

ShiffTERS



Muhammad Kumail Raza

01-135221-036

Hussain ibne Shah

01-135221-018

Supervisor: Jawwad Ijaz

Bachelors of Science in Information Technology

Department of Computer Science

Bahria University, Islamabad

December 2025

Certificate

We accept the work contained in the report titled "SHIFTERS", written by Kumail Raza AND Hussain ibne Shah as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by ...:

Supervisor: Jawwad Ijaz

Internal Examiner:

External Examiner:

Project Coordinator: Dr. Kabir Ahmed Bhatti

Head of the Department: Dr. Khurram Ehsan

Abstract

The relocation of residential properties in Pakistan presents significant challenges for consumers. Traditional moving services frequently employ opaque pricing structures, resulting in unexpected additional charges that were not communicated during initial consultations. Furthermore, customers regularly experience damaged or missing belongings, with limited accountability from service providers. These persistent issues within the industry necessitated the development of an innovative solution. This paper presents Shifters, a comprehensive mobile and web application designed to connect users with verified moving companies while ensuring transparency and accountability throughout the relocation process. The platform was developed using Flutter for cross-platform compatibility, Firebase for authentication and real-time data management, and YOLO (You Only Look Once) for artificial intelligence-powered object recognition and real-time inventory management capabilities. The development methodology followed a systematic approach encompassing user consultation, system architecture design, implementation, and rigorous testing protocols. A distinguishing feature of the application is its AI-powered camera functionality, which enables automatic recognition and cataloging of furniture and boxes through computer vision. This eliminates the traditional manual inventory process conducted by moving personnel, which is prone to errors and omissions. The platform provides users with real-time truck tracking capabilities, transparent cost estimation, and comprehensive digital documentation, thereby eliminating paper-based processes. Empirical validation conducted with actual users demonstrated measurable improvements across multiple metrics: significant reduction in lost items, decreased completion time for relocation services, and substantially lower stress levels reported by customers throughout the moving experience.

Acknowledgments

In the name of Allah, the most beneficent and the most merciful. We are highly grateful to the One who created us and blessed us with a privileged life. We would like to thank our parents who have always been a pillar of strength and support. We are thankful to our parents who taught us that how hard work, devotion, and working towards your goal with pure intention can take you to places.

Furthermore, we would like to thank and appreciate our supervisor Sir, Jawwad Ijaz who has given us a chance to work on a project that can help in creating value for our beloved country. His professional guidance, time, and effort will always be remembered. Last but not the least, we would like to appreciate our friends who make studying and hard times less challenging. May ALLAH (SWT) guide us to the right path.

Muhammad Kumail Raza & Hussain Ibne Shah
Bahria University
E-8, Islamabad, Pakistan

December, 2025

“Success belongs to those who remain patient in hardships.”

Imam Hussain عَلَيْهِ السَّلَام

Table of Contents

Abstract	iv
1 Introduction	1
1.1 Overview	1
1.2 Objective	2
1.3 Problem Description	2
1.4 Methodology	3
1.4.1 Collecting Requirements	3
1.4.2 Analysis	4
1.4.3 Design	4
1.4.4 Development	4
1.4.5 Maintenance/testing	4
1.5 Project Scope	4
1.6 Feasibility Study	5
1.7 Solution Application Areas	5
1.8 Tools and Technology	6
1.8.1 Flutter	6
1.8.2 Firebase	7
1.8.3 YOLO	7
1.8.4 APIs and Third-Party Services	7
1.8.5 Hardware	7

1.9	Expertise of the Team Members	7
1.10	Milestones	8
2	Literature Review	9
2.1	Related Work	9
2.1.1	Example Analysis	10
2.2	Comparative analysis	11
3	Requirements Specifications	12
3.1	Proposed System	12
3.2	Existing System	13
3.3	Functional Requirements	13
3.4	Non-Functional Requirements	14
3.5	Use Case Diagram	15
3.5.1	Register	16
3.5.2	Login	17
3.5.3	Forgot Password	18
3.5.4	Manage Profile	19
3.5.5	Requesting Moving Service	20
3.5.6	Get Price Estimate	21
3.5.7	Track Moving Status	22
3.5.8	Logout	23
3.5.9	Driver Register	24
3.5.10	Driver View Available Jobs	25
3.5.11	Driver Update Job Status	26
3.5.12	Driver Accept/Reject Job	27
3.5.13	AI Scan and Recognize Items	28
3.5.14	AI Compare Loaded Unloaded Items	29
3.5.15	AI Alert Missing Items	30
3.5.16	View Total Drivers Available	31
3.5.17	Update/Delete Orders	32

3.5.18	Update/Delete Users	33
3.5.19	Manage Users	34
3.5.20	Manage Drivers	35
3.5.21	Manage Orders	36
3.5.22	Verify Driver Documents	37
4	System Design	38
4.1	System Architecture	38
4.2	System Architecture Diagram	38
4.3	High Level Design	40
4.3.1	Sequence Diagram	40
4.4	Sequence of Actions	63
4.4.1	Activity Diagram	63
4.5	GUI Design	67
4.5.1	Splash Screen and Sign Up / Sign In Screen	67
4.5.2	Login Screen and Create Account Screen	68
4.5.3	Profile Screen and Home Screen	69
4.5.4	View Orders Screen and In Transit Orders Screen	70
4.5.5	Driver Dashboard Screen and Driver Profile Screen	71
4.6	Activity Workflow	72
4.7	GUI Design Overview	72
5	System Implementation	73
5.1	Introduction	73
5.2	System Architecture	73
5.3	System Internal Components	74
5.3.1	Registration and Authentication Module	74
5.3.2	Password Recovery Module	74
5.3.3	Inventory Management Module	74
5.3.4	Real-Time Tracking Module	74
5.3.5	Quote Request and Cost Estimation Module	74

5.3.6	Order Management Module	75
5.3.7	Driver Management Module	75
5.3.8	Report Generation Module	75
5.3.9	Session Management Module	75
5.3.10	Query Management Module	75
5.3.11	Alert Missing Items Module	75
5.3.12	Driver Job Status Update Module	75
5.3.13	Admin Dashboard Module	76
6	System Testing	77
6.1	System Testing	77
6.2	Graphical User Interface (GUI) Testing	78
6.2.1	Register	78
6.2.2	Login	78
6.2.3	Forgot Password	79
6.2.4	Logout	79
6.3	Usability Testing	80
6.3.1	Quote Request and Cost Estimation	80
6.4	Performance Testing	81
6.4.1	System Performance Testing	81
7	Conclusion	82
	References	84

List of Figures

1.1	Milestones	8
3.1	Use Case Diagram for Shifters	15
3.2	Register Use Case	16
3.3	Login Use Case	17
3.4	Forgot Password Use Case	18
3.5	Manage Profile Use Case	19
3.6	Requesting Moving Service Use Case	20
3.7	Get Price Estimate Use Case	21
3.8	Track Moving Status Use Case	22
3.9	Logout Use Case	23
3.10	Driver Register Use Case	24
3.11	Driver View Available Jobs Use Case	25
3.12	Add Products to Cart Use Case	26
3.13	Driver Accept/Reject Job Use Case	27
3.14	AI Scan and Recognize Items Use Case	28
3.15	AI Compare Loaded Unloaded Items Use Case	29
3.16	AI Alert Missing Items Use Case	30
3.17	View Total Drivers Available Use Case	31
3.18	Update/Delete Orders Use Case	32
3.19	Update/Delete Users Use Case	33
3.20	Manage Users Use Case	34

3.21	Manage Drivers Use Case	35
3.22	Manage Orders Use Case	36
3.23	Verify Driver Documents Use Case	37
4.1	System Architecture Diagram	39
4.2	Registration Sequence Diagram	41
4.3	User Login Sequence Diagram	42
4.4	Forgot Password Sequence Diagram	43
4.5	AI Scan and Recognize Items Sequence Diagram	44
4.6	AI Compare Loaded and Unloaded Items Sequence Diagram	45
4.7	AI Alert Missing Items Sequence Diagram	46
4.8	Driver Registration Sequence Diagram	47
4.9	Driver Accept / Reject Job Sequence Diagram	48
4.10	Driver Update Job Status Sequence Diagram	49
4.11	Driver View Available Jobs Sequence Diagram	50
4.12	Admin Manage Drivers Sequence Diagram	51
4.13	Admin Manage Users Sequence Diagram	52
4.14	Admin Manage Orders Sequence Diagram	53
4.15	Admin Update / Delete Orders Sequence Diagram	54
4.16	Admin Update / Delete Users Sequence Diagram	55
4.17	User Get Price Estimate Sequence Diagram	56
4.18	User Manage Profile Sequence Diagram	57
4.19	User Request Moving Service Sequence Diagram	58
4.20	User Track Moving Status Sequence Diagram	59
4.21	Admin Verify Driver Documents Sequence Diagram	60
4.22	Admin View Total Drivers Available Sequence Diagram	61
4.23	Logout Sequence Diagram	62
4.24	Activity Diagram	64
4.25	Activity Diagram	65
4.26	Activity Diagram	66
4.27	Key GUI screens of the Shifters application	67

4.28 Login and Account Creation Interfaces	68
4.29 User Profile and Home Interface	69
4.30 View Orders Screen	70
4.31 In Transit Orders Screen	70
4.32 Driver Dashboard and Driver Profile Screens	71

List of Tables

2.1	Comparison of Existing Solutions with Shifters	11
3.1	Use Case Description for Register	16
3.2	Use Case Description for Login	17
3.3	Use Case Description for Forgot Password	18
3.4	Use Case Description for Manage Profile	19
3.5	Use Case Description for Requesting Moving Service	20
3.6	Use Case Description for Get Price Estimate	21
3.7	Use Case Description for Track Moving Status	22
3.8	Use Case Description for Logout	23
3.9	Use Case Description for Driver Register	24
3.10	Use Case Description for View Available Jobs	25
3.11	Use Case Description for Update Job Status	26
3.12	Use Case Description for Accept/Reject Job	27
3.13	Use Case Description for Scan and Recognize Items	28
3.14	Use Case Description for Compare Loaded and Unloaded Items	29
3.15	Use Case Description for Alert Missing Items	30
3.16	Use Case Description for View Total Drivers Available	31
3.17	Use Case Description for Update/Delete Orders	32
3.18	Use Case Description for Update/Delete Users	33
3.19	Use Case Description for Manage Users	34
3.20	Use Case Description for Manage Drivers	35

3.21	Use Case Description for Manage Orders	36
3.22	Use Case Description for Verify Driver Documents	37
6.1	Test Case for Register	78
6.2	Test Case for Login	78
6.3	Test Case for Forgot Password	79
6.4	Test Case for Logout	79
6.5	Test Case for Quote Request and Cost Estimation	80
6.6	Test Case for System Performance	81

Chapter 1

Introduction

1.1 Overview

Shiffters is a ground breaking mobile application that is designed to take the moving and relocation industry by storm with innovative technology and user convenience. The app matches customers with trusted movers while solving common challenges such as inefficiencies, miscommunication and lack of transparency.

Logistics companies are more and more turning to AI models for finding the optimal routes and recognising the inventory, which reduces the operational cost and enhances the accuracy [1]. We made Shiffters based on this idea but we looked particularly at what about moving being stressful. The app uses object recognition to scan your belongings while loading and unloading so you get a complete digital record of everything in your belongings. No more worrying about whether something was left out, or a dispute on what was packed.

The price estimation also works in real time. Point your phone at objects and the phone works out the price on the spot. If something is missing during unloading, you will know as soon as you do. Everyone involved can track the tasks as they occur, whether the customer who is waiting at home, the driver on the road or the moderator who is coordinating everything.

1.2 Objective

We made Shifters to repair what's broken in Pakistan's moving industry. The object recognition feature tracks your stuff during the loading and unloading process so no confusion of what got loaded or unloaded. This cuts down on the disputes and things being lost. Real-time updates keep people informed instead of having to make educated guesses with things. The interface is simple because nobody wants to be dealing with complicated apps at a time when they're already stressed about moving. We wanted to bring some order to an industry that's been very chaotic for decades-make it easier for customers to find reliable movers to help them move, and actually trust the process.

1.3 Problem Description

We built Shifters to make what's broken in Pakistan's moving industry. The object recognition feature keeps track of your stuff during loading and unloading, and you'll never have a question as to what was moved. This reduces the number of disputes and lost items. Real-time updates ensure that everyone is in the know about where things are as opposed to guessing. The interface is relatively simple because no one wants to deal with complicated apps when they are already stressed out from moving. We wanted to bring some order to an industry that's been in chaos for decades, it being easier for customers to find reliable movers and actually trust the process.

Lack of Transparency: You ring off for a quote and receive some vague quote. Then when the bill comes, it's way more because there's no proper system for keeping track of what actually got moved.

Inefficient Processes: Everything's done manually. Writing down inventory lists, assigning jobs, it is all. This implies delays and wasted resources. You've got movers who are overbooked, or sitting around with nothing to do, customers who stress about their tight moving schedules.

Unreliable Service Providers: It is not easy to find a mover you can trust in

Pakistan. Too many people have a story about damaged furniture or a missing box or movers just not showing up professionally. It's killed off the trust in the whole industry.

Communication Gaps: Nobody knows what is happening. Customers don't know where their stuff is, movers don't get well streamlined instructions, drivers are confused about schedules. Without getting all that information at the time, everyone's just guessing and getting twisted.

Low Adoption of Technology: The industry has not kept up with technology at all. Nothing in terms of apps to track, digital inventory, or online payments. This means it is impossible to package it up on the scale to provide the fundamental amenities which people have become used to nowadays.

These problems aren't just annoying, they're holding them all back, the whole industry. With the help of object recognition technology and real-time updates of this technology we can actually fix these issues and the services of moving can work properly in Pakistan. More efficient and reliable and something that customers can actually depend on.

1.4 Methodology

1.4.1 Collecting Requirements

We began with talking to people that would use the app themselves. Did surveys - interviewed potential customers as well as moving companies - looked at what competitors were doing wrong. Our next step was to ask them about their biggest frustrations around moving, to check what would be reasonably possible from the existing technology and check that the cultural stuff isn't getting lost on us when we are in Pakistan.

1.4.2 Analysis

After getting all that data collected we sat down and looked for patterns. What were some of the common complaints? What could there be a realistic solution that we fix? And then we took all those messy notes and feedback and we turned it into clear requirements as to what the app needed to do.

1.4.3 Design

We designed interfaces that people in Pakistan would easily use. Not just translating some sort of app from the West, but actually thinking about the way people here interact with technology and what they're okay with and what makes sense culturally.

1.4.4 Development

Built app using Flutter so that it will work on different devices, middleware with the help of Firebase to perform all the back-end stuff for the app and TensorFlow for all the smart features. We didn't just code everything all one creates. Kept Testing Pieces As We Went Got Feedback Fixed Problems Tested Again Not just in general, but made sure it worked for Pakistan's certain situation.

1.4.5 Maintenance/testing

Conducted tons of tests, automated and manual. Wanted to eat bugs before users did pc catch We're still watching things, repairing things quickly whenever they come up. The aim is to keep the app running smoothly for the Pakistani market in particular.

1.5 Project Scope

Shiffters is focused on solving the largest issues faced by the moving industry in Pakistan with the application of practical tech solutions. The functions include object recognition giving trace of inventory automatically, real-time updates, so ev-

everyone knows what's happening, user authentication which provides security and an integrated payment system. We wanted to keep things fairly simple rather than attempting to try and do everything at once.

What we're not doing is a hardware development, the offline mode, not doing complex analytics. Those would've pulled resources away at getting the core app working really well. Pakistan had some very specific logistical headaches and the understanding was, let's build something people can count on, not something that's going to have great features, just because we can.

1.6 Feasibility Study

1. **Risks Involved:** There definitely are some risks that we needed to think about. The object recognition model might not always get things right, how to integrate different systems can be tricky, app performance can potentially slow down if we're not careful, users might not want to go for it, and we need to bother with keeping their data private. We're dealing with these through a lot of testing, making sure to create proof-of-concept versions, making sure that it runs smoothly on phones, making sure the interface is simple, as well as making sure to use proper security.
2. **Resource Requirement:** We need people that know what they're doing, decent computers to work on, phones to test on, Flutter SDKs, YOLO to recognize objects in and payment gateways and cloud servers to make it all work.

Studies of last mile logistics reports that user acceptance is a key factor for successful deployment of consumer-facing delivery apps [2].

1.7 Solution Application Areas

Target Industry: Moving and Relocation Services

Real Value:

- Makes moving faster and makes it less complicated
- Fewer errors with inventory and price
- Customers actually have a better experience

Benefits to the Industry:

- **Efficiency:** Tracking of the items during the process (loading and unloading) ensures that nothing gets lost and everybody remains accountable.
- **Accuracy:** Digital records mean that items don't need to get lost during the move.
- **Transparency:** Customers can see exactly what has been loaded, no more disputes.
- **Competitive Edge:** Companies utilizing tech are unique to younger and tech savvy customers.
- **Data Insights:** The data collected helps companies to make better business decisions.
- **Cost Reduction:** Automation reduces manual labor and reduces cost.
- **Scalability:** Moving companies can manage much more clients without hiring tons more people.

1.8 Tools and Technology

The core technology stack includes:

1.8.1 Flutter

Google's Maven Googfeet, two Google software developers from Google project, allows us to code once and run on phones, web browsers and computers. Instead of having to write an Android app and an iOS app we get both from the same codebase.

1.8.2 Firebase

In our case, the backend needs are taken care of by Firebase. Real time database, user logins, cloud storage, analytics, all in one place. Works great with Flutter and is able to keep sync of data across devices instantly.

1.8.3 YOLO

We use YOLO for object recognition since it's fast to use and it's suitable for use on mobile devices. Point your camera at furniture, boxes, etc., and YOLO will tell you what it's looking at via the camera, and in real-time. Perfect for the sort of quick scanning that we need during the loading and unloading.

1.8.4 APIs and Third-Party Services

We use existing APIs for things like payments and maps and not build them ourselves. Why reinvent the wheel when there are some sound solutions out there? It's more on top speed and security and allows us to focus on the things Shifters are unique.

1.8.5 Hardware

- Laptop
- Mobile Phone

1.9 Expertise of the Team Members

We're Kumail Raza and Hussain Ibne Shah, we've been using some different technologies with flutter, firebase, yolo, we're going to use all of it here. Between those two of us, we have mobile app development covered pretty well. We've built apps before but the reason why Shifters is different, way more real world project that we saw an issue in Pakistan.

1.10 Milestones

Note: This is the approximate milestone.

This timeline shows the key development phases of the Shifters project from start to finish. It includes requirements gathering, system design, implementation of core features like object recognition and real-time tracking, testing phases, and final deployment. Each milestone represents a major step in bringing the app from concept to a working product.

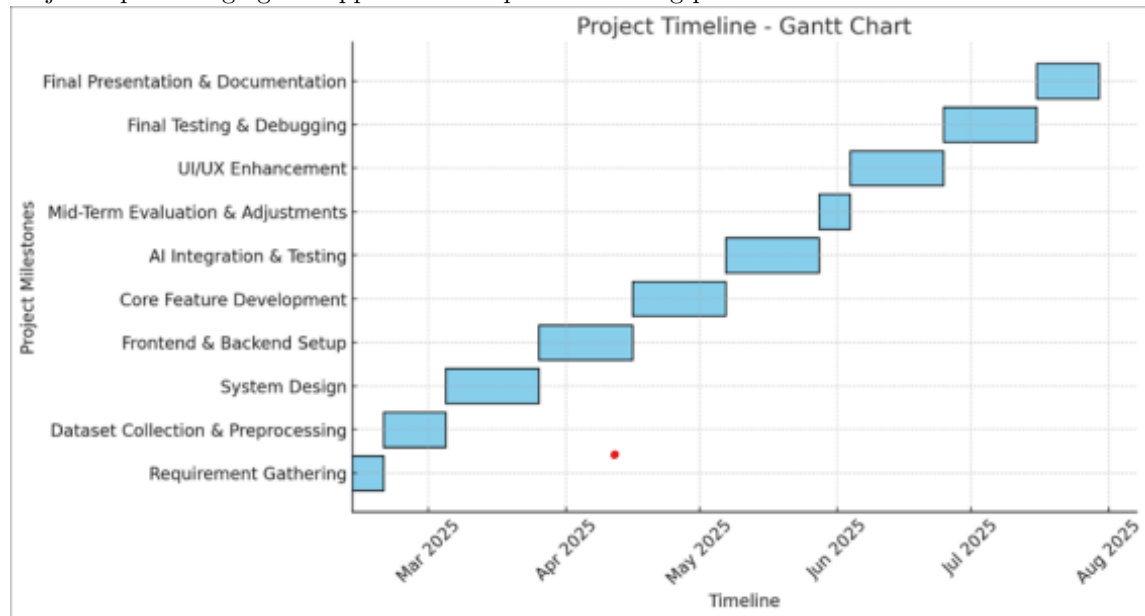


Figure 1.1: Milestones

Chapter 2

Literature Review

This chapter examines the existing capacities in the space of moving and relocation and where things fall short. "Part of what we wanted to see were what other people had tried, what worked, what didn't work and where Shifters could actually make a difference." Moovaz is an example of a company that is changing the way people move by putting pricing, coordinating, and tracking all in one place by way of digital platforms. [3].

The moving industry does have real problems. Things are inefficient and there is not much transparency and mostly companies have not adopted much technology. Recent research on mobile logistics apps illustrate some of the useful techniques to manage inventory by using images [4]. We're looking at looking at what's missing in current systems to figure out what Shifters needs to do differently. Yembo uses computer vision to scan household items out of videos from people's phones, which demonstrates that this type of automation can actually work [5].

2.1 Related Work

There are existing platforms which take care of transport management and illustrate the improvement that software can bring in route planning and resource allocation.

Recent studies cover these systems pretty well [6].

We also saw how technology is applied in warehouses by logistics companies. Things such as autonomous robots and image recognition for supply chains are an example of what is possible. But nobody's really adapted this stuff for regular home moves as yet. Lugg does real-time tracking and on-demand moving through a mobile app to demonstrate how technology can make traditional moving much easier. [7].

2.1.1 Example Analysis

Academic research on route optimization and resource planning gave us some ideas on designing the logistics module [8].

A popular method in the field of logistics is the use of barcodes used for package identification. Barcodes are perfectly sufficient to keep track of shipments and keep things running smoothly. The problem is you need people who are constantly looking and checking all the time manually.

The following formula is a popular method to estimate transportation costs:

$$C_t = \sum_{i=1}^n (w_i \cdot d_i) + F$$

Where:

- C_t : Total transportation cost
- w_i : Weight of item i
- d_i : Distance traveled for item i
- F : Fixed logistical overhead

Bar codes and cost formula are helpful, but they don't offer you realtime tracking and the possibility to validate items via images. That's where **Shifters** comes in with object recognition that actually can see what you're moving.

2.2 Comparative analysis

The table below shows what existing solutions are missing: Looking at the table,

Table 2.1: Comparison of Existing Solutions with Shiffters

Features	Wave Associates	Godo Movers	Shiffters (Proposed)
Inventory Tracking	Manual	Barcode-based	AI-driven object detection
Real-Time Cost Estimation	Absent	Limited	Fully automated
Communication	Email/Phone	General updates	Real-time notifications
Transparency	Low	Moderate	High with live tracking

you can see where Shiffters is beating the competition. Better Accuracy with object detection, instant notification and tech that actually works in the real time.

Chapter 3

Requirements Specifications

3.1 Proposed System

Shiffters is a moving app that we created to fix the mess in moving industry of Pakistan. The biggest idea is linking customers and movers together, while providing a solution to the problems on moving that is such a headache. Inaccurate inventories, surprise costs, poor communication, chaotic job assignments - we're solving all of it with technology that actually works.

The fundamental feature is the object recognition by YOLO. Instead of typing everything you are moving on a keyboard, you simply scan things with your phone camera. The system identifies them, creates a digital inventory automatically and you are able to verify all of this when it arrives. N-i-the-batcher No more arguments about what wasn't there or what got damaged.

Up to date news is provided in real-time to keep everyone informed. We use Google Maps and WebSocket communication to let you know what is happening - Exactly when a mover accepted your job, where the truck is, what is being loaded or unloaded, and payment confirmations. You're not left guessing.

The interface works when three types of users. Customers are able to book services, and keep track of everything and make easy changes. Drivers get definite task assign-

ment and job details so they can handle their timetable better. Moderators ensure the smooth running of operations and roles are assigned where they are required.

Price estimation occurs in real-time depending on what you are scanning. No surprise charges later. We have integrated Stripe so that we can accept your payments safely, securely, downloadable receipts for your records. There's also a rating system to ensure the service quality as your customers can review movers, which keeps up the quality of the service.

3.2 Existing System

Yeah there are some apps out there of moving and of being relocated. But honestly? Most of them are pretty basic. They don't have object recognition, real-time tracking is not great, and you never really know what you're gonna pay till the end. A lot of them services, they still do things manually, so that just creates more problems - miscommunication, delays, customers getting annoyed. Like traditional movers don't even bother to track your stuff and plan efficient routes. That's why we built Shifters - to actually solve these problems with technology that's going to work.

3.3 Functional Requirements

Here's what needs to be done with the system. These are the main features that we gathered from talking to users:

1. **Login and registration of users:** Users can register and log into the applications in a secure way. Multi-factor authentication was added for additional security.
2. **Cataloging Items:** Users can scan anything using their phone's camera and the object detection model automatically adds it to the user's inventory.
3. **Item details:** Each item indicates a category, a size and how it is to be handled during the move.

4. **Real-time tracking:** You can see where your stuff would be during the move, such as arrival times and if there are any delays.
5. **Cost estimation:** The app determines how much your move will cost according to what you're moving, the distance and what you need.
6. **Classification model to verify cnic** When drivers upload their CNIC, the model verifies if it is a real Pakistani ID card or not.

3.4 Non-Functional Requirements

1. **Availability:** It's a mobile app that is available whenever you need it. We're aiming for 99.9. It's a mobile app that will work whenever you need it to. We're looking for 99.9 % uptime hence pretty much always accessible.% uptime hence pretty much always accessible.
2. **Usability:** The interface is simple to use and works well on different devices - phones, tablets, whatever.
3. **Performance:** The appiously does multiple moving requests at once without slowing down. Object recognition and calculation of prices are fast.
4. **Reliability:** When something happens in the app, the database is updated immediately so that everything is in sync.
5. **Scalability:** As more and more people using the app more, it can handle the growth without getting slower or crashing.
6. **Security:** Your data is encrypted when it's in transit when it's being sent and when it's stored.
7. **Modifiability:** It was built in such a way that we may easily modify it to incorporate new features or to update things in the future without breaking things.

3.5 Use Case Diagram

This diagram shows who uses the Shifters system - Customers Drivers and Admins. You can see the main things people do such as signing up, logging in, requesting moves and tracking everything. Basically it's a visual of how everybody interacts with the app.

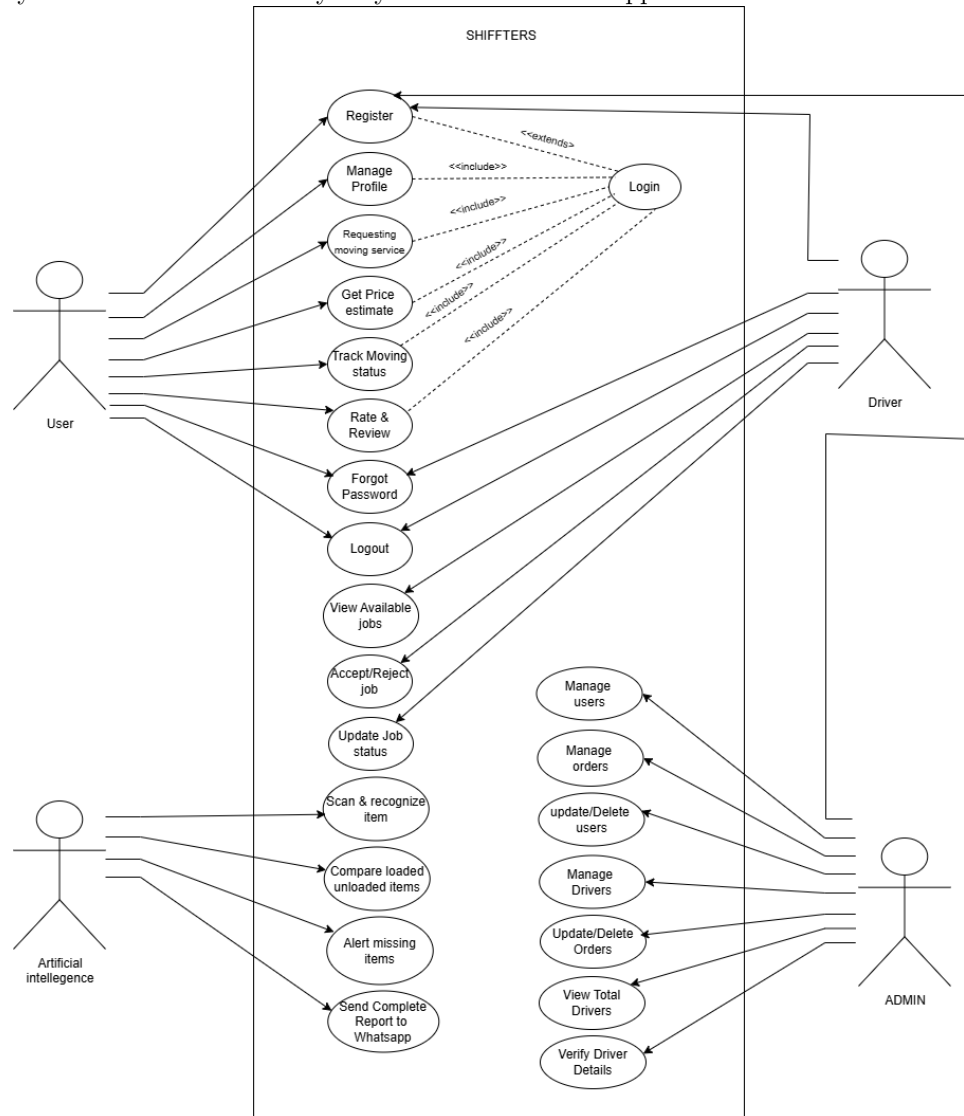


Figure 3.1: Use Case Diagram for Shifters

3.5.1 Register

This displays how new users register for Shifters. Customers and Drivers enter their info, we check the info and store it in the database. Pretty standard signup flow to make sure we are letting in real users only.

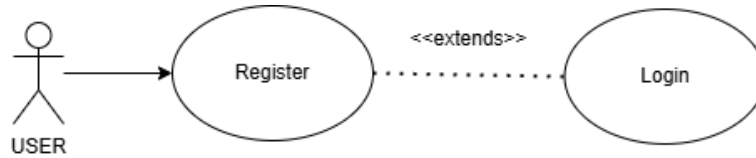


Figure 3.2: Register Use Case

Table 3.1: Use Case Description for Register

Use Case ID	UC01
Title	Register
Description	Customer signs up for an account on the Shifters mobile application as well as access to moving services.
Actor	Customer
Pre-Condition	Register screen of Shifters mobile application is loaded.
Basic Flow	Customer opens the registration screen. Fills in details for e-mail, passwords, etc. Submits the form. System Does a Check if Everything's good. Account gets created. Customer is taken to login page.
Alternate Flow	If email's already taken, we ask them to use some other one. If something's missing in wrong, We tell them to fix it
Post-Condition	The customer creates a new account successfully and gets access to the features of the Shifters application.

3.5.2 Login

Shows how login works. Users type their username and password We consult it in the database If it matches, they get in. If not, error message. Simple security check.



Figure 3.3: Login Use Case

Table 3.2: Use Case Description for Login

Use Case ID	UC02
Title	Login
Description	The user can log in to their account with security in order to access the features and services of the application.
Actor	User
Pre-Condition	The user must have a valid account with a set of credentials for login.
Basic Flow	1. User opens the app, incurs to login. 2. Provides email/phone and password 3. We check if it's correct. 4. If yes, they're in.
Alternate Flow	1. Wrong password or email. 2. We tell them that this is wrong and ask them to have another go.
Post-Condition	The user is successfully logged into his or her account and his or her session is initiated.

3.5.3 Forgot Password

Password recovery flow. User enters their email or their phone, we send that user a link or code to reset their password and user will set up his or her password. Pretty straightforward.



Figure 3.4: Forgot Password Use Case

Table 3.3: Use Case Description for Forgot Password

Use Case ID	UC03
Title	Forgot Password
Description	User can reset password through secure recovery process.
Actor	User
Pre-Condition	User needs active account with access to registered email or phone.
Basic Flow	1. User clicks "Forgot Password." 2. Enters email address. 3. System sends reset link or code. 4. User clicks link or enters code and sets new password. 5. Password reset completed.
Alternate Flow	1. Email or phone not in system. 2. System prompts for valid information.
Post-Condition	User password is successfully changed and can log in with new password.

3.5.4 Manage Profile

Smaller elements on the right side of the page that display how users edit their profiles. They can change their name, contact info, address and upload a photo. We validate the changes and committing them.

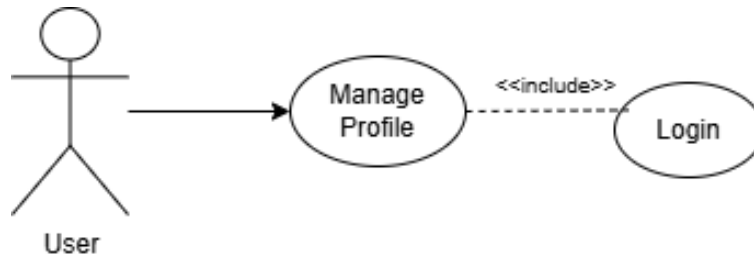


Figure 3.5: Manage Profile Use Case

Table 3.4: Use Case Description for Manage Profile

Use Case ID	UC04
Title	Manage Profile
Description	The user can update his or her information on its profile such as name, contact details, and password.
Actor	User
Pre-Condition	The user have to be logged in the application.
Basic Flow	1. User goes to "Manage Profile." 2. Current info shows up. 3. They do not only edit what they want but save it.
Alternate Flow	1. Required field is left blank. 2. We ask them to do a fill in first before saving it.
Post-Condition	The updated information of profile is saved and reflected in the application.

3.5.5 Requesting Moving Service

How booking a move works. Customer enter pickup and drop-off locations plus are what they're moving. In the end we calculate the price and send the request to drivers. When a driver accepts customer gets confirmation.

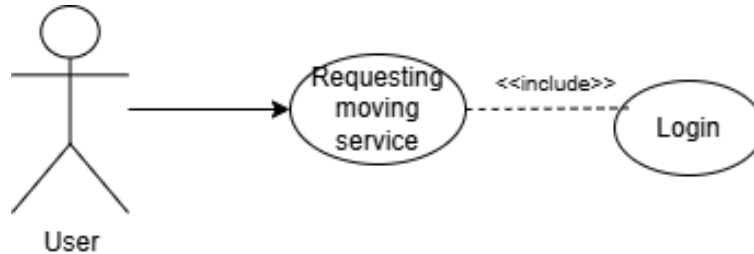


Figure 3.6: Requesting Moving Service Use Case

Table 3.5: Use Case Description for Requesting Moving Service

Use Case ID	UC05
Title	Requesting Moving Service
Description	The user can order a moving service through providing necessary details of pickup location and drop-off location, item description, and date of preference.
Actor	User
Pre-Condition	The user must be logged into the application.
Basic Flow	1. User clicks on "Request Moving Service." 2. Form pops up. 3. They fill it out and submit. 4. We prove and demonstrate the estimated price.
Alternate Flow	1. Something's missing or wrong in the form. 2. We make a few requests of them to fix it before they submit.
Post-Condition	The moving service request is successfully submitted and confirmation and tracking details are sent to the user.

3.5.6 Get Price Estimate

Getting a price quote. Customer enters where they're moving from, to, what they're moving. We calculate the cost per distance and weight of the items.

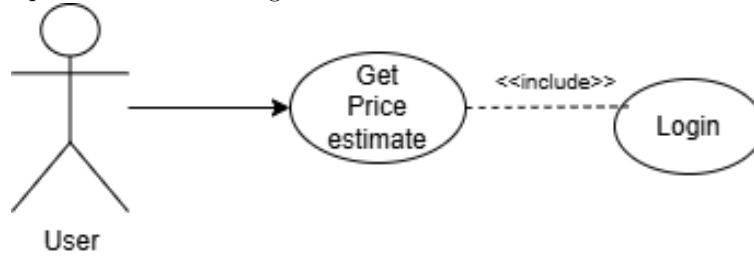


Figure 3.7: Get Price Estimate Use Case

Table 3.6: Use Case Description for Get Price Estimate

Use Case ID	UC06
Title	Get Price Estimate
Description	The user can get a priceestimate for a service in motion based upon the details given by the user like the locations and item descriptions.
Actor	User
Pre-Condition	The user will have to be logged into the application and basic moving details should have been filled in.
Basic Flow	1. User clicks on 'Get Price Estimate.' 2. We request pickup, drop-off, and item information from them. 3. We play the numbers and provide the estimation.
Alternate Flow	1. User refuses to provide some info that is required. 2. We get them to fill in what they can, the things that are missing for them first.
Post-Condition	The user receives price estimation of the moving service based on the provided details.

3.5.7 Track Moving Status

Real time tracking for your move. You can see when the driver accepts and picks up your stuff and delivers it. Status updates throughout.

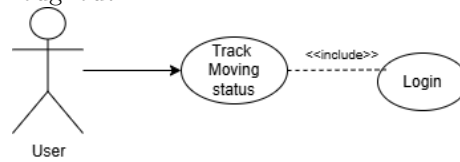


Figure 3.8: Track Moving Status Use Case

Table 3.7: Use Case Description for Track Moving Status

Use Case ID	UC07
Title	Track Moving Status
Description	The user can track the status of his or her requested moving service through real-time tracking.
Actor	User
Pre-Condition	The user must have an active moving request and be logged onto the application.
Basic Flow	1. User clicks on "Track Moving Status." 2. We demonstrate where the driver is, what stage is the move at, estimated arrival time.
Alternate Flow	1. Something goes wrong with the loading of the status. 2. We tell them there is an error and they have to try again.
Post-Condition	The user successfully views the real-time status of their moving service.

3.5.8 Logout

Logging out. User clicks log out we destroy their session and redirect them to the login screen. Basic security stuff.

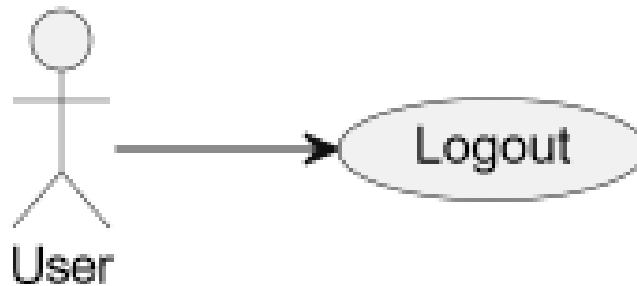


Figure 3.9: Logout Use Case

Table 3.8: Use Case Description for Logout

Use Case ID	UC09
Title	Logout
Description	The user can safely log out of his/her account from the application.
Actor	User
Pre-Condition	The user must be logged into the application.
Basic Flow	1. User is taken to profile or settings. 2. Clicks "Logout." 3. We we log them out and we take them to the login screen.
Alternate Flow	1. Network issue at the time of logging out 2. At one point when connection's back, we tell them and retry.
Post-Condition	The user is successfully logged out and his/her session is terminated.

3.5.9 Driver Register

Driver signup process. They enter their info, and upload their required documents such as license and vehicle registration. Shows what will happen if something goes wrong too.

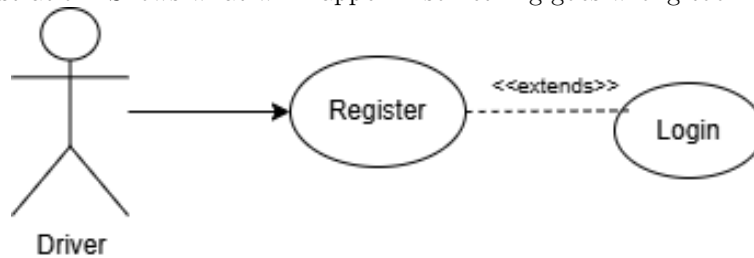


Figure 3.10: Driver Register Use Case

Table 3.9: Use Case Description for Driver Register

Use Case ID	UC10
Title	Register
Description	The driver signs up using his personal information, his contact details and vehicle information.
Actor	Driver
Pre-Condition	Valid email and necessary documents are required.
Basic Flow	1. Driver clicks "Register." 2. Fills out the form, uploads docs 3. We check everything and we make their account. 4. They get a confirmation.
Alternate Flow	1. There is something wrong with what they entered, we ask them to fix it. 2. Upload won't allow them, they can complete it later on.
Post-Condition	Driver account is created and ready for use.

3.5.10 Driver View Available Jobs

Drivers search for available jobs After logging in, they are presented with a list of moving requests based on location, vehicle type and availability. They pick what works for them.

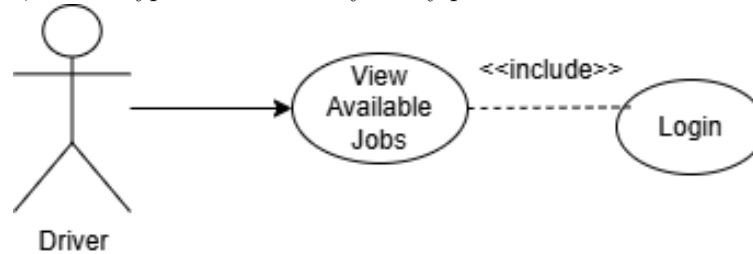


Figure 3.11: Driver View Available Jobs Use Case

Table 3.10: Use Case Description for View Available Jobs

Use Case ID	UC11
Title	View Available Jobs
Description	The driver can see a list of jobs that are available based on his profile and preferences in the application.
Actor	Driver
Pre-Condition	The driver: the driver needs to be logged in to the application with a valid profile, with relevant preferences.
Basic Flow	1. Driver logs in. 2. Goes to "Available Jobs." 3. We show jobs in their vehicle with their location.
Alternate Flow	1. No jobs available right now. 2. We say to them to check back later on.
Post-Condition	The driver is able to see a list of the jobs that are available and waiting for him to choose from or perform additional actions.

3.5.11 Driver Update Job Status

How drivers 2 job status. After accepting a job, they mark as 'Accepted', 'Picked Up', 'In Transit' or 'Completed'. We take it down and inform the customer.

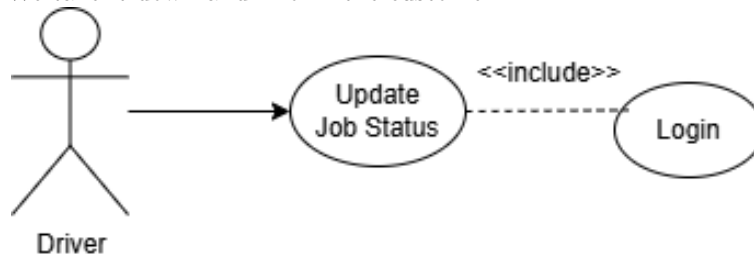


Figure 3.12: Add Products to Cart Use Case

Table 3.11: Use Case Description for Update Job Status

Use Case ID	UC12
Title	Update Job Status
Description	The driver updates the status of a job as "In Progress," "Completed" or "Canceled" to know this and to users of the system.
Actor	Driver
Pre-Condition	Driver is logged in and has an assigned job.
Basic Flow	1. Driver opens the job details. 2. Picks a new status. 3. We update it and tell all of those who are involved.
Alternate Flow	1. Update will not succeed due to network problems. 2. We retry when internet's back.
Post-Condition	Job status is updated and visible to all parties.

3.5.12 Driver Accept/Reject Job

Drivers deciding on jobs. They see the details - Where to pickup, drop off, distance, payment. They have a choice of accepting it or rejecting it. If they reject we send it to another driver.

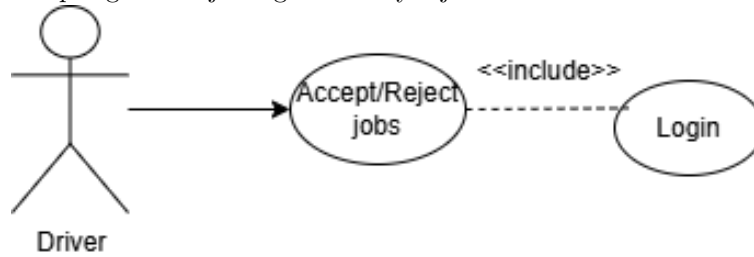


Figure 3.13: Driver Accept/Reject Job Use Case

Table 3.12: Use Case Description for Accept/Reject Job

Use Case ID	UC13
Title	Accept/Reject Job
Description	The driver can accept or reject job offers presented to him/her depending on his/her availability and preferences.
Actor	Driver
Pre-Condition	The driver should be logged into the application and viewed the available job details.
Basic Flow	1. Driver examines the details of the job. 2. Clicks "Accept" or "Reject." 3. If it is accepted, we give it to them. 4. If rejected, we take them off their list.
Alternate Flow	1. They try to accept but someone else already took it. 2. We tell them the list has gone and renew the list.
Post-Condition	The job is either accepted, and assigned to the driver, or rejected, and noted to be unavailable to the driver.

3.5.13 AI Scan and Recognize Items

Inventory detection of an item User scans items using phone camera, YOLO recognizes what they're and we have a classic inventory built automatically.

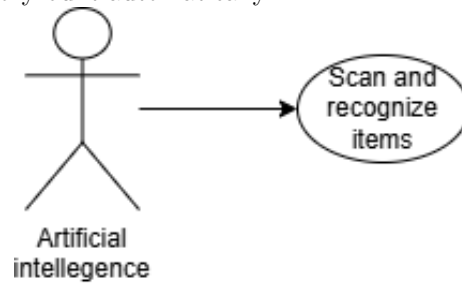


Figure 3.14: AI Scan and Recognize Items Use Case

Table 3.13: Use Case Description for Scan and Recognize Items

Use Case ID	UC14
Title	Scan and Recognize Items
Description	The AI system scans items to identify and recognize each item so that it can be accurately tracked and its inventory can be managed.
Actor	Artificial Intelligence
Pre-Condition	Scanning technology must be available and items must be prepared and arranged for scanning.
Basic Flow	1. Camera turns on. 2. User scans items. 3. What YOLO does is identify what they are. 4. We save them in the database.
Alternate Flow	1. Can't recognize something. 2. We flag it and we ask user to manually enter the details
Post-Condition	All of the recognized items are recorded in the system and a complete inventory list is updated.

3.5.14 AI Compare Loaded Unloaded Items

Verification on loading and unloading. Users take pictures of things at both ends. YOLO compares them to ensure that nothing has been lost during the move.

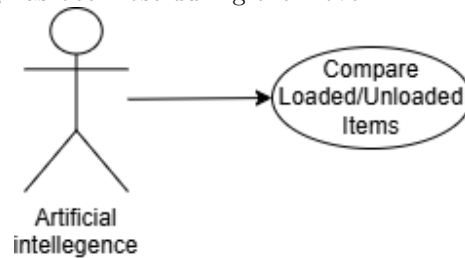


Figure 3.15: AI Compare Loaded Unloaded Items Use Case

Table 3.14: Use Case Description for Compare Loaded and Unloaded Items

Use Case ID	UC15
Title	Compare Loaded and Unloaded Items
Description	The list of items loaded and unloaded is compared with the AI system to ensure that all the items are accounted for in a move.
Actor	Artificial Intelligence
Pre-Condition	Loaded and unloaded items must be scanned and recorded in the system.
Basic Flow	1. We go ahead and recall the list of loaded items. 2. We get out list of unloaded items. 3. YOLO compares both. 4. Flags anything missing and/or extra
Alternate Flow	1. Loaded and unloaded of data's incomplete. 2. We tell our user and wait until we get it from them.
Post-Condition	Any discrepancies between the loads and the unloads, if any, are identified and reported to concerned people.

3.5.15 AI Alert Missing Items

Alert system for stuff that is missing After comparing loaded vs unloaded inventory, when something is off, automatically we send an alert to everybody's knowing about what's missing..

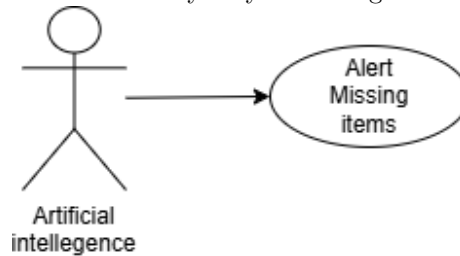


Figure 3.16: AI Alert Missing Items Use Case

Table 3.15: Use Case Description for Alert Missing Items

Use Case ID	UC16
Title	Alert Missing Items
Description	There the AI system detects the missing items at the moment of move and creates an alert that informs the concerned individuals to take necessary action right away.
Actor	Artificial Intelligence
Pre-Condition	The comparison on items loaded and unclear must have been carried out and the discrepancies to be detected.
Basic Flow	1. We see what is missing from the comparison. 2. Create an alert with details. 3. Send it to the customer/dispatcher. 4. Log it for records.
Alternate Flow	1. Is not able to send the alert due to connection error. 2. We retry in a bit. If it continues failing, admin gets notified
Post-Condition	Relevant parties are informed about the missing pieces and corresponding updates occur in the system.

3.5.16 View Total Drivers Available

Admin dashboard showing the number of drivers online and ready to take jobs now.

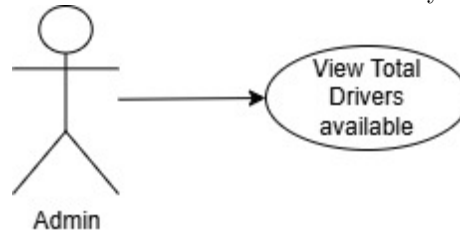


Figure 3.17: View Total Drivers Available Use Case

Table 3.16: Use Case Description for View Total Drivers Available

Use Case ID	UC18
Title	View Total Drivers Available
Description	Admin is viewed total number of the drivers available in the system in
Actor	Admin
Pre-Condition	Admin must be logged into the system.
Basic Flow	1. Admin logs in. 2. Goes to "Drivers" section. 3. Performs checks on total available drivers. 4. We show the count.
Alternate Flow	1. Can't load the data. 2. Error message pops up, admin is able to retry or contact support.
Post-Condition	The admin gets successful to get to see the total number of drivers available in the system.

3.5.17 Update/Delete Orders

Admin has the ability to edit or delete orders. Good for making corrections or taking care of cancellations.

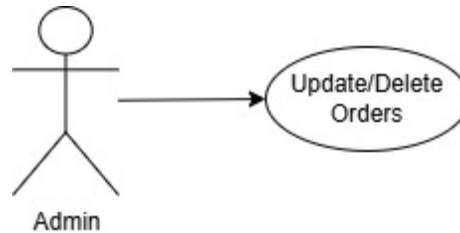


Figure 3.18: Update/Delete Orders Use Case

Table 3.17: Use Case Description for Update/Delete Orders

Use Case ID	UC19
Title	Update/Delete Orders
Description	Admin updates or deletes already existing orders in the system.
Actor	Admin
Pre-Condition	Admin must be logged into the system and have the necessary permissions.
Basic Flow	1. Admin logs in. 2. Opens "Orders" section. 3. Picks order to change or to deletion. 4. Makes the changes or getting rid of it. 5. We process it and confirm.
Alternate Flow	1. Order doesn't exist. 2. Error happens We show a message and they are able to retry
Post-Condition	The order is successfully changed or deleted, and reflected by the system.

3.5.18 Update/Delete Users

Admin facility to manage our user accounts. They are able to update info or delete accounts if necessary.

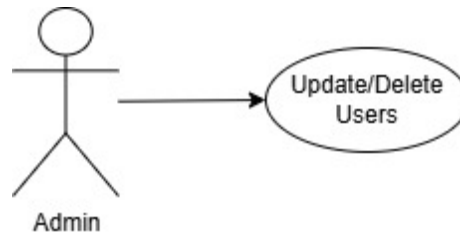


Figure 3.19: Update/Delete Users Use Case

Table 3.18: Use Case Description for Update/Delete Users

Use Case ID	UC20
Title	Update/Delete Users
Description	Admin add remove user accounts from the system.
Actor	Admin
Pre-Condition	Admin needs to be logged in to the system with required permissions.
Basic Flow	1. Admin logs in. 2. Goes to "Users" section. 3. Picks a user to edit or delete. 4. Makes changes/to confirms deletion. 5. We take certain action with it and provide us confirmation.
Alternate Flow	1. User account doesn't exist. 2. Error occurs and message pops up and can retry message.
Post-Condition	The user account is successfully updated/ deleted and the changes are reflected in the system.

3.5.19 Manage Users

User management screen where admins may view all users, edit user information, or remove them from the platform.

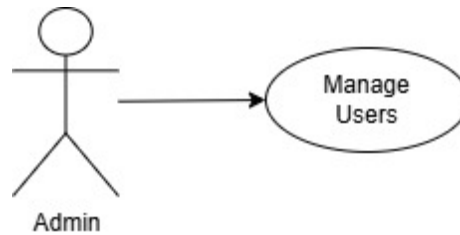


Figure 3.20: Manage Users Use Case

Table 3.19: Use Case Description for Manage Users

Use Case ID	UC21
Title	Manage Users
Description	Admin Controls user accounts (user details can be viewed, updated, or deleted)
Actor	Admin
Pre-Condition	Admin has to be logged in to the system with the necessary permissions.
Basic Flow	1. Admin logs in. 2. Opens "Manage Users." 3. Selection of a user to see, edit or delete. 4. Does whatever action they need. 5. We confirm it went through.
Alternate Flow	1. User doesn't exist. 2. Something goes wrong, having the error message and the possibility to retry it.
Post-Condition	User account is successfully controlled (viewed, updated, or deleted) and the system represents the changes.

3.5.20 Manage Drivers

Driver management user interface. Admins can see driver profiles, change their details or uninstall them from the system.

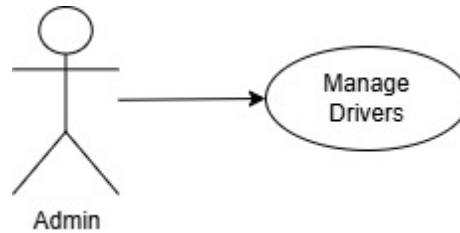


Figure 3.21: Manage Drivers Use Case

Table 3.20: Use Case Description for Manage Drivers

Use Case ID	UC22
Title	Manage Drivers
Description	Admin manages account information for the drivers, such as adding, updating or removing the driver information.
Actor	Admin
Pre-Condition	Admin has to be logged in to the system with the required permissions.
Basic Flow	1. Admin logs in. 2. Opens "Manage Drivers." 3. Choosing a driver to add, update or delete. 4. Does the action - add new driver, change info or remove. 5. Confirmation shows up.
Alternate Flow	1. Driver account doesn't exist. 2. Option of error, we show message and retry option.
Post-Condition	The driver account is successfully managed (added, updated, or deleted), and the system reflects the changes.

3.5.21 Manage Orders

Admin order management dashboard. They can view all orders, update information or cancel them.

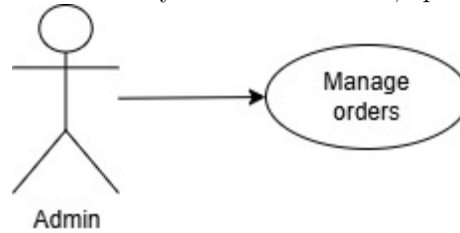


Figure 3.22: Manage Orders Use Case

Table 3.21: Use Case Description for Manage Orders

Use Case ID	UC23
Title	Manage Orders
Description	Admin manages orders, such as view, update, or cancel orders in the system.
Actor	Admin
Pre-Condition	Admin must be logged into the system with the appropriate permissions.
Basic Flow	1. Admin logs in. 2. Opens "Manage Orders." 3. Picks an order to check for, update or to cancel the order. 4. Makes the change they need. 5. We confirm it's done.
Alternate Flow	1. Order doesn't exist. 2. Something blowing, error message with retry option.
Post-Condition	The order is handled successfully (viewed, updated or canceled), and the system takes the changes into account.

3.5.22 Verify Driver Documents

Steps in the verification of documents. admin checks whether license and registration uploaded by the driver are legit before he/she approves to take jobs.

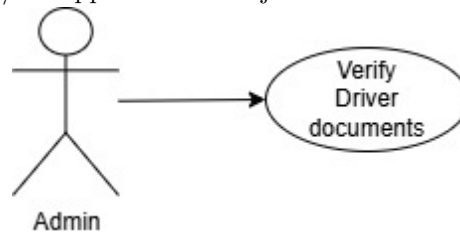


Figure 3.23: Verify Driver Documents Use Case

Table 3.22: Use Case Description for Verify Driver Documents

Use Case ID	UC24
Title	Verify Driver Documents
Description	The uploaded documents of drivers are verified by the admin who ensures they are following and original documents.
Actor	Admin
Pre-Condition	The uploaded documents of drivers are verified by the admin who ensures they are following and original documents.
Basic Flow	1. Admin opens up "Verify Driver Documents." 2. Examines the uploaded docs by a driver. 3. Approves/ rejects based on if they're legit. 4. Driver's status gets updated.
Alternate Flow	1. Docs are not complete or not clear. 2. We reject them, and say to the driver, Upload better
Post-Condition	Driver's documents are checked and their state is changed to system.

Chapter 4

System Design

4.1 System Architecture

We made Shiffters in a modular way so that it can be easily scaled and adapted. There are three main layers:

- **User Interaction Layer:** The mobile and web applications, in which users write down things and trace their movements.
- **Processing Layer:** Runs the object recognition (YOLO), route optimization, and cost estimation algorithms.
- **Data Layer:** user-data, picture and logs storage in the cloud.

4.2 System Architecture Diagram

This diagram shows the three-layer architecture of Shifters - the User Interaction Layer where customers and drivers access the app, the Processing Layer that handles AI recognition and route optimization, and the Data Layer that stores all information in the cloud. All components work together to provide a seamless moving experience.

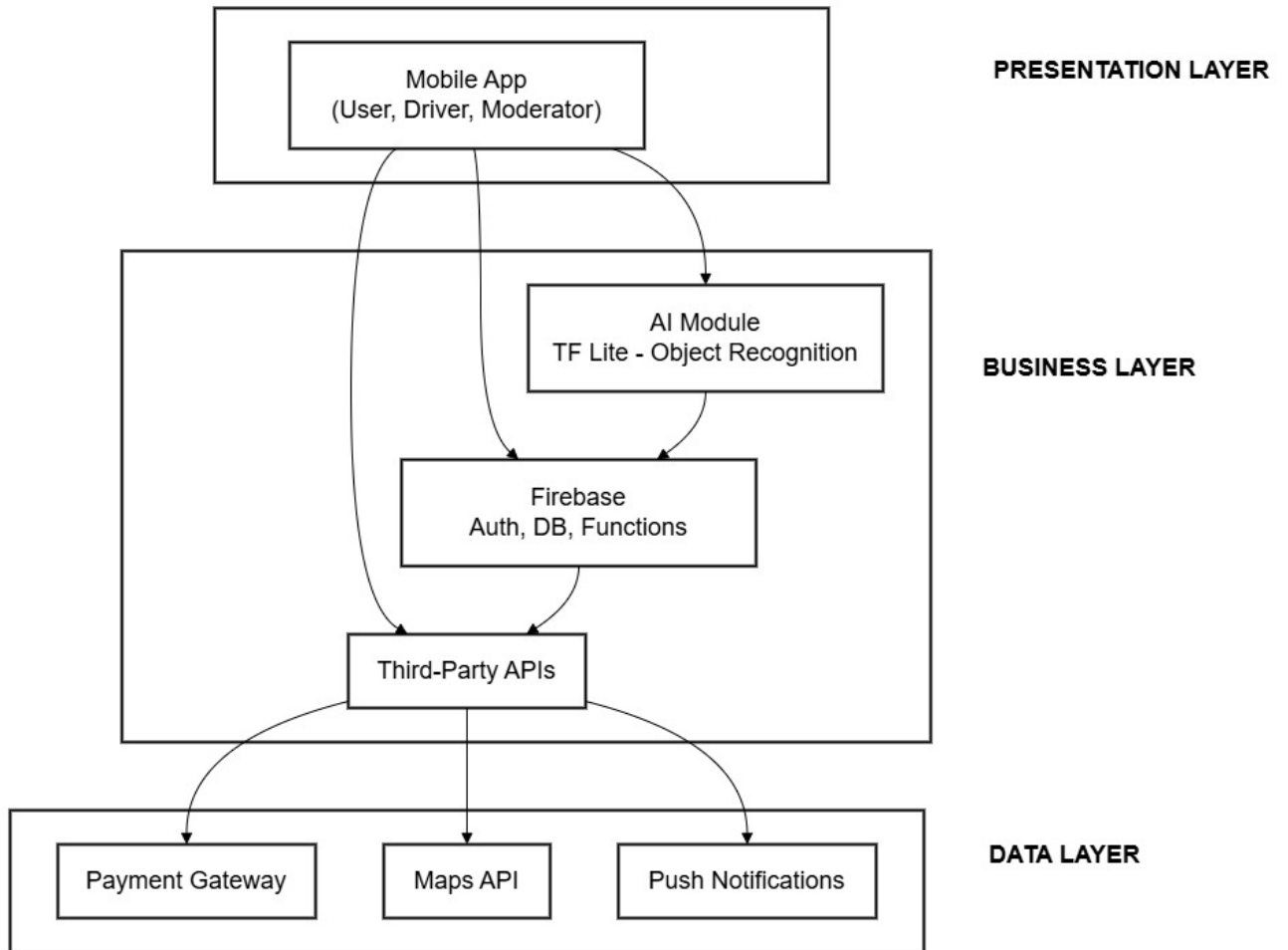


Figure 4.1: System Architecture Diagram

4.3 High Level Design

The design is a link between the user interface and backend processing. Main modules we have:

- **Item Recognition Module:** Uploads pictures and classifies them with the help of YOLO.
- **Logistics Module:** Real time tracking of logistics and route optimization for transport.
- **User Module:** To save user preferences and to create progress reports.

4.3.1 Sequence Diagram

Sequence diagrams illustrate how various parts of the system communicate with each other and in which order. They essentially trace the interactions to the end when a person does something in the application. We have used modular UI with backend services that is pretty standard for web-based moving apps [9].

User Registration

Shows how a new user enters registration details, the system validates them, and creates an account.

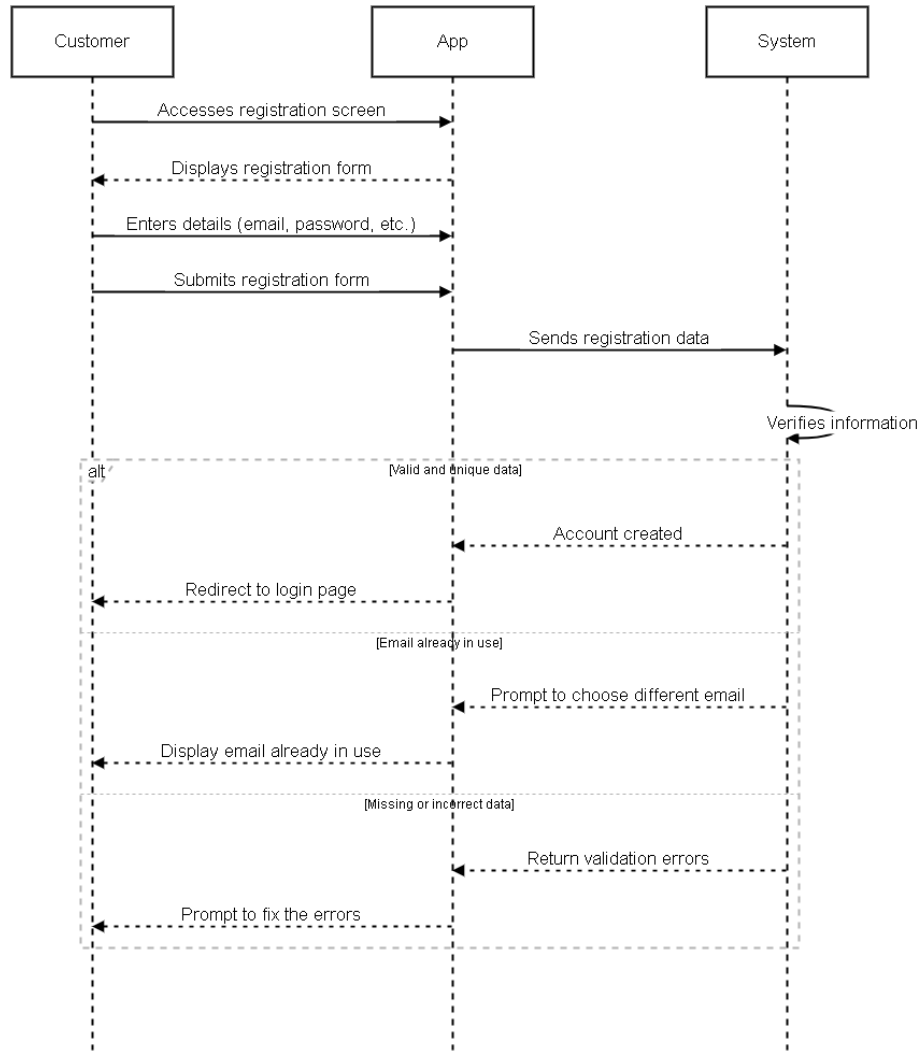


Figure 4.2: Registration Sequence Diagram

User Login

Illustrates how users submit login credentials, which the system verifies before granting access.

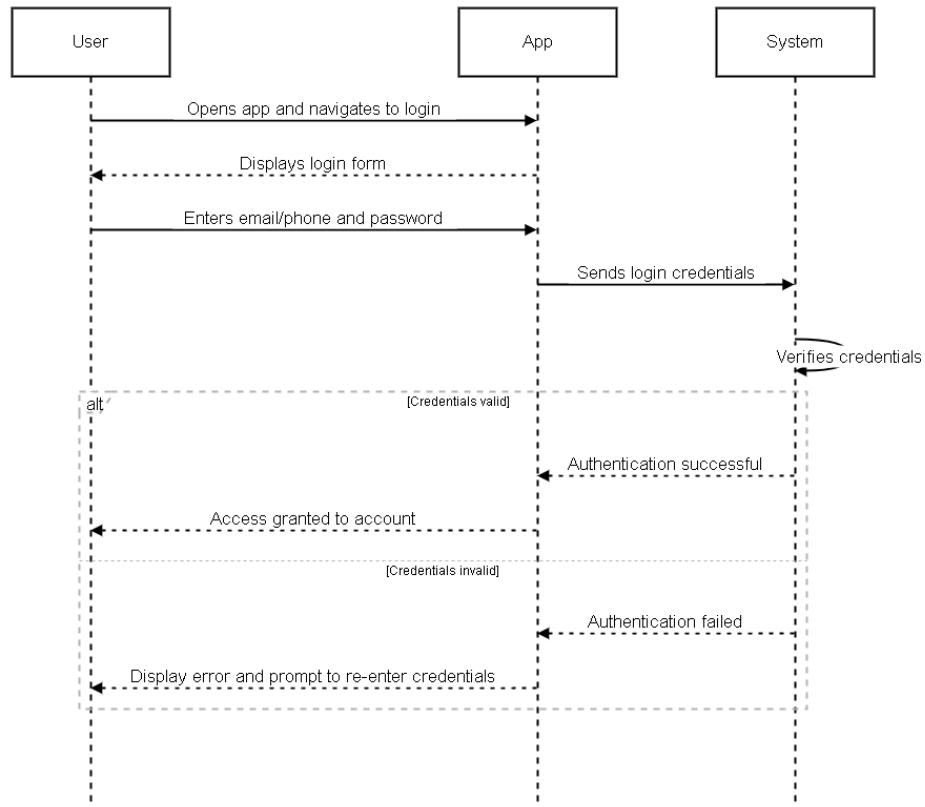


Figure 4.3: User Login Sequence Diagram

Forgot Password

Explains how the user requests a password reset and receives a secure link or code to create a new password.

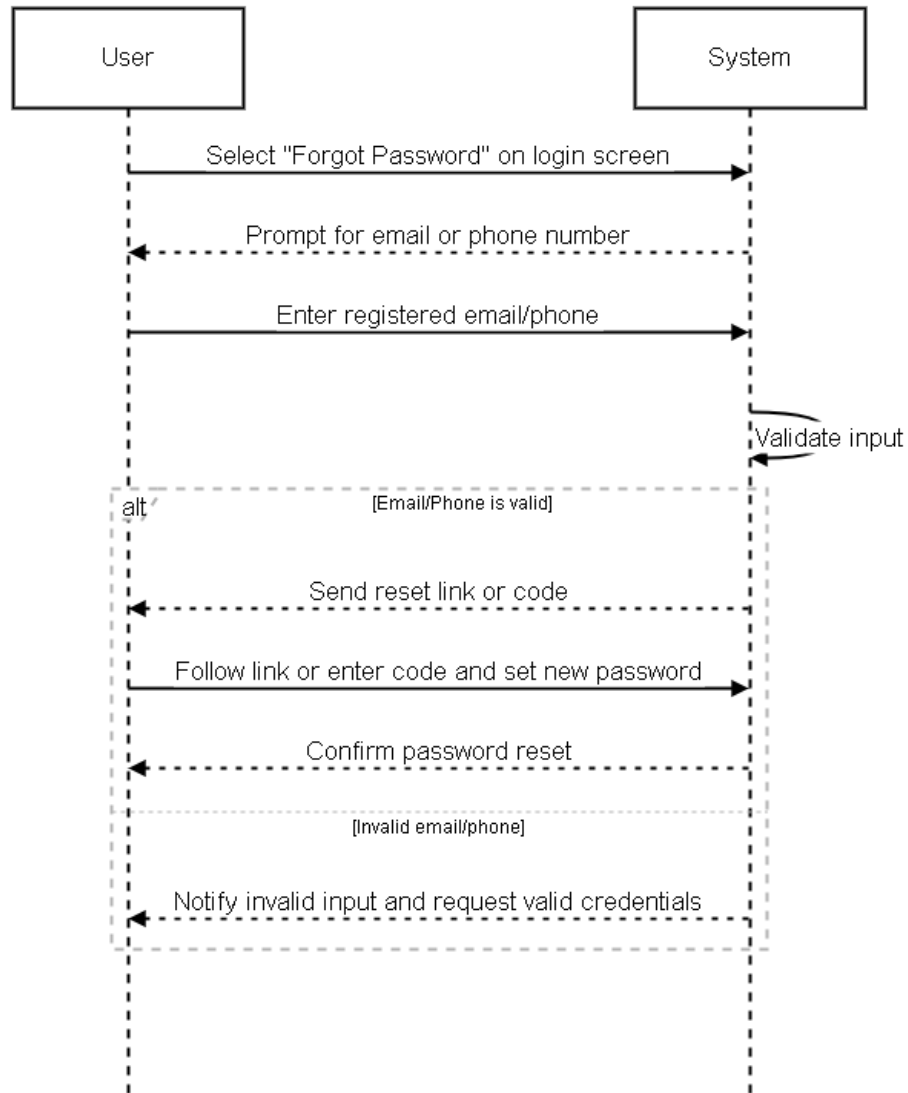


Figure 4.4: Forgot Password Sequence Diagram

AI Scan and Recognize Items

Shows how uploaded images are processed by the AI module to identify, classify, and catalog items.

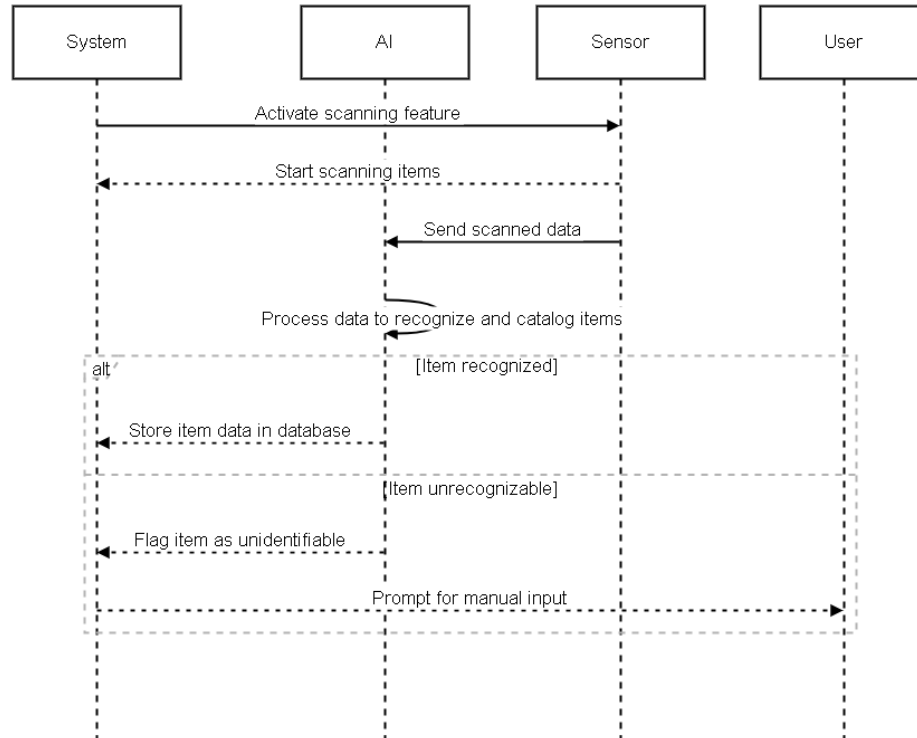


Figure 4.5: AI Scan and Recognize Items Sequence Diagram

AI Compare Loaded and Unloaded Items

Demonstrates how the AI compares pre-move and post-move item lists to detect any differences.

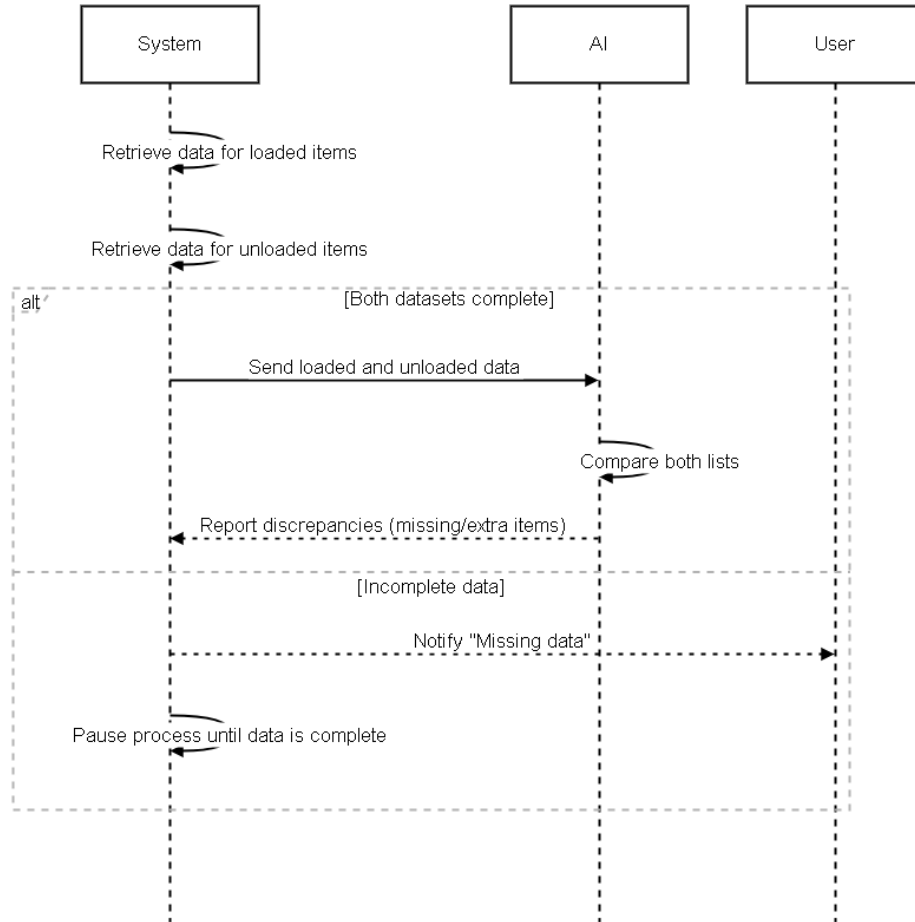


Figure 4.6: AI Compare Loaded and Unloaded Items Sequence Diagram

AI Alert Missing Items

Describes how the system identifies missing items after comparison and automatically generates alerts.

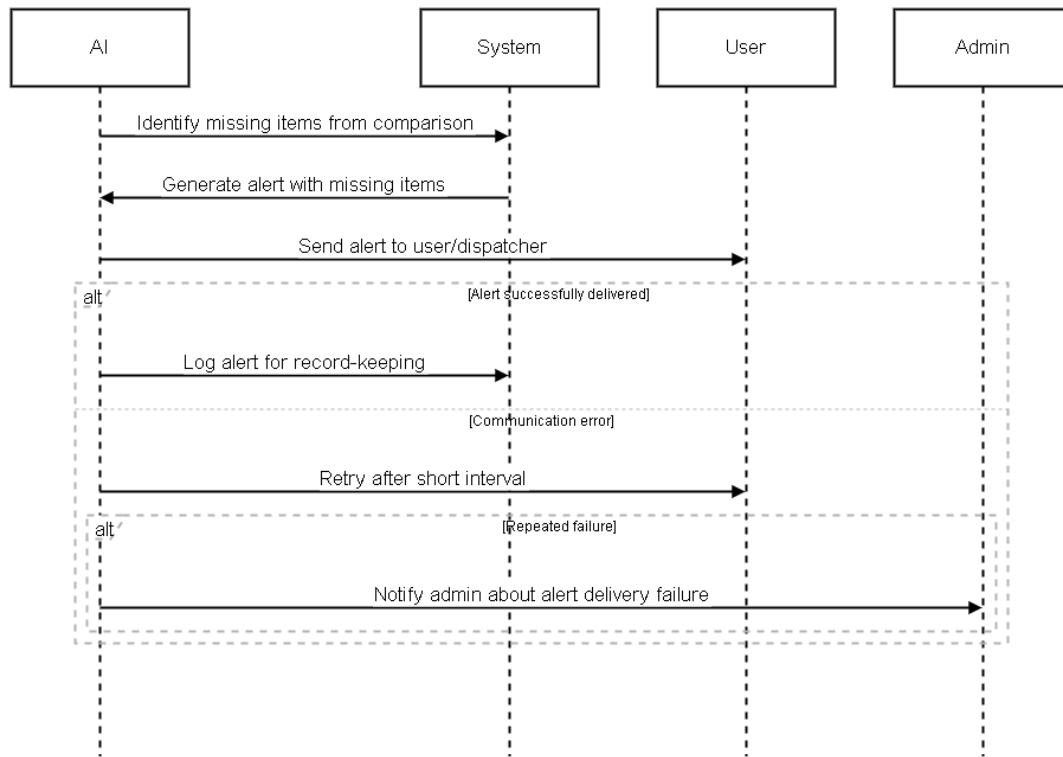


Figure 4.7: AI Alert Missing Items Sequence Diagram

Driver Registration

Explains how a driver provides personal and vehicle information, which the system validates to create an account.

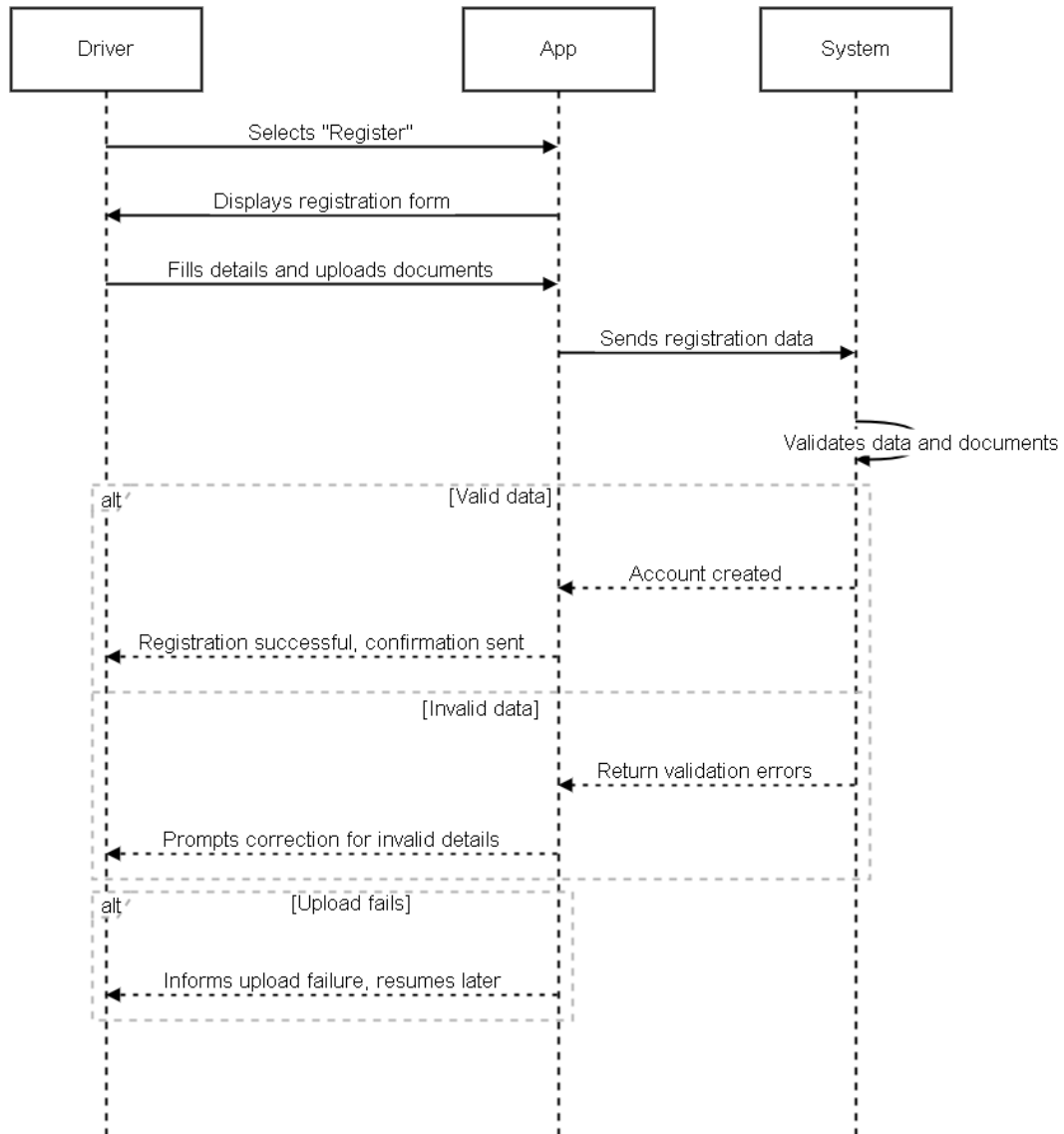


Figure 4.8: Driver Registration Sequence Diagram

Driver Accept / Reject Job

Illustrates how drivers review assigned jobs and respond with acceptance or rejection.

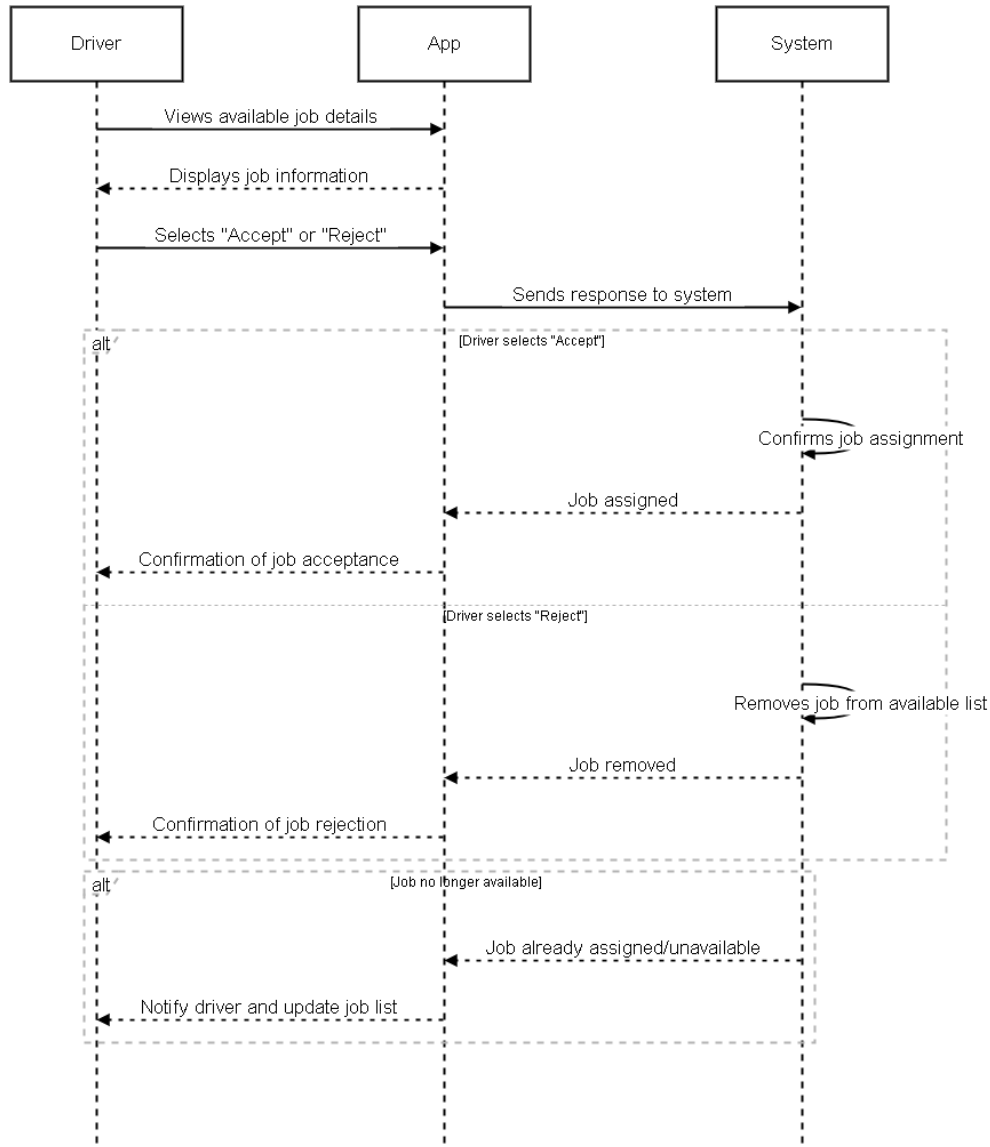


Figure 4.9: Driver Accept / Reject Job Sequence Diagram

Driver Update Job Status

Shows how drivers update job status at each stage, and how the system records and notifies users.

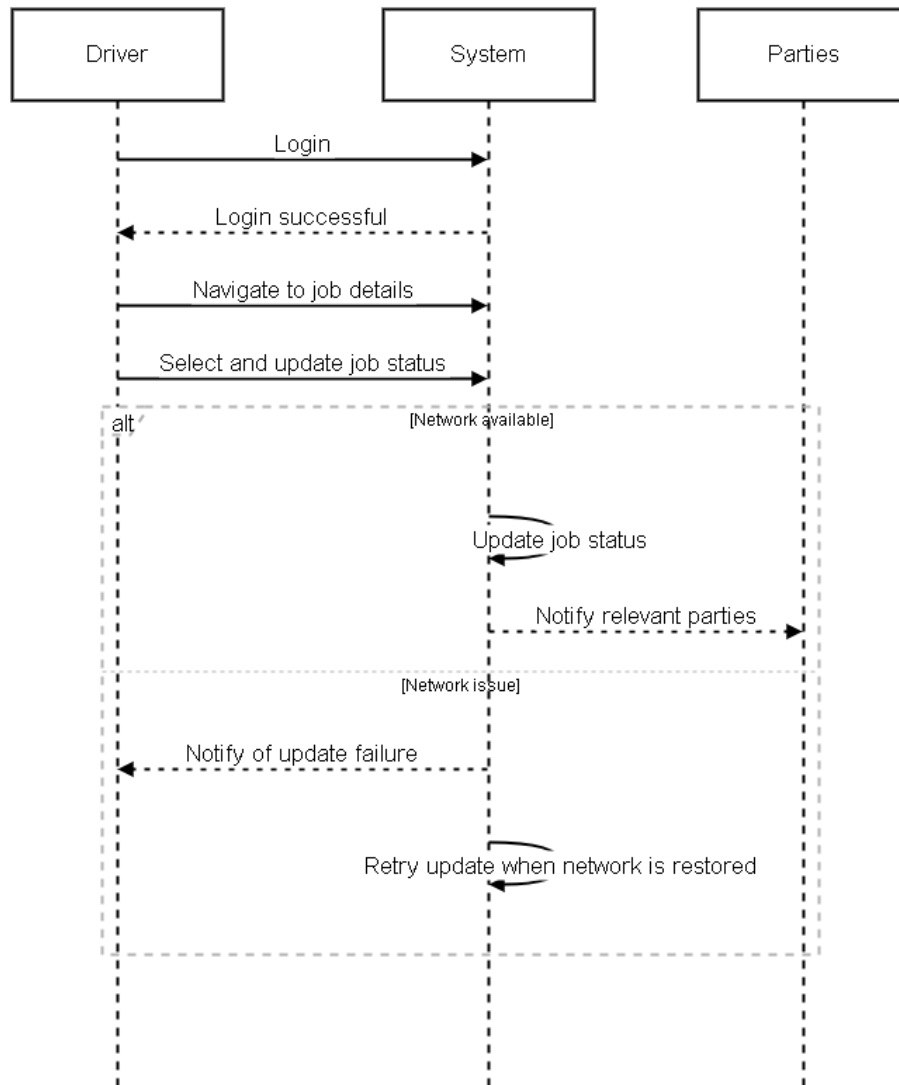


Figure 4.10: Driver Update Job Status Sequence Diagram

Driver View Available Jobs

Describes how the system retrieves and displays available jobs that match the driver's profile.

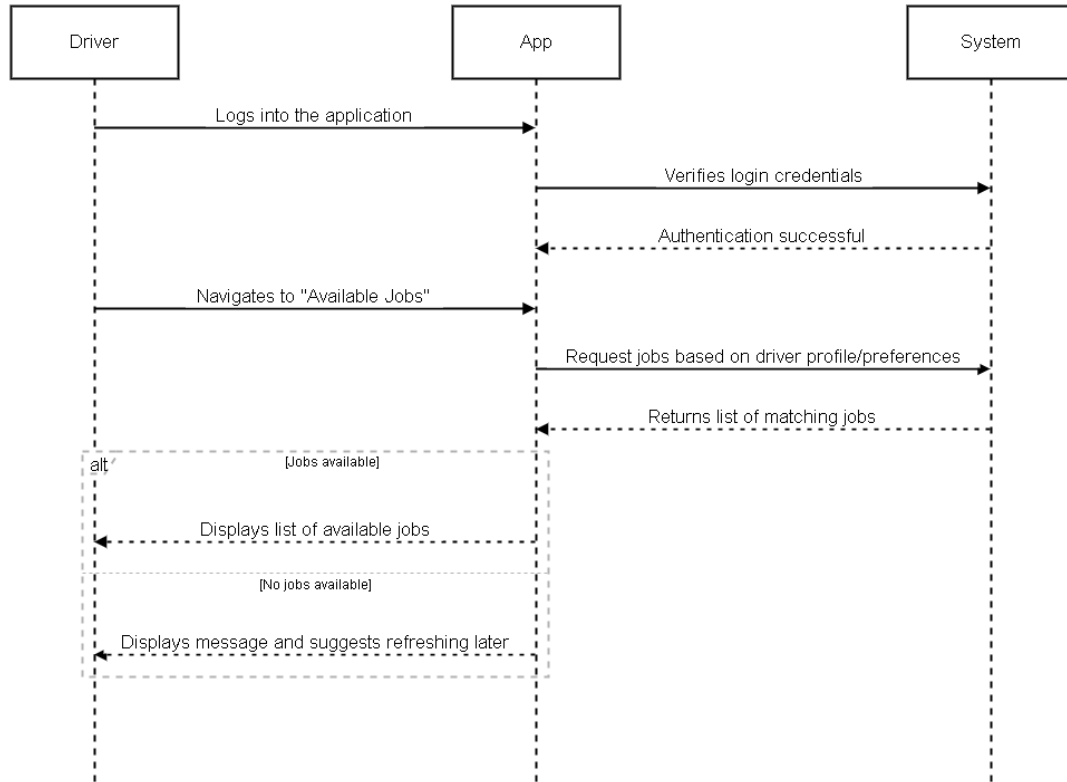


Figure 4.11: Driver View Available Jobs Sequence Diagram

Admin Manage Drivers

Shows how the admin views, edits, or deletes driver accounts in the system.

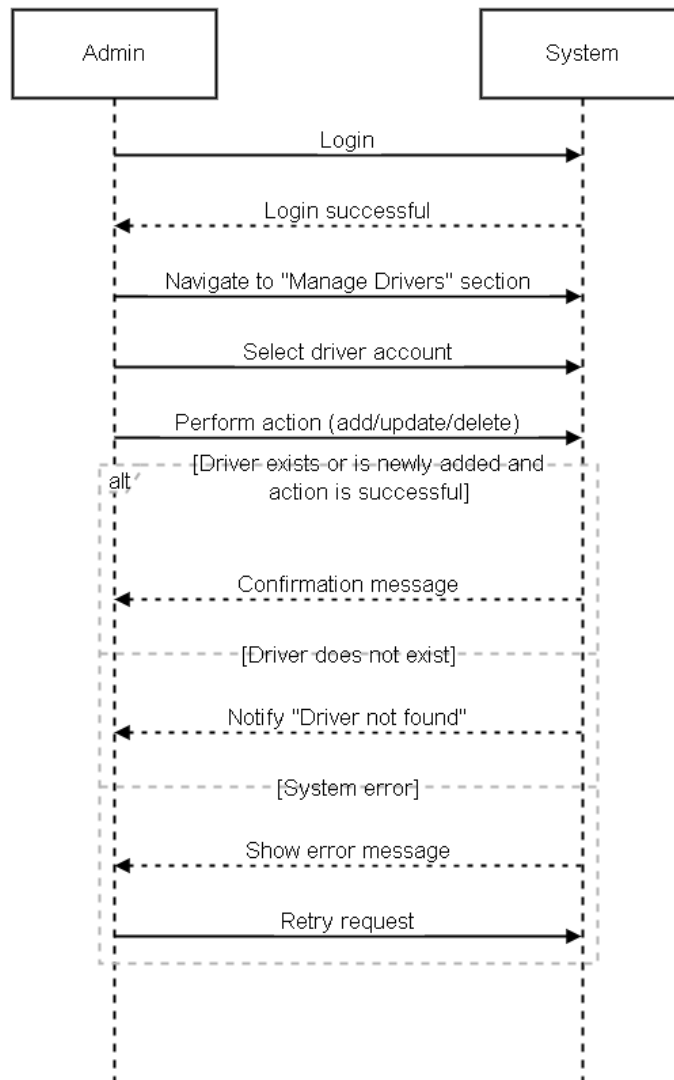


Figure 4.12: Admin Manage Drivers Sequence Diagram

Admin Manage Users

Explains how the admin manages user accounts, including updates and removals.

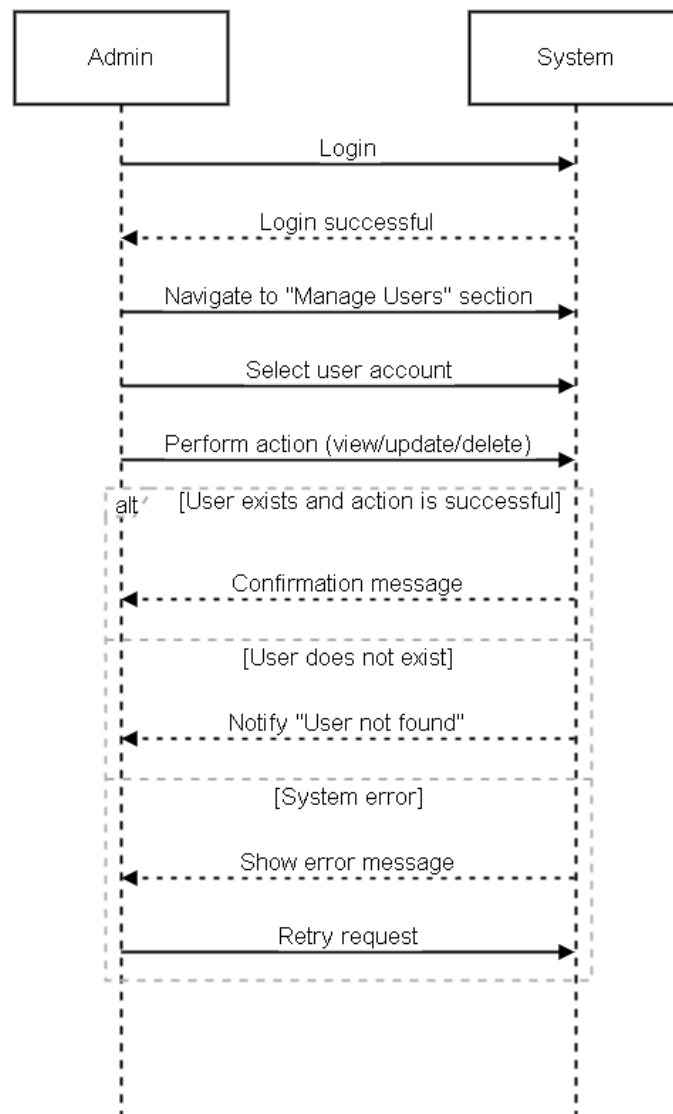


Figure 4.13: Admin Manage Users Sequence Diagram

Admin Manage Orders

Demonstrates how the admin views, updates, or cancels customer orders within the system.

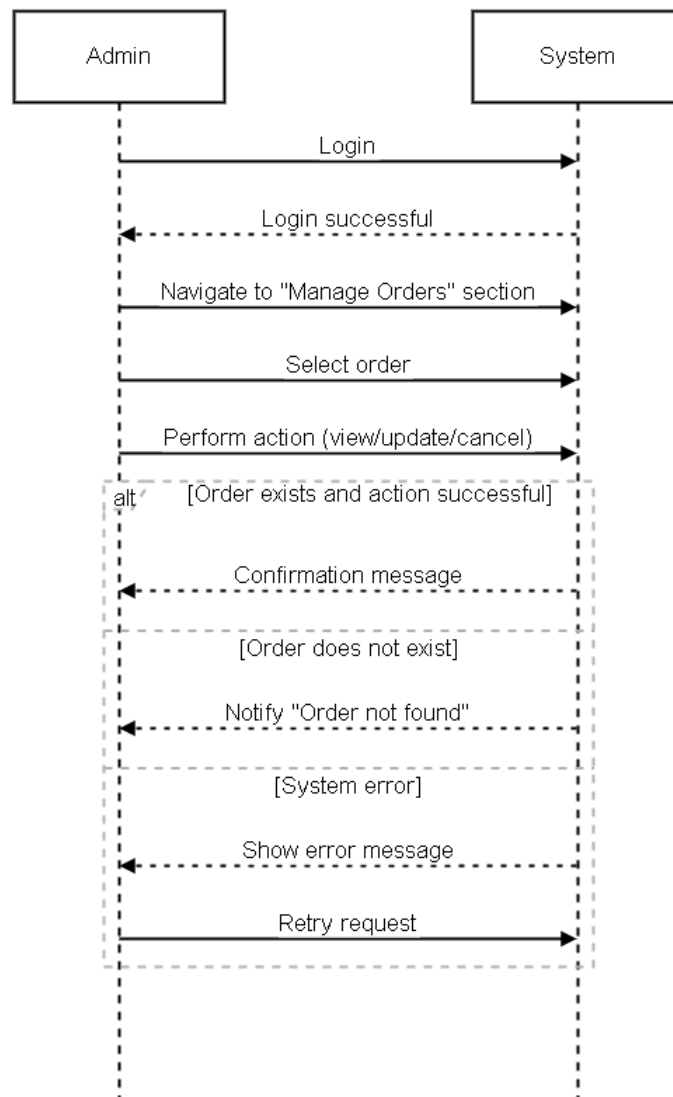


Figure 4.14: Admin Manage Orders Sequence Diagram

Admin Update / Delete Orders

Explains how the admin modifies or removes existing orders.

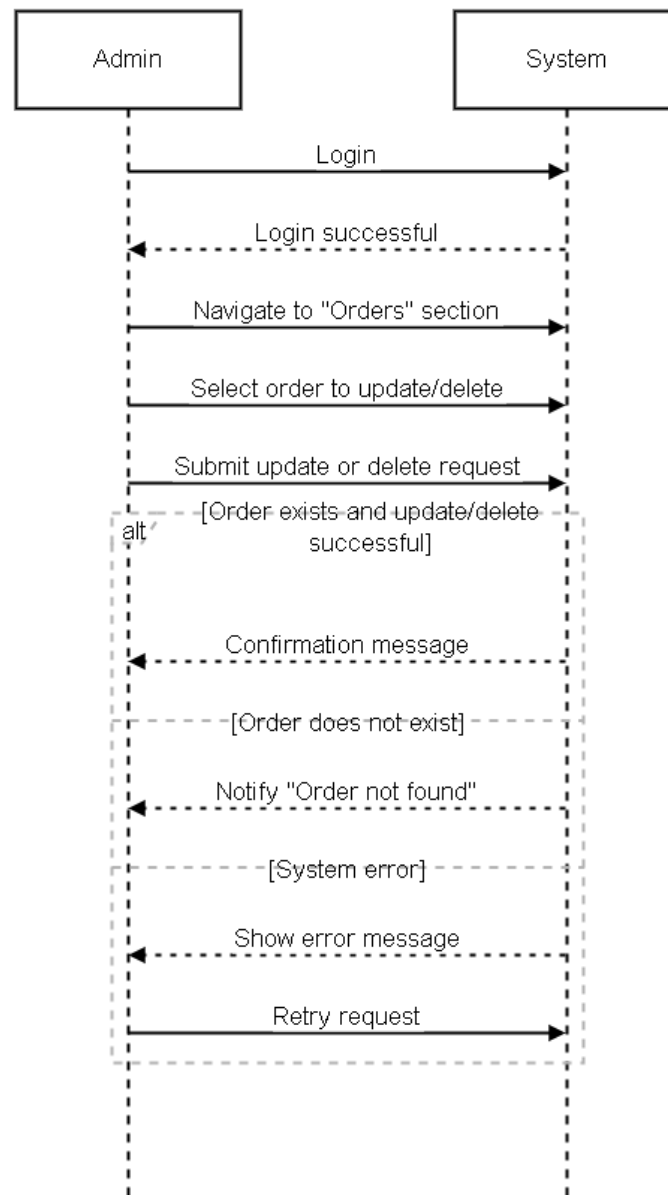


Figure 4.15: Admin Update / Delete Orders Sequence Diagram

Admin Update / Delete Users

Shows how the admin updates details or deletes user accounts.

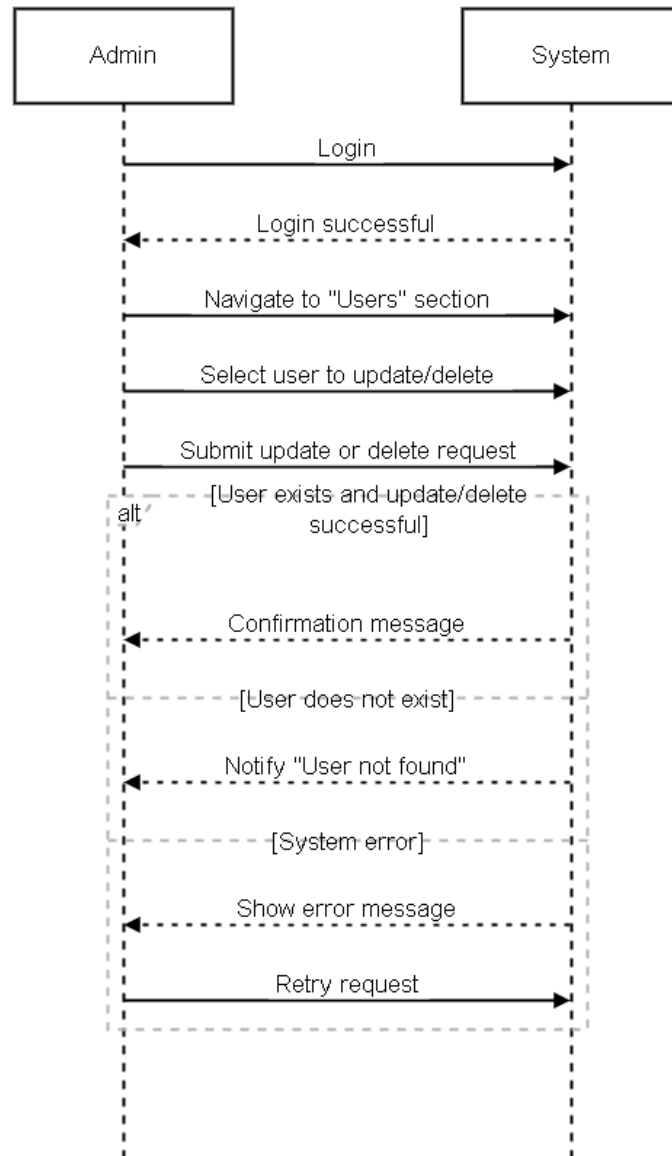


Figure 4.16: Admin Update / Delete Users Sequence Diagram

User Get Price Estimate

Illustrates how users enter moving details to receive an estimated cost from the system.

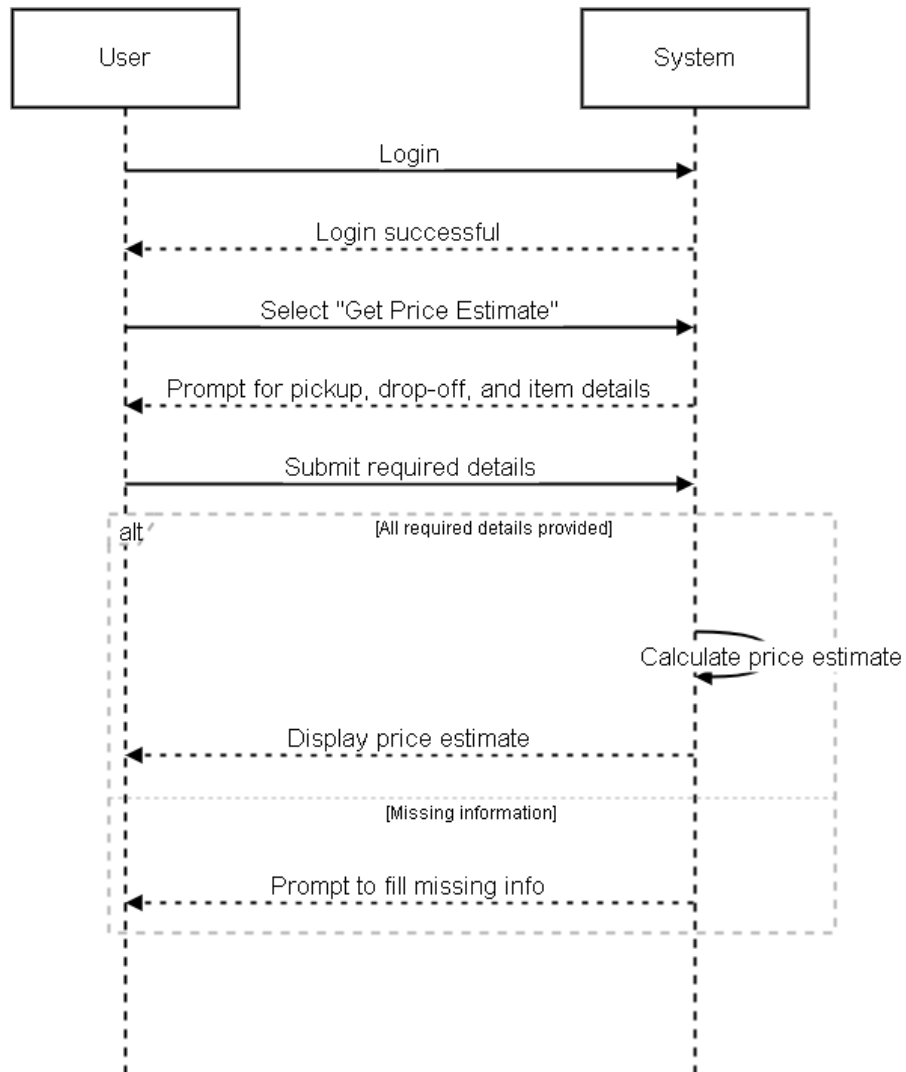


Figure 4.17: User Get Price Estimate Sequence Diagram

User Manage Profile

Describes how users update their personal information and the system saves the changes.

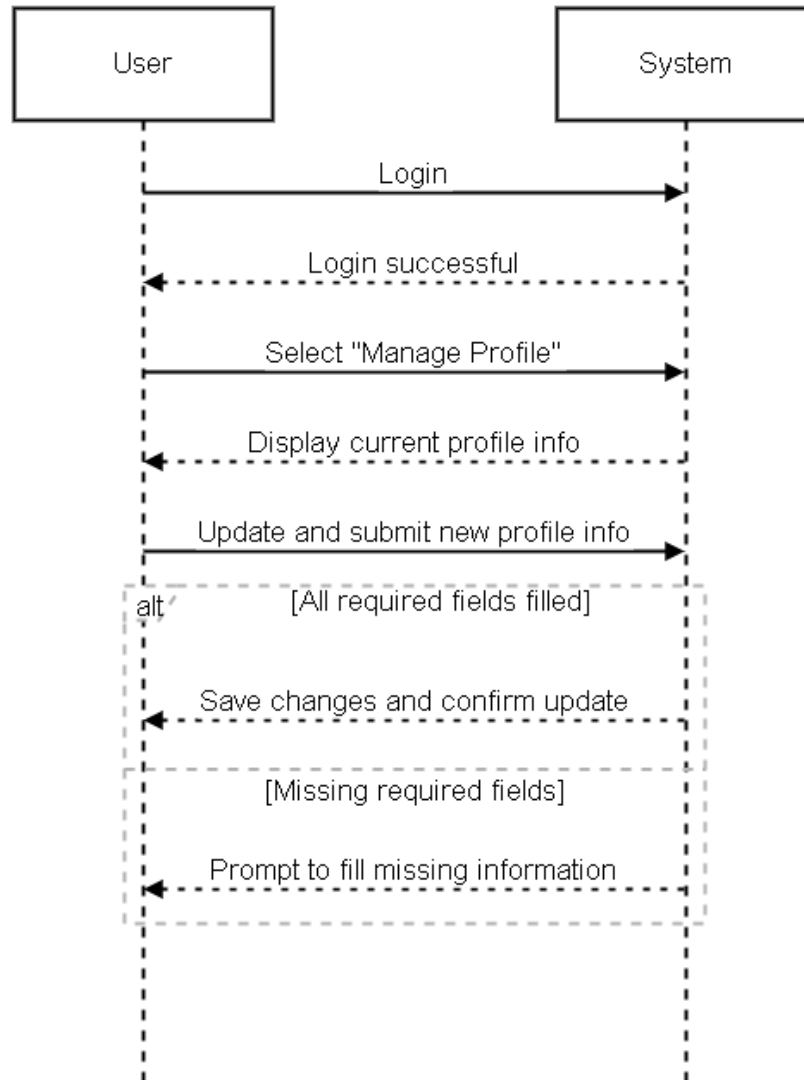


Figure 4.18: User Manage Profile Sequence Diagram

User Request Moving Service

Shows how users submit a moving request and receive confirmation from the system.

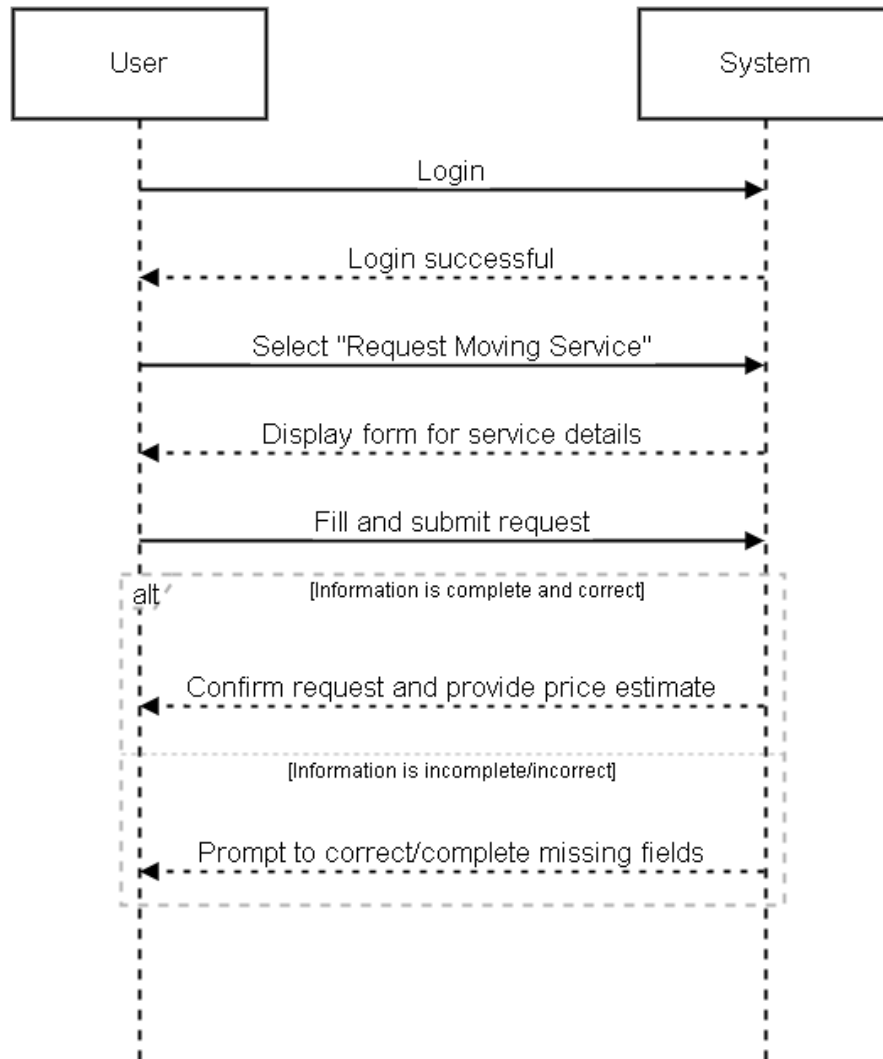


Figure 4.19: User Request Moving Service Sequence Diagram

User Track Moving Status

Explains how users view real-time updates and progress for their ongoing moving job.

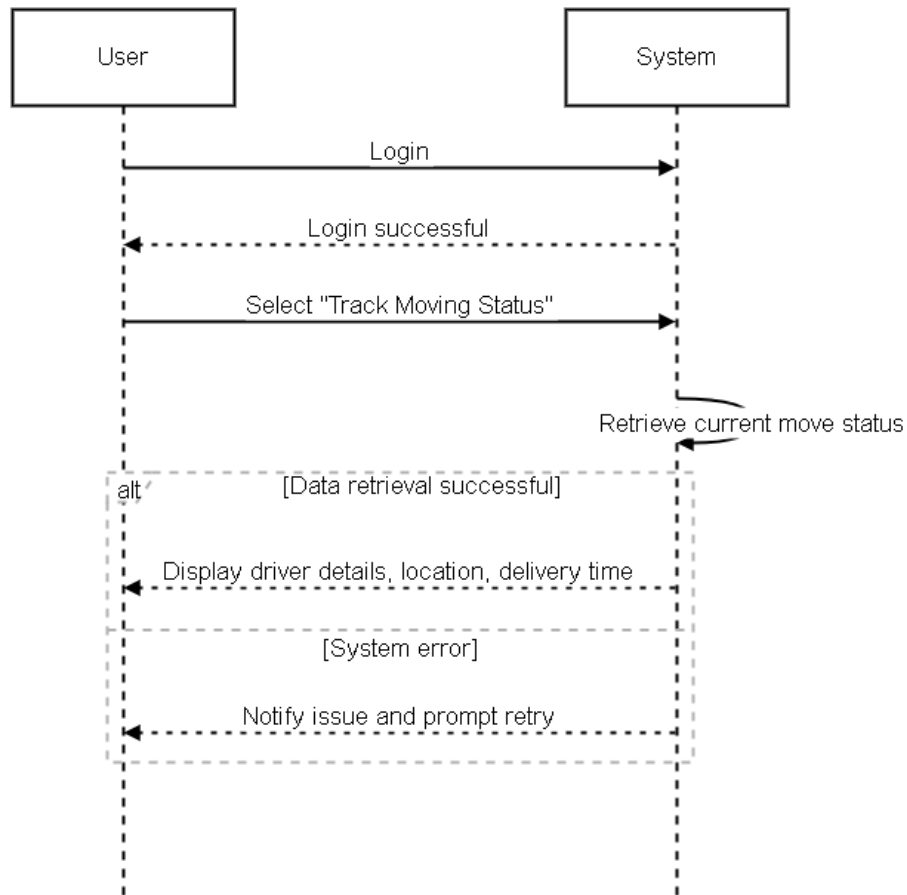


Figure 4.20: User Track Moving Status Sequence Diagram

Admin Verify Driver Documents

Shows how the admin reviews, approves, or rejects driver documents for authenticity.

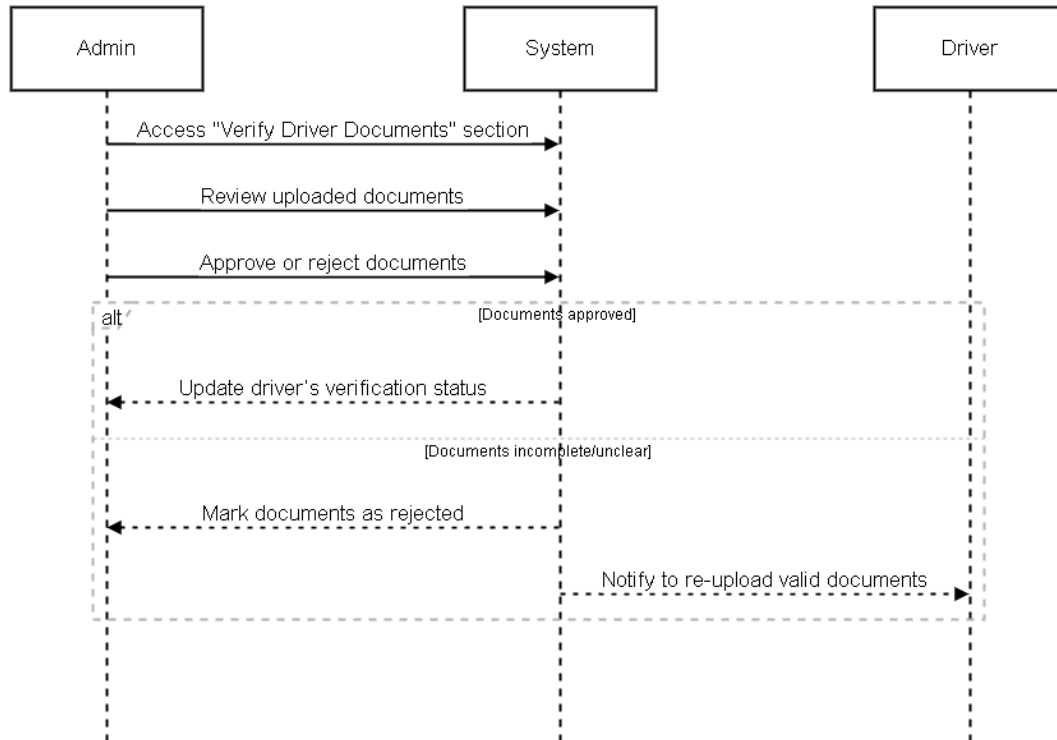


Figure 4.21: Admin Verify Driver Documents Sequence Diagram

Admin View Total Drivers Available

Illustrates how the system retrieves and displays the total number of currently available drivers.

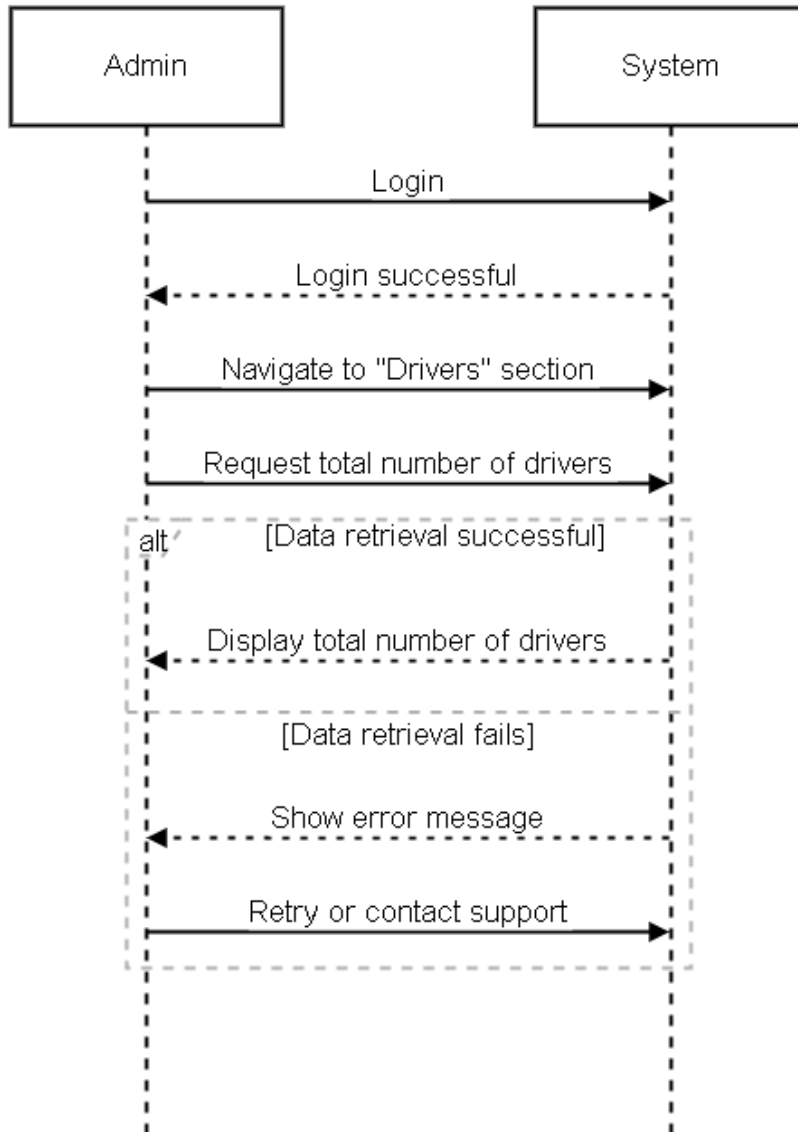


Figure 4.22: Admin View Total Drivers Available Sequence Diagram

Logout

Shows how the user initiates the logout process and the system terminates the active session securely.

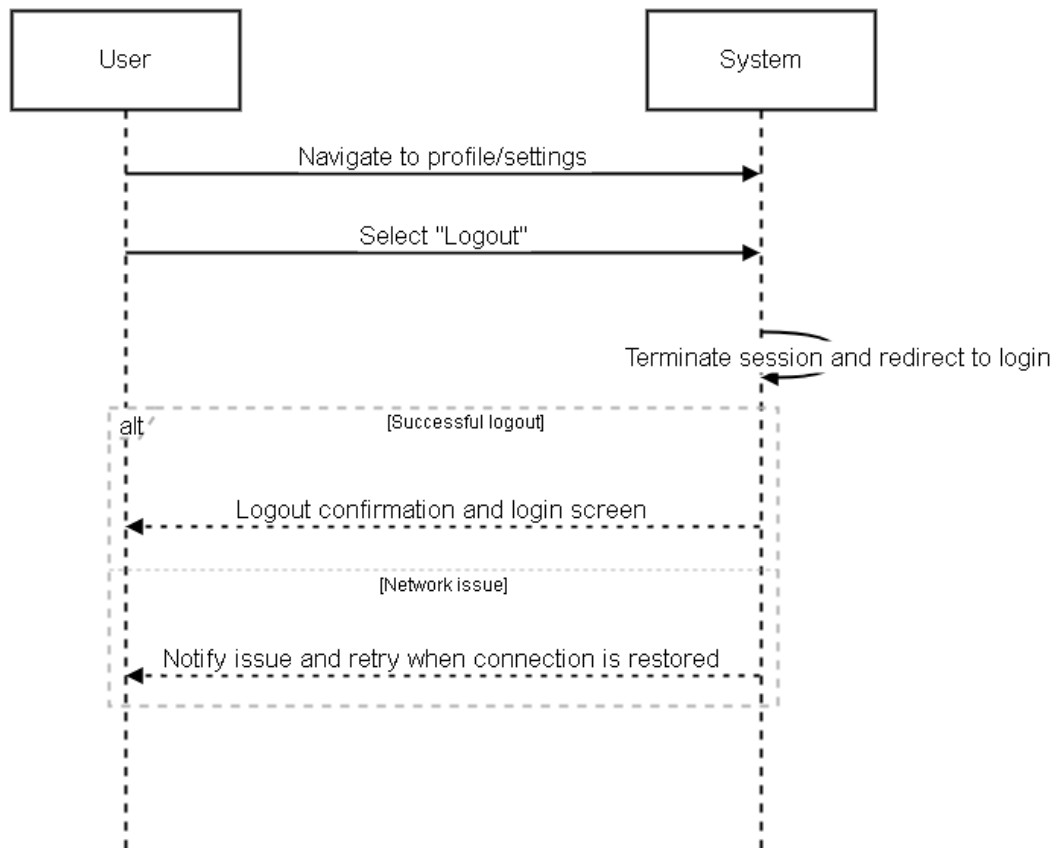


Figure 4.23: Logout Sequence Diagram

4.4 Sequence of Actions

A normal request that is being moved is as follows:

1. User uploads photos of stuff through the app.
2. YOLO analyses the images and categorizes everything to build an inventory.
3. Logistics module determines the path with the best route and computes the cost.
4. User gets real time tracking updates during the move.

4.4.1 Activity Diagram

Activity diagrams are in a way similar to flowcharts that describe the system from start to finish. Ours demonstrates the contact between all the various pieces during a move.

User

Shows the end-to-end flow of a typical relocation request from the user's perspective, including uploading item photos, requesting service, and tracking progress.

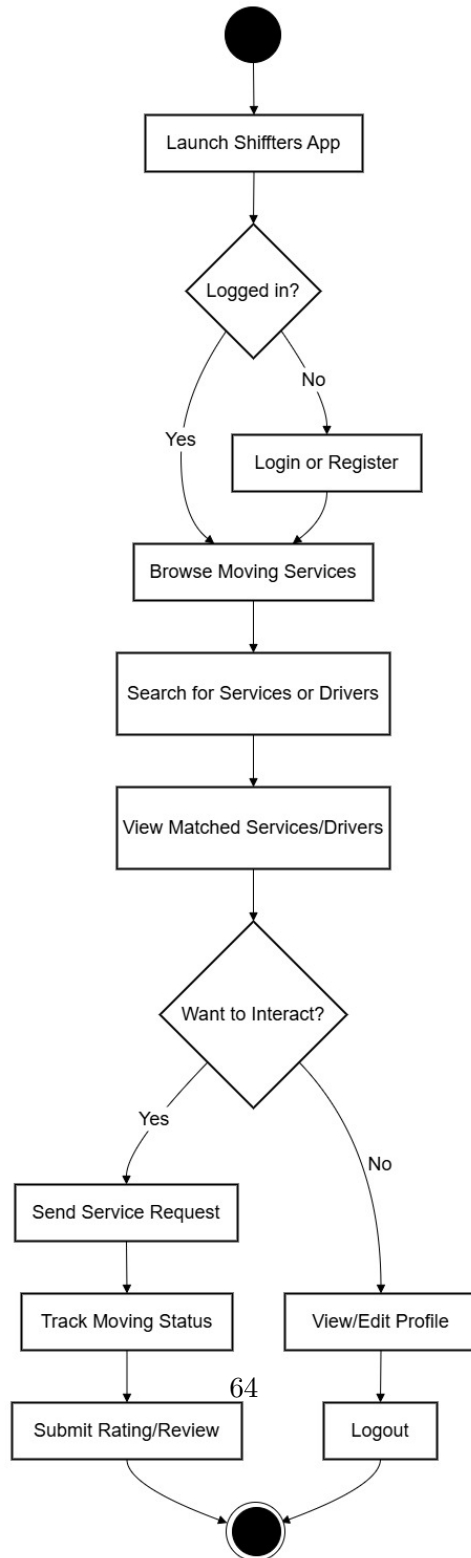


Figure 4.24: Activity Diagram

Admin

Illustrates how the admin oversees relocation requests, manages system updates, and responds to issues throughout the moving workflow.

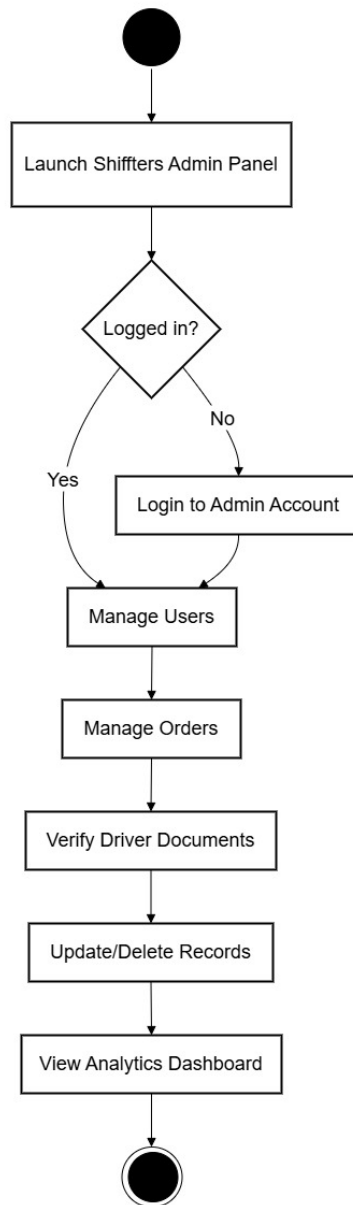


Figure 4.25: Activity Diagram

Driver

Depicts the driver's actions during a move, from receiving and accepting a job to picking up items, transporting them, and marking the job as completed.

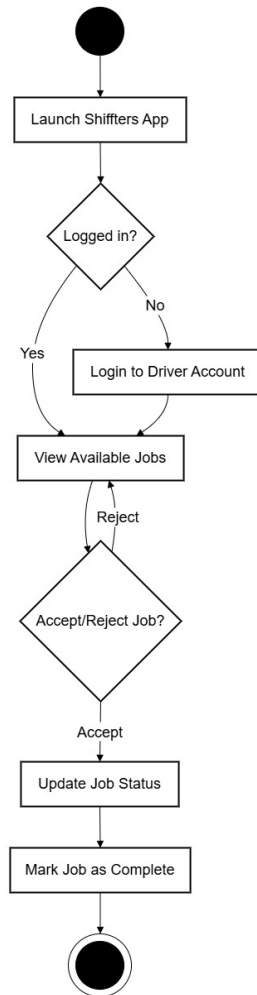


Figure 4.26: Activity Diagram

4.5 GUI Design

4.5.1 Splash Screen and Sign Up / Sign In Screen

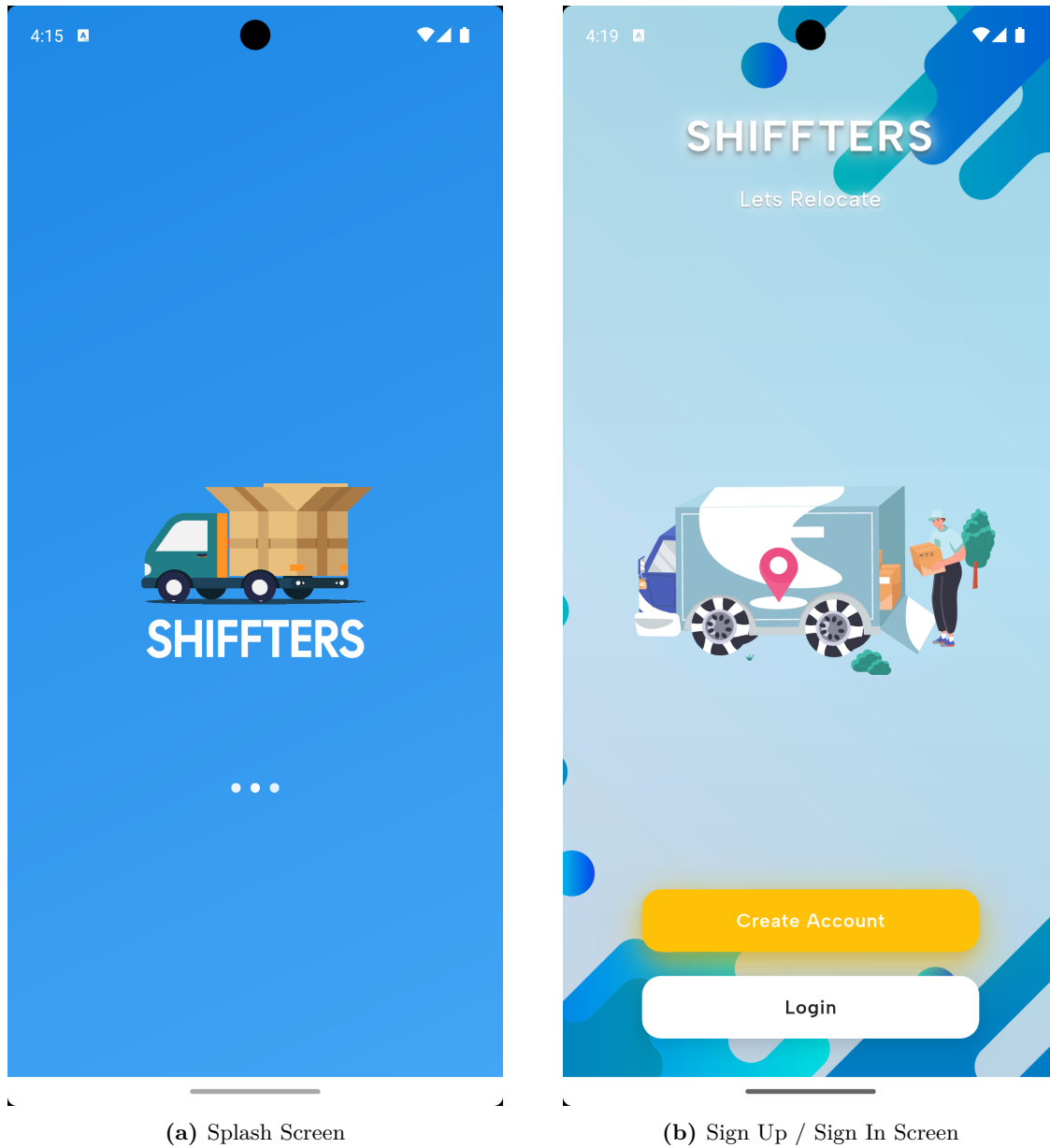
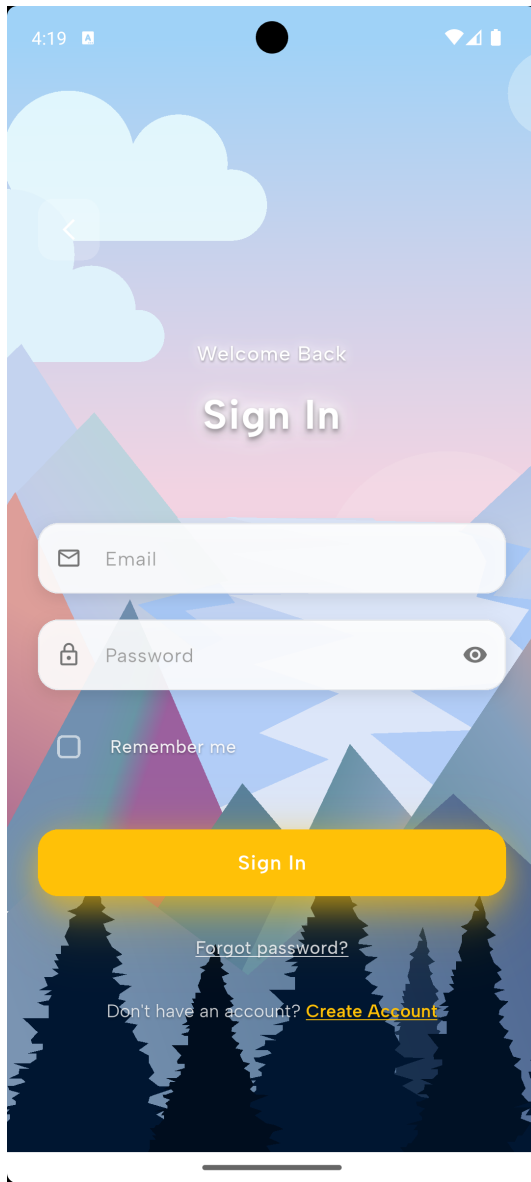
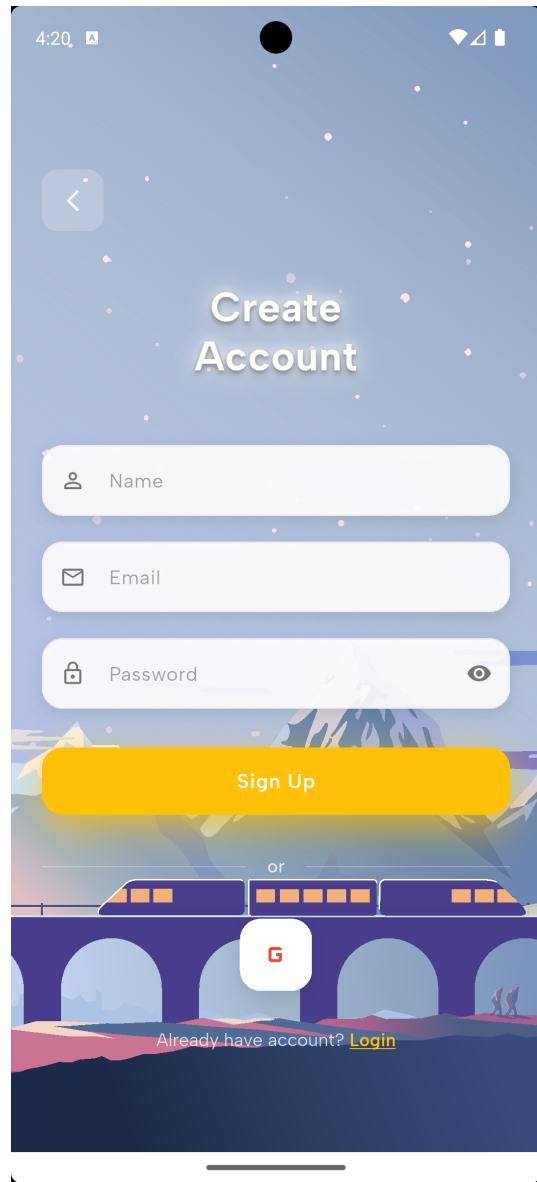


Figure 4.27: Key GUI screens of the Shifters application

4.5.2 Login Screen and Create Account Screen



(a) Login Screen



(b) Create Account Screen

Figure 4.28: Login and Account Creation Interfaces

4.5.3 Profile Screen and Home Screen

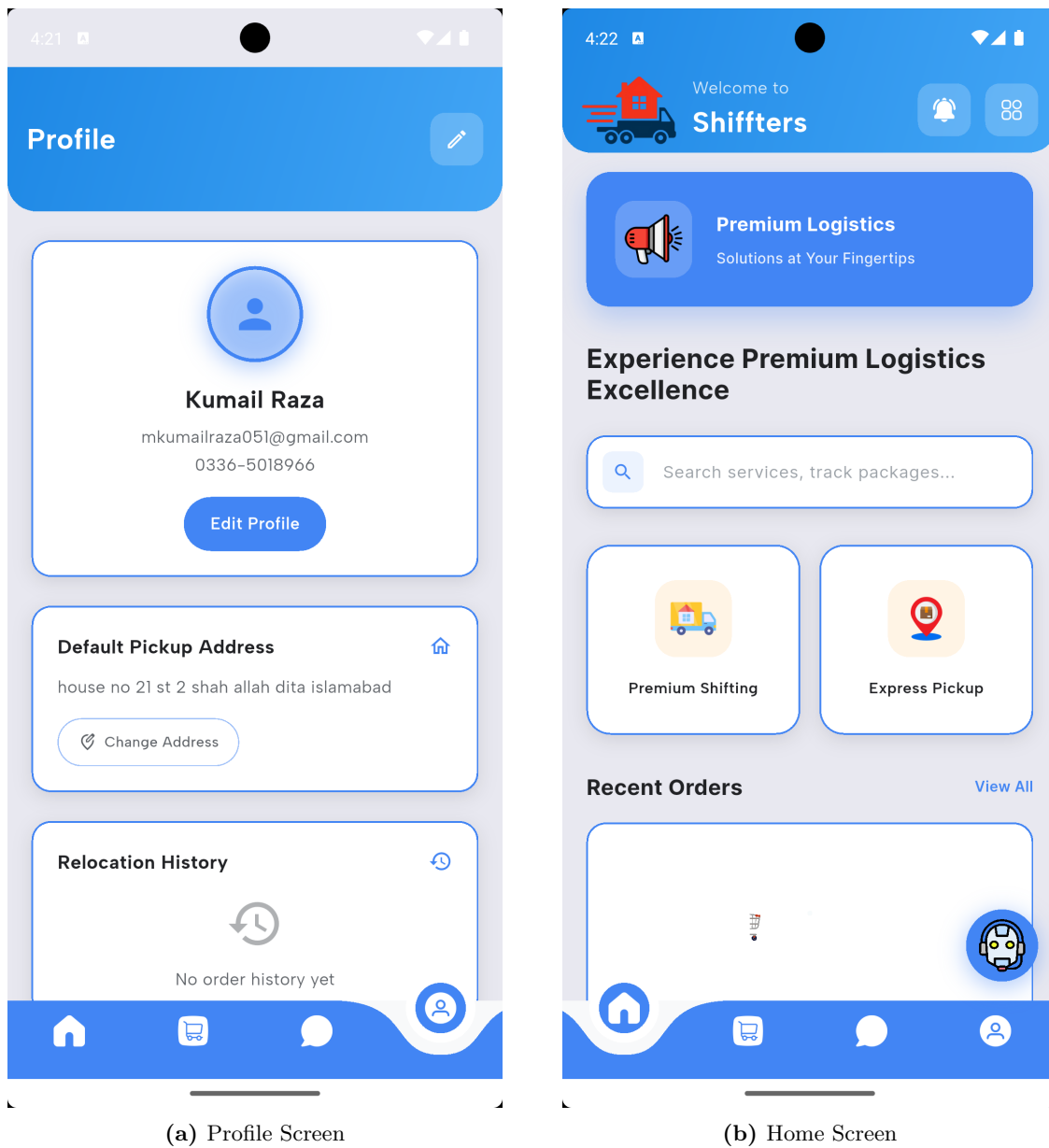


Figure 4.29: User Profile and Home Interface

4.5.4 View Orders Screen and In Transit Orders Screen

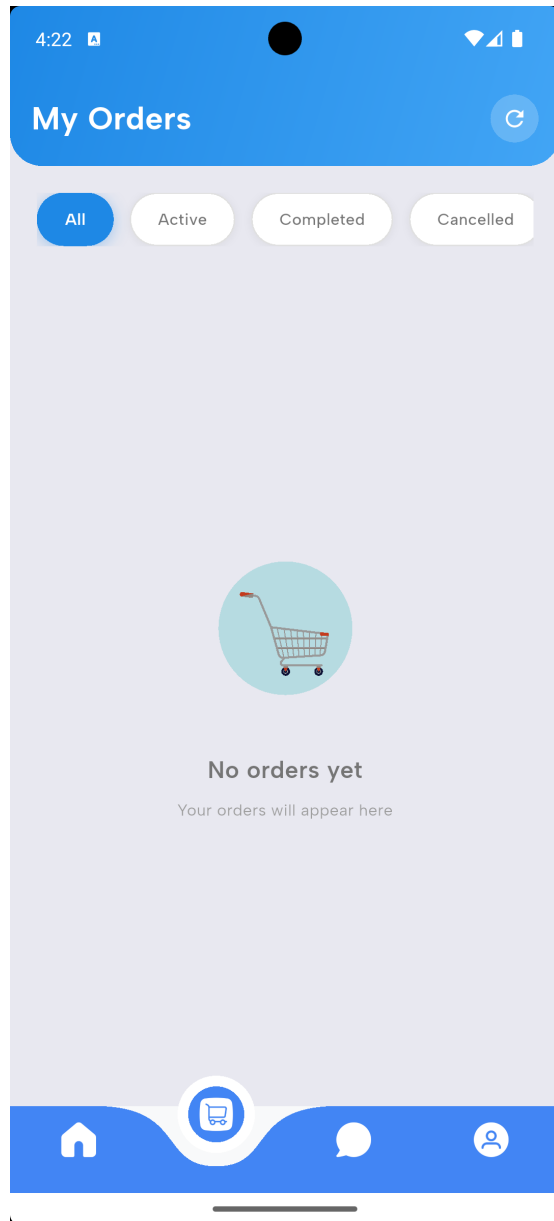


Figure 4.30: View Orders Screen

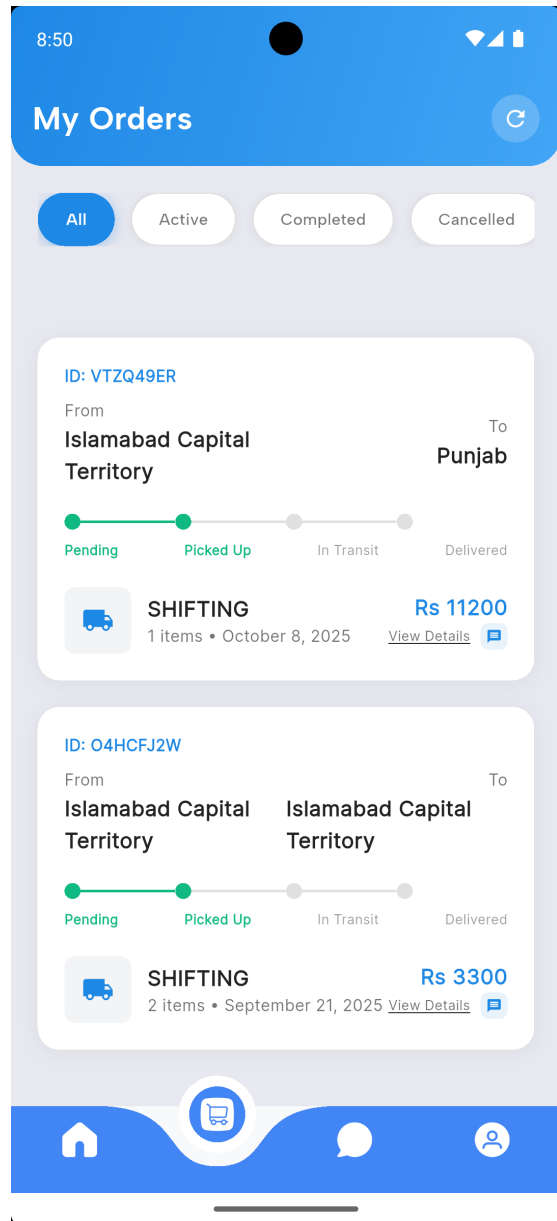
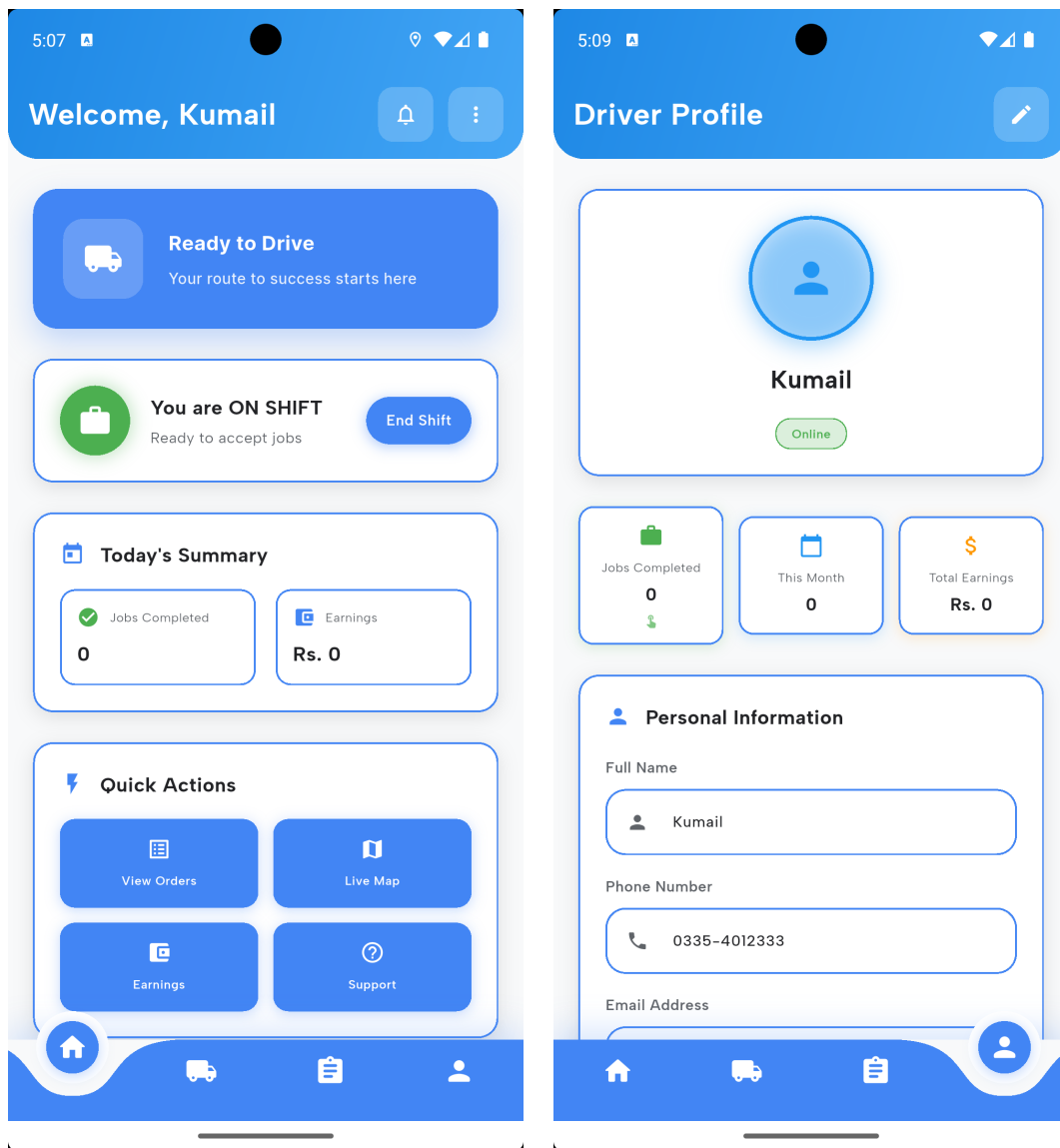


Figure 4.31: In Transit Orders Screen

4.5.5 Driver Dashboard Screen and Driver Profile Screen



Driver Dashboard Screen

Driver Profile Screen

Figure 4.32: Driver Dashboard and Driver Profile Screens

4.6 Activity Workflow

How the workflow goes:

1. User initiates a move request, puts their preferences and uploads items photos.
2. YOLO goes through the images and finds items and flags anything fragile or special.
3. System allows for the best route to be generated and for details to be given to the user [8].
4. Everything's tracked in real-time with every update sent for the user.

4.7 GUI Design Overview

We have designed the interface to be simple and easy to use. Key features:

- Dashboards for all of your things and their categories.
- Real time tracking with progress of route
- Easy menus for managing preferences, costs and move details.

Chapter 5

System Implementation

5.1 Introduction

The implementation is the simple stage in which the design is converted into actual, functioning code, such as when you have finally completed that piece of code assignment and the code begins to do what you have told it to do. We construct, assemble, test, and report that all things are well in making sure that it works well to the users. Shiffters relies on Flutter on the front-end, Firebase in the back-end, and YOLO to identify objects, and we have a good foundation of a scalable platform of a modern moving-industry solution.

5.2 System Architecture

The architecture diagram indicates the connection between all the components of Shiffters as well as the input of the user. Flutter uses one codebase to run the Android and iOS versions of the mobile app. Firebase is used to run our back-end, which is real-time database, authentication, cloud storage, and notifications. YOLO takes the feed of the camera and recognizes objects. We have experimented with the cloud-based logistics tracking in the past and it is efficient, thus we are relying on similar concepts to track the real-time tracker [10].

5.3 System Internal Components

Shifters is divided into various modules, which address individual characteristics. Here's what each one covers:

5.3.1 Registration and Authentication Module

The registration and authentication module deals with enrolling of all users into the system. Handles sign-up and log-in. It gathers email, password, authenticate all and store safely in the database. We have also included multi-factor authentication as an added security.

5.3.2 Password Recovery Module

Allows users to change lost passwords. They insert their email, we send them a verification code, and they form a new password -that is rather simple.

5.3.3 Inventory Management Module

Scans and recognizes items during a move using YOLO, which automatically generates a digital inventory, reducing the number of disputes. It also contrasts between that which was loaded and unloaded to highlight missing things and warn users.

5.3.4 Real-Time Tracking Module

This gives live updates to the user on their move. GPS follows the position of the truck and informs about the delays. All this can be found in the app [10].

5.3.5 Quote Request and Cost Estimation Module

Users place moving orders here- pickup, drop-off, what they are moving and when they desire it. It is processed through the system, and they are provided with a price estimate in real-time with the help of the data of YOLO.

5.3.6 Order Management Module

Takes care of all the orders that are in motion and making sure that they are documented, followed and updated in real-time. The app allows users to locate the status of their order.

5.3.7 Driver Management Module

This helps admins to add, update, and check the driver accounts. Drivers will be able to view jobs available, accept or reject, as well as keep the job status updated.

5.3.8 Report Generation Module

Gathers a full report of the relocation including inventory information and discrepancy and emails it to the user.

5.3.9 Session Management Module

Monitors user activities on the computer by assigning users unique IDs, making things safe and tracking user activity during a user session.

5.3.10 Query Management Module

Here, admins have access to customer questions and can reply to them. All the information is under one roof so that they can respond in a quick manner.

5.3.11 Alert Missing Items Module

Makes a comparison between loaded and unloaded inventory to identify items that are missing. Alerts user and admins so that they can deal with it immediately.

5.3.12 Driver Job Status Update Module

Here drivers update their job status which can be either In Progress, Completed, or Cancelled. This keeps everybody posted at any moment.

5.3.13 Admin Dashboard Module

Managing user, driver and order administration make point. They will be able to check the driver documents, modify or cancel the orders, and view the number of drivers available.

Chapter 6

System Testing

6.1 System Testing

We literally whacked Shifters to ensure that all really works. Each of the modules was individually tested to determine whether it does what it is supposed to do. The entire idea was to ensure that the app satisfies all the requirements that we had set initially. We saw what is working, what is not and what should be fixed. Tested a few cases to determine its performance.

Here's what we tested:

1. Test GUI Graphical User Interface.
2. Usability Testing
3. Software Performance testing.
4. Compatibility Testing
5. Exception Handling
6. Load Testing
7. Security Testing
8. Installation Testing

6.2 Graphical User Interface (GUI) Testing

6.2.1 Register

Table 6.1: Test Case for Register

Steps	Tasks	Expected Output	Result (Pass/Fail)
1	Open register screen.	Register screen is displayed.	Pass
2	Enter required fields.	Fields are validated.	Pass
3	Press register button.	User is registered successfully.	Pass
4	Check button is submitting data.	Data is submitted to the database.	Pass

6.2.2 Login

Table 6.2: Test Case for Login

Steps	Tasks	Expected Output	Result (Pass/Fail)
1	Open login screen.	Login screen is displayed.	Pass
2	Enter email and password.	Fields are validated.	Pass
3	Press login button.	User is logged in successfully.	Pass
4	Check button is submitting data.	Data is submitted to the database.	Pass

6.2.3 Forgot Password

Table 6.3: Test Case for Forgot Password

Steps	Tasks	Expected Output	Result (Pass/Fail)
1	Open forgot password screen.	Forgot password screen is displayed.	Pass
2	Enter email.	Email is validated.	Pass
3	Press submit button.	Verification code is sent to the email.	Pass
4	Check button is submitting data.	Data is submitted to the database.	Pass

6.2.4 Logout

Table 6.4: Test Case for Logout

Steps	Tasks	Expected Output	Result (Pass/Fail)
1	User is logged in.	User session is active.	Pass
2	Press logout button.	User is logged out successfully.	Pass
3	Redirect to login screen.	Login screen is displayed.	Pass

6.3 Usability Testing

6.3.1 Quote Request and Cost Estimation

Table 6.5: Test Case for Quote Request and Cost Estimation

Steps	Tasks	Expected Output	Result (Pass/Fail)
1	Open quote request screen.	Quote request screen is displayed.	Pass
2	Enter pickup and drop-off locations.	Locations are validated.	Pass
3	Enter item details.	Item details are validated.	Pass
4	Press submit button.	Cost estimate is displayed.	Pass

6.4 Performance Testing

6.4.1 System Performance Testing

Table 6.6: Test Case for System Performance

Steps	Tasks	Expected Output	Result (Pass/Fail)
1	Run the system.	System starts successfully.	Pass
2	Login to the system.	User is logged in successfully.	Pass
3	Navigate through the app.	Navigation is smooth and responsive.	Pass
4	Submit a quote request.	Quote is processed successfully.	Pass
5	Add items to the cart.	Items are added successfully.	Pass
6	Place an order.	Order is placed successfully.	Pass
7	Logout from the system.	User is logged out successfully.	Pass

Chapter 7

Conclusion

Pakistan had a dire need of catching up with technology in the moving industry. We created Shifters to rectify the inefficiencies, transparency deficiencies and old fashioned processes that have been a pain to everyone. With Flutter, Firebase, and YOLO as a tool to identify objects, we have developed a platform that really works with their customers, drivers, and admins.

Upon launching this project, the same grievances continued to reoccur. Individuals would outsource movers and not know where their goods were. The prices would take a sharp increase since there was no clear monitoring of what was moved. Boxes disappeared and nobody owned them. It was dismembered in every direction.

The object recognition of inventory, real time tracking, and automated pricing is what makes Shifters unique. These are not just smacky features but they do address genuine issues and create confidence amongst the customers and movers. The camera scans the items, recognizes them automatically, creates a digital inventory, and one does not have to write something manually. You are able to track the location of your truck in real time. The price is calculated in advance, depending on the quantity of what you are actually transporting, rather than a haphazard estimate that will be altered later.

We have made it to be friendly to all the people involved and constructed it to enlarge

as more people use it. Customers are provided with transparency that they have never enjoyed. Drivers receive specific job descriptions and equitable assignments. All of that is controlled by admins with the help of a single dashboard rather than phone calls and paperwork. It is not ideal, but it is much better than what was the case.

The project is important in the sense that it indicated that you can find yourself competing globally with the service industries in Pakistan. We do not need to bring in solutions that are developed in other markets. Here, we are able to create structures that are useful in our particular context, our user behaviours, and our infrastructure reality. That is what we planned to do to Shiffters, and, frankly speaking, we believe that we were quite close.

References

- [1] A. Solanki and S. Jadiga. Ai applications for improving transportation and logistics operations. *International Journal of Intelligent Systems and Applications in Engineering*, 12(3):2607–2617, 2024.
- [2] E. P. Sinesio, M. E. Fontana, J. C. F. de Guimarães, and P. C. Marques. Autonomous last-mile logistics in emerging markets: A study on consumer acceptance. *Logistics*, 9(3):106, 2025.
- [3] Moovaz. Moovaz expands beyond relocation services to ecosystem for human mobility. <https://www.moovaz.com/moovaz-ecosystem-announcement>, 2021. Accessed Nov. 8, 2021.
- [4] A. B. Tineo Díaz. Impacto de una aplicación móvil con inteligencia artificial en la gestión de almacenamiento en empresas logísticas: Una revisión de literatura, 2024.
- [5] E. Gent. This startup’s ai tool makes moving day easier. *IEEE Spectrum*, 2024. Accessed Mar. 27, 2024.
- [6] T. Gajewska and D. Walczyk. Development of transport management software. *Sustainability*, 15(15):12083, 2023.
- [7] H. Benjamin. Celebrating 10 years of lugg. <https://blog.lugg.com/celebrating-10-years-of-lugg-9fc94b9f63fb>, 2024. Accessed Oct. 16, 2024.

- [8] B. R. Aquino Muñoz and A. D. Valencia De León. Propuesta de optimización de las rutas de distribución en una empresa de productos de primera necesidad mediante la mejora de la gestión logística y la planificación de recursos, 2025.
- [9] K. Shoilekova, B. Ivanova, and B. Blazhev. A web application for moving from one spot to another by using different transport—a scenario of the research. In *Software Engineering Research in System Science (CSOC 2023)*, volume 722 of *Lecture Notes in Networks and Systems*, pages 608–612. Springer, 2023.
- [10] J. F. Odesanya and O. G. Famosipe. Development of a cloud-based mobile-app tracking system for outbound logistics distributions functions. *Transportation Research Procedia*, 89:42–50, 2025.