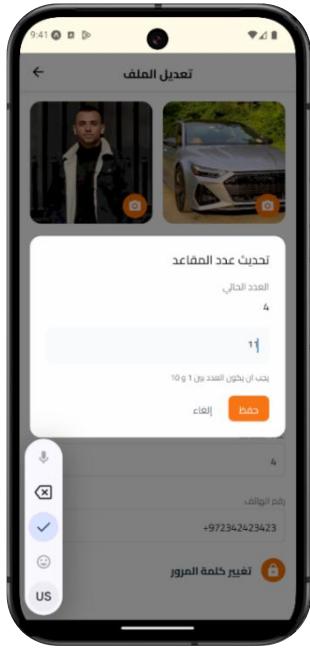


Figure 44 Car Seats Change Model.



Figure 45 Car Seats Must be Between 1 and 10 Error.



Figure 46 Modifying Car Seats to be Compatible with the Condition.



Figure 47 Car Seats Changed Successfully.

## 5.9 Language Button

When this button is pressed, the entire application switches to English, allowing users to view all content and interface elements in their preferred language.

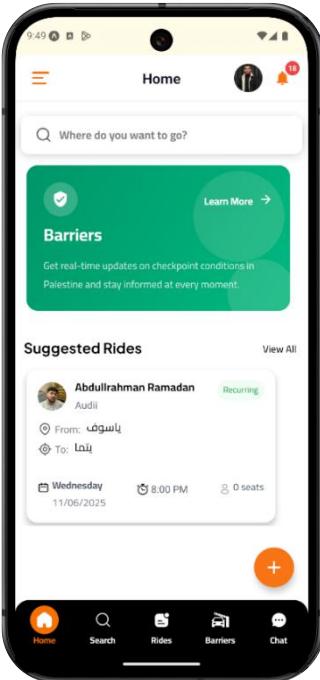


Figure 48 English Home Page.

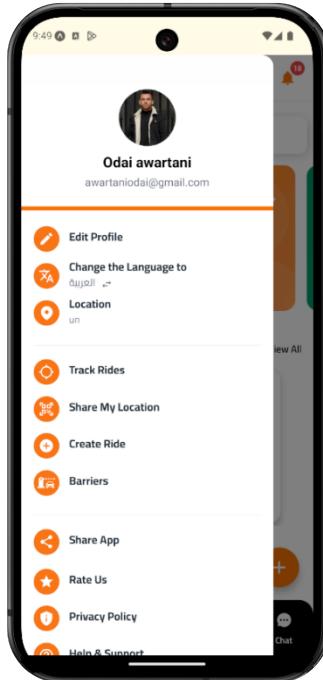


Figure 49 English Side Menu.

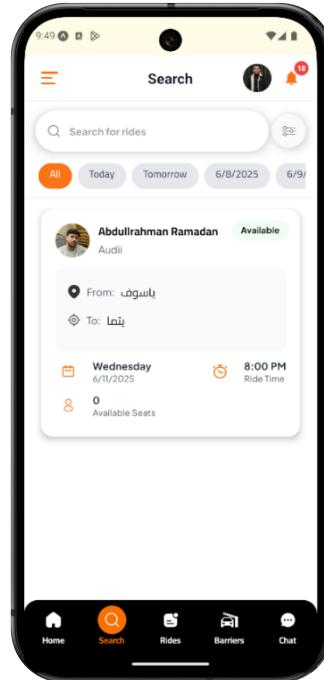


Figure 50 English Search.

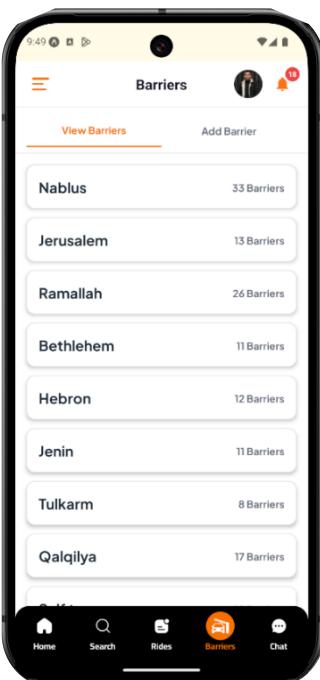


Figure 51 English Barrier.



Figure 52 English Chat.

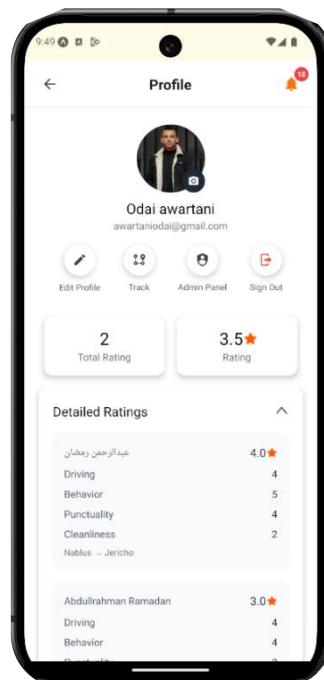


Figure 53 English Profile.

## 5.10 Location Management

Every day when the user opens the application, a location modal will appear, asking: “From which location do you want to travel today?” This modal displays the locations the user has previously saved—such as Home, University, or Work—as these are commonly used starting points.

If the user is new and hasn’t saved any locations yet, they will see a button prompting them to add a new location, which will then lead them to the Location Page to set it up.

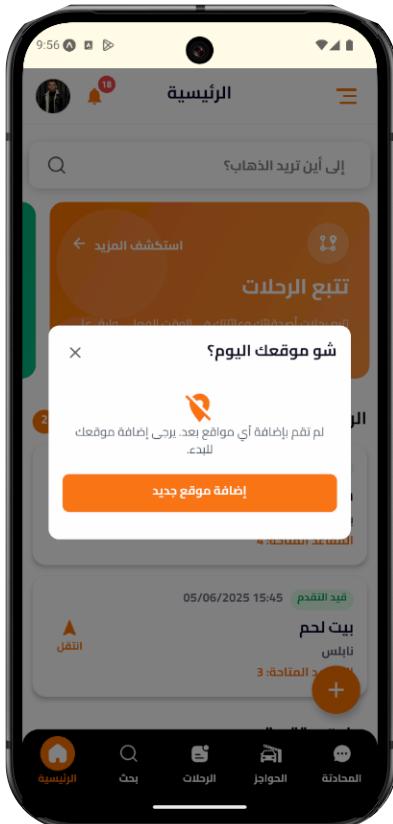


Figure 54 Location Model if User didn't Sign Any Location.



Figure 55 Location Model if the User Already Signed Locations.

Here, I can add a new location, switch between saved locations, or delete any of my existing locations as needed.

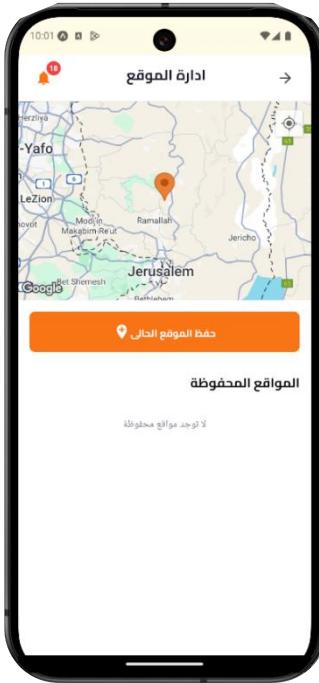


Figure 56 Location Manager.

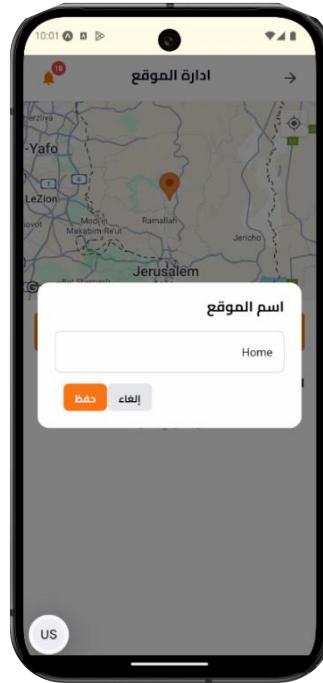


Figure 57 Adding New Location.

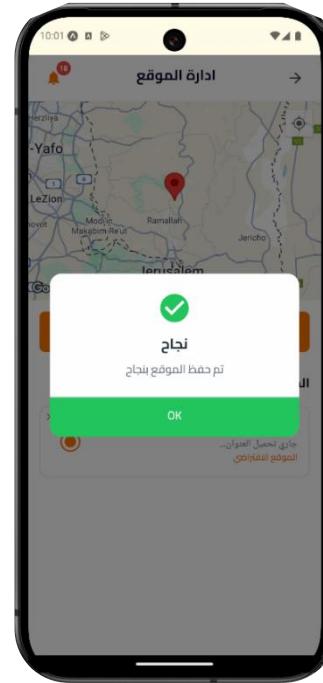


Figure 58 Location Adding Success

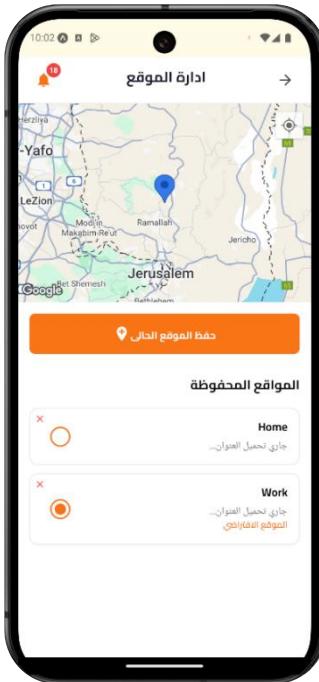


Figure 59 Clicking on x sign to Remove the Home Location.



Figure 60 Confirmation Message to Delete the Location.

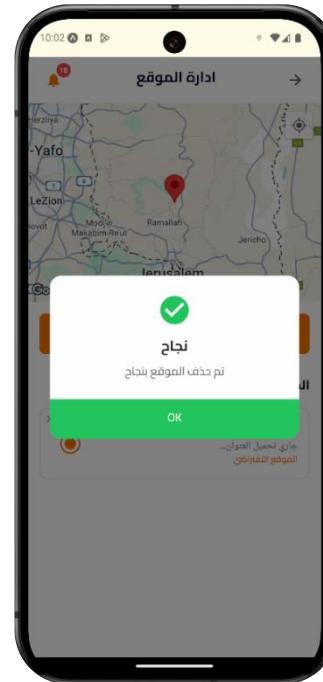


Figure 61 Location Delete Success.

## 5.11 Share Button

This button allows you to easily share the application with others. When you tap it, you can choose from various sharing options like WhatsApp, Messenger, or email to invite friends and family to download and use the app.

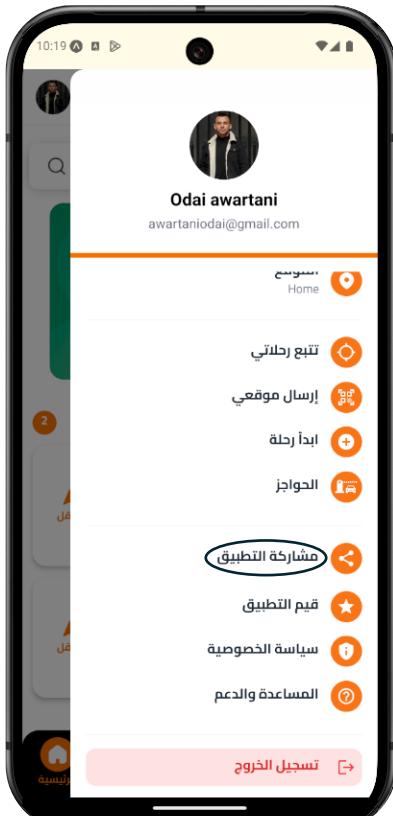


Figure 62 Application Share Button.



Figure 63 Sharing Contacts.

## 5.12 Application Rate

This button allows you to rate the app on the store. By tapping it, you'll be redirected to the app's page on the App Store or Google Play, where you can leave a rating and review to share your experience and help us improve.

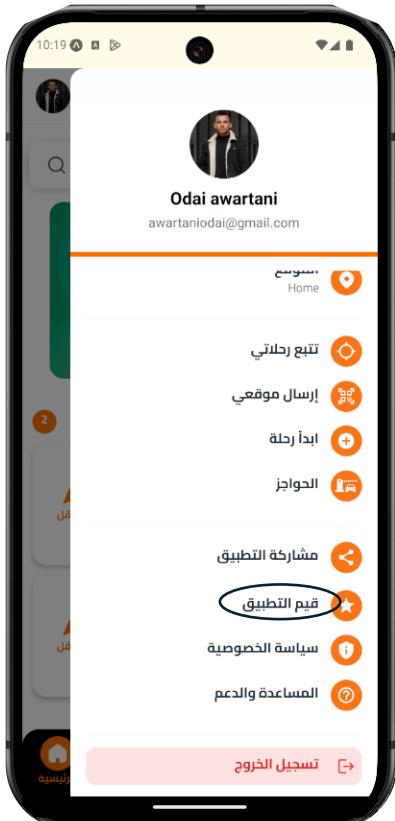


Figure 64 Application Share Button.

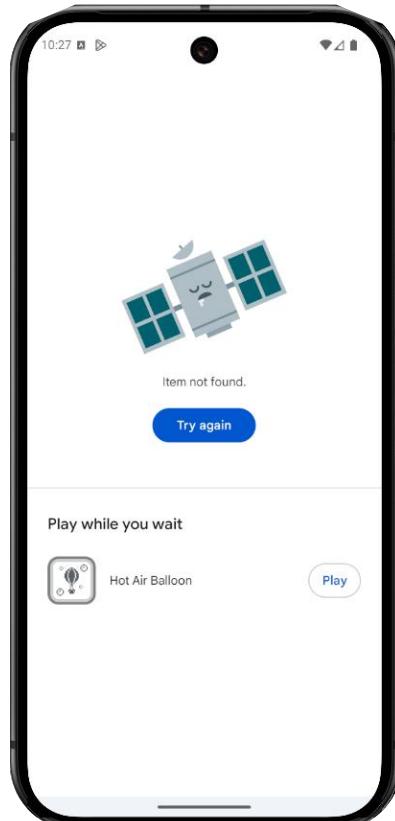


Figure 65 Google Play or App Store.

## 5.13 Privacy Policy

This button opens the app's Privacy Policy, where you can learn how your personal data is collected, used, and protected. It ensures transparency and helps you understand your rights as a user.

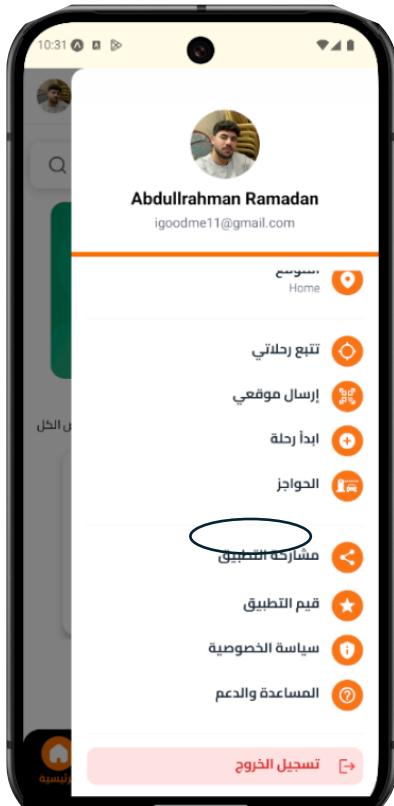


Figure 66 Privacy Policy Button.



Figure 67 Privacy Policy 1.



Figure 68 Privacy Policy 2.

## 5.14 Help & Support

When the user faces a problem, they can open this page, type the issue, and press Send. If the box is empty, an error appears: "Please fill in the box." If filled, a success message appears, and the problem is sent to the admin.

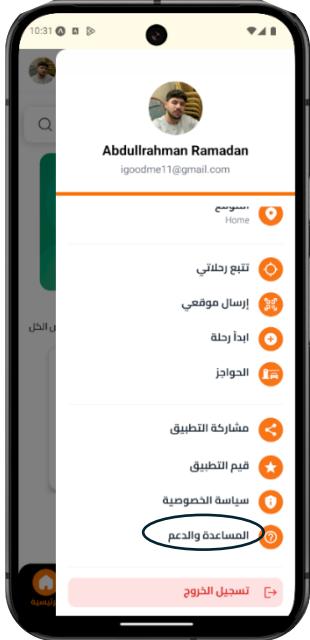


Figure 69 Help & Support Button.

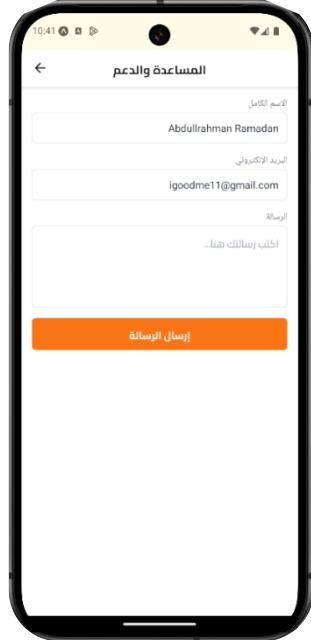


Figure 70 Help & Support Page.

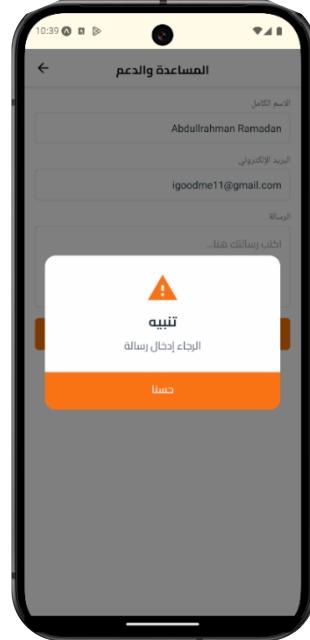


Figure 71 Sending Empty Problem.

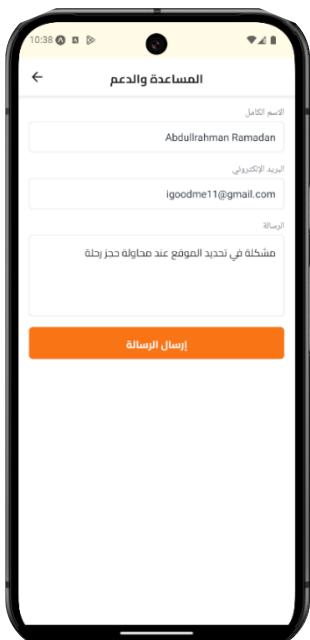


Figure 72 Filling the Problem.



Figure 73 Problem Sent Successfully.



Figure 74 Message Received When the Problem Solved

The message goes to the Support Messages section. The admin clicks "Start Processing", and once the issue is resolved, the user receives: "Problem solved, thank you!"

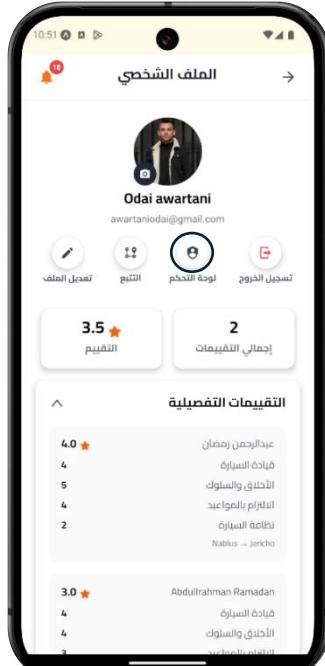


Figure 75 Control Panel Button.

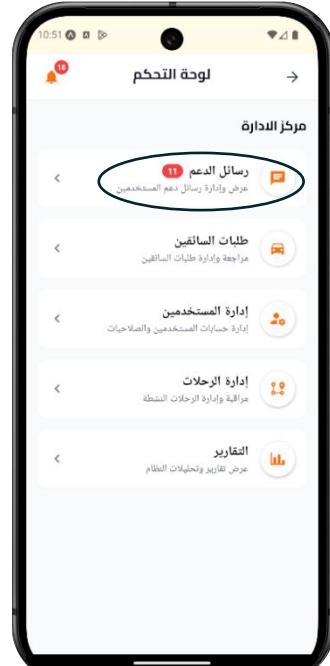


Figure 76 Support Messages.

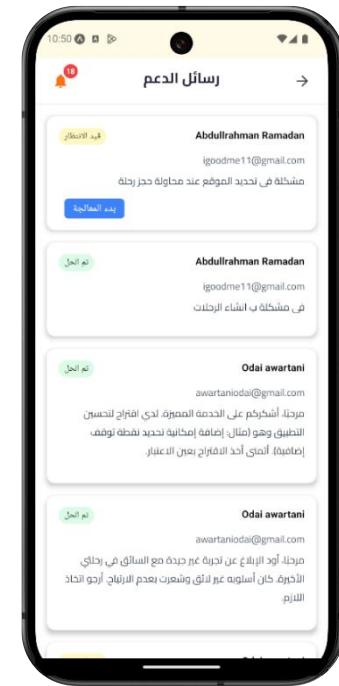


Figure 77 Start Processing button.

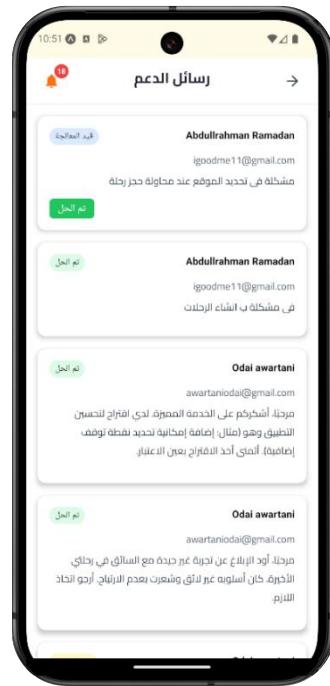


Figure 78 Solved Button.

## 5.15 Home Page

The homepage serves as the main interface of the Wasselny app, providing users with quick access to essential features and trip planning tools. It is designed to be intuitive, informative, and responsive to the user's real-time needs, making the ride-sharing experience smoother and more reliable.

Directly beneath the header, the homepage features a prominent search bar. Users can search for rides by entering destination names or cities, making it easy to find available trips or plan routes. This search functionality is powered by Google, ensuring fast and accurate location results.



Figure 79 Search Bar.



Figure 80 Searching for village.

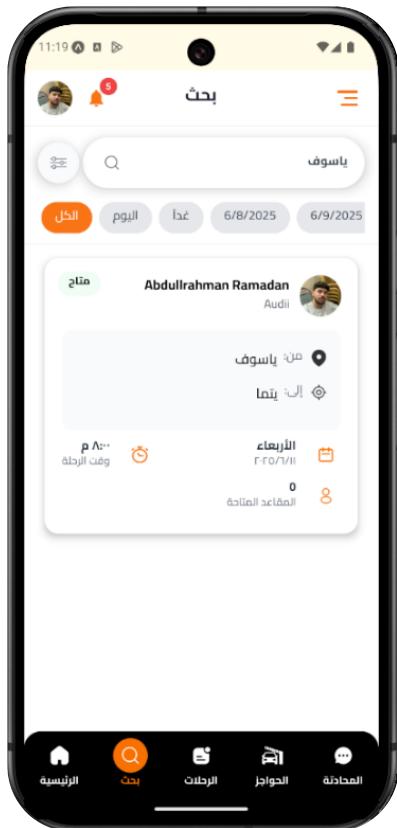


Figure 81 The Results.

## Application Feature

Following the search bar is a carousel featuring three informative slides that highlight key app functionalities. These slides include: "Become a Driver & Make Money," which encourages users to register as drivers; "Track Users (e.g., your son) for Safety," which promotes the real-time tracking feature; and "Barrier Check," which helps users check the status of military checkpoints before starting their trip. Each slide is interactive and links directly to the relevant functionality.

### 5.15.1 Become A Driver

**Note:** Whenever you become a driver, this card will disappear from the entire application.



Figure 82 Homepage card introducing the "Become a Driver" feature.

After the user presses the card, they will be taken to a page to fill in the following information: car type, number of seats, user picture, and car picture. We have added validation to ensure the user completes all fields. If any field is left empty, an error message will appear asking the user to fill in all the fields.

Once the user submits all the information, a message will inform them that the admin will review their application as soon as possible. While waiting for the review, if the user tries to submit another application, an alert will notify them that they already have an application under review by the admin.

## How to be a Driver in wasselni



Figure 83 Driver Application Information.

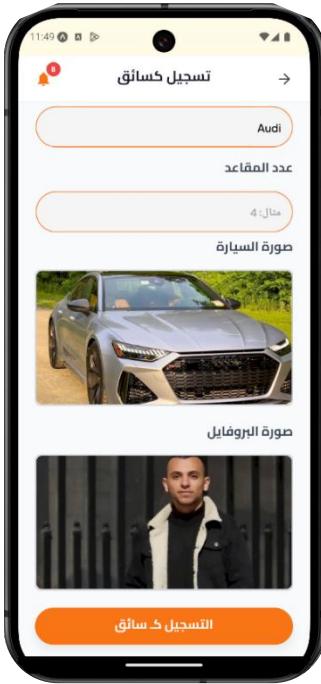


Figure 84 Filling All the Information Except the Car Seats.

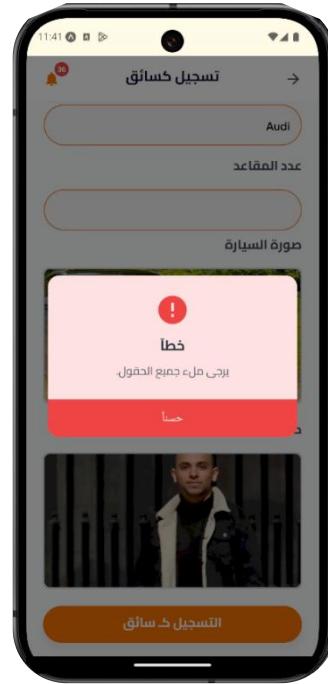


Figure 85 Fill All the Field Message.

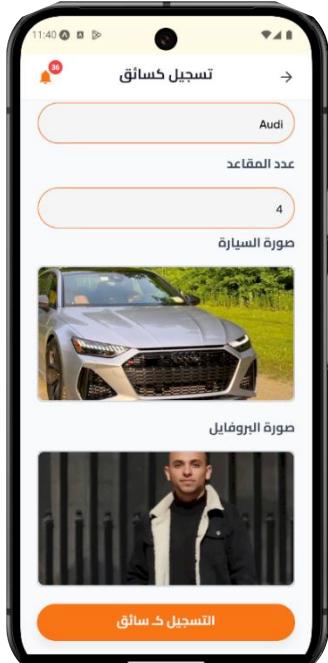


Figure 86 Fill the Missing Box.



Figure 87 Application Delivered Successfully.



Figure 88 What I Will See if I Tried to send Another Application.

## How the Admin Accept the Application



Figure 89 Control Panel Showing New Application.



Figure 90 Checking on the Driver Application.



Figure 91 Accept or Reject the Application.



Figure 92 User Receive the Acceptance on Notification.

## How the Admin Rejects the Application

The admin has the ability to reject a driver's application based on specific criteria. When an application is rejected, the admin provides a clear reason for the rejection, allowing the driver to understand the issue, make the necessary improvements, and submit a new application if desired.



Figure 93 Reject the Driver.



Figure 94 Cause of Rejection.



Figure 95 Rejection Done Successfully.

## Driver Receive the rejection with the cause



Figure 96 Notification Received.



Figure 97 Receive the Rejection.

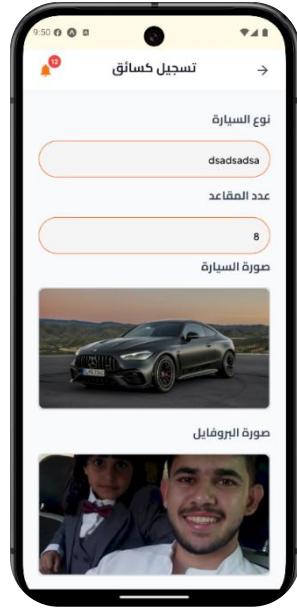


Figure 98 Old Information.



Figure 99 Modified Information.



Figure 100 Resend the Application.

### 5.15.2 Track Page



Figure 101 Homepage card introducing the "Track" feature.

To enable live tracking in Wasselni, tap the "Share Location" card on the homepage. Then, click the share icon and select the person you want to share your location with. They will be able to track your movement in real time, with your location being automatically updated every 30 seconds. This ensures consistent and reliable tracking throughout your journey for enhanced safety

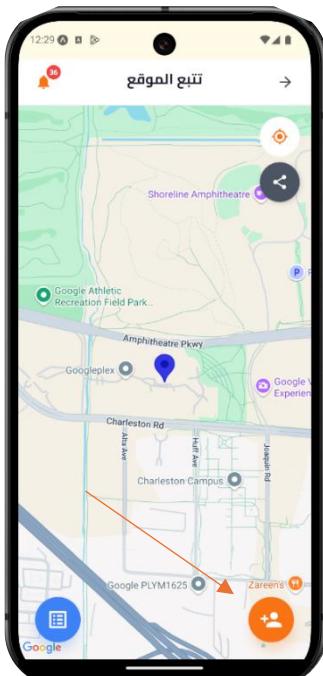


Figure 102 Track Page.

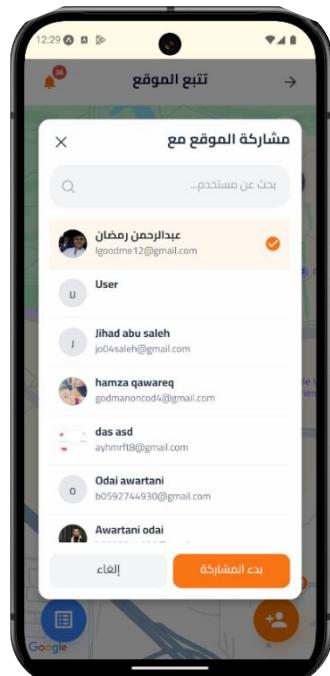


Figure 103 Share My Location model.

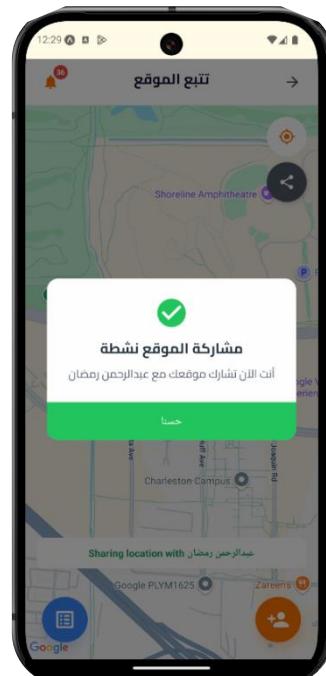


Figure 104 Location shared Successfully.

To view users who are sharing their location with you, go to the "Shared Locations" section in the app. You'll see a list of users currently sharing their live location. Simply click on "View" next to a user's name, and the map will automatically zoom in on their current location, allowing you to track their movement in real time with updates every 30 seconds.

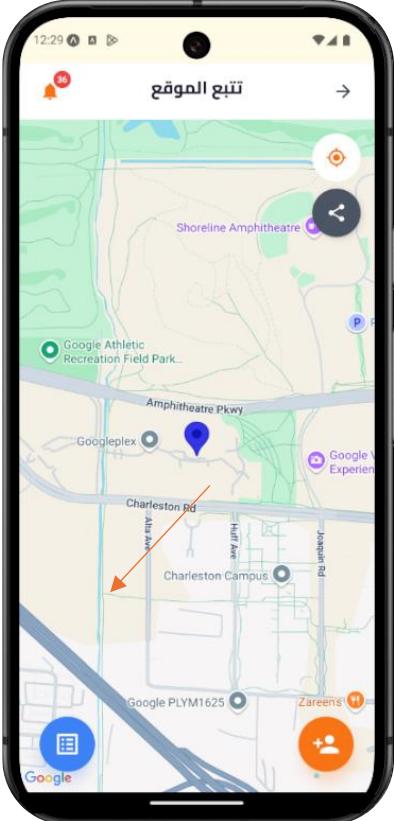


Figure 105 Track Page.

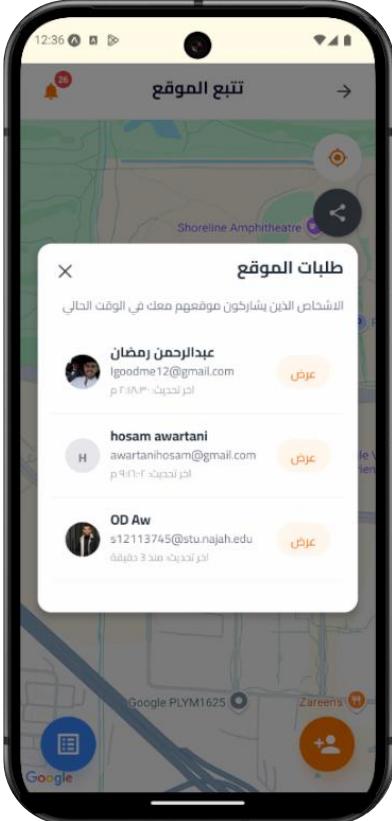


Figure 106 People Sharing their location with me.

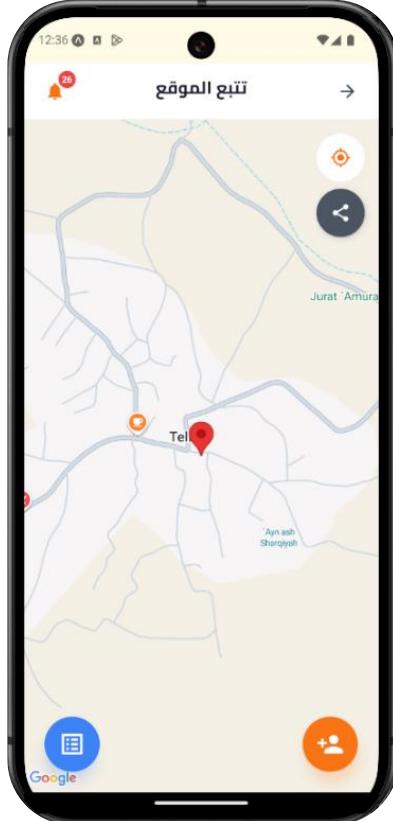


Figure 107 Location of the user

To manage your location sharing in Wasselni, go to the "My Shares" section. There, you can view who you're sharing with, update your location, or stop someone from tracking you at any time.

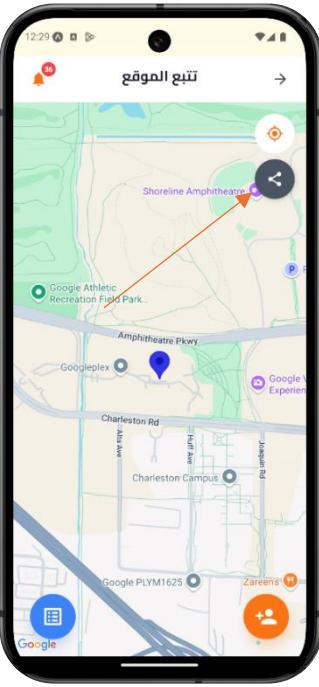


Figure 108 Track Page Click on Shares.

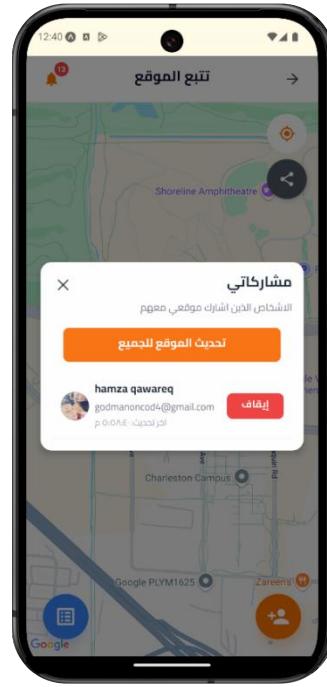


Figure 109 Update my Location.



Figure 110 Update Done Successfully.



Figure 111 Stop Someone from Tracking me.

### **5.15.3 Suggested Rides**

The Suggested Rides section appears on the home screen as a set of interactive cards that automatically rotate every 5 seconds to display different ride options. These rides are prioritized and sorted based on proximity to the user, helping them quickly find the most relevant options nearby. Each card displays key information such as the origin and destination of the ride, allowing users to instantly determine if the ride matches their route. A "View All" button is provided to let users explore the complete list of available rides beyond those currently visible.

Each ride card is designed to highlight the most important ride details in a concise format. When a user taps on a ride card, a bottom sheet opens, providing full information about the ride. This includes the number of available car seats, the driver's starting point, waypoints along the route (to help the user determine if the driver passes near them), and the ride's final destination. The date and time of the ride are also clearly shown. Additional ride preferences are displayed, such as gender preferences (if any) and ride rules, including whether smoking, children, or music are allowed during the ride.

The bottom sheet also includes a passengers table that lists all currently booked passengers, their names, the number of seats they have reserved, and the location where each passenger will get off. A "Book Request" button is available, allowing the user to send a booking request to join the ride. Below this section, there is a timer indicating when the ride can begin. If the ride hasn't started yet, the "Start Ride" button remains gray and inactive. Once the scheduled start time arrives, the button turns blue and becomes clickable. There is also a "Cancel Ride" button in red, which allows the user to cancel their booking if needed.

If the user closes the bottom sheet, a map view is displayed showing the full route of the driver. This map includes the starting point, all waypoints along the route, and the final destination, giving the user a clear visual overview of the trip. This feature ensures that users can make an informed decision before booking a ride by seeing whether the ride path aligns with their own route.

All of these features and components will be clearly illustrated with pictures on the next page for better understanding.



Figure 112 Suggested Rides Cards.



Figure 113 Ride Details.

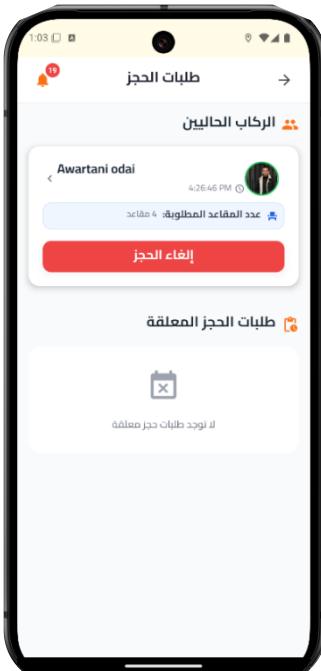


Figure 114 Book Requests



Figure 115 Map Ride Route.

#### 5.15.4 View All Button

The "View All" button allows users to browse all available rides in the system. It includes filtering options to sort rides by **date** (newest to oldest or vice versa) and by **number of available seats** (more or fewer). This feature enhances user experience by making it easier to find suitable rides quickly and efficiently.

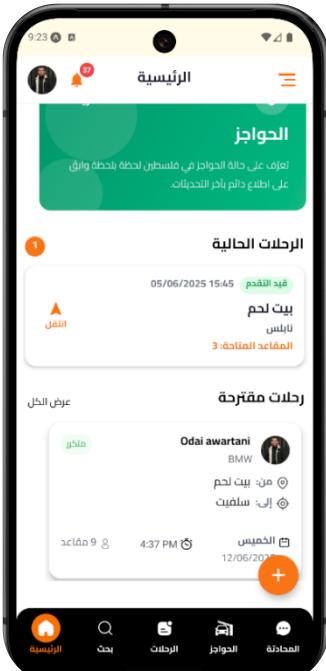


Figure 116 View All Button.



Figure 117 All Rides



Figure 118 All Rides Filter.

### **5.15.5 Create Ride**

After a user becomes a driver, they can create a new ride by tapping the “+” icon located at the bottom of the screen. This action opens a new page where the driver is prompted to fill in the essential ride details. The first step includes entering the starting location (“From”) and the destination (“To”), with the option to specify exact street names for more accurate routing. These street fields are optional but can improve the quality of the route for passengers. In the middle of this page, there is also an “Add Waypoint” button, allowing the driver to insert optional stops along the way. Including waypoints increases the chances of matching with more nearby passengers.

Once this initial information is filled in, the driver can tap “Next” to proceed to the second step. Here, additional ride details must be provided, including the number of available car seats, the date, time, and day of the ride, as well as any gender preferences the driver wishes to set. At the bottom of this page, the driver can also specify whether the ride will be repeated regularly in the upcoming weeks by marking it as a recurring ride.

After pressing “Next” again, the driver is taken to the ride rules section. In this step, the driver can choose which rules will apply to the ride by checking the available options: No Smoking, No Children, and No Music. These rules help set expectations and provide a clear environment for passengers.

Finally, after all required information is reviewed and completed, the driver can tap the “Create Ride” button to publish the ride. A confirmation box will appear to verify that the ride has been created successfully, providing the driver with feedback and assurance.

All the steps mentioned above will be visually demonstrated with screenshots on the following pages to ensure clarity and ease of understanding.

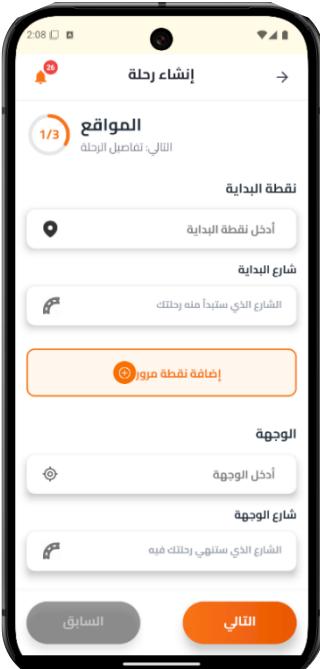


Figure 119 Adding One Waypoint.

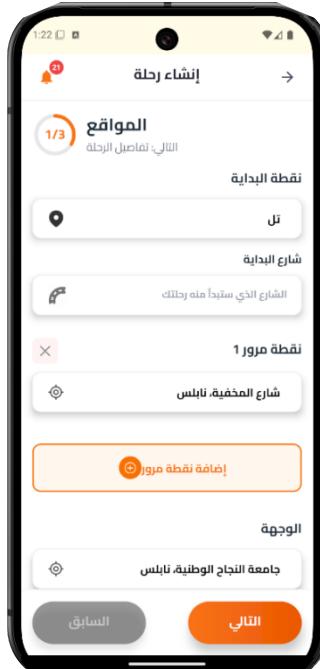


Figure 120 Filling Ride Locations.

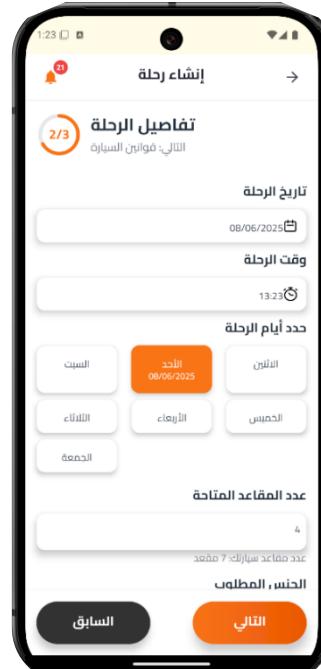


Figure 121 Filling Date Time & Car Seats.



Figure 122 Other Information.

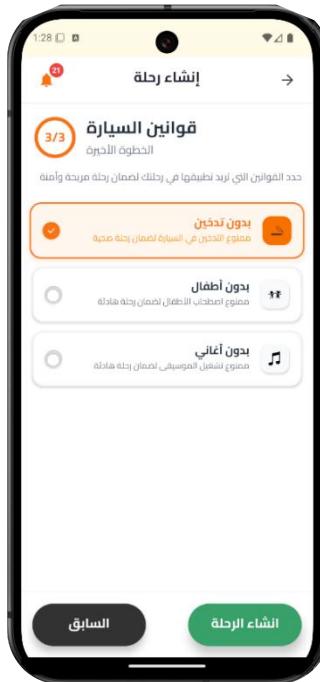


Figure 123 Ride Rules.

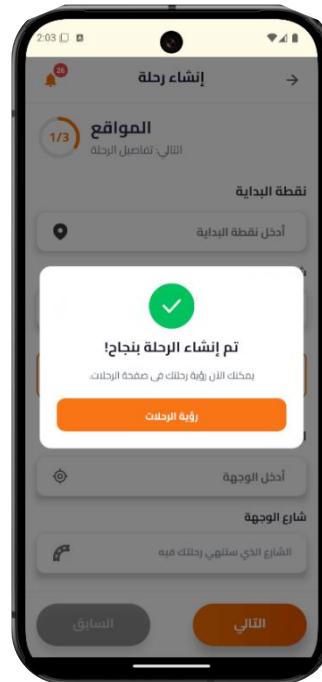


Figure 124 Ride Created Successfully.

If the driver misses the start or end point, date or time, or car seats, an error will appear. Also, selecting a past date/time shows: "Ride time must be in the future."



Figure 125 Fail All Locations Error.



Figure 126 Pick Time & Date for Ride Error.



Figure 127 Fill Car Seats Number Error.



Figure 128 Time & Date Should be in Future.

### 5.15.6 Book Ride

When a user wants to travel from one place to another, they can open the application and either browse the Suggested Rides section or use the search bar to look for a specific destination or waypoint. Matching rides will then be displayed on the screen, as shown in the examples below. After selecting a ride, the user can view the ride details, starting with the route on the map to decide where they can meet the driver. To book the ride, the user presses the Book button, which opens a modal showing the available waypoints. The user can then choose a waypoint to meet the driver, select the number of car seats they need, and confirm the booking. After that, the user will wait for the driver to accept or reject the booking request.

These steps will be illustrated with screenshots on the following pages.



Figure 129 Search for the Area.



Figure 130 Search Result.

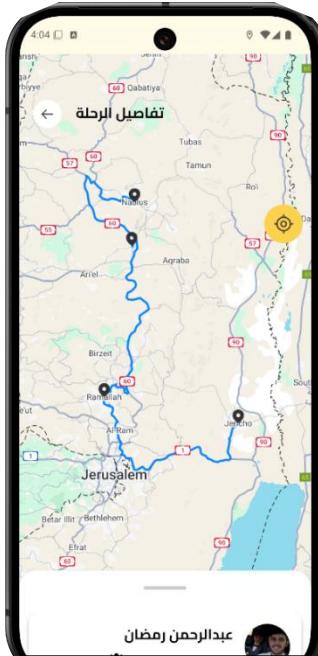


Figure 131 Ride Route Check.



Figure 132 Ride Details.



Figure 133 Choosing Car Seats.



Figure 134 Picking Stop Point.



Figure 135 Ride Details After Booking.

### 5.15.7 Driver Accept or Reject the Passenger

When a user sends a booking request, the driver gets a notification like: "Odai wants to book a ride with you from [Area A] to [Area B]." By tapping it, the driver opens the ride details, views booking requests, and can Accept or Reject each user.



Figure 136 Book Notification.



Figure 137 Book requites.



Figure 138 Accept or Reject the Passenger.



Figure 139 Confirm Passenger Alert.

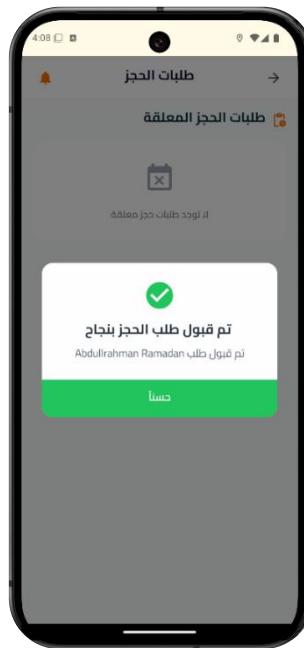


Figure 140 Passenger Accepted.

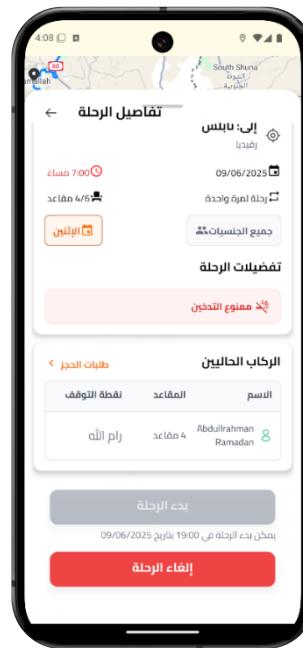


Figure 141 Current Passengers.

## Booking in a Fully Booked Ride

If a user tries to book a seat in a ride that is already fully booked, an alert will appear saying the ride is full. However, the booking request will still be saved, and if another passenger cancels, the driver can approve the pending request.



Figure 142 Book full Booked Ride.



Figure 143 Ride full Booked Alert.



Figure 144 Fill Car Seats.



Figure 145 Pick Stop Point.

## Accepting a Passenger in a Fully Booked Ride

If the driver tries to accept a passenger while the ride is already fully booked, an error message will appear saying: "Ride is fully booked. Not enough available seats."

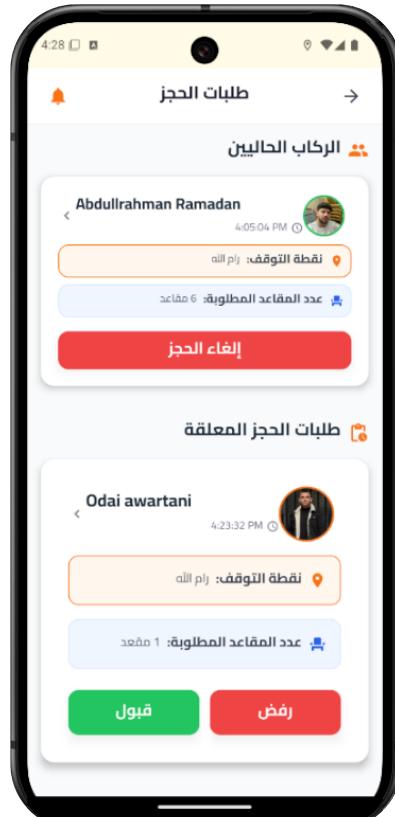
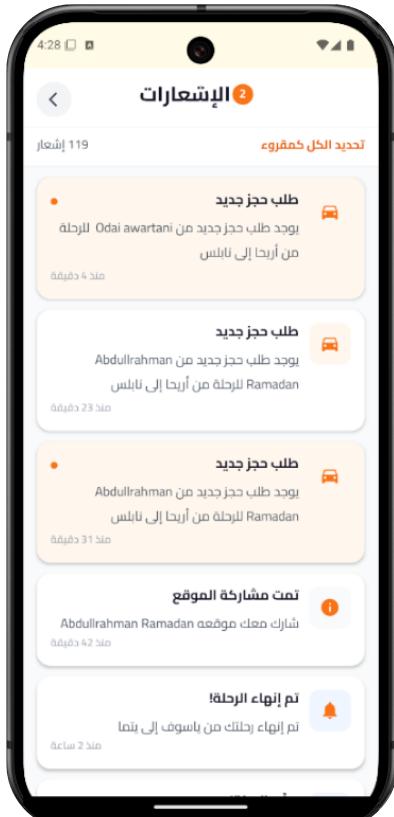


Figure 146 Passenger Book Request.

Figure 147 Try to Accept the Passenger.

Figure 148 Not Enough Car Seats Error.

### 5.15.8 Driver Starts a Ride

When the scheduled ride time arrives, the driver must open the ride and press "Start Ride." The ride will then be pinned on the home page, and a notification will be sent to all passengers informing them that the driver is on the way to their location.



Figure 149 Start the Ride.



Figure 150 Ride Started Successfully.



Figure 151 Home Page When I start a Ride.

### 5.15.9 Driver Communication with Passenger

The driver can go to the booking requests and tap on a passenger's image to open their profile. There, the driver can view key details such as the passenger's name and image, selected location, phone number, email, and also start a chat within the app if communication is needed.



Figure 152 Click on the picture.

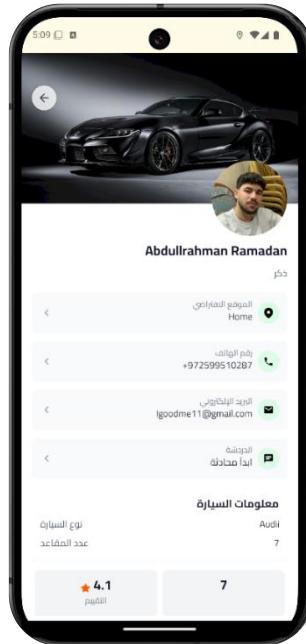


Figure 153 Click on Location.

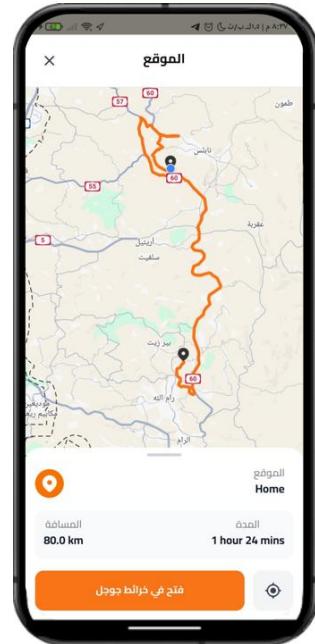


Figure 154 Passenger Location.

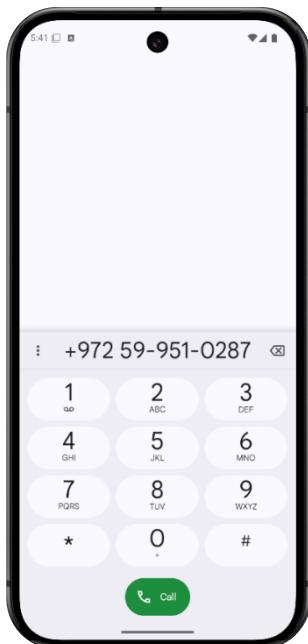


Figure 155 Clicked on Phone Number.

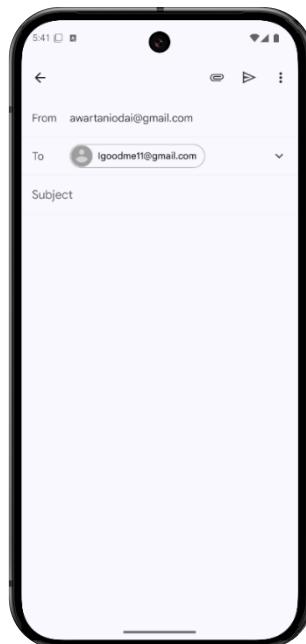


Figure 156 Clicked on Email.



Figure 157 Clacked on Chat.

### 5.15.10 Passenger Check-In for the Ride

Once the passenger gets into the car, they need to check in through the app. This action will share their live location with the people they've chosen to share it with, ensuring added safety and transparency.



Figure 158 Press on Check in Button.



Figure 159 Do you Want to Share Your Location with the People Sharing their Location With you.



Figure 160 Check in Done Successfully.



Figure 161 Home Page After Checking in.

### 5.15.11 Passenger Check-Out and Rate the Driver

After the ride ends, the passenger must check out and rate the driver. Once submitted, the driver receives a notification with the rating provided.



Figure 162 Check-out.

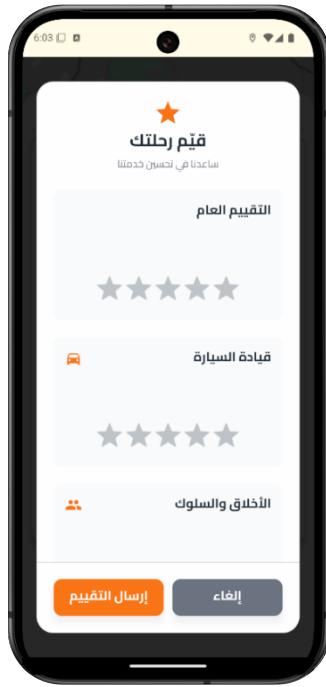


Figure 163 Rate Model 1.

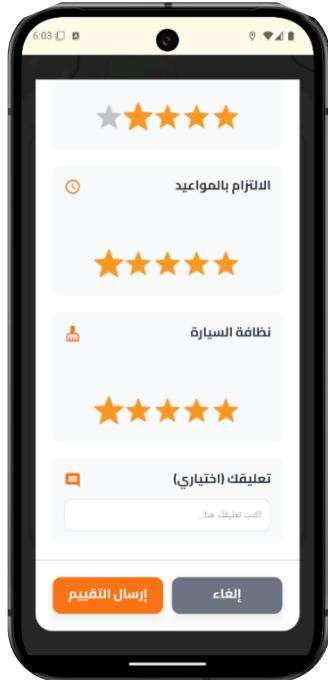


Figure 164 Rate Model 2.

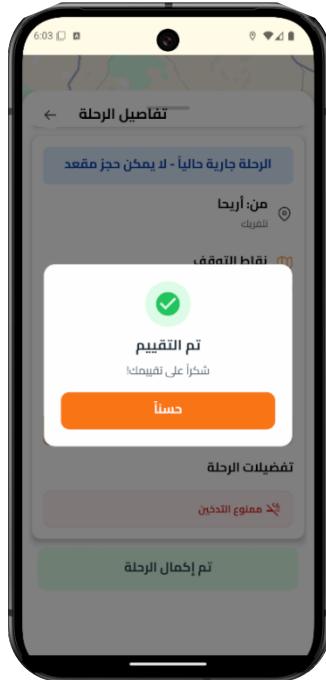


Figure 165 Thanks for Rating.

### 5.15.12 Driver Receives the Rating and Ends the Ride

After the passenger submits their rating, the driver can view it as feedback to help improve future rides. To finish the process, the driver must press the “End Ride” button. If the ride was marked as frequent, a modal will appear asking whether to create the same ride for the next week. If confirmed, the ride will be automatically created.



Figure 166 Rating Notification.



Figure 167 End Ride Button.



Figure 168 Ride Ended Alert.

## 5.16 Barriers Page

This feature is considered fundamental in the app, as it helps you stay informed about the status of all barriers in real time. You can check whether a barrier is open, closed, or partially open, along with any reports of police presence. Drivers can also contribute by updating the status of a barrier they've passed through simply by going to the "Barriers" tab, selecting the relevant barrier, and submitting the current state along with any additional comments or feedback. Additionally, users have the option to upload a photo of the barrier to give others a clearer view of the situation.

**Note:** This feature adding (barriers) can only be used by drivers. However, the barriers added will be visible to everyone.



Figure 169 Homepage card introducing the “Barriers” feature.

### 5.16.1 Viewing Barrier Status

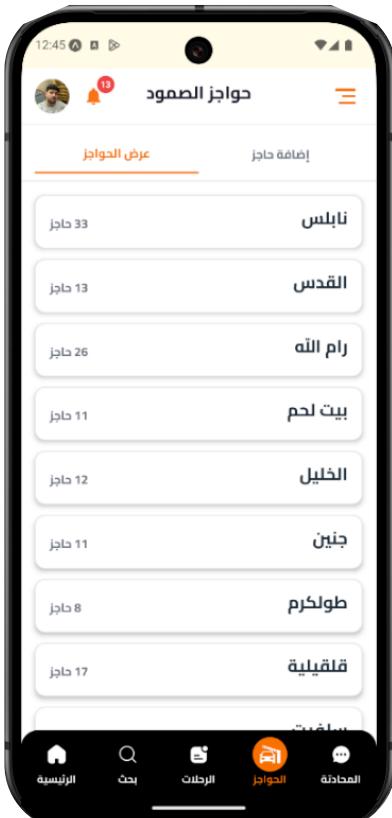


Figure 170 Barriers Page.

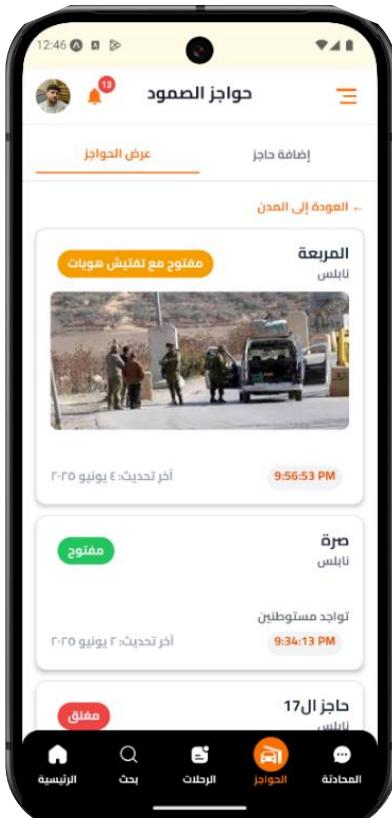


Figure 171 Nablus City Barriers.



Figure 172 Barrier Details for mrabaa

It displays the number of barriers in each city along with their current status. You can also see the exact time the status was last updated for example, if someone reported it as open at 7:15 PM, you'll know when that information was provided, helping you make more informed decisions.

### 5.16.2 Updating Barrier Status:

As shown in the screens, you can easily update a barrier's status or add a new one by selecting the city, choosing the barrier name, and setting its current status. You can also report details like traffic, police presence, inspections, or incidents.



Figure 173 Add Barrier Situation.

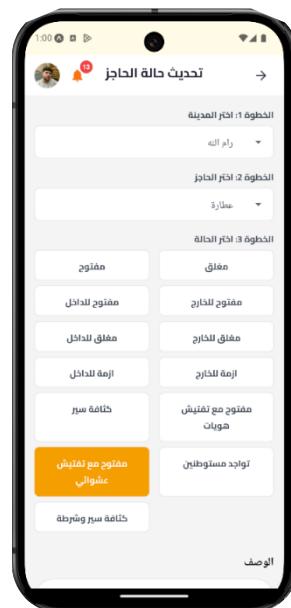


Figure 174 Filling Barrier Information



Figure 175 Choosing Picture and Updating.

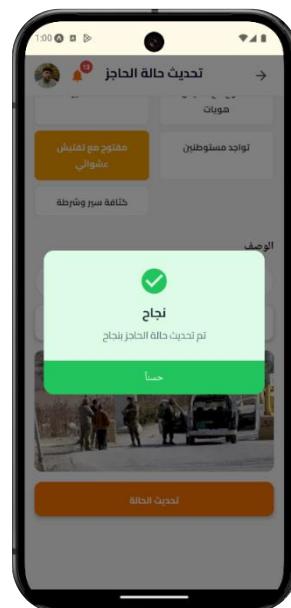


Figure 176 Barrier Updating Successfully.

## 5.17 Search Page

On the Search page, users can perform several actions. They can search for rides, view recent searches, and filter results by time such as Today, Tomorrow, All, or the next 5 days. Additional filters are available to narrow down rides based on distance (km), ride status, date, and sex preferences. Once filters are selected, users can apply them using the Apply button at the bottom.



Figure 177 Search Page.



Figure 178 Search Filter.

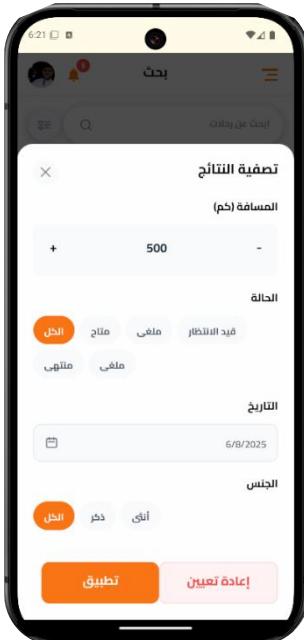


Figure 179 Choosing Criteria.



Figure 180 Search Results.

## 5.18 Rides Page

The Rides page displays all rides related to the user and is divided into two sections: Upcoming Rides and Recent Rides. In the Upcoming section, users can filter rides by status or choose to view only their own rides. The Recent section shows both the rides the user has driven and the rides they have joined as a passenger.

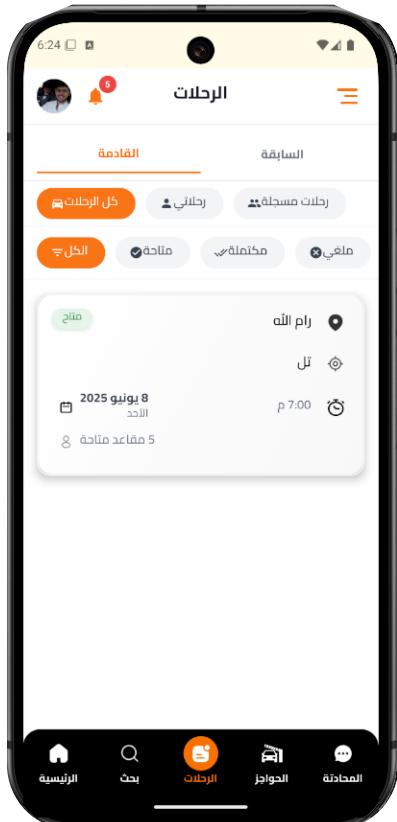


Figure 181 Upcoming Rides.



Figure 182 Recent Rides

## 5.19 Chat Page

The Chat page provides a fully functional messaging feature. Passengers can search for drivers to ensure smooth communication. It includes typical chat features like “Active Now” status, and conversations are sorted by most recent, so the last person you messaged appears at the top of the list.

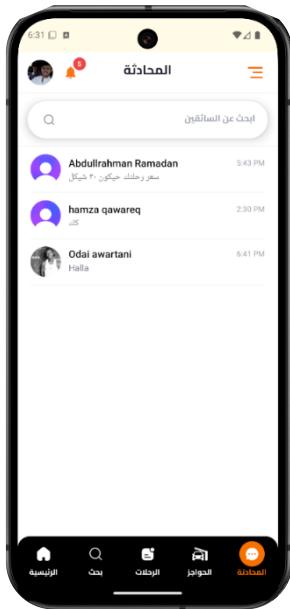


Figure 183 Chat Page.



Figure 184 Search for Driver.

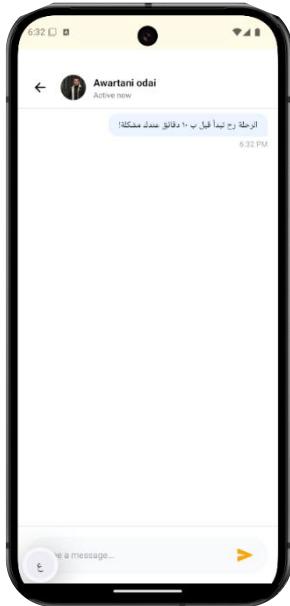


Figure 185 Text the Driver.



Figure 186 The Driver I Texted is the First on Screen.

## 5.20 Driver Profile from Passenger Perspective

When a passenger taps on the driver's name or photo, they are taken to the driver's profile. Here, they can view the driver's rating to assess their reliability and see a list of the driver's recent rides for better insight into their activity and experience.

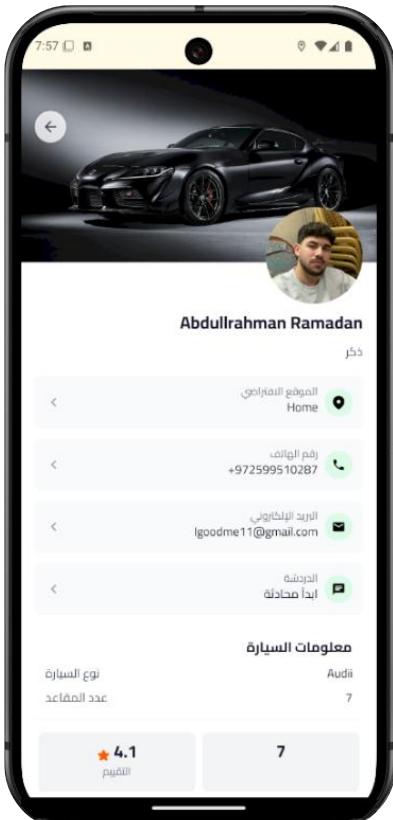


Figure 187 Driver Profile from Passenger Perspective.

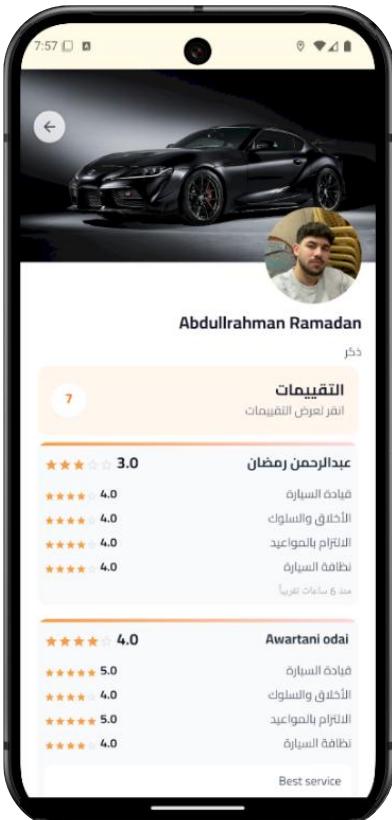


Figure 188 Driver Rating.

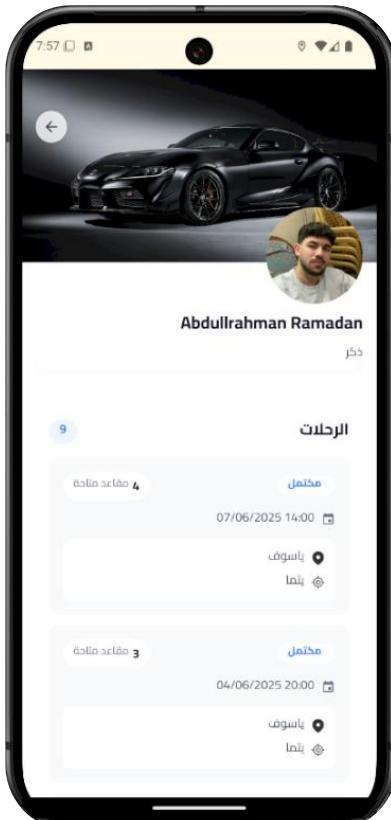


Figure 189 Recent Rides.

## 5.21 Passenger Profile from Driver Perspective

By navigating to the booking requests, the driver can see the list of users who have requested to join the ride. By tapping on a passenger's name or image, the driver is taken to the passenger's profile, where they can view personal details and initiate communication if needed.



Figure 190 Click on Book Requests.



Figure 191 Click on Passenger Picture.

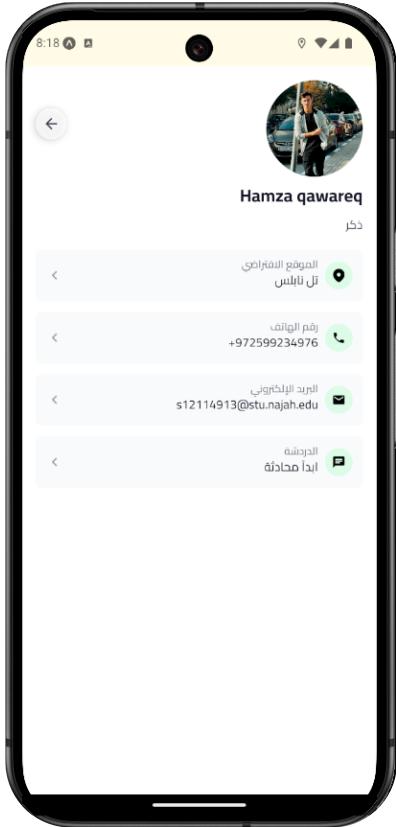


Figure 192 Passenger Profile.

## 6. Admin Dashboard

The Admin Dashboard includes several features designed to improve the application and enhance user experience. It provides access to user support messages, allowing the admin to respond to issues. It also includes driver applications, enabling the admin to verify submitted information before approving driver accounts. Additionally, the dashboard offers user management tools, including the ability to suspend drivers if needed, as well as ride management features. Lastly, it displays ride statistics and other relevant data to monitor app activity.

### 6.1 Driver Applications and Support Messages

These sections have already been explained earlier in this document, where I detailed how driver applications are reviewed for authenticity and how support messages from users are handled.



Figure 193 Click on Control Panel.

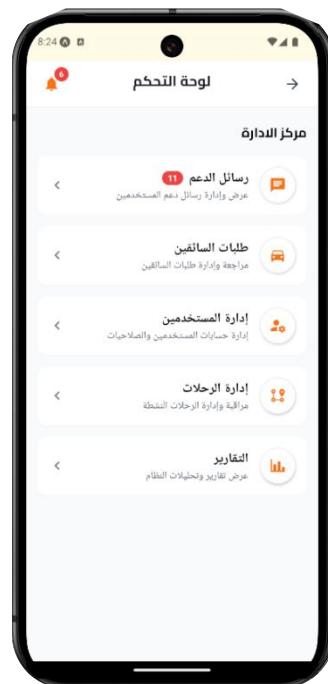


Figure 194 Dashboard Features.

## 6.2 User Management

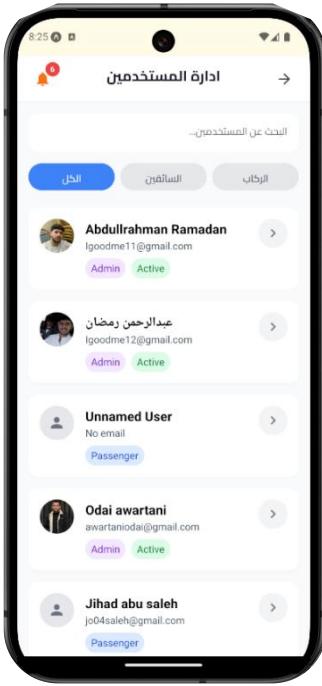


Figure 195 User Mngmt Page.

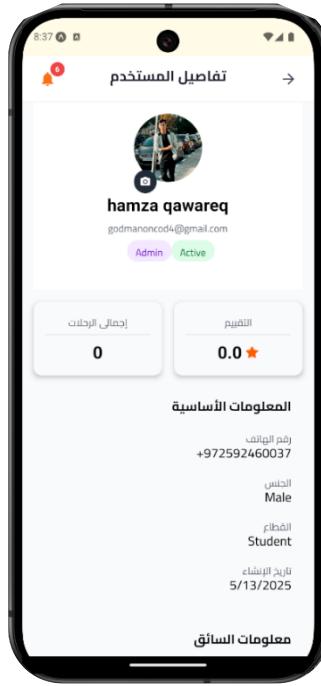


Figure 196 User Details.



Figure 197 Click on Stop Driver.

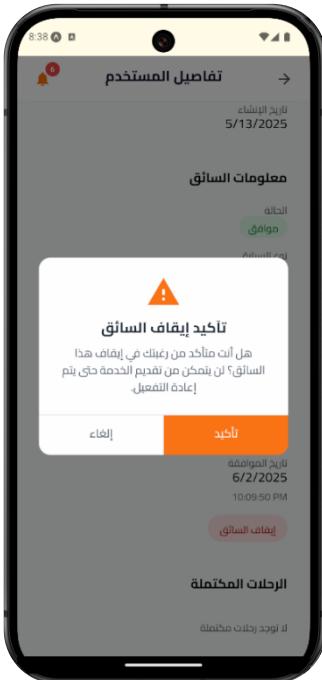


Figure 198 Alert to Confirm Stop Driver.



Figure 199 Driver Stopped Successfully.

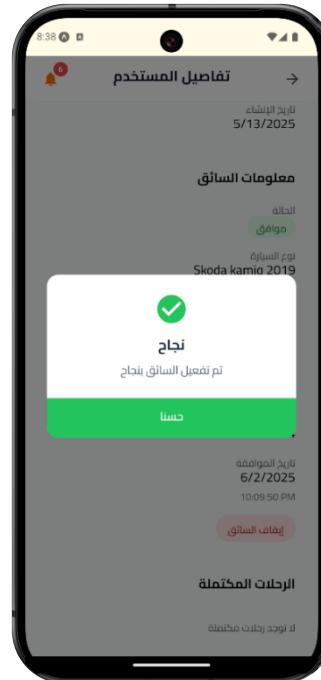


Figure 200 Reactive Driver.

### 6.3 Rides Management



Figure 201 Rides Mngmt Page.



Figure 202 Ride Details 1.



Figure 203 Ride Details 2.

## 6.4 Reports

The Reports section provides key statistics such as the total number of drivers, total rides, overall users, and ride cancellation counts. These insights help monitor the Application performance and user activity.



Figure 204 Reports.

## **7. Technology used**

To ensure optimal performance, scalability, and a seamless user experience, the following technologies and tools were utilized in the development of the mobile application:

### **7.1 Frontend (User Interface):**

- **React Native:** Chosen for its ability to build cross-platform applications using a single codebase, enabling efficient development for both iOS and Android.
- **Expo Go:** Used for streamlined testing and development without needing native builds during early stages.
- **TypeScript:** Provides static typing to catch errors early and improve code quality across both frontend and backend.
- **NativeWind:** Utilized for styling UI components using TailwindCSS principles in React Native.
- **Figma:** Employed for designing intuitive and consistent user interfaces that enhance usability.

### **7.2 Backend (Server-Side):**

- **Node.js:** Used for creating scalable and high-performance APIs with its event-driven, non-blocking I/O model.
- **Firebase:** Integrated to offer essential backend services such as real-time database and cloud storage.

### **7.3 Database:**

- **Firestore (Firebase):** Chosen for its real-time synchronization, scalability, and offline persistence, ensuring smooth user experience even without a stable internet connection.

## **7.4 APIs and Integrations:**

- **Google Maps API:** Enables geolocation, mapping, and route planning features.
- **Cloudinary:** Used for uploading, optimizing, and delivering media content such as images in a scalable and secure way.
- **Expo Notifications:** Integrated to provide real-time notifications to users for ride statuses and updates.

## **7.5 Additional Tools and Services:**

- **Clerk:** Handles all aspects of user authentication, including secure login, session management, and access control, with features like passwordless login and multi-factor authentication.
- **Firebase Storage:** Stores and serves user-generated media securely.
- **AsyncStorage:** Enables local data persistence to cache important data and improve app responsiveness during offline scenarios.
- **ESLint:** Ensures consistent code quality and helps catch errors during development.

## **8. Frequently Asked Questions (FAQ):**

### **1. How can I make a trip?**

To enhance user experience and ensure seamless navigation, we've integrated an intuitive in-app guidance system. This feature assists users in understanding the necessary steps to book a ride, making the process straightforward and user-friendly.

Recognizing the diverse preferences of our user base, we're also incorporating a concise tutorial video. This visual aid will provide a step-by-step walkthrough, catering to visual learners and ensuring that every user, regardless of technical proficiency, can utilize the app effectively.

By combining interactive guidance with visual tutorials, we aim to minimize user confusion, promote feature adoption, and elevate overall satisfaction.

## **2. What should I do if my driver doesn't arrive?**

If your driver hasn't arrived at the designated pickup location:

- Contact the Driver: Use the in-app messaging or call feature to reach out to the driver for an update on their location and estimated arrival time.
- Cancel and Rebook: If the driver is unresponsive or significantly delayed, you may cancel the ride and request a new one. Be aware of the app's cancellation policy to avoid potential fees.
- If the driver was late or failed to arrive, you can provide a lower rating in the 'Punctuality' section to reflect your experience."

## **3. How can I pay?**

Currently, payments are arranged directly between the passenger and the driver. After booking a ride, passengers and drivers coordinate through in-app communication to agree on the payment in cash.

## **4. Does the app support multiple languages?**

Currently, our application supports two languages: English and Arabic. You can switch between these languages by using sideMenu in the header and selecting your preferred language.

## **9. Future Plans:**

Looking ahead, several future plans are planned to continuously improve the functionality and user satisfaction. These include:

- Admin-Set Trip Pricing: Administrators will have the ability to define and standardize trip prices across different routes. This feature aims to ensure fairness, prevent disputes between drivers and passengers, and enhance overall transparency.
- Enhanced Navigation Features: Integration of more advanced route optimization algorithms to provide users with the most efficient travel routes.
- Payment Integration: Adding a secure payment gateway to allow in-app payments for rides, making the transaction process smoother and more secure.
- User Loyalty System: Introducing a rewards or loyalty program to incentivize frequent users and encourage more sustainable ride-sharing behavior.
- Accessibility Enhancements: Improving the app's accessibility for users with disabilities, including voice commands and screen reader compatibility.
- Providing all languages: Currently, our application supports two languages: English and Arabic. We recognize the importance of catering to a diverse user base and are actively working on integrating additional languages to enhance accessibility and user experience. While these features are not available at the moment, we are committed to expanding our language offerings in future updates.
- Multiple methods for payment: At present, our application exclusively supports cash payments. This means that users are required to pay the driver in cash at the end of each trip.

We recognize the importance of offering diverse and convenient payment options to cater to our users' preferences. Therefore, we are actively working on integrating additional payment methods, such as credit/debit cards and digital wallets, to enhance your experience. These features will be introduced in future updates.

- Collecting points: The "Collecting Points" feature is designed to incentivize drivers by allowing them to earn points based on their performance. These points will be accumulated through positive actions such as completing rides, receiving high ratings from passengers, and providing excellent service. The more points a driver collects, the better their reputation and standing within the app.

**Postconditions:**

- Users feel heard and supported.
  - Feedback loop helps improve app performance
  - The points system will serve two main purposes:
    1. Driver Feedback: Points will act as a form of feedback on the driver's performance. A higher point count will indicate that the driver consistently offers great service and is trusted by passengers. This will help passengers make informed decisions when choosing a driver for their ride.
    2. Rewards and Benefits: Accumulating a certain number of points will unlock various benefits for the driver, such as access to exclusive features, priority ride requests, or even discounts on service fees. This creates an incentive for drivers to maintain high standards of service.
- By implementing the "Collecting Points" system, we aim to create a motivated and high-quality driver community while ensuring that passengers feel confident and secure when selecting a driver.

## **10. Conclusion**

In conclusion, this mobile application was developed to support individuals across all demographics within the borders of Palestine, regardless of their background or status. While the initial focus included addressing the challenges faced by students, the app is designed to benefit everyone experiencing transportation delays and restrictions caused by Israeli occupation checkpoints. and restrictions due to Israeli occupation checkpoints. These barriers often lead to long, uncertain commutes. Our goal is to provide a reliable, efficient, and community-driven solution that makes commuting easier and more accessible for everyone.

The app enhances the ride-sharing experience through a user-friendly interface that connects drivers and passengers seamlessly. With features such as ride creation and joining, user ratings, real-time chat, GPS navigation, and customizable profiles, we aim to improve both convenience and trust between users.

Looking ahead, we plan to introduce future updates that include a secure payment system, a loyalty points program, and expanded ride-sharing options to further elevate the user experience and encourage sustainable transportation habits.

We invite users to explore the app, share their feedback, and become part of a growing community working toward smarter and more inclusive mobility solutions.

Done