
Software Requirements Specification

for
BinBuddy

Version 1.0 approved

Prepared by:

HTOO MYAT NOE (U2422977K)
ARMAN KHAN (U2421760A)
CHUA YUE JUN (U2423015D)
NIKHIL MADETI (U2421243D)
SOH CEK CONG (U2423500C)
YOONG HONG JUN, NICHOLAS (U2321582L)

Group 3, SCSI, Nanyang Technological University

5 November 2025

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References	3
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	4
2.3 User Classes and Characteristics	5
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
3. External Interface Requirements	7
3.1 User Interfaces	7
3.2 Hardware Interfaces	22
3.3 Software Interfaces	22
3.4 Communications Interfaces	22
4. System Features	23
4.1 User Features	23
4.2 General Features	27
4.3 Admin Features	30
5. Other Nonfunctional Requirements	36
5.1 Performance Requirements	36
5.2 Safety Requirements	36
5.3 Security Requirements	36
5.4 Software Quality Attributes	37
5.5 Business Rules	37
6. Other Requirements	37

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This purpose of this Software Requirements Specification (SRS) is to outline the following:

- Scope and objectives of the BinBuddy software.
- Functional and non-functional requirements for our BinBuddy web application.
- User roles and interactions (users and administrators).
- External interface requirements.

It serves as a common reference for developers, testers, stakeholders and product managers for development, testing and maintenance of the platform. The scope of this SRS is limited to BinBuddy's software components, which includes the frontend web interface, backend server logic and database layer.

1.2 Document Conventions

The SRS follows the following documentation conventions:

Font: Arial
Headings: Size 14, Bold
Content: Size 12
Spacing: Single Spacing

All requirements have equal priority unless stated.

1.3 Intended Audience and Reading Suggestions

It is advisable for all stakeholders to go through the SRS in detail. The document is separated by sections and a suggested reading order is as follows:

1. Section 1 - Overview and purpose of BinBuddy web application
2. Section 2 - System design and external interfaces
3. Section 3 - Functional requirements
4. Section 4 - Non-functional requirements

Developers	Developers require deep understanding of the functional and non-functional requirements in order to implement the system components.
Project Managers	Project managers need the SRS to track project progress and ensure the product meets the client's expectations.

Graphic Designers	Graphic designers can design the user interface based on the requirements stated by the client.
Users	Users can use the SRS to validate whether the requirements stated meet their expectations.
Testers	Testers can create test cases based on the requirements.
Documentation writers	Documentation writers can use this SRS as a reference to produce user guides and update the SRS when needed.

1.4 Product Scope

BinBuddy is a web application designed to assist users in locating nearby e-waste disposal bins and submitting queries related to e-waste recycling. The system targets electronic device users and e-waste disposers who are seeking a convenient and environmentally responsible way to discard electronic waste.

The primary objective of BinBuddy is to promote recycling by improving accessibility and awareness. The application allows users to:

- Locate nearby e-waste disposal bins quickly based on their location.
- Access educational information and disposal guidelines.
- Submit queries or feedback related to e-waste bins.

By making recycling easier and more user-friendly, BinBuddy supports national sustainability initiatives and contributes to reducing environmental impact.

1.5 References

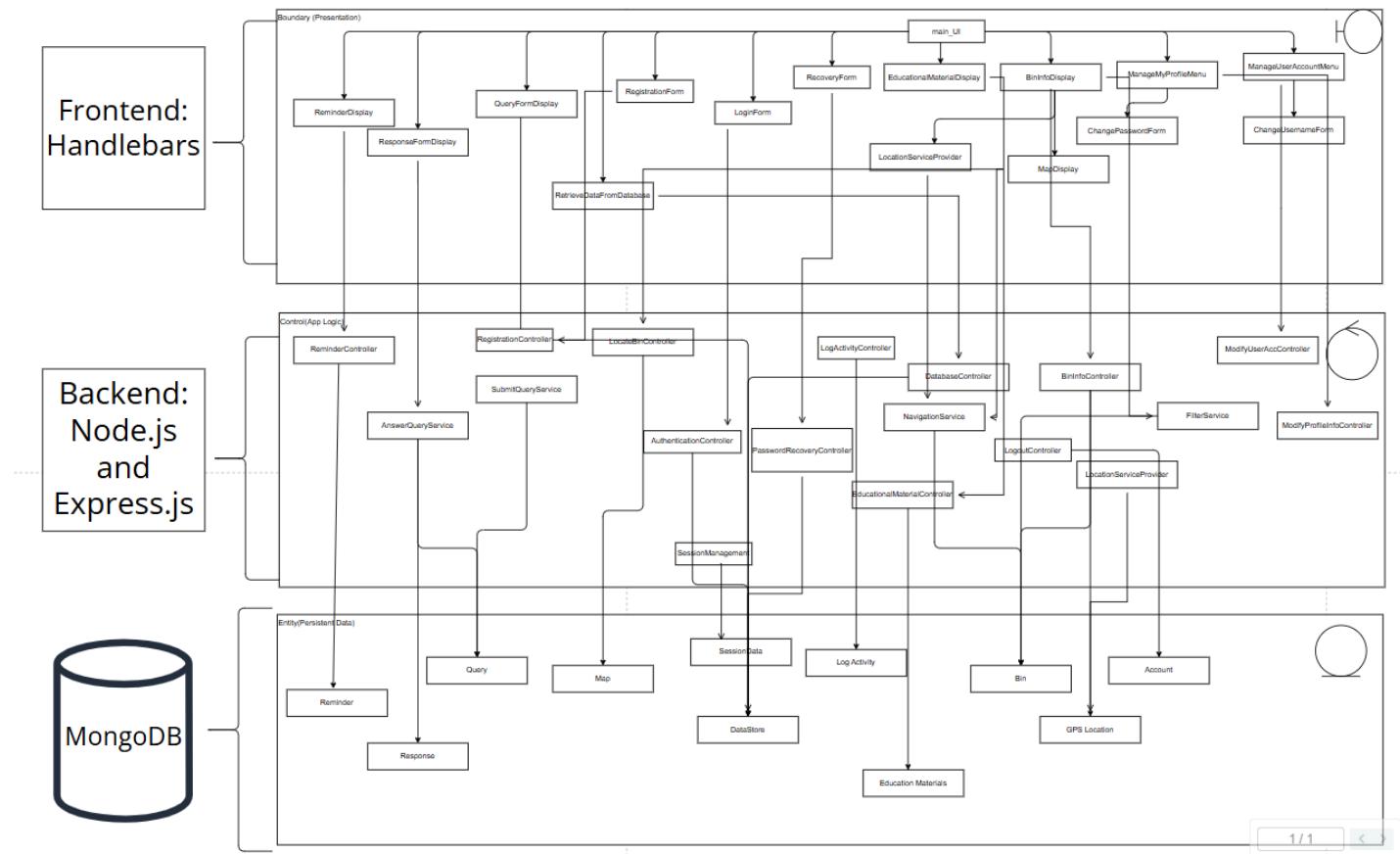
- I. MongoDB documentation: <https://www.mongodb.com/docs/>
- II. Express.js reference: <https://expressjs.com/>
- III. handlebars.js reference: <https://handlebarsjs.com/>
- IV. Nodemailer Documentation: <https://nodemailer.com/>
- V. Google Maps API reference: <https://developers.google.com/maps/documentation/javascript>

2. Overall Description

2.1 Product Perspective

BinBuddy is a new standalone web-based application developed to help users in locating e-waste bins nearby in an effort to increase Singapore's e-waste recycling rate. The system retrieves data related to e-waste bins and presents them visually on a map using Google Maps API. Unlike other commercial or government systems, BinBuddy does not receive real-time bin capacity, as such data is not available.

BinBuddy depends on several external services to work. This includes Node.js , Express.js, Google Maps API, SMTP, Bcrypt and MongoDB database.



2.2 Product Functions

The following use case diagram summarises the core functionalities that BinBuddy includes. There are 2 main main actors that are involved - User and Admin.

User

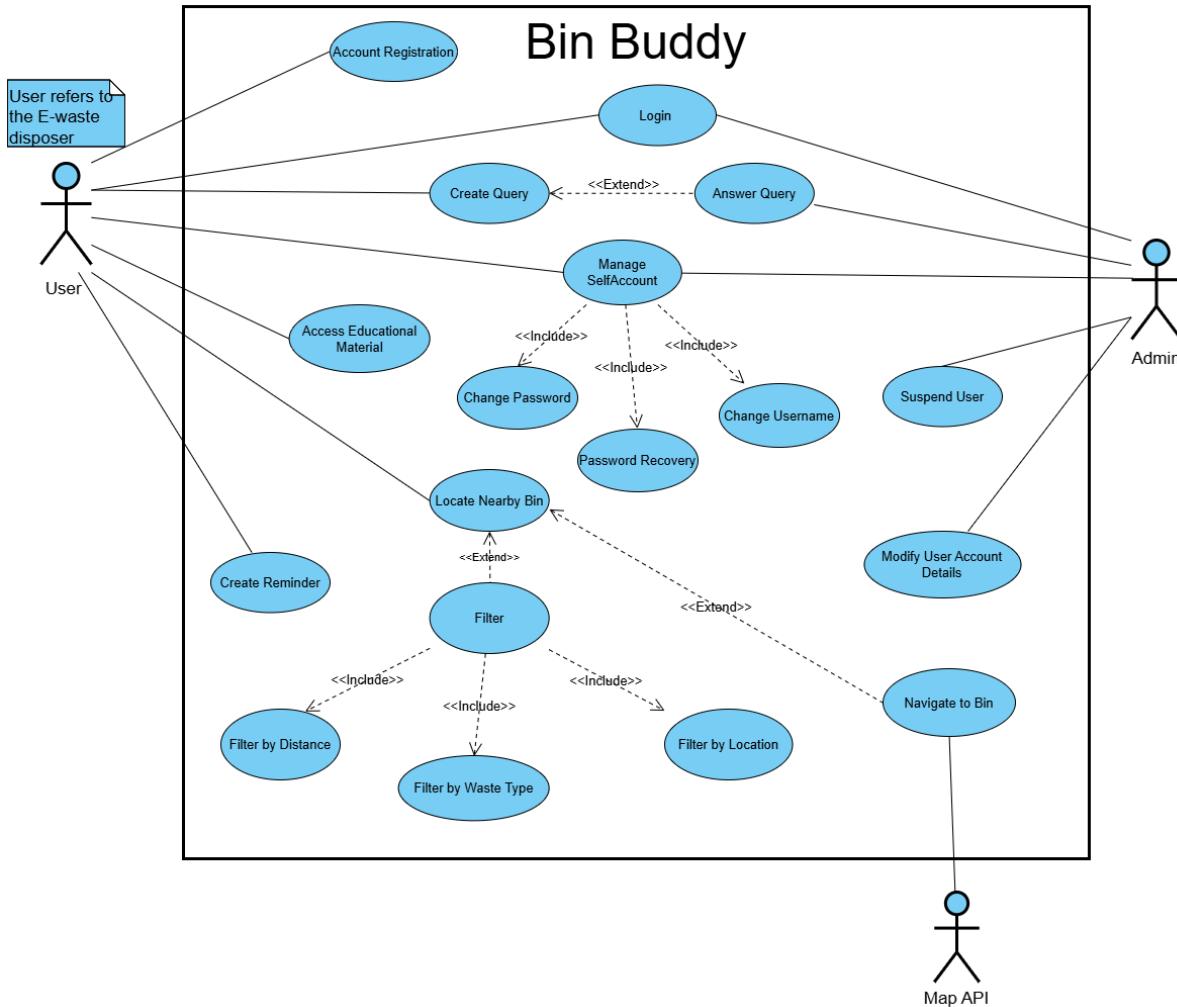
- Registration, Login and Password recovery
- Editing account profile
- Viewing e-waste related educational materials
- Creating reminders and queries
- Locating nearby bins as well as filtering bins by distance,waste type, and name of location

Admin

- Viewing and answering queries
- Changing user account (including recovering accounts and suspending accounts)

Others

- Auto flagging of user account



2.3 User Classes and Characteristics

The anticipated user classes include:

Consumers of electronic devices

Aspect	Description
Frequency of Use	Occasionally
Subset of product functions used	All user features
Technical expertise	Individuals should have a working device that can access the users' location.
Characteristics	Individuals who often use electronic products and hence may want to engage in more environmentally

	friendly practices by disposing their e-waste responsibly at designated e-waste bins.
--	---

Company technicians

Aspect	Description
Frequency of Use	Frequently
Subset of product functions used	All user features
Technical expertise	Individuals should have a working device that can access the users' location.
Characteristics	Technicians are responsible for equipment lifecycle management and are trained to dispose of faulty equipment in an environmentally-friendly manner.

2.4 Operating Environment

BinBuddy is a web-based application that operates on browsers such as Chrome, Firefox, Safari and Microsoft Edge. For it to operate all functionalities, the following components must be available:

Internet Connectivity (Wi-Fi / Mobile Data Connection)

- Internet Connectivity is required to access the web application, retrieve bin data and communicate between the cloud database and backend server

GPS Location Service

- Location Service must be available on the user's device to support the navigation features. (**Node.js and Express**)
- Our system logic runs on the backend service, which handles user requests and communicates with external API.

MongoDB Database

- Used to store data including user account information.

SMTP Email Service

- SMTP email service is required for sending automated password recovery email to the user.

Google Maps API

- Google Maps API is used to display the map interface which shows the location of bins as well as the directions.

2.5 Design and Implementation Constraints

For this project, we have to use data that is publicly and readily available on the government website. So in our case, the National Environment Agency (NEA) publishes the e-waste bin locations as well as the type of e-waste accepted at each location. However, real-time bin occupancy or capacity status is not available in the government dataset.

This means that BinBuddy cannot show whether a specific bin is near its capacity or is currently unavailable. If possible, we would like to partner with NEA to integrate sensors that monitor the bin current occupancy so that users can make smarter choices when choosing e-waste bins. With that information, NEA can also monitor the bins and deploy people to clear the bins when it's full.

2.6 User Documentation

A demonstration video will be made available on our Github repository to showcase the core functionalities of BinBuddy. The README file will also provide an overview view of how to set up the application.

2.7 Assumptions and Dependencies

The following assumption and dependencies apply to the operations of our BinBuddy system:

NEA dataset
Backend services: Node.js, Express.js, multer, Flash Messengers, Cookie Parser, Bcrypt
Third-part APIs: Google Maps API, SMTP.
Database cloud services: MongoDB Atlas
Internet / GPS availability
User device and Browser compatibility

3. External Interface Requirements

3.1 User Interfaces

3.1.1 UI Samples

3.1.1.1 Home

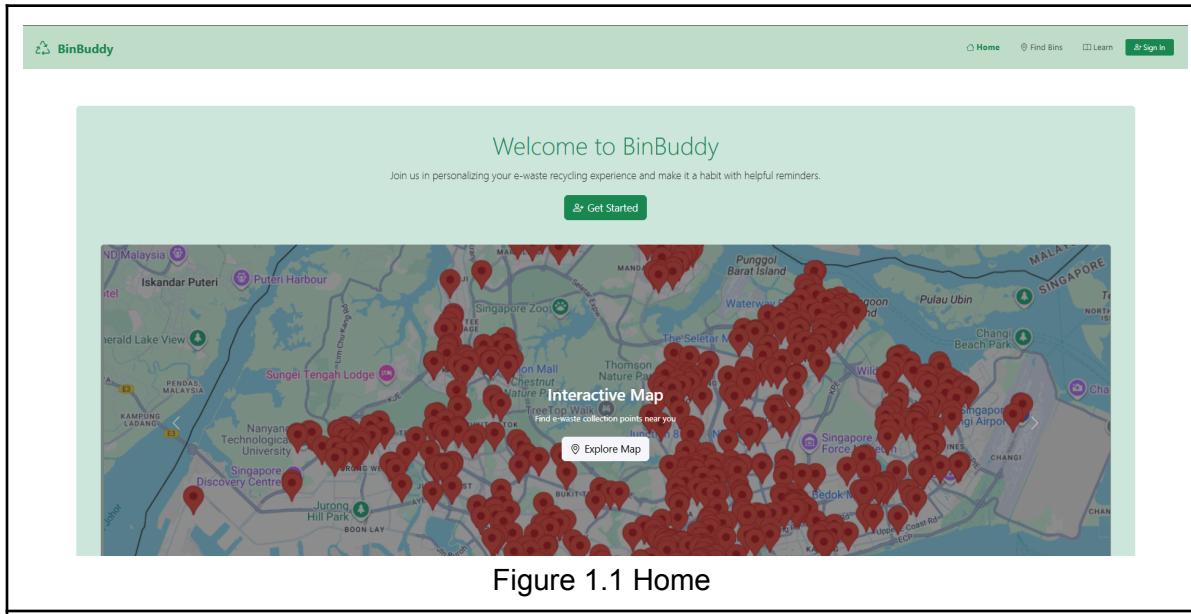


Figure 1.1 Home

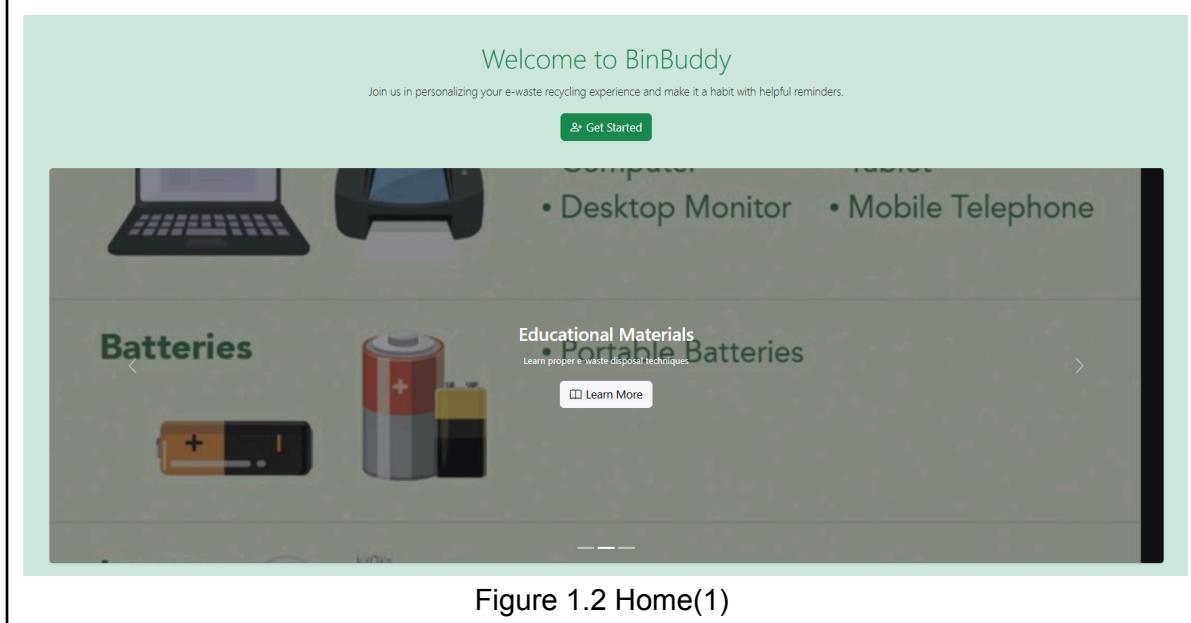


Figure 1.2 Home(1)

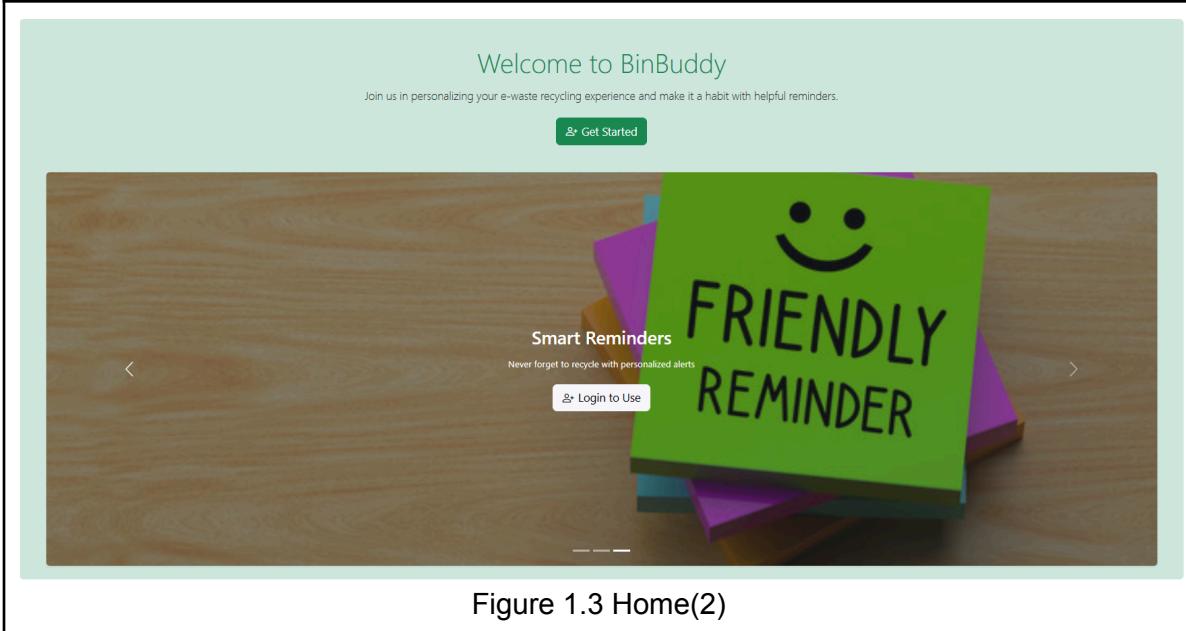


Figure 1.3 Home(2)

3.1.1.2 Find Bin

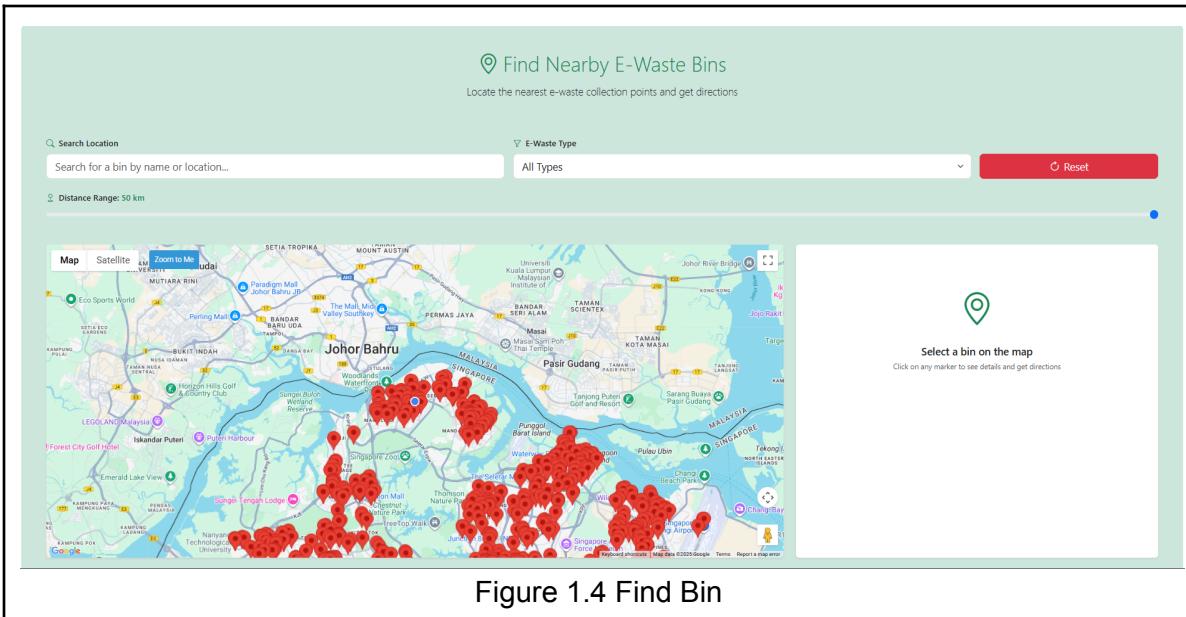


Figure 1.4 Find Bin

3.1.1.3 Educational Materials

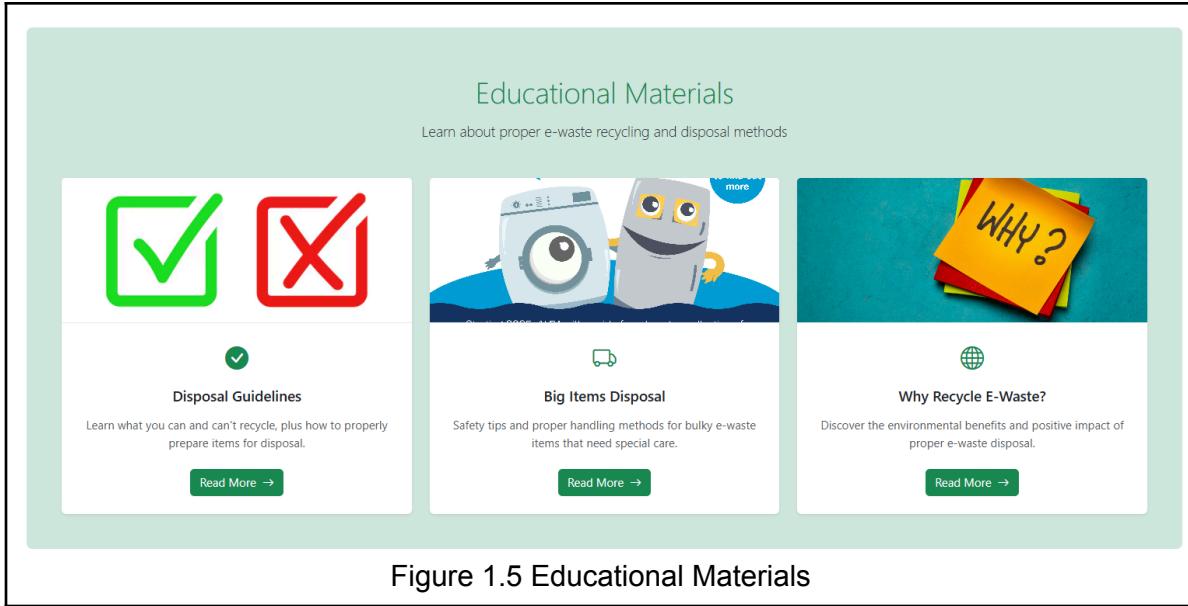


Figure 1.5 Educational Materials

3.1.1.4 Account Creation

The screenshot shows the "Join BinBuddy" account creation form. The title is "Join BinBuddy" with the subtitle "Create your account and start your recycling journey". The form fields are:

- Full Name**: "Enter your full name"
- Username**: "Choose a username" with a note "At least 3 characters long"
- Email Address**: "Enter your email address"
- Password**: "Create a strong password" with a note "At least 6 characters with 1 number and 1 special character"
- Confirm Password**: "Confirm your password"
- Admin Code (optional)**: "Enter admin code if you have one" with a note "Only provide this if you were given a valid admin code by the system administrator."

A large green button at the bottom left says ">Create Account". At the bottom right, it says "Already have an account? [Sign in here](#)".

Figure 1.6 Account Creation

3.1.1.5 Login

Welcome Back

Sign in to your BinBuddy account

Email Address
Enter your email

Password
Enter your password

Forgot your password?

Sign In

Admin Sign In

Don't have an account? [Create one here](#)

Admin Sign In

Sign in with an admin account

Email Address
Enter admin email

Password
Enter your password

Sign In as Admin

Figure 1.7 User Login and Admin Login

3.1.1.6 Reset Password

Reset Password

Enter your email address and we'll send you a recovery link

Email Address
Enter your registered email

We'll send password reset instructions to this email

Send Recovery Email

Remember your password? [Sign in here](#)

Figure 1.8 Reset Password

3.1.1.7 User Profile

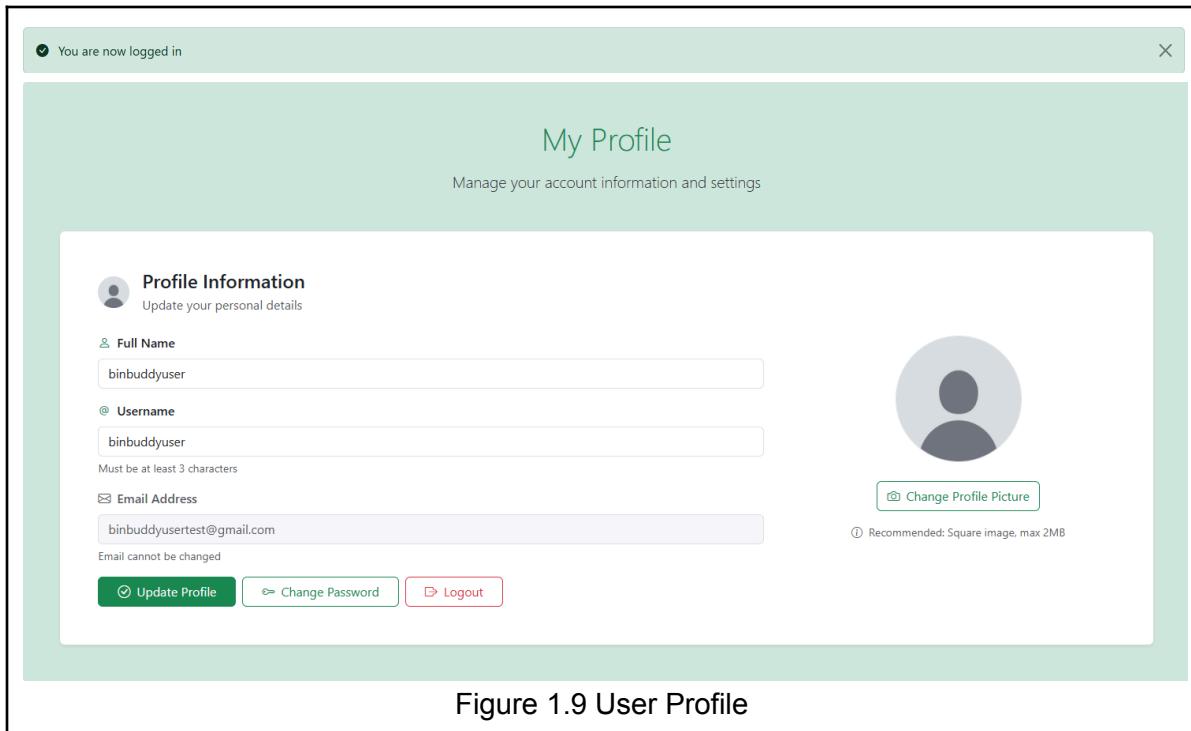


Figure 1.9 User Profile

3.1.1.8 Change Password

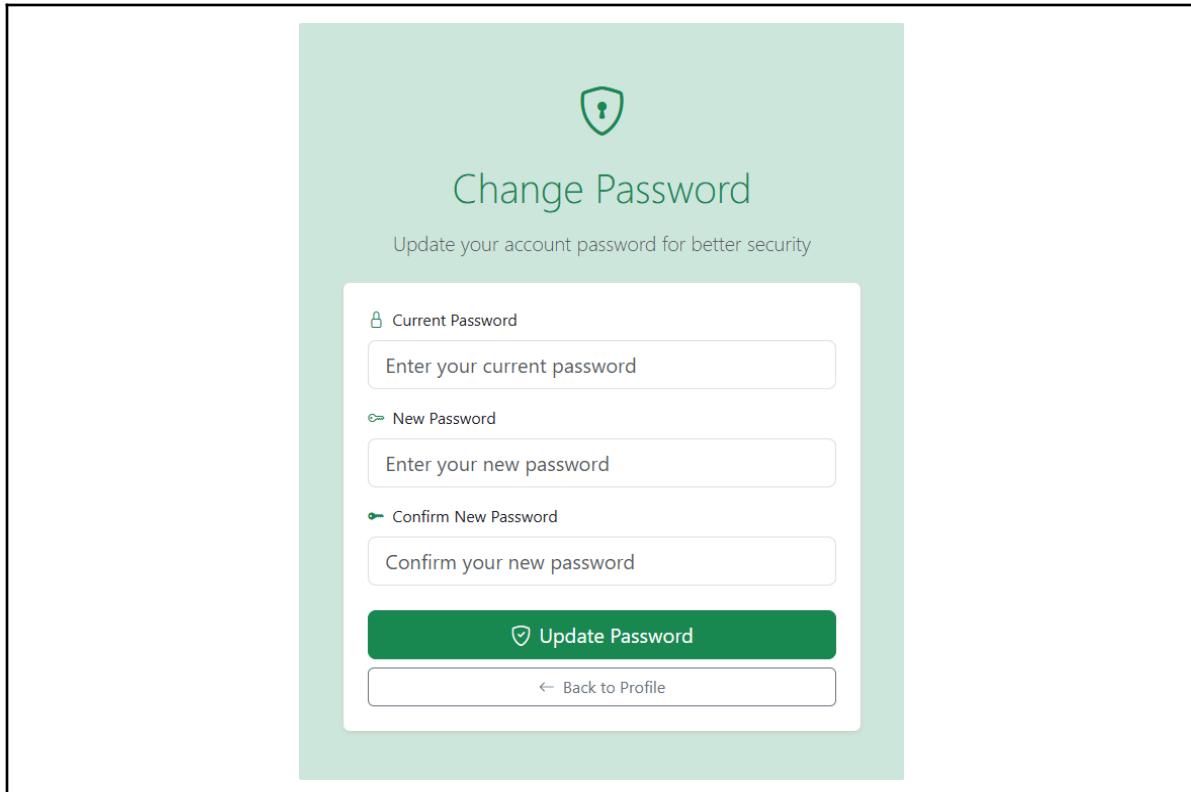


Figure 1.10 Change Password

3.1.1.9 Personalised Home

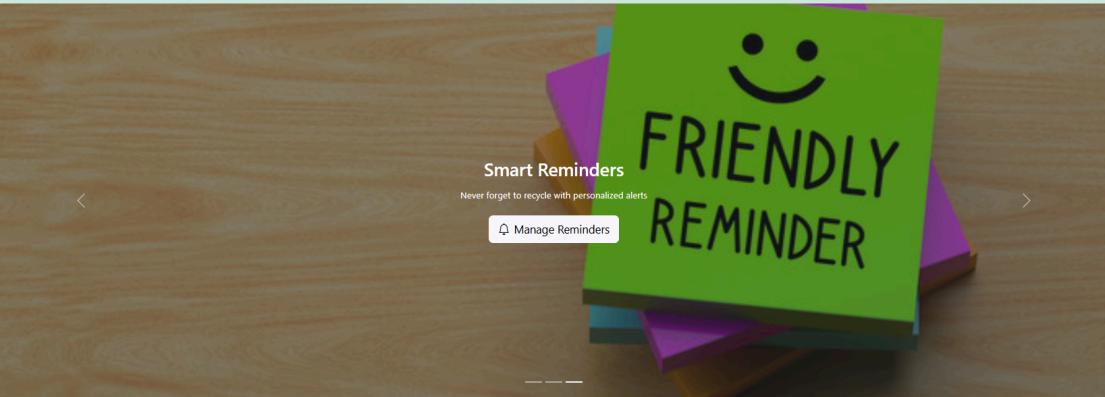


Figure 1.11 Personalised Home

3.1.1.10 Reminder

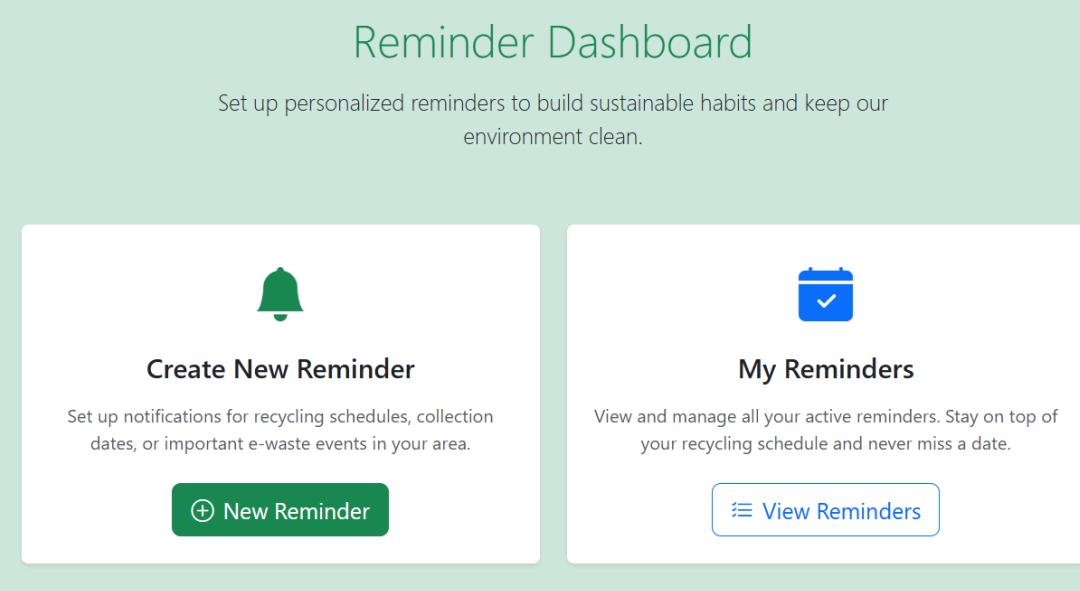


Figure 1.12 Reminder

Create New Reminder

Set up a personalized reminder to help you stay on track with your recycling goals

Reminder Message

Enter your reminder message...

What would you like to be reminded about?

Date & Time

dd/mm/yyyy --:-- --

When would you like to receive this reminder?

Save Reminder

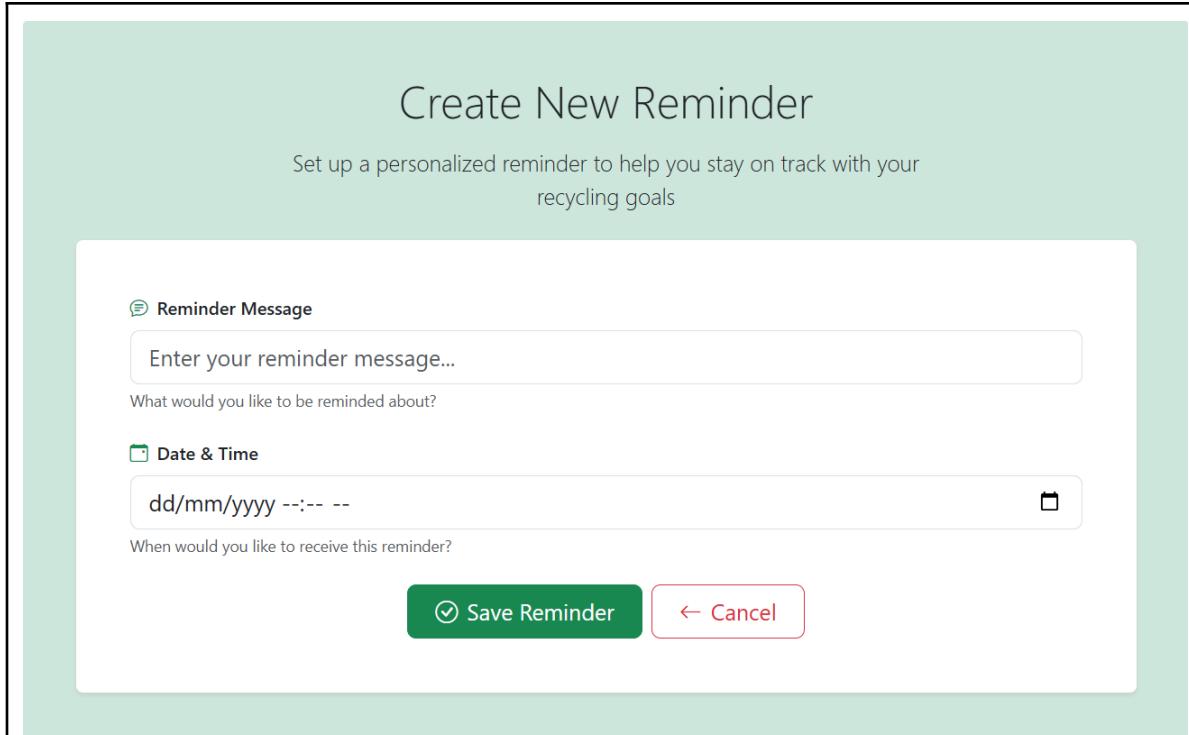


Figure 1.13 New Reminder

My Reminders

Keep track of all your recycling reminders and stay committed to a sustainable lifestyle.

No Reminders Yet

You haven't set up any reminders. Start building sustainable habits today!

Create Your First Reminder

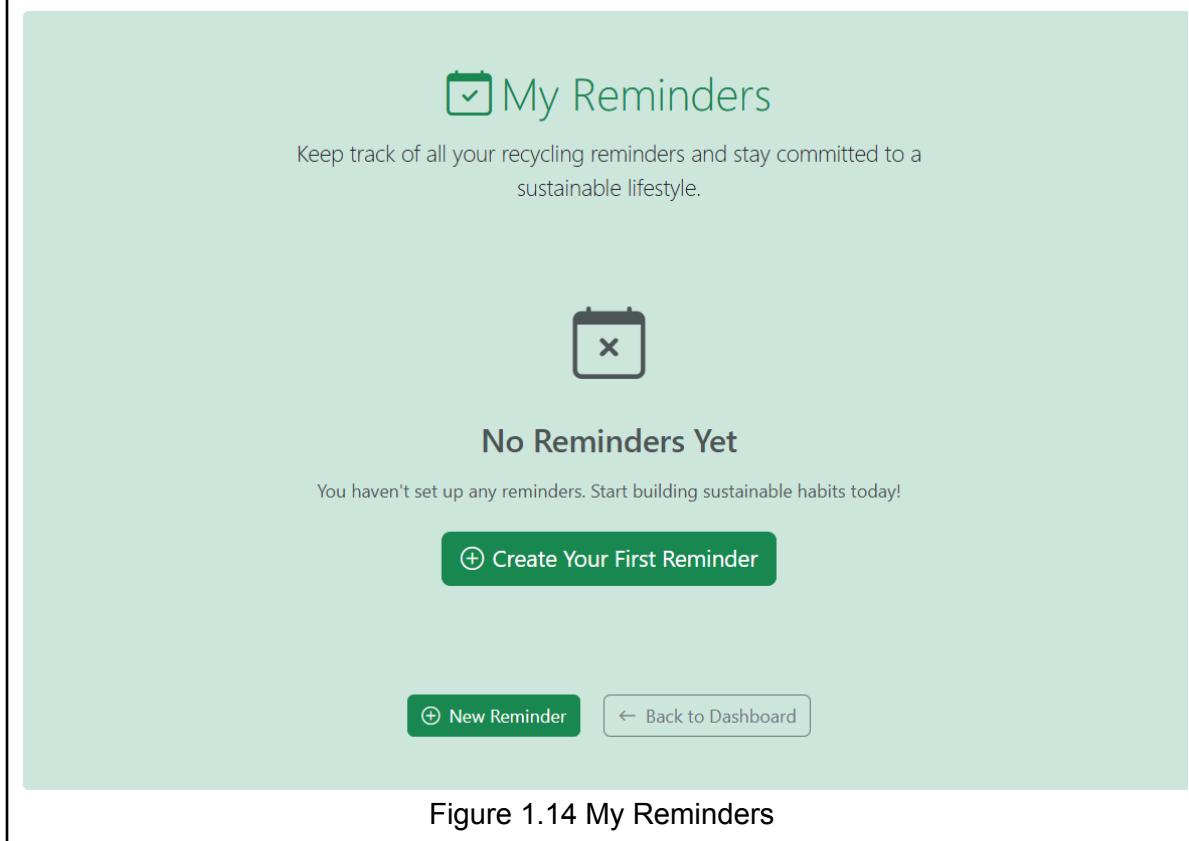


Figure 1.14 My Reminders

3.1.1.11 Query

The screenshot shows the 'Query Dashboard' interface. At the top center is the title 'Query Dashboard'. Below it is a sub-instruction: 'Have questions about e-waste recycling? Submit your queries and get guidance to make recycling easier and more effective.' The interface is divided into two main sections. The left section, titled 'Submit New Query', features a large question mark icon and a green button labeled '+ Ask a Question'. A descriptive text below the button reads: 'Ask questions about recycling processes, disposal methods, or anything related to e-waste management.' The right section, titled 'My Queries', features a speech bubble icon and a blue button labeled ': View My Queries'. A descriptive text below the button reads: 'View all your submitted questions and track their status. Get answers from our team.'

Figure 1.15 Query

The screenshot shows the 'Ask a Question' interface. At the top center is the title 'Ask a Question'. Below it is a sub-instruction: 'Get expert guidance on e-waste recycling, disposal methods, and environmental practices'. The main area contains a text input field with a placeholder: 'Ask anything about e-waste recycling, disposal guidelines, bin locations, or environmental practices...'. Below the input field is a note: 'Be as specific as possible to get the most helpful answer'. At the bottom are two buttons: a green 'Submit Question' button with a paper airplane icon and a red 'Cancel' button with a left arrow icon. At the very bottom, there is a link: 'Browse our educational guides for quick answers' and a blue 'My Questions' button.

Figure 1.16 Ask a Question

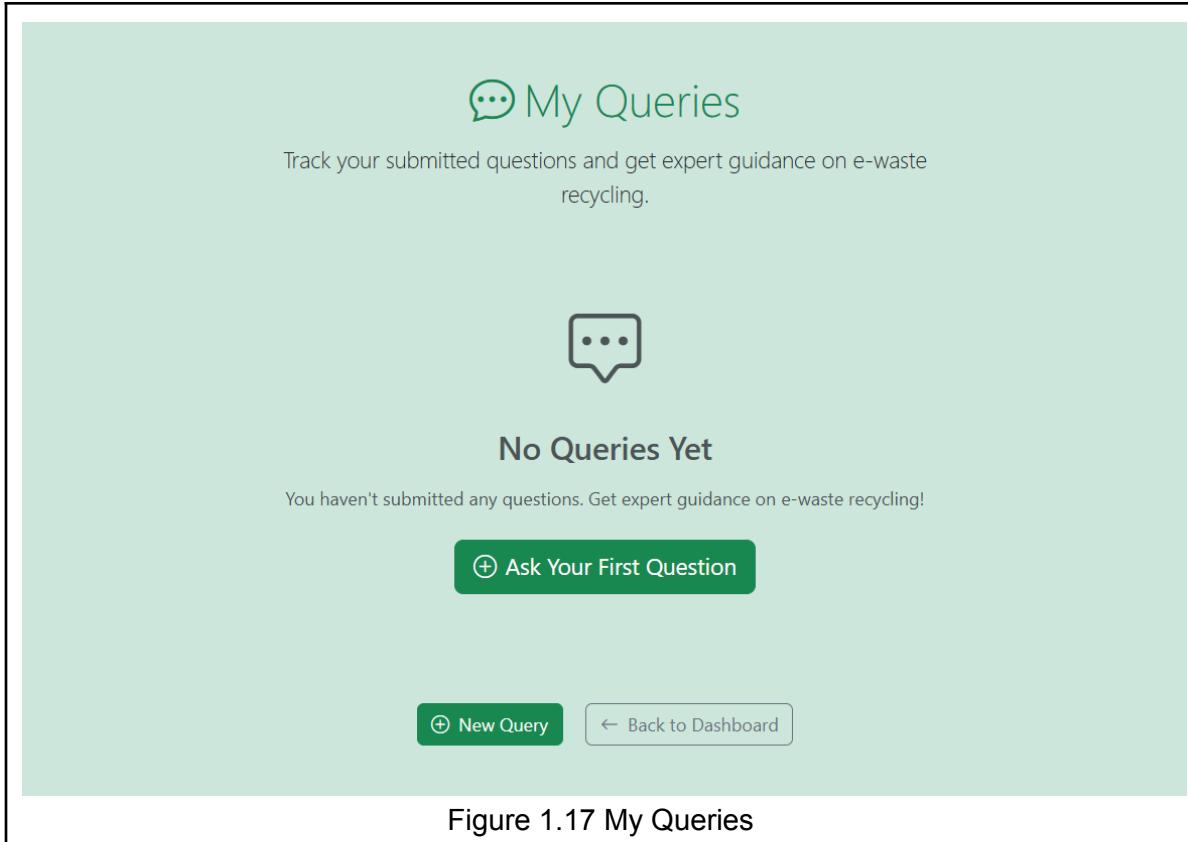


Figure 1.17 My Queries

3.1.1.12 Admin Dashboard

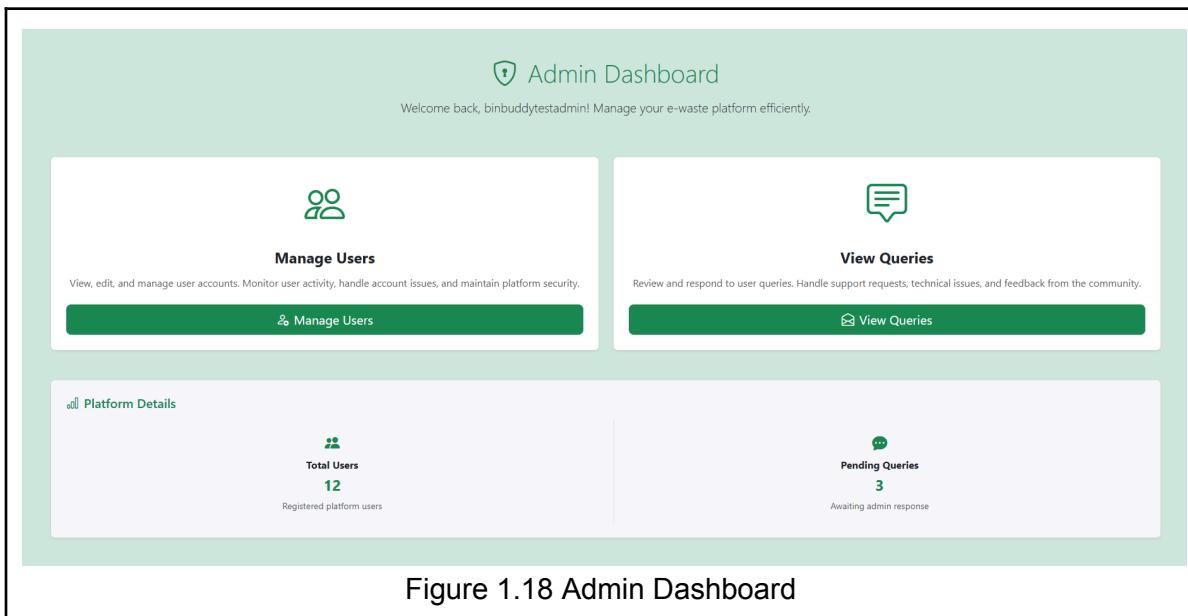


Figure 1.18 Admin Dashboard

3.1.1.13 Manage Users

The screenshot shows a user management interface titled "Manage Users". It displays a list of 12 users with the following details:

Profile	Email	Date Joined	Role	Status	Actions
Tan @t2e209	binbuddy30@gmail.com	November 6, 2025	User	Active	
julia @julia	julialevelup@gmail.com	November 5, 2025	User	Active	
test1 @test1	Test12345@gmail.com	November 5, 2025	User	Active	
sc2006user @sc2006username	arman15forex@gmail.com	November 5, 2025	User	Active	
testAdmin @testAdmin	testAdmin@gmail.com	November 5, 2025	User	Flagged	
Admin @Admin123	Admin123@gmail.com	November 4, 2025	Admin	Active	
testUser2 @testUser2	arman154khan@gmail.com	November 4, 2025	User	Active	
Admin @admin2	Test12345@gmail.com	November 4, 2025	Admin	Active	
Soh Cek Cong @cekong124	cekong123@gmail.com	November 4, 2025	User	Active	
binbuddyuser @binbuddyuser1	binbuddyuser@gmail.com	November 3, 2025	User	Active	
binbuddytestadmin @binbuddytestadmin	noreply.binbuddy.test@gmail.com	November 2, 2025	Admin	Active	
test1234 @sc2006123	test@test.com	November 2, 2025	User	Active	

Figure 1.19 Manage Users

3.1.1.14 Edit Users

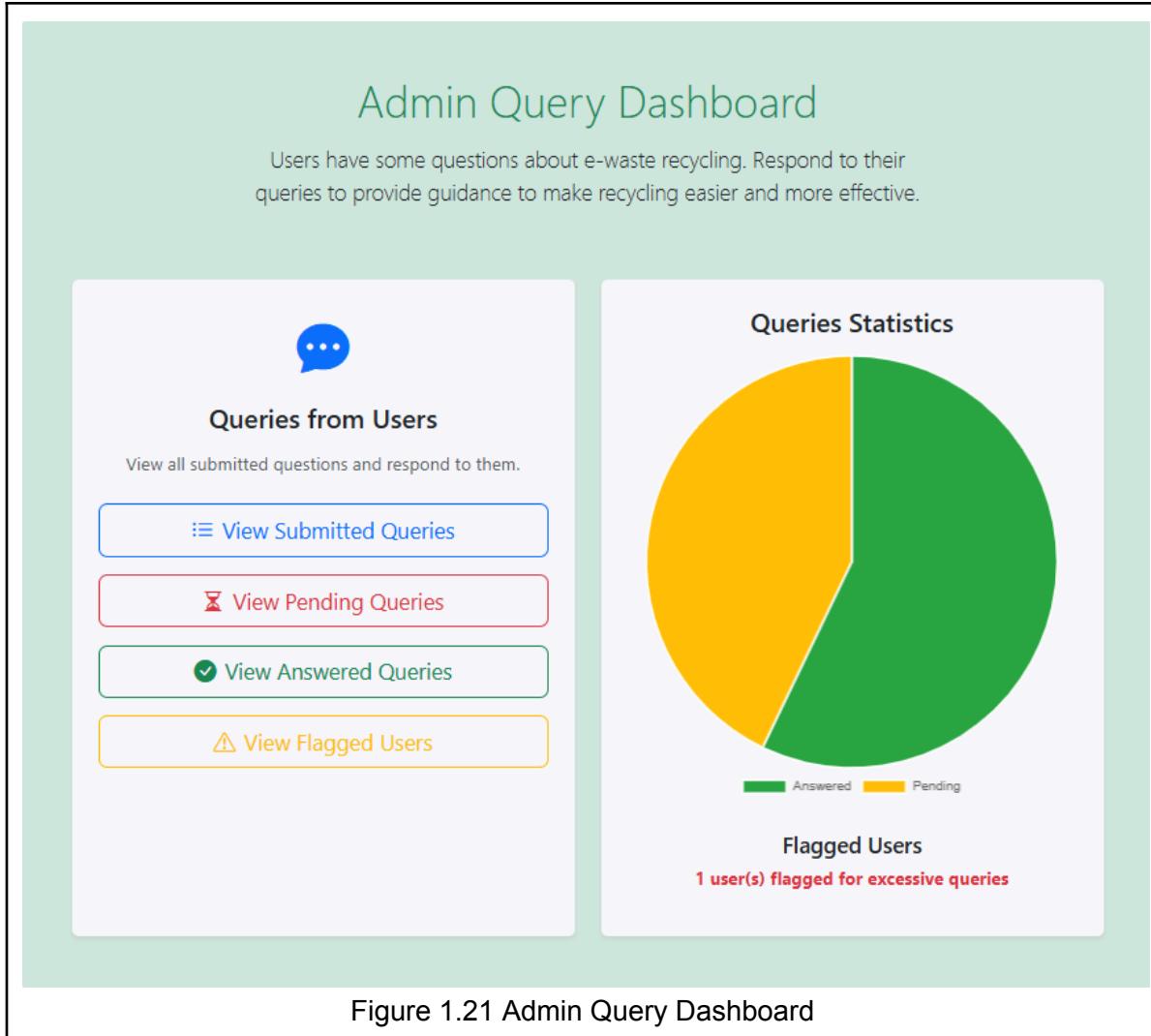
The screenshot shows a user profile edit form titled "Edit User". The form includes the following fields:

- Profile Information**: Update user personal details.
- Full Name**: binbuddyuser
- Username**: binbuddyuser1 (Note: Must be at least 3 characters)
- Email Address**: binbuddyuser@gmail.com (Email cannot be changed)
- Date Joined**: November 3, 2025
- New Password**: Leave blank to keep current password (Enter new password only if you want to change it)
- Confirm Password**: Confirm new password (Must match new password if changing)
- Admin Status**: No
- Flagged**: No

Buttons at the bottom include "Update Profile" and "Reset Picture".

Figure 1.20 Edit Users

3.1.1.15 Admin Query Dashboard



3.1.1.16 View Submitted Queries

All User Queries

View all submitted questions and respond to them.

Filter: All ← Back to Dashboard

How do i dispose appropriately Answered

Created: Tue Nov 04 2025 22:41:36 GMT+0800 (Singapore Standard Time)

Answer:
You need to look at the type of bin and only throw items that are accepted.

Can I throw bulky items at the Bin? Answered

Created: Wed Nov 05 2025 00:21:48 GMT+0800 (Singapore Standard Time)

Answer:
No you can to look at the bin type.

Can I throw e-waste at any Bins? Answered

Created: Wed Nov 05 2025 00:22:10 GMT+0800 (Singapore Standard Time)

Answer:
Only appropriate bins and you need ot look at the waste type.

Why should I throw responsibly? Pending

Created: Wed Nov 05 2025 00:22:26 GMT+0800 (Singapore Standard Time)

Waiting for response...

Type your response here...

Figure 1.22 View Submitted Queries

3.1.1.17 View Pending Queries

The screenshot shows a web application interface titled "All User Queries". At the top, there is a header with a message icon and the title "All User Queries". Below the header, a sub-header says "View all submitted questions and respond to them." There are three main query cards displayed:

- Why should I throw responsibly?** (Created: Wed Nov 05 2025 00:22:26 GMT+0800 (Singapore Standard Time))
Status: Pending
Response status: Waiting for response...
Text input field: Type your response here...
Response button: Respond
- can you help with disposing bigger items** (Created: Wed Nov 05 2025 14:44:15 GMT+0800 (Singapore Standard Time))
Status: Pending
Response status: Waiting for response...
Text input field: Type your response here...
Response button: Respond
- Test** (Created: Thu Nov 06 2025 23:02:16 GMT+0800 (Singapore Standard Time))
Status: Pending
Response status: Waiting for response...
Text input field: Type your response here...
Response button: Respond

At the bottom of the page, there is a caption: "Figure 1.23 View Pending Queries".

3.1.1.18 View Answered Queries

The screenshot shows a web application interface titled "All User Queries". At the top, there is a header with a message icon and the title "All User Queries". Below the header, a sub-header says "View all submitted questions and respond to them." There are two buttons: "Filter:" and "Answered" (with a dropdown arrow), and a "Back to Dashboard" button.

1. How do i dispose appropriately (Answered)

Created: Tue Nov 04 2025 22:41:36 GMT+0800 (Singapore Standard Time)

Answer:
You need to look at the type of bin and only throw items that are accepted.

2. Can I throw bulky items at the Bin? (Answered)

Created: Wed Nov 05 2025 00:21:48 GMT+0800 (Singapore Standard Time)

Answer:
No you can to look at the bin type.

3. Can I throw e-waste at any Bins? (Answered)

Created: Wed Nov 05 2025 00:22:10 GMT+0800 (Singapore Standard Time)

Answer:
Only appropriate bins and you need ot look at the waste type.

4. How do i recycle ewaste (Answered)

Created: Wed Nov 05 2025 16:34:57 GMT+0800 (Singapore Standard Time)

Answer:
burn it

Figure 1.24 View Answered Queries

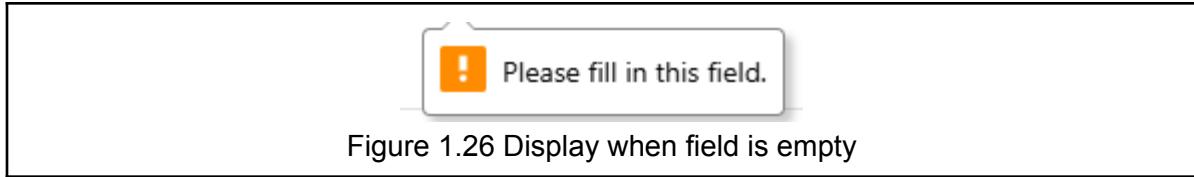
3.1.1.19 View Flagged Users

The screenshot shows a web application interface titled "Flagged Users". The title includes a circular icon with a flag. Below the title, a subtitle says "Manage suspended and flagged user accounts". A table displays user information:

#	Name	Email	Status
0	testAdmin	testAdmin@gmail.com	Suspended

Figure 1.25 View Flagged Users

3.1.1.20 Display when field is empty



3.2 Hardware Interfaces

BinBuddy is designed to operate primarily as a web-based application, minimizing the need for direct hardware interfaces. However, the following outlines the logical and physical characteristics of the interfaces between the software and hardware:

3.2.1 Device Types

The software is designed to be compatible with various devices, including desktop computers, laptops, tablets, and smartphones.

3.2.2 Data and Control Interactions

The user's device interacts with the software through standard input/output hardware such as keyboards, touchscreens, and mice for input; displays for output; and network interface cards for internet connectivity.

3.3 Software Interfaces

Our application consists of interactions between several software components:

3.3.1 Operating System

Any Operating System that is able to run a modern browser software (eg. Google Chrome) is compatible (Windows, macOS, Linux etc.).

3.3.2 Web Browsers

Latest versions of modern web browsers that are capable of supporting HTML5, CSS3 and JavaScript.

3.3.3 Database

MongoDB/NoSQL Databases must be supported. These are accessed through the backend of our system.

The backend communicates with MongoDB through the Mongoose ODM, which provides schema validation and query handling.

3.3.4 Backend Tools

- 1) [Node.js](#) (Version 24.11.0)
- 2) [Express.js](#) (Version 5.1.0)

3.3.5 Frontend Tools

- 1) Handlebars (for HTML templating)
- 2) CSS
- 3) JavaScript

3.3.6 Email Service Interface

Password recovery and notifications are handled using the SendGrid SMTP service through the nodemailer package.

3.3.7 Environment Configuration

Sensitive credentials (API keys, database URLs) are loaded from environment variables defined in the .env file using the dotenv module.

3.3.8 Externals APIs

Google Maps API integration allows geolocation features such as venue location display.

3.4 Communications Interfaces

BinBuddy communicates over standard web protocols to enable client-server interactions. Transport protocol: HTTP/HTTPS for client-server communication. When a user interacts with the UI (such as to view the map), the browser sends a HTTP request to the back end server. The types of request are GET, POST,PUT, DELETE. The express.js server will receive the HTTP request and execute the required logic, database access and API calls. The server then returns an HTTP response with a status code and response body.

All HTTP communications are encrypted using SSL/TLS for security purposes. API keys are also not exposed in our frontend code.

4. System Features

4.1 User Features

4.1.1 Login

4.1.1.1 Description and Priority

Description:

- This feature enables a user to log in into a created account.

Priority:

- High. This is an essential feature as it gives a user the ability to ask queries, set reminders for himself, and save this information to a personal account.

4.1.1.2 Stimulus/Response Sequences

1. To log in, a user will press the Sign In button located at the top right of the application.
 - a. The system displays the login UI with options to input an email address and a password.
2. The user can now input his email address and password:
 - a. The user inputs his email address.
 - i. The system will verify if the email address inputted fulfills the correct format requirements.
 - b. The user inputs his password.
 - i. The system will verify if the password is correct and matches the records in the database
3. Once the user is logged in, he is able to access his own profile.

4.1.1.3 Functional Requirements

REQ-1: Application must verify that the user credentials are correct before authentication.

REQ-2: Application must verify that the email and password fields meet the input format requirements.

4.1.2 Manage Reminders

4.1.2.1 Description and Priority

Description:

- This feature enables a user to manage his own personal reminders. The user is able to set reminders relating to e-waste management. The user is also able to view all his reminders and delete them where

necessary. After the reminder time has passed, the reminder status will change from “Upcoming” to “Past Due”

Priority:

- High. This is an essential feature as it gives a user the ability to set his own personal reminders. This will be tied to personal user accounts and is not something that can be done without an account. .

4.1.2.2 Stimulus/Response Sequences

1. Upon logging in, the user can press “Reminders” on the top of the webpage.
 - a. The system displays the reminder dashboard with options to manage his own reminders
2. The user can create his own reminder:
 - a. The user inputs a reminder message.
 - i. This field cannot be blank.
 - b. The user sets the time for the reminder.
 - i. The system will verify if the time for the reminder has not passed yet.
 - c. A new reminder is successfully created.
3. The user can view his existing reminders:
 - a. The user clicks on “View Reminders” in the reminder dashboard
 - i. The full list of the user’s reminders is then displayed.
 - ii. The user is able to delete any reminder that he does not require anymore.

4.1.2.3 Functional Requirements

REQ-1: Application must verify that the reminder field is not blank.

REQ-2: Application must verify that the time set for the reminder is in the future and has not passed yet.

REQ-3: Application must be able to display the full list of user reminders after successful creation

REQ-4: Application must be able to give the user the option to delete reminders that have been previously created.

4.1.3 Ask Queries

4.1.3.1 Description and Priority

Description:

- This feature enables a user to ask queries to an admin. The user is able to ask anything related to e-waste management or environmental issues.

Priority:

- Medium. This feature is relatively important as users may have specific questions concerning their own personal situations to ask admins on e-waste disposal

4.1.3.2 Stimulus/Response Sequences

1. Upon logging in, the user can press “Queries” on the top of the webpage.
 - a. The system displays the query dashboard.
2. The user can create his own query. The user clicks on “Ask a Question”:
 - a. The user inputs a query.
 - i. This field cannot be blank.
 - b. The user clicks on Submit.
 - c. A new query is successfully created.
3. The user can view his existing queries:
 - a. The user clicks on “View Queries” in the reminder dashboard
 - i. The full list of the user’s queries is then displayed.
 - ii. The user can see which queries have been answered to or are still pending a response.

4.1.3.3 Functional Requirements

REQ-1: Application must verify that the query field is not blank.

REQ-2: Application must be able to display the full list of user queries after successful creation

REQ-3: Application must be able to update the query status from Pending to Answered once the query has been responded to.

4.1.4 Edit Profile

4.1.4.1 Description and Priority

Description:

- This feature enables a user to edit his own profile. The user is able to change his name, username and profile picture

Priority:

- Medium. This feature is relatively important as it allows for user customisation of their own profile

4.1.4.2 Stimulus/Response Sequences

1. Upon logging in, the user can click the profile image on the top right corner
 - a. The user is then directed to the edit profile page.
2. The user can edit his profile information:
 - a. Should the user want to change his full name, he can do so by altering the field.
 - i. This field cannot be blank.
 - b. The user can also change his username and can type a new username in.

- i. This field cannot be blank and must be at least 3 characters long
 - c. The user can also change his profile picture by uploading a local image.
3. Once the changes are made, the user can click “Update Profile” to save the changes.

4.1.4.3 Functional Requirements

REQ-1: Application must verify that the new Full Name and Username is not blank

REQ-2: Application must be able to verify that the new username is at least 3 characters long

REQ-3: Application must be able to give users the ability to upload their own local photos for changing their profile picture

REQ-4: Application must be able to save the user changes once the user clicks on “Update Profile”. These changes must then be updated in the database.

4.1.5 Change Password

4.1.5.1 Description and Priority

Description:

- This feature enables a user to change his own password

Priority:

- High. This feature concerns security of user accounts.

4.1.5.2 Stimulus/Response Sequences

1. Upon logging in, the user can click the profile image on the top right corner
 - a. The user is then directed to the edit profile page.
2. Once this is done, the user can click on “Change Password” on the edit profile page.
3. The user can edit his profile information:
 - a. The user has to enter his existing password first.
 - b. The user next enters the new password that he wants to set. This has to be at least 6 characters long, contain at least 1 numerical character and contain at least 1 special character.
 - c. The user then fills in the confirm new password field to ensure that the new password is confirmed.
 - i. This must match the new password for the new password to be saved.
4. Once the changes are made, the user can click “Update Password” to save the changes.

4.1.5.3 Functional Requirements

REQ-1: Application must verify that the old password is accurate

REQ-2: Application must be able to verify that the new password is at least 6 characters long, with at least 1 numerical character and 1 special character.

REQ-3: Application must be able to verify that the new password and the confirmed new password entries are matching.

4.2 General Features

4.2.1 Educational Material

4.2.1.1 Description and Priority

Description:

- This feature enables a user to view the educational material in BinBuddy. User login is not required

Priority:

- High. It allows users to learn key information concerning the disposal of e-waste.

4.2.1.2 Stimulus/Response Sequences

1. Upon opening BinBuddy, the user can click on “Learn” on the top right corner.
 - a. The system displays all the educational material on the website.
2. The user can then click on the article that he wants to view.

4.2.1.3 Functional Requirements

REQ-1: Application must display all the educational material pages on the main landing page for “Learn”.

4.2.2 Locate Bin

4.2.2.1 Description and Priority

Description:

- This feature enables a user to find e-waste bins based on location.

Priority:

- High. This is a key feature of BinBuddy that enables users to identify where to dispose of waste.

4.2.2.2 Stimulus/Response Sequences

1. The user clicks on “Find Bins” on the top navigation bar.
 - a. The Find Bins main landing page is displayed.
2. The user can then make use of the various filters to find an appropriate bin

- a. The user can type a specific location to find a bin near it.
 - b. The user can also filter by e-waste type to find bins that match the type of e-waste that they want to dispose
 - c. There is also a distance filter present where the user can filter bins based on how far they are from the selected location.
3. The user can also reset the search parameters where necessary.

4.2.2.3 Functional Requirements

REQ-1: Application must display a map of all bins in Singapore

REQ-2: Application must be able to filter bins by type of e-waste accepted.

REQ-3: Application must be able to accept searches based on a typed location.

REQ-4: Application must be able to filter bins by the distance from the specified location.

REQ-5: Application should give users the ability to reset the search parameters.

4.2.3 Create Account

4.2.3.1 Description and Priority

Description:

- This feature enables any user to create an account.

Priority:

- High. This is important as it enables a user to have a personalised account. This will give access to features such as asking queries or setting reminders.

4.2.3.2 Stimulus/Response Sequences

1. A user can click on “Sign In” at the top right corner of the application
2. The user scrolls down and clicks on the link to create a new account
 - a. This launches the account creation page.
3. The user then enters all the relevant fields needed to create an account.
 - a. He is invited to enter a Full name and username. Both of the fields cannot be blank. Additionally, the username must be at least 3 characters long
 - b. The user enters an email address to be tied to the account.
 - c. The user then enters a valid password to be set as the account password. This means the password must be at least 6 characters long, with at least 1 numerical and 1 special character.
 - d. The user then confirms the new password.

4. If all the fields are valid and fulfill the requirements, the user can create an account. The account details are stored in the database

4.2.3.3 Functional Requirements

REQ-1: Application must verify that all relevant fields are not blank

REQ-2: Application must verify that the username is unique, and must be at least 3 characters long.

REQ-3: Application must verify that an email with valid format is inputted. This means the email must contain the character '@'.

REQ-4: Application must verify that the password set conforms to the password requirements.

REQ-5: Application must verify that the confirm password field matches the new password.

4.2.4 Forgot Password

4.2.4.1 Description and Priority

Description:

- This feature enables a user to reset their account password if it has been forgotten.

Priority:

- High. It is an essential feature that allows a user to enter their account if they have forgotten their password.

4.2.4.2 Stimulus/Response Sequences

1. The user clicks on "Sign In" on the top right corner of the screen.
2. The user then clicks on "Forgot Your Password".
3. The user then enters the email associated with the account that he forgot the password for.
 - a. The field to enter an email cannot be blank
4. Once this is done, a recovery email is sent to the user's registered email account.

4.2.4.3 Functional Requirements

REQ-1: Application must be able to verify that the email field is not blank.

REQ-2: Application must have a working mail server in order to send the recovery email instructions.

4.3 Admin Features

4.3.1 Admin Dashboard

4.3.1.1 Description and Priority

Description:

- This feature enables admin to access admin features to manage users and view queries.

Priority:

- High. It allows admin access to admin features to manage applications and assist users.

4.3.1.2 Stimulus/Response Sequences

1. After the admin successfully logs in, it will show the admin dashboard.
 - a. The system displays the admin dashboard with 2 options available to admin: Manage Users and View Queries.
2. Admin can now select an option:
 - a. Admin selects Manage Users option.
 - i. The system will carry out the managed users features, allowing admin to manage users accounts.
 - b. Admin selects View Queries option.
 - i. The system will carry out the view queries features, allowing admin to manage user queries.

4.3.1.3 Functional Requirements

REQ-1: Application must verify that it is an admin before allowing access to administrative functions.

REQ-2: Application must display the options for admin to select.

REQ-3: Application must navigate to the features based on the selected options.

4.3.2 Admin Query Dashboard

4.3.2.1 Description and Priority

Description:

- This feature enables admin to access query features.

Priority:

- Medium.

4.3.2.2 Stimulus/Response Sequences

1. After admin selects the View Queries from the admin dashboard.
 - a. The system displays the admin queries dashboard with 4 options available to admin: View Submitted Queries, View Pending Queries, View Answered Queries and View Flagged Users.

- b. The system also displays the queries statistics in a form of pie chart with 2 fields of pending and answered.
- c. The system also displays the flagged users status if any user(s) is/are flagged for excessive queries.

4.3.2.3 Functional Requirements

REQ-1: Application must display the admin queries dashboard.

REQ-2: Application must display the queries statistics in the form of a pie chart that updates the status live.

REQ-3: Application must display the flagged user status if any user(s) is/are flagged and updates the status live.

4.3.3 View Queries

4.3.3.1 Description and Priority

Description:

- This feature enables the admin to view all queries posted.

Priority:

- Medium.

4.3.3.2 Stimulus/Response Sequences

1. After the admin selects the view queries button, from the admin query dashboard, admin can select these 4 options available: View Submitted Queries, View Pending Queries, View Answered Queries and View Flagged Users.
2. Admin can now select an option:
 - a. Admin select the View Submitted Queries option.
 - i. The System displays the query list, allowing the admin to view all queries.
 - b. Admin select the View Pending Queries option.
 - i. The System displays the pending query list, allowing the admin to view all pending queries.
 - c. Admin select the View Answered Queries option.
 - i. The System displays the answered query list, allowing the admin to view all answered queries.
 - d. Admin select the View Flagged Users option.
 - i. The System displays the flagged user list, allowing the admin to view all flagged users.
3. Admin can toggle between View Submitted Queries, View Pending Queries and View Answered Queries when admin is in any of these 3 options: All, Answered and Pending.
 - a. Admin select the All option.
 - i. The System displays the query list, allowing the admin to view all queries.

- b. Admin select the Pending option.
 - i. The System displays the pending query list, allowing the admin to view all pending queries.
- c. Admin select the Answered option.
 - i. The System displays the answered query list, allowing the admin to view all answered queries.

4.3.3.3 Functional Requirements

REQ-1: The system must display the admin query dashboard with the options available to admin: View Submitted Queries, View Pending Queries, View Answered Queries and View Flagged Users.

REQ-2: The system must display all queries when admin selects View Submitted Queries.

 REQ-2.1: The system must display the selected view option of All.

REQ-3: The system must display pending queries when admin selects View Pending Queries.

 REQ-3.1: The system must display the selected view option of Pending.

REQ-4: The system must display answered queries when admin selects View Answered Queries.

 REQ-4.1: The system must display the selected view option of Answered.

REQ-5: The system must display all flagged users when admin selects Flagged Users.

4.3.4 Answer Queries

4.3.4.1 Description and Priority

Description:

- This feature enables admin to answer to users' queries

Priority:

- Medium. It serves as a communication point between users and admin to address any queries regarding E-waste disposal procedure or to provide feedback with regards to the application itself.

4.3.4.2 Stimulus/Response Sequences

1. After admin successfully logs in, from the admin dashboard, admin will select the view queries button. From the admin query dashboard 2 options available to admin to respond to queries: View Submitted Queries and View Pending Queries.
2. Admin can now select an option:
 - a. Admin select the View Submitted Queries option.
 - i. The System displays the query list, allowing the admin to view all queries.

- b. Admin select the View Pending Queries option.
 - i. The System displays the pending query list, allowing the admin to view all pending queries.
3. Admin can type in the response box the response to the query and press respond button to post the response in these 2 options: View Submitted Queries and View Pending Queries.
 - a. Admin select the View Submitted Queries(ALL) option.
 - i. The System displays the query response box at each pending queries, allowing the admin to input his response.
 - ii. The system posts the response back to users' queries.
 - iii. The system changes the tag from pending to answered.
 - b. Admin select the View Pending Queries(Pending) option.
 - i. The System displays the query response box at each pending queries, allowing the admin to input his response.
 - ii. The system posts the response back to users' queries.
 - iii. The system changes the tag from pending to answered and removes from pending view.

4.3.4.3 Functional Requirements

REQ-1: The system must display the admin query dashboard with the options available to admin: View Submitted Queries and View Pending Queries.

REQ-2: The system must display all queries when admin selects View Submitted Queries.

REQ-2.1: The system must display the response field for pending queries to take in admin response and a response button to submit response.

REQ-2.2: The system must send the response to the user who posted the query.

REQ-2.3: The system must update the tag from pending to answered.

REQ-3: The system must display pending queries when admin selects View Pending Queries.

REQ-3.1: The system must display the response field for pending queries to take in admin response and a response button to submit response.

REQ-3.2: The system must send the response to the user who posted the query.

REQ-3.3: The system must update the tag from pending to answered and remove it from the pending list.

4.3.5 Manage Users

4.3.5.1 Description and Priority

Description:

- This feature enables the admin to view all active users of the application and allow the admin to edit any user's information.

Priority:

- High. It allows admin access to users' profiles and assists with updating of any fields it is able to access.

4.3.5.2 Stimulus/Response Sequences

1. After admin selects Manage Users, all present users will be an option for the admin to select for edit function.
 - a. The system displays the user list with all users becoming an option available to admin: all users.
2. After admin selects a user to edit and press the edit button.
 - a. The system displays the selected user's profile with Username, Full name, Password, Confirm Password, Flagged status, Admin Status and Reset Picture editable. Email and date joined will not be editable.

4.3.5.3 Functional Requirements

REQ-1: Application must display the list of users with an edit button besides each one.

REQ-2: Application must display that user's profile once the edit button of that user is selected.

REQ-3: Application must ensure that both Email and date joined fields are not editable but the rest of the fields (Username, Full name, Password, Confirm Password, Flagged status, Admin Status and Reset Picture) are editable.

4.3.5.4 Change User information

4.3.5.5 Description and Priority

Description:

- This feature enables admin to update users' profile information upon user's request or any violation occurred.

Priority:

- High. It allows admin access to assist users with their profile information.

4.3.5.6 Stimulus/Response Sequences

1. After the admin selects the user to edit, there are up to 6 fields that admin can edit and update the user's profile information.
 - a. The system displays the user's profile information with Username, Full name, Password, Confirm Password, Flagged Status, Admin Status and Reset Picture editable. Email and date joined will not be editable.
2. Admin can now select any field that is editable to edit or choose not to edit:
 - a. Admin edits Full Name field.

- i. The system will display the new input of the selected field.
 - b. Admin edits Username field.
 - i. The system will display the new input of the selected field.
 - c. Admin edits Flagged status field.
 - i. The system will display the new input of the selected field.
 - d. Admin edits Admin status field.
 - i. The system will display the new input of the selected field.
 - e. Admin edits Reset Picture field.
 - i. The system will display the default image and remove the current one.
 - f. Admin edits Password field which is linked together with Confirm Password field.
 - i. The system will display the new input of the selected field.
 - ii. The system requires both fields to be entered with the same input to be considered valid.
3. After admin is done with the edits of at least 1 field, and press the update button, the user profile information will be updated immediately on those fields that are changed.
 - a. The system displays a flash message saying user update successfully.
 4. If the admin does not wish to update any information, the admin can press the cancel button.
 - a. The system will redirect back to the user list after the cancel button is pressed.

4.3.5.7 Functional Requirements

REQ-1: Application must ensure that both Email and date joined fields are not editable but the rest of the fields (Username, Full name, Password, Confirm Password, Flagged status, Admin Status and Reset Picture) are editable.

REQ-2: Application must check the input of the fields and ensure that there is no duplicate fields in relation to the data based that is not the current user's data, then update the field(s) that are amended with new data and update in the system.

REQ-2.1: Application must ensure that when password field is changed, both password and confirm password must match before the change can occur.

REQ-3: Application must ensure the flash message shown user update successfully.

REQ-4: Application must ensure that when the cancel button is pressed, no update is done to the user profile information and redirect back to the user list.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- 1) *Each page should load in no more than 3 seconds, including logging in and authentication.*
- 2) *The reminder date and time should be set from current to the future and no setting of past reminders.*
- 3) *The auto flagging should flag the user as intended when conditions are met.*
- 4) *The map filters should work as intended when toggling the distance, input address or different bin type.*
- 5) *The function of each button must work as intended.*
- 6) *The platform will operate for 24/7, but the administrative works will be done during business hours unless otherwise stated.*

5.2 Safety Requirements

- 1) *Application must not collect the user's live location data after exiting the webpage.*
- 2) *Application must not collect additional personal information besides the required Email for verification purposes and account ownership.*

5.3 Security Requirements

- 1) *Besides the user, only authorised personnel like the admin will be able to access the user's profile information to assist with regards to the user's queries.*
- 2) *The system should implement password hashing for storing the users' passwords to protect users' accounts.*
- 3) *There should be zero tolerance for data leaks or compromise and the database should be stored securely by limiting access to the database to key developers and admins of the application.*
- 4) *The user's live location will only be viewed by the user when using the application and will be accessed by the system when using the map to plot and direct the users to the bins they require*
- 5) *The user has to grant permission to share his live location.*

5.4 Software Quality Attributes

- 1) *The respective functions will be named as its intended purpose by using meaningful names of variables, functions and files to be self-explanatory.*
- 2) *Helpful comments will be written to briefly explain what this part of the code does and will not be overly clustered with comments.*
- 3) *The system will display the appropriate error or success messages when actions are done by users, to improve testability of the product.*
- 4) *API keys used by the system will be changed when the dataset is updated to ensure the business continuity.*

5.5 Business Rules

- 1) Based on the bin information and what type of E-waste is collected at the bins, users must ensure they follow the instructions and dispose of the correct E-waste to the right bins that collect them.

6. Other Requirements

Database Requirements

- Our BinBuddy system uses MongoDB as the primary database to store user account information, reminders and queries.
- All data stored should follow the Mongoose schema structure.
- Passwords should be hashed before storing into the database.
- Data backups should be performed periodically to ensure reliability.

Internationalisation and Localisation requirements

- The initial release of the application supports English as the primary language. Future releases will consider other national languages such as Chinese, Malay and Tamil.
- All date and time is in Singapore timezone (UTC+8).

Legal requirements

- Our BinBuddy system should comply with Singapore's Personal Data Protection Act (PDPA) for the handling of user data.
- The system will display a consent notice requesting for user permission if user data is being used.

Open Data and API usage requirements

- Our BinBuddy system will use publicly available dataset provided by the National Environment Agency (NEA) through data.gov.sg.
- Map visualization features will use the Google Maps API and our system must comply with Google's terms of service, usage and API key security requirements.

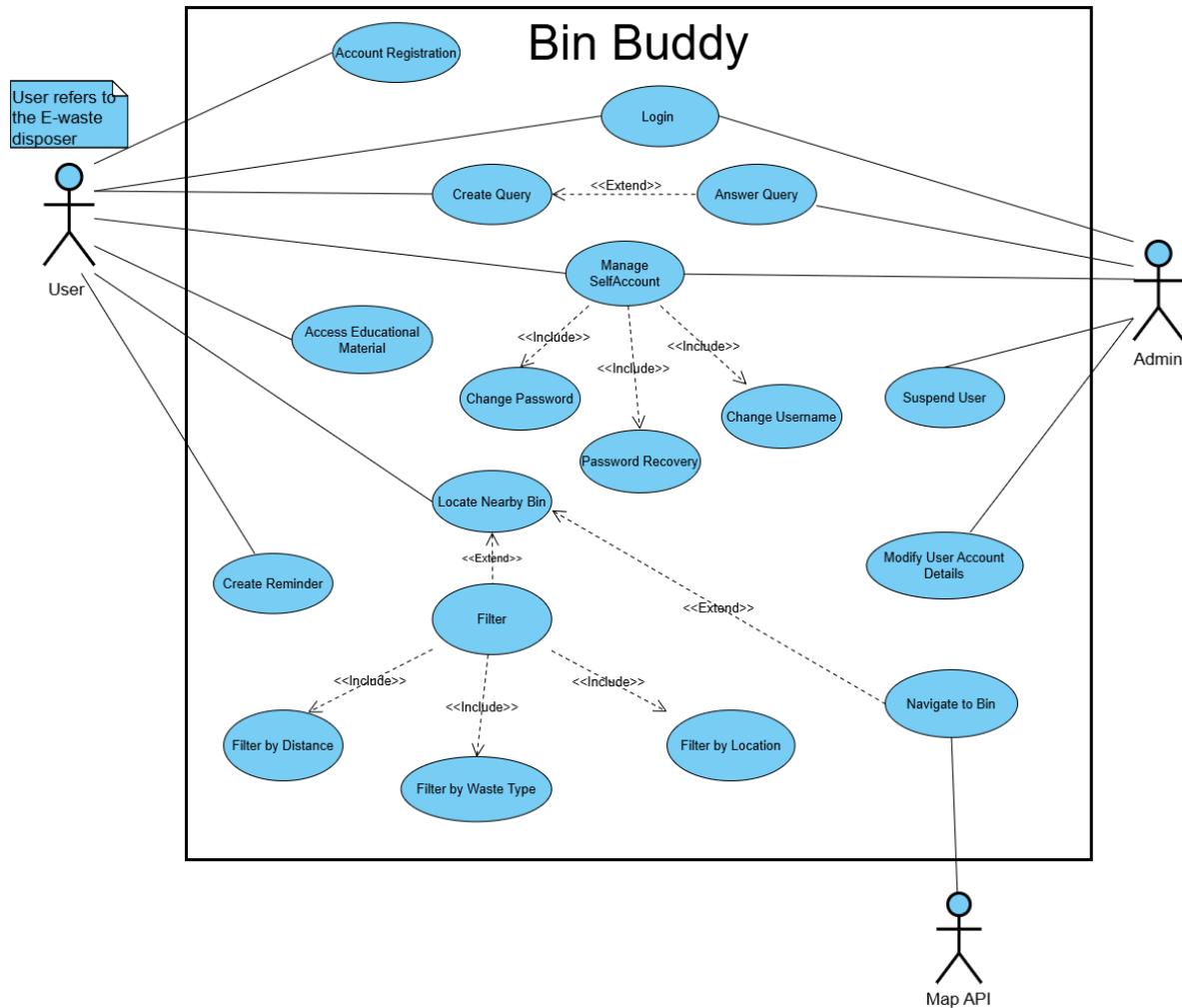
Appendix A: Glossary/Data Dictionary

Term	Definition
E-waste	E-waste is electrical and electronic equipment of any kind that has been discarded. It includes practically anything powered by an electrical source. Common types of E-waste include the following: desktops, laptops, tablets, mobile phones, refrigerators, batteries, and lamps.
User	User refers to anyone who wishes to dispose of their E-waste.
Admin	Admin refers to administrators who manage and oversee the app. Admins have privileges that allow them to remove accounts or to modify data in the database.

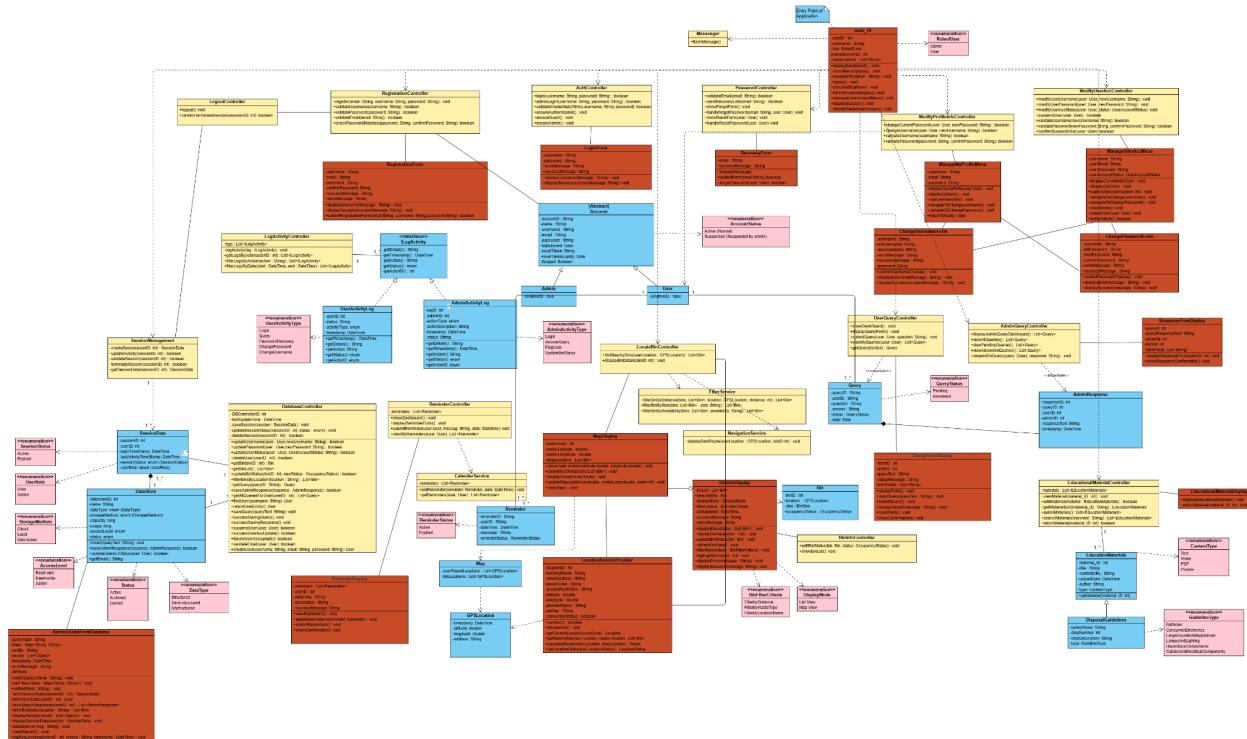
E-waste bins	E-waste bins are bins that are designed for e-waste collection. There are more than 870 E-waste collection points across Singapore. The bin sizes come in different sizes - small is suitable for batteries, phones and bulbs; medium is suitable for laptops; large is suitable for monitors and printers. The suitability for each bin is specified on the App itself.
Account Information	Account Information refers to the email, username and password of an account. The username and password of an account can be changed, but the email of an account can't be changed. The username must be between 5 to 30 letters, while the password must be between 10 to 30 letters.
Query	A query can be submitted by users when they want to submit a clarification to the administrators. The administrators will then answer the query.
Ban User	An admin can ban a user account if unsavoury activities are noticed, such as spamming queries.
Occupancy	The extent to which a particular E-waste disposal bin is filled to its maximum capacity.
Small-sized items	Batteries, phones and bulbs.
Medium-sized items	Laptops
Large-sized items	Monitors and printers.

Appendix B: Analysis Models

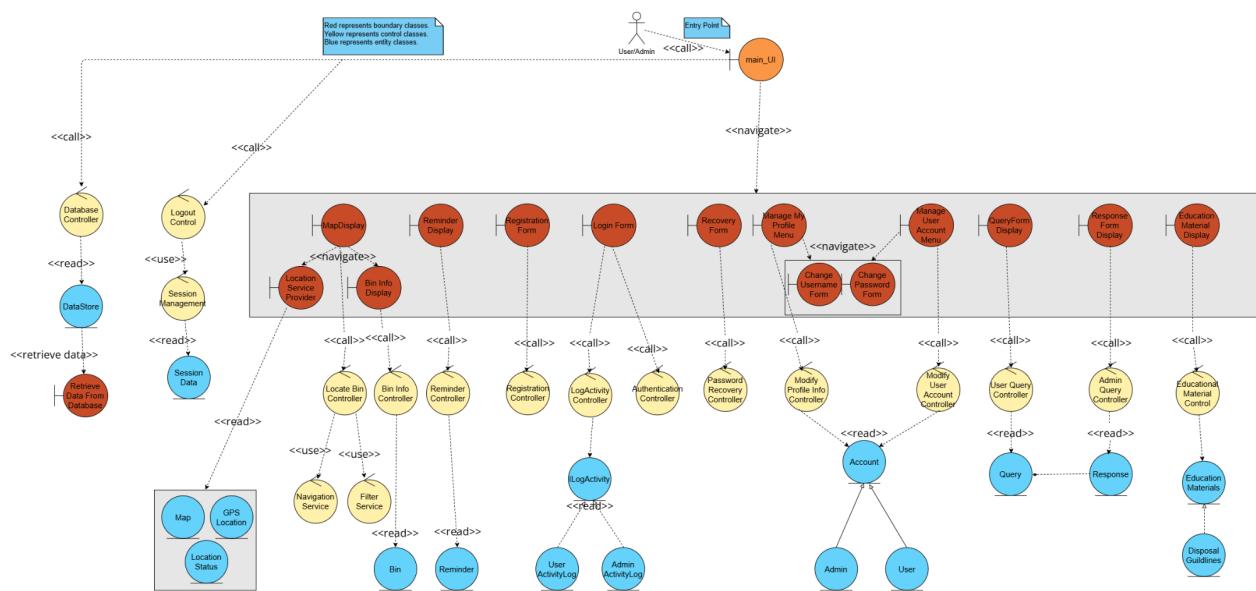
I. Use Case Model



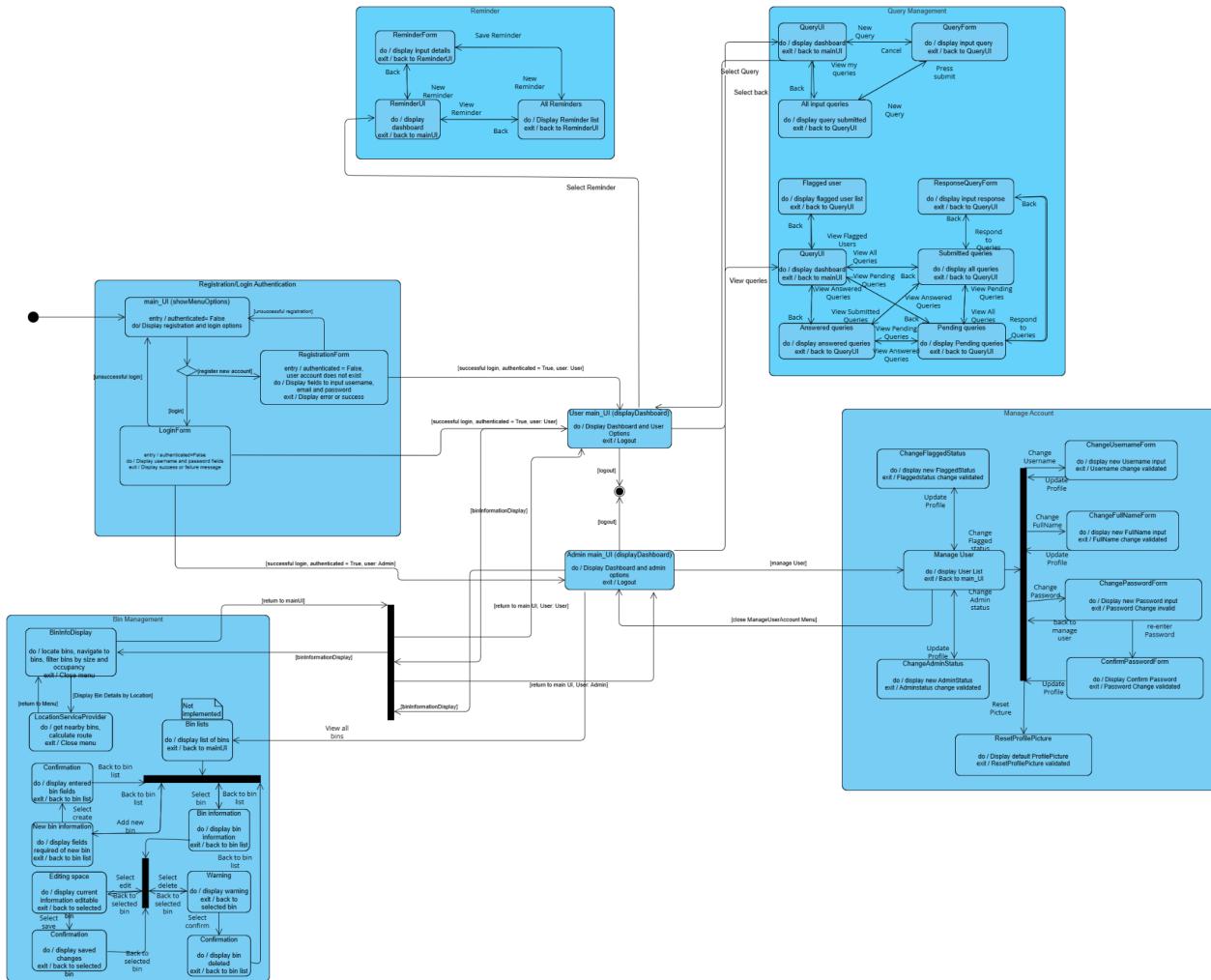
II. Class Diagram



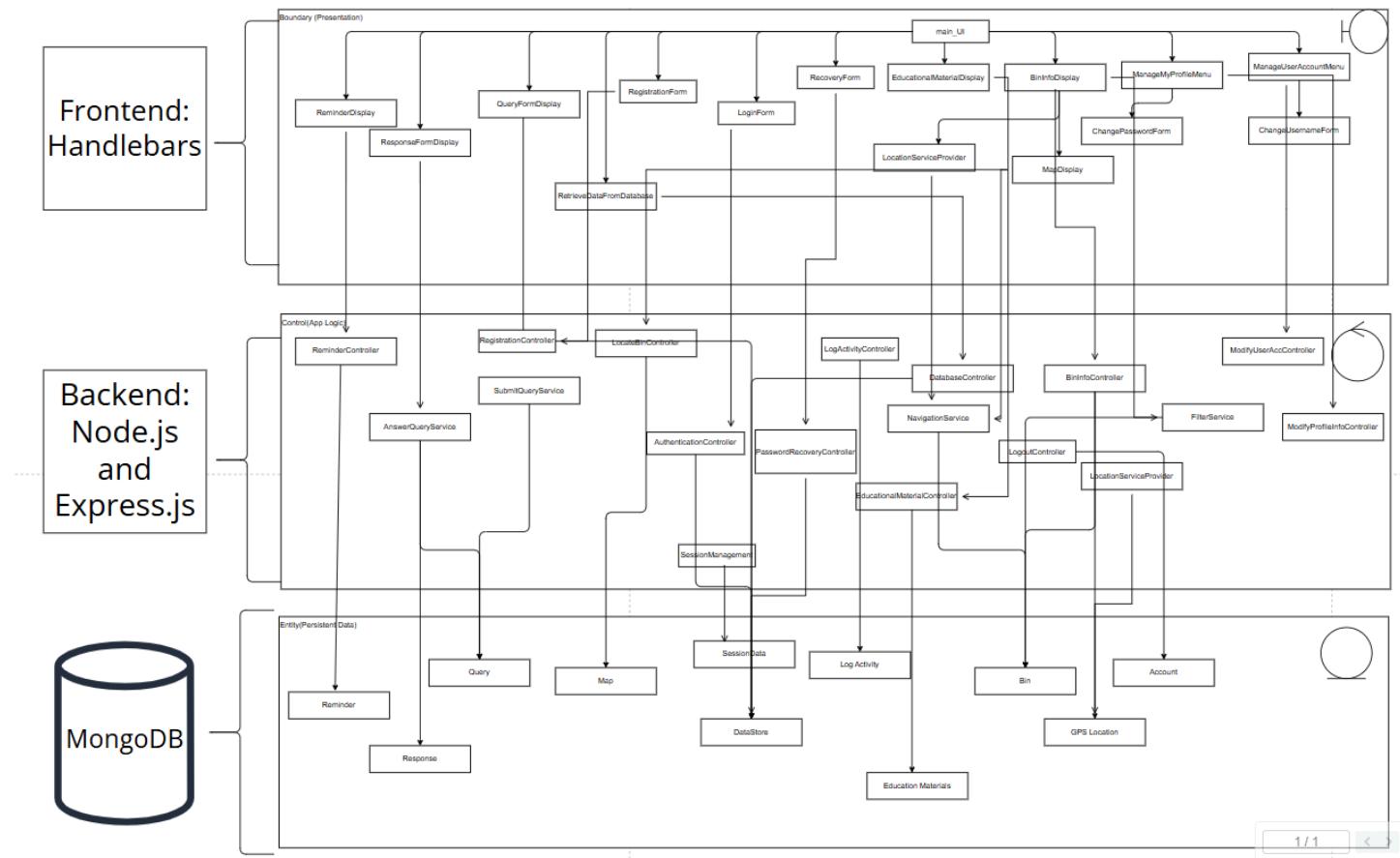
III. Class Stereotype



IV. Dialog Map



V. System Architecture



VI. Application Skeleton

```

    ✓ controllers
      JS adminQueryController.js
      JS passwordController.js
      JS reminderController.js
      JS userQueryController.js
    ✓ helpers
      JS hbs.js
      JS messenger.js
    ✓ middleware
      JS auth.js
    ✓ models
      JS Query.js
      JS Reminder.js
      JS User.js
    > node_modules
    > public
    > routes
    > services
    > views
    ⚙ .env
    ⚡ .gitignore
    JS app.js
    {} package-lock.json
    {} package.json
    ⓘ README.md
  
```

Controllers: Containing controller files that define the logic and behavior of the app when certain routes are triggered. They handle user input, interact with the database and render responses.

Helpers: Folder to put utility codes that extend our app's functionalities.

Middleware: Folder containing functions that run before or after route handlers.

Models: Folder containing the Mongoose schema that describes how data is stored in the MongoDB database.

Node_modules: Folder containing all installed dependencies from npm (Express, Handlebars etc) that our app relies on.

Public:

- CSS: Stylesheets for frontend design.
- Img: All images used in the app.
- JS: Client-side JavaScript files.

Routes: Folder containing application logics for our webpages.

Views: Folder containing handlebars templates which defines how our pages look. This includes our base.handlebars which defines the overall HTML structure of our app, and home.handlebars which is the template of our homepage.

Root Files:

- .env: Stores environment variables like database URLs and API keys, which are kept secret.
- .gitignore: List of files that are not uploaded to github.

- App.js : Entry point of the server, it sets up express, connects to MongoDB, loads routes and start the app.
- Package.json
- Package-lock.json

Appendix C: To Be Determined List

There are no TBD items for this SRS.