Smart Recycle

A Group Project

By

Nikhil Mishra (2141123)

Rishi Vagadia (2141132)

Roshan Kataruka (2141133)

Under the Guidance of

Dr. Smitha Vinod

And

Dr. Kirubanand

Department of Computer Science

Christ (Deemed to be University)

Bengaluru India

Introduction

Recycle Smart is a mobile app that will let users provide their waste for recycling. The app is designed to make recycling easy and convenient. Users who have recyclable waste in their homes can book an appointment for collecting the waste, which will collected at the user's convenient time by different pickup teams after which it will be recycled. The app also provides users with information about recycling, such as what items can be recycled and what's the benefit of recycling. Recycle Smart is a great way to reduce waste and help the environment. By rewarding users through appreciation certificates for recycling, the app encourages people to recycle more often. This can help to reduce the amount of waste that ends up in landfills, and it can also help to conserve natural resources.

Problem Definition:

The problem that the "Recycle Smart" mobile app aims to address is the lack of a convenient and user-friendly platform for recycling waste. Many individuals may have recyclable waste in their homes but find it challenging to dispose of it properly. Additionally, the lack of awareness about recycling practices and their environmental benefits can contribute to a significant amount of waste ending up in landfills.

The "Recycle Smart" app seeks to tackle these issues by providing a solution that offers the following features:

Easy Waste Collection: The app enables users to book appointments for waste collection, making the process of recycling more convenient. Users can schedule pickups at their preferred time and location.

Information on Recycling: The app serves as an educational tool by providing users with valuable information about recycling. This includes details about which items can be recycled and the environmental benefits of recycling.

Environmental Impact: By encouraging and rewarding users for recycling through appreciation certificates, the app aims to increase the frequency of recycling. This, in turn, helps to reduce the amount of waste sent to landfills, contributing to a cleaner environment.

Conservation of Resources: The app aligns with the goal of conserving natural resources by promoting recycling practices. Recycling materials reduces the need for raw materials, leading to more sustainable resource usage.

The "Recycle Smart" app seeks to bridge the gap between individuals who want to recycle but lack convenient means to do so and the importance of recycling for environmental preservation. By providing a user-friendly platform and promoting awareness, the app aims to foster a recycling culture and contribute to a positive impact on the environment.

Requirement Specification:

Software Requirement Specification (SRS) for "Recycle Smart" Mobile App

1. Introduction

1.1 Purpose

- The purpose of the "Recycle Smart" app is to provide users with a convenient and user-friendly platform for recycling waste materials.
- The app aims to increase recycling rates by offering appointment-based waste collection and educational resources about recycling.
- By rewarding users with appreciation certificates for recycling, the app encourages regular recycling practices.

1.2 Scope

- The "Recycle Smart" app will be available for mobile devices running on both Android and iOS platforms.
- It will serve users within a specific geographical region initially, with the potential for future expansion.

2. Overall Description

2.1 Product Perspective

- The "Recycle Smart" app will function as a standalone application, independent of any other systems.

- It will interface with the user's mobile device's camera, location services, and push notification capabilities.

2.2 User Classes and Characteristics

- The app will cater to two primary user classes:
- Regular Users: Individuals who want to recycle their waste and access recycling-related information.
- Pickup Teams: Teams responsible for collecting waste from users' locations.

2.3 Operating Environment

- The app will operate on Android and iOS devices with a minimum supported version specified.

2.4 Design and Implementation Constraints

- The app's design should prioritize simplicity, ease of use, and an intuitive user interface.

2.5 User Documentation

- The app will include in-app help and instructions to guide users through its features.
- An online user guide and FAQs will be made available on the app's website.

Functional Requirements

3.1 User Registration and Authentication

- Users should be able to register and create accounts using their email or social media accounts.
- Account authentication should be implemented to ensure data security and user privacy.

3.2 Waste Collection Booking

- Users should be able to schedule waste collection appointments using a calendar-based interface.
- They must provide location details and preferred time slots for pickups.

3.3 Pickup Team Assignment

- Pickup teams should receive notifications and details about scheduled appointments.
- Teams should be able to view their assigned pickups and navigate efficiently to users' locations.

3.4 Recycling Information

- The app should provide information about recyclable materials and their proper disposal methods.
- Users can access articles or FAQs related to recycling.

3.5 Reward System

- The app should track user recycling activity and reward them with appreciation certificates based on their recycling frequency.

Non-Functional Requirements

4.1 Performance

- The app should load quickly and respond to user actions within acceptable time frames.
- It should be able to handle concurrent user requests without significant delays.

4.2 Security

- User data should be encrypted and stored securely to prevent unauthorized access.
- The app must implement proper authentication mechanisms to protect user accounts.

4.3 Reliability

- The app should be robust and reliable, minimizing downtime and system failures.

4.4 Usability

- The user interface should be intuitive and user-friendly for users of all ages and technological backgrounds.

4.5 Compatibility

- The app should be compatible with a wide range of mobile devices and screen sizes.

System Requirements

5.1. Development Environment Requirements:

- Operating System compatibility for Android Studio and Flutter.

- Sufficient resources for smooth development.

5.2. Target Mobile Application Requirements:

- Compatibility with Android and/or iOS platforms.

- Responsive design for various screen sizes and orientations.

5.3. User Interface (UI) and User Experience (UX) Requirements:

- Intuitive and user-friendly UI design.

- Consistent branding and visual elements.

5.4. Functionality Requirements:

- Implementation of required features and functionalities.

- Proper error handling and validation.

5.5. Performance Requirements:

- Fast loading times and response rates.

- Minimal battery consumption and resource usage.

5.6. Testing Requirements:

- Thorough testing on multiple devices.

- Bug fixing and continuous improvement.

5.7. Documentation Requirements:

- Comprehensive codebase and architecture documentation.

- User guides or manuals, if applicable.

5.8. Compliance Requirements:

- Adherence to relevant app store guidelines.

- Compliance with legal and privacy requirements.

5.9. Maintenance and Support:

- Regular updates and improvements.

- Support for bug fixes and inquiries.

5.10. Scalability:

- Consideration for future expansion.

- Scalability of features and services.

Conceptual Model

Conceptual Model for the "Recycle Smart" Mobile App:

6.1. Entities:

- User: Represents individuals who want to recycle waste and access recycling-related information.

- Pickup Team: Represents teams responsible for collecting waste from users' locations.

- Recyclable Material: Represents the different types of waste that can be recycled.

6.2. Attributes:

- User:

  - Name: Stores the user's name.

  - Email: Stores the user's email address used for registration.

  - Password: Stores the encrypted password for account authentication.

- Recycling Frequency: Keeps track of the user's recycling activity.

- Pickup Team:

  - Team Name: Stores the name of the pickup team.

  - Contact: Stores contact information for the pickup team.

- Recyclable Material:

  - Material Name: Stores the name of the recyclable material (e.g., plastic, paper, glass).

  - Disposal Method: Provides information about the proper disposal method for each recyclable material.

6.3. Relationships:

- User to Recyclable Material: Users can access information about recyclable materials and their disposal methods.

- User to Pickup Team: Users schedule waste collection appointments with pickup teams.

- Pickup Team to Recyclable Material: Pickup teams are responsible for collecting specific recyclable materials from users.

6.4. Actions/Operations:

- User Registration and Authentication:

  - User can register and create an account using their email or social media accounts.

  - The app authenticates users' accounts to ensure data security.

- Waste Collection Booking:

  - Users can schedule waste collection appointments with pickup teams.

  - Users provide location details and preferred time slots for pickups.

- Recycling Information:

  - Users can access information about different recyclable materials and their proper disposal methods.

- The app provides articles and FAQs related to recycling.

- Reward System:

- The app tracks users' recycling frequency and rewards them with appreciation certificates.

6.5

- User Registration and Authentication:

- User provides registration details (name, email, password) -> Account created and authenticated.

- Waste Collection Booking:

- User schedules a waste collection appointment with preferred details -> Pickup team receives notification and appointment details.

- Recycling Information:

- User accesses recyclable material information and disposal methods.

- Reward System:

- User recycling activity is tracked -> Users receive appreciation certificates based on recycling frequency.

6.6. Use Cases:

- User Registration: Users can register and create an account on the app.

- Waste Collection Booking: Users can schedule waste collection appointments.

- Access Recycling Information: Users can access information about recycling and recyclable materials.

- Reward for Recycling: Users are rewarded with appreciation certificates based on their recycling frequency.

6.7. Diagrams:

- Entity-Relationship Diagram (ERD):

- Illustrating the relationships between User, Pickup Team, and Recyclable Material entities.

6.8. Constraints:

- User Interface Constraints:

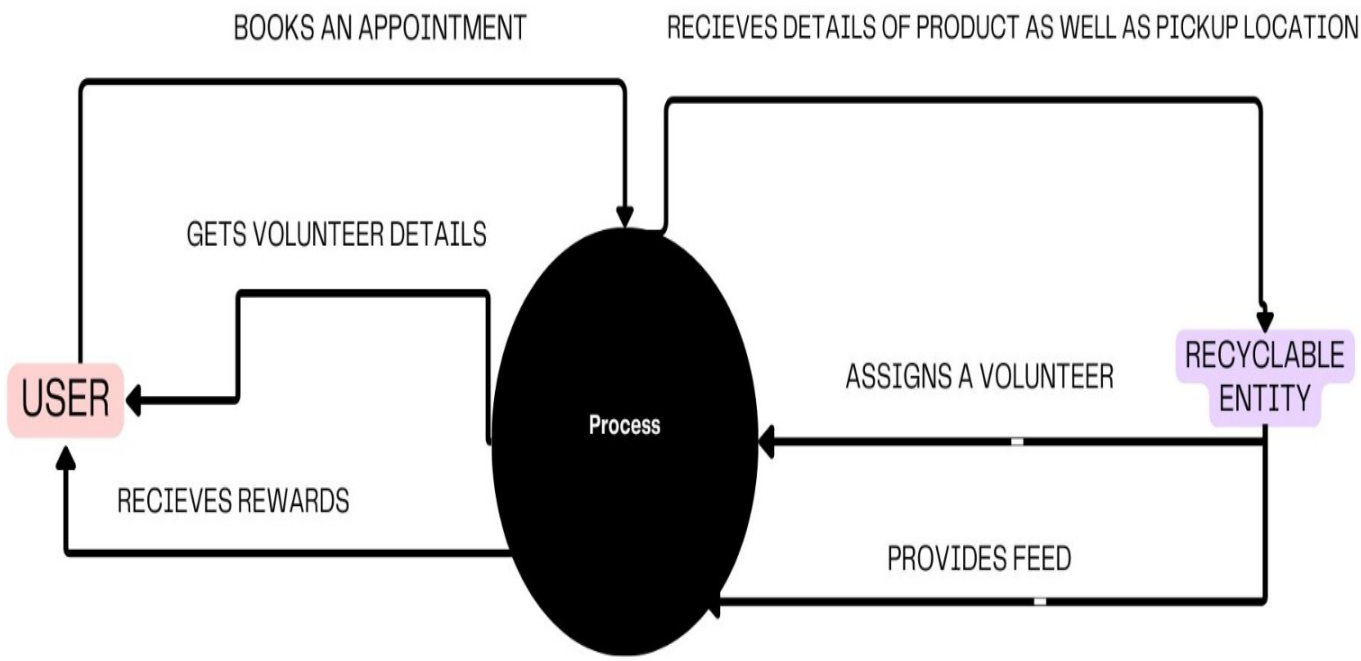
- The app's user interface should prioritize simplicity and ease of use to cater to users of all ages and technological backgrounds.

- Data Security Constraints:

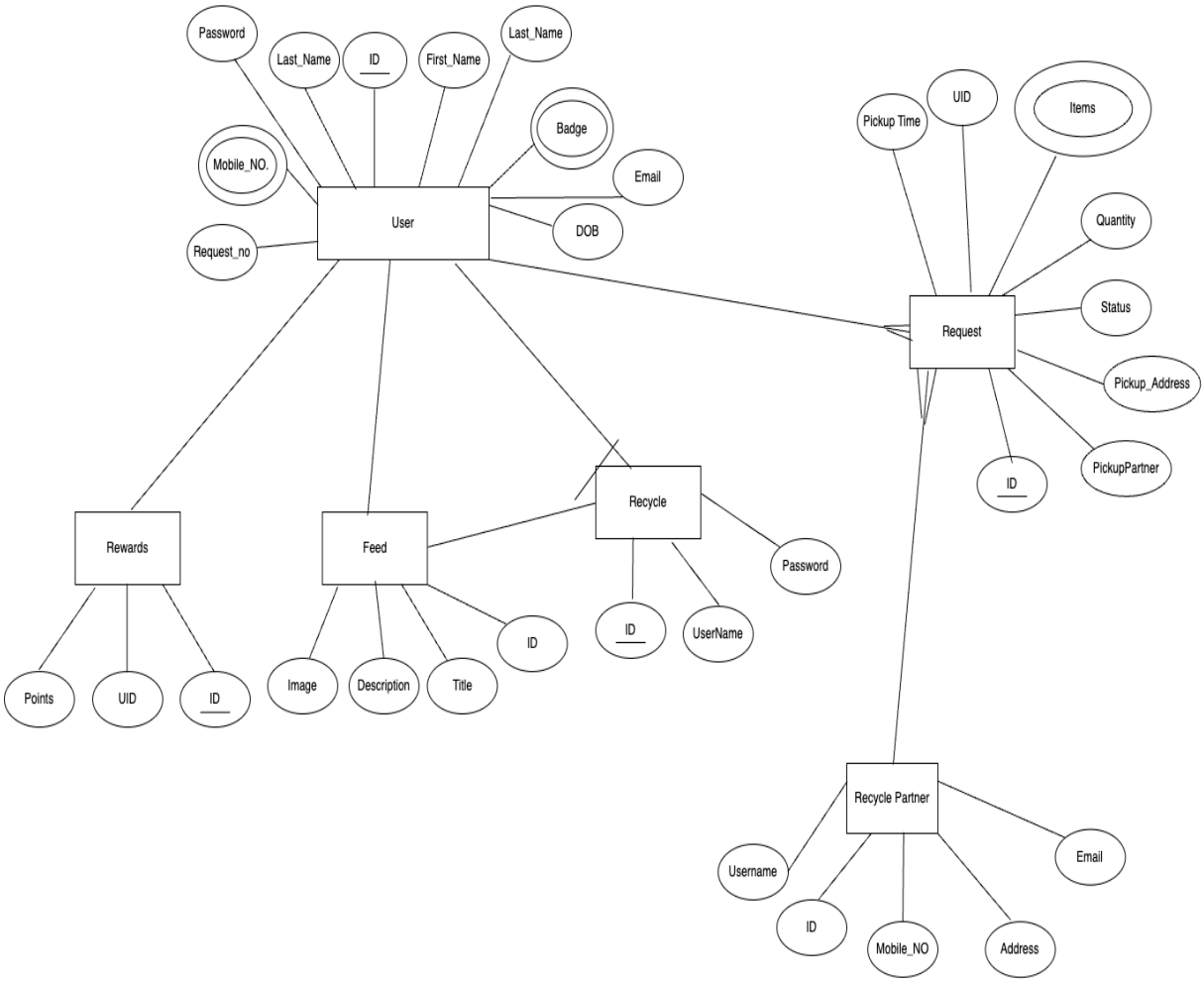
- User data should be encrypted and stored securely to prevent unauthorized access.

The Conceptual Model provides an abstract representation of the "Recycle Smart" mobile app, focusing on its core entities, their attributes, relationships, actions, and interactions. This high-level view aids in understanding the app's fundamental structure and behaviour, serving as a foundation for further development and design efforts.

Data Flow Diagram



ER Diagram



Planning And Scheduling

Planning and Scheduling for the "Recycle Smart" Mobile App:

7.1. Project Scope and Objectives:

- Define the scope of the project, including the functionalities and features to be developed.

- Set clear objectives, such as increasing recycling rates and providing educational resources.

7.2. Project Timeline:

- Create a detailed project timeline with milestones for each development phase.

- Allocate sufficient time for design, development, testing, and deployment.

7.3. Resource Allocation:

- Identify and allocate resources required for the development, including developers, designers, testers, and project managers.

- Ensure access to the necessary hardware and software tools for development.

7.4. User Documentation and Help:

- Plan for in-app help and instructions to guide users through the app's features.

- Develop an online user guide and FAQs on the app's website for additional support.

7.5. Risk Management:

- Identify potential risks and their impact on the project.

- Develop mitigation plans and contingency strategies to minimize risks.

7.6. Communication and Reporting:

- Establish clear communication channels among team members, stakeholders, and clients.

- Regularly provide progress reports and updates on the project's status.

By planning and scheduling the development process effectively, the "Recycle Smart" app project can be efficiently executed, meeting its objectives of providing a user-friendly platform for recycling and promoting environmental awareness. Regular monitoring and adjustments to the plan will help ensure that the project progresses smoothly and delivers a successful mobile application.

Tools Proposed

Flutter is an open-source mobile UI framework created by Google. It is used to develop native mobile apps for Android and iOS from a single codebase. Flutter is known for its high performance, beautiful visuals, and ease of use.

Frontend

Flutter's frontend framework is based on the Dart programming language. Dart is a compiled language that is fast, expressive, and type-safe. Flutter's frontend framework provides a rich set of widgets that can be used to create beautiful and interactive user interfaces.

Backend

Flutter can be used with a variety of backend technologies. Some popular options include:

Firebase is a BaaS (backend as a service) platform that provides a number of features for Flutter apps, including user authentication, database storage, and push notifications.

Node.js is a popular JavaScript runtime environment that can be used to create web servers and APIs.

Django is a Python web framework that can be used to create web servers and APIs.

Database

Flutter can be used with a variety of databases. Some popular options include:

Firebase Realtime Database is a NoSQL database that is well-suited for real-time applications.

MongoDB is a NoSQL database that is flexible and scalable.

MySQL is a relational database that is widely used and supported.

THANK YOU!