

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

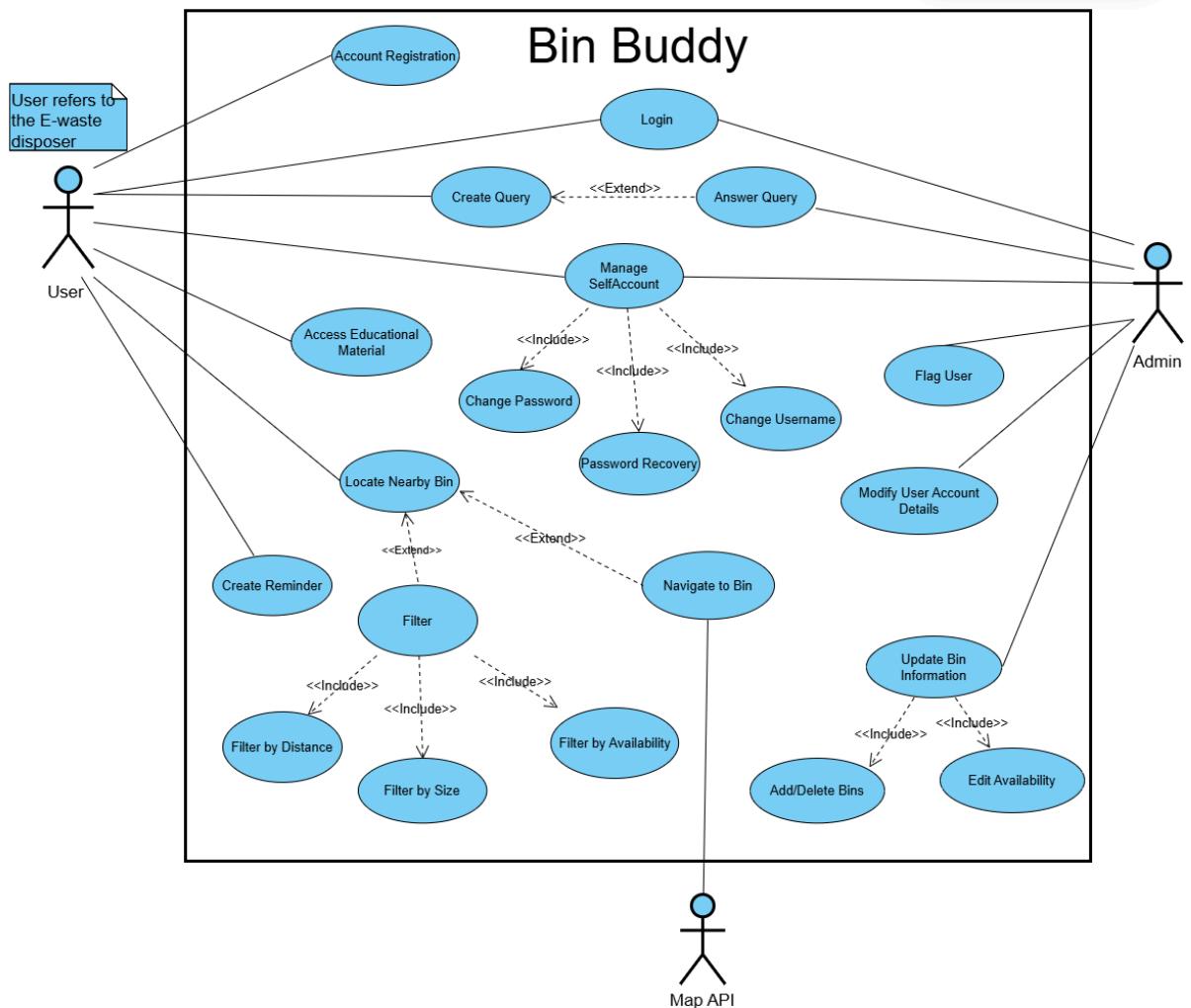
SC2006 - Software Engineering

Bin Buddy - An E-waste Recycling App

Lab 3 Deliverables

Lab Group	SCSI
Team	Group 3
Title	Bin Buddy
Member	HTOO MYAT NOE (U2422977K)
	ARMAN KHAN (U2421760A)
	CHUA YUE JUN (U2423015D)
	NIKHIL MADETI (U2421243D)
	SOH CEK CONG (U2423500C)
	YOONG HONG JUN, NICHOLAS (U2321582L)

I. Use Case Model



II. Use Case Descriptions

1. Account Registration

Use Case ID:	UC1		
Use Case Name:	Account Registration		
Created By:	Htoo Myat Noe	Last Updated By:	Htoo Myat Noe
Date Created:	24 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User (E-waste disposer)
Description:	The E-waste disposer user may register for a new account within the E-waste recycling system in order to gain personalized access and convenience features. Registration enables users to securely store

	<p>their personal details and home/work location, making it easier to locate nearby recycling bins without having to repeatedly input their information. By creating an account, the user gains access to a persistent profile that can be used for future interactions such as checking nearby bins, receiving notifications to remind them on their next recycling trip, and bin availability in their vicinity.</p> <p>During registration, the user provides basic details such as their username and password credentials. The system ensures that user information is valid, unique (no duplicate username), and compliant with password security requirements. The account creation process is designed to be simple and user-friendly while maintaining strict security standards.</p> <p>The successful creation of an account establishes the user's identity within the system, enabling them to log in with their username.</p>
Preconditions:	<ul style="list-style-type: none"> • The user can access the application. • The user's device has internet connectivity. • The user does not already have an account (for the same email/username).
Postconditions:	<ul style="list-style-type: none"> • A new account is created and stored in the database. • The user can log in using the newly created credentials.
Priority:	High
Frequency of Use:	Low (usually only done once for every account to be created)
Flow of Events:	<ol style="list-style-type: none"> 1. The user opens the portal. 2. The user selects "Register / Create new Account." 3. The user enters required information: <ul style="list-style-type: none"> ○ Email ○ Username ○ Password (with confirmation) 4. The user clicks "Sign up" 5. System validates input: <ul style="list-style-type: none"> ○ Checks if email already exists. ○ Ensures password is between 8 to 30 characters, with at least a number and a special character. 6. If validation passes, the system creates the account in the database. 7. The system displays "Registration successful" message and provides the option to log in.
Alternative Flows:	<p>If the user provides missing or invalid input, the system displays an error message specifying the missing/incorrect field (e.g., "Password too short"). The user will correct the input and resubmit.</p> <p>If the email already exists, the system displays: "An account with this email already exists." The user can choose to log in or use a different email.</p> <p>If the registration fails due to network/database issues, the system</p>

	displays: “Registration failed. Please try again later.” The user can retry after the internet connection stabilises or wait for the system recovery.
Exceptions:	If the system cannot connect to the database, display “Account registration service unavailable. Please try again later.”
Includes:	NIL
Special Requirements:	Clear instructions for password requirements (e.g., min. 8 characters, max. 30 characters 1 number, 1 special character).
Assumptions:	The user provides accurate information for registration.
Notes and Issues:	<p>Consider adding biometric login (fingerprint/face ID) as a future enhancement.</p> <p>Consider Multi-Factor Authentication for enhanced security.</p> <p>Consider a session timeout for inactivity.</p> <p>Consider secure storage of passwords (hashed and salted).</p>

2. Login (User)

Use Case ID:	UC2		
Use Case Name:	Login (User)		
Created By:	Htoo Myat Noe	Last Updated By:	Nicholas
Date Created:	24 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User (e-waste disposer)
Description:	<p>The E-waste disposer user may log in to the E-waste recycling application to gain secure access to its personalized features and services. Logging in establishes the user's identity within the system and ensures that their account data, such as saved locations, are available. This allows users to enjoy a more seamless experience compared to guest access.</p> <p>Through the login process, users can conveniently access features such as locating nearby recycling bins tailored to their stored address, retrieving disposal guidelines for different categories of electronic waste, and viewing updates or notifications related to recycling trips. The authentication process also safeguards sensitive user data and ensures that only authorized users gain access to the account.</p> <p>The login process typically requires the user to provide registered credentials (email/username and password). The system validates</p>

	<p>these details against securely stored data in the database. On successful authentication, the user is granted access to the main dashboard of the application, where all features are accessible. This process reinforces trust and security, while also enabling the app to deliver a personalized and engaging experience.</p> <p>In essence, login serves as the critical gateway between the user and the E-waste recycling ecosystem, ensuring that users can securely interact with the system while enabling the application to provide tailored services that encourage proper recycling practices.</p>
Preconditions:	<ul style="list-style-type: none"> • The user can access the application. • The user's device has internet connectivity. • The user has previously registered an account.
Postconditions:	<ul style="list-style-type: none"> • The user is successfully logged into the app and can access personalized features.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user opens the app. 2. The user selects "Login". 3. The user enters personal credentials (email address and password). 4. The user clicks "Login". 5. The system verifies the credentials with the database. 6. If credentials are correct, the system logs in the user and redirects to the home/dashboard screen.
Alternative Flows:	<p>If the user forgets his/her password, he/she can click "Forgot Password". The system prompts the user to enter the registered email. The system sends a password reset link to the email. The user resets the password and returns to the login flow.</p> <p>If the user tries to log in without an internet connection, the system displays: "Internet connection required. Please connect and try again."</p> <p>If the user enters incorrect credentials, the system displays: "Incorrect username or password. Please try again." The user can retry entering credentials.</p>
Exceptions:	<p>If the system cannot connect to the database, display "Login service unavailable. Please try again later."</p> <p>If the account is locked due to multiple failed login attempts (max 5), the system displays "Account temporarily locked. Please reset your password."</p>
Includes:	Account Registration: Optional, if the user does not have an account.
Special	Nil

Requirements:	
Assumptions:	<ul style="list-style-type: none"> Users have a valid account to log in. The user's device has a working internet connection.
Notes and Issues:	<p>Users can only have up to 5 login attempts (for security purposes).</p> <p>Consider adding biometric login (fingerprint/face ID) as a future enhancement. Optional integration with OAuth/Google/Facebook login for convenience.</p> <p>Error messages should be clear but not reveal sensitive info (e.g., avoid stating whether username or password is incorrect).</p> <p>Consider logging of admin login attempts for audit purposes.</p> <p>Secure storage of passwords (hashed and salted).</p>

3. Login (Admin)

Use Case ID:	UC3		
Use Case Name:	Login		
Created By:	Htoo Myat Noe	Last Updated By:	Cek Cong
Date Created:	24 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	Admin
Description:	<p>The Admin user logs in to the E-waste management application to gain secure access to administrative features that are not available to regular users. Through the login process, the admin establishes their identity and authorization level, ensuring that only verified personnel can manage critical aspects of the system. This protects sensitive operational data and prevents unauthorized modifications.</p> <p>Once logged in, the admin gains access to a dedicated dashboard that provides tools for overseeing the overall recycling system. These tools include updating bin location details, adjusting occupancy levels, adding or removing bins when necessary, and monitoring user activity. The admin can also generate and review statistical reports to gain insights into bin usage trends, recycling participation levels, and other key performance metrics. These features support informed decision-making and ongoing system maintenance.</p> <p>The login process requires the admin to supply valid credentials (email/username and password), which are verified against the secure admin database. Successful login not only confirms identity but also initiates a secure session that allows the admin to perform high-level tasks with accountability. Security is paramount in this process, as administrative actions can significantly impact the</p>

	<p>usability and reliability of the entire E-waste management system.</p> <p>In essence, the admin login function is the gateway to the “control center” of the E-waste management platform. By restricting access to trusted personnel, the system ensures integrity, security, and smooth operation of the recycling ecosystem while enabling administrators to maintain accurate, up-to-date information for end-users.</p>
Preconditions:	<ul style="list-style-type: none"> • The admin can access the application. • The admin's device has internet connectivity. • The admin has the essential credentials to log into the admin account, and is authorised to do so.
Postconditions:	<p>Admin is successfully logged in and can access administrative functions such as:</p> <ul style="list-style-type: none"> • Updating bin locations and occupancy data • Adding or removing bins • Viewing usage statistics and reports
Priority:	High
Frequency of Use:	Medium (used periodically for maintenance)
Flow of Events:	<ol style="list-style-type: none"> 1. The admin opens the app. 2. The admin selects “Login”. 3. The admin enters credentials (email address and password). 4. The admin clicks “Login”. 5. The system verifies credentials against the admin database. 6. If credentials are correct, the system logs in the admin and redirects to the admin dashboard.
Alternative Flows:	<p>If the admin forgets his/her password, he/she can click “Forgot Password”. The system prompts the admin to enter the registered email. The system sends a password reset link to the email. The admin resets password and returns to login flow.</p> <p>If the admin tries to log in without an internet connection, the system displays: “Internet connection required. Please connect and try again.”</p> <p>If the admin enters incorrect credentials, the system displays: “Incorrect username or password. Please try again.” He/she can retry entering credentials (for up to 3 tries).</p>
Exceptions:	<p>If the system cannot connect to the database, display “Login service unavailable. Please try again later.”</p> <p>If the account is locked due to multiple failed login attempts, display “Account temporarily locked. Please try again after 10 minutes.”</p>
Includes:	NIL
Special Requirements:	Optional integration with OAuth/Google/Facebook login for convenience.

Assumptions:	<ul style="list-style-type: none"> • Admin has a valid account to log in. • Admin's device has a working internet connection.
Notes and Issues:	<p>The admins can only have up to 5 login attempts (for security purposes).</p> <p>Consider adding biometric login (fingerprint/face ID) as a future enhancement.</p> <p>Consider Multi-Factor Authentication for enhanced security.</p> <p>Consider a session timeout for inactivity.</p> <p>Consider logging of admin login attempts for audit purposes.</p> <p>Secure storage of passwords (hashed and salted).</p> <p>Error messages should be clear but not reveal sensitive info (e.g., avoid stating whether username or password is incorrect).</p>

4. Manage Account

Use Case ID:	UC4		
Use Case Name:	Manage Account		
Created By:	Nicholas	Last Updated By:	Myat Noe
Date Created:	26 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User / Admin
Description:	<p>Both users and admins may occasionally need to modify their account details within the E-waste management application. These modifications typically include updating their username for personalization or security reasons, or changing their password to maintain account safety. Over time, actors may forget their credentials or enter incorrect passwords multiple times, which may result in temporary account lockouts as part of the system's security protocol. To address this, the system provides a password recovery function that ensures actors can securely regain access to their accounts.</p> <p>For routine account management, an actor who is already logged in can access the "Manage Account" section. This area displays their current information and allows them to update sensitive details such as username and password. Changes are validated by the system and securely stored in the database. By allowing actors to update their information, the system supports both personalization and ongoing security practices, such as encouraging stronger passwords.</p> <p>For password recovery, the system ensures that even if access credentials are forgotten or compromised, actors can safely reset</p>

	<p>their accounts. By verifying the actor's identity through their registered email and username, the system sends a secure password reset link to the associated email account. This ensures that only the rightful account owner can initiate the reset. Once the reset link is accessed, the actor can create a new password, which replaces the old one in the database. This prevents unauthorized access while restoring system usability.</p> <p>Overall, the account management and password recovery functions are critical components of the E-waste management system. They ensure that both users and admins can maintain secure, up-to-date accounts while preserving system integrity and providing a seamless way to regain access in case of forgotten credentials.</p>
Preconditions:	<ul style="list-style-type: none"> • The actor can access the web portal and has internet connectivity. • The actor has logged in. (for changing of password and username). • The actor has access to his/her email. (for password recovery only).
Postconditions:	<ul style="list-style-type: none"> • The actor's account information will be updated in the system. (for changing of password and username) • An email will be sent to the corresponding email address where the actor can reset their password. (for password recovery)
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<p><u>For changing account information:</u></p> <ol style="list-style-type: none"> 1. The actor will click on the "Manage Account". 2. The actor will be presented with their current information. He/she will type in the new username or password in their respective input field and click save. <p><u>For password recovery:</u></p> <ol style="list-style-type: none"> 1. The actor opens the portal. 2. The actor selects "Forgot Password" 3. The actor enters their username and email used to register the account 4. If the username and email match, an email will be sent to the corresponding email address where the actor can reset their password. This can only be done through the link in the email. 5. The account credentials are updated with the new password.
Alternative Flows:	NIL
Exceptions:	If the system cannot connect to the database, display "Please try again later."
Includes:	<ul style="list-style-type: none"> • Password Recover • Change Password • Change Username

	<ul style="list-style-type: none"> • Login
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

5. Password Recovery

Use Case ID:	UC5		
Use Case Name:	Password Recovery		
Created By:	Nicholas	Last Updated By:	Nicholas
Date Created:	26 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User/Admin
Description:	The actor has forgotten their password. The actor needs to reset or recover their password in order to log in.
Preconditions:	<ul style="list-style-type: none"> • The actor can access the web portal and has internet connectivity. • The actor has logged in. (for changing of password and username).
Postconditions:	<ul style="list-style-type: none"> • The user/admin account is updated with a new password, with the old password being replaced.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The actor opens the portal. 2. The actor selects “Forgot Password”. 3. The actor enters his/her email that was used to register the account. 4. An email will be sent to the corresponding email address where the actor can reset their password. This can only be done through the link in the email. 5. The actor should ensure that the new password is between 8-30 characters with at least 1 number and a special character. 6. The account credentials are updated with the new password.
Alternative Flows:	If the actor provides missing or invalid input, the system displays an error message specifying the missing/incorrect field (e.g., “The email address does not have an account”). The actor has to correct the input and resubmit.

	If the password reset fails due to network/database issues, the system displays: "Please try again later." The actor can retry after ensuring an internet connection or waiting for system recovery.
Exceptions:	If the system cannot connect to the database, display "Please try again later."
Includes:	NIL
Special Requirements:	The actor has to be able to access his email account to be able to access the recovery link.
Assumptions:	The actor can recall the email used to create the account.
Notes and Issues:	NIL

6. Change Password

Use Case ID:	UC6		
Use Case Name:	Change Password		
Created By:	Nicholas	Last Updated By:	Myat Noe
Date Created:	26 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User / Admin
Description:	Users or admins may wish to change their password.
Preconditions:	<ul style="list-style-type: none"> The actor can access the web portal and has internet connectivity. The actor has logged in.
Postconditions:	<ul style="list-style-type: none"> The actor's account password will be updated in the system.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> The actor navigates to the application's main interface after logging in. The actor selects the "Manage Account" option from the menu. The system retrieves the actor's current account information and displays it on the screen (excluding sensitive data such as the current password). The actor clicks on the "Change Password" section. The system presents input fields for: <ul style="list-style-type: none"> - Current password (for verification) - New password

	<ul style="list-style-type: none"> - Confirm new password <p>6. The actor enters their current password and provides the new password in the designated field, then re-enters the new password in the confirmation field.</p> <p>7. The actor clicks “Save” to submit the change.</p> <p>8. The system validates the following:</p> <ul style="list-style-type: none"> - The current password matches the one stored in the database. - New password and confirmation match. - The new password meets security criteria: length of 8–30 characters, contains at least one number, and at least one special character. <p>9. If all validations pass, the system securely updates the password in the database</p> <p>10. The system displays a success message: “Password updated successfully.”</p> <p>11. The actor is redirected to their Manage Account dashboard and can continue using the application with the updated credentials.</p>
Alternative Flows:	<ol style="list-style-type: none"> 1. If the actor types a password that is not strong enough or does not meet the criteria, the system will show a message “The password does not meet the criteria, please try again”. 2. The actor will key in the new password again until the system approves.
Exceptions:	If the system cannot connect to the database, display “Please try again later.”
Includes:	Login: the actor has to log in to change his/her password.
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

7. Change Username

Use Case ID:	UC7		
Use Case Name:	Change Username		
Created By:	Nicholas	Last Updated By:	Nicholas
Date Created:	26 Aug 2025	Date Last Updated:	31 Aug 2025

Actor:	User / Admin
--------	--------------

Description:	Users or admin may wish to change their username.
Preconditions:	<ul style="list-style-type: none"> The actor can access the web portal and has internet connectivity. The actor has logged in.
Postconditions:	The actor's account username will be updated in the system.
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> The actor logs into the system and navigates to the application's main interface. The actor selects the "Manage Account" option from the menu. The system retrieves the actor's current account information (excluding sensitive details such as the password) and displays it on the screen. The actor clicks on the "Change Username" section. The system presents an input field for the actor to enter a new username. The actor types in the desired new username. The actor clicks "Save" to submit the change request. The system performs validation checks: <ul style="list-style-type: none"> Ensures the new username is between 5–30 characters in length. Ensures the username is not already taken by another registered account. Ensures the username does not contain invalid or restricted characters (e.g., spaces). If all validations pass, the system updates the username in the database. The system displays a confirmation message: "Username updated successfully." The actor is redirected back to the Manage Account dashboard, where the updated username is now visible.
Alternative Flows:	<ol style="list-style-type: none"> If the actor types a username that is not between 8 to 30 characters, the system will show a message "The username is not between 8 to 30 characters, please try again". The actor will key in the new username again until the system approves.
Exceptions:	If the system cannot connect to the database, display "Please try again later."
Includes:	Login: the actor has to log in to change his/her username.
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

8. Access Educational Material

Use Case ID:	UC8		
Use Case Name:	Access Educational Material		
Created By:	Nikhil	Last Updated By:	Yue Jun
Date Created:	28 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User
Description:	The user wants to be able to view the disposal guidelines for a selected e-waste bin or educational videos concerning the environment/proper waste disposal processes.
Preconditions:	<ul style="list-style-type: none"> • The actor can access the application and has internet connectivity. • The actor has logged in.
Postconditions:	<ul style="list-style-type: none"> • User successfully views the educational content and logs out of the system
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The user logs into the application. 2. From the home or dashboard screen, the user navigates to the “Educational Materials” or “Disposal Guidelines” tab/menu option. 3. The system retrieves and displays a list of available educational resources (e.g., video thumbnails, guideline categories). 4. The user browses through the available content. 5. The user selects the specific guideline or video they wish to view. 6. The system loads and displays the chosen material: <ul style="list-style-type: none"> - For guidelines: the system displays formatted text instructions with optional downloadable resources. - For videos: the system streams the video content either from the app's backend server or via an integrated video platform (e.g., YouTube). 7. The user views the material and may choose to: <ul style="list-style-type: none"> - Return to the educational materials list. - Select another resource to view. - Exit the section and return to the main dashboard.
Alternative Flows:	If the user attempts to view the videos or guidelines without an internet connection, the system displays: “Internet connection required. Please connect and try again.”

Exceptions:	If the educational materials cannot be loaded properly, or are not stored in the backend server properly, an error message will display to indicate that the resource requested is unavailable.
Includes:	Login (User): The user has to be logged in to access this functionality.
Special Requirements:	Optional integration with YouTube or other video hosting websites
Assumptions:	<ul style="list-style-type: none"> • User account is valid. • All guidelines and educational videos are loaded properly and are accessible to users.
Notes and Issues:	Ensure timely updates of educational videos and guidelines. Videos should be embedded in the web application properly.

9. Locate Nearby Bin

Use Case ID:	UC9		
Use Case Name:	Locate Nearby Bins		
Created By:	Myat Noe	Last Updated By:	Myat Noe
Date Created:	26 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User
Description:	The user wants to view the locations of nearby e-waste bins. This use case allows e-waste disposers to locate the nearest e-waste bins based on their current locations.
Preconditions:	<ul style="list-style-type: none"> • The user can access the application and has internet connectivity. • The user has logged in. • Global Positioning System (GPS) is enabled on the user's device. • The system has the latest information on the bins' location.
Postconditions:	<ul style="list-style-type: none"> • A map of nearby e-waste bins will be shown on the user's device, and the user may be instructed on how to navigate there if the user wishes to be directed.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user opens the app and selects "Locate Bin".

	<ol style="list-style-type: none"> 2. The system requests permission from the user to access his/her location. 3. The user grants location access. 4. The system retrieves the user's current location. 5. The system searches in the database for nearby bins. 6. The system displays a map with markers for nearby bins and a list view. 7. The user can select bins to view their details (address, type of e-waste accepted, availability).
Alternative Flows:	<p>If the user does not grant precision to access his/her location, the system will display a message "The app requires location permission to work". The user can either:</p> <ol style="list-style-type: none"> 1) Grant permission → return to the main flow. 2) Cancel → system returns to the home screen. <p>If the user is not in Singapore, the system will display a message saying it's not available in his/her country. The system will offer options to the user:</p> <ol style="list-style-type: none"> 1) Go back to the home screen 2) View educational content/videos, which are available internationally.
Exceptions:	If the map cannot be loaded properly, or data is not stored in the backend server properly, an error message will display to indicate that the resource requested is unavailable.
Includes:	Login
Special Requirements:	NIL
Assumptions:	The user's device has GPS and internet access.
Notes and Issues:	NIL

10. Filter

Use Case ID:	UC10		
Use Case Name:	Filter		
Created By:	Cek Cong	Last Updated By:	Cek Cong
Date Created:	25 Aug 2025	Date Last Updated:	28 Aug 2025

Actor:	User	
Description:	The user wants to be able to filter the bins to identify the most	

	suitable bins for the disposal of e-waste. The user will be able to enter the type of waste that they want to dispose of, as well as the estimated size/quantity.
Preconditions:	<ul style="list-style-type: none"> • The user can access the application and has internet connectivity. • The user has logged in. • Global Positioning System (GPS) is enabled on the user's device. • The system has the latest information on the bins' location. • The system has a tracker to estimate the remaining capacity of bins/ can be manually edited by admins.
Postconditions:	<ul style="list-style-type: none"> • The user can identify a list of suitable bins for his disposal purposes
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The user logs in with a valid username and password. 2. The system verifies credentials and grants access to the main dashboard. 3. The user enters the app and selects "Filter Bins". 4. The system loads the filter interface and displays available filtering options. 5. Subsequently, the user will be invited to select the type of e-waste he wants to dispose of from a drop down box. 6. The system displays a reference image and example items for each category to guide user input. 7. The user confirms the selection. 8. The system retrieves the user's current GPS location. 9. The system cross-checks available bins within a defined radius (e.g., 2 km). 10. The system filters bins based on: <ul style="list-style-type: none"> - Accepted waste type. - Estimated available capacity (Low, Medium, High). 11. The system generates a ranked list of suitable bins, prioritizing the nearest bins with sufficient capacity. 12. The system highlights the top recommended bin as the "Best Option." 13. The user is presented with a map view showing: <ul style="list-style-type: none"> - User's current location (red marker). - Recommended bins (green/yellow/red markers depending on occupancy). 14. The user can toggle between map view and list view of bins. 15. Each bin result displays: <ul style="list-style-type: none"> - Location address. - Distance from user. - Occupancy level. - Accepted waste type(s). 16. The user clicks on a specific bin in the list or map. 17. The system shows detailed bin information, including the last update timestamp.

	<p>18. 6.3 The user can click “Navigate”, which redirects to Google Maps for directions.</p> <p>19. The user can choose to save the bin as a preferred location for future use.</p> <p>20. The system confirms that the bin has been saved (if chosen).</p> <p>21. The user can return to the dashboard or log out.</p>
Alternative Flows:	<p>If the user does not grant precision to access his/her location, the system will display a message “The app requires location permission to work”. The user can either:</p> <ol style="list-style-type: none"> 1. Grant permission → return to the main flow. 2. Cancel → system returns to the home screen. <p>If no bins meet the user’s criteria (waste type + capacity + proximity): The system displays: “No suitable bins are currently available for your selection.”</p> <p>The user may:</p> <ol style="list-style-type: none"> 1. Retry Later → The system suggests checking again after some time. 2. View Closest Available Bins → The system presents the nearest bins regardless of capacity or waste type, so the user still has disposal options.
Exceptions:	If the map cannot be loaded properly, or data is not stored in the backend server properly, an error message will display to indicate that the resource requested is unavailable.
Includes:	Login (User): The user must already be logged in to the portal to access this functionality.
Special Requirements:	NIL
Assumptions:	There are e-waste bins nearby (within 2km).
Notes and Issues:	NIL

11. Navigate to Bin

Use Case ID:	UC11		
Use Case Name:	Navigate to Bin		
Created By:	Myat Noe	Last Updated By:	Nicholas
Date Created:	24 Aug 2025	Date Last Updated:	31 Aug 2025

Actor:	User
Description:	The user wants to get directions to the nearest e-waste bin.
Preconditions:	<ul style="list-style-type: none"> • The user has access to the application.

	<ul style="list-style-type: none"> • The user is logged in. • Global Positioning System (GPS) is enabled on the user's device. • The system has the latest information on the bins' location. • The device has internet connectivity for map navigation.
Postconditions:	<ul style="list-style-type: none"> • The user can open Google Maps for navigation.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects a bin from the list/map. 2. The user clicks the "Navigate" button. 3. The system confirms the user's location is accessible. 4. Google Maps opens with the route to the selected bin. 5. The user follows the route to the bin.
Alternative Flows:	<p>If the user has no internet connection, the system displays: "Internet connection is required to open maps." The user can retry after connecting to the internet.</p> <p>If the user's location is unavailable, the system displays: "Unable to detect current location. Please enable GPS and try again." The user can enable GPS and then return to the main flow.</p> <p>If the user chooses a bin outside Singapore, the system displays: "Selected bin is outside the service area. Navigation unavailable." The user can choose another bin.</p>
Exceptions:	<p>If the GPS signal is lost during navigation, Google Maps will show a warning and attempt to recalculate.</p> <p>If the selected bin is removed or marked unavailable, the system displays "Bin is currently unavailable. Please select another bin."</p>
Includes:	<p>Locate Nearby Bins: The navigation feature depends on the user first locating a nearby bin.</p> <p>Check Bin Availability: Shows bin capacity status before navigating.</p>
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • The users have a GPS-enabled device. • The users are familiar with basic map navigation. • Bins' location data in the database is accurate.
Notes and Issues:	NIL

12. Suspend User

Use Case ID:	UC12		
Use Case Name:	Suspend User		
Created By:	Nicholas	Last Updated By:	Nicholas
Date Created:	28 Aug 2025	Date Last Updated:	31 Aug 2025

Actor:	Admin
Description:	The admins will be able to suspend a user should a normal user perform an action that is undesirable (eg. spamming the query functionality)
Preconditions:	<ul style="list-style-type: none"> The admin has logged into the system and has internet connectivity. The user to be suspended has already performed an action that is deemed to be worthy of being flagged by an admin
Postconditions:	<ul style="list-style-type: none"> The user's account is suspended. The user will no longer be able to log in with his account.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> The admin selects log in and enters personal credentials to log in. The admin clicks on "Manage Users" menu functionality The admin scrolls/searches to locate the specific user to be suspended. The system displays the user's profile details (username, email, recent activity logs, current account status). The admin clicks the "Suspend User" option. The system displays a confirmation dialog: "Are you sure you want to suspend this user? This action will prevent them from logging in." The admin confirms the action by clicking "Yes". The system updates the user's account status to "Suspended" in the database. The system logs the action in the Admin Activity Log (admin ID, timestamp, user suspended, reason if provided). The system displays a success message: "User has been suspended successfully." The flagged user is immediately prevented from logging in on their next attempt. The admin remains in the Manage Users menu and can continue to manage other accounts.
Alternative Flows:	NIL
Exceptions:	The same exceptions for Login (Admin) apply here.
Includes:	Login (Admin): The admin must be verified in order to perform such

	functionality. Normal users will not have any access to this functionality
Special Requirements:	Storage of user flag status in the database.
Assumptions:	<ul style="list-style-type: none"> The user to be suspended already has an existing account in the system The admin has a valid internet connection and is able to successfully log into the system
Notes and Issues:	NIL

13. Create Query

Use Case ID:	UC13		
Use Case Name:	Create Query		
Created By:	Cek Cong	Last Updated By:	Myat Noe
Date Created:	29 Aug 2025	Date Last Updated:	31 Aug 2025

Actor:	User
Description:	The user has some query or inquiry to make concerning any of the application services. The user may also have an inquiry on more general issues relating to e-waste logistics etc.
Preconditions:	<ul style="list-style-type: none"> User has to be logged in into the portal Device has internet access
Postconditions:	<ul style="list-style-type: none"> The user is able to successfully submit a query to an admin, who will respond to the user's query accordingly.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> The user opens the application portal. The user selects "Log In." The system prompts the user for credentials (username and password). The user enters credentials. The system validates the credentials against the user database. Upon successful authentication, the user is redirected to the User Dashboard. The user selects the "Query / Support" menu option. The system displays the Query Management Page, with options such as Create New Query, View Submitted Queries, and Query Status.

	<p>9. The user selects “Create New Query.”</p> <p>10. The system displays a form with required fields (e.g., Subject, Category, Description, and optional attachments).</p> <p>11. The user enters the query details into the fields provided.</p> <p>12. If needed, the user attaches supporting files or screenshots (optional).</p> <p>13. The system validates mandatory fields (e.g., Subject and Description cannot be empty).</p> <p>14. The user clicks “Submit.”</p> <p>15. The system displays a confirmation screen summarizing the entered query details (Subject, Category, Description, Attachments).</p> <p>16. The user reviews the information.</p> <p>17. The user clicks “Confirm” to proceed.</p> <p>18. The system saves the query into the database, tagged with:</p> <ul style="list-style-type: none"> - User ID - Query details - Date and timestamp - Status = “Pending” <p>19. The system logs the submission in the User Activity Log.</p> <p>20. The system displays a success message: “Your query has been submitted successfully. An admin will respond shortly.”</p> <p>21. The system sends an acknowledgment notification (e.g., in-app or email) to the user with a reference ID for tracking.</p> <p>22. The user remains on the Query Management Page and can choose to create another query or return to the dashboard.</p>
Alternative Flows:	<ol style="list-style-type: none"> 1. If the user enters a query that is too short (20 characters or less), the system will show a message “Your query is too short.”. This is to prevent users from spamming the submit query functionality. 2. The actor will input a new valid query that fulfills the requirements.
Exceptions:	The user inputs a query that is too short, in which case the user will be invited to input a longer query.
Includes:	Login (user): The user has to be logged in and have a valid account in order to access this functionality.
Special Requirements:	A query downtime has to be implemented, with users limited to submitting a fixed number of queries during a certain timeframe. This is to prevent spam from the end of the users.
Assumptions:	User has a valid account already registered in the system
Notes and Issues:	NIL

14. Answer Query

Use Case ID:	UC14
Use Case Name:	Answer Query

Created By:	Nicholas	Last Updated By:	Myat Noe
Date Created:	28 Aug 2025	Date Last Updated:	31 Aug 2025

Actor:	Admin
Description:	The admin will respond to queries created by various users.
Preconditions:	<ul style="list-style-type: none"> • Admin has to be logged in into the portal • Device has internet access • There exists at least 1 unanswered query in the database from a user
Postconditions:	<ul style="list-style-type: none"> • User's query is successfully answered • Query from the user can be marked as closed
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The Admin opens the application portal. 2. The Admin selects "Log In." 3. The system prompts for admin credentials (username + password). 4. The Admin enters credentials. 5. The system validates the credentials against the Admin Database. 6. Upon successful authentication, the Admin is redirected to the Admin Dashboard. 7. The Admin selects the "Query Management" option from the menu. 8. The system displays a list of queries with details such as: <ul style="list-style-type: none"> - Query ID - User ID / Name - Subject / Category - Submission Timestamp - Status (Open, Pending, Closed) 9. The Admin filters or searches queries (optional). 10. The Admin selects a specific query marked as "Open." 11. The system displays the full query details, including: <ul style="list-style-type: none"> - User's original message - Attachments (if any) - Timestamp of submission - Query status = "Open" 12. The Admin clicks "Respond to Query." 13. The system presents a text input field (Answer Box) and optional attachment upload. 14. The Admin enters the response/answer to the query. 15. The system validates the input (e.g., ensures response is not empty). 16. The Admin clicks "Submit." 17. The system prompts for confirmation: "Are you sure you want to submit this response? Once submitted, no other admin can

	<p>respond.”</p> <p>18. The Admin clicks “Confirm.”</p> <p>19. The system saves the Admin’s response into the Query Database with:</p> <ul style="list-style-type: none"> - Answer text - Admin ID - Timestamp - Status updated to “Closed” <p>20. The system logs the action in the Admin Activity Log.</p> <p>21. The system sends a notification (in-app or email) to the User:</p> <p>22. “Your query (Ref: #QueryID) has been answered by an Admin.”</p> <p>23. The Admin sees a success message: “Response submitted successfully. This query is now closed.”</p> <p>24. The Admin is redirected back to the query list, where the answered query now appears as Closed.</p>
Alternative Flows:	<p>1. If the admin notices that the query submitted is spam or dubious in nature, the flag user use case will be executed.</p>
Exceptions:	<p>1. Changes to the answer will not be saved. The admin must submit the query for the answer to be saved. This is to prevent multiple admins from editing a query.</p>
Includes:	<p>Login (Admin): The Admin has to be logged in and have a valid account in order to access this functionality.</p>
Special Requirements:	<p>Admins can respond to any query. However, once a query is responded to, other admins cannot respond to the query anymore. It will be regarded as closed in the system</p>
Assumptions:	<ul style="list-style-type: none"> Admin has a valid account already registered in the system Admin has a valid internet connection
Notes and Issues:	<p>Need to track the status of queries in the system (whether they are open or closed)</p>

15. Create Reminder

Use Case ID:	UC15		
Use Case Name:	Create Reminder		
Created By:	Cek Cong	Last Updated By:	Myat Noe
Date Created:	28 Aug 2025	Date Last Updated:	31 Aug 2025

Actor:	User	
Description:	The user should be able to create a reminder with a custom memo that reminds the user to dispose of his/her E-waste.	

Preconditions:	<ul style="list-style-type: none"> The user has to be logged in into the portal. The user's device has internet access.
Postconditions:	<ul style="list-style-type: none"> User's reminder will appear on the "reminders" section of the user dashboard on the web application.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> The user opens the application portal. The user selects "Log In." The system prompts the user for credentials (username and password). The user enters credentials. The system validates the credentials against the user database. Upon successful authentication, the user is redirected to the User Dashboard. The user selects the "Reminder" menu option. The system displays the Reminder Management Page, with options such as Create New Reminder, View Reminders The user selects "Create New Reminder" The system displays a form with required fields (e.g. Location, Description, Date, Time, and optional attachments). The user enters the reminder details into the fields provided. If needed, the user attaches photos (optional). The system validates mandatory fields (e.g., Location, Description, Date and Time cannot be empty). The input for Date and Time should be later than the date and time in which the user created the reminder. The user clicks "Submit." The system displays a confirmation screen summarizing the entered reminder details. The user reviews the information. The user clicks "Confirm" to proceed. The system saves the reminder into the database, tagged with: <ul style="list-style-type: none"> - User ID - Reminder ID - Date and Time - Status = "Active" (to signal the reminder is still ongoing) The system logs the submission in the User Activity Log. The system displays a success message: "Your reminder has been created successfully. A notification will be sent to you to remind you." The user can log out of the system. The system sends a notification (e.g., in-app or email) to the user when the date and time has come.
Alternative Flows:	If the user submitted a reminder with a date/time that has already passed, the system will not accept the input and will display the error message: "Please input a valid date and time."

Exceptions:	NIL
Includes:	Login (User): The User has to be logged in and have a valid account in order to access this functionality.
Special Requirements:	The user has to enable notification (etc on the app or email) in order to receive the notification on time.
Assumptions:	<ul style="list-style-type: none"> • The user has a valid account already registered in the system • The user has a valid internet connection
Notes and Issues:	The notification can either be an app notification or an email notification.

16. Modify User Account Details

Use Case ID:	UC16		
Use Case Name:	Modify User Account Details		
Created By:	Cek Cong	Last Updated By:	Cek Cong
Date Created:	11 Sep 2025	Date Last Updated:	11 Sep 2025

Actor:	Admin
Description:	Admin may receive requests from users to modify their account information on their behalf. The information includes username and password.
Preconditions:	<ul style="list-style-type: none"> • The user has to be logged in into the portal. • The user's device has internet access.
Postconditions:	<ul style="list-style-type: none"> • Information of the user account gets updated in the database.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The admin logs in successfully. 2. The admin clicks on “Manage Users” menu functionality 3. The admin scrolls/searches to locate the specific user to be suspended. 4. The system displays the user’s profile details (username, email, recent activity logs, current account status). 5. The admin enters the new username/password/confirmation password/ 6. The admin clicks save. 7. The system performs validation checks: <ul style="list-style-type: none"> - Ensures the new username is between 5–30 characters in length. - Ensures the username is not already taken by another

	<p>registered account.</p> <ul style="list-style-type: none"> - Ensures the username does not contain invalid or restricted characters (e.g., spaces). <p>OR</p> <ul style="list-style-type: none"> - New password and confirmation match. - The new password meets security criteria: length of 8–30 characters, contains at least one number, and at least one special character. <ol style="list-style-type: none"> 8. If all validations pass, the system updates the username in the database. 9. The system displays a success message “You have successfully updated the user’s information”. 10. The system logs the action in the Admin Activity Log. 11. The user can log out of the system.
Alternative Flows:	<p>If the username/password does not meet the requirement, the system displays an error message “Username/Password does not meet the</p> <p>If the username is already taken, the system displays an error message “Username is taken. Please try another username.”</p>
Exceptions:	NIL
Includes:	Login (Admin): The admin has to be logged in and have a valid account in order to access this functionality.
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • The admin has a valid account already registered in the system • The admin has a valid internet connection
Notes and Issues:	NIL

17. Update Bin Occupancy Status

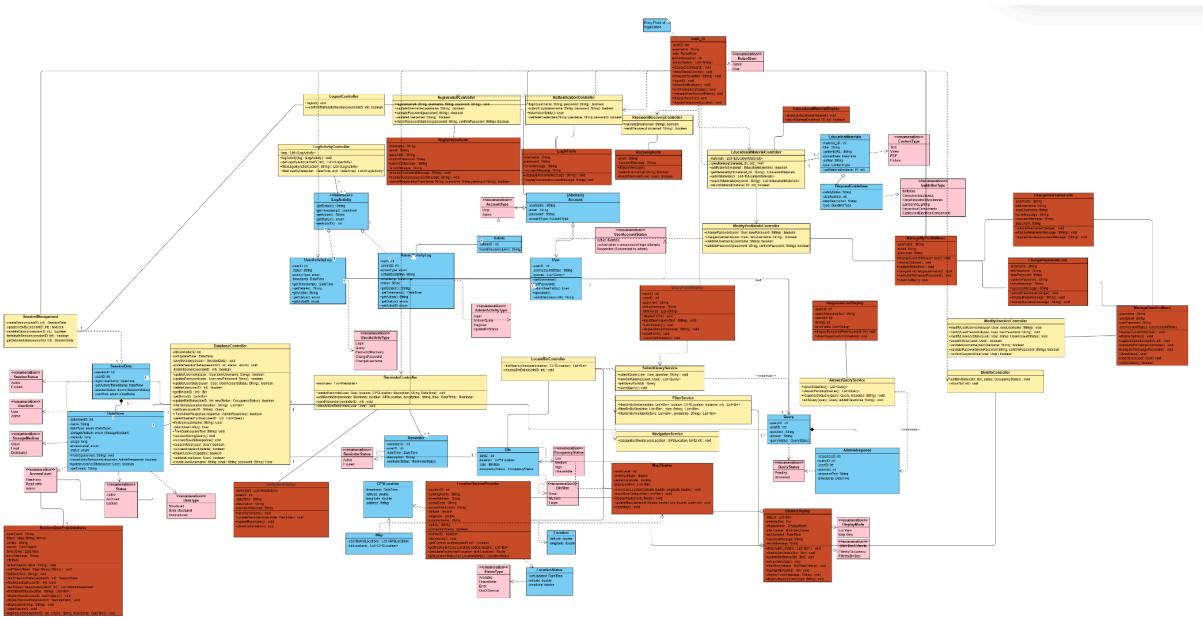
Use Case ID:	UC17		
Use Case Name:	Update Bin Occupancy Status		
Created By:	Cek Cong	Last Updated By:	Cek Cong
Date Created:	11 Sep 2025	Date Last Updated:	11 Sep 2025

Actor:	Admin
Description:	The admin may wish to update the occupancy status of certain bins.
Preconditions:	<ul style="list-style-type: none"> • The admin has to be logged in into the portal. • The admin device has internet access.

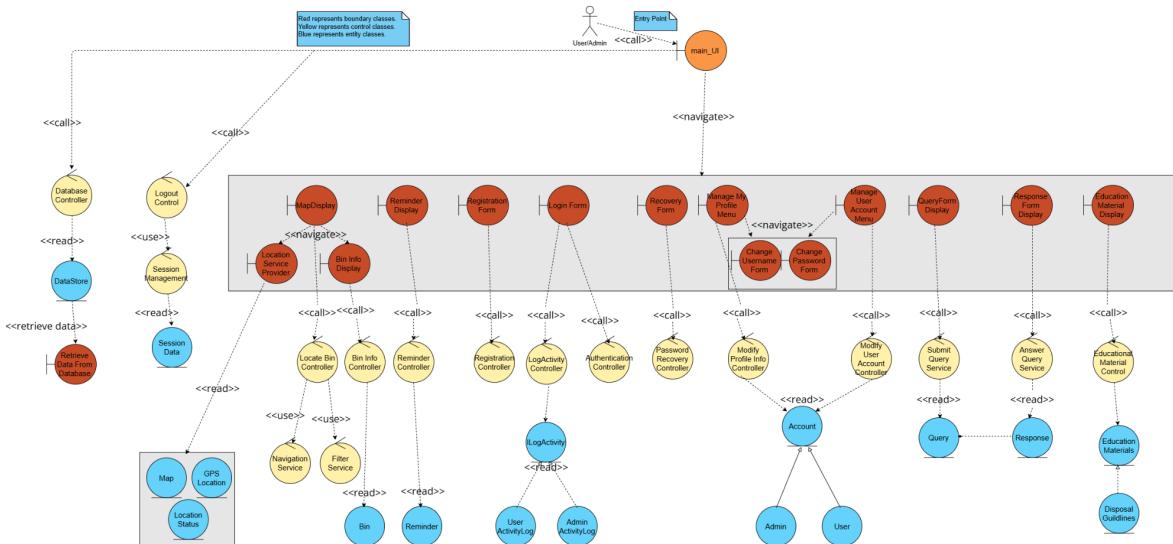
Postconditions:	<ul style="list-style-type: none"> • Bins' information gets updated in the database.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The admin logs in successfully. 2. The admin clicks on the “Bin Information Display”. 3. The system will display a list of bins stored in the database. 4. The admin clicks onto a particular bin. 5. The admin clicks on “Update Bin Occupancy Status” and clicks onto the new occupancy status. 6. Admin clicks save and the system will display a success message “You have successfully updated the Bin's information”. 7. The system logs the action in the Admin Activity Log. 8. The user can log out of the system.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login (Admin): The admin has to be logged in and have a valid account in order to access this functionality.
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • The admin has a valid account already registered in the system • The admin has a valid internet connection
Notes and Issues:	NIL

III. Class diagram

Our class diagram is as follows: blue represents entity classes, yellow represents control classes, the maroon represents boundary classes, and pink represents enumeration classes.

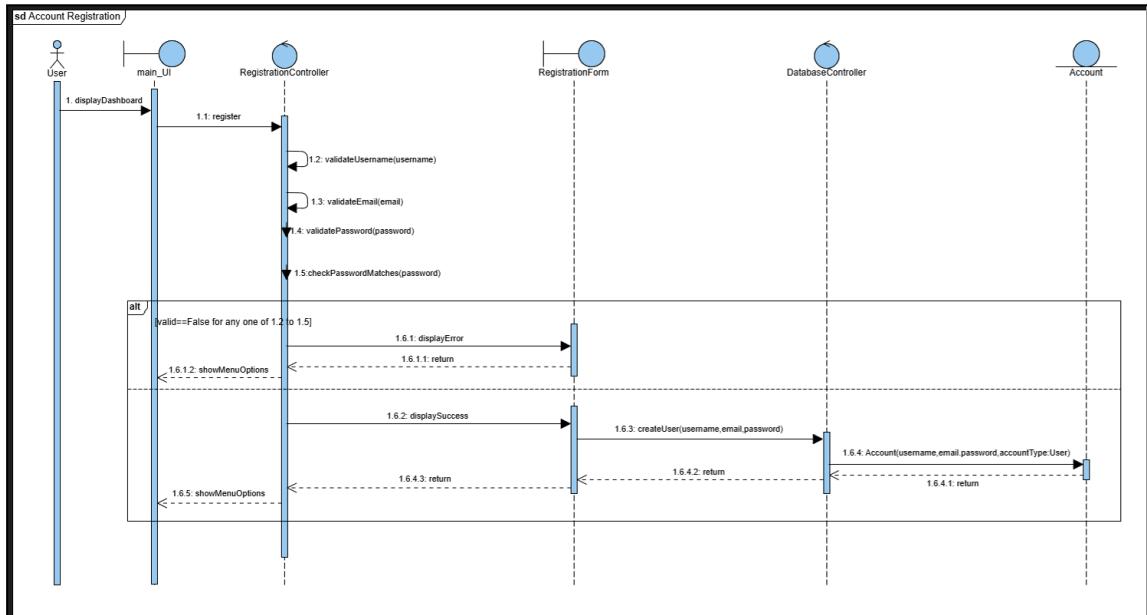


IV. Class Stereotype

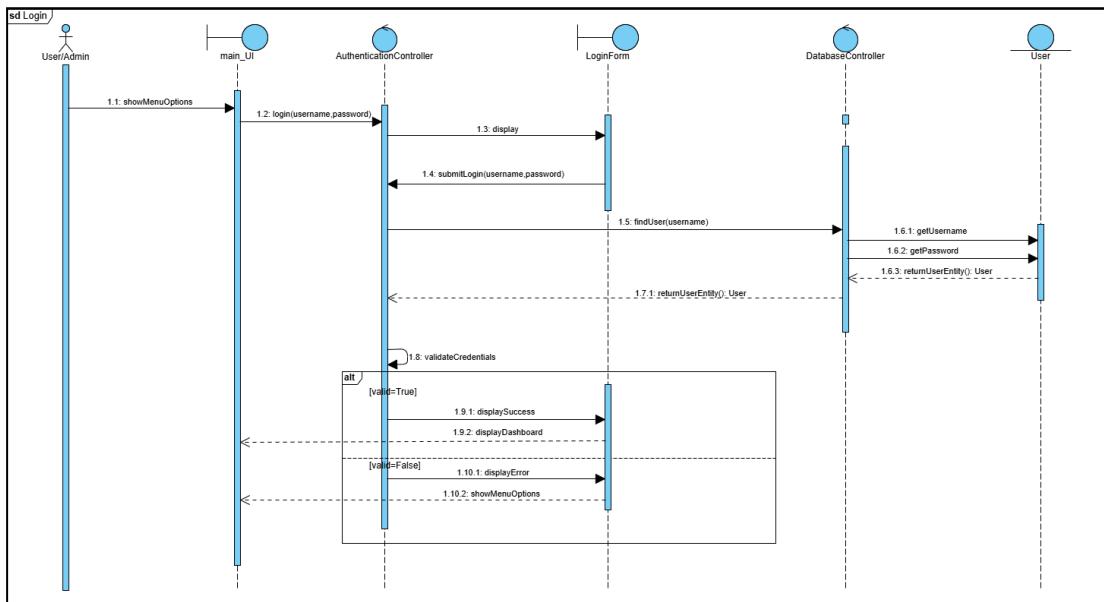


V. Sequence Diagrams

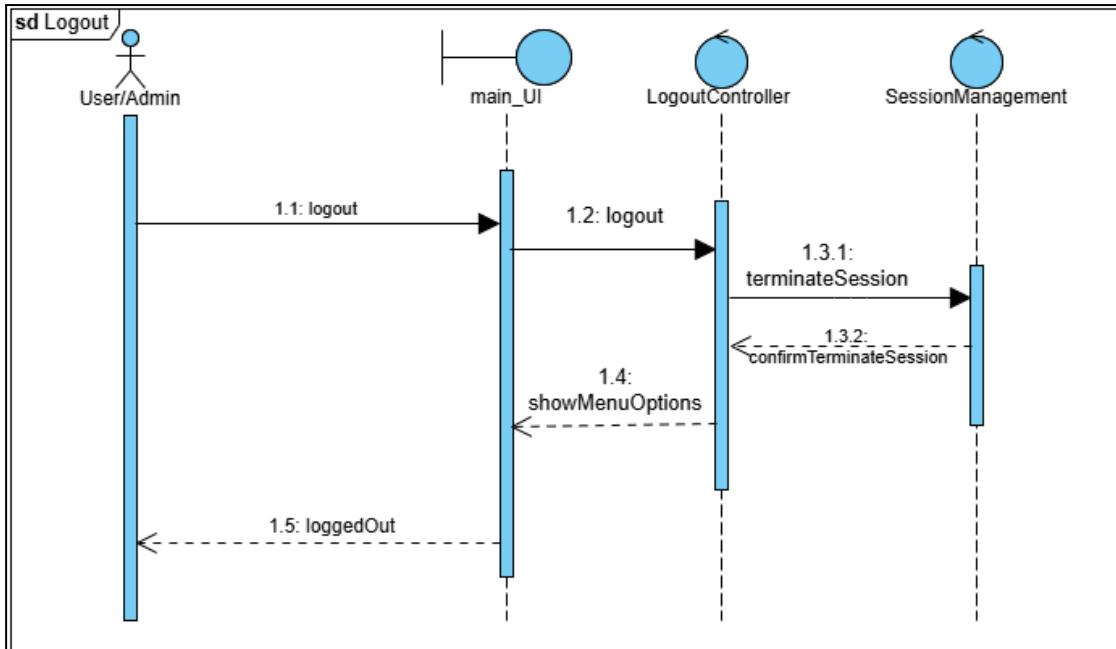
1. Account Registration



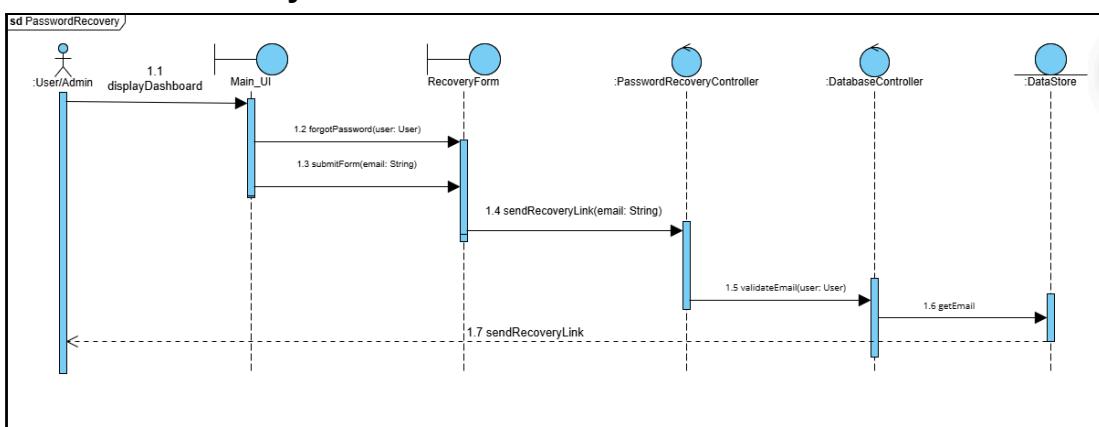
2. Login



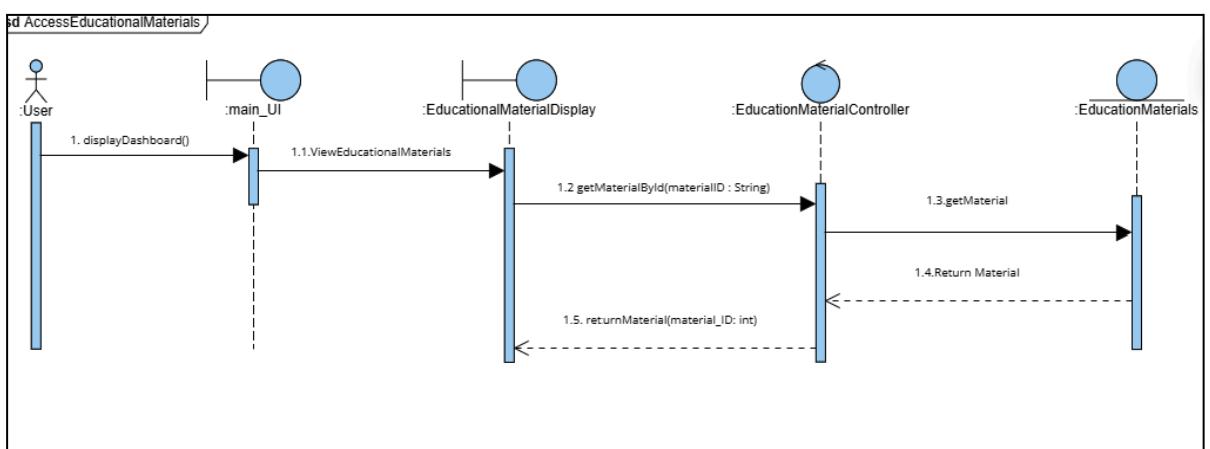
3. Logout



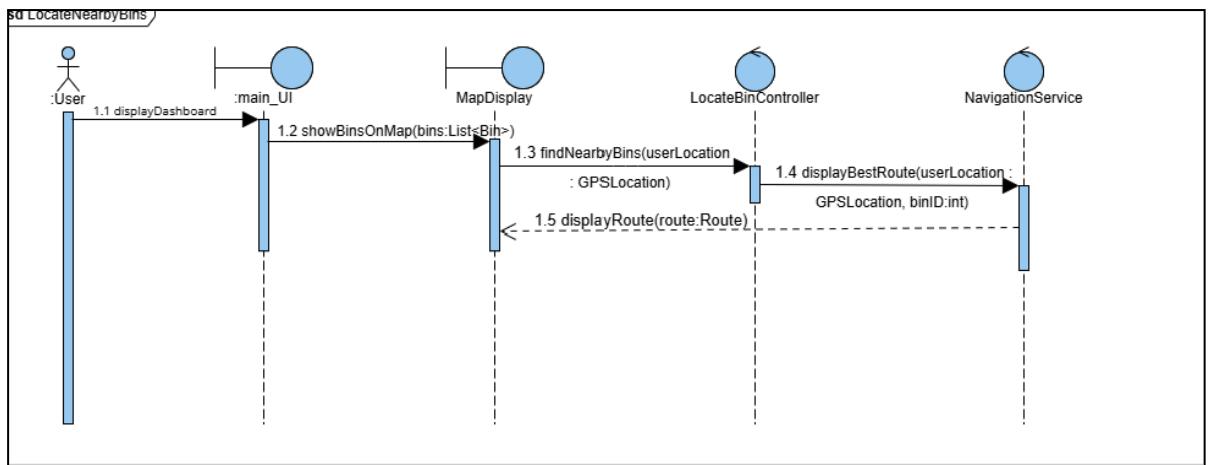
4. Password Recovery



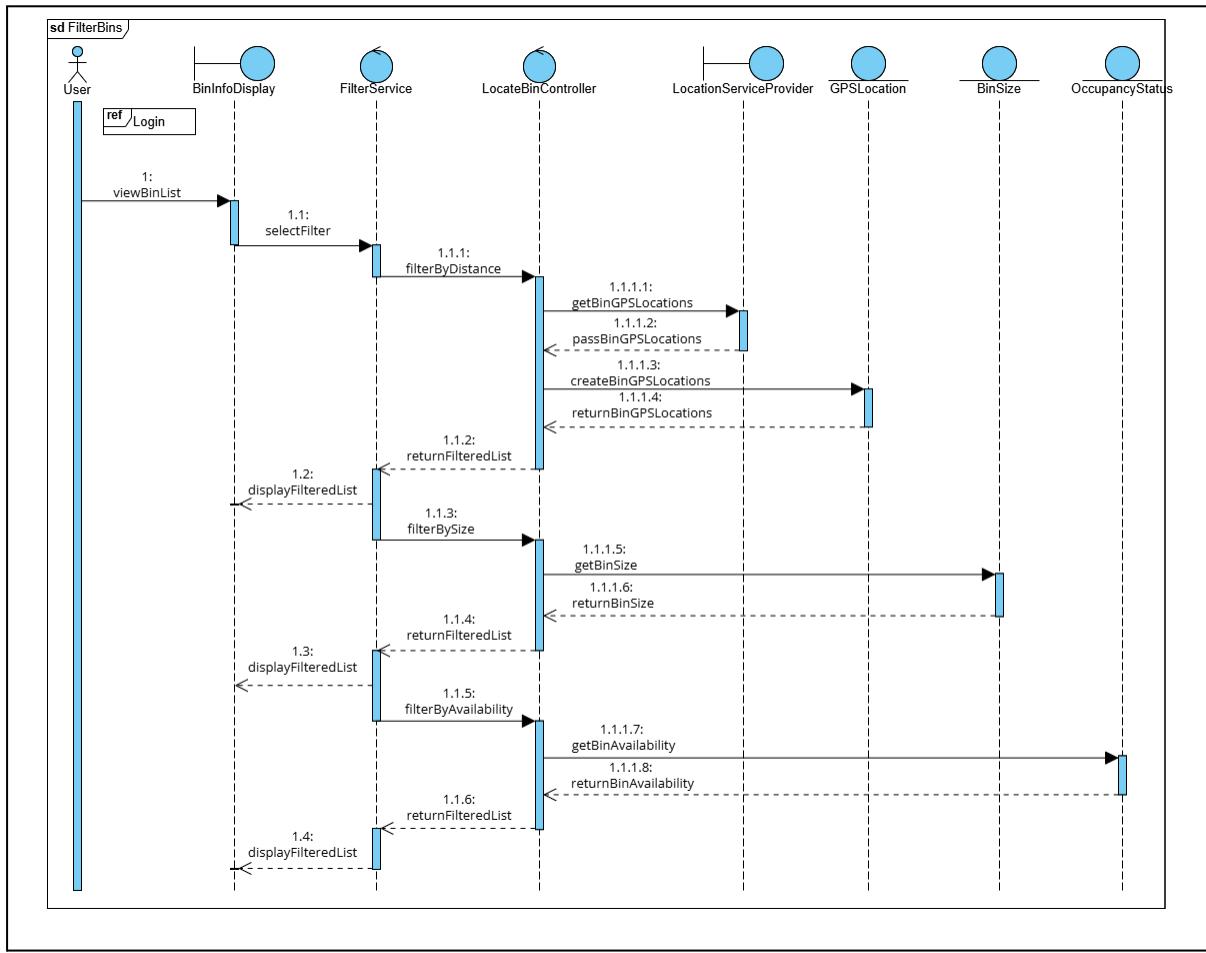
5. Access Educational Materials



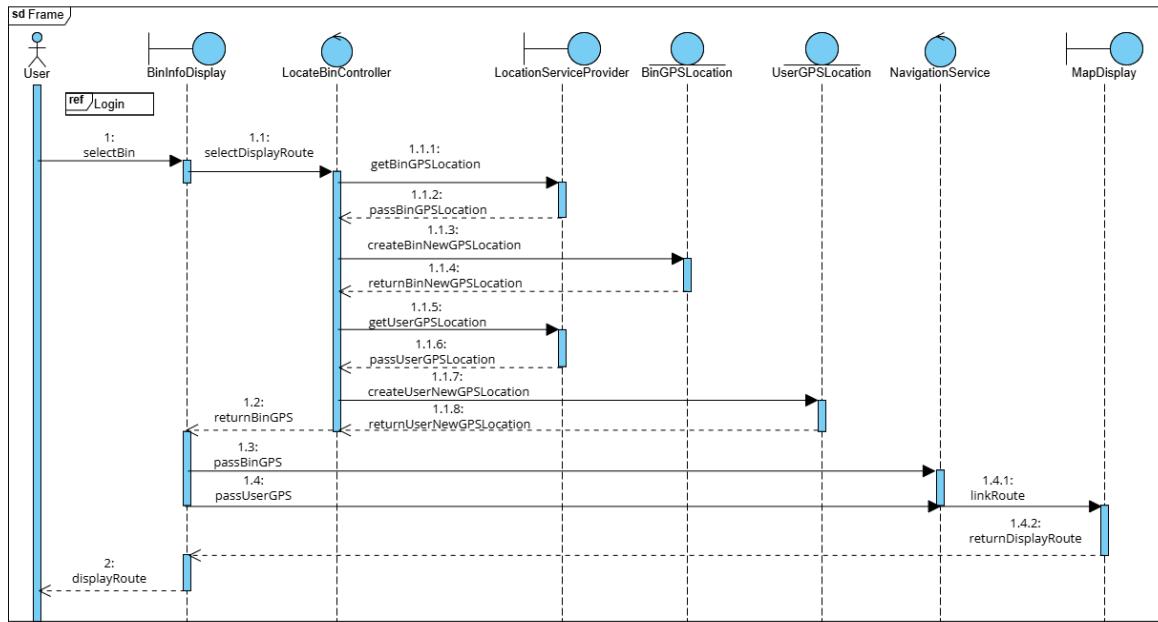
6. Locate Nearby Bins



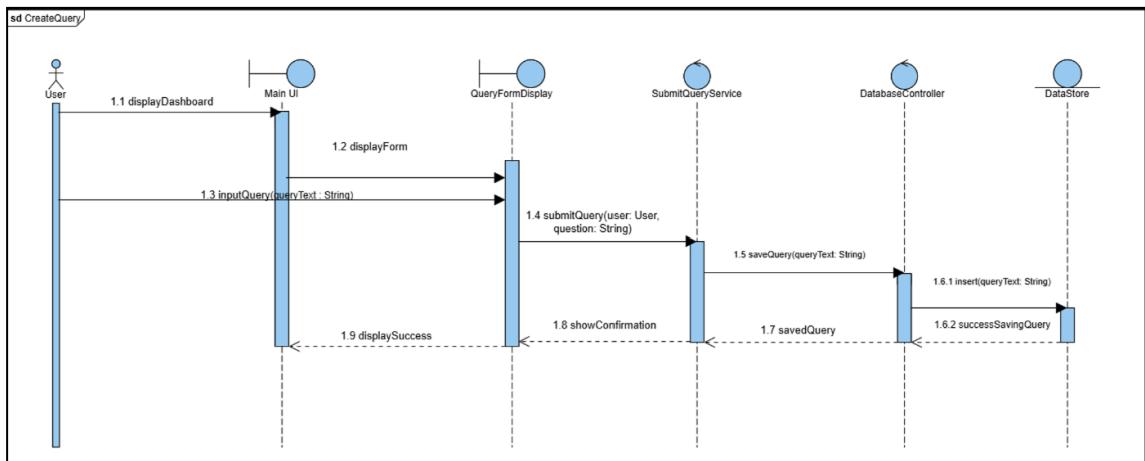
7. Filter Bins by Distance/Availability



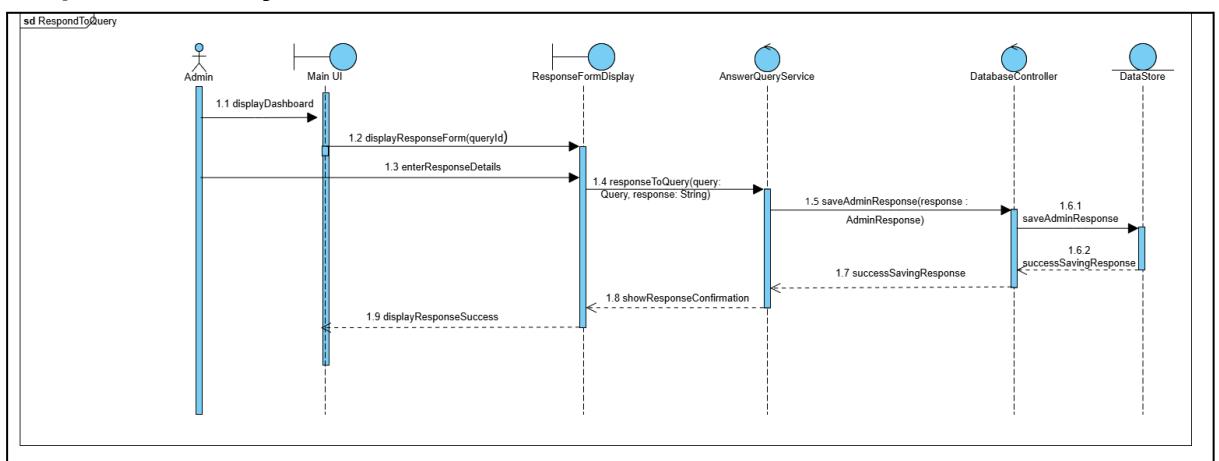
8. Display Route



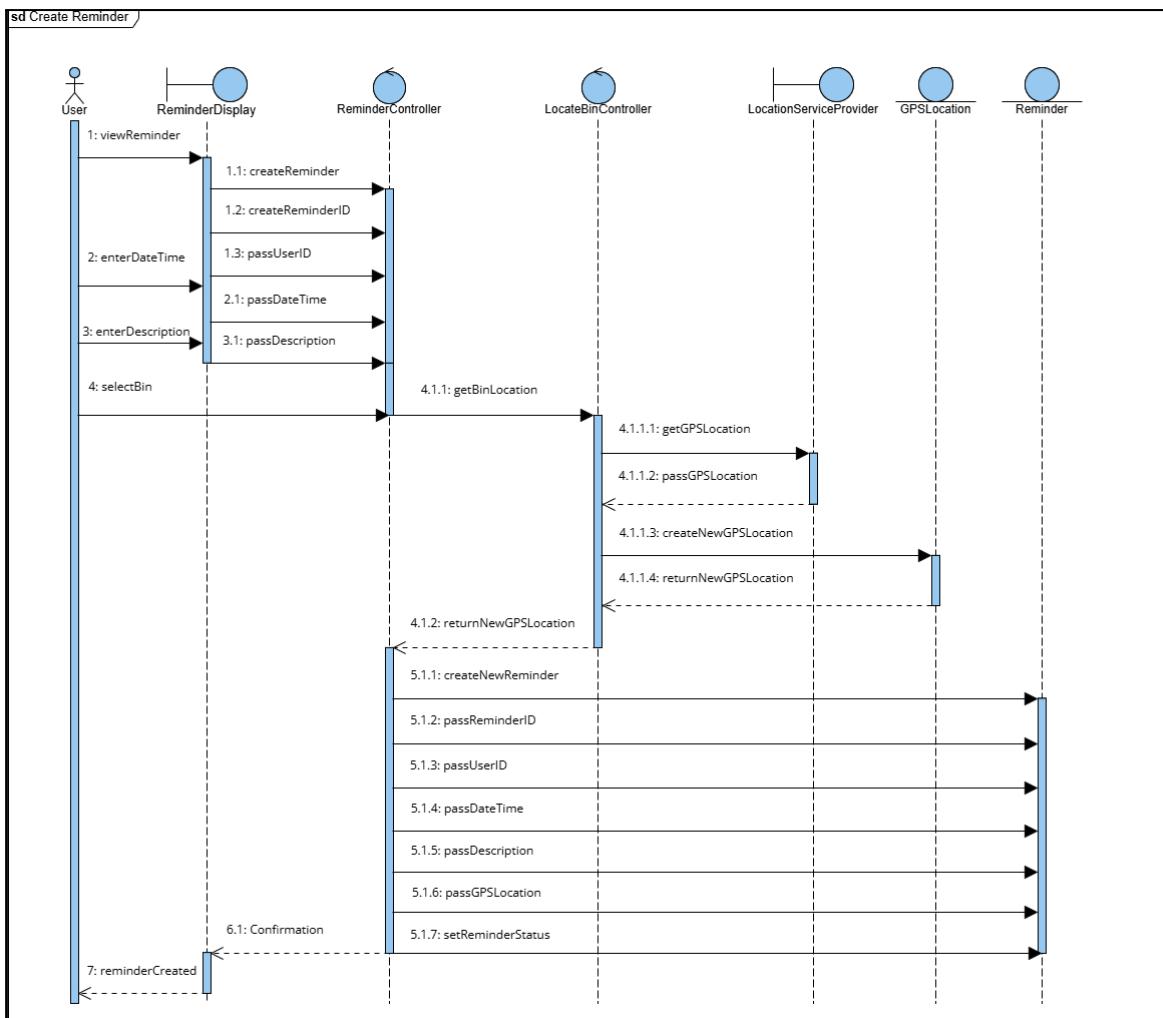
9. Create Query



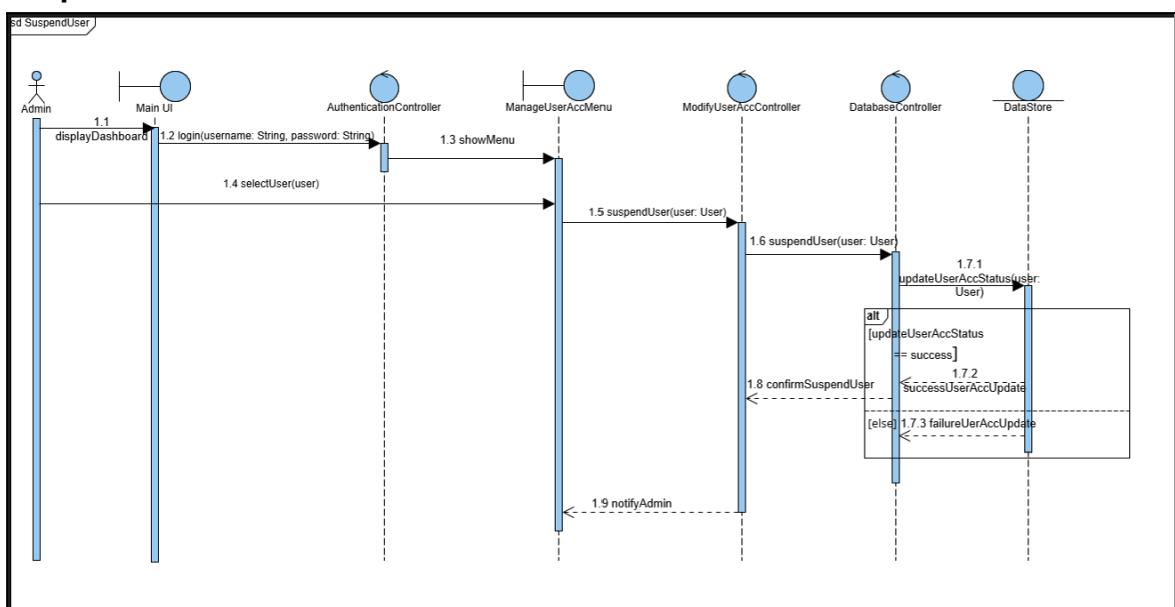
10. Respond to Query



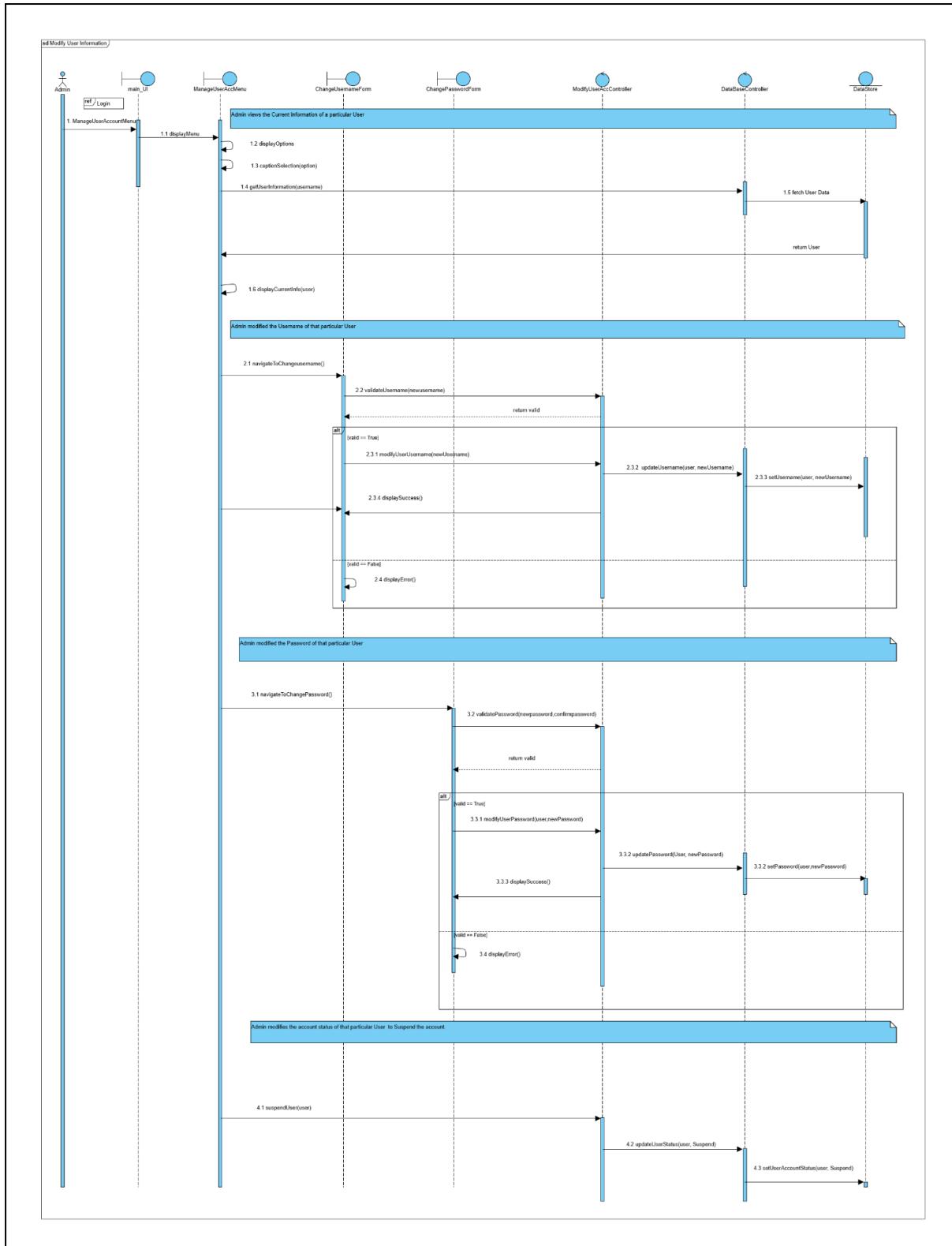
11. Create Reminder



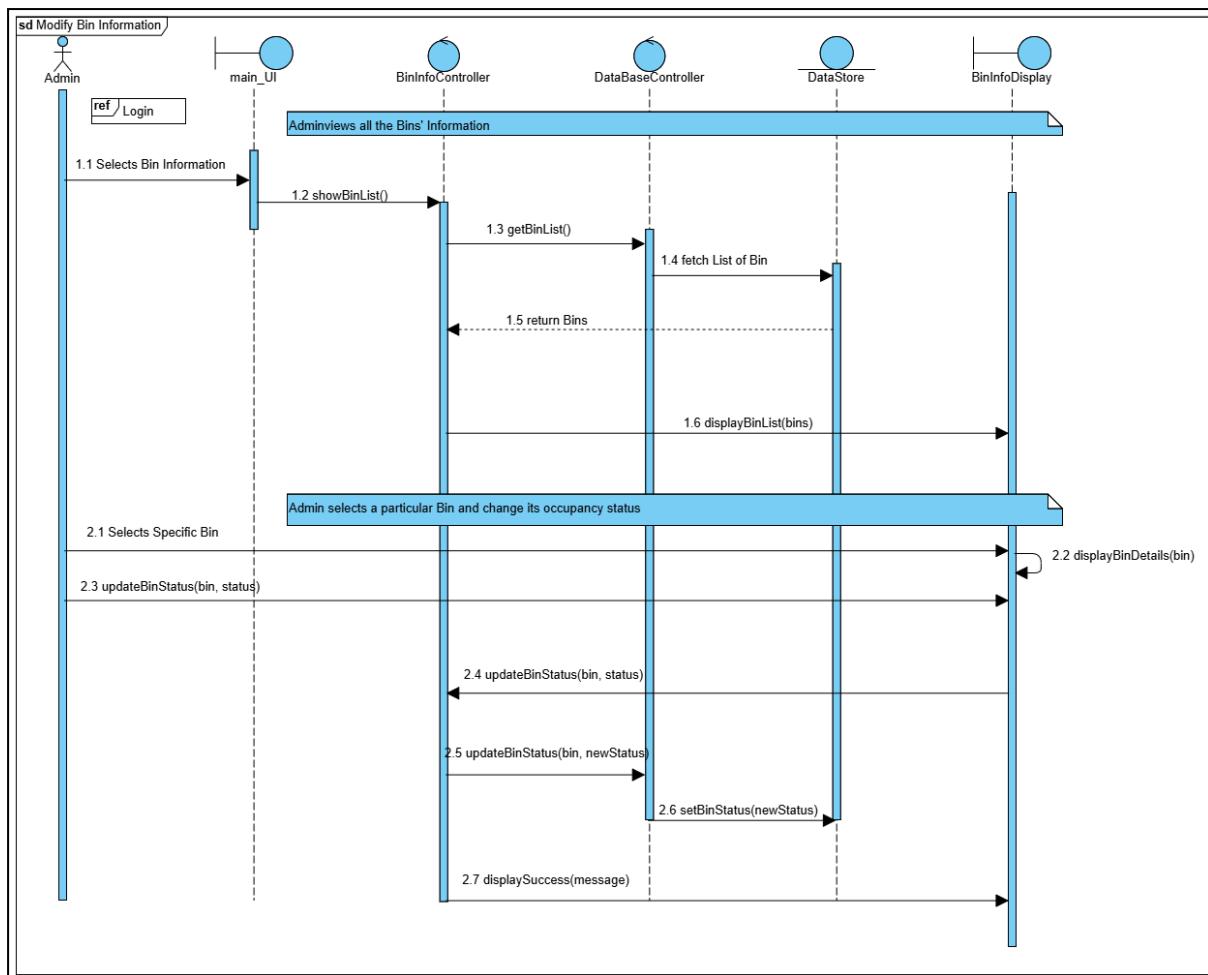
12. Suspend User



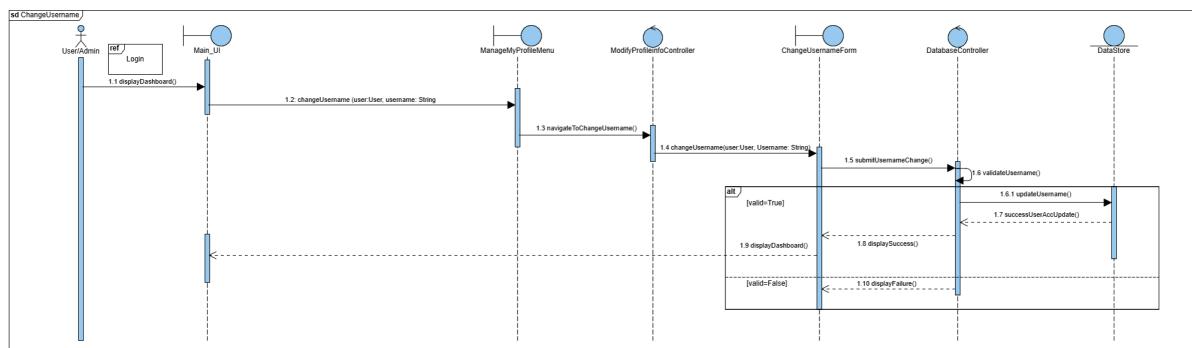
13. Modify User Information



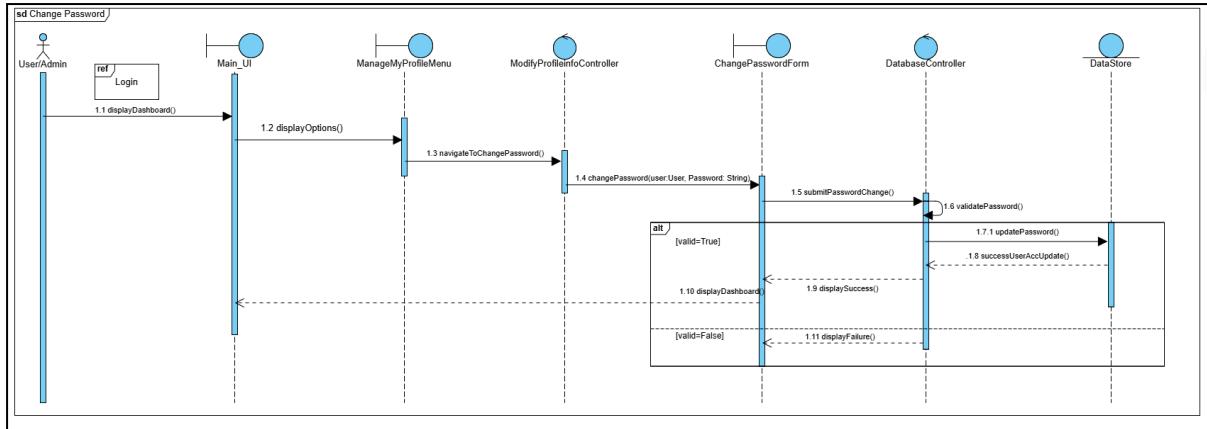
14. Modify Bin Information



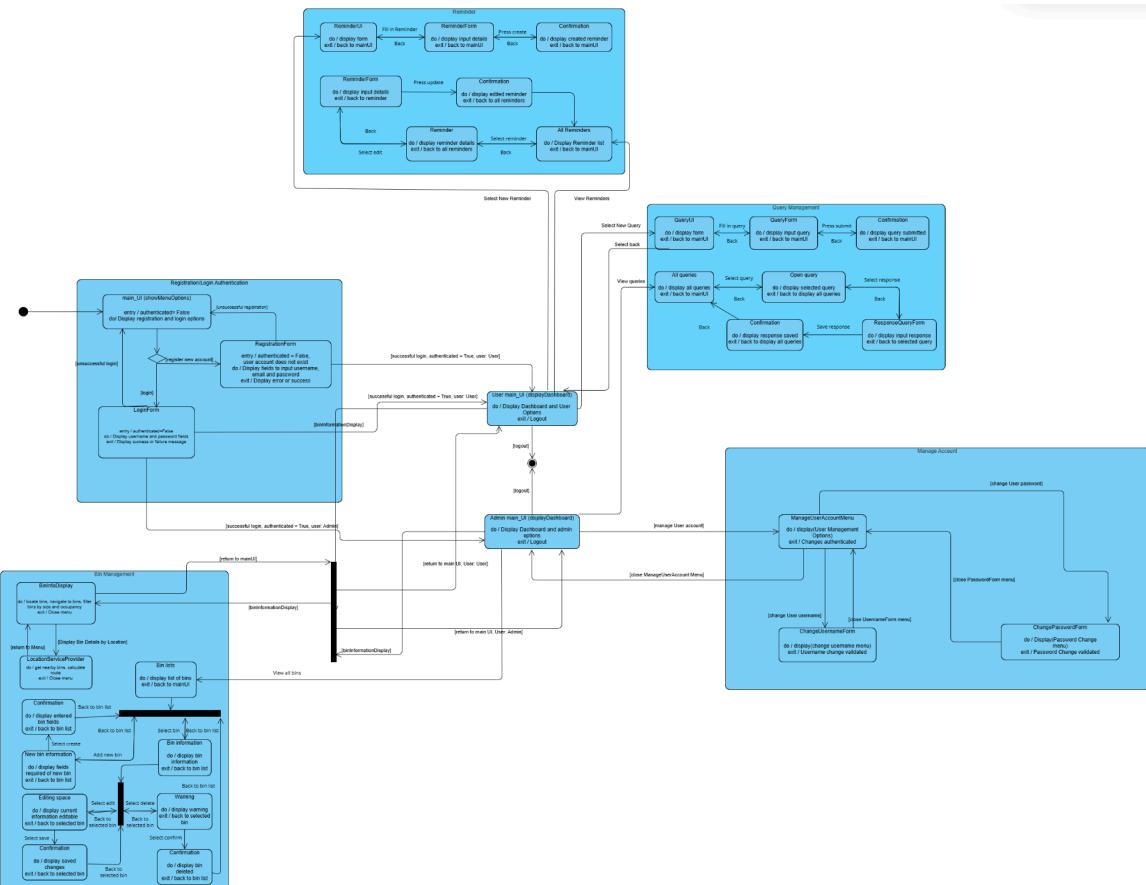
15. Change Username



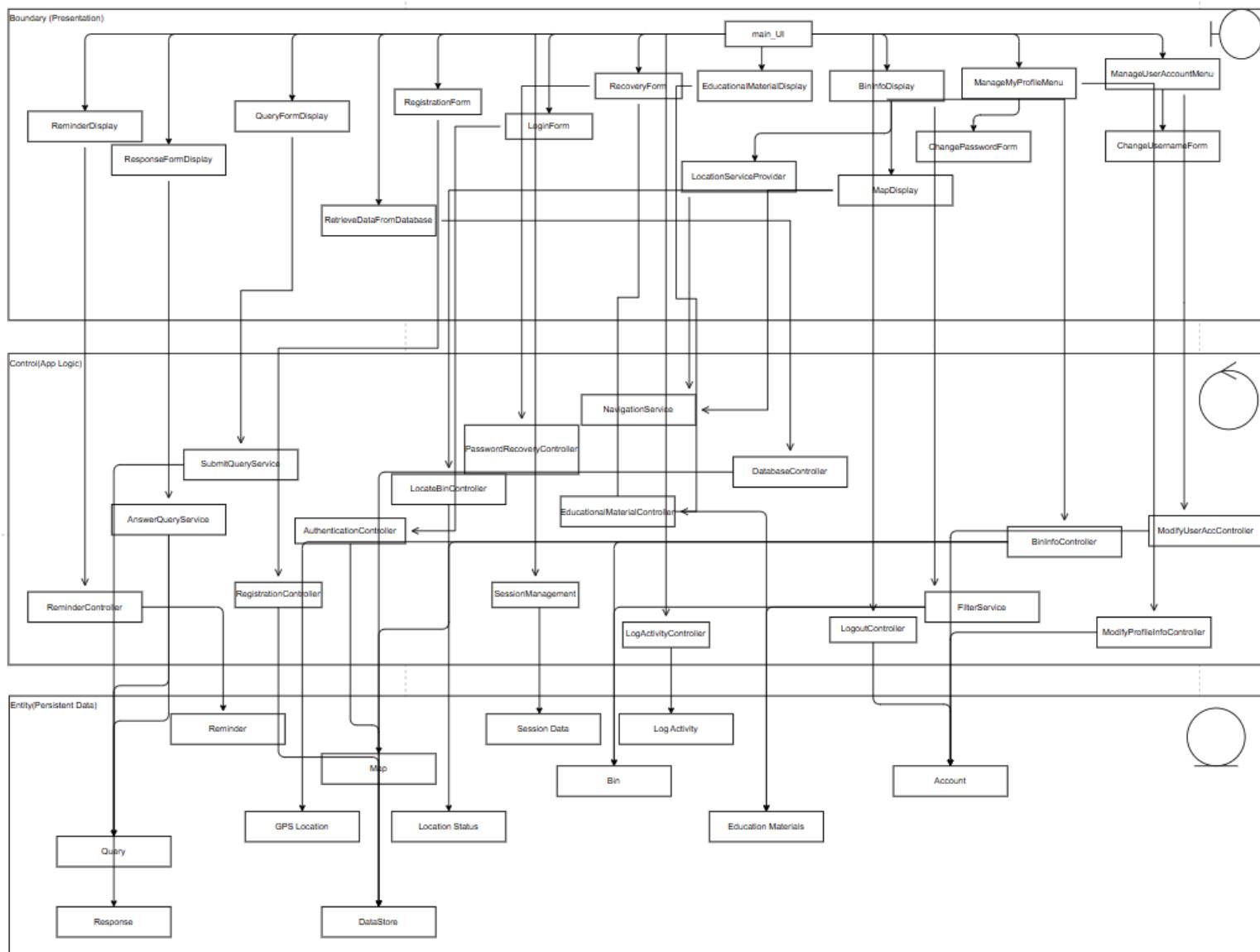
16. Change Password



VI. Initial Dialog Map



VII. System Architecture



Entity Layer

This layer is responsible for managing all persistent data within the Bin Buddy system. It defines the core data structures and entities that are stored and retrieved from the database. These entities encapsulate the system's long-term information such as user accounts, queries, reminders, logs, and educational content, and they are designed to maintain data integrity and support the logic implemented in the control layer.

The layer consists of the following key classes:

1. **Account:** Represents the user or admin account information stored in the system. It contains essential attributes such as username, password, and user role, which are referenced during login, registration, and profile management processes
2. **Query:** Stores user-submitted queries or questions regarding recycling or

other system-related topics. Each query entry includes details such as the query text, submission time, and user identifier

3. **Response:** Represents the admin's reply to a user's query. It is linked to the corresponding Query entity, allowing a one-to-one or one-to-many relationship
4. **Reminder:** Contains data related to user reminders for bin collection or other notifications. This entity stores the reminder message, date, and associated account information
5. **Map:** Acts as a central data structure that connects various location-related entities such as GPS Location, Location Status, and DataStore. It helps in retrieving and storing location-based data within the system.
6. **GPS Location:** Represents the geographical coordinates used by the system to identify bin locations and user positions.
7. **Location Status:** Stores dynamic data about the current state of a particular bin location
8. **DataStore:** Acts as a unified repository for managing and storing location and mapping data persistently. It serves as the main interface between location-related entities and the system database.
9. **Session Data:** Captures user session information for active users. This helps maintain login sessions, track activity, and improve user experience.
10. **Log Activity:** Maintains system logs of user and admin actions, enabling monitoring and auditing for system maintenance and security purposes.
11. **Bin:** Represents bin-related data such as type, location, and capacity. It is directly linked to the Map and other location entities for accurate representation and updates.
12. **Education Materials:** Stores educational content that users can access through the interface. It includes articles, guides, and other relevant materials related to recycling and environmental awareness.

Boundary (Presentation) Layer

This layer is responsible for the main interaction between users and Bin Buddy. Depending on which options the user selects, Bin Buddy will call the corresponding control classes responsible for the logic.

The layer consists of the following key classes:

1. **main_UI:** This is the main user landing page and is the centralised user interface for all user actions. It has dependencies with all other classes in the boundary layer
2. **ReminderDisplay:** This is part of the main UI. It allows users to display their personal reminders tied to their account that have been stored in the system.

It calls ReminderController and RetrieveDataFromDatabase to access the details.

3. **ResponseFormDisplay:** Allows admins to respond to queries by users. This gives them direct access to the response form to respond to appropriate queries. The class will call AnswerQueryService to execute the relevant logic for answering queries.
4. **QueryFormDisplay:** Allows users to ask questions regarding recycling or other related matters. This allows them to access the query form to ask questions. The class will call SubmitQueryService where the query logic is processed and the query stored in the database.
5. **RetrieveDataFromDatabase:** General UI class that enables users to retrieve information from the database. Privacy is ensured through the calling of DatabaseController, that allows the user to only retrieve information for its intended purpose.
6. **RegistrationForm:** Allows a new user to register for an account in the system. It works with RegistrationController to enable a new user to create an account.
7. **LoginForm:** Allows an existing user/admin to log in to their account stored in the system. LoginForm has a dependency with AuthenticationController to check if the credentials given by the user/admin are valid.
8. **RecoveryForm:** Allows an existing user to recover their password should it have been forgotten. It will call PasswordRecoveryController to allow the necessary recovery logic to be executed.
9. **EducationalMaterialDisplay:** Allows a user to display educational material after clicking the corresponding article or page in the application. The educational material is embedded in the frontend, thus no control classes are called.
10. **LocationServiceProvider:** Called by BinInfoDisplay to show the accurate locations of bins and other relevant location information.
11. **BinInfoDisplay:** Displays the information of sorted bins based on the filters implemented by the user (distance from current location etc.). This calls both MapDisplay and LocationServiceProvider to support the bin display.
12. **MapDisplay:** Displays the map of bins and locations.
13. **ManageMyProfileMenu:** Allows the user/admin to manage their own profile. This includes changes to passwords, usernames or other settings. It calls both the DatabaseController and the ModifyProfileInfoController to execute the profile management logic, as well as ChangePasswordForm and ChangeUsernameForm where required.
14. **ChangePasswordForm:** Allows the user/admin to change their password.
15. **ManagerUserAccountMenu:** Allows the admin to manage the various user accounts. The admin is able to change certain user information directly using their privileges. This calls the ModifyUserAccController and the DatabaseController to modify information where required.
16. **ChangeUsernameForm:** Allows the user/admin to change their username

Control Layer (App Logic)

1. **ReminderController:** The controller that handles the application logic when users set reminders for themselves to dispose of E-waste at a particular bin at a particular date and time.
2. **SessionManagement:** Handles the logic when a user logs into his/her

account, and a new session is created under that account, which will then be stored in the database for future reference, and that session will terminate when the user logs out of his/her account.

3. **AnswerQueryService:** The controller class that handles when an admin wishes to answer a particular query from a particular user.
4. **SubmitQueryService:** The controller that handles when a user submits a query regarding E-waste or E-waste bins through a query form.
5. **DatabaseController:** The central controller that handles the storage of data that consist of usernames, passwords, bin capacity information, user locations, reminders that user set etc.
6. **RegistrationController:** The controller class that handles the app logic when a user registers for an account.
7. **AuthenticationController:** The controller that handles the logic when a non-first-time user wants to log into his/her account, assuming they already have an account. This class will check that the username and password of that particular user match with the information in the database.
8. **PasswordRecoveryController:** Handles the logic when a user forgets their password. This class will send a recovery link to the email that the user provided when he/she first registered for his/her account.
9. **EducationalMaterialController:** Manages the educational materials like documents, images or videos regarding E-waste and E-waste bins on our website so that users can view these materials and learn about disposal guidelines, what E-waste is, or about why we should recycle E-waste.
10. **LocateBinController:** Handles the logic of showing the GPS location of the E-waste bins around Singapore using the dataset from data.gov.sg, as well as Google Maps API, and the nearest bins to a particular user, as well as the bins near the user when the user filters by the size capacity of the bin or when the user filters by distance.
11. **NavigationService:** Handles the logic of guiding a user to the nearest E-waste bin, or to the nearest E-waste bin that satisfies the user's size capacity and distance requirements using Google Maps API.
12. **BinInfoController:** Handles the logic of where the bin is located using Google Maps API and the dataset from data.gov.sg, and the size capacity of the bin.
13. **FilterService:** Handles the logic of filtering what bins are shown to the user based on the user's size capacity requirements inputted into the system, as well as the user's distance requirements inputted into the system.
14. **ModifyProfileInfoController:** Handles the logic when the user wishes to change their password or username by sending an email to the email account provided by the user during registration.
15. **ModifyUserAccController:** Handles the logic when an admin wishes to modify the account of a user. This class enables the admin to help the user change their password and/or username when the user is unable too, as well as to suspend the user in the event of malicious activity like spamming queries.
16. **LogActivityController:** Handles the logic of keeping an actor's (user or admin) actions on record, and allows the actions to be filtered by date or by the type of action.
17. **LogoutController:** Handles the logic for the termination of a session when an actor logs out of their account.

VIII. Application Skeleton

```
✓ helpers
  JS hbs.js
  JS messenger.js
> node_modules
✓ public
  > css
  > img
  ✓ js
    JS home.js
    JS main.js
    JS map.js
  {} EwasteRecycling.json
✓ service
  JS aboutRoute.js
  JS router.js
✓ views
  > layouts
  > partials
  ~ about.handlebars
  ~ articles.handlebars
  ~ home.handlebars
  ~ login.handlebars
  ~ map.handlebars
  ~ profile.handlebars
  ~ query.handlebars
  ~ reminder.handlebars
JS app.js
{} package-lock.json
{} package.json
```

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

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.

Service: Contain app 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:

- App.js : Entry point of the server.
- Package.json
- Package-lock.json

