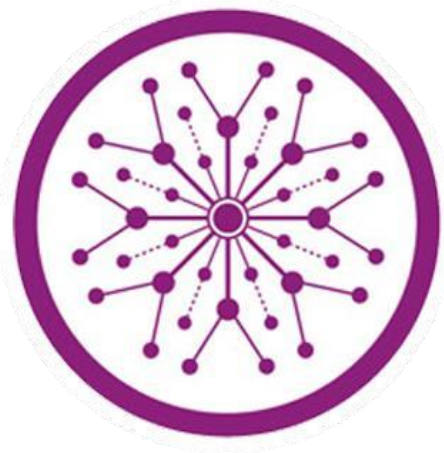
Rejunkify

Final Year Project

Session 2020-2024

A project submitted in partial fulfilment of the degree of

BS in Computer Science



Department of Computer Sciences

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2024

Type (Rejunkifyt)	[✓] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	Development
FYP ID	FYP-BCSM-F23-044

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-F20-173	Muaaz Ahmad	bcsm-f20-173@superior.edu.pk	
(ii)	BCSM-F20-232	Abdullah Asim	bcsm-f20-232@superior.edu.pk	
(iii)	BCSM-F20-237	Shahroz	bcsm-f20-237@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I _____ S/D of Manzoor Ahmad, group leader of FYP under registration no _____at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: _____ Signature: _____

Name of Supervisor: ***Ma'am Humaira Muqadus,*** Co-Supervisor: Mam Maria

Designation: Lecturer Designation: Lecturer

Signature: _____ Signature: _____

HoD: Dr. Irfan ud Din

Signature: _____

ReJunkify

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

We dedicate this Mobile Application Development project to our dedicated supervisor, **Ma'am Humaira Muqadus**, whose unwavering guidance and expertise have been the cornerstones of our learning journey. Your constructive feedback, encouragement, and insightful suggestions have been instrumental in shaping this project and enhancing our skills in mobile app development.

We also extend my sincere appreciation to my FYP manager, **Sir Jawad Ahmad**, for their continuous support and mentorship throughout this project. Your vision and encouragement have been crucial in navigating the challenges and complexities of mobile application development.

This project stands as a testament to the invaluable lessons and experiences gained under your supervision. We are truly grateful for the opportunity to work under your guidance, and we believe the skills acquired will serve as a solid foundation for our future endeavors in the field.

Thank you for being not just mentors but also inspirations on this educational journey.

Acknowledgement

We would like to express our sincere gratitude to our supervisor, **Ma'am Humaira Muqadus**, for their invaluable guidance, unwavering support, and continuous encouragement throughout the entire journey of our final year project. Their expertise, insightful feedback, and dedication to our academic and professional development have been instrumental in the successful completion of this project. We also extend our appreciation to **Ma'am Maria**, whose feedback and suggestions enriched the project and contributed to its overall quality.

Our heartfelt thanks go to our classmates and friends for their camaraderie, shared insights, and collaborative spirit. Your support has been a source of motivation during the challenging phases of this project.

Additionally, we would like to acknowledge the Superior University for providing us with the resources and environment conducive to our academic pursuits. This project marks the culmination of our academic journey, and it wouldn't have been possible without the guidance and encouragement we received along the way. Thank you, **Ma'am Humaira Muqadus**, for being a mentor and guide who believed in our potential and nurtured our growth.

Executive Summary

The House-to-House Junk Collection App is a pioneering Final Year Project (FYP) centered on revolutionizing waste recycling. This initiative harnesses the power of Flutter, a versatile mobile app development framework, to facilitate homeowners in scheduling junk pickups from their residences. By bridging the gap between households and recyclers, the project promotes responsible disposal and environmental sustainability.

Introducing "**ReJunkify**": Revolutionize your recycling journey with our innovative FYP project. Say goodbye to clutter and waste by seamlessly connecting with a skilled network of recyclers through our user-friendly app. With **ReJunkify**, you can effortlessly schedule and book experienced recyclers to collect and purchase your home junk, turning your trash into cash. Join us in creating a sustainable future while earning from your clutter. Embrace convenience, environmental consciousness, and economic benefit all in one app.

Table of Contents

Plagiarism Free Certificate ii
Dedication v
Acknowledgement vi
Executive Summary vii
Table of Contents viii
List of Figures xi
4.11. Data Flow diagram 48 xi
Chapter 1 2
Introduction 2
Introduction 3
1.1. Background 3
1.2. Motivations and Challenges 3
1.3. Goals and Objectives 4
1.4. Literature Review/Existing Solutions 4
1.5. Gap Analysis 4
1.6. Proposed Solution 5
1.7. Project Plan 5
1.7.1. Work Breakdown Structure 6
1.7.2. Roles & Responsibility Matrix 7
1.7.3. Gantt Chart 8
1.9. Empathy Map 9
Chapter 2 10
Software Requirement Specifications 10
2.1. Introduction 11
2.1.1. Purpose 11
2.1.2. Document Conventions 11
2.1.3. Intended Audience and Reading Suggestions 12
2.1.4. Product Scope 12
2.2. Overall Description 13
2.2.1. Product Perspective 13
2.2.2. User Classes and Characteristics 13
2.2.3. Operating Environment 14
2.2.4. Design and Implementation Constraints 15
2.2.5. Assumptions and Dependencies 15
2.3. External Interface Requirements 16
2.3.1. User Interfaces 16
2.3.2. Hardware Interfaces 16
2.3.3. Software Interfaces 17
2.3.4. Communications Interfaces 17
2.4. System Features 18

2.4.1. System Feature 18
2.4.1.1. Description and Priority 18
2.4.1.2. Stimulus/Response Sequences 19
2.4.1.3. Functional Requirements 19
2.4.2. System Feature 20
2.4.2.1. Description and Priority 20
Description: 20
Priority: 20
2.4.2.2. Stimulus/Response Sequences 20
2.4.2.3 Functional Requirements 21
2.5. Nonfunctional Requirements 22
2.5.1. Performance Requirements 22
2.5.2.Safety Requirements 23
2.5.4.Usability Requirements 24
2.5.5. Reliability Requirements 24
2.5.6.Maintainability/Supportability Requirements 25
2.5.7. Portability Requirements 25
2.5.8.Efficiency Requirements 26
2.6. Domain Requirements 26
Chapter 3 27
Use Case Analysis 27
3.1. Use Case Model 29
3.2. Use Cases Description 30
Chapter 4 31
System Design 31
4.1. Architecture Diagram 33
4.2. Domain Model 34
4.3.Entity Relationship Diagram with data dictionary 35
4.4. Class Diagram 36
4.5.Sequence / Collaboration Diagram 37
4.6. Operation contracts 38
4.7. Activity Diagram 41
4.8. State Transition Diagram 42
4.9. Component Diagram 43
4.10. Deployment Diagram 44
Chapter 5 45
Implementation 45
5.1. Important Flow Control/Pseudo codes 46
5.2. Components, Libraries, Web Services and stubs 49
5.3. Deployment Environment 51
5.4. Tools and Techniques 53
5.5. Best Practices / Coding Standards 55
5.6. Version Control 55
Chapter 6 56

Testing and Evaluation 56
6.1. Use Case Testing 57
6.2. Equivalence partitioning 59
6.3. Boundary value analysis 59
6.4. Unit testing 60
6.5. Integration testing 61
6.6. Performance testing 61
6.7. Stress Testing 62
6.8. Application Interface 63

Chapter 7 66
Summary, Conclusion and Future Enhancements 66
7.1. Project Summary 67
7.2. Achievements and Improvements 68
7.3. Critical Review 70
7.4. Lessons Learnt 72
7.5. Future Enhancements/Recommendations 73
Reference and Bibliography 75

List of Figures

1.7.1 Work Breakdown Structure 7
1.7.2 Roles & Responsibility Matrix 8
1.7.3 Gantt Chart 9
1.9.1 Empathy Map 11
3.1 Use Case Model 33
4.1 Architecture Diagram 37
4.2 Domain Model 38
4.3 ERD Diagram 40
4.4 Class Diagram 41
4.5 Sequence / Collaboration Diagram 42
4.8. State Transition Diagram 45
4.9. Component Diagram 46
4.10. Deployment Diagram 47
4.11. Data Flow diagram 48

Chapter 1
Introduction

Chapter 1: Introduction

Introduction

Welcome to ReJunkify, your go-to app for a cleaner environment! We've made recycling and junk collection easier than ever. If you have items you no longer need, simply book a recycler through our app, and they'll come right to your doorstep to pick up your junk. Junk collectors can also use the app to connect with households looking to declutter. Let's work together to make recycling a breeze and contribute to a greener, healthier planet

1.1. Background

In some new housing communities in Pakistan, recyclers face a challenge—they're not allowed to collect junk from these areas. As a result, valuable items that could be recycled end up in the regular garbage, causing harm to the environment. But fear not, because ReJunkify is here to offer a solution! Our app connects recyclers with households, making it easy for residents to responsibly dispose of their junk. By using ReJunkify, we're not just cleaning up homes; we're also making a positive impact on the environment by reducing waste and promoting recycling. Together, let's make our communities cleaner and greener

1.2. Motivations and Challenges

ReJunkify aims to be a catalyst for positive environmental change, empowering users to reduce their ecological impact. Through the app, individuals can easily participate in responsible waste disposal, contributing to the global effort to minimize landfill waste and conserve resources. The platform fosters a sense of community engagement, showcasing the collective impact of users in creating a cleaner and healthier neighbourhood. Convenience and efficiency are core motivations, making the process of decluttering homes and connecting junk collectors with households seamless. ReJunkify also serves as an educational tool, encouraging users to learn about the environmental benefits of their actions.

Resistance to change poses an initial hurdle, as some individuals may be accustomed to traditional waste disposal methods. Overcoming community regulations restricting access for recyclers requires active engagement with community leaders and advocacy for responsible recycling. Addressing technological barriers involves providing user-friendly tutorials and support to ensure accessibility for all. Raising awareness is a challenge, met through a robust marketing strategy leveraging social media and community partnerships. Ensuring the quality of recycling involves clear guidelines and partnerships with reputable facilities, building trust among users.

1.3. Goals and Objectives

ReJunkify's primary goals are to significantly reduce landfill waste, fostering a sense of community engagement and responsibility. The platform aims to streamline the junk collection process, providing economic opportunities for collectors and supporting local economies. Additionally, ReJunkify serves as an educational hub, promoting awareness and understanding of responsible waste disposal practices.

To achieve these goals, ReJunkify has set specific objectives. These include increasing user adoption rates, establishing partnerships with local recycling centers, enhancing app usability based on user feedback, expanding market penetration to additional cities, and implementing a comprehensive training program for recyclers. These objectives are designed to measure and guide the platform's progress in fulfilling its broader mission.

1.4. Literature Review/Existing Solutions

Existing solutions in the waste management app landscape, such as "Recycle Now" and "WasteWizard," have effectively heightened public awareness about recycling practices. These apps prioritize user education, aiding individuals in understanding recyclable items. Noteworthy platforms like "Freecycle" focus on reusability, encouraging users to give away items to minimize waste. Challenges faced by these apps include user resistance to technology adoption and the need for robust educational components. Successful solutions often feature userfriendly interfaces, clear recycling guidelines, and community engagement features. ReJunkify aims to learn from these experiences, prioritizing not only the convenience of junk collection but also integrating educational elements to promote responsible waste disposal within local communities. By addressing challenges encountered by existing solutions and incorporating user feedback, ReJunkify strives to offer a comprehensive and user-centric approach to recycling and waste management.

1.5. Gap Analysis

While existing waste management apps provide valuable insights and solutions, there are notable gaps that ReJunkify aims to address. Many current platforms focus primarily on user education and recycling awareness, often overlooking the critical aspect of junk collection logistics. ReJunkify aims to bridge this gap by placing a strong emphasis on the seamless coordination between households and junk collectors, ensuring an efficient and convenient process for both parties.

Furthermore, there is a gap in the integration of educational features within the collection process. While some apps prioritize user education separately, ReJunkify seeks to embed educational elements directly into the user experience, providing real-time information about the environmental impact of each recycling action.

Community engagement is another area where existing solutions often fall short. ReJunkify aims to go beyond individual actions by fostering a sense of community responsibility, encouraging collective efforts in waste reduction initiatives. Additionally, there is a gap in addressing the challenges posed by community regulations that restrict recyclers' access to certain areas. ReJunkify plans to actively engage with local leaders and communities to navigate and overcome these regulatory hurdles.

By identifying and addressing these gaps, ReJunkify aims to offer a more comprehensive and user-focused waste management solution that not only educates users but also seamlessly facilitates the collection and recycling of household junk while actively involving communities in the process.

1.6. Proposed Solution

ReJunkify makes junk collection easy and informative. You can schedule pickups with a simple tap, making it hassle-free for both homes and collectors. As you recycle, get real-time info on the positive impact you're making, right on the app.

But ReJunkify isn't just about you—it's about your community too. Join hands with neighbors for clean-up events and big projects to reduce waste together. Plus, we're not just an app; we're on your side. If rules make it tricky for collectors, we'll talk to local leaders and sort things out. ReJunkify is here to make recycling a breeze for you and your community.

1.7. Project Plan

Getting Ready (Month 1-2):

Gather a team of treehouse builders (our developers and designers).

Plan what our treehouse (app) will look like and what cool features it'll have. Laying the Foundation (Month 3-4):

Start building the base of our treehouse (develop the app structure).

Test it to make sure it's strong and can hold lots of cool things (junk collection features). Adding Branches (Month 5-6):

Put in branches (features) like easy scheduling and real-time info on recycling impact.

Make sure everything works smoothly like leaves blowing in the wind.

Decorating the Treehouse (Month 7-8):

Design the inside (user interface) to be super easy and fun.

Add educational posters (info pop-ups) about recycling while you play in the treehouse. Inviting Friends (Month 9-10):

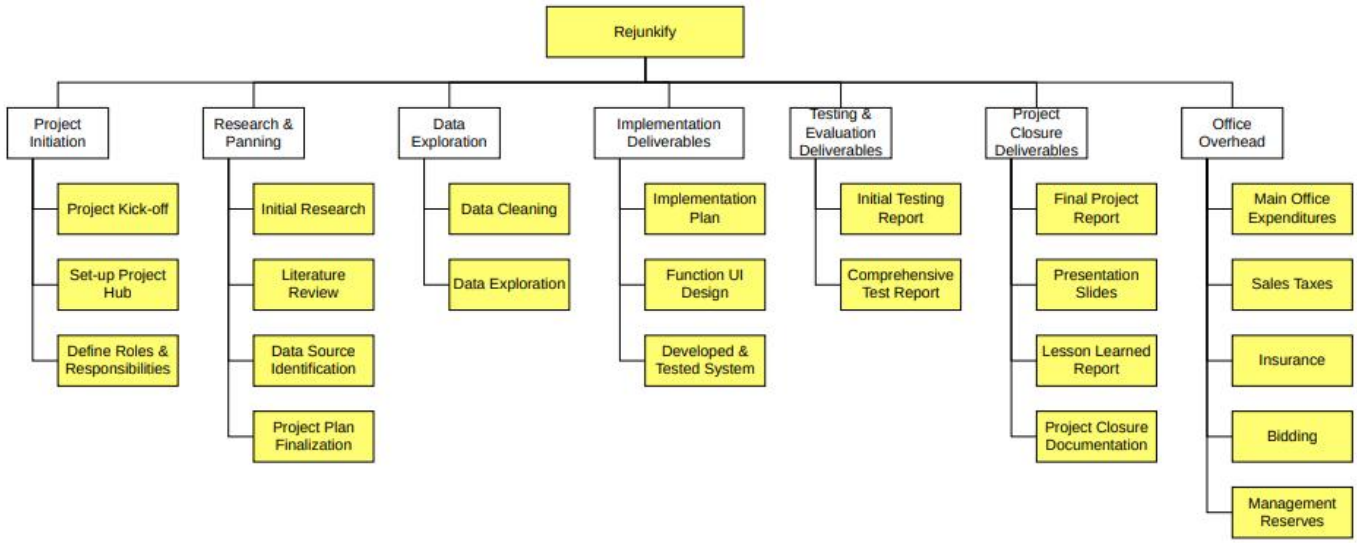
Open the treehouse to everyone (launch the app).

Tell all our friends (marketing) about our cool new treehouse (app). Growing Together (Ongoing):

Listen to what everyone thinks (user feedback).

Keep adding swings and slides (new features) to make the treehouse even more awesome.

1.7.1. Work Breakdown Structure

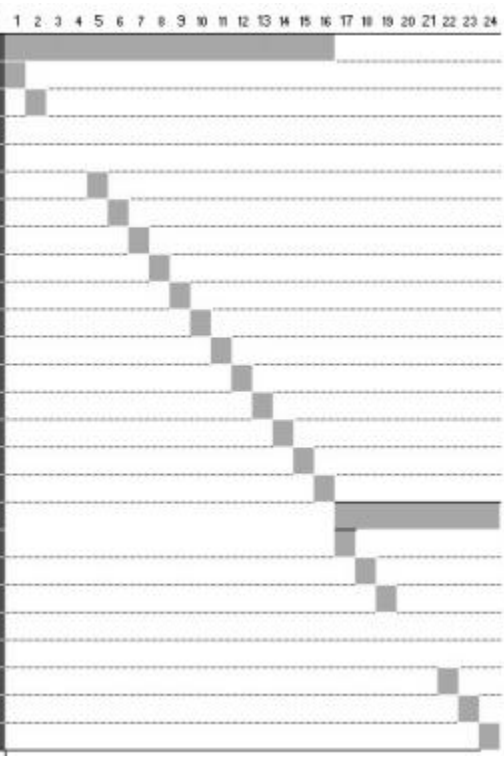


1.7.2. Roles & Responsibility Matrix

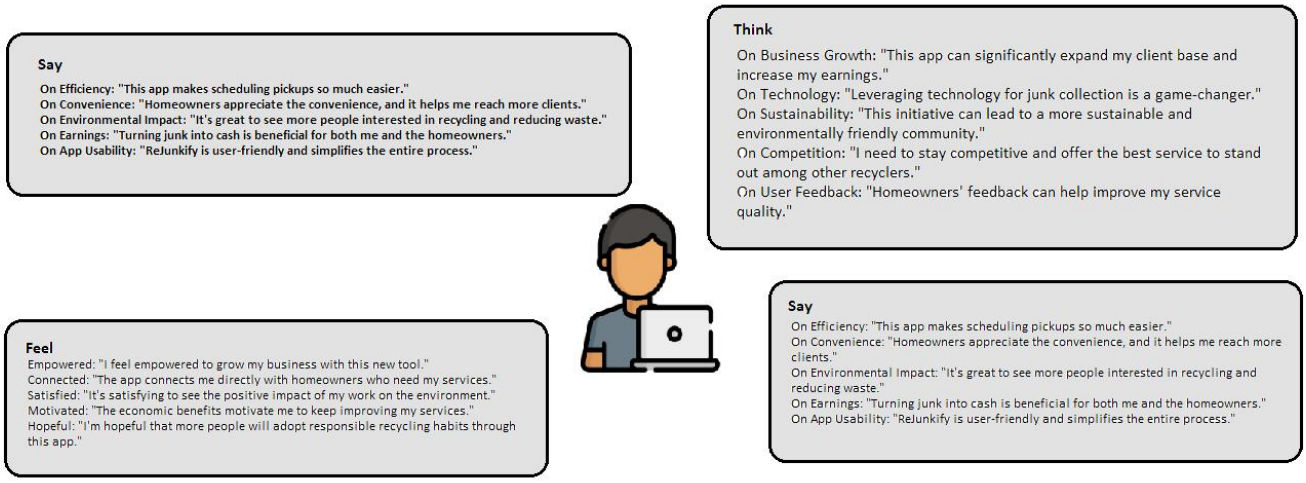
Task	Software Engineer(Abdullah Asim)	Developer(Shahroz & Muaaz)
Kick-off Meeting	Abdullah,Shahroz and Muaaz	Abdullah,Shahroz and Muaaz
Set up ProjectHub	Abdullah	Shahroz & Muaaz
Initial Research	Abdullah	Shahroz & Muaaz
Identify Data Sources	Abdullah	Shahroz & Muaaz
Begin Literature Review	Abdullah	Shahroz & Muaaz
Data Cleaning and Exploration	Abdullah	Shahroz & Muaaz
Disuss Challenges	Abdullah	Shahroz & Muaaz
Finalize Project Plan	Abdullah	Shahroz & Muaaz
Complete Literature Review	Abdullah	Shahroz & Muaaz
Regular Team Check-Ins	Abdullah	Shahroz & Muaaz
Continue Project Report Draft	Abdullah	Shahroz & Muaaz
Intensive Literature	Abdullah	Shahroz & Muaaz
Review Discussions	Abdullah	Shahroz & Muaaz
Begin Implementation	Abdullah	Shahroz & Muaaz
UI Design Discussions	Abdullah	Shahroz & Muaaz
Testing and Evaluation	Abdullah	Shahroz & Muaaz
Comprehensive Testing and Evaluation	Abdullah	Shahroz & Muaaz
Finalize Project Report and Presentations	Abdullah	Shahroz & Muaaz

1.7.3. Gantt Chart

#	Activity			Start	End
	Data Exploration and Planning	08/07/23	03/08/23	1	16
1	Kick-off Meeting	09/07/23	04/08/23	1	1
2	Set up ProjectHub	10/07/23	05/08/23	2	2
3	Initial Research	11/07/23	06/08/23	3	3
4	Identify Data Sources, Begin Lit. Review	12/07/23	07/08/23	4	4
5	Complete Lit. Review	13/07/23	08/08/23	5	5
6	Initial Project Report Draft	14/07/23	09/08/23	6	6
7	Continue Lit. Review Synthesis	15/07/23	10/08/23	7	7
8	Regular Team Check-ins	16/07/23	11/08/23	8	8
9	Continue Project Report Draft	17/07/23	12/08/23	9	9
10	Intensive Lit. Review Discussions	18/07/23	13/08/23	10	10
11	Data Analysis and Visualization	19/07/23	14/08/23	11	11
12	Finalize Project Plan and Report	20/07/23	15/08/23	12	12
13	Begin Implementation	21/07/23	16/08/23	13	13
14	Regular Team Check-ins	22/07/23	17/08/23	14	14
15	UI Design Discussions	23/07/23	18/08/23	15	15
16	Testing and Evaluation	24/07/23	19/08/23	16	16
	Implementation and Testing	25/07/23	20/08/23	17	24
17	Testing and Evaluation	26/07/23	21/08/23	17	17
18	Regular Team Check-ins	27/07/23	22/08/23	18	18
19	UI Design and Refinement	28/07/23	23/08/23	19	19
20	Initial Testing and Evaluation	29/07/23	24/08/23	20	20
21	Continue Implementation	30/07/23	25/08/23	21	21
22	Regular Team Check-ins	31/07/23	26/08/23	22	22
23	Comprehensive Testing and Evaluation	01/08/23	27/08/23	23	23
24	Finalize Project Report and Presentation	02/08/23	28/08/24	24	24



1.9. Empathy Map



Chapter 2
Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

The Rejunkify project strives for an eco-friendly solution connecting individuals disposing of items with recyclers. This Software Requirements Specification (SRS) details the mobile app for easy junk material booking. User-centric, it offers a friendly interface for convenient scheduling of recyclable material pickups. Key features include authentication, material categorization, scheduling, real-time tracking, and payment integration. This SRS comprehensively outlines system requirements, defining the Rejunkify app's scope, functionality, and constraints. This document is crucial for laying the groundwork for the successful development and implementation of Rejunkify

2.1.1. Purpose

ReJunkify is like a helpful friend on your phone that makes cleaning up and recycling super easy! It uses cool technologies called Flutter and Firebase to create a simple app where you can book someone to pick up your junk. The app also teaches you about why recycling is good for the planet while looking really nice with bright colors and clear letters. It wants to bring neighborhoods together to keep things clean, help junk collectors make money, and make recycling a fun and easy thing for everyone.

2.1.2. Document Conventions

The documentation for ReJunkify incorporates a visually engaging design with a yellow color scheme and the use of the Roboto font. This includes a comprehensive Color Scheme Guide detailing the specific shades of yellow employed, coupled with Font Usage Guidelines that outline the consistent application of the Roboto font throughout the app. Brand Guidelines encompass the visual identity, logo, and iconography, offering a cohesive representation. Development Documentation guides developers through the implementation of Flutter and Firebase, ensuring compatibility and providing clarity through inline code comments. A detailed User Guide facilitates user onboarding, explaining functionalities like scheduling junk pickups, accessing real-time education, and engaging with the community. Testing Documentation

encompasses test cases and user feedback analysis, while the Deployment Guide outlines steps for a smooth app release, accompanied by release notes. Marketing and Outreach Documentation include promotional materials and a community engagement strategy, fostering a holistic approach to the ReJunkify project.

2.1.3. Intended Audience and Reading Suggestions

The documentation for ReJunkify caters to a diverse audience involved in various facets of the app's development and promotion. Developers will find valuable insights in the Flutter and Firebase implementation guides, along with detailed code documentation. Designers can refer to the color scheme and font usage guidelines on Figma, gaining an understanding of the visual identity and user interface design. Testers will benefit from comprehensive test cases and user feedback analysis for a thorough understanding of app functionality. Marketing and outreach teams will find promotional materials and community engagement strategies to effectively promote the app. Project managers and stakeholders can rely on the executive summary for an overarching view of the project, explore project goals in the project plan, and delve into postlaunch analysis in the Evaluation and Optimization section. This tailored approach ensures that each audience can easily access relevant information aligned with their specific roles and responsibilities.

2.1.4. Product Scope

ReJunkify is a cutting-edge mobile application designed to streamline and revolutionize waste management practices. With a vibrant yellow color scheme and the contemporary Roboto font, the app's purpose is to connect households seamlessly with junk collectors, promoting responsible waste disposal and recycling. The software aims to educate users in real-time about the environmental impact of their recycling actions while fostering community engagement and advocacy. The key benefits include a user-friendly interface developed with Flutter, integration with Firebase for efficient backend processes, and a visually appealing design on Figma. The

objectives encompass efficient junk collection, real-time user education, community responsibility, and economic opportunities for junk collectors. By aligning with these goals, ReJunkify directly contributes to corporate goals centered on environmental sustainability, community engagement, and technological innovation. The app envisions a cleaner environment, empowered communities, and economic opportunities within the waste management sector, creating a positive impact aligned with broader business strategies. For a more comprehensive understanding, the vision and scope document provide detailed insights into the overarching goals and strategies driving ReJunkify's development.

2.2. Overall Description

2.2.1. Product Perspective

The Rejunkify mobile application represents a pioneering solution in the domain of waste management, serving as a novel addition to existing recycling systems. Originating from the need to streamline and modernize the process of discarding junk materials, our application is a self-contained product designed to facilitate users in booking the collection and recycling of their waste. It serves as an innovative follow-on member of the waste management product family, aiming to bridge the gap between users and recyclers through a user-friendly interface. This mobile application not only acts as a replacement for traditional methods of junk disposal but also enhances the overall efficiency and accessibility of recycling systems. The Rejunkify system integrates seamlessly with existing waste management infrastructures, creating a comprehensive and sustainable solution for users to contribute to environmental conservation conveniently. A system diagram illustrates the major components, subsystem interconnections, and external interfaces, showcasing the holistic approach adopted to redefine the user experience in waste disposal and recycling.

2.2.2. User Classes and Characteristics

Our rejunkify mobile application caters to a diverse range of user classes, each with distinct characteristics and requirements.

1. **Regular Users:** The largest user class, regular users represent individuals who periodically dispose of their recyclable materials through the app. They have varying levels of technical expertise, ranging from tech-savvy users who engage frequently with the application to those less familiar with mobile interfaces. The primary requirement for this user class is an intuitive and user-friendly experience to effortlessly schedule and manage their junk pickups.

2. **Recyclers:** This user class comprises recycling professionals and organizations registered on our platform. Recyclers leverage the app to receive and manage pickup requests, making their operations more streamlined and efficient. These users require a robust and secure system for managing multiple pickups, real-time updates, and a seamless interface for communication with the app's end-users.

3. **Administrators:** As the overseers of the rejunkify platform, administrators possess elevated privileges and access levels. Their responsibilities include monitoring system performance, resolving escalated issues, and ensuring compliance with environmental regulations. Administrators require a comprehensive dashboard for system management, data analytics, and the ability to enforce security protocols.

2.2.3. Operating Environment

The "Rejunkify" mobile application is designed to operate seamlessly within a diverse technological environment, catering to users who seek to responsibly dispose of their junk materials. The application is optimized for use on smartphones running on both Android and iOS operating systems, ensuring accessibility for a broad user base. It leverages the capabilities of modern mobile devices, requiring minimal hardware specifications to guarantee a smooth user experience. "Rejunkify" interfaces efficiently with various versions of Android (from KitKat onwards). The application is designed to coexist harmoniously with standard mobile software components, leveraging user-friendly interfaces and adhering to best practices for mobile application development. Its compatibility ensures a user-friendly experience across a range of

devices and operating system versions, promoting widespread adoption and facilitating the seamless booking of junk material for recycling.

2.2.4. Design and Implementation Constraints

The design and implementation of the Rejunkify mobile application are subject to several constraints that shape the development process. Firstly, adherence to corporate and regulatory policies regarding data privacy and waste management standards imposes stringent security considerations on the application. Additionally, the integration of the platform with recycling facilities necessitates adherence to specific interfaces and protocols for seamless communication. Hardware limitations, particularly on user devices, may influence timing and memory requirements, requiring an optimized design to ensure efficient performance. The project is bound by the choice of technologies and databases suitable for waste management and mobile application development. Compliance with design conventions and programming standards is crucial, especially if the responsibility for maintaining the software is eventually transitioned to the customer's organization. These constraints collectively guide the development team in crafting a robust and compliant mobile application for the streamlined booking and recycling of junk materials.

2.2.5. Assumptions and Dependencies

This project, Rejunkify, relies on several key assumptions and dependencies that may impact the successful implementation of the stated requirements in the Software Requirements Specification (SRS). Firstly, it assumes that users will have reliable and consistent access to mobile devices with internet connectivity to effectively use the Rejunkify mobile application. The project is dependent on the availability and cooperation of recycling facilities and their willingness to participate in the platform. Additionally, the accuracy and timely updating of recycling facility information, such as location and acceptable materials, are assumed to be maintained by third-party sources. The seamless integration of payment gateways for transaction processing is also a critical dependency for the successful execution of the application's functionality. Any changes in the regulatory environment or recycling policies may affect the project's viability, and it is assumed that these factors will remain consistent during the project's development and operational phases. Lastly, the successful implementation of the

Rejunkify mobile application is dependent on the continued compatibility and support of the chosen mobile platforms (iOS and Android) and their respective versions throughout the project lifecycle.

2.3. External Interface Requirements

2.3.1. User Interfaces

The user interface of Rejunkify is thoughtfully designed to provide an intuitive and user-friendly experience for individuals looking to responsibly dispose of their junk materials through our mobile application. The interface comprises a clean and visually appealing design, with sample screen images showcasing the easy navigation process. Users will encounter a streamlined booking system where they can schedule the collection of their recyclable items effortlessly. The graphical user interface (GUI) adheres to industry standards and embraces a modern design language, ensuring consistency across screens. Standard buttons and functions such as 'Book Now,' 'View History,' and 'Help' are prominently displayed for easy access. The layout is designed with efficiency in mind, emphasizing simplicity without compromising on functionality. Error messages are displayed in a clear and concise manner, adhering to established standards, to assist users in resolving any issues seamlessly. The software components that necessitate a user interface include the booking system, user profiles, and a feedback mechanism, all seamlessly integrated to enhance the overall user experience. Further details of the user interface design are outlined in the accompanying user interface specification document.

2.3.2. Hardware Interfaces

In our Rejunkify project, the hardware interfaces play a pivotal role in facilitating seamless communication between the software application and the associated hardware components of the system. The primary interface involves the integration of mobile devices, including smartphones and tablets, as user interfaces for the application. These devices act as the conduits through which users book their junk material pickups. The logical characteristics of this interface encompass user input functionalities such as touch-based interactions and GPS capabilities for location services. Moreover, the physical characteristics involve the utilization of device sensors for accurate data inputs. The communication protocols employed leverage standard wireless technologies such as Bluetooth and internet connectivity to transmit user

requests and relevant data securely to the designated recyclers. This dynamic interaction ensures efficient coordination between the software and the hardware, enabling a user-friendly and effective junk material booking experience through our mobile application.

2.3.3. Software Interfaces

The Rejunkify mobile application interfaces with various software components to facilitate seamless booking and recycling processes. The application relies on a backend database, powered by MySQL version 8.0, to store and retrieve user information, junk materials data, and recycling schedules. Operating on both Android and iOS platforms, the app leverages native development tools such as Android Studio and Xcode. Additionally, it integrates with Google Maps API (version 3.0) for location services, enabling users to input and visualize their pickup locations. Communication between the mobile application and the backend server occurs through RESTful APIs, employing JSON as the data interchange format. The application utilizes Retrofit library (version 2.9) for efficient HTTP communication. Data items such as user credentials, junk material details, and pickup preferences are transmitted to the server, while notifications and status updates are pushed back to the user interface. The integration ensures a secure and responsive platform for users to seamlessly book and manage their junk recycling services.

2.3.4. Communications Interfaces

The communication interfaces of the ReJunkify mobile application are designed to seamlessly connect users with recycling services, ensuring efficient and secure interactions. The system incorporates email notifications to keep users informed about their junk material pickup schedules, providing a convenient way to stay updated on the recycling process. Additionally, the application relies on a web browser interface for users to access detailed information, track their recycling history, and manage account preferences. The communication protocols are built on secure HTTP connections, prioritizing data encryption to safeguard sensitive user information during the submission and processing of junk material requests. To maintain optimal performance, the application adheres to industry-standard data transfer rates and incorporates synchronization mechanisms to ensure real-time updates on the status of recycling requests. The utilization of electronic forms within the application streamlines the

booking process, allowing users to input necessary details easily. Overall, the ReJunkify communication interfaces prioritize user convenience, data security, and seamless interactions with the recycling service.

2.4. System Features

The Rejunkify mobile application boasts a comprehensive set of system features designed to streamline and enhance the user experience in booking and recycling junk materials. The primary functionality revolves around an intuitive and user-friendly interface that allows users to effortlessly schedule pickups for their unwanted items. The system features include a secure user authentication process, enabling registered users to create and manage their accounts seamlessly. The application supports a dynamic booking system, providing users with the flexibility to choose specific time slots for junk removal. Additionally, an integrated payment gateway ensures a smooth and secure transaction process. The system employs real-time tracking, allowing users to monitor the status of their booked pickups. Furthermore, a robust notification system keeps users informed about the progress of their recycling request. The Rejunkify app also incorporates a comprehensive catalog of recyclable items and provides users with relevant information on sustainable disposal practices.

2.4.1. System Feature

2.4.1.1. Description and Priority

Rejunkify is an innovative mobile application designed to revolutionize waste management by seamlessly connecting users with recyclers. The key feature of the application allows users to effortlessly schedule the pickup of their junk materials, facilitating a convenient and ecofriendly disposal process. Through the intuitive interface, users can book appointments, specify the type and quantity of recyclables, and receive real-time updates on the recycling status. This not only streamlines the recycling process but also promotes sustainability by encouraging responsible waste disposal.

2.4.1.2. Stimulus/Response Sequences

Users initiate the process by opening the Rejunkify mobile application and selecting the "Book Pickup" option. The system responds with a user-friendly interface prompting them to input details such as their location, the type and quantity of junk materials, and preferred pickup time. Upon submitting this information, the system validates and confirms the booking, providing users with a confirmation message and a unique booking reference number. To enhance user engagement, the system sends timely push notifications, reminding users of the scheduled pickup and offering recycling tips. Users can also track the real-time location of the assigned recycling vehicle through the application, promoting transparency and convenience. Once the pickup is completed, the system generates a digital receipt and sends it to the user, along with an eco-friendly impact report, showcasing the positive environmental contribution of their recycled materials. This seamless and interactive process encourages users to actively participate in sustainable practices while utilizing the Rejunkify platform.

2.4.1.3. Functional Requirements

Functional Requirements for Junk Material Booking Feature:

REQ-RJF1-1: The mobile application shall provide a user-friendly interface allowing users to log in or create an account.

REQ-RJF1-2: Upon successful login, users should be able to input details about the junk materials they wish to dispose of, including type, quantity, and preferred pickup time.

REQ-RJF1-3: The system shall validate user inputs, ensuring that all mandatory fields are filled, and provide appropriate error messages for any missing or incorrect information.

REQ-RJF1-4: Users should have the capability to view a summary of their booking request before confirming, with the option to make any necessary changes.

REQ-RJF1-5: Once the user confirms the booking, the system shall generate a unique confirmation ID and provide an acknowledgment to the user.

REQ-RJF1-6: The application should allow users to track the status of their booked junk material pickup, providing real-time updates on the recycler's location and estimated arrival time.

REQ-RJF1-7: In case of any errors or disruptions in the pickup process, the system shall notify the user promptly and provide assistance options.

REQ-RJF1-8: Users shall have the ability to cancel a booking within a reasonable time frame before the scheduled pickup, with the system providing confirmation of the cancellation.

REQ-RJF1-9: The application shall maintain a record of past bookings and provide users with access to their booking history.

2.4.2. System Feature

2.4.2.1. Description and Priority

Description:

Rejunkify, our innovative mobile application, revolutionizes waste management by providing users with a seamless platform to book the collection of their junk materials and connect with recyclers effortlessly. Users can easily schedule pick-ups for various recyclable items, promoting environmental sustainability and responsible waste disposal practices. The application streamlines the process, allowing users to contribute actively to recycling efforts with just a few taps on their mobile devices.

Priority:

The feature of booking junk material pickups through the Rejunkify mobile application is of High Priority. The benefits are substantial, including the promotion of eco-friendly practices, reduction of landfill waste, and the facilitation of a more sustainable lifestyle. The potential penalties for not implementing this feature promptly involve missed opportunities to engage users in environmentally conscious behaviors. While the initial development cost is moderate, the long-term benefits far outweigh the investment. The associated risks, such as user adoption challenges, are relatively low given the growing awareness and concern for environmental issues, making this feature a pivotal and high-impact component of the Rejunkify project.

2.4.2.2. Stimulus/Response Sequences

User Action: User launches the Rejunkify mobile application and navigates to the "Book Junk Pickup" feature.

System Response: The application prompts the user to log in or create an account if they haven't already. Once logged in, the user is presented with a user-friendly interface displaying options for selecting the type and quantity of junk they wish to recycle.

User Action: The user selects the appropriate categories for their junk items, specifying quantities and any additional notes.

System Response: The application generates a real-time summary of the selected items, providing an estimated recycling cost and available time slots for pickup. The user can review and confirm their booking.

User Action: Upon confirmation, the user proceeds to the payment section to complete the transaction securely.

System Response: The application processes the payment and sends an instant confirmation, along with details of the booked pickup, including the scheduled time and a unique reference number.

User Action: On the scheduled day, the user prepares the designated junk materials for pickup.

System Response: The application sends a reminder notification to the user, confirming the scheduled pickup and providing details of the assigned Rejunkify collection team.

User Action: The user acknowledges the notification and places the designated junk materials outside for collection.

System Response: The Rejunkify collection team updates the user on their way via the application, ensuring a seamless and transparent process. After the successful pickup, the user receives a notification confirming the completion of the recycling process, along with an ecofriendly impact report showcasing the positive environmental contribution of their actions.

2.4.2.3 Functional Requirements

REQ-BJMUP-1: The application shall provide a user-friendly interface allowing users to input details of the junk materials they want to recycle, including type, quantity, and preferred pickup date.

REQ-BJMUP-2: Upon submission of the junk material booking request, the system shall generate a confirmation notification to the user, providing a summary of the booked pickup, including date, time, and a unique reference number.

REQ-BJMUP-3: The application shall validate user inputs for the junk material details, ensuring the accuracy and completeness of information. In case of invalid or incomplete inputs, appropriate error messages shall be displayed to guide users in correcting the information.

REQ-BJMUP-4: The system shall integrate with the recycling service provider's schedule and logistics system to check the availability of pickup slots for the specified date. If no slots are available, the system shall inform the user and suggest alternative dates.

REQ-BJMUP-5: Users shall have the ability to view and modify their booked pickups, including changing the pickup date or canceling the request. The system shall update the schedule and notify the user of any changes.

REQ-BJMUP-6: In the event of a canceled pickup or rescheduled date, the system shall automatically notify the user and provide reasons for the change.

REQ-BJMUP-7: The application shall securely store and manage user booking history, allowing users to review past junk material pickups and associated details.

REQ-BJMUP-8: The system shall have mechanisms in place to handle and recover from unexpected errors or interruptions, ensuring the reliability and availability of the junk material booking functionality.

2.5. Nonfunctional Requirements

2.5.1. Performance Requirements

The performance requirements for the Rejunkify mobile application are designed to ensure a seamless and efficient user experience, facilitating the booking of junk material pickups in a timely manner. The system must handle concurrent user requests without degradation in response time, aiming for a maximum response time of three seconds for any user action. The booking confirmation process should be completed within 15 seconds to enhance user satisfaction. In the case of simultaneous booking requests, the system should prioritize based on proximity and real-time availability of the recycling service. For real-time updates on the

recycling truck's location, the application must provide location data with a refresh rate of at least 10 seconds. To minimize downtime, the application should have a system uptime of 99.5%, ensuring users can access and utilize the service reliably. These performance requirements are essential to guarantee a smooth and responsive experience for users engaging with the Rejunkify application, aligning with our commitment to efficiency and customer satisfaction.

2.5.2. Safety Requirements

Safety is of paramount importance in the Rejunkify mobile application to ensure the well-being of users and the environment. To address potential risks associated with the use of the application, strict safety requirements are implemented. Users are required to adhere to guidelines for proper handling and packaging of junk materials to mitigate any risk of injury during transportation. The application incorporates secure payment gateways to safeguard financial transactions, and user data is encrypted to protect privacy. Additionally, Rejunkify complies with environmental regulations for the proper disposal and recycling of materials, aligning with waste management policies. To ensure user safety, the application emphasizes the use of certified recycling facilities, and users are cautioned against submitting hazardous materials. Moreover, Rejunkify adheres to industry safety standards and certifications to guarantee the reliability and security of the entire junk material booking process. It is essential for users to familiarize themselves with these safety requirements outlined in the application to promote a secure and responsible recycling experience.

2.5.3. Security Requirements

Security is a paramount concern for the Rejunkify project, which facilitates users in booking junk material pickups through a mobile application. To ensure the protection of user data and privacy, the application must implement stringent security measures. This includes robust user identity authentication requirements to prevent unauthorized access to sensitive information. The system should adhere to industry-standard encryption protocols to secure data transmission and storage. Compliance with relevant external policies and regulations, such as data protection laws and recycling industry standards, is imperative. Additionally, the project

must adhere to any security or privacy certifications applicable to the handling of user data and transactions, demonstrating a commitment to safeguarding user information and maintaining the trust of both recyclers and users alike.

2.5.4. Usability Requirements

The usability requirements of the Rejunkify project are centered on creating an intuitive and seamless experience for users who are booking junk material pickups through the mobile application. The application should feature a user-friendly interface, allowing users to easily navigate through the various features such as material selection, date and time preferences, and location details. Clear and concise instructions should guide users through the booking process, ensuring minimal effort is required to complete a transaction. The application must also be responsive and compatible across different devices and screen sizes to accommodate a diverse user base. Additionally, an efficient and transparent communication system should be integrated to keep users informed about the status of their bookings, providing real-time updates and notifications. Overall, the usability requirements aim to enhance user satisfaction, streamline the booking process, and encourage frequent and straightforward interaction with the Rejunkify mobile application.

2.5.5. Reliability Requirements

Reliability is paramount in the success of Rejunkify, the mobile application designed for users to effortlessly book their junk material pickups with recyclers. To ensure a dependable user experience, the application must exhibit robust and consistent performance in processing and confirming booking requests. The reliability requirements encompass seamless integration with the recycler's scheduling system to accurately allocate time slots for pickups. The application should also feature real-time updates and notifications to keep users informed about the status of their junk material collection, ensuring transparency and trust. Moreover, a reliable data storage and retrieval mechanism is essential to maintain the integrity of user profiles, booking history, and transaction records. System downtime must be minimized through effective error handling and recovery strategies, guaranteeing uninterrupted service for users and recyclers

alike. Overall, the reliability requirements of Rejunkify are designed to instill confidence in users, fostering a dependable and efficient platform for sustainable waste management.

2.5.6. Maintainability/Supportability Requirements

Maintainability and supportability are pivotal aspects of the Rejunkify project, ensuring the seamless operation and longevity of the mobile application. To guarantee ease of maintenance, the codebase has been meticulously organized and documented, facilitating efficient updates and future enhancements. The implementation of modular and well-commented code not only eases troubleshooting but also enables new team members to swiftly comprehend and contribute to the project. Regular training sessions for support staff are integrated into the plan, ensuring that they remain proficient in handling user queries and technical issues. Moreover, the application incorporates automatic update mechanisms to streamline the deployment of bug fixes and feature enhancements, promoting a user-friendly and continuously improving experience. The overarching goal is to create a sustainable and adaptable platform that evolves alongside technological advancements and user requirements, fostering a reliable and enduring solution for junk material recycling.

2.5.7. Portability Requirements

The portability requirements of our "Rejunkify" mobile application are designed to ensure seamless functionality across various devices and operating systems, promoting accessibility and convenience for users. The application will be developed with cross-platform compatibility, ensuring it runs smoothly on both iOS and Android devices. The user interface will be responsive, adapting to different screen sizes and resolutions to provide a consistent and userfriendly experience across smartphones and tablets. Furthermore, the application will prioritize efficient data transfer, optimizing for varying network conditions to guarantee reliable performance regardless of the user's location or internet connectivity. These portability measures aim to empower users to effortlessly book their junk material for recycling from any device, enhancing the overall accessibility and usability of the "Rejunkify" platform.

2.5.8. Efficiency Requirements

The efficiency requirements of our "Rejunkify" project are designed to ensure a seamless and prompt experience for users engaging with the mobile application. The booking process for junk material pick-up is optimized for efficiency, with a user-friendly interface that allows swift selection and scheduling of recycling services. The application is engineered to handle a high volume of simultaneous bookings, ensuring that users can easily secure slots for recyclable material collection without delays. Real-time updates and notifications provide users with instant confirmation of their bookings and timely alerts on the status of the recycling process. Additionally, the backend system is streamlined for optimal resource allocation, minimizing wait times and maximizing the overall efficiency of the recycling workflow. Through these efficiency requirements, "Rejunkify" aims to enhance user satisfaction and contribute to a sustainable and timely waste management solution.

2.6. Domain Requirements

In addition to the core functionalities outlined in the Software Requirements Specification (SRS), the Rejunkify mobile application introduces specific domain requirements crucial for the seamless functioning of the platform. Database requirements include efficient storage and retrieval mechanisms for user profiles, transaction history, and recycling center details, ensuring data integrity and optimal performance. Given the potential international user base, internationalization requirements mandate support for multiple languages and regional settings to enhance user accessibility. Legal requirements encompass compliance with environmental regulations, data protection laws, and adherence to waste disposal guidelines, reinforcing the ethical and legal aspects of the recycling process facilitated by the application. Furthermore, the project emphasizes the importance of code reusability to facilitate future feature expansions and modifications, aligning with our long-term objectives for sustained project growth. These domain requirements collectively contribute to the comprehensive and responsible functionality of the Rejunkify mobile application.

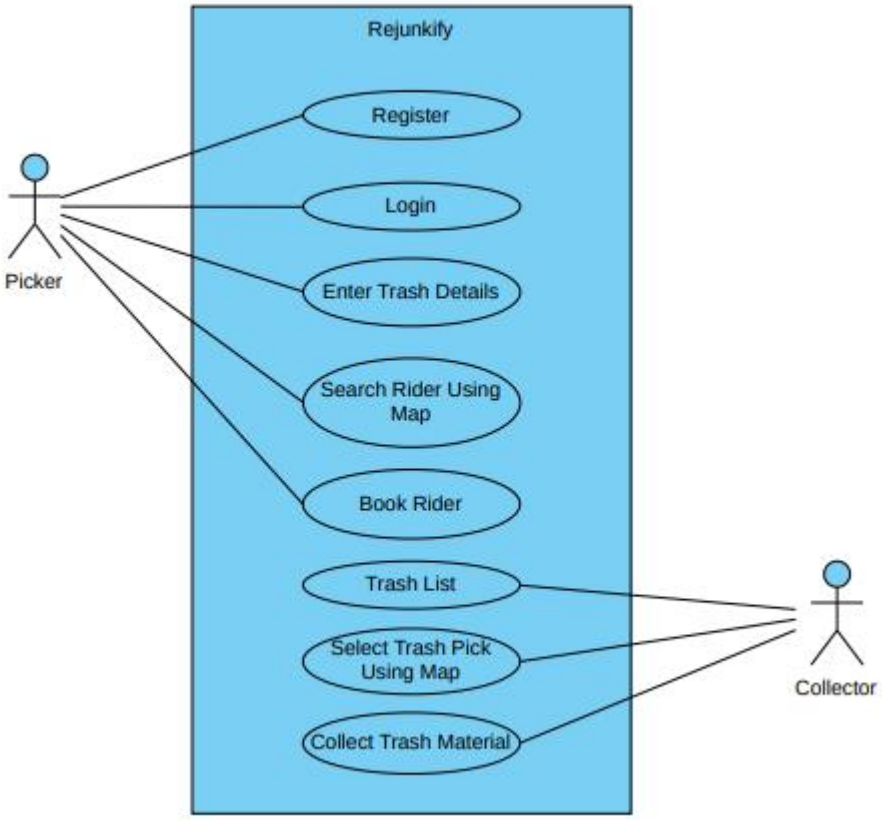
Chapter 3
Use Case Analysis

Chapter 3: Use Case Analysis

Use case analysis is a technique used to identify the requirements of a system (normally associated with software/process design) and the information used to both define processes used and classes (which are a collection of actors and processes) which will be used both in the use case diagram and the overall use case in the development or redesign of a software system or program.

The use case analysis is the foundation upon which the system will be built

3.1. Use Case Model



3.2. Use Cases Description

Use Case ID	RE-UC-001
Use Case Name	Login
Pre-Condition	User has to have a valid account.
Description	This use case allow user to login into the system to access relevant function.
Actor	User
Post-Condition	The system displays the relevant homepage

Use Case ID	RE-UC-002
Use Case Name	Take customer order
Pre-Condition	Rider must login successfully. User must have a account.
Description	This use case allow rider to take order from customer by accessing it using map feature.
Actor	Recycler/Rider
Post-Condition	The system creates a new order for rider.

Chapter 4
System Design

Chapter 4: System Design

In sharing how our Junk Recycling App works, we've made sure it's easy and friendly for everyone. First off, we put a lot of thought into keeping everything smooth and simple for users. We take your security seriously – making sure only the right people get access to the app. Our app is like your helpful guide. You can easily put up items to recycle, and we sort them out neatly. If you need to schedule a pickup, no worries – we've got real-time scheduling and maps to show where things are happening. Payments are a breeze with our safe system, and taking pictures of your recyclables is super easy with our in-app tool. We keep things organized and keep you updated with a tagging system and notifications. We also like hearing from you, so we keep a record of all your activities and let you share your thoughts on recyclers. We speak your language too – the app can talk in different languages. Even if your internet is slow or off, our app still works. Our app looks good on all devices, and we're connected to social media to keep you in the loop. There's a lot more to come, like special offers, ways for you to tell us what you think, and a place for everyone to chat in a community forum. And of course, we make sure everything is safe and secure for you and everyone using our app.

4.1. Architecture Diagram

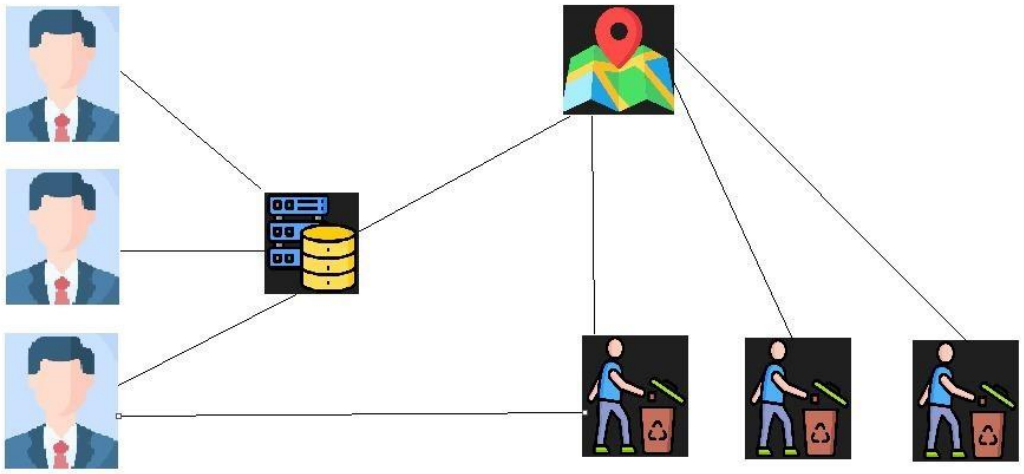
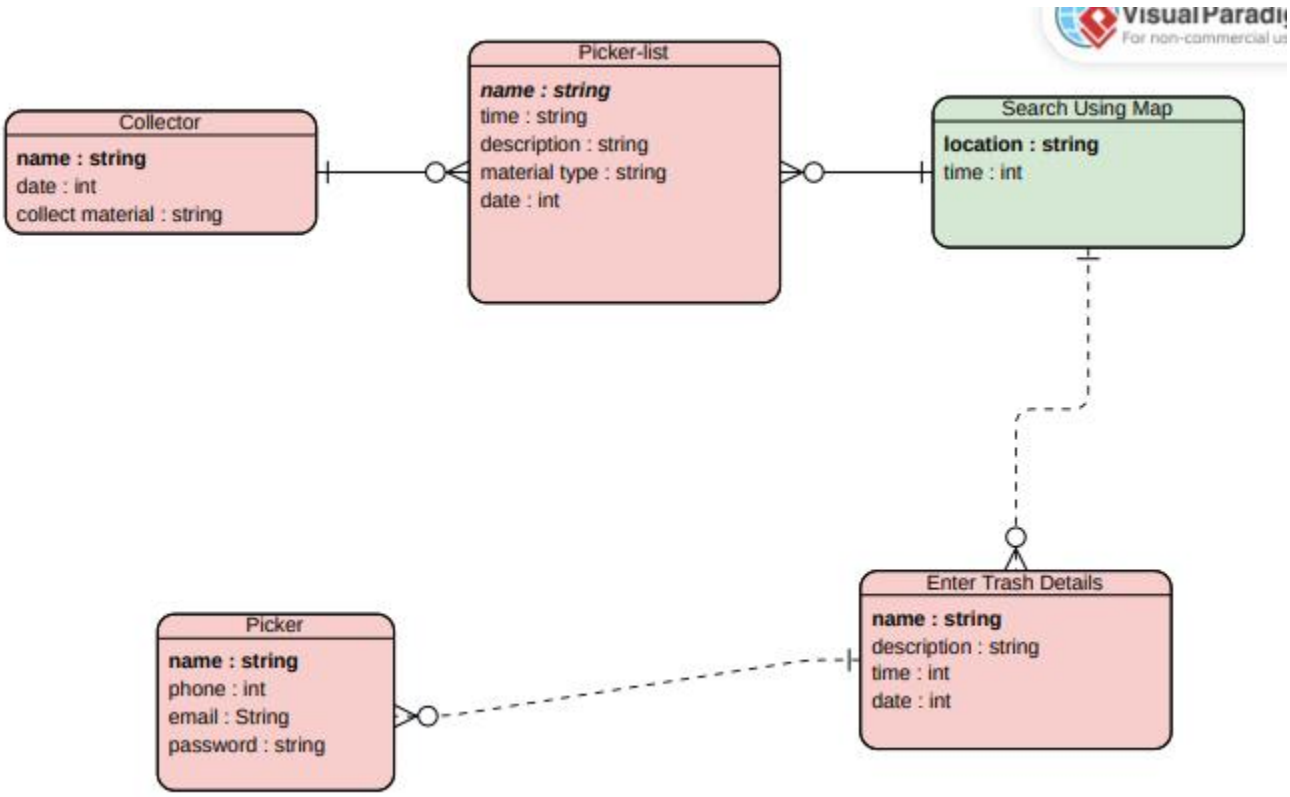
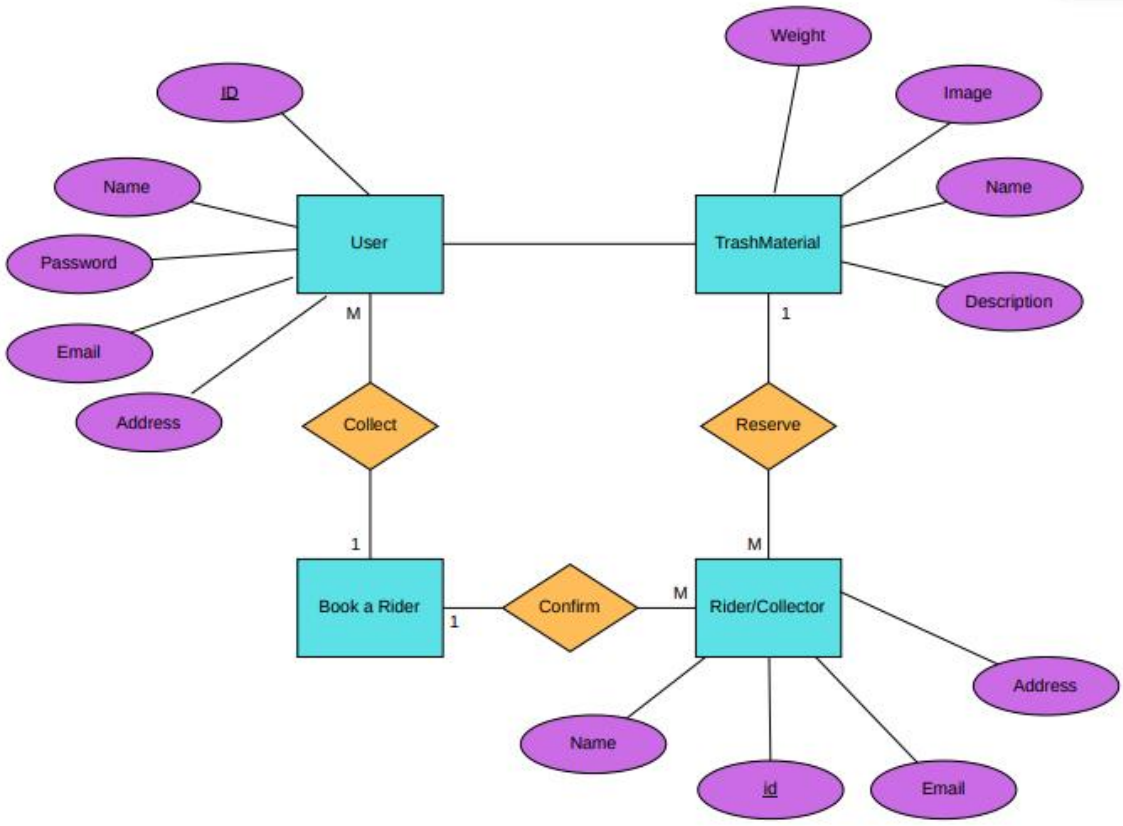


Figure 3 Architecture Diagra

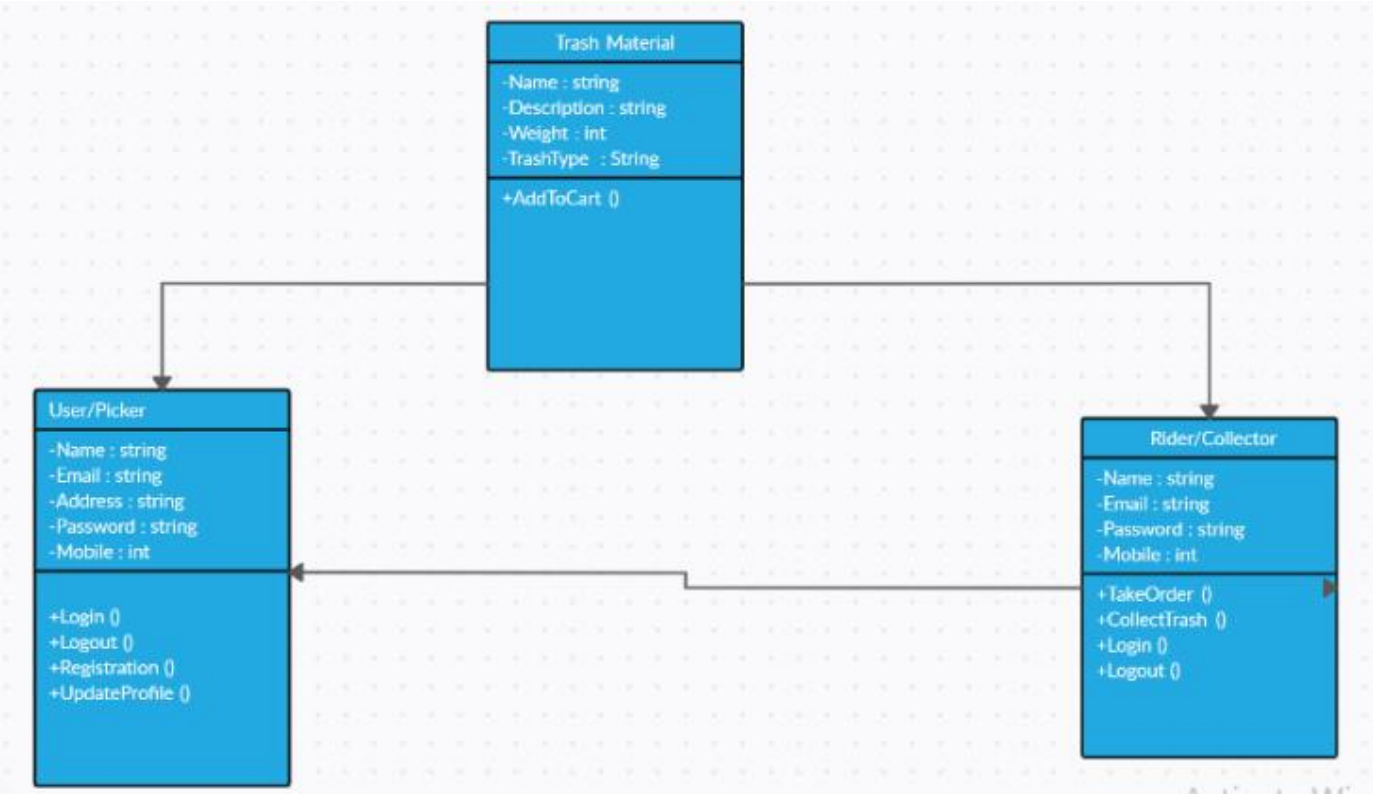
4.2. Domain Model



4.3. Entity Relationship Diagram with data dictionary



4.4. Class Diagram



4.5. Sequence / Collaboration Diagram

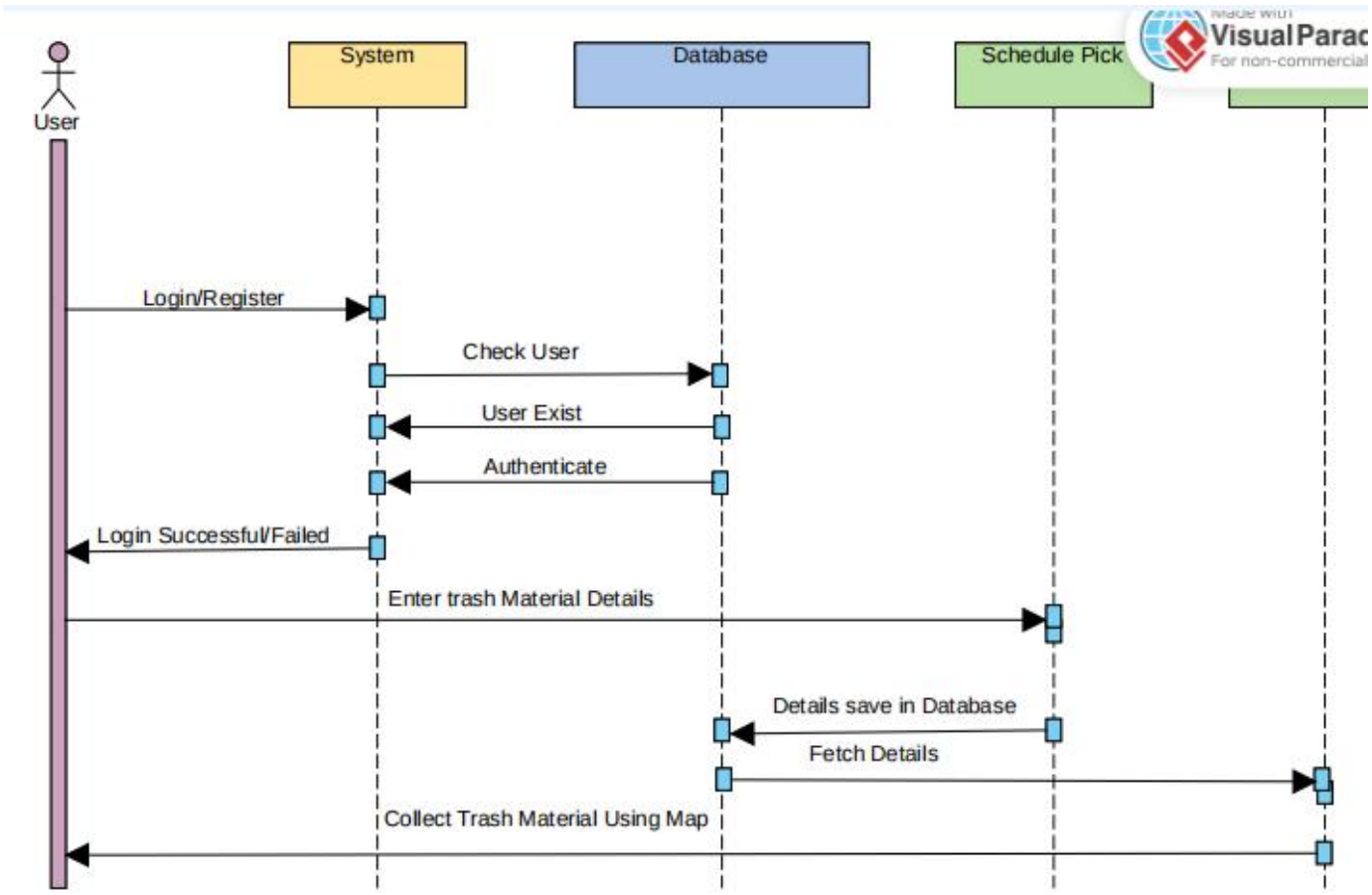


Figure 5 Sequence Diagram

4.6. Operation contracts

Operation: login(username: String, password: String)

Preconditions:

Username and password are provided.

Postconditions:

If credentials are correct, the user is authenticated and logged in.

If credentials are incorrect, an authentication failure is returned.

Exceptions:

UserNotFoundException: If the username does not exist.

AuthenticationException: If the password is incorrect.

Operation: register(username: String, password: String, email: String)

Preconditions:

Username, password, and email are provided.

Username is unique.

Postconditions:

A new user account is created with the provided credentials.

The user is logged in after successful registration.

Exceptions:

UserAlreadyExistsException: If the username is already taken.

InvalidInputException: If the input data is invalid.

Junk Collection and Recycling Services

Operation: schedulePickup(date: Date, address: String)

Preconditions:

Date and address for pickup are provided.

The user is logged in.

Postconditions:

A pickup request is scheduled with the specified date and address.

The request is sent to registered junk collectors for confirmation.

Exceptions:

InvalidDateException: If the provided date is invalid.

PickupRequestException: If there is an issue with scheduling the pickup.

Environmental Impact Tracking

Operation: getEnvironmentalImpact()

Preconditions:

The user is logged in.

Postconditions:

Environmental impact data related to the user's recycling activities is retrieved.

Exceptions:

ImpactDataNotAvailableException: If environmental impact data is not available.

DataRetrievalException: If there is an issue retrieving environmental impact data.

Community Engagement and Events

Operation: organizeCommunityEvent(eventName: String, eventDate: Date, description: String)

Preconditions:

Event name, date, and description are provided.

The user is logged in as an admin or organizer.

Postconditions:

A community event is organized with the specified details.

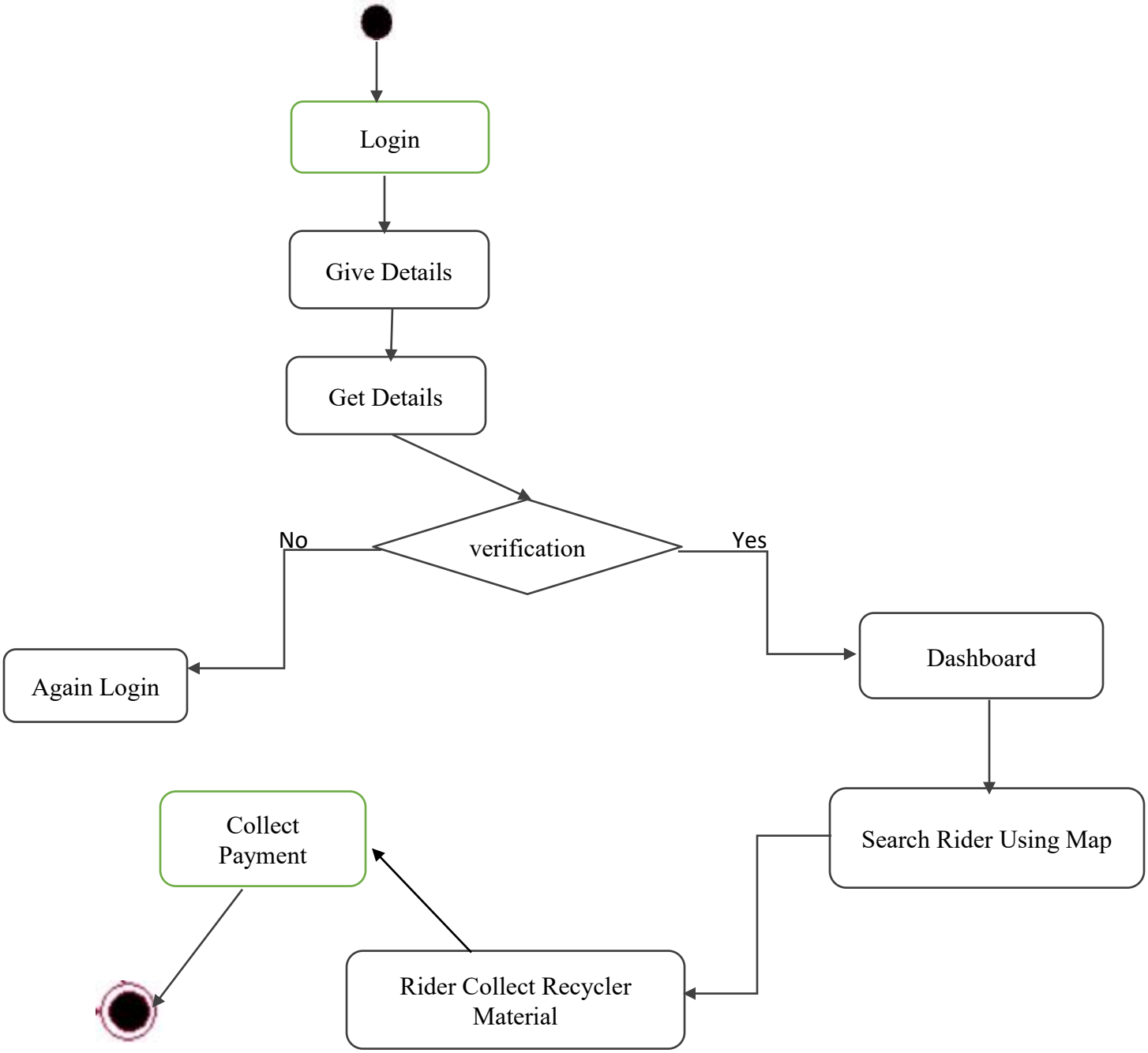
Event details are published and shared with ReJunkify users.

Exceptions:

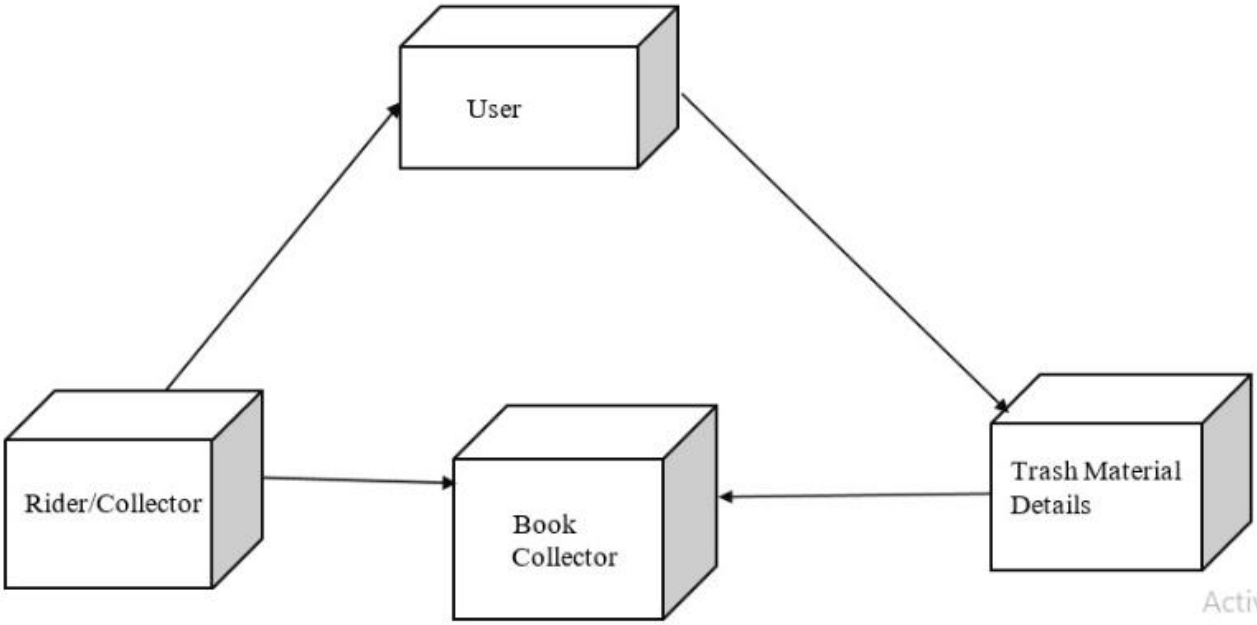
InvalidEventDetailsException: If provided event details are incomplete or invalid.

EventOrganizingException: If there is an issue organizing the community event.

4.7. Activity Diagram

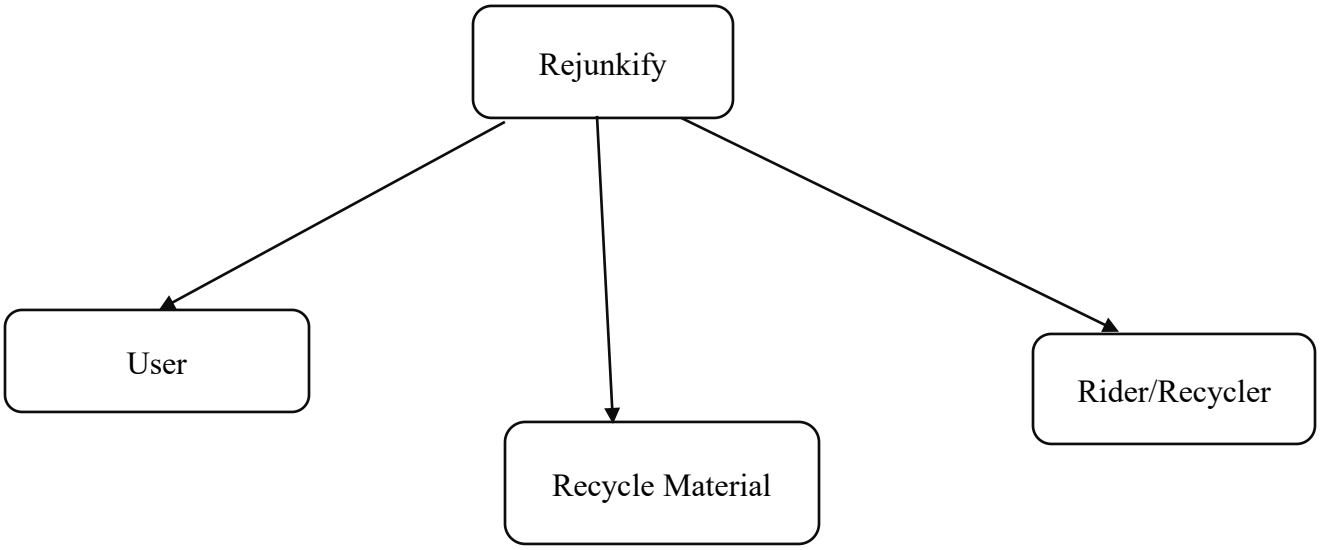


4.8. State Transition Diagram

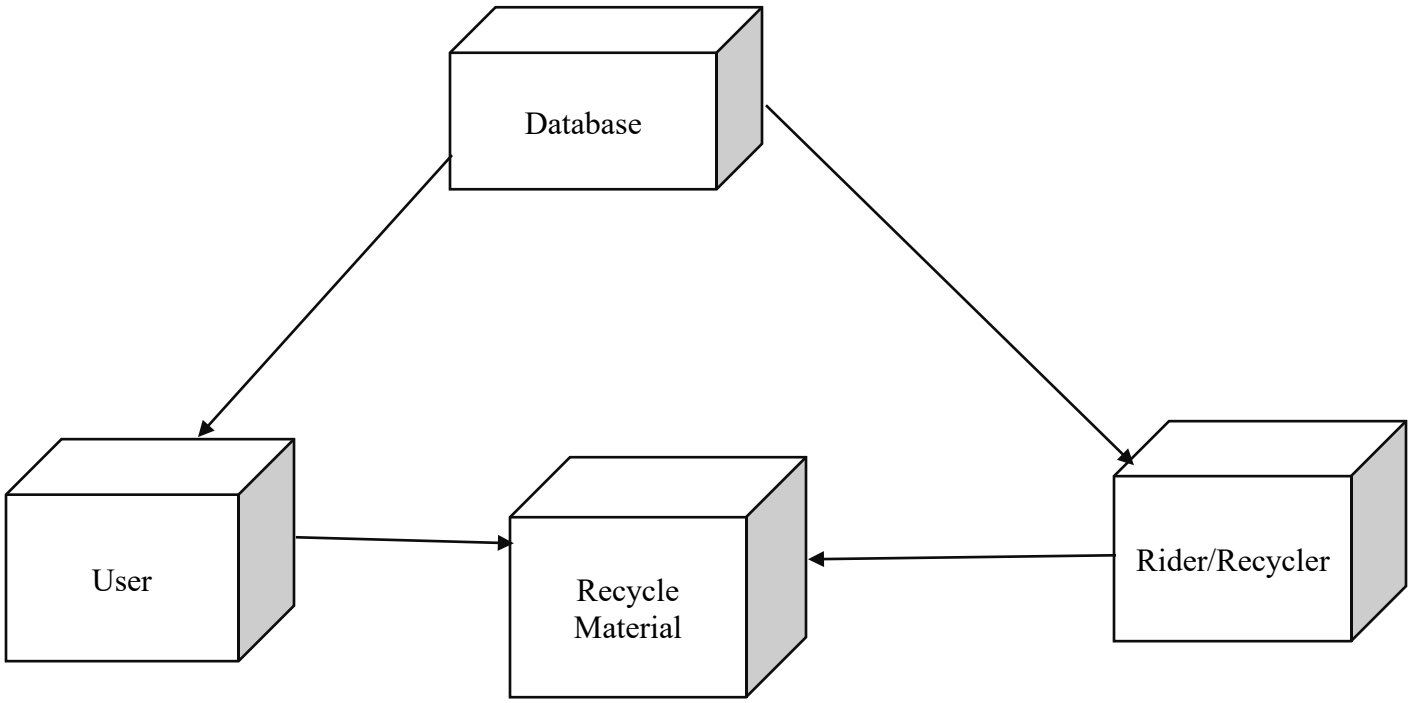


Activate
Go to Settir

4.9. Component Diagram



4.10. Deployment Diagram



s

Chapter 5
Implementation

Chapter 5: Implementation

Our project, implemented with Flutter, revolutionizes waste management by introducing the Rejunkify mobile application. Users can seamlessly book the collection of their recyclable materials through the app, streamlining the process of connecting individuals with local recyclers. The Flutter-based interface ensures a user-friendly experience, allowing clients to schedule pickups, track their recycling history, and contribute to sustainable practices with ease. With real-time notifications and an intuitive design, Rejunkify promotes environmental consciousness while utilizing the power of technology to simplify and enhance the recycling process.

5.1. Important Flow Control/Pseudo codes

1. User Authentication:

- Pseudo-code for user authentication:

Dart:

```
// Check if the user is already authenticated
if (userIsAuthenticated()) {
  // User is authenticated, proceed to the main screen
navigateToMainScreen();
} else {
  // User is not authenticated, navigate to the authentication screen
navigateToAuthenticationScreen();
}
```

2. Main Screen:

- Display options for the user, such as booking a junk pickup, viewing past bookings, etc.

Dart:

```
// Main screen UI
Widget buildMainScreen()
{ return Scaffold(
appBar: AppBar(title: Text('Rejunkify')),
body: Column( children:
[ ElevatedButton(
onPressed: () => navigateToBookingScreen(),
child: Text('Book Junk Pickup'),
),
ElevatedButton(
onPressed: () => navigateToPastBookings(),
child: Text('View Past Bookings'),
),
],
),
);
}
```

3. Booking Screen:

- Allow the user to select the type of junk, quantity, and schedule a pickup.

Dart:

```
// Booking screen UI
Widget buildBookingScreen()
{ return Scaffold(
appBar: AppBar(title: Text('Book Junk Pickup')),
body: Column( children: [
// UI components for selecting junk type, quantity, etc.
// ...
ElevatedButton(
```

```
          onPressed:  ()  =>  submitBookingRequest(),
child: Text('Submit Booking'),
          ),
        ],
      ),
    );
  }
```

4. Booking Logic:

- Pseudo-code for handling the booking logic:

Dart:

```
// Function to submit a booking request
void submitBookingRequest() {      //
Validate user inputs      if
(inputsAreValid()) {
      // Create a booking request object
      BookingRequest booking = createBookingObject();

      //      Send      the      booking      request      to      the      server
sendBookingRequestToServer(booking);

      //      Show      a      confirmation      message      to      the      user
showConfirmationDialog();
    } else {
      // Show an error message to the user
      showErrorDialog();
    }
  }
```

5. Past Bookings Screen:

- Display a list of past booking history for the user.

Dart:

```
// Past bookings screen UI
Widget buildPastBookingsScreen(List<Booking> pastBookings)
{ return Scaffold(
appBar: AppBar(title: Text('Past Bookings')),
body: ListView.builder( itemCount:
pastBookings.length, itemBuilder: (context,
index) { return ListTile(
title: Text('Booking #${pastBookings[index].bookingId}'),
// Display more details or actions if needed
);
},
),
);
}
```

5.2. Components, Libraries, Web Services and stubs

1. User Authentication:

- Firebase Authentication: For secure user authentication.

2. UI/UX Design:

- Flutter Material Components: Default set of Material Design widgets.
- Flutter Cupertino: If you prefer an iOS-style design.
- Flutter SVG: For displaying SVG images/icons.

3. Maps Integration:

- Google Maps Flutter: For integrating maps and location services.
- Geolocator: To get the user's current location.

4. Database:

- Firebase Firestore/Realtime Database: For storing user data, booking history, and other information.
- SQLite: For local storage of essential data on the user's device.

5. State Management:

- Provider or Riverpod: For managing app-wide state.
- Bloc: For implementing the Business Logic Component pattern.

6. Networking:

- Dio or HTTP: For making API calls to communicate with the server.

7. Image Handling:

- CachedNetworkImage: For efficiently loading and caching network images.
- ImagePicker: For capturing or selecting images from the device.

8. Forms and Validation:

- Flutter Form and TextFormField: For creating input forms.
- Formik (if needed): For more advanced form management.

9. Notifications:

- Firebase Cloud Messaging (FCM): For push notifications.

10. Dependency Injection:

- GetX or Provider: For managing and injecting dependencies.

11. Animations:

- Flutter Animation: For creating smooth and appealing animations.

12. Testing:

- Flutter Test: For writing unit and widget tests.

13. Logging and Analytics:

- Firebase Analytics: For tracking user behavior.
- Logger: For efficient logging during development.

14. Payment Integration:

- Stripe or Braintree: For processing payments.

15. Internationalization:

- Flutter Intl: For managing and displaying localized strings.

16. Backend/API:

- Node.js with Express (or any preferred backend technology): For handling requests and managing the database.

5.3. Deployment Environment

Frontend:

- **Framework:** Flutter
- **Platform:** Cross-platform (iOS and Android)
- **UI/UX Design**: Material Design for Android and Cupertino for iOS

Backend:

- **Server-side Framework:** Django or Flask (Python)
- **Database:** PostgreSQL or MongoDB

- **API Integration:** RESTful APIs for communication between the mobile app and server

Authentication:

- **User Authentication:** Firebase Authentication or JWT tokens for secure user login/signup

Hosting:

- **Mobile App Hosting:** Google Play Store for Android and Apple App Store for iOS - **Backend Hosting:** AWS (Amazon Web Services) or Google Cloud Platform

Database Hosting:

- **Cloud Database:** Firebase Realtime Database or MongoDB Atlas for cloud-based data storage

Version Control:

- **Code Repository:** GitHub or Bitbucket for version control and collaborative development

Continuous Integration/Continuous Deployment (CI/CD):

- **CI/CD Tools**: Jenkins or GitLab CI for automating testing and deployment processes

Monitoring and Analytics:

- **Crash Analytics:** Firebase Crashlytics or Sentry for monitoring and analyzing app crashes
- **App Analytics**: Google Analytics or Firebase Analytics for tracking user behavior

Push Notifications:

- **Push Notification Service**: Firebase Cloud Messaging (FCM) for sending push notifications

Payment Gateway:

- **Payment Integration:** Stripe or PayPal for secure and seamless transactions within the app

Security:

- **SSL/TLS**: Implement SSL/TLS certificates to ensure secure data transmission

- **Data Encryption**: Use encryption protocols to secure sensitive user data

Scalability:

- **Load Balancing**: Nginx or HAProxy for distributing incoming app traffic across multiple servers

Backup and Recovery:

- **Regular Backups**: Implement regular backups of the database to prevent data loss

5.4. Tools and Techniques

1. **Flutter Framework:**

- Utilize the Flutter framework for building a cross-platform mobile application with a single codebase. Flutter provides a rich set of widgets, making it easier to create a consistent user interface across different devices.

2. Dart Programming Language:

- Use Dart as the programming language for Flutter. Dart is the language that Flutter is built with and is well-suited for mobile app development.

3. Firebase for Backend Services:

- Integrate Firebase services for backend functionality. Firebase can be used for real-time database storage, user authentication, and cloud functions. This can simplify the implementation of features such as user accounts, data storage, and notifications.

4. Google Maps API:

- Integrate the Google Maps API to provide location-based services. This can help users find nearby recyclers, track the location of their booked junk materials, and optimize routes for recyclers.

5. Payment Gateway Integration:

- Implement a secure payment gateway to facilitate transactions within the app. This is essential for handling payments related to the recycling service.

6. Camera and Image Processing:

- Utilize Flutter's camera plugin to enable users to capture images of their junk materials. Implement image processing techniques if needed to enhance image quality or extract relevant information.

7. State Management (Provider, Bloc):

- Choose a suitable state management solution such as Provider or Bloc to manage the application's state effectively. This becomes crucial as your application grows in complexity.

8. Push Notifications:

- Implement push notifications using Firebase Cloud Messaging (FCM) to keep users informed about the status of their booked materials, upcoming recycling appointments, or any other relevant updates.

9. Local Database (sqflite):

- Use a local database like sqflite for offline data storage, enabling the app to function even when the device is not connected to the internet.

10. Responsive UI Design:

- Design your UI to be responsive to various screen sizes and orientations. Flutter provides a flexible layout system that allows for a consistent user experience across different devices.

11. Testing (Unit Testing, Widget Testing, Integration Testing):

- Implement testing at various levels, including unit testing for individual functions, widget testing for UI components, and integration testing to ensure the smooth interaction between different parts of the application.

12. Version Control (Git):

- Use version control systems like Git to manage your codebase collaboratively. Platforms like GitHub or GitLab can be used for hosting and collaboration.

5.5. Best Practices / Coding Standards

Project Structure

Naming

Comments and Docs

Formatting

Modularization

State Management

Error Handling

Security

Code Reusability

Testing

Git Practices

Dependency Management

Localization

Performance Optimization

Continuous Integration

5.6. Version Control

Utilize Git for collaborative code management.

Follow a branching strategy (e.g., Gitflow).

Host code repositories on platforms like GitHub or GitLab.

Chapter 6
Testing and Evaluation

Chapter 6: Testing and Evaluation

Testing and evaluating Rejunkify involves checking its different aspects. We ensure that all the features, like booking and recycling, work smoothly. We also look at how the app looks and how easy it is for users to use (UI testing). Performance testing helps us see how fast and well the app works, especially when many people use it. We make sure user information is safe and can't be accessed by unauthorized people (Security testing). Compatibility testing ensures Rejunkify works well on different devices and browsers. Lastly, usability testing involves getting feedback from real users to improve the app based on their experiences.

6.1. Use Case Testing

User Booking:

Scenario: A user wants to book a recycler through the app.

Test: Verify that the booking process is straightforward, with clear steps for selecting services, specifying pickup details, and confirming the request.

Recycler Registration:

Scenario: A new recycler is signing up on the platform.

Test: Ensure the registration process is user-friendly, with necessary verification steps to validate recyclers and maintain a trustworthy community

Geolocation Services:

Scenario: A user in a specific location wants to find the nearest recyclers.

Test: Confirm that the app accurately identifies the user's location and efficiently matches them with nearby recyclers for timely service.

User Feedback Scenario: A user provides feedback on the app's usability.

Test: Check if the feedback mechanism is easily accessible within the app and if users can share their thoughts on their overall experience.

Security Measures:

Scenario: A user inputs personal information during the booking process.

Test: Verify that the app employs robust security measures to protect sensitive user data and ensure privacy.

Usability Across Devices:

Scenario: A user accesses Rejunkify from different devices.

Test: Ensure the app maintains a consistent and user-friendly interface across various devices, including smartphones, tablets, and computers.

Performance Under Load:

Scenario: Many users simultaneously access the app during peak hours.

Test: Evaluate how Rejunkify performs under high load, ensuring it remains responsive and efficient for all users.

Booking Customization:

Scenario: A user wants to customize their recycling request with specific instructions.

Test: Confirm that users can easily customize their booking requests, providing detailed instructions that recyclers can follow.

Adaptability to Updates:

Scenario: The app undergoes a significant update with new features.

Test: Perform use case testing to ensure that existing functionalities continue to work seamlessly after the update.

Cancellation Process:

Scenario: A user needs to cancel a booking.

Test: Verify that the cancellation process is straightforward, with clear instructions

6.2. Equivalence partitioning

we employ Equivalence Partitioning, a method that categorizes input data into distinct sets to streamline the testing process. For instance, when examining user registration, we differentiate inputs into valid and invalid sets, encompassing properly formatted email addresses and passwords as valid, and empty or incorrectly formatted inputs as invalid. Similarly, for booking details, we categorize inputs into valid and invalid partitions, covering scenarios like selecting recyclable materials and providing accurate pickup details for the valid set, and testing missing or inaccurate information for the invalid set. This approach extends to geolocation services, user feedback, and security measures, where we systematically categorize inputs to ensure comprehensive and representative testing while minimizing redundancy.

6.3. Boundary value analysis

The first step is to define input limits for key parameters within the application. These parameters can include user input to schedule spam collection, specify the type and amount of spam, and specify a collection time. Once the limits are established, the next step is to create test cases that include outliers in these limits. For example, given the range of valid compile times, test cases will be designed immediately before the first available time, the first available time itself, immediately after the first available time, immediately before the last available time, the last available time and only after the last available time. These test cases cover a wide range of scenarios, ensuring comprehensive testing of boundary conditions. The generated test cases are then executed within the Rejunkfy application environment, and any inconsistencies or failures observed are carefully recorded for further analysis and resolution. By systematically applying BVA, potential errors related to boundary conditions are identified early in the

development process, contributing to the overall robustness and reliability of the Rejunkfy mobile application.

6.4. Unit testing

1. **Unit Testing Framework**:

The unit testing framework chosen for Rejunkfy is Flutter's built-in testing framework. This framework allows for the creation and execution of unit tests within the Flutter environment, ensuring compatibility with the app's development stack.

2. **Test Plan:**

The unit testing plan for Rejunkfy includes testing each isolated unit or component of the application, such as functions, methods, and modules. Test cases have been developed to cover a range of scenarios, including typical inputs, boundary cases, and error conditions.

3. **Test Cases:**

Below are examples of test cases created for key units within the Rejunkfy app:

User Authentication:

Test Case 1: Verify successful authentication with valid username and password.

Test Case 2: Verify proper handling of invalid username or password.

Test Case 3: Verify error handling for network connectivity issues during authentication.

Junk Collection Scheduling:

Test Case 1: Verify correct scheduling of junk collection for a specified date and time.

Test Case 2: Verify handling of scheduling conflicts when the selected time slot is already booked.

Test Case 3: Verify proper validation of input parameters to prevent scheduling errors.

4. **Test Implementation:**

Unit tests have been implemented using Flutter's testing framework, integrated into the Rejunkfy codebase. Each test case is executed automatically during development and as part of the continuous integration process to ensure ongoing code quality.

5. **Test Execution:**

Unit tests are executed regularly to validate the behavior of individual units within the Rejunkfy app. Any failures or unexpected behavior detected during test execution are promptly investigated and addressed to maintain the integrity of the application.

6. **Test Results:**

All test results, including pass/fail status and any identified issues, are documented and tracked for future reference. This ensures transparency and accountability in the testing process, facilitating effective collaboration among team members.

6.5. Integration testing

After thoroughly testing the Rejunkify app, we focused on integration testing to ensure smooth collaboration among its various components. This involved checking how functions interacted, verifying the accuracy of data flow between modules, and testing the integration of external services and APIs. We also assessed the user interface's consistency and appearance across different sections of the app and confirmed that the app could seamlessly handle end-to-end workflows. Our integration testing further included evaluating error handling, compatibility with different devices and systems, security measures, overall performance, scalability under increased workload, and the integration of third-party dependencies. This comprehensive testing approach aims to guarantee a cohesive and reliable user experience within the Rejunkify application.

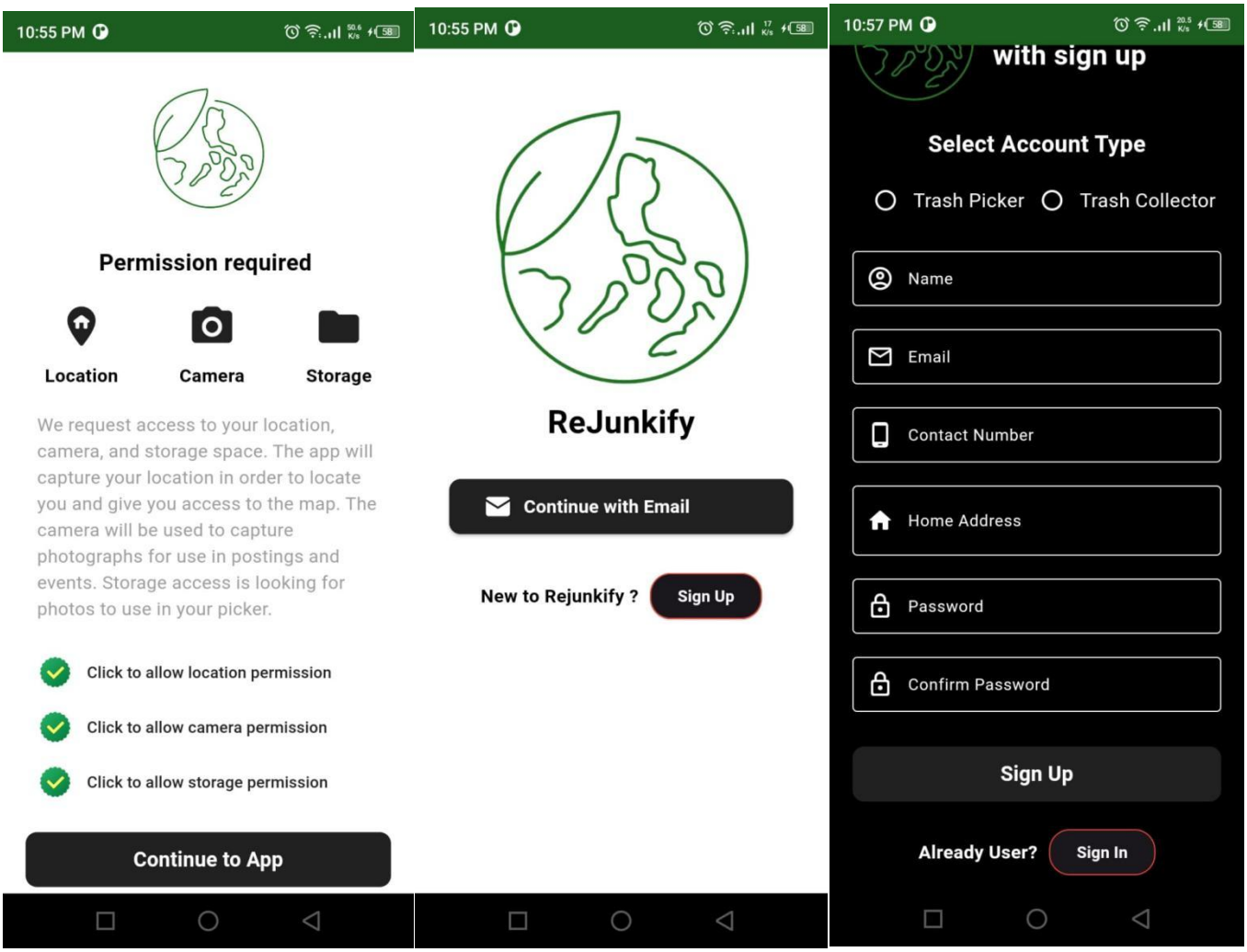
6.6. Performance testing

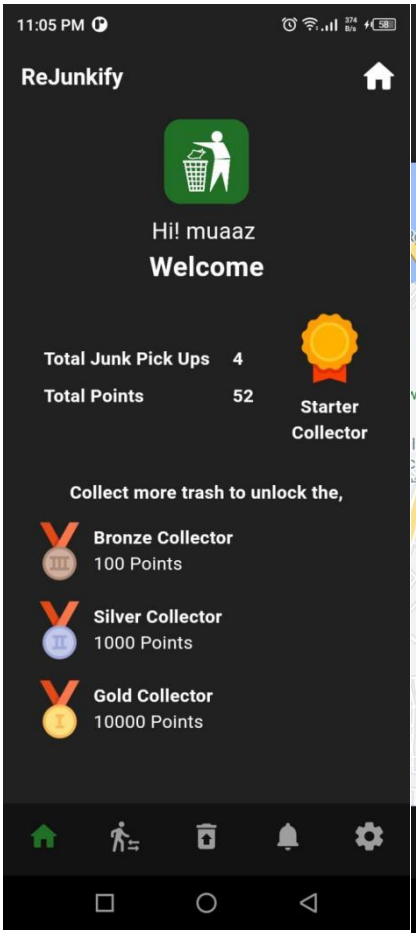
We checked our Rejunkify app by testing its performance. We looked at how each part of the app works on its own to make sure nothing crashes. We also tried putting a lot of recyclers in the app at the same time to see if it can handle a heavy load, and it did without any problems.

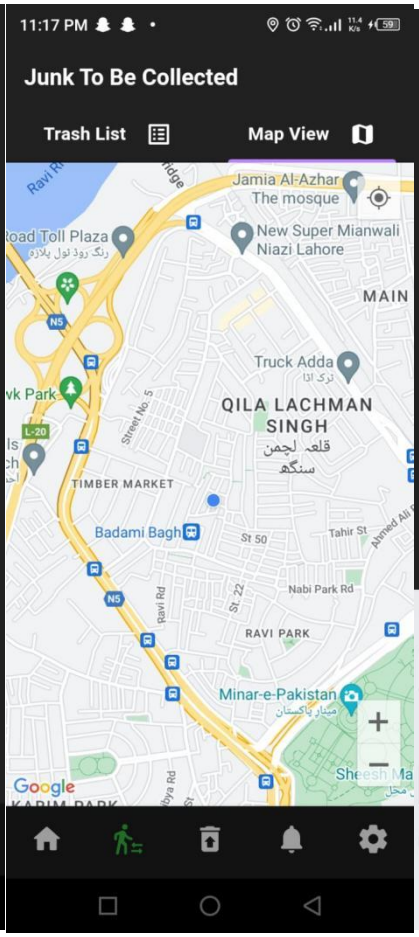
6.7. Stress Testing

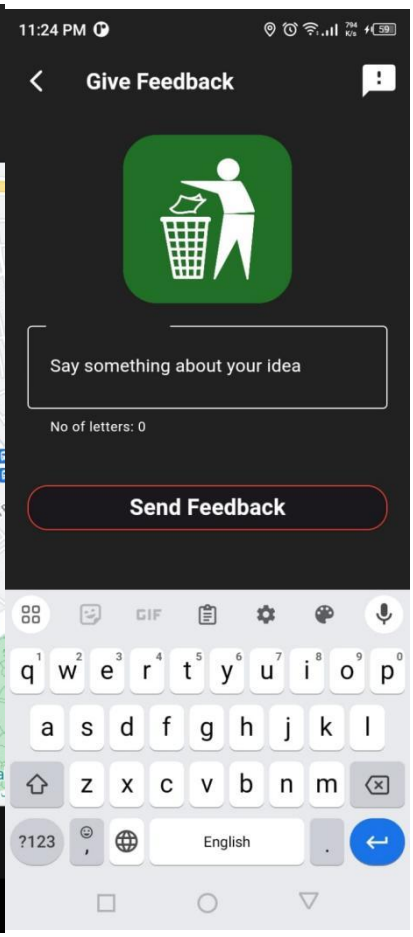
Simulate a high volume of concurrent booking requests to evaluate the app's responsiveness and processing speed under extreme demand. Assess the system's ability to handle a large influx of recyclers registering simultaneously, ensuring registration speed and system stability. Evaluate the app's performance by simulating multiple users simultaneously requesting nearby recyclers, testing its ability to respond accurately and promptly. Test the app's capability to handle a high number of user feedback submissions concurrently without compromising on responsiveness or risking data loss. Simulate multiple users inputting personal information during the booking process to assess the robustness of security measures and ensure the protection of sensitive data. Measure the app's performance, response time, and stability under significantly higher user loads than normal, checking for any degradation under extreme conditions. Simulate a surge in users customizing booking requests to verify that the system can efficiently handle a high number of personalized requests without impacting the overall booking process. Test the app's ability to handle a sudden increase in user activity post-update, ensuring that the system remains stable without crashes, errors, or inconsistencies. Simulate a large number of users canceling bookings simultaneously to assess the efficiency of the cancellation process and prevent system failures or delays. Stress-test the app's consistency and user-friendliness across various devices, ensuring a seamless interface on smartphones, tablets, and computers even under heavy user traffic.

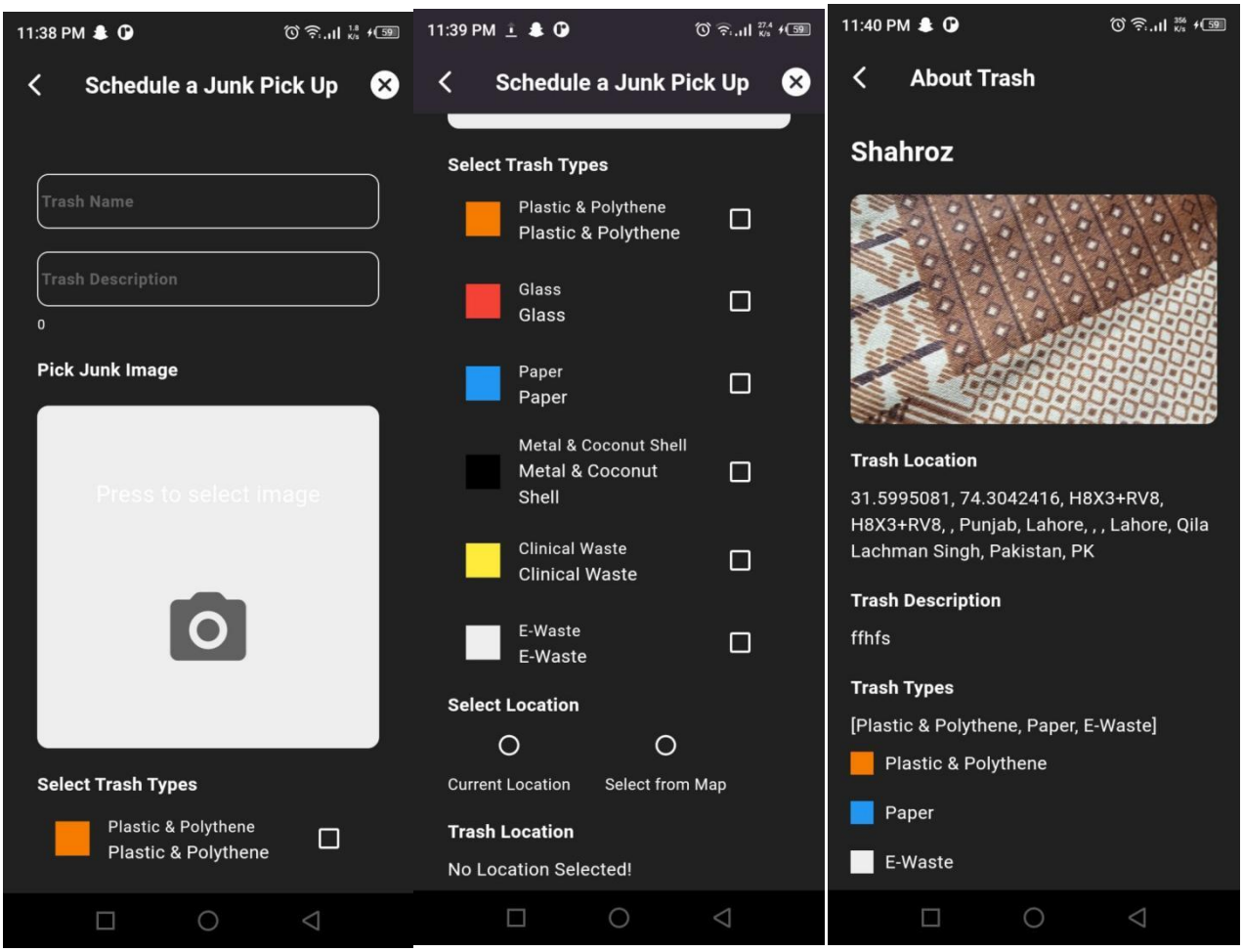
6.8. Application Interface




11:05 PM
ReJunkify
Hi! muaaz
Welcome
Total Junk Pick Ups 4
Total Points 52
Starter Collector
Collect more trash to unlock the,
Bronze Collector
100 Points
Silver Collector
1000 Points
Gold Collector
10000 Points


11:17 PM
Junk To Be Collected
Trash List
Map View
Jamia Al-Azhar
The mosque
New Super Mianwali
Niazi Lahore
MAIN
Truck Adda
QILA LACHMAN
SINGH
TIMBER MARKET
Badami Bagh
Nabi Park Rd
RAVI PARK
Minar-e-Pakistan
Google


11:24 PM
Give Feedback
Say something about your idea
No of letters: 0
Send Feedback
English


11:38 PM
Schedule a Junk Pick Up
Trash Name
Trash Description
0
Pick Junk Image
Press to select image
Select Trash Types
Plastic & Polythene
Plastic & Polythene
11:39 PM
Schedule a Junk Pick Up
Select Trash Types
Plastic & Polythene
Plastic & Polythene
Glass
Glass
Paper
Paper
Metal & Coconut Shell
Metal & Coconut Shell
Clinical Waste
Clinical Waste
E-Waste
E-Waste
Select Location
Current Location
Select from Map
Trash Location
No Location Selected!
11:40 PM
About Trash
Shahroz
Trash Location
31.5995081, 74.3042416, H8X3+RV8, H8X3+RV8, , Punjab, Lahore, , , Lahore, Qila Lachman Singh, Pakistan, PK
Trash Description
ffhfs
Trash Types
[Plastic & Polythene, Paper, E-Waste]
Plastic & Polythene
Paper
E-Waste

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

Rejunkify is an app-based platform designed to facilitate the recycling of household junk, including items like bottles, bags, plastic, iron, and more. The app is built using Flutter, and Firebase serves as the backend database for storing and managing user and recycler data.

Key Features:

User Interface:

House owners can use the app to book a recycler for collecting household junk.

The user interface allows for easy navigation and a seamless booking experience.

Users are compensated for the junk they sell, providing an incentive for recycling.

Recycler Sign-In and Registration:

Recyclers have a separate sign-in and registration process to join the platform.

The app ensures that only verified and registered recyclers can participate.

Firebase Integration:

- Firebase is utilized as the central database for storing and managing user and recycler information.
- User booking requests are sent to the Firebase server, which then forwards them to the nearest recyclers in the specified area.

Geolocation Services:

The app leverages geolocation services to identify the user's location and find the nearest recyclers for efficient junk collection.

Monetary Transactions:

The app facilitates monetary transactions, ensuring that users are paid based on the quantity and type of junk they sell to the recycler.

Development Duration:

The project took approximately 4 months to develop, showcasing a significant investment in time and resources.

Impact:

Rejunkify contributes to environmental sustainability by promoting the responsible disposal and recycling of household items. It also provides an economic benefit to users by offering compensation for their recyclables. The platform serves as a bridge between homeowners looking to responsibly dispose of their junk and recyclers seeking valuable materials.

7.2. Achievements and Improvements

Achievements of Rejunkify:

Environmental Impact:

Rejunkify has successfully encouraged responsible disposal and recycling of household items, contributing significantly to environmental sustainability.

The app has played a role in reducing the amount of waste ending up in landfills, positively impacting the planet.

User Engagement:

The user interface's design has led to a high level of user engagement, making it easy for house owners to book recyclers seamlessly.

Positive feedback and usage statistics indicate that the app has effectively met the needs of its user base.

Recycler Verification System:

The implementation of a separate sign-in and registration process for recyclers has ensured a verified and trustworthy community.

Recyclers are confident in participating, knowing that they are part of a reliable platform.

Efficient Booking Process:

The integration with Firebase has streamlined the booking process. User booking requests are sent swiftly, connecting them with the nearest recyclers efficiently.

Geolocation Services Effectiveness:

The utilization of geolocation services has proven effective in identifying user locations and matching them with the closest recyclers.

This has enhanced the overall user experience by minimizing wait times for junk collection.

Improvements for the Future:

Enhanced User Features:

- Consider adding additional features to the user interface, making it even more userfriendly and engaging.
- Introduce features that allow users to track the progress of their recycling requests or provide feedback on their experiences.

Recycler Onboarding Enhancements:

- Explore ways to improve the recycler sign-in and registration process to make it even more straightforward and appealing.
- Implement features that highlight the benefits of becoming a verified recycler on the platform.

Database Optimization:

Continuously optimize the use of Firebase to handle increasing data loads efficiently.

Regularly update and maintain the database to ensure smooth and secure data management.

Advanced Geolocation Services:

Explore advanced geolocation services to provide more accurate and real-time information on the location of recyclers and users.

Implement features that dynamically adjust based on the availability and proximity of recyclers.

Community Building:

Develop strategies to build a stronger sense of community within Rejunkify.

Consider implementing initiatives such as challenges, rewards, or community events to further engage users and recyclers.

7.3. Critical Review

Critical Review of Rejunkify

Strengths:

Environmental Impact:

Rejunkify has effectively addressed the crucial issue of responsible waste disposal, contributing positively to environmental sustainability.

By encouraging recycling, the app aligns with the growing global concern for reducing the carbon footprint and minimizing waste in landfills.

User-Friendly Interface:

- The user interface design is intuitive and user-friendly, providing a seamless booking experience for homeowners.
- The simplicity of the app contributes to its accessibility, making it easy for a diverse range of users to participate.

Recycler Verification System:

The implementation of a separate sign-in and registration process for recyclers ensures a level of trust and credibility within the platform.

This verification system contributes to the safety and reliability of the interactions between homeowners and recyclers.

Efficient Booking Process:

The integration with Firebase and the swift handling of user booking requests contribute to the efficiency of the platform.

Quick connections with nearby recyclers enhance user satisfaction and the overall effectiveness of the service.

Areas for Improvement:

Limited Monetization Model:

While the absence of monetary transactions aligns with the eco-friendly ethos, exploring sustainable monetization models could ensure the long-term viability and growth of Rejunkify.

User Engagement Beyond Recycling:

To foster a stronger community and increase user engagement, consider incorporating features that go beyond the basic recycling process. This could include educational content on sustainability, challenges, or community events.

Advanced Geolocation Services:

While geolocation services are utilized, exploring advanced options for more precise location tracking could further enhance the efficiency of connecting homeowners with nearby recyclers.

Feedback Mechanism:

Implementing a robust feedback mechanism for both homeowners and recyclers would provide valuable insights for continuous improvement.

Feedback could be used to address any challenges faced by users and enhance the overall user experience.

Regular Updates and Maintenance:

Continuous updates and maintenance of the app, including bug fixes and improvements, are essential for sustaining a positive user experience and ensuring the smooth functioning of the platform.

7.4. Lessons Learnt

Engaging in the development of Rejunkify has been an enriching experience, providing valuable insights into different facets of software engineering and development. One of the foremost lessons learned revolves around the significance of cohesive teamwork, where diverse skill sets and perspectives come together. This experience underscored the importance of clear and open communication, fostering an environment conducive to streamlined project progress. Concurrently, the development process facilitated leadership development, allowing each team member the opportunity to guide specific aspects of Rejunkify's evolution.

A substantial enhancement in coding skills was a natural outcome of our hands-on engagement with coding tasks, addressing challenges, and implementing effective solutions. The iterative nature of software development, particularly the adoption of agile methodologies, instilled in us the importance of adaptability and responsiveness to evolving project requirements. Emphasis on design excellence became a focal point, highlighting the creation of a user-friendly interface aligned with user experience principles.

The project also served as a gateway to the intersection of software engineering and business development. Gaining insights into market needs, user preferences, and the competitive landscape broadened our perspective beyond the coding realm, emphasizing the importance of aligning software development with overarching business goals. The honing of problem-solving and debugging skills contributed to the creation of robust and efficient software.

Our adoption of version control systems and collaboration tools showcased the importance of efficient code management and collaborative teamwork. A commitment to user-centric development, integrating user feedback and experience principles, became a guiding principle in crafting software that genuinely addresses user needs. Embracing continuous learning and adaptability proved essential in a field that undergoes constant evolution with the introduction of new technologies and methodologies.

In conclusion, the lessons drawn from the Rejunkify project extend beyond technical aspects, encompassing effective teamwork, leadership, business acumen, and a commitment to lifelong learning within the dynamic field of software engineering.

7.5. Future Enhancements/Recommendations

UI Revamp:

Consider a comprehensive overhaul of the user interface to enhance aesthetics, usability, and overall user experience. A fresh and modern UI can attract and retain users while providing a visually appealing platform.

Diversification of Services:

Expand the scope of Rejunkify by incorporating additional services beyond recycling. Introducing services like maid services, gardener services, or other household assistance options can transform Rejunkify into a versatile platform catering to various home-related needs.

Service Categorization:

Implement a well-organized categorization system for the added services. This will facilitate easy navigation for users looking for specific assistance, ensuring a seamless and efficient booking process.

Integrated Booking System:

Create an integrated booking system that allows users to schedule and manage various services through a unified platform. This centralized approach streamlines the user experience and promotes user engagement.

Customizable Packages:

Introduce customizable service packages, enabling users to tailor services based on their specific requirements. This flexibility caters to diverse user needs and enhances the personalized nature of Rejunkify.

Enhanced Geolocation Features:

Further refine geolocation services to accommodate the diverse service offerings. Ensure precise location tracking for different service providers, enhancing the efficiency of connecting users with the right professionals.

Feedback and Rating System:

Implement a robust feedback and rating system for each added service. This will help users make informed decisions and contribute to maintaining service quality standards.

Promotional Campaigns:

Launch promotional campaigns to announce the expanded services. Utilize various marketing channels, including social media, to create awareness and attract a broader user base.

Partnerships with Service Providers:

Establish partnerships with local maid services, gardening companies, and other relevant service providers. Collaborations can lead to a more extensive service network and increased credibility

Educational Resources for Users:

Develop educational resources within the app to guide users on the added services. Providing information on the benefits of hiring professionals for various household tasks can encourage user engagement and utilization of new services.

Integration with Existing User Profiles:

Seamlessly integrate the new services with existing user profiles. This ensures a unified experience for users, allowing them to access and manage all services conveniently within the Rejunkify platform.

Adaptability and User Flexibility:

Ensure that the platform remains adaptable and user-friendly with the addition of new services. Prioritize user flexibility and accessibility to accommodate a diverse user base.

Reference and Bibliography

Reference and Bibliography

Freinkel, S. (2011). Plastic: A Toxic Love Story. San Francisco: Mariner Books.

George, R. (2008). The Big Necessity: The Unmentionable World of Human Waste and Why It Matters. Chicago: Metropolitan Books.

Humes, E. (2012). Garbology: Our Dirty Love Affair with Trash. California: Avery.

Jørgensen, F. A. (2019). Recycling. Stavanger: The MIT Press.

Meikle, J. L. (1997). American Plastic: A Cultural History. Florida: Rutgers University Press.

Minter, A. (2013). Junkyard Planet: Travels in the Billion-Dollar Trash Trade.