

# User Interface (UI) User Experience (UX) Analysis



## TRANSPORT SERVICE MATCH PLATFORM

Version	Description of Change	Author	Date
1.0	First Draft	Ahmed Berdai / Aadel Bouzambou / Maria Deffense / Moustapha El Zeir / Ophelia Karapetyan /	28/02/2025

## **Table of Contents**

- 1. Introduction**
- 2. User's Personas**
  - 2.1 Logistic Requester (LR) - Emma
  - 2.2 Logistic Provider (LP) - Mark
  - 2.3 Logistic Requester (LRi) - Ivo
  - 2.4 Logistic Provider (LP) - Aga
  - 2.5 Support – Adam
- 3. User Case Journey**
  - 3.1 Logistic Requester (LR) - Emma Creates a Transport Request
  - 3.2 Logistic Provider (LP) - Mark Accepts and Fulfills a Transport Request
  - 3.3 Logistic Requester (LR) - Emma Tracks Delivery Status
  - 3.4 Logistic Provider (LRi) - Ivo Abandons a Transport Request
  - 3.5 Support – Adam Resolves a User's Issue
- 4. Layouts Example**
  - 4.1 Sign In / Sign Up Interfaces
  - 4.2 Logistic Requester (LR) Interfaces
  - 4.3 Logistic Provider (LP) Interfaces
- 5. Other Interface Elements**


# 1. Introduction

This document provides essential guidelines for the UX/ UI design analysis conducted for the Transport Service Match System. Introducing user personas, visual elements such as screen prototypes for different roles: Logistic Requesters (LR) and Logistic Providers (LP), along with common interface elements and support features. It aims to convey a clear idea of the application design and the user interactions.


## 2. User's Personas

This section outlines the key user types for the Transport Service Match System. Each persona represents a distinct user role, their goals, pain points, and behaviors.


### 2.1 Logistic Requester (LR) - Emma

<b>EMMA</b>	<b>Role:</b>	◆ Creates delivery requests.
	<b>Goals:</b>	◆ Easily schedule and track deliveries. ◆ Ensure accurate and timely delivery of goods.
	<b>Pain Points:</b>	◆ Difficulty finding reliable logistic providers. ◆ Confusion in inputting correct delivery details.
	<b>Behavior:</b>	◆ Regularly uses the app to request deliveries. ◆ Prefers clear instructions and confirmation of delivery status.


### 2.2 Logistic Provider (LP) - Mark

<b>MARK</b>	<b>Role:</b>	◆ Fulfills delivery requests.
	<b>Goals:</b>	◆ Find and accept delivery requests efficiently. ◆ Ensure timely and accurate delivery of goods.
	<b>Pain Points:</b>	◆ Difficulty understanding delivery requirements. ◆ Lack of clear communication with requesters.
	<b>Behavior:</b>	◆ Frequently checks the app for new requests. ◆ Prefers straightforward request details and easy navigation.


## 2.3 Logistic Requester (LRi) - Ivo

	<b>IVO</b>	<b>Role:</b>	<ul style="list-style-type: none"> <li>Starts but rarely completes delivery requests</li> </ul>
		<b>Goals:</b>	<ul style="list-style-type: none"> <li>Quickly explore delivery options.</li> <li>Avoid committing to requests without full clarity.</li> </ul>
		<b>Pain Points:</b>	<ul style="list-style-type: none"> <li>Finds the request process too complex.</li> <li>Often inputs incorrect data due to confusion.</li> </ul>
		<b>Behavior:</b>	<ul style="list-style-type: none"> <li>Frequently abandons requests midway.</li> <li>Prefers simpler forms and clearer instructions.</li> </ul>

## 2.4 Logistic Provider (LP) - Aga

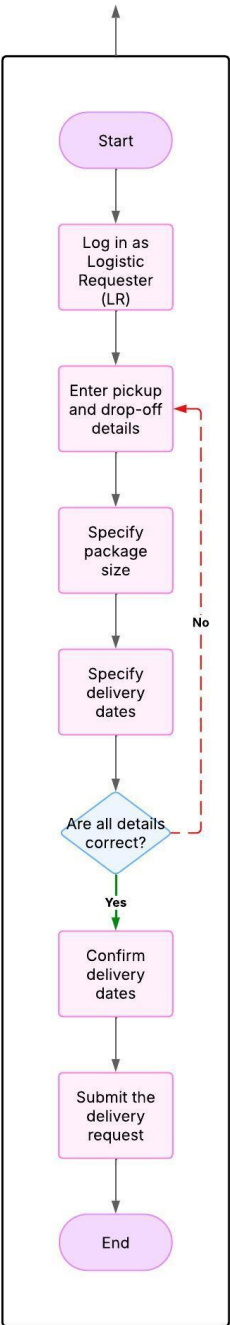
	<b>AGA</b>	<b>Role:</b>	<ul style="list-style-type: none"> <li>Struggles to fulfill delivery requests.</li> </ul>
		<b>Goals:</b>	<ul style="list-style-type: none"> <li>Accept and complete delivery requests.</li> <li>Avoid errors in delivery execution</li> </ul>
		<b>Pain Points:</b>	<ul style="list-style-type: none"> <li>Difficulty understanding delivery details.</li> <li>Often fails to deliver due to unclear instructions.</li> </ul>
		<b>Behavior:</b>	<ul style="list-style-type: none"> <li>Frequently makes mistakes in fulfilling requests.</li> <li>Needs more guidance and support from the app</li> </ul>

## 2.5 Support – Adam

	<b>ADAM</b>	<b>Role:</b>	<ul style="list-style-type: none"> <li>Backend support for the system.</li> </ul>
		<b>Goals:</b>	<ul style="list-style-type: none"> <li>Resolve user issues quickly.</li> <li>Ensure smooth operation of the app.</li> </ul>
		<b>Pain Points:</b>	<ul style="list-style-type: none"> <li>Lack of clear logs or reports to diagnose issues.</li> <li>Difficulty tracking user complaints.</li> </ul>
		<b>Behavior:</b>	<ul style="list-style-type: none"> <li>Monitors app performance and user feedback.</li> <li>Prefers detailed logs and easy-to-use support tools.</li> </ul>

### 3. User Case Journey

#### 3.1 Logistic Requester (LR) - Emma Creates a Transport Request

USER JOURNEY FLOWCHART Scenario: Emma Creates a Transport Request	Comments
 <pre>graph TD; Start([Start]) --&gt; Login[Log in as Logistic Requester (LR)]; Login --&gt; Enter[Enter pickup and drop-off details]; Enter --&gt; Size[Specify package size]; Size --&gt; Dates[Specify delivery dates]; Dates --&gt; Check{Are all details correct?}; Check -- No --&gt; Enter; Check -- Yes --&gt; Confirm[Confirm delivery dates]; Confirm --&gt; Submit[Submit the delivery request]; Submit --&gt; End([End]);</pre> <p>The flowchart illustrates the process of creating a transport request. It begins with a 'Start' oval, followed by a 'Log in as Logistic Requester (LR)' rectangle. The next step is 'Enter pickup and drop-off details', which leads to 'Specify package size', then 'Specify delivery dates'. A decision diamond asks 'Are all details correct?'. If the answer is 'No', a dashed red arrow loops back to the 'Enter pickup and drop-off details' step. If 'Yes', a solid green arrow leads to 'Confirm delivery dates', followed by 'Submit the delivery request', and finally 'End'.</p>	<ul style="list-style-type: none"><li>◆ <b>Oval (Start/End):</b><ul style="list-style-type: none"><li>○ Start: Clearly marks the beginning of the user journey.</li><li>○ End: Indicates the successful completion of the transport request process.</li></ul></li><li>◆ <b>Rectangle (Process Steps):</b><ul style="list-style-type: none"><li>○ Log in as Logistic Requester (LR): Essential first step to authenticate the user.</li><li>○ Enter pickup and drop-off details: Captures critical location information for the delivery.</li><li>○ Specify package size: Ensures the system knows the dimensions/weight of the package.</li><li>○ Specify delivery dates: Allows the user to set preferred delivery timelines.</li><li>○ Confirm delivery dates: Finalizes the timing details before submission.</li><li>○ Submit the transport request: Completes the process, sending the request to the system.</li></ul></li><li>◆ <b>Diamond (Decision Point):</b><ul style="list-style-type: none"><li>○ Are all details correct?: A crucial checkpoint to ensure accuracy before submission.<ul style="list-style-type: none"><li>■ Yes: Proceeds to confirm and submit the request.</li><li>■ No: Loops back to allow corrections (implied but not shown in detail).</li></ul></li></ul></li></ul>

### 3.2 Logistic Requester (LP) - Mark Accepts and Fulfills a Transport Request

<b>USER JOURNEY FLOWCHART</b> <b>Scenario: Mark accepts and fulfills a transport request.</b>	<b>Comments</b>
<pre> graph TD     Start([Start: Log in as Logistics Provider (LP)]) --&gt; Search[Search for available delivery requests]     Search --&gt; Decision1{Are suitable delivery requests available?}     Decision1 -- No --&gt; Search     Decision1 -- Yes --&gt; Accept[Accept a delivery request]     Accept --&gt; View[View delivery request details]     View --&gt; Decision2{Are all details correct and feasible?}     Decision2 -- No --&gt; Accept     Decision2 -- Yes --&gt; Proceed[Proceed with fulfilling the delivery]     Proceed --&gt; Begin[Begin delivery process]     Begin --&gt; Complete[Complete delivery operations]     Complete --&gt; Confirm[Confirm delivery completion]     Confirm --&gt; End([End])     </pre>	<ul style="list-style-type: none"> <li>◆ <b>Oval (Start/End):</b> <ul style="list-style-type: none"> <li>○ Start: Clearly marks the beginning of the Logistic Provider (LP) journey.</li> <li>○ End: Indicates the successful completion of the delivery process.</li> </ul> </li> <li>◆ <b>Rectangle (Process Steps):</b> <ul style="list-style-type: none"> <li>○ Log in as Logistics Provider (LP): Essential first step to authenticate the user.</li> <li>○ Search for available delivery requests: Allows the LP to find relevant delivery opportunities.</li> <li>○ Accept a delivery request: Enables the LP to commit to a specific request.</li> <li>○ View delivery request details: Provides critical information about the delivery (e.g., pickup/drop-off, package size).</li> <li>○ Proceed with fulfilling the delivery: Moves the LP into the execution phase.</li> <li>○ Begin delivery process: Marks the start of physical delivery operations.</li> <li>○ Complete delivery operations: Ensures the delivery is carried out as planned.</li> <li>○ Confirm delivery completion: Finalizes the process, updating the system and requester.</li> </ul> </li> <li>◆ <b>Diamond (Decision Point):</b> <ul style="list-style-type: none"> <li>○ Are suitable delivery requests available?: A key checkpoint to ensure the LP finds relevant requests. <ul style="list-style-type: none"> <li>▪ Yes: Proceeds to accept and view details.</li> <li>▪ No: Implies the LP may need to search again or wait for new requests (loop not explicitly shown).</li> </ul> </li> <li>○ Are all details correct and feasible?: Ensures the LP can fulfill the request before proceeding. <ul style="list-style-type: none"> <li>▪ Yes: Moves forward with delivery fulfillment.</li> <li>▪ No: Implies the LP may need to reject or request clarification (loop not explicitly shown).</li> </ul> </li> </ul> </li> </ul>

### 3.3 Logistic Requester (LR) - Emma Tracks Delivery Status

<b>USER JOURNEY FLOWCHART</b> <b>Scenario: Emma Tracks Delivery Status</b>	<b>Comments</b>
<pre> graph TD     Start([Start - Log in as Logistic Requester (LR)]) --&gt; View[View Active Delivery Requests]     View --&gt; Check[Check Delivery Status]     Check --&gt; Status{Delivery Status?}     Status -- "Awaiting Confirmation" --&gt; Notify[Notify LR of pending confirmation]     Status -- "In Transit" --&gt; Monitor[Monitor progress and estimate delivery time]     Status -- "Other Status" --&gt; Investigate[Investigate issue with the provider]     Notify --&gt; Contact{Contact Provider?}     Investigate --&gt; Contact     Monitor --&gt; Contact     Contact -- Yes --&gt; Chat[Open Chat with Provider]     Contact -- No --&gt; End([End - Communication with Provider])     Chat --&gt; Completed[Delivery Process Completed]     Completed --&gt; End     Monitor --&gt; Confirm[Confirm delivery completion with LR]     Confirm --&gt; End </pre>	<ul style="list-style-type: none"> <li>♦ <b>Oval (Start/End):</b> <ul style="list-style-type: none"> <li>○ Start: Clearly marks the beginning of the user journey.</li> <li>○ End: Indicates completion of the process, focusing on communication, confirmation.</li> </ul> </li> <li>♦ <b>Rectangle (Process Steps):</b> <ul style="list-style-type: none"> <li>○ Log in as Logistic Requester (LR): Essential first step to authenticate the user.</li> <li>○ View Active Delivery Requests: Allows the LR to see ongoing deliveries.</li> <li>○ Check Delivery Status: Provides real-time updates on the delivery progress.</li> <li>○ Notify LR of pending confirmation: Alerts the LR if the delivery is awaiting confirmation.</li> <li>○ Investigate issue with the provider: Ensures problems are addressed swiftly.</li> <li>○ Monitor progress and estimate delivery time: Keeps the LR informed about timeline.</li> <li>○ Open Chat with Provider: Enables direct communication between the LR and LP.</li> <li>○ Confirm delivery completion with LR: Finalizes the process, ensuring the LR is satisfied.</li> </ul> </li> <li>♦ <b>Diamond (Decision Point):</b> <ul style="list-style-type: none"> <li>○ Delivery Status?: A key checkpoint to determine current state of the delivery. <ul style="list-style-type: none"> <li>▪ Awaiting Confirmation: Triggers a notification to the LR.</li> <li>▪ In Transit: Leads to monitoring progress and estimated delivery time.</li> <li>▪ Other Status: Implies additional actions may be needed (not explicitly detailed).</li> </ul> </li> <li>○ Contact Provider?: Decides whether the LR needs to communicate directly with the provider. <ul style="list-style-type: none"> <li>▪ Yes: Opens a chat with LP.</li> <li>▪ No: Implies the LR is satisfied with the current status (not explicitly shown).</li> </ul> </li> </ul> </li> </ul>

### 3.4 Logistic Requester (LRi) - Ivo Abandons a Transport Request

<b>USER JOURNEY FLOWCHART</b> <b>Scenario: Ivo Abandons a Transport Request</b>	<b>Comments</b>
<pre> graph TD     Start([Start]) --&gt; Login[Log in as Logistic Requester (LRi)]     Login --&gt; StartReq[Start delivery request]     StartReq --&gt; Error{Is there an error in the delivery request process?}     Error -- Yes --&gt; Encounter[Encounter error: Prompt error message]     Encounter --&gt; Attempt[Attempt to resolve the error]     Attempt --&gt; Resolved{Successfully resolved the error?}     Resolved -- Yes --&gt; Navigate[Navigate through delivery options]     Resolved -- No --&gt; Stuck[Stuck: Unable to resolve]     Navigate --&gt; Clear{Is the process clear and simple to follow?}     Clear -- No --&gt; Stuck     Clear -- Yes --&gt; Consider[Consider abandoning the delivery request]     Stuck --&gt; Consider     Consider --&gt; Decision{Decision to proceed or abandon?}     Decision -- Proceed --&gt; Continue[Continue with delivery request]     Decision -- Abandon --&gt; End([End: Delivery request abandoned])     Continue --&gt; Complete[Complete the delivery request successfully]     Complete --&gt; End   </pre>	<ul style="list-style-type: none"> <li>◆ <b>Oval (Start/End):</b> <ul style="list-style-type: none"> <li>○ Start: Marks the beginning of the Logistic Requester (LRi) journey.</li> <li>○ End: Indicates the user abandons the process due to complexity or errors.</li> </ul> </li> <li>◆ <b>Rectangle (Process Steps):</b> <ul style="list-style-type: none"> <li>○ Log in as Logistic Requester (LRi): Essential first step to authenticate the user.</li> <li>○ Start transport request: Initiates the process of creating a transport request.</li> <li>○ Encounter error / Prompt error message: Highlights a potential issue in the process.</li> <li>○ Attempt to resolve the error: Shows the user's effort to fix the problem.</li> <li>○ Navigate through delivery options: Represents the user exploring available choices.</li> <li>○ Continue with delivery request: Indicates the user decides to proceed despite challenges.</li> <li>○ Complete the delivery request successfully: Marks the successful completion of the process.</li> <li>○ Abandon delivery request: Represents the user giving up due to complexity or errors.</li> </ul> </li> <li>◆ <b>Diamond (Decision Point):</b> <ul style="list-style-type: none"> <li>○ Is there an error in the transport request process?: A key checkpoint to identify issues.               <ul style="list-style-type: none"> <li>▪ Yes: Triggers error message, resolution attempt.</li> <li>▪ No: Allows the user to proceed without issues.</li> </ul> </li> <li>○ Successfully resolved the error?: Determines whether the user can continue.               <ul style="list-style-type: none"> <li>▪ Yes: Moves forward with the transport request.</li> <li>▪ No: Leads to potential abandonment.</li> </ul> </li> <li>○ Is the process clear and simple to follow?: Evaluates the user's experience.               <ul style="list-style-type: none"> <li>▪ Yes: Encourages the user to proceed.</li> <li>▪ No: Leads to frustration and potential abandonment.</li> </ul> </li> <li>○ Decision to proceed or abandon?: Final choice for user.               <ul style="list-style-type: none"> <li>▪ Proceed: Continues with the delivery request.</li> <li>▪ Abandon: Ends the process without completion.</li> </ul> </li> </ul> </li> </ul>

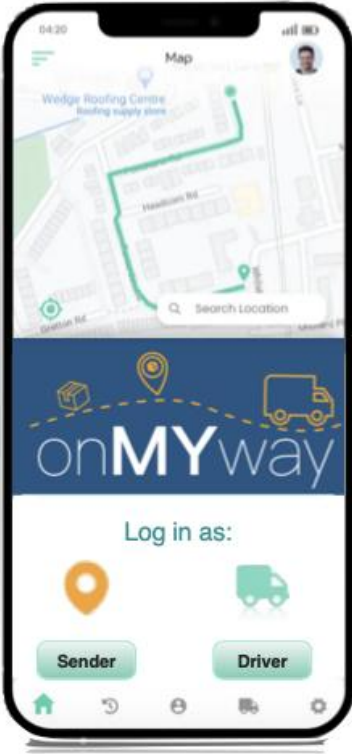



### 3.5 Support - Adam Resolves a User Issue


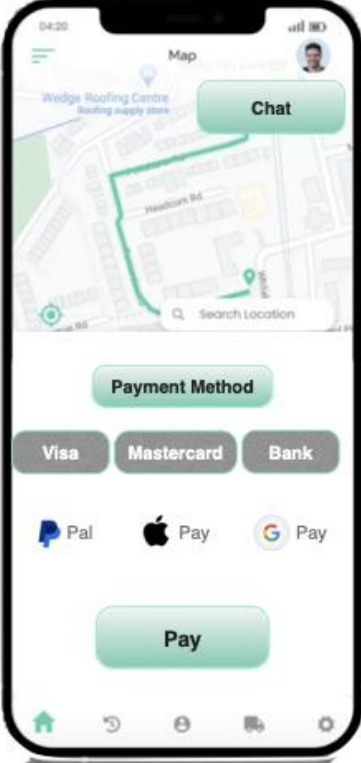
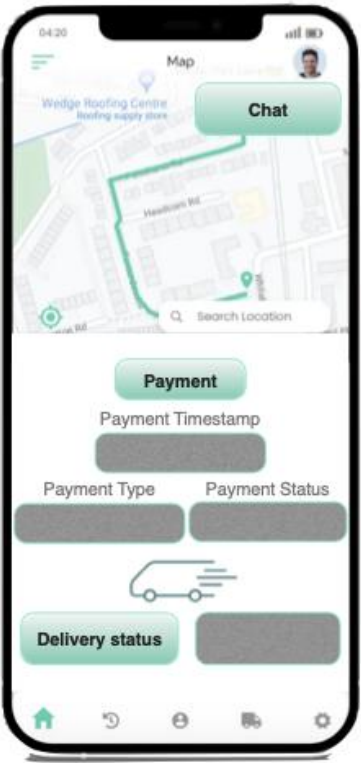
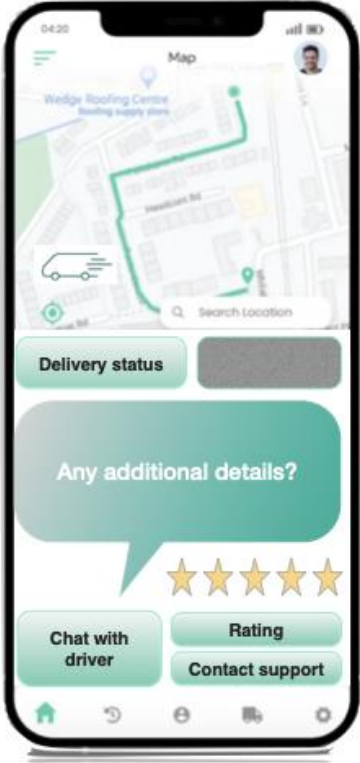
<b>USER JOURNEY FLOWCHART</b> <b>Scenario:</b> Adam resolves a user issue.	<b>Comments</b>
<pre> graph TD     Start([Start]) --&gt; Receive[Receive a support ticket or notification]     Receive --&gt; Access[Access user logs and delivery details]     Access --&gt; Decision1{Decision: Is the issue clear from the logs?}     Decision1 -- Yes --&gt; Resolve[Resolve the issue based on log information]     Decision1 -- No --&gt; Communicate[Communicate with the user to understand the issue]     Communicate --&gt; Decision2{Decision: Is the issue resolved?}     Decision2 -- Yes --&gt; Confirm[Confirm resolution and close the ticket]     Decision2 -- No --&gt; Communicate     Resolve --&gt; Confirm     Confirm --&gt; End([End])         </pre>	<ul style="list-style-type: none"> <li>♦ <b>Oval (Start/End):</b> <ul style="list-style-type: none"> <li>○ Start: Marks the beginning of the support process.</li> <li>○ End: Indicates the successful resolution and closure of the support ticket.</li> </ul> </li> <li>♦ <b>Rectangle (Process Steps):</b> <ul style="list-style-type: none"> <li>○ Receive a support ticket or notification: Initiates the support process.</li> <li>○ Access user logs and delivery details: Provides the support team with necessary information to diagnose the issue.</li> <li>○ Communicate with the user to understand the issue: Ensures clarity if the logs are insufficient.</li> <li>○ Resolve the issue based on log information: Directly addresses the problem using available data.</li> <li>○ Confirm resolution and close the ticket: Finalizes the process, ensuring the issue is resolved.</li> </ul> </li> <li>♦ <b>Diamond (Decision Point):</b> <ul style="list-style-type: none"> <li>○ Is the issue clear from the logs?: A key checkpoint to determine if the logs provide enough information.               <ul style="list-style-type: none"> <li>▪ Yes: Proceeds to resolve the issue directly.</li> <li>▪ No: Requires communication with the user for further clarification.</li> </ul> </li> <li>○ Is the issue resolved?: Ensures the problem is fully addressed before closing the ticket.               <ul style="list-style-type: none"> <li>▪ Yes: Confirms resolution and closes the ticket.</li> <li>▪ No: Implies further action is needed (though the loop is not explicitly shown).</li> </ul> </li> </ul> </li> </ul>

## 4. Layout Examples (Screens Simulations)

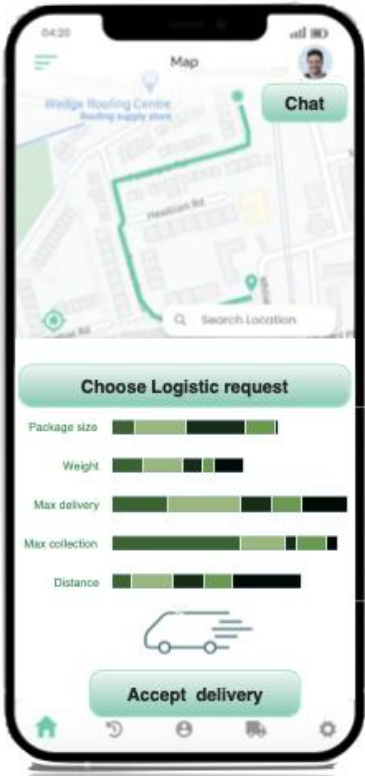


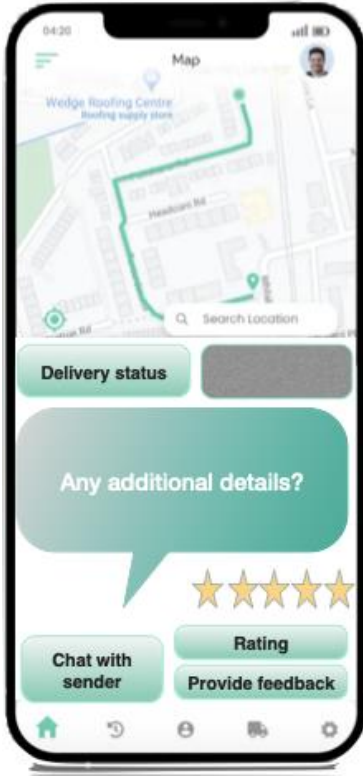
### 4.1 Sign In / Sign Up Interfaces

Sign In Screen	Sign Up Screen
 The Sign In screen features a map at the top showing a route. Below the map is the 'onMYway' logo with a truck icon. Underneath the logo, it says 'Log in as:' followed by two buttons: 'Sender' with a location pin icon and 'Driver' with a truck icon. The bottom of the screen has a navigation bar with five icons: a house, a clock, a person, a truck, and a gear.	 The Sign Up screen features a map at the top showing a route. Below the map is the 'onMYway' logo with a truck icon. Underneath the logo, it says 'Sign up' followed by six input fields: 'First name', 'Last name', 'Phone number', 'Adress', 'Email', and 'Pasword'. The bottom of the screen has a navigation bar with five icons: a house, a clock, a person, a truck, and a gear.

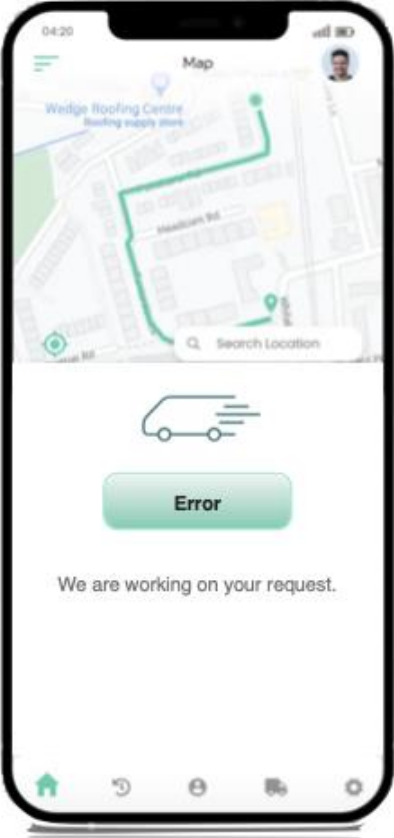
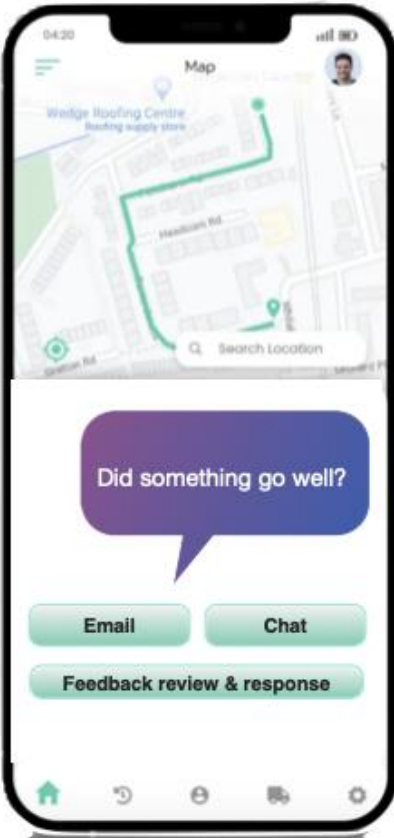

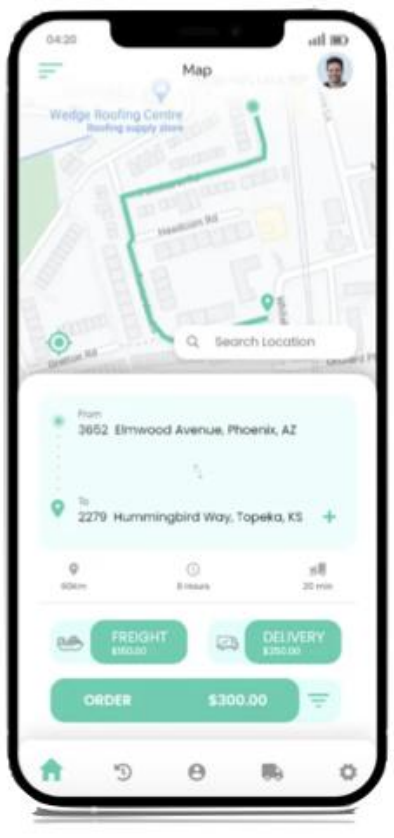
4.2 Logistic Requester (LR) Interfaces

Create a Request Screen	Payment Method Screen	Delivery Status Screen	Chat Screen
 The 'Create a Request Screen' shows a map at the top with a location pin. Below the map are two columns of input fields. The left column has 'Price' (with a 'Euro' button), 'Pick up City', 'House Number', 'Drop off City', 'House Number', 'Min delivery Date', 'Max delivery Date', and 'Package size: KG' (with size buttons: XS, S, M, L, XL, XXL). The right column has 'Pick up Street', another 'House Number', 'Drop off Street', and another 'House Number'. A 'Create a request' button is at the bottom.	 The 'Payment Method Screen' features a map at the top with a 'Chat' button. Below the map is a 'Payment Method' button, followed by three buttons: 'Visa', 'Mastercard', and 'Bank'. Below these are three logos: 'Pal', 'Apple Pay', and 'Google Pay'. A large 'Pay' button is at the bottom.	 The 'Delivery Status Screen' shows a map at the top with a 'Chat' button. Below the map is a 'Payment' button, followed by a 'Payment Timestamp' input field. Below that are two input fields for 'Payment Type' and 'Payment Status'. A delivery truck icon is shown, followed by a 'Delivery status' button and an input field.	 The 'Chat Screen' displays a map at the top with a 'Chat' button. Below the map is a 'Delivery status' button and an input field. A large speech bubble contains the text 'Any additional details?'. Below the speech bubble are five yellow stars. At the bottom are three buttons: 'Chat with driver', 'Rating', and 'Contact support'.

4.3 Logistic Provider (LP) Interfaces

Accept a Request Screen	Search Screen	Registration Screen	Chat Screen
 The 'Accept a Request Screen' shows a map at the top with a green route and a 'Chat' button. Below the map is a 'Choose Logistic request' section with sliders for 'Package size', 'Weight', 'Max delivery', 'Max collection', and 'Distance'. At the bottom is a green 'Accept delivery' button.	 The 'Search Screen' features a map at the top. Below it are input fields for 'Pick up City', 'Pick up Street', 'House Number', 'Drop off City', 'Drop off Street', 'House Number', 'Min delivery Date', and 'Max delivery Date'. There are also 'Price' and 'Euro' buttons, a 'Package size' section with 'KG' and size buttons (XS, S, M, L, XL, XXL), and a 'Search' button at the bottom.	 The 'Registration Screen' has a green header with 'Vehicals' and a truck image. It lists 'STRAALIS XS Hi-WAY' and 'Max Load 850 Kgs'. Below are input fields for 'Name & Surname', 'Email', 'Address', 'Phone number', and 'Fuel type'. There are also icons for '80 hp (375 Kw)' and 'Ignore City Road'.	 The 'Chat Screen' shows a map at the top. Below it is a 'Delivery status' section with a green button. A large green speech bubble contains the text 'Any additional details?'. Below the speech bubble are five yellow stars, a 'Rating' button, and a 'Provide feedback' button. At the bottom is a 'Chat with sender' button.

5. Other Interface Elements

Error Screen	Contact Support Screen	Support Audit Screen	Order Recap Screen
 <p>The Error Screen displays a map at the top with a green route and a location pin. Below the map is a green button labeled "Error" and the text "We are working on your request." The bottom navigation bar is visible.</p>	 <p>The Contact Support Screen displays a map at the top. Below the map is a purple speech bubble containing the text "Did something go well?". Underneath the speech bubble are three green buttons: "Email", "Chat", and "Feedback review &amp; response". The bottom navigation bar is visible.</p>	 <p>The Support Audit Screen displays a map at the top. Below the map are three green buttons: "Performance Analysis", "Compliance Analysis", and "Ratings". Underneath the "Ratings" button is a pie chart showing the following data: 38%, 31%, 12%, 11%, and 9%. The bottom navigation bar is visible.</p>	 <p>The Order Recap Screen displays a map at the top. Below the map is a green button labeled "ORDER" with the text "\$300.00". Above the "ORDER" button are two green buttons: "FREIGHT \$150.00" and "DELIVERY \$200.00". The bottom navigation bar is visible.</p>

## Appendix A – Glossary

- ◆ **LR:** Logistic Requester (someone who needs transportation).
- ◆ **LP:** Logistic Provider (someone who offers transportation).
- ◆ **LRi:** Logistic Requester (Incomplete): User who starts but rarely completes Transport requests.
- ◆ **UI:** User Interface, the visual and interactive elements of the application.
- ◆ **UX:** User Experience, The Overall experience of users interacting with the system.<sup>1</sup>

---

### <sup>1</sup>Methodology Note:

This document was developed through a structured, user-centered design process by a collaborative team:

1. **User Personas:** The team identified key user roles (e.g., Logistic Requester, Logistic Provider, Support) and detailed their goals, pain points, and behaviors.
2. **User Case Journeys:** Based on the personas, the team mapped out workflows for scenarios like creating requests, fulfilling deliveries, tracking status, and resolving issues. Flowcharts were created to visualize these journeys, highlighting decision points and pain points.
3. **Screen Drafts:** The team designed draft layouts for key interfaces (e.g., Sign In, Create Request, Delivery Status, Chat) to align with the user journeys and personas.
4. **Refinement:** Through iterative feedback and collaboration, the team refined the personas, journeys, and screens to ensure they accurately reflected user needs and system goals.