



BEACONHOUSE NATIONAL UNIVERSITY

PRJ-F23/335

**SAAFPAKISTAN:
REVOLUTIONIZING RECYCLING IN
PAKISTAN**

INTERNAL SUPERVISOR

MS. HUDA SARFRAZ

GROUP MEMBERS

MIAN FAIZAN MUNAWER	F2020-148
NOUMAN ALI	F2020-149
ALI SHER	F2020-158

SCHOOL OF COMPUTER & INFORMATION TECHNOLOGY

Project Title	SAAFPAKISTAN	
Project ID	PRJ-F23/335	
Project Supervisors	Ms. Huda Sarfraz (Internal)	
Group Members	Mian Faizan Munawer	F2020-148
	Nouman Ali	F2020-149
	Ali Sher	F2020-158
Academic Session	2023-24 (Sept 2023 to June 2024)	
Credit Hours	6	

PROJECT APPROVAL

This Project is approved in partial fulfilment of the requirements of BSc (Hons.) in Software Engineering degree conducted by the School of Computer and IT, Beaconhouse National University, Lahore.

Ms. Huda Sarfraz
Internal Supervisor

Prof. Dr. Shafaat Ahmed Bazaz
Dean School of Computer and IT

Date: _____

Acknowledgement

We would like to extend our heartfelt gratitude to all those who have stood by us throughout our final year project. This project has been a significant aspect of our academic journey, and we are deeply appreciative of the invaluable support and guidance we have received.

Firstly, we express our sincere thanks to our families for their unwavering support, encouragement, and belief in our capabilities. Their steadfast encouragement has served as a constant source of inspiration and motivation throughout this endeavour.

We would also like to acknowledge the guidance and support provided by our internal advisor, Ms. Huda Sarfraz. Her expertise and commitment to our project have played a pivotal role in shaping our research and its outcomes.

Furthermore, we extend our gratitude to our classmates, friends, and peers for their active participation in discussions and for providing constructive feedback on our work. Their diverse perspectives have contributed significantly to the refinement of our project.

Special recognition is due to Ms. Shazia Rizwan for her invaluable assistance and support throughout this project. Her guidance and insights have been instrumental in overcoming obstacles and achieving our objectives.



SaafPakistan

Scan To View Code & Implementation Files

Table of Contents

1	Introduction.....	1
1.1	Problem Statement	2
1.2	Existing Systems	2
1.3	Literature review	3
1.3.1	Gamification for Recycling.....	3
1.3.2	Financial Incentives and Gamification.....	3
1.3.3	Carbon footprint calculators.....	4
1.3.4	Gamification to Promote Tourist Recycling Behavior.....	4
1.4	Additional research	4
1.4.1	iRecycle Mobile App	4
1.4.2	Recycle Coach Mobile App	4
1.4.3	Recycle Smart Mobile App.....	5
1.4.4	The Click Mobile App	5
1.4.5	Zeloop Mobile App.....	6
1.4.6	Gaia Mobile App.....	7
1.5	Solution.....	8
1.5.1	Recycling Pickup Scheduling	8
1.5.2	Gamification Leaderboard System.....	8
1.5.3	Add Friends and Friends Leaderboard	8
1.5.4	Rewards and Compensation.....	8
1.5.5	Corporate Onboarding.....	8
1.5.6	Advertisement as a Green Company.....	8
1.5.7	Educational Content	8
1.5.8	Dashboard Summary.....	9
2	Requirement Analysis	10
2.1	User Roles.....	11
2.2	User Story	11
2.2.1	Scheduling a Pickup for Recycling	11
2.2.2	Collecting Recyclable Waste from Customers' Homes	11
2.3	Prototypes	12
2.3.1	Customer App Prototypes	12
2.3.2	Rider App Prototypes.....	13
2.4	Use Case Diagram.....	14

2.5	Functional Requirements Mobile App.....	15
2.5.1	Customer Requirements	15
2.5.2	Rider Requirements.....	21
2.6	Functional Requirements Web App.....	23
2.6.1	Warehouse Manager Requirements	24
2.6.2	Admin Requirements	25
2.7	Non-Functional Requirements	28
3	System Design & Architecture	30
3.1	Technologies	31
3.1.1	WebApp Technology	31
3.1.2	MobileApp Technology	31
3.2	System Context Diagram	32
3.3	High-level Architecture.....	33
3.4	Low-Level Architecture.....	34
3.4.1	Mobile App Component Diagram.....	34
3.4.2	Web App Component Diagram.....	35
3.5	Database Schema	36
3.6	Sequence Diagram	37
3.6.1	Schedule Pickup (FR 1.6)	37
3.6.2	Complete Pickup (FR 1.22-1.26)	38
4	Implementation	39
4.1	Test Cases	40
4.1.1	Test Cases Mobile App (FR 1.1 – FR 1.28).....	40
4.1.2	Test Cases Web App (FR 2.1 – FR 2.19).....	47
4.2	Test Case Grid.....	52
4.3	Mobile App Implementation.....	54
4.3.1	Navigating Customer to Schedule Pickup (Frontend)	54
4.3.2	Displaying Top 3 Organizations (Backend).....	55
4.4	Web App Implementation.....	56
4.4.1	Fetching Leaderboard Data From API.....	56
4.4.2	Fetching Leaderboard Data From Firebase.....	57
4.5	Screenshots Of Key Features	59
4.5.1	SaafPakistan Mobile Application Screenshots.....	59
4.5.2	SaafPakistan Web Application Screenshots.....	63
5	Conclusion	64

6	References.....	66
---	-----------------	----

Table of Figures

Figure 1.1 Recycle Coach App Preview	5
Figure 1.2 iRecycle Mobile App Preview.....	5
Figure 1.3 Recycle smart Mobile App Preview	6
Figure 1.4 The Click Mobile App Preview.....	6
Figure 1.5 Gaia Mobile app Preview	7
Figure 2.1 SaafPakistan Customer's App prototypes	12
Figure 2.2 SaafPakistan Rider's App prototypes	13
Figure 2.3 Use Case diagram	14
Figure 3.1 System Context diagram.....	32
Figure 3.2 High-level Architecture	33
Figure 3.3 Mobile App Component Diagram	34
Figure 3.4 Web App Component Diagram	35
Figure 3.5 Schema Diagram.....	36
Figure 3.6 Sequence diagram: Schedule Pickup	37
Figure 3.7 Sequence Diagram: Complete Pickup	38
Figure 4.1 Flutter Widget Code Snippet (Frontend)	54
Figure 4.2 Dart Function Code Snippet (Backend).....	55
Figure 4.3 React code Snippet	56
Figure 4.4 Controller Code Snippet	58
Figure 4.5 Leaderboards Screenshot Admin panel	58
Figure 4.6 Customer Dashboard	59
Figure 4.7 Maps Integration for Address Registration.....	60
Figure 4.8 Customer Recycling Leaderboards.....	61
Figure 4.9 Map Integration for Rider.....	62
Figure 4.10 Admin Dashboard.....	63

Abstract

This report discusses the SaafPakistan project, which aims to revolutionize waste management and recycling practices in Pakistan, primarily focusing on Lahore. The primary objective is to address the city's pressing waste disposal issues by introducing a structured, efficient, and engaging system that promotes responsible recycling habits among individuals and businesses. The mobile application incorporates features such as recycling pickup scheduling, leaderboards, rewards and compensation, corporate onboarding, and educational content. The development process involved extensive research into existing systems and technologies, leading to the creation of a user-friendly interface and a robust backend system. Testing and implementation phases demonstrated the application's potential to significantly improve user engagement and operational efficiency. The results suggest that integrating gamification and financial incentives can effectively encourage sustainable behaviors, contributing to a cleaner and more environmentally conscious community in Lahore. Future enhancements will focus on refining the application's features and expanding its scope to include a broader range of waste management services across Pakistan.

1 Introduction

Waste management has emerged as a pressing issue in cities worldwide, with Lahore, Pakistan, being no exception. The prevailing waste disposal practices in Lahore are largely informal and inconsistent, resulting in environmental degradation, health hazards, and unsightly conditions. However, amidst these challenges, initiatives like the Solid Waste Management Initiative by the Aabroo Educational Welfare Organization have showcased the potential for change. Recognizing the need for sustainable solutions, organizations and researchers have been exploring innovative approaches to encourage responsible waste management practices.

One such approach gaining traction is the concept of gamification, which applies game-like elements to non-game contexts, including waste management. Research studies have highlighted the effectiveness of gamification in increasing recycling rates, particularly when combined with financial incentives, as identified by the World Bank. Moreover, the proliferation of mobile applications dedicated to waste management and recycling, such as Recycle Coach, iRecycle, and Recycle Smart, underscores the growing interest in leveraging technology to address environmental challenges.

In this context, the proposed solution, SaafPakistan, aims to revolutionize waste management and recycling practices in Lahore through a mobile application. By incorporating features like recycling pickup scheduling, gamification leaderboards, rewards and compensation, corporate onboarding, and educational content, SaafPakistan seeks to engage individuals and businesses in sustainable practices while fostering a sense of community and environmental responsibility.

Through the development and implementation of SaafPakistan, it is envisioned that Lahore can move towards a more sustainable and environmentally-conscious future, where waste is managed efficiently, and recycling becomes an integral part of daily life for residents and businesses alike.

1.1 Problem Statement

In Pakistan, the absence of a structured recycling system, marked by the absence of separate bins for recyclables, has led to a widespread habit of indiscriminate waste disposal. This contributes significantly to environmental pollution and the buildup of recyclable materials in landfills. The problem is intensified by a lack of awareness among the public about the environmental consequences and the absence of incentives to promote responsible waste management. The absence of initiatives addressing these issues not only worsens environmental pollution but also hinders the sustainable utilization of valuable recyclable resources.

1.2 Existing Systems

In Lahore, Pakistan, the prevailing waste management system heavily relies on informal and inconsistent practices. Residents and businesses typically dispose of their waste by dumping it in open areas, streets, or makeshift trash piles near their homes or places of business. These haphazard disposal methods often result in unsightly, unsanitary, and environmentally harmful conditions.

The collected waste is typically gathered by local waste collector or municipal workers who manually sort through it to salvage recyclable materials. Non-recyclable waste is then transported to landfills, which are often poorly managed and insufficiently regulated, leading to pollution and health hazards. Households in Lahore rarely segregate waste, with organic and inorganic materials tossed in together. This means that otherwise recyclable materials, such as paper and plastic, become contaminated with bacteria, reducing their quality and usefulness.

Even if waste was segregated at the basic level with separate bins for organic and inorganic waste, we could significantly reduce the amount of garbage that ends up in landfill sites in Lahore. Due to the absence of a well-structured recycling system, valuable recyclable resources are frequently mixed with non-recyclable waste, exacerbating environmental issues.

An NGO Aabroo Educational Welfare Organization, based in Lahore, has recognized the need for sustainable waste management solutions and has taken a pioneering step in this direction. Since 2007, they have initiated a successful Solid Waste Management Initiative, collecting recyclable waste from households and institutions. Aabroo Welfare Organization has introduced the concept of in-house segregation of biodegradable and dry recyclable domestic waste, along with its door-to-door collection from affluent households, as a donation for the first time in Pakistan. Their smart workers visit households weekly on a fixed schedule to collect garbage bags filled with recyclable dry waste. They also provide fresh, clean empty garbage bags free of cost for the next week. The dry waste collected daily from the community is taken to their waste recycling facility, where it is sorted by well-trained staff before being sold to recycling companies.

This program not only generates funds to support the organization but also promotes environmental awareness with a focus on the 3Rs (Reduce-Reuse-Recycle). Currently serving around 7,000 households and 28,000 donors each month, the initiative covers one-third of Aabroo's monthly expenses, with the goal of achieving full financial sustainability in the near future.[11]

Still, there is a lack of incentives for responsible waste management in Lahore, discouraging individuals and businesses from actively participating in recycling efforts. These challenges highlight the urgent need for a comprehensive and organized waste disposal and recycling solution in Lahore, such as SaafPakistan.

1.3 Literature review

As we researched for smart ideas to encourage good habits and positive results, we found one interesting thing, “Gamification”, which means using game-like features in non-game situations. We're also really interested in tools like carbon footprint calculators that help figure out how our actions affect the environment. We have learned that combining these game ideas with tools like carbon calculators could help make our future more sustainable.

1.3.1 Gamification for Recycling

A study "Gamification for Recycling: A Review of the Literature" by the University of California, Berkeley (2020) found that gamification can also be effective at increasing recycling rates. The study found that a program that rewarded people with points for recycling led to a 15% increase in the recycling rate.[1]

1.3.2 Financial Incentives and Gamification

A study "Financial Incentives and Gamification to Increase Recycling Rates" by the World Bank (2021) found that financial incentives and gamification are most effective at increasing recycling rates in countries with high levels of income and education.[2]

1.3.3 Carbon footprint calculators

The article "The state of carbon footprint calculators: An evaluation of calculator design and user interaction feature by John Mulrow" is about the state of carbon footprint calculators. It discusses the growing interest in these tools and the variety of calculators available. The authors note that there is no standardization in the way calculators are designed or the data they use. This makes it difficult to compare results from different calculators. The authors also discuss the importance of user engagement and how calculators can be used to educate people about their carbon footprint.[3]

1.3.4 Gamification to Promote Tourist Recycling Behavior

Gamification is a promising approach to promoting tourist recycling behavior, as shown in the article "Gamification as An Approach to Promote Tourist Recycling Behavior" by Lidia Aguiar-Castillo et al. (2019). It can make recycling more fun and engaging for tourists, and encourage them to recycle more often. Some examples of gamification for tourist recycling include awarding points and badges, using leaderboards, and creating challenges.[4]

In light of these research findings, it becomes evident that targeted strategies, such as gamification and financial incentives, hold the power to significantly enhance recycling rates. Furthermore, standardizing the design and features of carbon footprint calculators remains crucial. This not only facilitates precise environmental impact assessment but also functions as an educational instrument, guiding individuals toward more responsible and sustainable behaviors. Collectively, these approaches offer a promising pathway toward a more sustainable and environmentally-conscious future.

1.4 Additional research

1.4.1 iRecycle Mobile App

iRecycle is a mobile app that serves as an invaluable tool for anyone looking to make a positive impact on the environment. With access to over 110,000 recycling programs across the United States, the app simplifies the process of sorting and disposing of recyclables. Not only does it provide location, telephone, and website connections for recycling facilities, but it also offers insights into what can be dropped in your blue bin. [6]

1.4.2 Recycle Coach Mobile App

Recycle Coach is a mobile app and website that helps users recycle and compost correctly. It provides information on what materials can be recycled or composted in their area, as well as tips on reducing waste. Recycle Coach also has a feature that allows users to set reminders for their recycling and composting pickup days. [5]

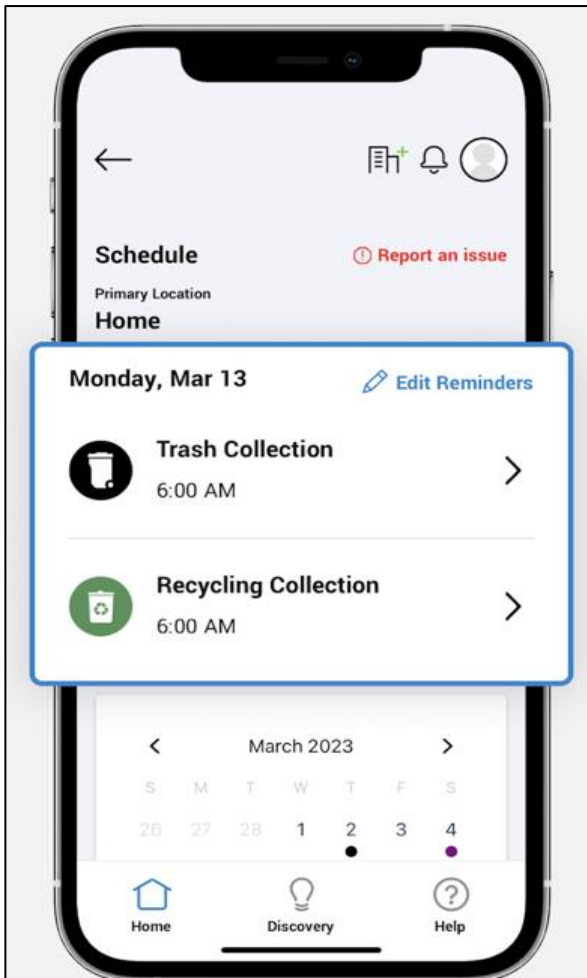


Figure 1.1 Recycle Coach App Preview

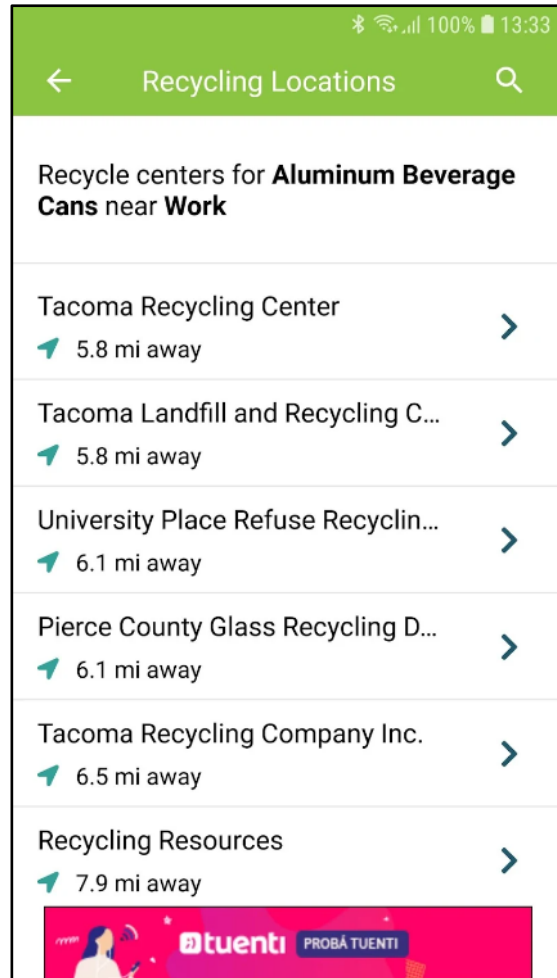


Figure 1.2 iRecycle Mobile App Preview

1.4.3 Recycle Smart Mobile App

Recycle Smart is a recycling app that simplifies the recycling process by providing information on what items belong in which recycling stream and offering a doorstep collection service for various recyclables. Users can easily schedule pickups, receive notifications, and manage their pickup schedule with just a tap. This helps users divert resources from landfills and contribute to a more sustainable environment. [7]

1.4.4 The Click Mobile App

The Click is a Belgian app that rewards users for disposing of their waste properly and picking up litter. Users can scan QR codes on waste bins to receive Circular UCoins, which can be redeemed for discounts and rewards at participating merchants. [9]

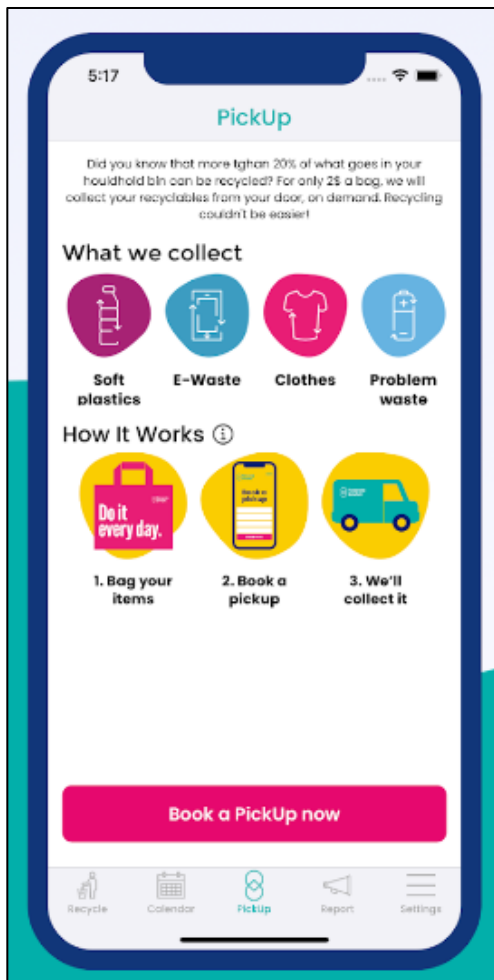


Figure 1.3 Recycle smart Mobile App Preview

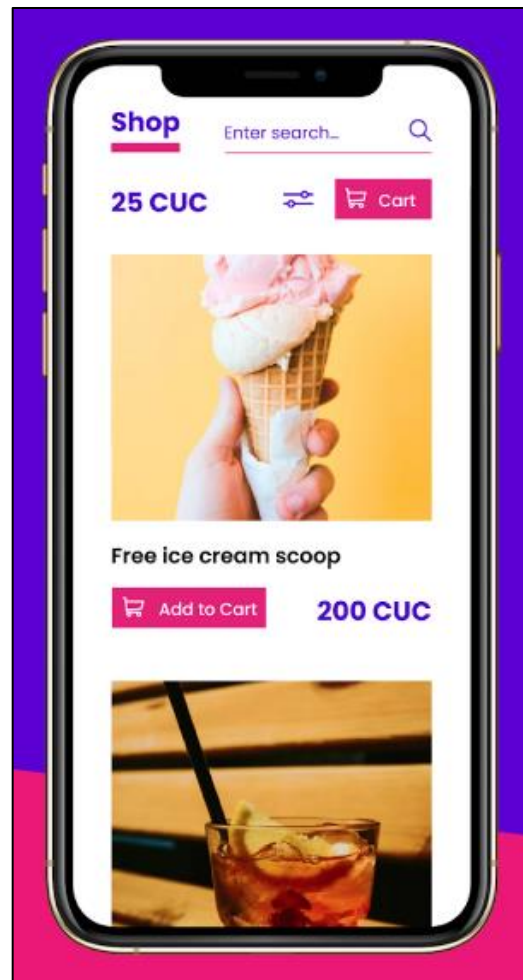


Figure 1.4 The Click Mobile App Preview

1.4.5 Zeloop Mobile App

The ZeLoop app incentivizes pro-environmental behavior by rewarding users for collecting and recycling plastic bottles. Users can join a community, deposit bottles at designated collection points, and earn Eco Rewards tokens which can be exchanged for prizes. The app also features daily actions, challenges, and a forum for sharing recycling tips and reducing environmental impact. [8]

1.4.6 Gaia Mobile App

Gaia is gamified recycling app that uses machine learning to provide users with recycling information based on the item they scan. The app would reward users for recycling correctly with points that can be redeemed for rewards. This would make recycling more engaging and rewarding, and help to increase recycling rates and reduce waste. [10]

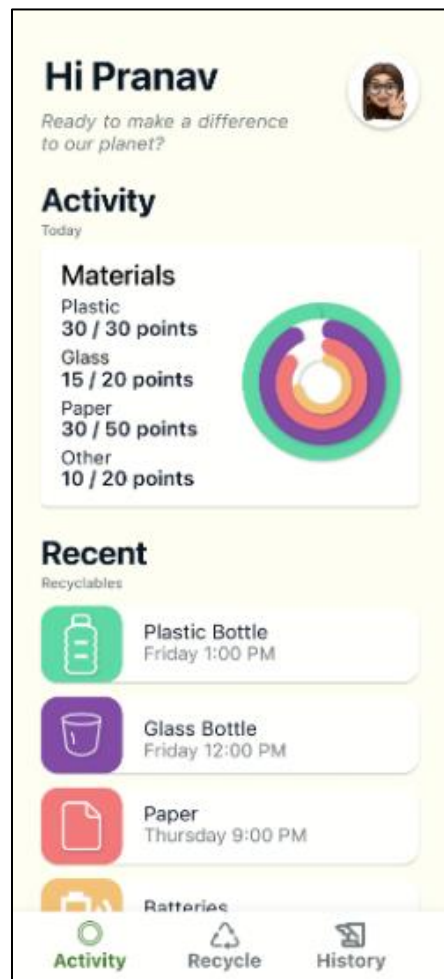


Figure 1.5 Gaia Mobile app Preview

1.5 Solution

SaafPakistan is a revolutionary mobile application designed to transform waste management and recycling practices in Pakistan. It tackles the challenges posed by unstructured waste disposal while promoting responsible recycling habits among both individuals and businesses. Here are the key features of this innovative solution:

1.5.1 Recycling Pickup Scheduling

The app empowers users to effortlessly schedule pickups for recyclable materials, making it convenient and accessible for everyone to participate in recycling.

1.5.2 Gamification Leaderboard System

To inspire a sense of competition and engagement, the app integrates a dynamic leaderboard system for both individuals and businesses. Users earn points and rankings based on their recycling activities, and real-time updates keep them motivated and involved.

1.5.3 Add Friends and Friends Leaderboard

Users can connect with others on the platform by sending and accepting friend requests. Once connected, they can view a separate "Friends Leaderboard" to compare their recycling achievements with their friends.

1.5.4 Rewards and Compensation

To encourage continued participation, users receive monetary compensation for their contributions to recycling. The app provides transparent and real-time tracking of users' earnings, keeping them informed about their achievements in terms of monetary rewards.

1.5.5 Corporate Onboarding

Companies can register to participate in the recycling program, extending the gamification aspect to the corporate sector. This feature allows businesses to showcase their environmental initiatives and commitment to sustainability through the app.

1.5.6 Advertisement as a Green Company

Businesses actively participating in the program receive recognition as eco-friendly and socially responsible partners. This heightened visibility can attract environmentally conscious consumers, potentially leading to increased customer engagement.

1.5.7 Educational Content

The app includes educational content like tips and tricks to guide users on how to recycle effectively and why it matters.

1.5.8 Dashboard Summary

Users are presented with a motivating summary on their dashboard, which encourages them to earn more points and recycle more. The summary includes details such as the total amount they have earned, the total number of waste items recycled, and the total reduction in carbon emissions, offering a clear view of their positive environmental impact.

Our aim is to develop an app that not only streamlines waste management and recycling but also raise awareness about responsible waste disposal and show the importance of sustainability practices.

2 Requirement Analysis

2.1 User Roles

SaafPakistan app involves the following user roles:

- **Customer:** A user who is an individual or corporate organization looking to recycle waste.
- **Rider:** A user who uses the system as a rider to complete recycling pickups.
- **Warehouse Manager:** A user who uses the system as a warehouse manager to oversee recycling pickups at the warehouse.
- **Admin:** A user with administrative privileges who manages and oversees the entire system.

2.2 User Story

2.2.1 Scheduling a Pickup for Recycling

As an environmentally-conscious customer, I want the ability to schedule a pickup for recycling from my desired location, so that I can conveniently and responsibly dispose of my recyclable materials.

Acceptance Criteria:

- When I access the Recycling App, I should find a user-friendly option on the main screen that allows me to schedule a pickup.
- I can select a "Schedule Pickup" button, which prompts me to enter details for the pickup request.
- I should have the option to specify the types of materials I want to recycle, such as paper, plastic, glass, or other recyclables.
- The app should prompt me to specify my pickup address, either by manually entering the address or selecting it from a list of saved addresses.
- After entering the details and confirming my pickup request, the app should provide a confirmation notification.
- If I need to cancel the pickup request, I should have the option to do so within the app.

2.2.2 Collecting Recyclable Waste from Customers' Homes

As a dedicated recycling rider, I want a smooth and efficient process for picking up recycling waste from customers' homes, ensuring that I can provide a valuable service while maintaining a positive customer experience.

Acceptance Criteria:

- When I log in to the Rider App, I should see a list of assigned pickup orders on the main screen.
- I should be able to select an order from the list to view more details, including the customer's name, address, and contact information.
- After selecting an order, the app should display a map with directions from my current location to the customer's address for easy navigation.
- I should be able to see contact info of the customer, if necessary, for example, if I have trouble locating their address.

- Once all recyclables are collected and weighed, I should be able to generate an order receipt, which lists the individual items, their exact weights, and the total estimated price for the customer's records.
- If the customer does not respond or there is any issue, the app should provide options to handle the situation such as cancelling the pickup.

2.3 Prototypes

SaafPakistan embodies a dual purpose: not only is it a tool for environmental consciousness and waste management, but it also serves as a symbol of national pride. To reflect this, we drew inspiration from Pakistan's flag, integrating its iconic green color as a central theme throughout the app's interface. Making the app easy to use for everyone was also one of our main goals. User-centric design principles drove the development process, ensuring an intuitive experience for individuals of all backgrounds.

2.3.1 Customer App Prototypes

The SaafPakistan customer app prototype features a user-friendly design and a green color scheme to promote environmental consciousness.

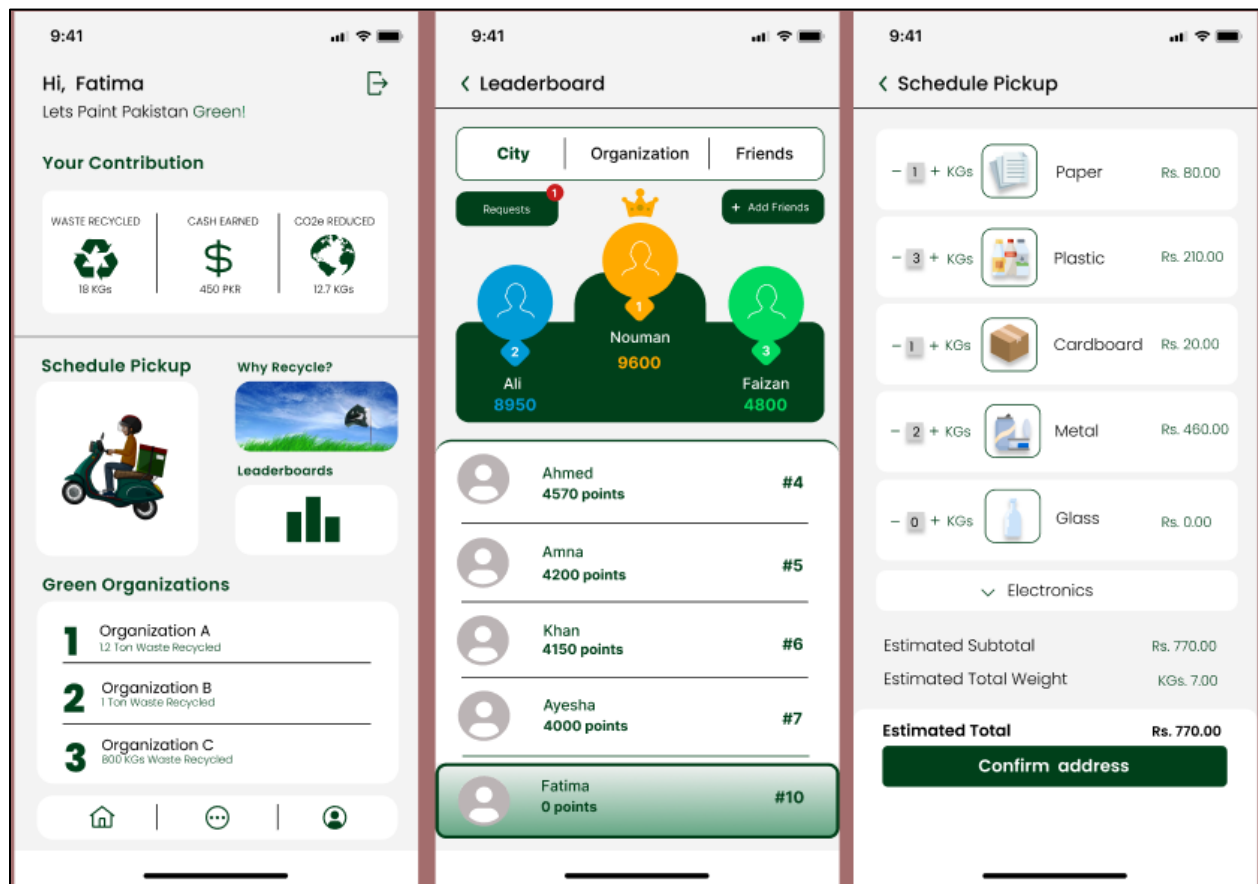


Figure 2.1 SaafPakistan Customer's App prototypes

2.3.2 Rider App Prototypes

The SaafPakistan rider app is tailored to facilitate the efficient collection of recyclable materials from customers. The prototypes emphasize functionality and ease of use to support riders in managing their pickups effectively. The app's interface is straightforward, with a focus on essential features that enhance the rider's experience.

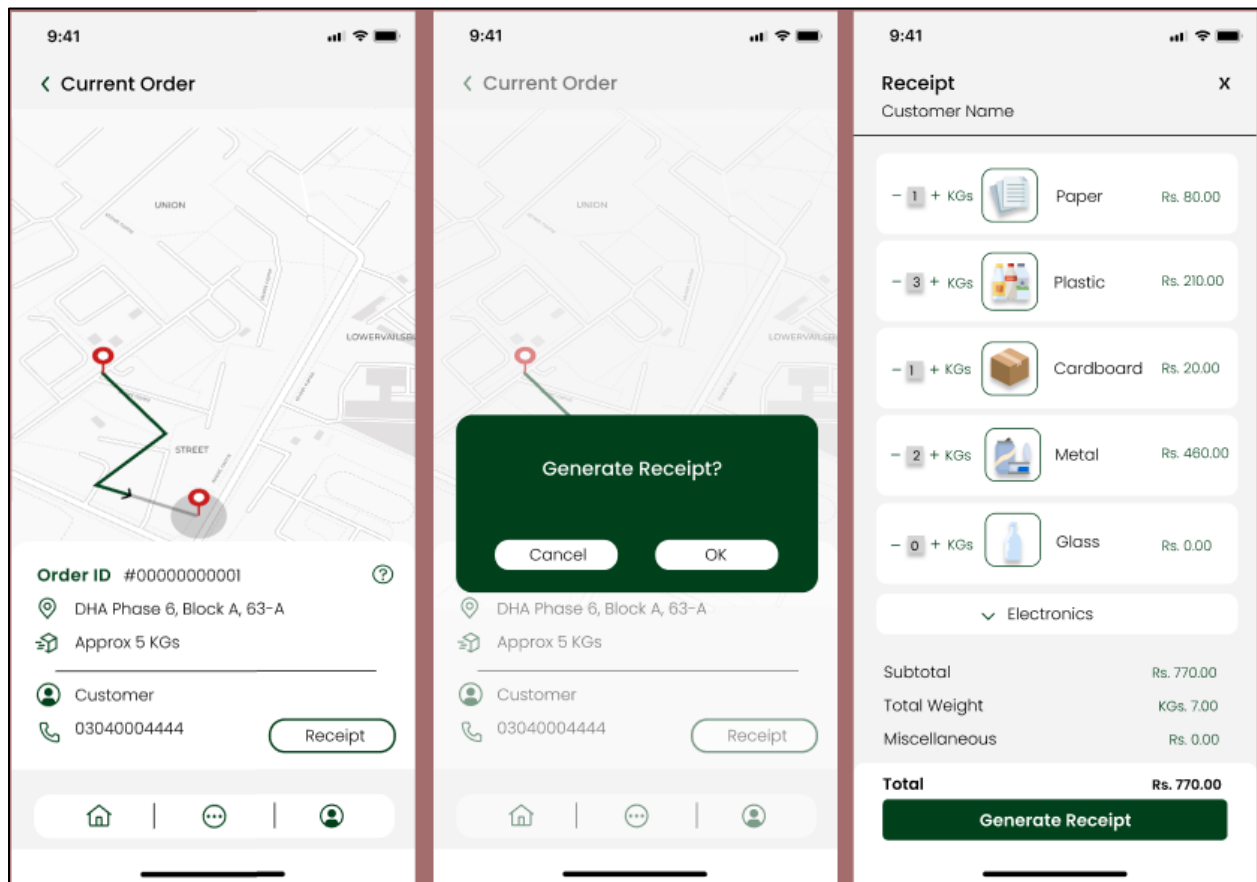


Figure 2.2 SaafPakistan Rider's App prototypes

2.4 Use Case Diagram

The following Use Case diagram helps to visualize all the Use Cases documented and helps in communicating the systems intended functionality, from the perspective of the users.

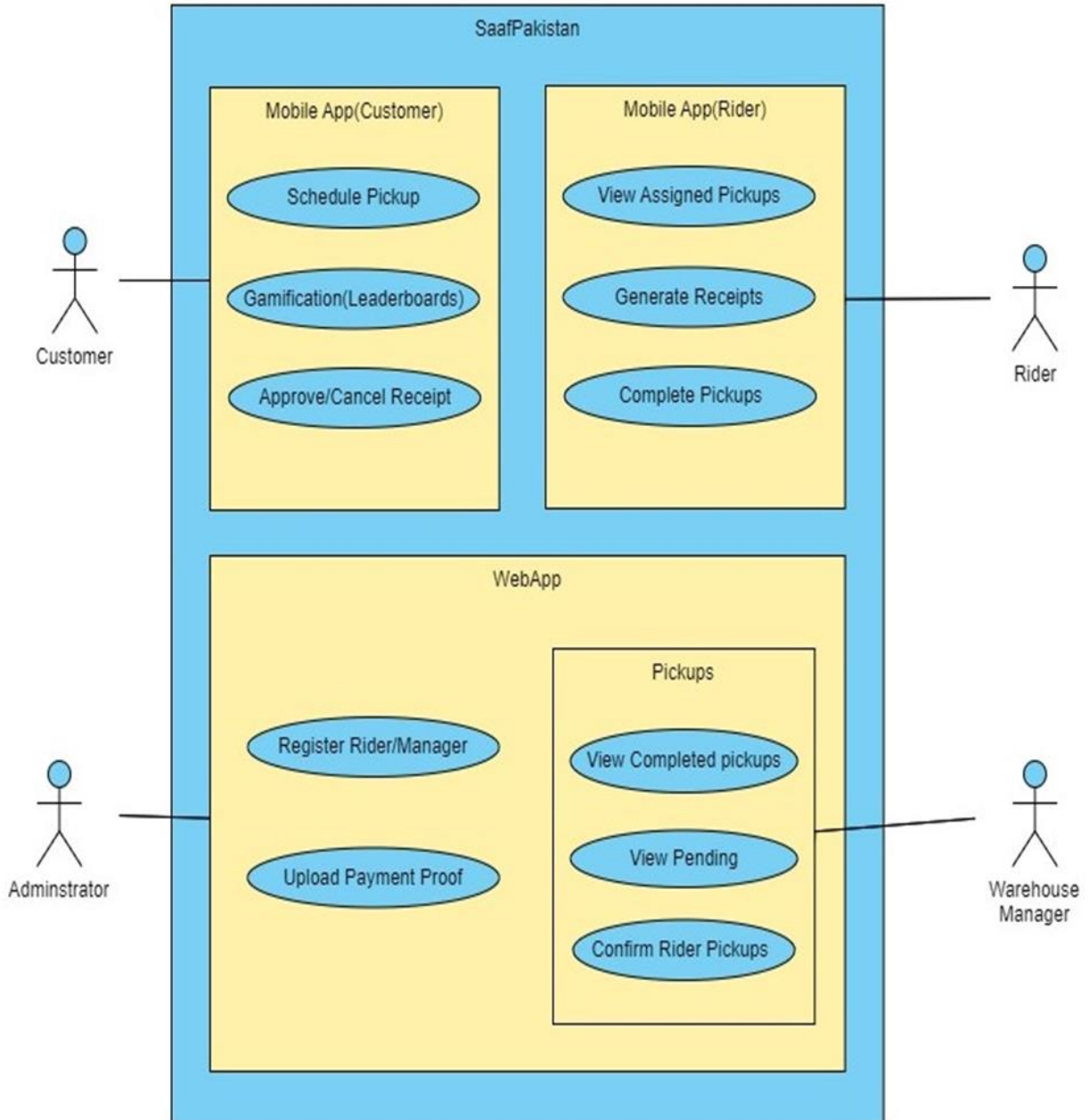


Figure 2.3 Use Case diagram

2.5 Functional Requirements Mobile App

This section lays out the functional requirements of the SaafPakistan Mobile app. The following statements lay out the basic tasks, capabilities and actions that the system is required to be able to perform to meet the needs of the user. They are broken down into two categories, Customer requirements and Rider requirements, which are specific to each individual user.

2.5.1 Customer Requirements

FR1.1: Customer Registration

Description: The system shall allow the customer to register an account.

Actors: Customer

Precondition: Customer is not registered in the system.

Postcondition: Customer is registered in the system and directed to the home page.

Main Flow:

1. Customer provides valid registration information, including name, email, phone number, and selects whether the account is personal or for a company.
2. The system validates the information and ensures it is unique.
3. User account information, including the chosen account type (personal or company), is securely stored in the database.

Alternate Flow: Registration information is invalid or not unique, and the system provides an error message.

FR1.2: Customer Sign In

Description: The system shall allow the customer to sign in to account.

Actors: Customer

Precondition: Customer is not logged in.

Postcondition: Customer is logged in and gains access to their account.

Main Flow:

1. Customer provides valid email and password.
2. The system validates the customer's credentials.
3. Upon successful validation, the customer is granted access to their account.

Alternate Flow: Customer provides invalid credentials, and the system denies access.

FR1.3: Customer Logout

Description: The system shall allow the customer to log out of their account.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: Customer is logged out of their account.

Main Flow:

1. Customer selects the logout option.
2. The system logs the customer out and redirects them to the login screen.

Alternate Flow: None

FR1.4: View Account Profile

Description: The system shall allow the customer to view their account profile.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: Customers can view their name, phone number, email, and address in their account profile.

Main Flow:

1. Customer selects the profile icon.
2. The system displays the customer's account profile information, including:
 - Name
 - Phone number
 - Email
 - Address
 - Map location

Alternate Flow: None

FR1.5: Update Profile

Description: The system shall allow the customer to update profile.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: Customers can edit their account information, including name, email, phone number, address, and password.

Main Flow:

1. Customer selects the profile icon.
2. The system displays the customer's current profile information.
3. Customer selects the "Edit Profile" option.
4. The system shall allow the customer to update their name, email, phone number, address, and password.
5. Customer makes the desired changes.
6. The system validates the changes and updates the customer's account information.

Alternate Flow 1: The customer provides invalid information, and the system displays an error message.

Alternate Flow 2: The customer decides not to make any changes and cancels the editing process.

FR1.6: Schedule Pickup for Recycling

Description: The system shall allow the customer to schedule pickup for recycling.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: The pickup is scheduled, and the recycling pickup request is recorded.

Main Flow:

1. Customer clicks the "Schedule Pickup" button.
2. Customer is presented with a list of recyclable items.

3. Customers enter the estimated weight of each recyclable item that they want to recycle.
4. The system calculates the estimated total price for the recyclables based on the recyclable items and their weights.
5. The system shows the estimated pickup time and the customer's address.
6. Customer confirms the pickup by clicking “Confirm pickup”.
7. The system records the pickup recycling pickup request and provides a confirmation to the customer.

Alternate Flow 1: The total weight is less than minimum limit set by admin, the system displays an appropriate message indicating that the minimum weight requirement for pickup is not met, and the pickup is not scheduled.

Alternate Flow 2: Customer decides not to proceed with the pickup scheduling.

FR1.7: View Current Recycling Pickup Request

Description: The system shall allow the customer to view current recycling pickup requests.

Actors: Customer

Precondition: Customer has a scheduled recycling pickup request.

Postcondition: Customer can view details of the current recycling pickup request.

Main Flow:

1. Customer selects the “View Current recycling pickup request” option.
2. The system displays details of the current recycling pickup request.

Alternate Flow: None

FR1.8: Cancel Current Recycling Pickup Request

Description: The system shall allow the customer to cancel the current recycling pickup request.

Actors: Customer

Precondition: Customer has a scheduled recycling pickup request.

Postcondition: Customer recycling pickup request is canceled.

Main Flow:

1. Customer selects the “View Current recycling pickup request” option.
2. The system displays details of the current recycling pickup request.
3. Customer clicks “Cancel recycling pickup request”.

Alternate Flow: None

FR1.9: View Pickup History

Description: The system shall allow the customer to view their recycling pickup history.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: Customers can view their pickup history.

Main Flow:

1. Customer selects the “Recycling History” option.
2. The system displays a list of past recycling pickups.

Alternate Flow: None

FR1.10: View Pending/Paid Payments

Description: The system shall allow the customer to view their pending and completed payments for recycling services. This also includes Rewards won in the leaderboards.

Precondition: Customer is logged in.

Postcondition: Customer can view payment status.

Main Flow:

1. Customer selects the “Recycling History” option.
2. The system displays a list of recycling pickup requests in the customer's history, and for each recycling pickup request, it indicates the payment status, whether it's pending or paid.

Alternate Flow: None

FR1.11: View Recycling Statistics on Home Page

Description: The system shall allow the customer to view total waste recycled by the user from the app, cash earned and carbon emission reduced on the home page.

Actors: Customer

Precondition: Customer is logged in and on the home page.

Postcondition: Customers can view total waste recycled, cash earned and carbon emission reduced.

Main Flow:

1. Customer visits the home page.
2. The system displays:
 - Amount of waste recycled
 - Cash earned through recycling
 - Carbon emissions reduced through recycling

Alternate Flow: None

FR1.12: View Top 3 Organizations on Home Page

Description: The system shall allow the customer to view top 3 organizations that recycle on the home page.

Actors: Customer

Precondition: Customer is logged in and on the home page.

Postcondition: Customers can view top recycling organizations.

Main Flow:

1. Customer visits the home page.
2. The system displays the top 3 recycling organizations.

Alternate Flow: None

FR1.13: Access Educational Content on Recycling

Description: The system shall allow the customer to access educational content like tips and tricks related to recycling on each recyclable material category.

Precondition: Customer is logged in and on the homepage.

Postcondition: Customers can access educational content.

Main Flow:

1. Customer selects the “Why recycle” option.
2. The system displays each recyclable material category.
3. Customer selects the category about which the customer wants to get information.
4. The system provides access to tips and tricks related to recycling on each recyclable material category.

Alternate Flow: None

FR1.14: View Individual Leaderboard

Description: The system shall allow the customer to view a leaderboard of individuals, this leaderboard will be based on points, with 2 points awarded for every one rupee earned by selling their recyclables through the app. Top 3 Ranks of individual leaderboards will be awarded a reward in the form of cash prize.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: Customer can view the Individuals leaderboard.

Main Flow:

1. Customer clicks the “Leaderboard” option.
2. In Leaderboard customer clicks the “Individual” option
3. The system displays a leaderboard of Individuals.

Alternate Flow: None

FR1.15: View Organization Leaderboard

Description: The system shall allow the customer to view a leaderboard of organizations, this leaderboard will be based on points, with 10 points awarded for every one rupee earned by selling their recyclables through the app. Organization leaderboards will be reset every 30 days.

Actors: Customer

Precondition: Customer is logged in.

Postcondition: Customer can view the organization leaderboard.

Main Flow:

1. Customer clicks the “Leaderboard” option.
2. Customer clicks the "Organization" option.
3. The system displays a leaderboard of recycling organizations.

Alternate Flow: None

FR1.16: View Friend's Leaderboard

Description: The system shall allow the customer to view a leaderboard of their friends' recycling efforts which is ranked on points, with 10 points awarded for every one rupee earned by selling their recyclables through the app.

Actors: Customer

Precondition: Customer is logged in and has friends on the platform.

Postcondition: Customers can view the leaderboard of their friends' recycling efforts.

Main Flow:

1. Customer clicks the "Leaderboard" option.
2. Customer clicks the "Friends" option.
3. The system displays a leaderboard of the recycling efforts of the customer's friends, providing insight into their achievements in recycling.

Alternate Flow: None

FR1.17: Add Friends

Description: The system shall allow the customer to add friends to their network.

Actors: Customer

Precondition: Customer is on the Friends leaderboard screen.

Postcondition: Customers can add friends to their network.

Main Flow:

1. Customer clicks the "Add Friends" option.
2. Customers search for a friend using their phone number.
3. Customer clicks "Send Request".
4. The system sends friend requests to the selected users.

Alternate Flow: None

FR1.18: Accept/Delete Friend Requests

Description: The system shall allow the customer to accept or delete friend requests.

Actors: Customer

Precondition: Customer is logged in and has pending friend requests.

Postcondition: Friend requests are accepted or deleted.

Main Flow:

1. Customers receive friend requests.
2. Customers can accept or delete friend requests.
3. The system updates the friend list accordingly.

Alternate Flow: None

FR1.19: Approve/Cancel Rider recycling pickup request approval

Description: The system shall allow the customer to approve or cancel the rider recycling pickup request approval.

Actors: Customer

Precondition: Customer has a completed recycling pickup request and a receipt pending approval.

Postcondition: Recycling pickup request receipt is approved or canceled.

Main Flow:

1. Customer views the recycling pickup request receipt.
2. Customers can approve or cancel the receipt.
3. The system updates the recycling pickup request status based on the customer's choice.

Alternate Flow: None

2.5.2 Rider Requirements

FR1.20: Rider Sign in

Same as FR1.2.

FR1.21: Rider Logout

Same as FR1.3.

FR1.22: View Assigned recycling pickup requests

Description: The system shall allow the rider to view a list of assigned pickup requests by the system.

Actors: Rider

Precondition: Rider is logged in.

Postcondition: Rider can view the list of assigned recycling pickup requests.

Main Flow:

1. The system displays a list of recycling pickup requests assigned by the system to the rider.

Alternate Flow: None

FR1.23: Select recycling pickup request

Description: The system shall allow the rider to select a recycling pickup request from the list of assigned recycling pickup requests.

Actors: Rider

Precondition: Rider is viewing the list of assigned recycling pickup requests.

Postcondition: Rider selects a recycling pickup request for further action.

Main Flow:

1. Rider selects a recycling pickup request from the list.
2. The system provides information about the selected recycling pickup request.

Alternate Flow: If the customer does not pick up the call for the selected recycling pickup request, the rider has the option to skip that recycling pickup request and choose the next one in the list

FR1.24: View Recycling Pickup Request Information

Description: The system shall allow the rider to view customer location, contact and estimated weight to pick about the selected recycling pickup request.

Actors: Rider

Precondition: Rider has selected a recycling pickup request from the list.

Postcondition: Rider can view recycling pickup request details.

Main Flow:

1. Rider selects a recycling pickup request from the list.
2. The system displays detailed information about the recycling pickup request, including customer location, contact and estimated weight to pick.

Alternate Flow: None

FR1.25: View Map for Navigation

Description: The system shall allow the rider to view a map to assist in reaching the customer's location.

Actors: Rider

Precondition: Rider has selected a recycling pickup request and needs navigation assistance.

Postcondition: Rider can view a map for navigation.

Main Flow:

1. Rider selects a recycling pickup request.
2. The system displays a map with directions from the rider's current location to the customer's location.

Alternate Flow: None

FR1.26: Generate recycling pickup request Receipt

Description: The system shall allow the rider to generate the recycling pickup request receipt.

Actors: Rider, Customer

Precondition: Rider has picked up the recycling pickup request and needs to create a receipt.

Postcondition: The recycling pickup request receipt is generated.

Main Flow:

1. Rider enters the exact weight of each recyclable item.
2. Rider clicks the "Generate Receipt" option.
3. The system generates the recycling pickup request receipt, which includes:
 - A detailed list of recyclable items with their exact weights.
 - The total price for these recyclables.
4. The system sends the recycling pickup request receipt to the customer's app for approval.

Alternate Flow: None

FR1.27: Rider Actions on Receipt Approval

Description: The system shall allow the rider to take actions regarding the generated recycling pickup request receipt while waiting for customer approval. The rider can choose to either wait for the customer to approve the receipt or cancel it.

Actors: Rider, Customer

Precondition: Rider has generated a recycling pickup request receipt.

Postcondition: Rider either continues to wait for customer approval or cancels the receipt.

Main Flow:

1. Rider views the generated recycling pickup request receipt.
2. Rider has the option to:
 - a) Wait for customer approval, choosing to remain in a pending state.
 - b) Cancel the receipt, which updates the recycling pickup request status to "Canceled."

Alternate Flow 1: If the customer neither approves or rejects the receipt the rider has the option to cancel the receipt, updating the recycling pickup request status to "Canceled."

FR1.28: Enter Reason for Pickup Cancellation

Description: The system shall allow the rider to enter a reason for canceling a scheduled pickup.

Actors: Rider

Precondition: Rider is viewing the list of assigned recycling pickup requests and chooses to cancel a scheduled pickup.

Postcondition: The reason for the pickup cancellation is recorded.

Main Flow:

1. Rider selects an assigned recycling pickup request from the list.
2. The system displays detailed information about the selected recycling pickup request, including customer location and pickup details.
3. Rider decides to cancel the pickup and clicks the "Cancel Pickup" option.
4. The system prompts the rider to provide a reason for the cancellation.
5. Rider enters a text description of the reason.
6. Rider submits the reason.
7. The system records the provided reason for the canceled pickup.

Alternate Flow: If the rider changes their mind and decides not to cancel the pickup, they can go back to the recycling pickup request details without providing a reason and continue with the pickup.

2.6 Functional Requirements Web App

This section lays out the functional requirements of the SaafPakistan Web app. The following statements lay out the basic tasks, capabilities and actions that the system is required to be able to perform to meet the needs of the user. They are broken down into two categories, Warehouse manager requirements and Admin requirements, which are specific to each individual user.

2.6.1 Warehouse Manager Requirements

FR2.1: Warehouse Manager Sign In

Same as FR1.2.

FR2.2: Warehouse Manager Logout

Same as FR1.3.

FR2.3: View Pending Pickups

Description: The system shall allow the warehouse manager to view a list of pending pickups.

Actors: Warehouse Manager

Precondition: Warehouse manager is logged in.

Postcondition: Warehouse manager can view the list of pending pickups.

Main Flow:

1. The system displays a list of pending pickups, including details such as recycling pickup request information and rider details.

Alternate Flow: None

FR2.4: View Completed Pickups

Description: The system shall allow the warehouse manager to view a list of completed pickups.

Actors: Warehouse Manager

Precondition: Warehouse manager is logged in.

Postcondition: Warehouse manager can view the list of completed pickups.

Main Flow:

1. The system displays a list of completed pickups, including details such as recycling pickup request information, rider details, and pickup confirmation.

Alternate Flow: None

FR2.5: Confirm Completed Shift by Rider

Description: The system shall allow the admin to manage and confirm the completion of a rider's shift, including verifying that the total weight of recycling items picked up by the rider matches the total weight of recyclables delivered to the warehouse.

Actors: Admin, Rider

Precondition: Admin has access to the list of riders and a rider has returned to the warehouse after completing their pickups.

Postcondition: Admin confirms the completion of the rider's shift and verifies the accuracy of the total weight of recycling items.

Main Flow:

1. Admin accesses the list of riders and selects a specific rider who has returned to the warehouse after completing their pickups.

2. The system displays details of the selected rider's shift, including the list of completed pickups and the total weight of recyclables picked up by the rider.
3. Admin checks the accuracy of the total weight of recycling items picked up by the rider.
4. If the total weight of recycling items picked up by the rider matches the total weight of recyclables delivered to the warehouse, the admin confirms the rider's shift as "Completed and Verified."

Alternate Flow:

1. If the total weight of recycling items picked up by the rider does not match the total weight of recyclables delivered to the warehouse:
2. Admin marks the shift as "Completed with Discrepancies."
3. The system notifies the rider and warehouse manager about the discrepancy for resolution.
4. The resolution may involve rechecking the recyclables, addressing any potential discrepancies or issues in the collection process, and taking appropriate actions to ensure accurate records.
5. Once resolved, the system updates the shift status accordingly.

FR2.6: View Canceled Pickups

Description: The system shall allow the warehouse manager to view canceled pickups.

Actors: Warehouse Manager

Precondition: Warehouse manager is viewing a pending pickup.

Postcondition: Warehouse manager cancels the pending pickup.

Main Flow:

1. The system displays a list of canceled pickups, including details such as recycling pickup request information, rider details, and reason of cancellation.

Alternate Flow: None

2.6.2 Admin Requirements

FR2.7: Admin Sign In

Same as FR1.2.

FR2.8: Admin Logout

Same as FR1.3.

FR2.9: Register Warehouse Manager

Description: The system shall allow the admin to register a new warehouse manager.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: A new warehouse manager is registered in the system.

Main Flow:

1. Admin selects the "Register Warehouse Manager" option.
2. Admin provides the required information for the new warehouse manager, including name, contact details, and username.
3. The system validates the information.
4. Upon successful validation, the new warehouse manager account is created.

Alternate Flow: Information provided is invalid, and the system displays an error message.

FR2.10: Register New Rider

Description: The system shall allow the admin to register a new rider.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: A new rider is registered in the system.

Main Flow:

1. Admin selects the "Register New Rider" option.
2. Admin provides the required information for the new rider, including name, contact details, and username.
3. The system validates the information.
4. Upon successful validation, the new rider account is created.

Alternate Flow: Information provided is invalid, and the system displays an error message.

FR2.11: View Rider Details

Description: The system shall allow the admin to view details of a rider.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Admin can view rider details.

Main Flow:

1. Admin clicks the "Riders" option.
2. Admin specifies the rider for whom they want to view details.
3. The system displays the rider's information, including name, contact details, and performance statistics.

Alternate Flow: Rider details are not found, and the system provides an appropriate message.

FR2.12: View Pending Pickups

Description: The system shall allow the admin to view a list of pending pickups.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Admin can view a list of pending pickups.

Main Flow:

1. Admin selects the "Pending Pickups" option.
2. The system displays a list of pickups that are pending, along with relevant details.

Alternate Flow: No pending pickups are found, and the system provides an appropriate message.

FR2.13: View Completed Pickups

Description: The system shall allow the admin to view a list of completed pickups.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Admin can view a list of completed pickups.

Main Flow:

1. Admin selects the "Completed Pickups" option.
2. The system displays a list of completed pickups, including details such as rider, customer, and materials collected.

Alternate Flow: No completed pickups are found, and the system provides an appropriate message.

FR2.14: View Customer Details

Description: The system shall allow the admin to view details of a customer.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Admin can view customer details.

Main Flow:

1. Admin clicks the "Customers" option.
2. Admin specifies the customer for whom they want to view details.
3. The system displays the customer's information, including name, contact details, and recycling history.

Alternate Flow: Customer details are not found, and the system provides an appropriate message.

FR2.15: View Pending Payments

Description: The system shall allow the admin to view a list of pending payments.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Admin can view a list of pending payments.

Main Flow:

1. Admin selects the "Pending Payments" option.
2. The system displays a list of payments that are pending, along with relevant details.

Alternate Flow: No pending payments are found, and the system provides an appropriate message.

FR2.16: Upload Payment Proof

Description: The system shall allow the admin to upload payment proof for completed payments.

Actors: Admin

Precondition: Admin is logged in and a payment is marked as completed.

Postcondition: Payment proof is uploaded for the completed payment.

Main Flow:

1. Admin selects a completed payment.

2. Admin uploads payment proof.
3. The system associates the payment proof with the completed payment.

Alternate Flow: None

FR2.17: View Inventory

Description: The system shall allow the admin to view the inventory of recyclable materials.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Admin can view the current inventory.

Main Flow:

1. Admin clicks the "Inventory" option.
2. The system displays the current inventory of recyclable materials, including quantities and types.

Alternate Flow: None

FR2.18: Update Rates of Recycling Materials

Description: The system shall allow the admin to update the rates for recycling materials.

Actors: Admin

Precondition: Admin is logged in.

Postcondition: Rates for recycling materials are updated.

Main Flow:

1. Admin selects the "Update Rates" option.
2. Admin updates the rates for various recyclable materials.
3. The system saves the updated rates.

Alternate Flow: None

2.7 Non-Functional Requirements

The requirements listed below define how the system should perform and specify the criteria the application must meet in terms of Security, Performance, Availability and UX.

1. Security Requirements:

- **Authentication:** The system must implement a secure authentication mechanism that includes multi-factor authentication (MFA) to ensure that only authorized users can access the application.
- **Data Encryption:** All sensitive user data, including login credentials, payment information, and personal details, must be encrypted during transmission and storage using industry-standard encryption algorithms (e.g., SSL/TLS).
- **Authorization:** The system should have fine-grained role-based access control to restrict users' access to specific features and data based on their roles.

2. Performance Requirements:

- **Response Time:** The system should respond to user actions and requests within a maximum response time of [specify in milliseconds/seconds] under normal operating conditions.
- **Scalability:** The application should be capable of handling a scalable number of concurrent users to accommodate growth without significant performance degradation.

3. Availability Requirements

- The application shall be available 24/7, throughout the year (except scheduled downtime).

4. User Experience (Usability):

- **User Interface (UI) Design:** The application should follow modern design principles and provide a user-friendly and intuitive interface for all actors involved.

3 System Design & Architecture

This section provides an overview of the system's architecture, including high-level and low-level components, database schema, and sequence diagrams illustrating key functionalities such as scheduling pickups and completing orders. It reflects the iterative nature of the development process, with updates made after each implementation iteration to ensure alignment with project goals and requirements.

3.1 Technologies

SaafPakistan employs a combination of modern technologies to deliver a robust and efficient platform for both web and mobile applications. The chosen technologies are as follows:

3.1.1 WebApp Technology

React: React is a popular JavaScript library for building user interfaces, particularly well-suited for single-page applications. It offers a component-based architecture, making it easy to develop and maintain complex UIs. React's rich ecosystem of tools and libraries enhances development efficiency. For SaafPakistan, React ensures a smooth and interactive user experience for features like scheduling pickups, viewing recycling statistics, and managing account details.

Firebase: Firebase provides a comprehensive suite of backend services such as real-time databases, authentication, and hosting. Its seamless integration with React enables rapid development and deployment. Firebase's real-time capabilities ensure that updates to recycling pickups, statistics, and other dynamic data are instantly reflected across the SaafPakistan web application, enhancing user engagement and responsiveness.

3.1.2 MobileApp Technology

Flutter: Flutter is Google's UI toolkit for crafting natively compiled applications for mobile from a single codebase. It allows for fast development with expressive and flexible UI components. Flutter's hot-reload feature significantly speeds up the development process, and its rich set of pre-designed widgets makes it easy to create visually appealing and responsive applications. For SaafPakistan, Flutter ensures a consistent and high-performance experience across both Android and iOS devices, crucial for features like viewing and managing recycling pickups.

Firebase: Firebase serves as a versatile backend for the mobile application as well. It provides real-time data synchronization, authentication, and analytics, which are crucial for maintaining consistent and secure user experiences across both web and mobile platforms. The integration with Flutter is straightforward, enabling developers to focus on building features rather than managing backend infrastructure. For SaafPakistan, this means a reliable and scalable backend that supports the project's mission of promoting environmental sustainability through efficient recycling management.

3.2 System Context Diagram

This is the System Context Diagram of the system. It focuses on people and software systems and only shows the system landscape. There are four actors and two software systems.

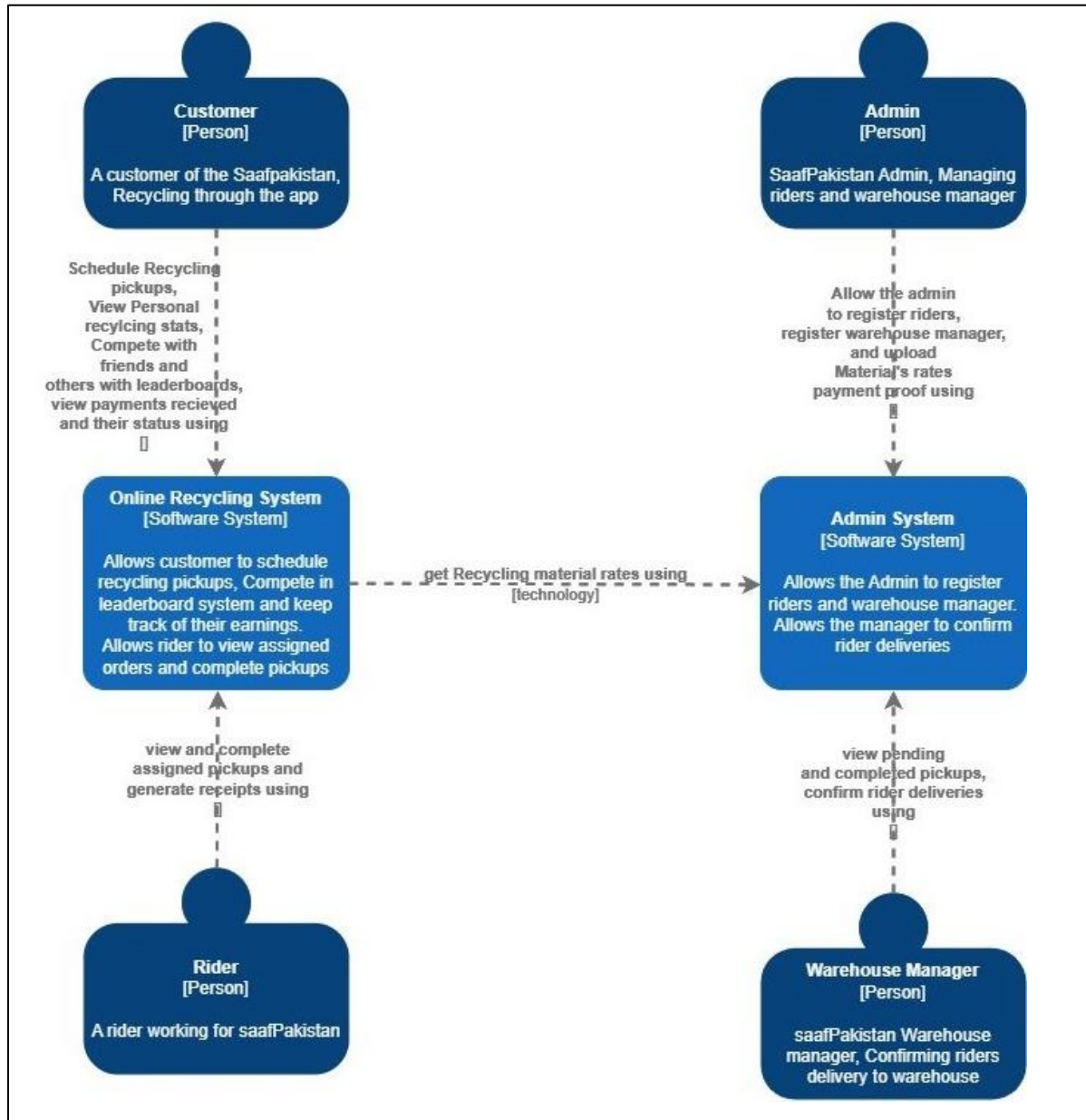


Figure 3.1 System Context diagram

3.3 High-level Architecture

This is the high-level architecture of the system. It shows major technology choices and how the software systems interact with each other. There are four actors and three major components of the system, a MobileApp (Flutter), a WebApp (React) and the database (Firebase Firestore).

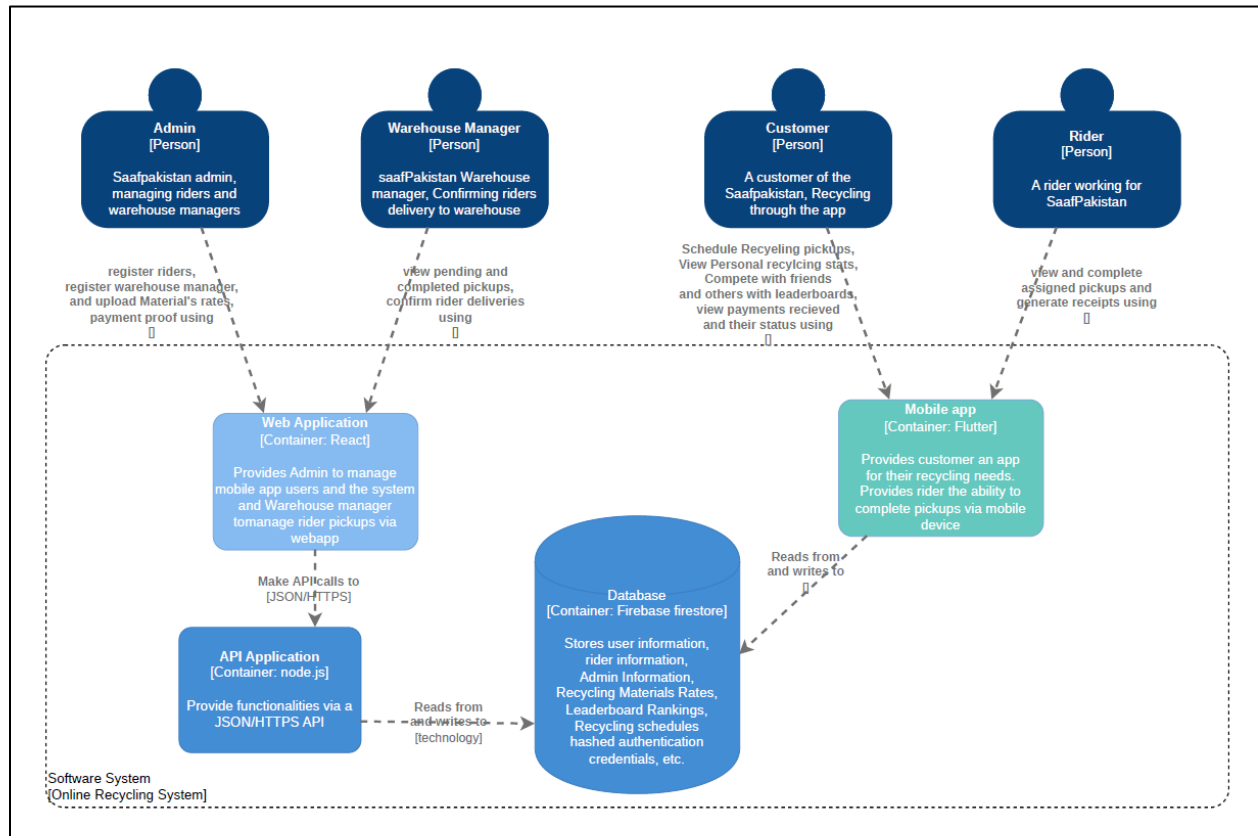


Figure 3.2 High-level Architecture

3.4 Low-Level Architecture

3.4.1 Mobile App Component Diagram

This is the component diagram of the SaafPakistan mobile app. It delves into the finer details of the system's low-level architecture. It provides a closer look at the key components within the mobile container, along with the underlying technologies and implementation specifics. Developed on Flutter, the app ensures consistent performance, leveraging Flutter's widget-based architecture for reusable UI components and Firebase for real-time data handling. Components such as Customer/Rider SignIn, Pickup, SchedulePickup, Friends, Leaderboards, and Maps are seamlessly integrated with Flutter, utilizing state management solutions and Firestore for various functionalities like authentication, scheduling, and data management. The database, powered by Firestore, stores essential app data while TPL Maps LBS provides location-based services. This architecture facilitates a high-quality user experience, secure authentication, and efficient development process.

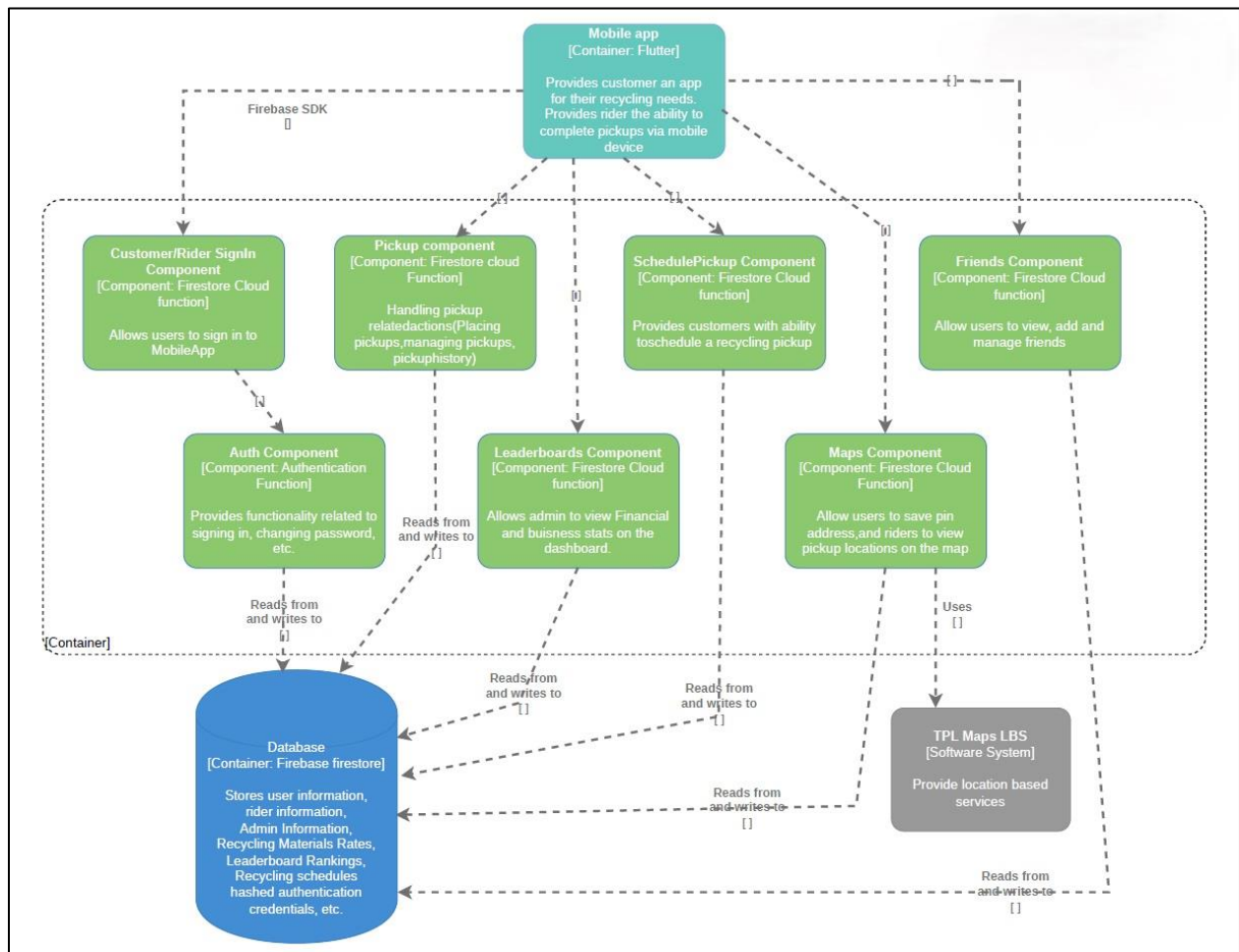


Figure 3.3 Mobile App Component Diagram

3.4.2 Web App Component Diagram

The component diagram of the SaafPakistan web application presents an in-depth exploration of the system's low-level architecture. It outlines the primary components residing within the web container, shedding light on the underlying technologies and implementation intricacies. The web application is developed using the React framework, which offers a modular and efficient approach to building user interfaces. The integration with Firebase, specifically Firestore for database operations and Firebase Authentication for secure sign-in processes, ensures that the application can handle real-time data synchronization and user management efficiently. By using React in combination with Firebase, the SaafPakistan web app benefits from a streamlined development process, where components can be developed, tested, and deployed independently, while still maintaining seamless communication with the backend services.

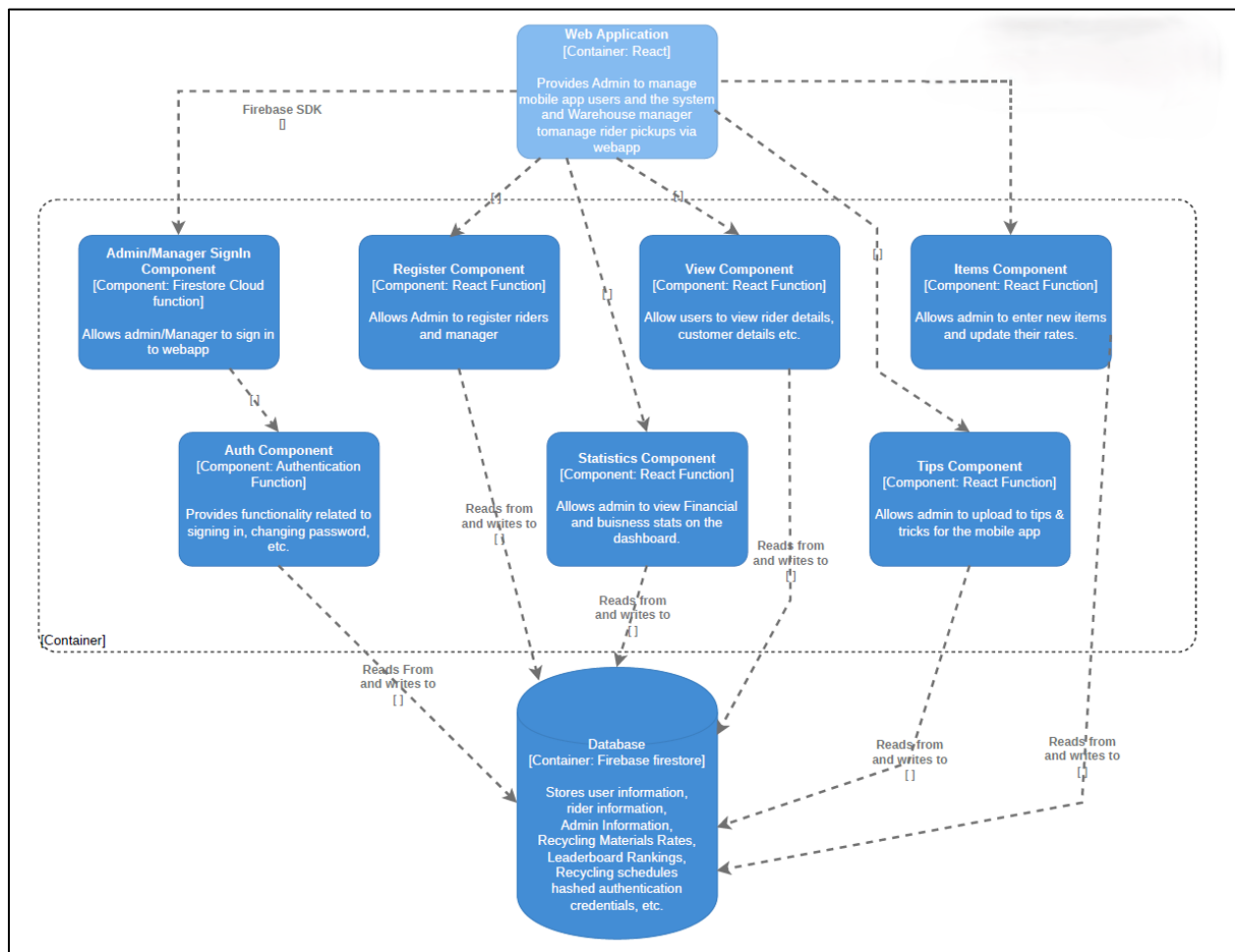


Figure 3.4 Web App Component Diagram

3.5 Database Schema

SaafPakistan is developed using Firebase Firestore as the database for the application. Firestore is a NoSQL serverless database with real-time notification capability, and together with the Firebase ecosystem it greatly simplifies common app development challenges while letting the application developer focus primarily on their business logic and user experience.

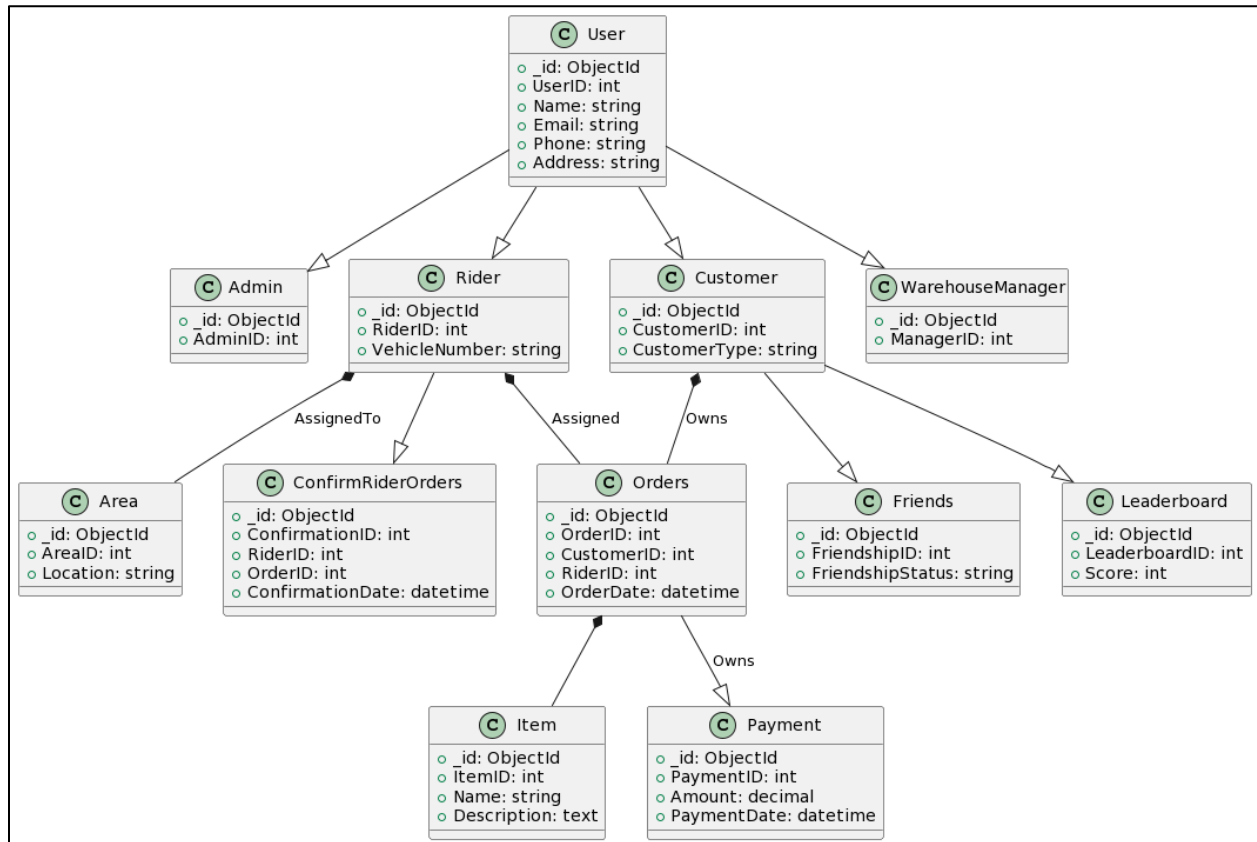


Figure 3.5 Schema Diagram

3.6 Sequence Diagram

Our project, which follows a functional programming paradigm, sequence diagrams diverge from the conventional use of objects and instead represent interactions through App components/pages. This approach aligns with the functional nature of our codebase, emphasizing the flow of data and operations within the distinct functional units rather than traditional object-oriented entities. By utilizing App components/pages in sequence diagrams, we tailor the representation to better reflect the functional architecture of our project.

Following are the sequence diagrams for the key (major) requirements.

3.6.1 Schedule Pickup (FR 1.6)

This is a sequence diagram of how a customer will schedule a pickup. It highlights how the customer navigates the screens, selects recyclables that he wants to recycle, and confirms the recycling pickup request.

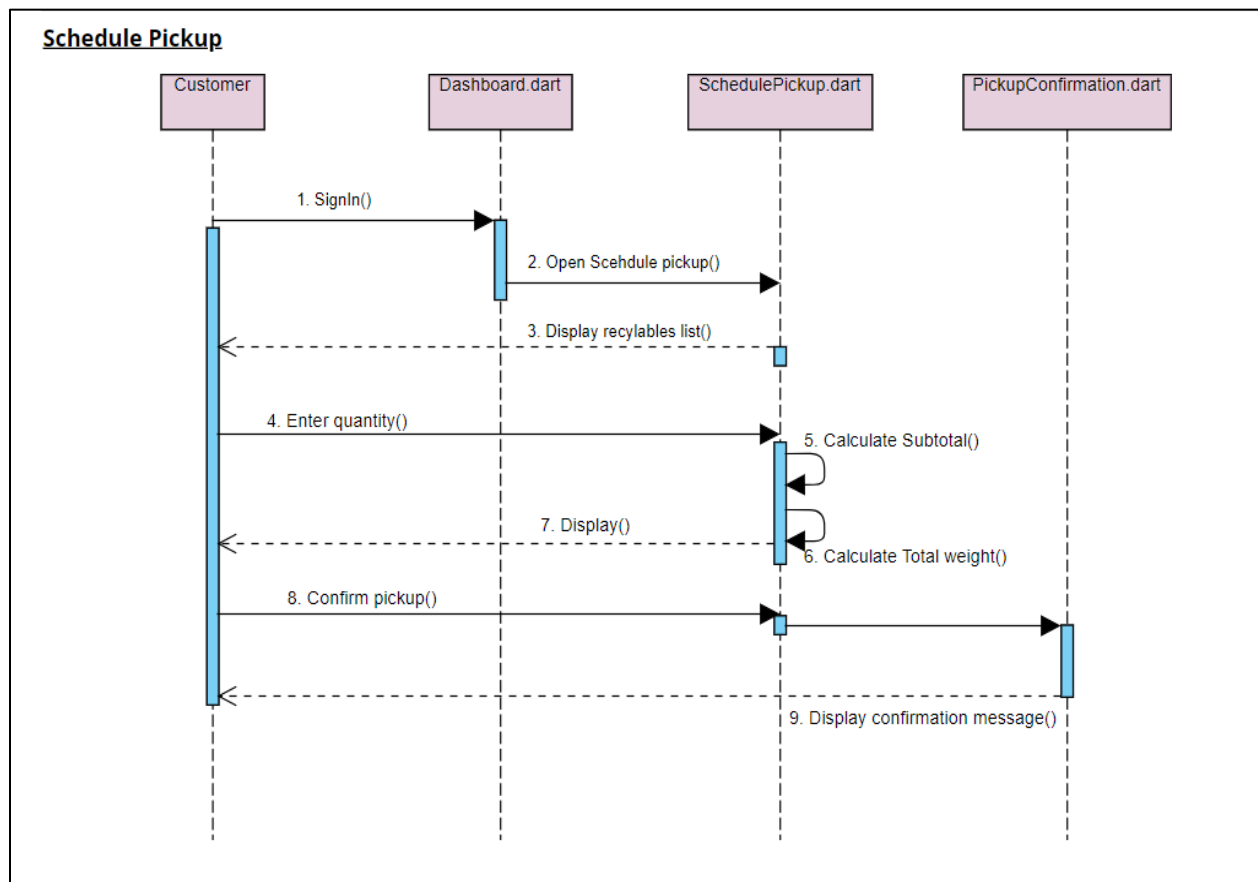


Figure 3.6 Sequence diagram: Schedule Pickup

3.6.2 Complete Pickup (FR 1.22-1.26)

This is a sequence diagram of how a rider will complete a pickup. It shows how the rider will select an order, complete pickup and generate receipt.

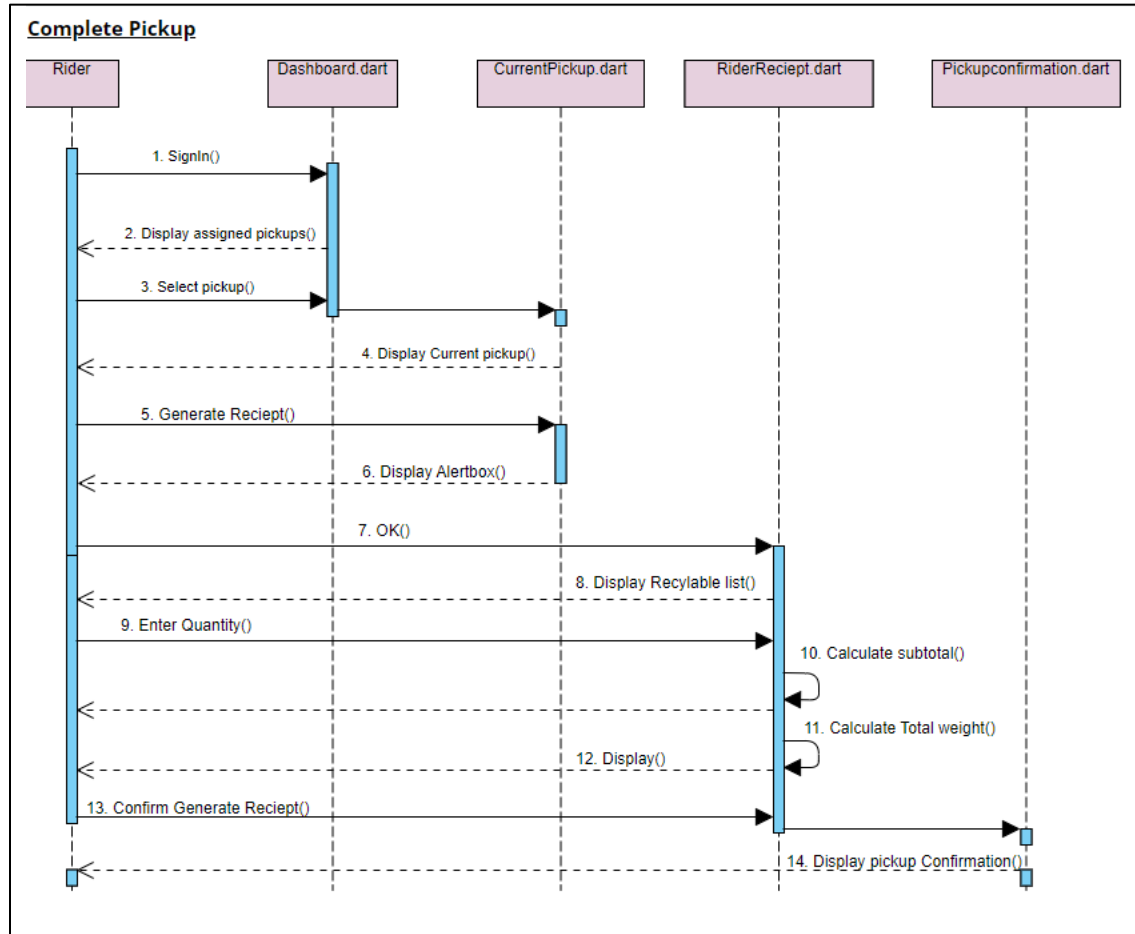


Figure 3.7 Sequence Diagram: Complete Pickup

4 Implementation

SaafPakistan's WebApp development is centered around the utilization of React, a powerful and versatile JavaScript library designed for building dynamic and interactive user interfaces. With React, we embrace a functional programming paradigm, emphasizing the creation of modular components that seamlessly integrate to form comprehensive views.

Moreover, for the mobile application development, SaafPakistan employs Flutter in conjunction with Firebase. Flutter provides a robust framework for building natively compiled applications for mobile from a single codebase. By harnessing the capabilities of Firebase, both the web and mobile applications benefit from real-time data synchronization, authentication, and cloud storage, ensuring a seamless and responsive user experience across platforms.

4.1 Test Cases

Test Case numbers correspond to the functional requirement numbers.

4.1.1 Test Cases Mobile App (FR 1.1 – FR 1.28)

4.1.1.1 Customer Module (FR 1.1 – FR 1.19)

Test Case 1.1: Customer Registration

Test Scenario: A new customer registers in the system.

Preconditions: The customer is not registered in the system.

Test Steps:

1. Open the SaafPakistan app.
2. Tap on the "Register" button.
3. Enter valid registration information, including name, email, phone number, and select "Personal" or "Company" account.
4. Enter the full address and select the main area and sub area.
5. Confirm the registration.

Expected Result: The customer is registered, and the system directs them to the home page. An email confirmation is sent to the customer's registered email address.

Alternate Flow 1: If the email address is already registered:

o **Test Steps:**

- In step 3, enter an email address that is already registered in the system.

o **Expected Result:** The system displays an error message indicating that the email is already in use, and the customer is prompted to use a different email.

Alternate Flow 2: If the phone number is already registered:

o **Test Steps:**

- In step 3, enter a phone number that is already registered in the system.

o **Expected Result:** The system displays an error message indicating that the phone number is already in use, and the customer is prompted to use a different phone number.

Test Case 1.2: Customer Sign In

Test Scenario: A registered customer logs into the system.

Preconditions: The customer is not logged in.

Test Steps:

1. Open the SaafPakistan app.
2. Tap on the "Login" button.
3. Enter a valid phone number and password.
4. Tap the "Login" button.

Expected Result: The customer is logged in and gains access to their account.

Alternate Flow 1: If the entered email or password is incorrect:

o **Test Steps:**

- In step 3, enter an incorrect phone number or password.

o **Expected Result:** The system displays an error message indicating that the login credentials are incorrect, and the customer is prompted to enter valid credentials.

Test Case 1.3: Customer Logout

Test Scenario: The customer logs out of their account.

Preconditions: The customer is logged in.

Test Steps:

1. Select the logout option.

Expected Result: The customer is logged out of their account, and the system redirects them to the login screen.

Alternate Flow: None

Test Case 1.4: View Account Profile

Test Scenario: The customer views their account profile.

Preconditions: The customer is logged in.

Test Steps:

1. Select the profile icon.

Expected Result: The system displays the customer's account profile information, including their name, phone number, email, and address.

Alternate Flow: None

Test Case 1.5: Update Credentials

Test Scenario: The customer updates their account information.

Preconditions: The customer is logged in.

Test Steps:

1. Select the profile icon.
2. The system displays the customer's current profile information.
3. Select the "Edit Profile" option.
4. The system shall allow the customer to update their name, email, phone number, and password.
5. The customer makes the desired changes.

6. The system validates the changes and updates the customer's account information.

Expected Result: The customer's account information is updated with the changes made by the customer. If any errors occur during the process, the system displays appropriate error messages.

Test Case 1.6: Schedule Pickups for Recycling

Test Scenario: The customer schedules a pickup for recycling.

Preconditions: The customer is logged in.

Test Steps:

1. Click the "Schedule Pickup" button.
2. Provide estimated weights for recyclable items.
3. Confirm the pickup.

Expected Result: The pickup is scheduled, and the system records the recycling pickup request. The customer receives a confirmation notification.

Alternate Flow 1: The total weight is less than 5 kg

- o **Expected Result:** The system displays an appropriate message indicating that the minimum weight requirement for pickup is not met, and the pickup is not scheduled.

Test Case 1.7: View Current recycling pickup request

Test Scenario: The customer views their current recycling pickup request.

Preconditions: The customer has a scheduled recycling pickup request.

Test Steps:

1. Select the "Current recycling pickup request" option.

Expected Result: The system displays details of the current recycling pickup request.

Alternate Flow: None

Test Case 1.8: Cancel Current recycling pickup request

Test Scenario: The customer cancels their current recycling pickup request.

Preconditions:

- The customer is logged into the system.
- The customer has a scheduled recycling pickup request.

Test Steps:

1. Select the "Current recycling pickup request" option.
2. Click on the "Cancel" button.
3. Confirm the cancellation by providing a reason (if required) and clicking the "Confirm Cancellation" button.

Expected Result: The system displays a confirmation message indicating that the current recycling pickup request has been canceled successfully.

Alternate Flow: None

Test Case 1.9: View Pickup History

Test Scenario: The customer views their recycling pickup history.

Preconditions: The customer is logged in.

Test Steps:

1. Select the “Recycling History” option.

Expected Result: The system displays a list of past recycling pickups in the customer's history.

Alternate Flow: None

Test Case 1.10: View Pending/Paid Payments

Test Scenario: The customer views their pending and paid payments for recycling services.

Preconditions: The customer is logged in.

Test Steps:

1. Select the “Recycling History” option.

Expected Result: The system displays a list of recycling pickup requests in the customer's history and indicates the payment status (pending or paid) for each recycling pickup request.

Alternate Flow: None

Test Case 1.11: View Recycling Statistics on Home Page

Test Scenario: The customer views recycling statistics on the home page.

Preconditions: The customer is logged in and on the home page.

Test Steps:

1. Visit the home page.

Expected Result: The system displays recycling statistics, including the amount of waste recycled, cash earned through recycling, and carbon emissions reduced through recycling.

Alternate Flow: None

Test Case 1.12: View Top 3 Organizations on Home Page

Test Scenario: The customer views the top 3 organizations that recycle on the home page.

Preconditions: The customer is logged in and on the home page.

Test Steps:

1. Visit the home page.

Expected Result: The system displays the top 3 recycling organizations.

Alternate Flow: None

Test Case 1.13: Access Educational Content on Recycling

Test Scenario: The customer accesses educational content related to recycling.

Preconditions: The customer is logged in.

Test Steps:

1. Select the “Why recycle” option.

Expected Result: The system provides access to articles and resources related to recycling.

Alternate Flow: None

Test Case 1.14: View Individuals Leaderboard

Test Scenario: The customer views a leaderboard of waste recycled by individuals in the Individuals.

Preconditions: The customer is logged in.

Test Steps:

1. Click the “Leaderboard” option.
2. In the Leaderboard customer clicks the “Individuals” option.

Expected Result: The system displays a leaderboard of Individuals.

Alternate Flow: None

Test Case 1.15: View Organization Leaderboard

Test Scenario: The customer views a leaderboard of recycling organizations based on various criteria.

Preconditions: The customer is logged in.

Test Steps:

1. Click the “Leaderboard” option.
2. Click the "Organization" option.

Expected Result: The system displays a leaderboard of recycling organizations.

Alternate Flow: None

Test Case 1.16: View Friend's Leaderboard

Test Scenario: The customer views a leaderboard of their friends' recycling efforts.

Preconditions: The customer is logged in and has friends on the platform.

Test Steps:

1. Click the “Leaderboard” option.
2. Click the "Friends" option.

Expected Result: The system displays a leaderboard of the recycling efforts of the customer's friends, providing insight into their achievements in recycling.

Alternate Flow: None

Test Case 1.17: Add Friends

Test Scenario: The customer adds friends to their network.

Preconditions: The customer is on the Friends leaderboard screen.

Test Steps:

1. Click the “Add Friends” option.
2. Search for friends using phone numbers.
3. Click “Send Request”.
4. The system sends friend requests to the selected users.

Expected Result: Friend requests are sent to the selected users.

Alternate Flow: None

Test Case 1.18: Accept/Delete Friend Requests

Test Scenario: The customer accepts or deletes friend requests.

Preconditions: The customer is logged in and has pending friend requests.

Test Steps:

1. The customer receives friend requests.
2. The customer can accept or delete friend requests.

Expected Result: The system updates the friend list accordingly.

Alternate Flow: None

Test Case 1.19: Approve/Cancel recycling pickup request Receipt

Test Scenario: The customer approves or cancels the recycling pickup request receipt.

Preconditions: The customer has a completed recycling pickup request and a receipt pending approval.

Test Steps:

1. The customer views the recycling pickup request receipt.
2. The customer can approve or cancel the receipt.

Expected Result: The system updates the recycling pickup request status based on the customer's choice.

Alternate Flow: None

4.1.1.2 Rider Module (FR 1.20 – FR 1.28)

Test Case 1.20: Rider Sign in

Test Scenario: The rider attempts to sign in.

Preconditions: Rider is not logged in.

Test Steps:

1. Open the SaafPakistan app.
2. Enter a valid email and password.
3. Tap the "Sign In" button.

Expected Result: The rider is successfully signed in and gains access to their account.

Alternate Flow 1: If the rider provides an incorrect email or password:

Test Steps:

1. In step 2, enter an incorrect email or password.
2. Tap the "Sign In" button.

Expected Result: The system displays an error message indicating that the login credentials are incorrect, and access is denied.

Test Case 1.21: Rider Logout

Test Scenario: The rider logs out of their account.

Preconditions: Rider is logged in.

Test Steps:

1. Tap the "Logout" option in the app.

Expected Result: The rider is successfully logged out of their account, and the app redirects them to the login screen.

Alternate Flow: None

Test Case 1.22: View Assigned recycling pickup requests

Test Scenario: The rider views a list of recycling pickup requests assigned to them.

Preconditions: Rider is logged in.

Test Steps:

1. Open the app.

Expected Result: The app displays a list of recycling pickup requests assigned to the rider, recycling pickup requested by proximity to the rider's current location.

Alternate Flow: None

Test Case 1.23: Select recycling pickup request

Test Scenario: The rider selects a recycling pickup request from the list of assigned recycling pickup requests.

Preconditions: Rider is viewing the list of assigned recycling pickup requests.

Test Steps:

1. Tap on a specific recycling pickup request in the list.

Expected Result: The app provides detailed information about the selected recycling pickup request.

Alternate Flow 1: If the customer does not answer the phone for the selected recycling pickup request, the rider has the option to skip that recycling pickup request and proceed to the next one in the list.

Test Case 1.24: View Detailed recycling pickup request Information

Test Scenario: The rider views detailed information about the selected recycling pickup request.

Preconditions: Rider has selected a recycling pickup request from the list.

Test Steps:

1. Tap on a selected recycling pickup request.

Expected Result: The app displays detailed information about the recycling pickup request, including the customer's location, contact details, and estimated weight to pick up.

Alternate Flow: None

Test Case 1.25: View Map for Navigation

Test Scenario: The rider views a map to assist in reaching the customer's location.

Preconditions: Rider has selected a recycling pickup request and needs navigation assistance.

Test Steps:

1. Tap on a selected recycling pickup request.

Expected Result: The app displays a map with directions from the rider's current location to the customer's location, assisting the rider in reaching the destination.

Alternate Flow: None

Test Case 1.26: Generate recycling pickup request Receipt

Test Scenario: The rider generates the recycling pickup request receipt.

Preconditions: Rider has picked up the recyclable items and needs to create a receipt.

Test Steps:

1. Enter the exact weight of each recyclable item.

2. Tap the "Generate Receipt" option.

Expected Result: The app generates the recycling pickup request receipt, including a detailed list of recyclable items with their exact weights and the total price for these recyclables.

Alternate Flow: None

Test Case 1.27: Rider Actions on Receipt Approval

Test Scenario: The rider decides on actions regarding the generated recycling pickup request receipt while waiting for customer approval. The rider can choose to either wait for the customer's approval or cancel the receipt.

Preconditions: Rider has generated a recycling pickup request receipt.

Test Steps:

1. View the generated recycling pickup request receipt.
2. Choose to either wait for customer approval, keeping the recycling pickup request in a pending state, or cancel the receipt.

Expected Result: The rider's choice is recorded by the app, and the recycling pickup request status is updated accordingly.

Alternate Flow 1: If the customer neither approves nor rejects the receipt, the rider has the option to cancel the receipt, updating the recycling pickup request status to "Canceled."

Test Case 1.28: Enter Reason for Pickup Cancellation

Test Scenario: The rider enters a reason for canceling a scheduled pickup.

Preconditions: Rider is viewing the list of assigned recycling pickup requests and decides to cancel a scheduled pickup.

Test Steps:

1. Select an assigned recycling pickup request from the list.
2. App displays detailed information about the selected recycling pickup request, including customer location and pickup details.
3. Decide to cancel the pickup and tap the "Cancel Pickup" option.
4. App prompts the rider to provide a reason for the cancellation.
5. Enter a text description of the reason.
6. Submit the reason.
7. The app records the provided reason for the canceled pickup.

Expected Result: The reason for the pickup cancellation is recorded by the app, allowing the rider to provide a valid explanation for the canceled recycling pickup request.

Alternate Flow: If the rider changes their mind and decides not to cancel the pickup, they can go back to the recycling pickup request details without providing a reason and continue with the pickup.

4.1.2 Test Cases Web App (FR 2.1 – FR 2.19)

4.1.2.1 Warehouse Manager Module (FR 2.1 – FR 2.6)

Test Case 2.1: Warehouse Manager Sign In

Test Scenario: The warehouse manager attempts to sign in.

Preconditions: Warehouse manager is not logged in.

Test Steps:

1. Open the warehouse manager web app.
2. Enter a valid email and password.
3. Tap the "Sign In" button.

Expected Result: The warehouse manager is successfully signed in and gains access to their account.

Alternate Flow 1: If the warehouse manager provides an incorrect email or password:

Test Steps:

1. In step 2, enter an incorrect email or password.
2. Tap the "Sign In" button.

Expected Result: The system displays an error message indicating that the login credentials are incorrect, and access is denied.

Test Case 2.2: Warehouse Manager Logout

Test Scenario: The warehouse manager logs out of their account.

Preconditions: Warehouse manager is logged in.

Test Steps:

1. Tap the "Logout" option in the web app.

Expected Result: The warehouse manager is successfully logged out of their account, and the app redirects them to the login screen.

Alternate Flow: None

Test Case 2.3: View Pending Pickups

Test Scenario: The warehouse manager views a list of pending pickups.

Preconditions: Warehouse manager is logged in.

Test Steps:

1. Open the app.
2. Navigate to the "Pending Pickups" section.

Expected Result: The app displays a list of pending pickups, including details such as recycling pickup request information and rider details.

Alternate Flow: None

Test Case 2.4: View Completed Pickups

Test Scenario: The warehouse manager views a list of completed pickups.

Preconditions: Warehouse manager is logged in.

Test Steps:

1. Open the app.
2. Navigate to the "Completed Pickups" section.

Expected Result: The app displays a list of completed pickups, including details such as recycling pickup request information, rider details, and pickup confirmation.

Alternate Flow: None

Test Case 2.5: Confirm Completed Shift by Rider

Test Scenario: The warehouse manager confirms the completion of a rider's shift and verifies the accuracy of the total weight of recycling items.

Preconditions: Admin has access to the list of riders, and a rider has returned to the warehouse after completing their pickups.

Test Steps:

1. Admin accesses the list of riders and selects a specific rider who has returned to the warehouse after completing their pickups.
2. The system displays details of the selected rider's shift, including the list of completed pickups and the total weight of recyclables picked up by the rider.
3. Admin checks the accuracy of the total weight of recycling items picked up by the rider.
4. If the total weight of recycling items picked up by the rider matches the total weight of recyclables delivered to the warehouse, the admin confirms the rider's shift as "Completed and Verified."

Expected Result: The admin successfully confirms the completion of the rider's shift, and the system updates the status accordingly.

Alternate Flow 1: If the total weight of recycling items picked up by the rider does not match the total weight of recyclables delivered to the warehouse, the admin marks the shift as "Completed with Discrepancies." The system notifies relevant parties for resolution.

Test Case 2.6: View Canceled Pickups

Test Scenario: The warehouse manager views canceled pickups.

Preconditions: Warehouse manager is viewing a pending pickup.

Test Steps:

1. The system displays a list of canceled pickups, including details such as recycling pickup request information, rider details, and reason for cancellation.

Expected Result: The system successfully displays a list of canceled pickups, allowing the warehouse manager to view details and reasons for each cancellation.

Alternate Flow: None

4.1.2.2 Admin Module (FR 2.7 – FR 2.19)

Test Case 2.7: Admin Sign In

Test Scenario: The admin attempts to sign in.

Preconditions: Admin is not logged in.

Test Steps:

1. Open the admin web app.
2. Enter valid admin credentials (username and password).
3. Tap the "Sign In" button.

Expected Result: The admin is successfully signed in and gains access to the admin dashboard.

Alternate Flow 1: If the admin provides incorrect credentials:

Test Steps:

1. In step 2, enter incorrect admin credentials.
2. Tap the "Sign In" button.

Expected Result: The system displays an error message indicating that the login credentials are incorrect, and access is denied.

Test Case 2.8: Admin Logout

Test Scenario: The admin logs out of their admin account.

Preconditions: Admin is logged in.

Test Steps:

1. Tap the "Logout" option in the admin app.

Expected Result: The admin is successfully logged out of their admin account, and the system redirects them to the login screen.

Alternate Flow: None

Test Case 2.9: Register Warehouse Manager

Test Scenario: The admin registers a new warehouse manager.

Preconditions: Admin is logged in.

Test Steps:

1. In the admin web app, select the "Register Warehouse Manager" option.
2. Provide valid information for the new warehouse manager, including name, contact details, and username.
3. The system validates the information.
4. Upon successful validation, the new warehouse manager account is created.

Expected Result: The new warehouse manager account is successfully registered in the system.

Alternate Flow: If the information provided is invalid, the system displays an error message.

Test Case 2.10: Register New Rider

Test Scenario: The admin registers a new rider.

Preconditions: Admin is logged in.

Test Steps:

1. In the admin app, select the "Register New Rider" option.
2. Provide valid information for the new rider, including name, contact details, and username.
3. The system validates the information.
4. Upon successful validation, the new rider account is created.

Expected Result: The new rider account is successfully registered in the system.

Alternate Flow: If the information provided is invalid, the system displays an error message.

Test Case 2.12: View Rider Details

Test Scenario: The admin views details of a rider.

Preconditions: Admin is logged in.

Test Steps:

1. Click the "Riders" option in the admin app.
2. Specify the rider for whom the admin wants to view details.

Expected Result: The system displays the rider's information, including name, contact details, and performance statistics.

Alternate Flow: Rider details are not found, and the system provides an appropriate message.

Test Case 2.13: View Pending Pickups

Test Scenario: The admin views a list of pending pickups.

Preconditions: Admin is logged in.

Test Steps:

1. Select the "Pending Pickups" option in the admin app.

Expected Result: The system displays a list of pickups that are pending, along with relevant details.

Alternate Flow: No pending pickups are found, and the system provides an appropriate message.

Test Case 2.14: View Completed Pickups

Test Scenario: The admin views a list of completed pickups.

Preconditions: Admin is logged in.

Test Steps:

1. Select the "Completed Pickups" option in the admin app.

Expected Result: The system displays a list of completed pickups, including details such as rider, customer, and materials collected.

Alternate Flow: No completed pickups are found, and the system provides an appropriate message.

Test Case 2.15: View Customer Details

Test Scenario: The admin views details of a customer.

Preconditions: Admin is logged in.

Test Steps:

1. Click the "Customers" option in the admin app.
2. Specify the customer for whom the admin wants to view details.

Expected Result: The system displays the customer's information, including name, contact details, and recycling history.

Alternate Flow: Customer details are not found, and the system provides an appropriate message.

Test Case 2.16: View Pending Payments

Test Scenario: The admin views a list of pending payments.

Preconditions: Admin is logged in.

Test Steps:

1. Select the "Pending Payments" option in the admin app.

Expected Result: The system displays a list of payments that are pending, along with relevant details.

Alternate Flow: No pending payments are found, and the system provides an appropriate message.

Test Case 2.17: Upload Payment Proof

Test Scenario: The admin uploads payment proof for completed payments.

Preconditions: Admin is logged in, and a payment is marked as completed.

Test Steps:

1. Select a completed payment in the admin app.
2. Upload payment proof.

Expected Result: The admin successfully associates the payment proof with the completed payment.

Alternate Flow: None

Test Case 2.18: View Inventory

Test Scenario: The admin views the inventory of recyclable materials.

Preconditions: Admin is logged in.

Test Steps:

1. Click the "Inventory" option in the admin app.

Expected Result: The system displays the current inventory of recyclable materials, including quantities and types.

Alternate Flow: None

Test Case 2.19: Update Rates of Recycling Materials

Test Scenario: The admin updates the rates for recycling materials.

Preconditions: Admin is logged in.

Test Steps:

1. Select the "Update Rates" option in the admin app.
2. Update the rates for various recyclable materials.

Expected Result: Admin successfully updates the rates for recycling materials.

Alternate Flow: None

4.2 Test Case Grid

Test Case ID	Test Case Description	Status
1.1	Customer Registration	✓
1.2	Customer Sign In	✓
1.3	Customer Logout	✓
1.4	View Account Profile	✓
1.5	Update Credentials	✓
1.6	Schedule Pickups for Recycling	✓
1.7	View Current recycling pickup request	✓
1.8	Cancel Current recycling pickup request	✓
1.9	View Pickup History	✓
1.10	View Pending/Paid Payments	✓
1.11	View Recycling Statistics on Home Page	✓
1.12	View Top 3 Organizations on Home Page	✓
1.13	Access Educational Content on Recycling	✓
1.14	View Individual Leaderboard	✓
1.15	View Organization Leaderboard	✓
1.16	View Friend's Leaderboard	✓
1.17	Add Friends	✓

1.18	Accept/Delete Friend Requests	✓
1.19	Approve/Cancel recycling pickup request	✓
1.20	Rider Sign In	✓
1.21	Rider Logout	✓
1.22	View Assigned recycling pickup requests	✓
1.23	Select recycling pickup request	✓
1.24	View Detailed recycling pickup request	✓
1.25	View Map for Navigation	✓
1.26	Generate recycling pickup request Receipt	✓
1.27	Rider Actions on Receipt Approval	✓
1.28	Enter Reason for Pickup Cancellation	✓
2.1	Warehouse Manager Sign In	✓
2.2	Warehouse Manager Logout	✓
2.3	View Pending Pickups	✓
2.4	View Completed Pickups	✓
2.5	Confirm Completed Shift by Rider	✓
2.6	View Canceled Pickups	✓
2.7	Admin Sign In	✓
2.8	Admin Logout	✓
2.9	Register Warehouse Manager	✓
2.10	Register New Rider	✓
2.11	View Rider Details	✓
2.12	View Pending Pickups	✓
2.13	View Completed Pickups	✓
2.14	View Customer Details	✓
2.15	View Pending Payments	✓
2.16	Upload Payment Proof	✓
2.17	View Inventory	✓
2.18	Update Rates of Recycling Materials	✓

*Note: ✓ indicates the test case is working as expected, while ✗ indicates the test case failed.

4.3 Mobile App Implementation

4.3.1 Navigating Customer to Schedule Pickup (Frontend)

SaafPakistan's mobile application is developed using Flutter, an open-source UI software development toolkit created by Google. Flutter enables developers to create natively compiled applications for mobile, web, and desktop from a single codebase. This approach provides a high-performance, beautiful, and consistent user experience across all platforms.

The frontend architecture of SaafPakistan's mobile app follows a structured approach to ensure modularity, maintainability, and scalability. The main components of the architecture include:

- **Widgets:** These are the building blocks of the Flutter UI. Each widget represents a part of the user interface.
- **Routes:** These manage the navigation and flow of the application, defining how users move between different screens.
- **State Management:** This handles the application's state, ensuring that the UI is updated in response to changes in the state.

Below is a code snippet which contains a Flutter widget that uses the **GestureDetector** to navigate to a different screen when tapped. This widget displays a container with an SVG icon and navigates to the **SchedulePickup** screen upon user interaction.

```
Flutter Widget Code Snippet

GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => SchedulePickup(
          uid: widget.uid,
          accountType: accountType,
        )),
    ),
  child: Container(
    width: screenWidth * 0.4,
    height: screenHeight * 0.245,
    decoration: ShapeDecoration(
      color: Colors.white,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
      ),
    ),
    child: SvgPicture.asset(
      "assets/icons/Pickup.svg",
    ),
  ),
),
```

Figure 4.1 Flutter Widget Code Snippet (Frontend)

4.3.2 Displaying Top 3 Organizations (Backend)

Our Mobile app's backend infrastructure utilizes Firebase Cloud Firestore for data management. The backend logic is implemented using Dart, specifically for Firestore operations, providing a scalable and efficient solution.

Below is a Dart function, **fetchTopOrganizations**, designed to fetch data from Firestore and retrieve the top organizations based on specific criteria. The function initializes a loading state to indicate the start of data retrieval. It then queries the Firestore collection 'leaderboards', filtering entries by account type ('Company') and sorting them by points. The result set is limited to the top 3 organizations.

Fetches documents are transformed into structured data, extracting organization name and waste recycled information. Upon successful fetch, the loading state is updated. Error handling is also implemented to log encountered errors, ensuring robustness of the data fetching process.

Dart Function Code Snippet

```
Future<void> fetchTopOrganizations() async {
  setState(() {
    organizationsLoading = true;
  });

  try {
    QuerySnapshot organizationsSnapshot = await
    FirebaseFirestore.instance
      .collection('leaderboards')
      .where('accountType', isEqualTo: 'Company')
      .orderBy('points', descending: true)
      .limit(3)
      .get();

    organizationList = organizationsSnapshot.docs.map((organization)
    {
      return {
        'organizationName': organization['cus'],
        'wasteRecycled': organization['points'],
      };
    }).toList();

    setState(() {
      organizationsLoading = false;
    });
  } catch (error) {
    print('Error fetching top organizations: $error');
    setState(() {
      organizationsLoading = false;
    });
  }
}
```

Figure 4.2 Dart Function Code Snippet (Backend)

4.4 Web App Implementation

4.4.1 Fetching Leaderboard Data From API

SaafPakistan's Web app frontend is developed using React, a popular JavaScript library for building user interfaces. React is known for its component-based architecture, which promotes the creation of reusable UI components. This modular approach makes the development process more efficient and the application easier to maintain. The frontend leverages React hooks such as 'useState' and 'useEffect' to manage state and lifecycle methods, providing a seamless user experience.

The styling of the frontend is done using the react-bootstrap library, which offers a collection of reusable UI components styled according to the Bootstrap framework. This allows for consistent, responsive, and visually appealing design throughout the application.

Below is a snippet of a React component for fetching and displaying leaderboard data. This component interacts with the backend to retrieve the leaderboard entries and displays them to the user.

```
React Code Snippet

1 import React, { useContext, useEffect, useState } from "react";
2 import { Spinner } from "react-bootstrap";
3 import { AuthContext } from "../../context/AuthContext";
4 import * as apiService from "../../api-service";
5
6 function Leaderboards({ type }) {
7   const [leaderboards, setLeaderboards] = useState([]);
8   const [loading, setLoading] = useState(true);
9   const [error, setError] = useState(null);
10  const { user } = useContext(AuthContext);
11
12  useEffect(() => {
13    if (user && user.getIdToken) {
14      (async () => {
15        const userIdToken = await user.getIdToken();
16        try {
17          const fetchedData = await apiService.getLeaderboardData({
18            userIdToken, type });
19          setLeaderboards(fetchedData);
20        } catch (error) {
21          setError("Error fetching leaderboard data: " + error.message);
22        } finally {
23          setLoading(false);
24        }
25      })();
26    }, [user, type]);
27
28    if (loading) return <Spinner animation="border" />;
29    if (error) return <div>Error: {error}</div>;
```

Figure 4.3 React code Snippet

4.4.1.1 Frontend Code Snippet Breakdown

Imports and Context Setup (lines 1-4): The component starts by importing necessary modules, including React, state and effect hooks, the **Spinner** component from **react-bootstrap**, the **AuthContext** for authentication, and the API service.

Component Definition (lines 6-10): The **Leaderboards** function component is defined, accepting a **type** prop to specify the type of leaderboard to fetch.

Fetching Data with useEffect (lines 12-26): The **useEffect** hook is used to fetch leaderboard data when the component mounts or when the **user** or **type** prop changes. The user's ID token is retrieved and used to authenticate the API request.

Conditional Rendering (lines 28-29): The component renders a loading spinner while data is being fetched, displays an error message if an error occurs.

4.4.2 Fetching Leaderboard Data From Firebase

Our Web App's backend is developed using Node.js with Firebase as the database service. Node.js is a powerful JavaScript runtime environment that allows for the development of scalable and efficient server-side applications. Firebase provides a real-time NoSQL cloud database that integrates seamlessly with Node.js, making it an excellent choice for handling dynamic data and real-time updates.

Backend architecture is designed using an organized structure that separates concerns and ensures maintainability and scalability. The architecture includes controllers, which handle incoming requests and interactions with the frontend, and the database access logic, which interfaces with Firebase to store and retrieve data.

Below is a code snippet containing a controller for fetching leaderboard data from Firebase. This controller fetches leaderboard entries, orders them by points, and sends the data back to the frontend.

Controller Code Snippet

```
1  const admin = require("firebase-admin");
2  const firestore = admin.firestore();
3
4  module.exports.getLeaderboard = async (req, res) => {
5    try {
6      const type = req.query.type || "Personal";
7
8      // Query Firestore for leaderboard documents of a specific type,
      // ordered by points
9      const querySnapshot = await firestore
10        .collection("leaderboards")
11        .where("accountType", "==", type)
12        .orderBy("points", "desc")
13        .get();
14
15      // Process the query results and build the leaderboard data
16      const leaderboardData = querySnapshot.docs.map((docSnap, index) => ({
17        id: docSnap.id,
18        uid: docSnap.data().uid,
19        cus: docSnap.data().cus,
```

```

19     cus: docSnap.data().cus,
20     points: docSnap.data().points,
21     accountType: docSnap.data().accountType,
22     rank: index + 1,
23   }));
24
25   res.status(200).json(leaderboardData);
26 } catch (error) {
27   console.error("Error fetching leaderboard data:", error);
28   res.status(500).json({ error: error.message });
29 }

```

Figure 4.4 Controller Code Snippet

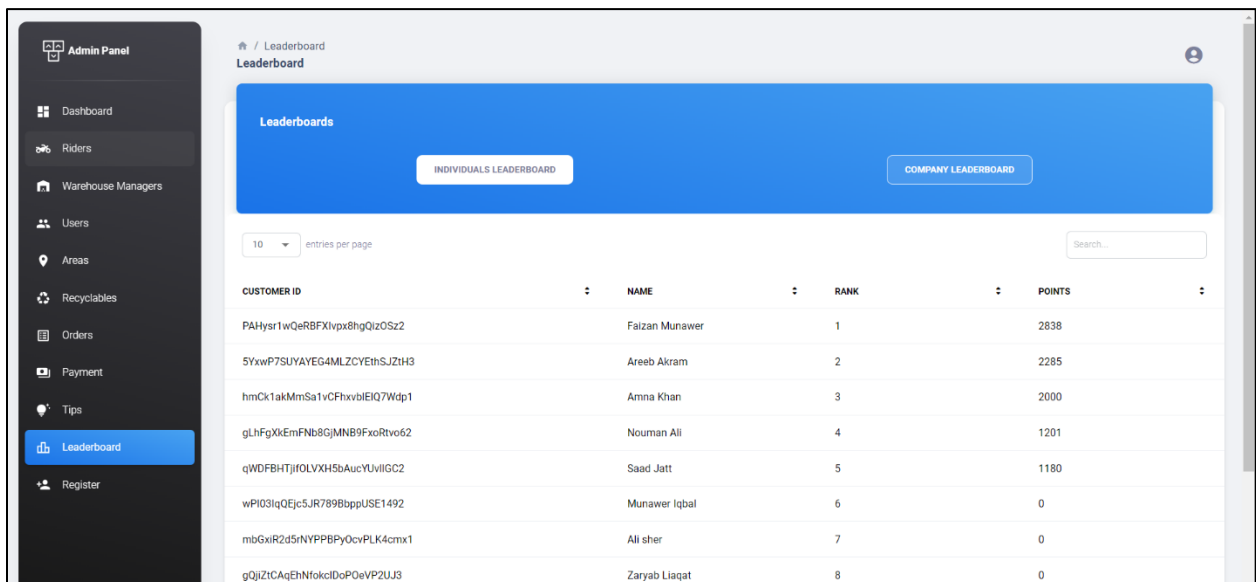
4.4.2.1 Backend Code Snippet Breakdown

Firestore Initialization (lines 1-2): The code begins by requiring the Firebase Admin SDK and initializing the Firestore database instance.

Controller Definition (line 4): The **getLeaderboard** function is defined as an asynchronous function that handles incoming requests.

Querying Firestore (lines 9-13): It queries the Firestore database to fetch leaderboard documents that match a specific account type, ordered by points in descending order.

Processing Data (lines 16-23): The results from the query are processed to construct a leaderboard with additional information like rank.



The screenshot shows the Admin Panel interface. On the left is a sidebar with navigation links: Admin Panel, Dashboard, Riders, Warehouse Managers, Users, Areas, Recyclables, Orders, Payment, Tips, Leaderboard (highlighted), and Register. The main content area is titled 'Leaderboard' and features a blue header with 'Leaderboards' and two buttons: 'INDIVIDUALS LEADERBOARD' and 'COMPANY LEADERBOARD'. Below the header, there is a dropdown for '10 entries per page' and a search bar. The main content is a table with the following data:

CUSTOMER ID	NAME	RANK	POINTS
PAHysr1wQeRBFxlvpx8hgQizOSz2	Faizan Munawer	1	2838
5YxwP7SUYAYEG4MLZCYEtH3	Areeb Akram	2	2285
hmCk1akMmSa1vCFhxbIEIQ7Wdp1	Amna Khan	3	2000
gLhFgXkEmfNB8GjMNB9FxoRtvof62	Nourman Ali	4	1201
qWDFBHTjIFOLVXH5bAucYUvIIGC2	Saad Jatt	5	1180
wPI03lqQEjc5JR789BppUSE1492	Munawer Iqbal	6	0
mbGxiR2d5NYPBPpOcvPLK4cmx1	Ali sher	7	0
gQjZtCAqEhNfokclDoPoVp2UJ3	Zaryab Liaqat	8	0

Figure 4.5 Leaderboards Screenshot Admin panel

Response Handling (lines 25-30): The constructed leaderboard data is sent back to the client with a 200-status code. In case of an error, a 500-status code and an error message are returned.

4.5 Screenshots Of Key Features

4.5.1 SaafPakistan Mobile Application Screenshots

4.5.1.1 Customer Dashboard

The dashboard provides an overview of the user's recycling activities, including total waste recycled, cash earned, and CO2 emissions reduced, along with options to schedule pickups, learn about recycling benefits, and view leaderboards and organizational rankings.



Figure 4.6 Customer Dashboard

4.5.1.2 Map Integration to Register Address for customer

The registration screen utilizes TPL Maps for location-based service, allowing users to select their area and confirm their address on a map for accurate pickup service.

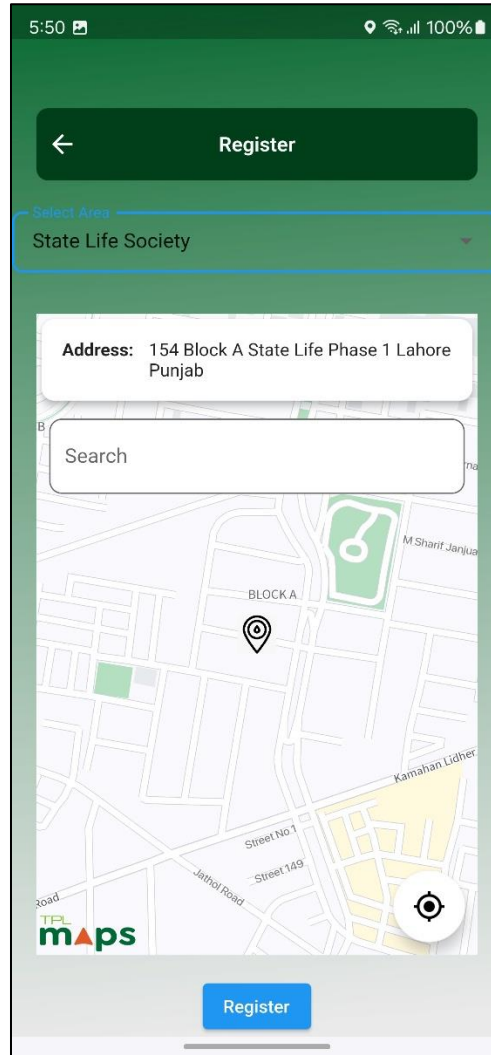


Figure 4.7 Maps Integration for Address Registration

4.5.1.3 Leaderboards in Mobile App

The leaderboard screen displays rankings based on points earned by recycling, allowing users to compare their performance with others.

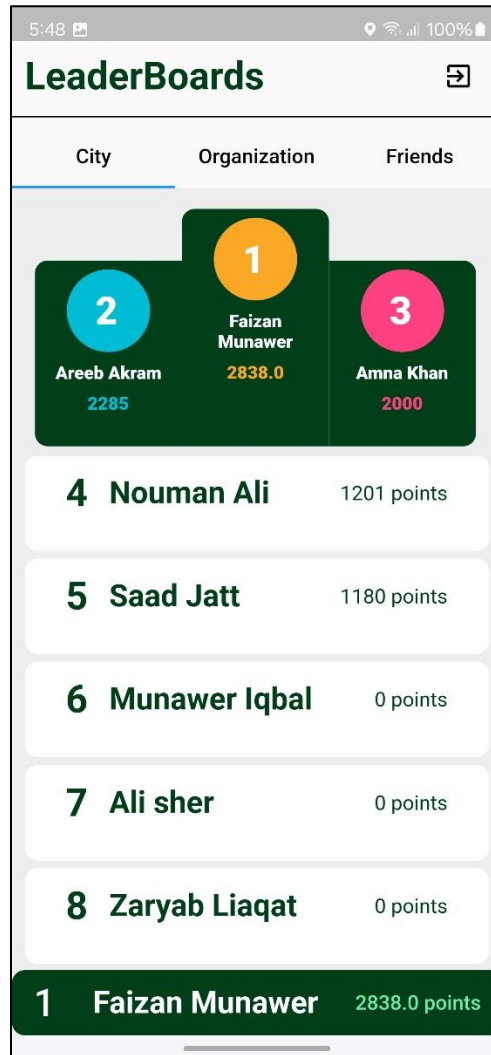


Figure 4.8 Customer Recycling Leaderboards

4.5.1.4 Map Integration for Rider

When the rider selects a pickup, Current order screen is displayed which displays order summary with navigation details, utilizing TPL Maps to help the rider navigate to the customer's home.

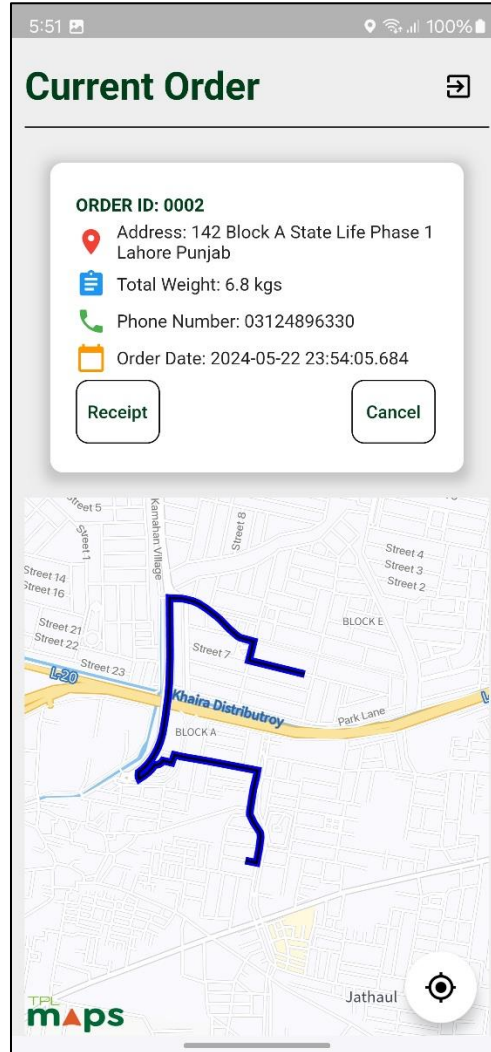


Figure 4.9 Map Integration for Rider

4.5.2 SaafPakistan Web Application Screenshots

4.5.2.1 Admin Dashboard

Admin dashboard displays key metrics such as total orders, pending orders, completed orders, cancelled orders, user sign-ups, payment statistics by month, and recyclable items data.

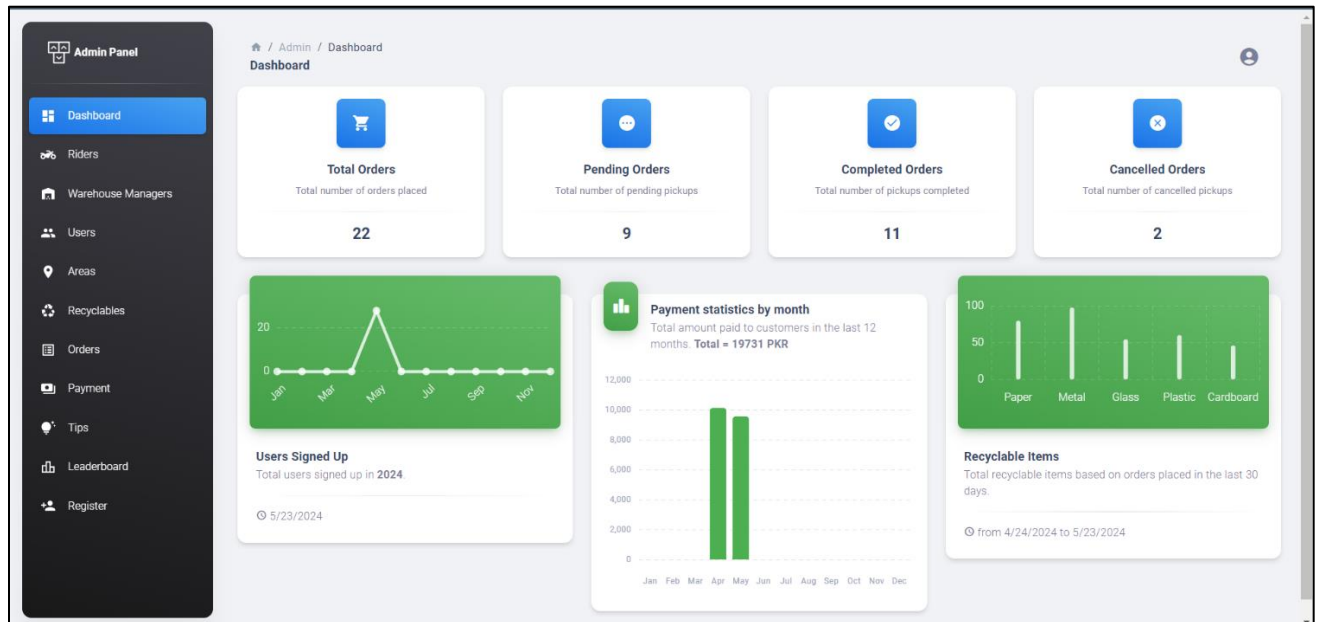


Figure 4.10 Admin Dashboard

5 Conclusion

Our team has successfully developed a comprehensive mobile application, SaafPakistan, for recycling as our final year project for our Software Engineering undergraduate degree. Simultaneously, we developed a web application to serve as an admin panel, complementing the mobile service. The project involved various phases of development, resulting in a highly functional and efficient mobile app aimed at revolutionizing recycling practices in Pakistan.

The initial phase focused on designing a user-friendly interface using Flutter, ensuring a seamless experience for users, including residents and businesses. The application incorporates features such as scheduling recycling pickups, gamification elements to encourage recycling and educational content on waste management and recycling to foster greater awareness and participation. Additionally, admin panel was developed to manage riders, warehouse managers, and app users, providing comprehensive control and coordination of the recycling process.

In the subsequent phase, we integrated Firebase Firestore for database management and authentication, ensuring secure storage and handling of user data for the mobile application. For the web application's backend, Node.js was utilized to, facilitate efficient processing of user requests and interactions with the Firestore database. This robust architecture supports scalable and responsive system performance.

The SaafPakistan app's gamification feature leverages point systems and leaderboards, motivating users through competitive and reward-based mechanisms. This approach, supported by research studies, demonstrates significant potential in increasing recycling rates and fostering a community-driven approach to waste management.

For future work, integrating additional features such as real-time tracking of recycling pickups, expanding reward partnerships, and incorporating machine learning algorithms to predict and optimize recycling patterns could further enhance the application's impact. Improving reporting and analytics capabilities would provide deeper insights into user behavior and the effectiveness of the recycling program.

In conclusion, the SaafPakistan mobile application successfully integrates Flutter and React to offer an intuitive interface, robust database management, and engaging gamification features. With further enhancements, it has the potential to revolutionize waste management practices in Pakistan, empowering the community to achieve greater environmental sustainability and cleanliness. The collaborative efforts of residents, businesses, and organizations, combined with the innovative features of the SaafPakistan app, can significantly contribute to reducing environmental pollution, promoting recycling, and ultimately leading to a more sustainable and environmentally conscious society

6 References

- [1] World Bank. (2022, May 11). Gamification for Recycling: A Review of the Literature. World Bank. [Online]. Available: <https://www.worldbank.org/en/events/2022/05/11/policy-research-report-improving-effective-coverage-in-health>
- [2] Soesatyo, Y. (2019). Financial Incentives and Gamification to Increase Recycling Rates. International Journal of Technology, 10(6), 1161–1170. [Online]. Available: <https://ijtech.eng.ui.ac.id/article/view/2644>
- [3] Mulrow, J. (2019). The state of carbon footprint calculators: An evaluation of calculator design and user interaction features. ResearchGate. [Online]. Available: https://www.researchgate.net/publication/329547708_The_state_of_carbon_footprint_calculators_An_evaluation_of_calculator_design_and_user_interaction_features
- [4] Aguiar-Castillo, L. (2019). Gamification as An Approach to Promote Tourist Recycling Behavior. Sustainability, 11(8), 2201. [Online]. Available: <https://www.mdpi.com/2071-1050/11/8/2201>
- [5] Recycle Coach. (n.d.). [Mobile app]. Google Play Store. [Online]. Available: https://play.google.com/store/apps/details?id=mobi.recyclecoach.worldster.pack&pcampaignid=web_share
- [6] iRecycle. (n.d.). [Mobile app]. Google Play Store. [Online]. Available: <https://play.google.com/store/apps/details?id=com.earth911.irecycle&hl=en&gl=US>
- [7] Recycle Smart. (n.d.). [Mobile app]. Google Play Store. [Online]. Available: https://play.google.com/store/apps/details?id=au.com.recyclesmart.recyclepedia&pcampaignid=web_share
- [8] Zeloop. (n.d.). [Mobile app]. Google Play Store. [Online]. Available: https://play.google.com/store/apps/details?id=com.zeloop.app&hl=en_US&gl=US
- [9] The Click. (n.d.). [Mobile app]. Google Play Store. [Online]. Available: <https://play.google.com/store/apps/details?id=be.fostplus.theclick>
- [10] Gaia. (n.d.). [Mobile app]. iOS Example. [Online]. Available: <https://iosexample.com/gamified-recycling-app-that-uses-machine-learning-to-provide-users-with-recycling-information-based-on-item-scanned/>
- [11] Aabroo Welfare Organization. (n.d.). Aabroo Welfare Organization. [Online]. Available: <https://aabroo.org/aabroo/>
- [12] Firebase. (n.d.). Firebase for Android. [Online]. Available: <https://firebase.google.com/docs/android/setup>
- [13] React. (n.d.). React Documentation. [Online]. Available: <https://legacy.reactjs.org/docs/getting-started.html>
- [14] Flutter. (n.d.). Flutter Documentation. [Online]. Available: <https://docs.flutter.dev/ui/widgets>