Table of contents

Web Application: Cash for Trash

DELIVERABLES

- Documentation of functional and non-functional requirements
- Data dictionary
- Initial Use Case Model, consisting of Use Case diagram and Use Case descriptions
- UI Mockups

Functional Requirements

1. **The system should allow the user to register for a new account**
    - 1.1. The required information includes
        - 1.1.1. Name
        - 1.1.2. Email Address
        - 1.1.3. Password & Confirm Password
    - 1.2. The system should validate that all required fields have the sufficient inputs.
    - 1.3. The system should validate that the email address entered by the user is valid and has not registered.
        - 1.3.1. If the input email address is not valid, the system should display an error message and restart the register progress
        - 1.3.2. If the input email address has registered, the system should display an error message and direct the user to login progress
        - 1.3.3. If the input email address is valid and has not registered, the system should send a verification code to the email address. Then the user needs to enter in the verification code to complete email verification
    - 1.4. The system should validate that Password and Confirm Password matches and meet requirements.
        - 1.4.1. If the password has less than 8 characters, the system should prompt the user to enter the password again.
        - 1.4.2. If input password and confirm password do not match, the system should display an error message and clear both the input boxes.
    - 1.5. Users should be reminded that they cannot modify personal information once they have registered.
    - 1.6. Alternatively, the user should be able to register for a new account via supported external provider(s): Google
    - 1.7. The system should send a welcome email to the user’s email once the user has successfully registered and logged in.

2. **The system should allow the user to be able to log in with their account**
    - 2.1. The system should prompt the user for their registered email address and password to log in.
        - 2.1.1. The system should validate that all required fields have the sufficient inputs.
        - 2.1.2. The system should validate that the input email address is from the existing account list.
            - 2.1.2.1. If the input email address is not from an existing account, the system should display an “invalid input of email or password” message.

2.1.2.2. If the input email is from an existing account, but the input password is wrong, the system should still display an "invalid input of email or password" message.

2.2. The user should be able to login via supported external provider(s): Google

2.3. The system should allow users to change their passwords if they have forgotten it.

2.3.1. The system should provide a "Forgot Password" option for the users to choose and input a recovery email address.

2.3.2. The system should send a password reset email to that email address with a link that will redirect the user to the password reset page.

2.4. Upon successful login, the system should redirect the user to the landing page.

3. The system should allow the user to be able to change their passwords.

3.1. The system should provide a "Change Password" option for the users to choose if they want to change their passwords.

3.1.1. The system should provide a page with places for users to input 'Password' and 'Confirm Password'.

3.1.2. The system should validate that Password and Confirm Password matches and meet requirements.

3.1.2.1. Password should have at least 8 characters

3.1.3. The system should check if the inputs to "Password" and "Confirm Password" are the same.

3.1.3.1. If the input passwords are the same:

3.1.3.1.1. The system must notify the user that the password reset is successful through a pop-up message "Password Successfully Changed" after clicking the "Submit" button.

3.1.3.2. If the input passwords are not the same:

3.1.3.2.1. The system must notify the user that the password reset is unsuccessful through a pop-up message "Passwords not match" after clicking the "Submit" button.

3.1.3.2.2. The system should ask the user to try again through a pop-up message "Please try again".

4. The system should allow the user to be able to log out from the system

4.1. After logging in, the user must be able to click "Account" on their Homepage

4.2. After clicking "Account", the user must be redirected to the Account Settings page.

4.3. The user must be able to click “Log out” on the Account settings page.

4.3.1. After clicking “Log out”, the system should redirect the user to the login page.

5. The system should allow the users to recycle trash through the system

5.1. The system should allow users to upload the type and the mass of their trash to the system.

5.1.1. The user should be able to find and click on the type of their trash from the given numbers of trash categories listed by the system.

5.1.2. The user should be able to type in the numeric mass(kg) of their trash to system type by type.

5.2. The system should be able to show the value of the trash that the user had recycled according to the type and the mass of the trash.

6. The system should all the users to be able to earn points by recycling trash

6.1. Users should be able to see the value of each type of trash (the number of points that can be earned)

6.2. After the user uploaded the recycling information, the system should be able to award the user with points calculated according to the value of recycled trash.

6.3. Users should be able to use points to redeem prizes

6.3.1. System should display a list of prizes and the corresponding points required

6.4. System should display a regional ranking of users points

6.4.1. Users should be able to change their privacy settings to keep themselves out of the ranking

7. The system should allow users to find nearby recycling centers.

7.1. Users shall be able to search for nearby recycling centers based on their current location.

7.1.1. The system shall be able to display the detailed information of a specific recycling center

7.1.2. The system shall be able to show users the details of rewards that specific recycling centers have available now.

7.2. Users shall be able to search for other recycling centers in Singapore

8. The system should allow the users to be able to access their account information.

8.1. The system should display a user's current points balance prominently on the user interface.

8.2. Users should be able to view a detailed history of their recycling activities:
   8.2.1. Points earned
   8.2.2. Date and time
   8.2.3. Trash type
   8.2.4. Trash collection method(Self-delivery or pickup)
   8.2.5. Trade ID

8.3. Users should be able to view a detailed history of their rewards redeem records:
   8.3.1. Points spent
   8.3.2. Items redeemed
   8.3.3. Date and time
   8.3.4. Trade ID

9. The system should provide a Help Center for the user to contact administrators.

9.1. The system should allow users to suggest improvements and report problems through sending the email or calling the administrators.

Non-Functional Requirements

1. Usability
    1.1. The system should have a user-friendly interface that is easy to navigate and use.
    1.2. The system should provide clear and concise instructions for using its features.
    1.3. The system should provide outputs in an easily readable form.

2. Performance
    2.1. The system should provide accurate results within 3 seconds of request from the user.
    2.2. The system should be able to handle high traffic loads during peak usage times.

3. Security
    3.1. The system should encrypt sensitive data and use secure data storages to provide secure user account management.
    3.2. The system should protect against malicious attacks such as SQL injection, cross-site scripting, and other common web application vulnerabilities.
    3.3. The system should be regularly tested for security vulnerabilities and updated as necessary to ensure continued security.
    3.4. The system should mask the password field to ensure protection against potential shoulder surfing.

4. Extendibility
    4.1. The system should be designed to easily incorporate new features and functionality as needed.
    4.2. The system should be able to integrate with other systems and services as required.
    4.3. The system should be designed to scale and accommodate growing user numbers and increasing amounts of data.

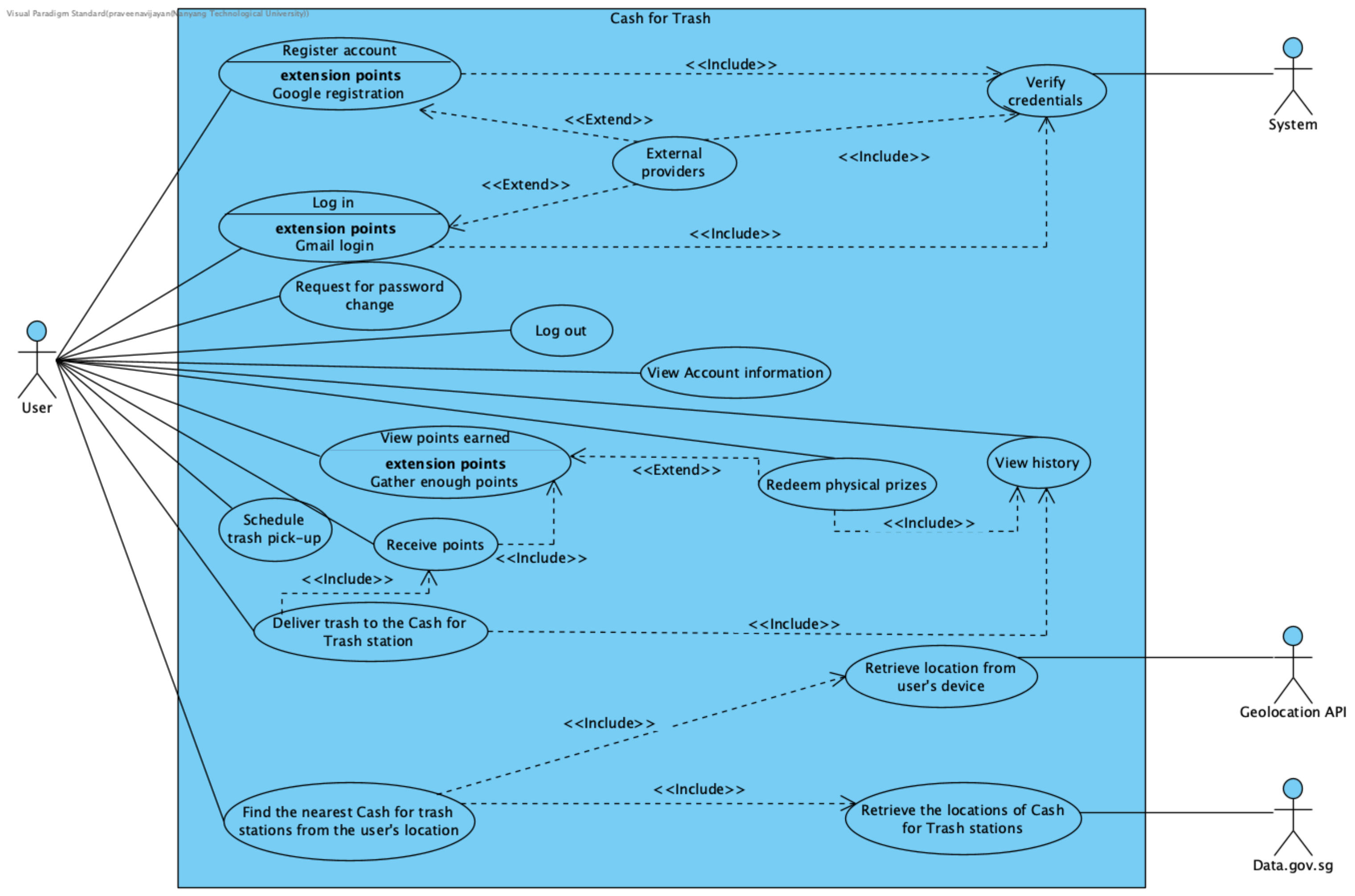
5. Supportability
    5.1. The system shall have robust logging and auditing capabilities, allowing administrators to track system activities, errors, and security events.
    5.2. The system shall provide detailed and user-friendly error messages, with error codes and recommended actions for common issues.
    5.3. The system shall have automated and regular backup procedures, and recovery processes shall be well-documented and tested.

Data dictionary

Field Name	Description
Email Address	User's email address, used for registering a new account and logging in.
Password	User's chosen password, used for logging into the system.
Verification Code	Code sent to the user's email for completing email verification.
Reset Link	Link included in the password reset email, redirecting the user to the password reset page.
Points Balance	Current balance of points for the user, used for redeeming prizes.
Recycling History	Detailed record of the user's recycling activities, including date, trash type, collection method, etc.
Redeem History	Detailed record of the user's prize redemption activities, including date, item redeemed, points used, etc.
Recycling Centers	Information about recycling centers provided by the system, including address.
Prizes	Information about prizes provided by the system, including name and required points for redemption.
Help Center	Information about the system's help center, including administrator contact details and user feedback channels.

Trash Type	Type of trash uploaded by the user for recycling calculations and points rewards.
Trash Mass	Mass of trash uploaded by the user for recycling calculations and points rewards.
Trade ID	ID for tracking recycling and prize redemption transactions.

Use Case Diagram



Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Register account		
Created By:	Vikass	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Allows the user to create a new account in the system by providing necessary information such as name, email address and password. Users can opt to register via external providers such as google or apple.
Preconditions:	1. User does not currently already have an account registered with the system.
Postconditions:	1. User will have a registered account in the system. 2. User will be directed to the home page.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system will prompt the user for details. 2. User enters their Name, Email address, and password. 3. System validates that the email has not been registered before. 4. System validates that the password meets the requirements. 5. User clicks on the ‘Sign up’ button. 6. System authenticates the sign up of the user successfully and redirects them to the home page.

Alternative Flows:	AF1: Sign up with External provider (Google) 1. User chooses to sign up with External provider. 2. System validates that the email has not been registered before. 3. Return to Step 6. AF2: Invalid Email address 1. System will inform the user of invalid email address through an error message. 2. System returns to Step 2. AF3: Password does not meet the requirements 1. System will inform the user that the password does not meet the requirements through an error message. 2. System returns to Step 2. AF4: Email address is already tied to an account 1. System displays a message that the email address is already tied to another account. 2. System will prompt the user to use a different email address instead. 3. System returns to Step 2.
Exceptions:	NIL
Includes:	Verify Credentials
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	2

Use Case Name:	Log in		
Created By:	Kenneth	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Enables a registered user to log in to their account using their email address and password.
Preconditions:	1. User currently already have an account registered in the system.
Postconditions:	1. User will be directed to the homepage.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system will prompt the user for details. 2. User enters their Name, Email address, and password. 3. System validates that the email has been registered before. 4. System validates that the password matches the registered email 5. User clicks on the ‘Sign in’ button. 6. System authenticates the sign up of the user successfully and redirects them to the home page
Alternative Flows:	AF1: Sign in with External provider (Google) 1. User chooses to sign in with External provider. 2. System validates that the email has been registered before.

	3. Return to Step 6. AF2: Invalid Email address 1. System will inform the user of invalid email address through an error message. 2. System returns to Step 2. AF3: Password does not match the username or email address 1. System will inform the user that the password is wrong or invalid through an error message. 2. System returns to Step 2.
Exceptions:	NIL
Includes:	Verify Credentials
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	3		
Use Case Name:	Log Out		
Created By:	Vikass	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Allows a logged-in user to log out of their account and terminate the current session.
Preconditions:	1. User must have an account registered with the system. 2. User must be currently logged in to the system.
Postconditions:	1. User will be logged out of the system, and the session will be terminated.
Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks on the ‘log out’ button. 2. System terminates the session. 3. User is brought back to the login page.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	4
Use Case Name:	Request for password change

Created By:	Kenneth	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Allows a logged-in user or the user who has forgotten their password to change their current password.
Preconditions:	1. User must have an account registered with the system.
Postconditions:	1. User will successfully changed their password
Priority:	
Frequency of Use:	
Flow of Events:	1. Once the user clicks on “Forget Password”, they must enter their email address registered in the system. 2. If the input email address is from an existing account, the system should inform the user that a password reset email has been sent to the email. 3. The password reset email should contain a link that will redirect the user to the password reset page. 4. User can change their password with similar requirements. 5. User has successfully changed their password.
Alternative Flows:	AF1: New Password does not meet the requirements 1. User will need to come up with another password that meets the requirement

	2. Return to step 5.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	5		
Use Case Name:	View account Information		
Created By:	Vikass	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Enables the user to view their account information such as profile information and points balance.
Preconditions:	1. User must have an account registered with the system. 2. User must be logged in to the system.
Postconditions:	1. User will be able to choose what information they want to see about their account.

Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks on the ‘account settings’ tab in the system. 2. System provides different information on the user’s account, such as points earned, points spent, items redeemed, etc.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	6		
Use Case Name:	View History		
Created By:	Vikass	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System

Description:	Enables the user to view their activity history within the system such as recycling history, rewards redeemed and points spent on each reward.
Preconditions:	1. User must have an account registered with the system. 2. User must be logged in to the system.
Postconditions:	1. The user will be able to view the activity history linked with that account.
Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks on the 'Account Settings' tab. 2. User clicks on the 'View History' tab. 3. User will be able to see details of their history, such as recycling history, rewards redeemed, points spent on each reward, etc.
Alternative Flows:	AF1: User does not have any history linked with that account 1. System will display the values under the 'View History' tab as NIL.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	7		
Use Case Name:	Receive Points		
Created By:	Kenneth	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Users should be able to receive points after submitting the trash and the information provided has been verified.
Preconditions:	1. Users must have an account registered in the system 2. Users must be logged in the system
Postconditions:	1. Users will receive a certain number of points from the system.
Priority:	
Frequency of Use:	
Flow of Events:	1. User deposits the trash into a collection point 2. The system will calculate the weight of the trash deposited and type of trash. 3. The system will convert the calculations into points that will be sent to the users account
Alternative Flows:	NIL

Exceptions:	NIL
Includes:	View points earned
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	8		
Use Case Name:	View Point Earned		
Created By:	Kenneth	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Enables the user to view the points they have earned through their past activities.
Preconditions:	1. Users must have an account registered in the system. 2. User must be logged in the system
Postconditions:	1. Users will be able to view the points the user has earned.
Priority:	

Frequency of Use:	
Flow of Events:	1. User clicks on the ‘Account Settings’ tab. 2. User clicks on the ‘View History’ tab. 3. Users will be able to see details of their history, such as points earned, rewards redeemed, points spent on each reward, etc.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	9		
Use Case Name:	Find the nearest Cash for Trash station from the user’s location.		
Created By:	Kenneth	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System, Geolocation API, data.gov.sg
Description:	User’s should be able to find the nearest cash for trash stations from their location to conveniently deliver the trash.

Preconditions:	1. Users must have an account registered in the system. 2. User must be logged in the system 3. The user must have a device capable of accessing their location
Postconditions:	1. Users will be able to find the nearest cash for trash station.
Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks on the “Locations” tab 2. Using the included ‘Retrieve locations of Cash for Trash stations’ use case, the system will display a map showing all the Cash for Trash stations in the area. 3. Using the included ‘Retrieve location from user’s device’ use case, system will also display the user’s current location on the map 4. Users can see the available Cash for Trash stations close to them.
Alternative Flows:	NIL
Exceptions:	The user’s location setting is not turned on.
Includes:	Retrieve location from user’s device, Retrieve the locations of Cash for Trash stations.
Special Requirements:	Location of the user must be available.
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	10

Use Case Name:	Redeem Prizes		
Created By:	Vikass	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	User, System
Description:	Users should be able to use the points earned to redeem prizes corresponding to the points.
Preconditions:	1. Users should have an account registered with the system. 2. Users should be logged in to the system.
Postconditions:	1. Users will gain prizes in their registered account, that they can use in future.
Priority:	
Frequency of Use:	
Flow of Events:	1. User clicks on the ‘Rewards’ tab. 2. System provides the user with options of different prizes that range in value, that have associated points values linked to it. 3. User selects the prize he/she wishes to get, using the points available. 4. The system deducts the respective number of points from the account. 5. The system registers the prize to that account, which is saved for future use.
Alternative Flows:	AF1: User does not have enough points for the prize 1. System prompts an error message saying that the user does not have enough points for that prize. 2. Return to step 3.

Exceptions:	NIL
Includes:	View history
Special Requirements:	NIL
Assumptions:	The prizes are always available to the user for redemption.
Notes and Issues:	NIL

Use Case ID:	11		
Use Case Name:	Retrieve location from the user’s device		
Created By:	Kenneth	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	System, Geolocation API
Description:	The location of the user is retrieved by using geolocation API.
Preconditions:	1. User must have an account registered in the system 2. User must be logged in the system
Postconditions:	1. The user’s location is made available.
Priority:	
Frequency of Use:	

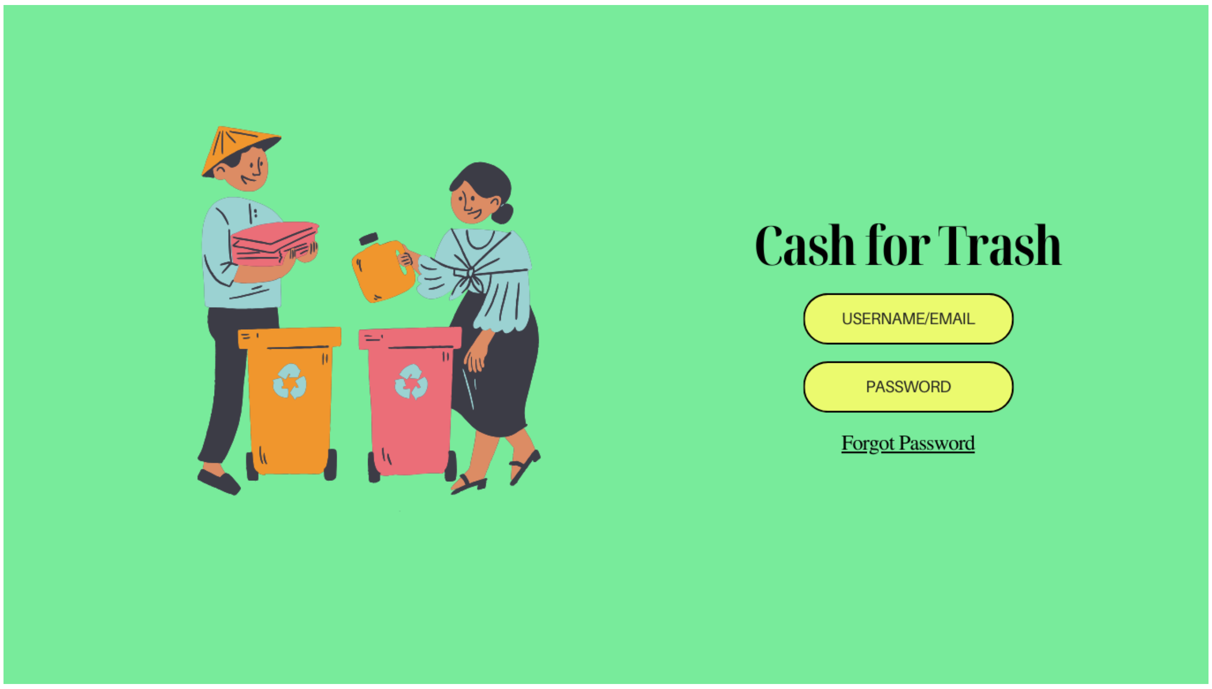
Flow of Events:	1. User turn on their location 2. The geolocation API obtains the current location of the user. 3. The geolocation API sends the location information of the user to the system.
Alternative Flows:	NIL
Exceptions:	EX1: The location is unable to be obtained from the geolocation API 1. System shows a message saying that it is unable to get the user’s location
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	12		
Use Case Name:	Retrieve locations of cash for trash stations		
Created By:	Vikass	Last Updated By:	
Date Created:	8/2/2024	Date Last Updated:	

Actor:	System, data.gov.sg
Description:	The locations of Cash for Trash stations are retrieved by the GEOJSON data from data.gov.sg

Preconditions:	1. Data.gov.sg must contain the data of the locations of the Cash for Trash stations.
Postconditions:	1. The system will be able to obtain the locations of the Cash for Trash stations across Singapore.
Priority:	
Frequency of Use:	
Flow of Events:	1. The system obtains the locations of the Cash for Trash stations from the data.gov.sg database.
Alternative Flows:	NIL
Exceptions:	EX1: The data is not obtainable from data.gov.sg 1. The system displays a message saying that the data of the Cash for Trash station locations is not available.
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

UI Mock Up

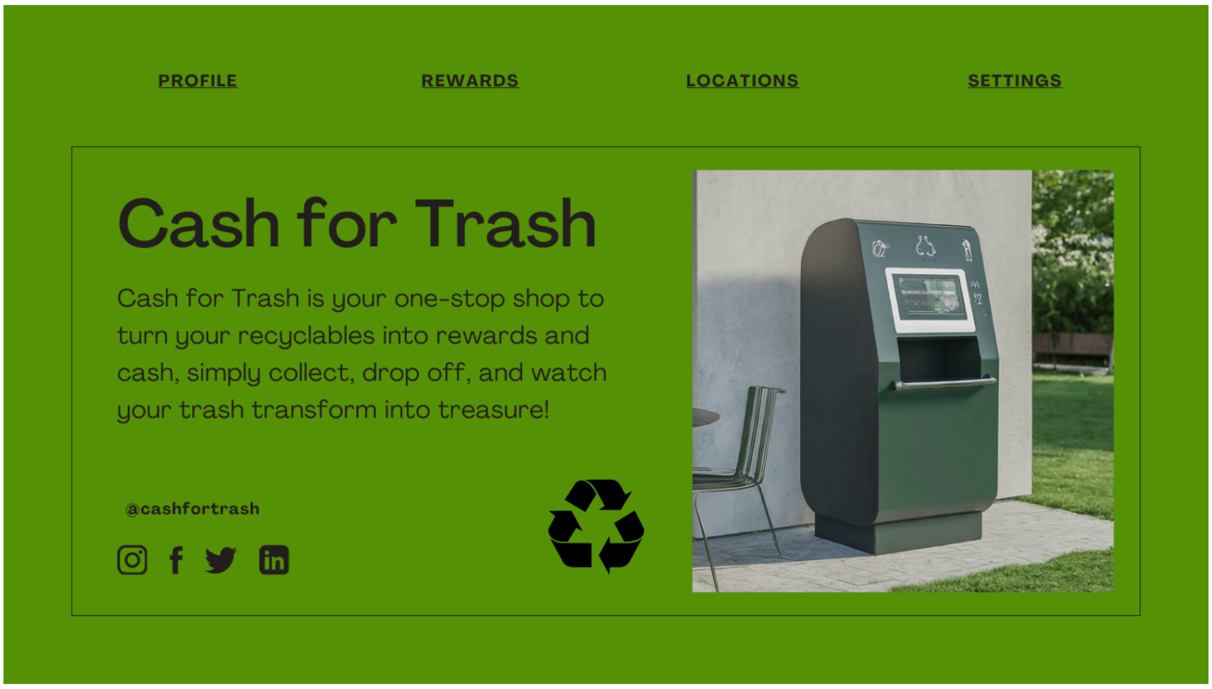


Cash for Trash

USERNAME/EMAIL

PASSWORD

Forgot Password

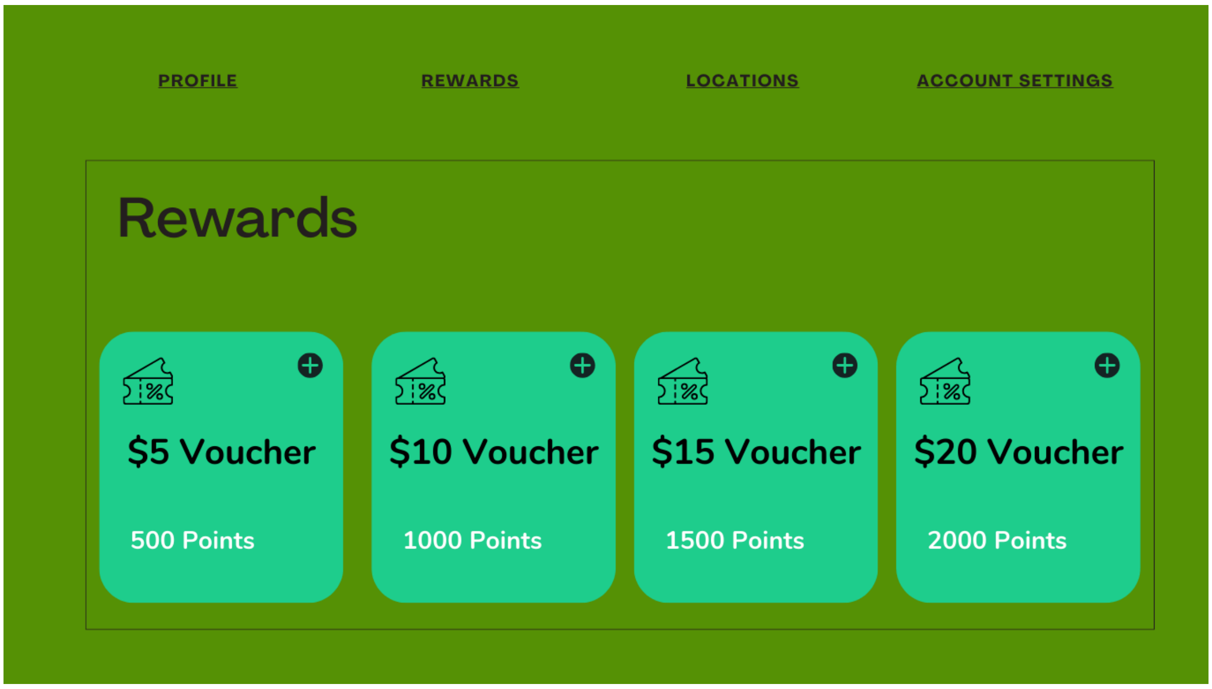


PROFILE REWARDS LOCATIONS SETTINGS

Cash for Trash

Cash for Trash is your one-stop shop to turn your recyclables into rewards and cash, simply collect, drop off, and watch your trash transform into treasure!

@cashfortrash



PROFILE REWARDS LOCATIONS ACCOUNT SETTINGS

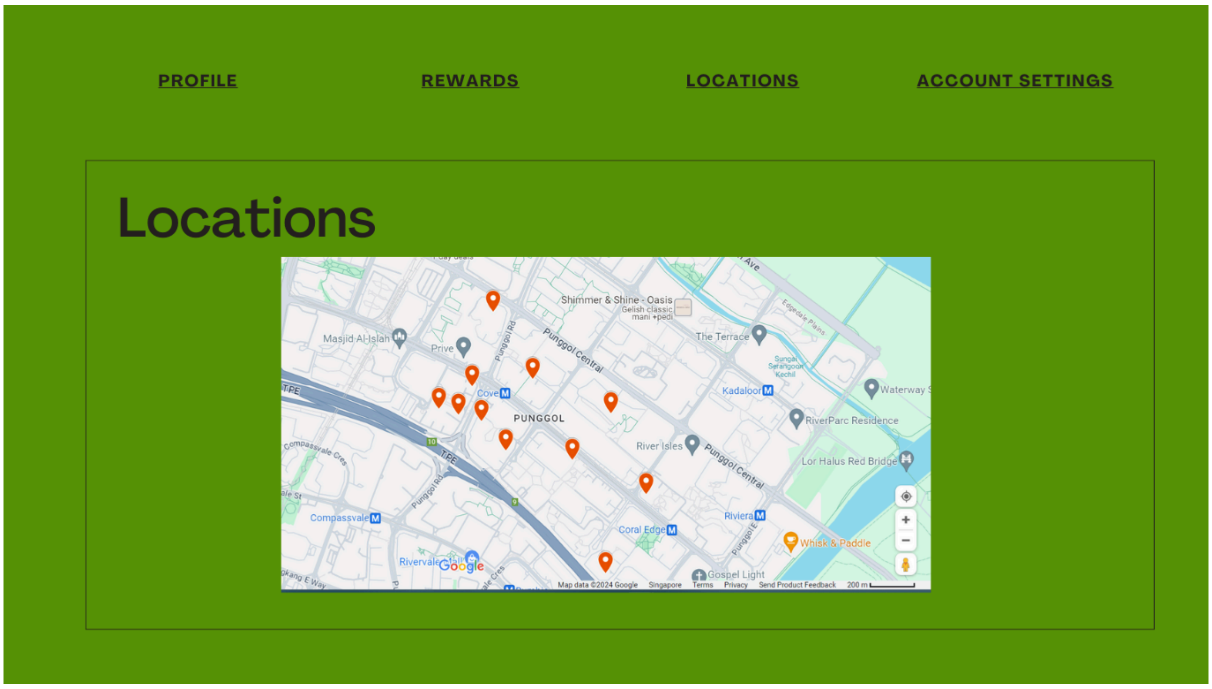
Rewards

$5 Voucher
500 Points

$10 Voucher
1000 Points

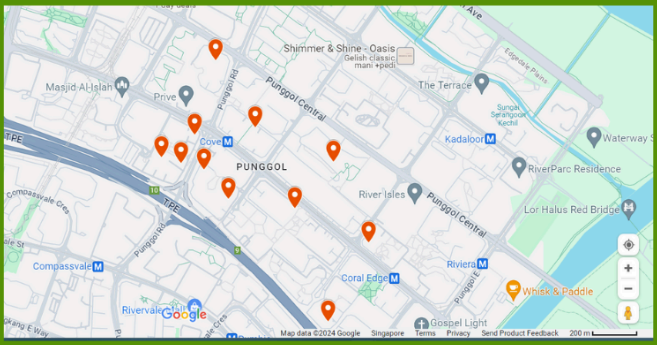
$15 Voucher
1500 Points

$20 Voucher
2000 Points



PROFILE REWARDS LOCATIONS ACCOUNT SETTINGS

Locations



PROFILE REWARDS LOCATIONS ACCOUNT SETTINGS

Settings

- Edit Account Details
- Transaction History
- Change Password
- Ranking Privacy Settings
- Log out