**BIN TRACKER (TrashTogether (tbc))**

**Functional Requirements**

1. Main menu
   1. The system must be able to display the option to find the locations of recycling bins and collection points.
   2. The system must allow user to register their account.
   3. The system must allow user to login to their account.
2. Registration
   1. The user must be able to register for a new account via our system registration.
      1. Information include:
         1. Name
         2. Password
         3. Confirm Password
         4. Valid Email Address
   2. The system must validate all required fields have been filled up.
   3. The system must validate the account availability
3. Login
   1. The user must be able to login using their email address and password.
   2. The system must validate that both fields have been filled up.
   3. The system must validate the email address and password are valid.
   4. If the login information is invalid, the system will display an error message.
   5. The system will redirect the user to the application’s main menu.
   6. The user must be able to request for a change of password.
   7. The system must be able to send email verification code upon the user's request for a change of password.
4. Location
   1. The system must be able to display recycling bin locations in ascending order of distance from the user.
   2. The user must be able to search for a variety of recycling bins based on types of recyclable material. (e.g. plastic, paper, clothes, e-waste, etc.)
   3. The system will redirect the user to the directions to the location of interest.
5. Collection points (cash for trash)
   1. The system must be able to display collection points for trading cash for trash in ascending order of distance from the user.
   2. The system will redirect the user to the directions to the location of interest.
6. Directions
   1. The system must be able to display directions to the user’s location of interest.
7. Estimate cash returns
   1. The system must be able to display the estimated cash returns of *common trash* traded for cash.
   2. The user must be able to record items to be traded for cash.
8. Interface with other systems
   1. The system must be able to retrieve location from user’s device via Geolocation API.
   2. The system must be able to retrieve the locations of recycling bins and collection points from Data.gov.sg.
9. Formats for information to be processed
   1. Location
      1. Distance format must be in “km” and two decimal places of accuracy (e.g. 1.50km) if distance >=1000m.
      2. Distance format must be in “m” if distance < 1000m.
   2. Price
      1. Price format must be in “S$” and two decimal places of accuracy. (e.g. S$1.50)
   3. Weight
      1. Weight format must be in “kg” and two decimal places of accuracy (e.g. 1.50kg) if weight >=1000g.
      2. Weight format must be in “g” if distance < 1000g.

**Non-functional requirements**

1. Security requirements
   1. The system will perform hashing on the password created for the account before storing the password into the database for better security of the user’s password.
   2. When the user inputs the password for the account, the system will hash the input password and compare it with the hashed password in the database for that user account.
2. Usability requirements
   1. The system must allow easy reversal of actions, where the user must be able to change the recorded items for cash for trash.
   2. The system must offer informative feedback, and display proper error messages and feedback when user keys in invalid inputs
3. Performance requirements
   1. The application should not have any bugs and the user interface must be fast and responsive according to the user’s action.
   2. The system should return results in the minimum time possible.
4. Data Integrity requirements
   1. The application must only provide accurate data of locations.
   2. The system should ensure that the directions to the recycling bins must be correct
5. Scalability requirements
   1. The system should perform efficiently when tasked with any number of users.
   2. The system should ensure that the application is reliable and minimize downtime as much as possible.
   3. The system must be designed with the Model-View-Controller architecture and design patterns to support future enhancements and increase scalability.

**Data dictionary**

|  |  |
| --- | --- |
| Term | Definition |
| User | A user is a person who is using the application/website to find recycling bins or collection points to trade trash for cash in Singapore |
| System | A system refers to the TrashTogether mobile application/website. |
| Filter | Filter is a feature that allows users to find specific types of recycle bins or collection points that they want to search for, such as plastic, paper, clothes, e-waste, etc. |
| Registration | Registration is a feature that allows users to sign up for an account. |
| RB Location | Recycling Bin Location is a feature that allows users to find recycling bins |
| Collection points | Collection points is a feature that allows users to find collection points |
| Direction | Direction is a feature that allows users to be guided to a location of interest using GPS. |
| Model-View-Controller | Model-View-Controller is a software architectural pattern for implementing user interfaces in the application. |
| GPS | Global Positioning System which will detect a user’s current location. Is used interchangeable with WIFI positioning system |

Use Case Description

**Use Case Model 1 : Creating a new account**

|  |  |
| --- | --- |
| **Actor:** | User |
| **Description:** | Creating a new account for a user. |
| **Preconditions:** | 1. New user account must not exist in the database previously. 2. Users must be connected to WiFi or mobile data during this process. |
| **Postconditions:** | A verification email will be sent to the email address that the user used to create a new account. |
| **Priority:** | Medium |
| **Frequency of Use:** | Once in a lifetime |
| **Flow of Events:** | 1. User enters username,account password, email address and password of email address. 2. User clicks on the “Create an account” button. 3. System sends a verification email to the email address indicated previously. 4. User inputs the verification code that is found in the verification email. 5. New user account is created. |
| **Alternative Flows:** | password does not match confirm password |
| **Exception:** | - |
| **Includes:** | Checking whether a user account has previously existed or not. |
| **Special**  **Requirements:** | - |
| **Assumptions:** | Each user only needs one account and doesn’t create a second account. |
| **Notes and Issues:** | - |

**Use Case Model 2 : Searching for near trash for cash locations**

|  |  |
| --- | --- |
| **Actor:** | User |
| **Description:** | Searching for near trash for cash locations |
| **Preconditions:** | 1. User must be connected to internet 2. User must turn on location/ set location |
| **Postconditions:** | System displays the closest collection point and offers to direct the user to it. |
| **Priority:** | Medium |
| **Frequency of Use:** | Once per week |
| **Flow of Events:** | 1. User searches for cash for trash locations 2. All Locations are displayed in ascending order of distance from the user 3. The user may select one of the locations 4. The system will then use gps to guide the user to the location chosen |
| **Alternative Flows:** | User is not connected to Internet and has not set a location:  System will inform the user that they need to connect.   Cash for trash stations are suspended:  System will inform the user that no stations are available at the moment |
| **Exception:** | - |
| **Includes:** | - |
| **Special**  **Requirements:** | - |
| **Assumptions:** | - |
| **Notes and Issues:** | - |

**Use Case Model 3 : Validating account availability**

|  |  |
| --- | --- |
| **Actor:** | System |
| **Description:** | Validate the account availability |
| **Preconditions:** | 1. User account must not already exist in the database 2. Mobile must be connected to WIFI/Mobile Data |
| **Postconditions:** | 1. User account is successfully created and stored in the database. 2. Users are able to login using their account successfully. |
| **Priority:** | Medium |
| **Frequency of Use:** | Once in a lifetime |
| **Flow of Events:** | 1. System will check if the account already exists in the database. 2. System will store the account details in the database |
| **Alternative Flows:** | AF-S1-1: The account already exists in the database.   1. System will display an error message 2. User will enter new account details 3. User will reattempt to register using the new details 4. Return to step 1 |
| **Exception:** |  |
| **Includes:** | 1. Validate the account availability |
| **Special**  **Requirements:** | - |
| **Assumptions:** | Database refers to system’s database |
| **Notes and Issues:** | - |

**Use Case Model 4 : Change password**

|  |  |
| --- | --- |
| **Actor:** | User |
| **Description:** | Allows User the option to change their account’s password |
| **Preconditions:** | 1. Mobile must be connected to WiFi/data |
| **Postconditions:** | 1. System successfully updates User’s new password. 2. User is able to login with the new password successfully. |
| **Priority:** | Medium |
| **Frequency of Use:** | 1 - 3 times per year |
| **Flow of Events:** | 1. User type in their email address and tap the Change password button. 2. System will check if the email address is registered. 3. System will send an OTP to the registered email address provided. 4. User will enter the OTP. 5. System will validate the OTP. 6. User will key in the new password and confirm new password, and tap Confirm password changes. 7. System will check if new password matches confirm new password. 8. System will save the new password into the database. |
| **Alternative Flows:** | AF-S2: The provided email address is not registered and cannot be found in the database.  AF-S5: User enters an invalid OTP.  AF-S7: New password does not match confirm new password. |
| **Exception:** |  |
| **Includes:** |  |
| **Special**  **Requirements:** | - |
| **Assumptions:** |  |
| **Notes and Issues:** | - |

**Use Case Model 5: Searching for recycling bin locations**

|  |  |
| --- | --- |
| **Actor:** | User |
| **Description:** | Searching for recycling bin locations |
| **Preconditions:** | 1. User must be connected to internet 2. User must turn on location/ set location |
| **Postconditions:** | System displays the closest recycling bin and offers to direct the user to it. |
| **Priority:** | High |
| **Frequency of Use:** | Once per week |
| **Flow of Events:** | 1. User searches for recycling bin locations 2. All locations are displayed in ascending order of distance from the user 3. The user may use the filter feature to find recycling bins for a certain recyclable material. 4. The user may select one of the locations 5. The system will then use gps to guide the user to the location chosen |
| **Alternative Flows:** | User is not connected to Internet and has not set a location:  System will inform the user that they need to connect. |
| **Exception:** | - |
| **Includes:** | - |
| **Special**  **Requirements:** | - |
| **Assumptions:** | - |
| **Notes and Issues:** | - |

**Use Case Diagram**Diagram

Description automatically generated

**UI design mockup:**

Log in / Sign up page

Graphical user interface

Description automatically generated

Searching for locations:

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated