**Household food tracker**

**Functional requirements**

Sign up

1. Creating new accounts

1.1. Choice of singpass or unique username and password to sign up

1.2. All login information must be saved in cloud

Login/out

1. User must be able to login to existing accounts

1.1. System must be able to verify login information with data stored

1.2. Prompted of any input information is wrong

1.3. Limited login attempts given

2. System must allow user change password

3. Must be able to conveniently logout at anytime

3.1. Automatically logout if user exits application

Food tracking

1. User must be able to manage food purchases

1.1. System must allow users to input items purchased and relevant product information.

1.2. Default date of purchase will be the day which items are keyed in, unless specified by the user.

1.3. System must allow user to modify list of purchased item

1.4. All inputs and changes must be saved by the system

1.5. System must be able to retrieve list of purchased item and relevant product attributes as requested by user

2. List of purchase product should be able to be sorted according to filters

3. System will be able to get information (calories and healthy logo)

3.1. Use spoonacular api to get calorie information

3.2. Use data.gov.sg Healthy logo csv to determine if product has healthy logo

3.3. If api return with no result, system will prompt user to add missing information

Set a meal/individual item consumption

1. User must be able to record items used

1.1. Items user has will be displayed

1.2. System accept quantity used from user and update item balance

1.2.1. Check if quantity is sufficient

1.2.2. Strike off item when quantity reach zero

1.3. System allows items to be group together and be seen as a meal

1.3.1. System allows user to input additional description for meals

1.3.2. System will total up calories in meal

1.3.3. All items in a meal, their quantity ,total calories and description all be recorded

1.4. Default case: single item consumption

1.4.1. Update item balance

1.4.2. System will only reflect name and calories

2. User can modify items used in meal

2.1. Limit modification period

3. For individual item or meal, date entered will be save for records function.

4. User can delete individual item/meal entries

4.1. Limit modification to before day ends

Records

1. User can retrieve records of consumption/calories

1.1. In the form of daily/weekly

1.2. System allows user to download statement

**Non-functional requirements**

Security

1. The system must hash the password created for the user's account.

2. When users try to login, the system must compare the input password with the hashed password in the database.

3. After 3 failed login attempts, the user account will be locked to protect the user information from hackers

Usability

1. Help messages must be displayed according to the user’s locale.
2. System must collect the feedback from users about improvement and Bugfix after users press the feedback button

Reliability

1. Data should be back up frequently

2. System should be available 24/7

3. If a reboot is required, full system functionality must be restored within 5 minutes

Performance

1. login verification process is done almost immediately
2. retrieval of information and after displaying results is done within seconds

.

Supportability

1. The database can be replaced with any commercial product supporting standard SQL queries.

**Data Dictionary**

1. Household - People living in the same accommodation with a common storage for food items.
2. User - Anyone in one household are users. They have access to a common account using a unique username and password.
3. items input - Date, Qty & balance with number of calories to be put in the available list.
4. Available list - items that the user has previously purchased but have yet to finish
5. Meal - a group of items selected from available list
6. output - number of calories one user consumed in a specified region of time.

**Use Case Model:**

**Use Case Model 1:Registration:**

| Actor: | User |
| --- | --- |
| Description: | This use case depicts how users create a new account. |
| Trigger: | Click on “create a new account” |
| Preconditions | User device should have access to Internet(WIFI/4G) |
| Failed End condition | 1. This user account has been in database;  2. Users do not input at least one capital letter in the password; |
| Basic Flow | 1.Users enter email/singpass, email/singpass password, username and password.  2.Users click on “create a new account”.  3.System receives users information.  4.System check if this new account is valid.  5.If valid, a new user account is created. |
| Alternate Flow | Scenario 1: Step 4 fails  1.System will display an error message.  2.Users need to re-enter information.  3.Return to step 1. |
| Post conditons | 1.New user account is added to the database successfully.  2.A verification email/SMS is sent to the user. |
| Frequency of occurrence | Once a lifetime |
| Special Requirement |  |

**Use Case Model 2:Change password:**

| Actor: | User |
| --- | --- |
| Description: | This use case depicts how users change passwords. |
| Trigger: | Click on “Change password” |
| Preconditions | User device should have access to Internet(WIFI/4G) |
| Failed End condition | This user account cannot be found in the database. |
| Basic Flow | 1.Users click on “Change password”.  2..Users enter a new password.  3.System check if this new password is valid.  4.System sends verification code to users through SMS/email.  5.Users input the verification code.  6.System receives users information.  7.System updates account information. |
| Alternate Flow | Scenario 1: Step 3 fails  1.System will display an error message.  2.Users need to re-enter password.  3.Return to step 2.  Scenario 2: Step 5 fails  1.System will display an error message.  2.Users need to re-enter/resend verification code.  3.Return to step 5. |
| Post conditons | 1.New password is updated to the account successfully.  2.A verification email/SMS is sent to the user. |
| Frequency of occurrence | 1-2 times a month |
| Special Requirement |  |

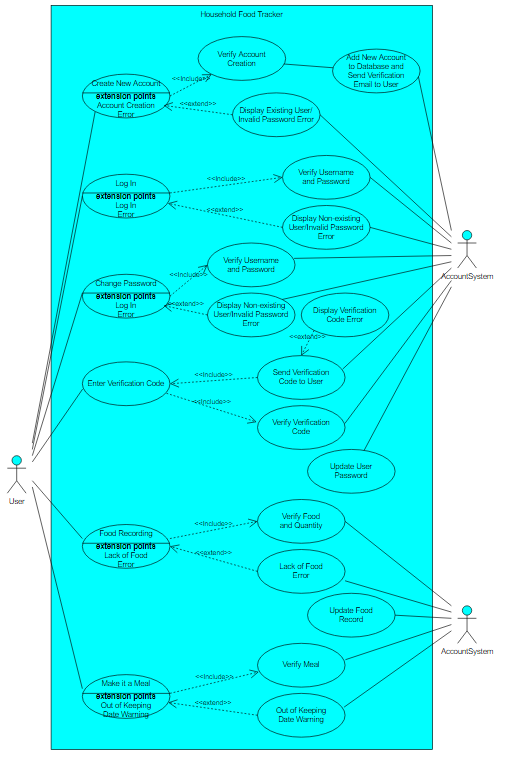
**Use Case Model 3:Food Recording**

| Actor: | User |
| --- | --- |
| Description: | This use case depicts how users record the food they purchased. |
| Trigger: | Click on “Start Recording” |
| Preconditions | User device should have access to Internet(WIFI/4G)  User must login to their accounts |
| Failed End condition |  |
| Basic Flow | 1.Users click on “Start Recording”.  2.Users enter the name of the food they purchased.  3.Users can enter the date they want to eat the food(optional),default is that day  4.Users enter the Qty/Balance of the food they purchased.  5.Users enter the time for each item they want to keep(optional),default depends on Healthy logo.  6.Then Users click on the ”Finish” button. |
| Alternate Flow | Scenario 1: lack of necessary items.  1.System will display an error message.  2.Users need to check and fill in the blanks that with”\*”.  3.Return to step 2. |
| Post conditons | 1.New record is updated to the user account successfully.  2.System will display “Congratulations! record successfully!”  3.The status of each item in this new record will be “In Progress” |
| Frequency of occurrence | not limited |
| Special Requirement |  |

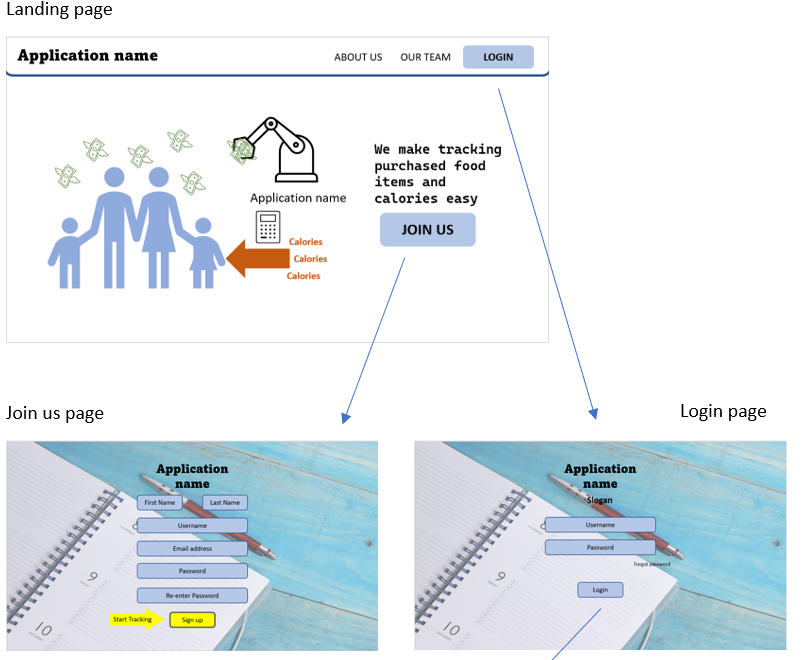
**Use Case Model 4:Make it a meal**

| Actor: | Users and System |
| --- | --- |
| Description: | This use case depicts how the system updates the calories users take. |
| Trigger: | User click on “make it a meal” |
| Preconditions | User device should have access to Internet(WIFI/4G)  User must login to their accounts |
| Failed End condition | Users select the item already out of use. |
| Basic Flow | 1.Users click on “make it a meal”.  2.Users select the food they owned in their record.  3.Users input the Qty/Balance they want to eat in this meal.  4.Users click on “Yes, I want to eat them!”  5.System receives the item users' consumption, and invokes calories data from spoonacular.  6.System will calculate total calories and display them in a list to users.  7.System will record the calories and delete the food users consume this time. |
| Alternate Flow | Scenario 1: if the food in one record is close to/out of keeping date.  1.System will display a reminder message.  2.Return to step 2. |
| Post conditons | 1.Food consumption is updated to the user account successfully.  2.System will display “Congratulations! Good meal!”  3.The balance of food will be reduced. |
| Frequency of occurrence | not limited |
| Special Requirement |  |

**Use Case Diagram:**

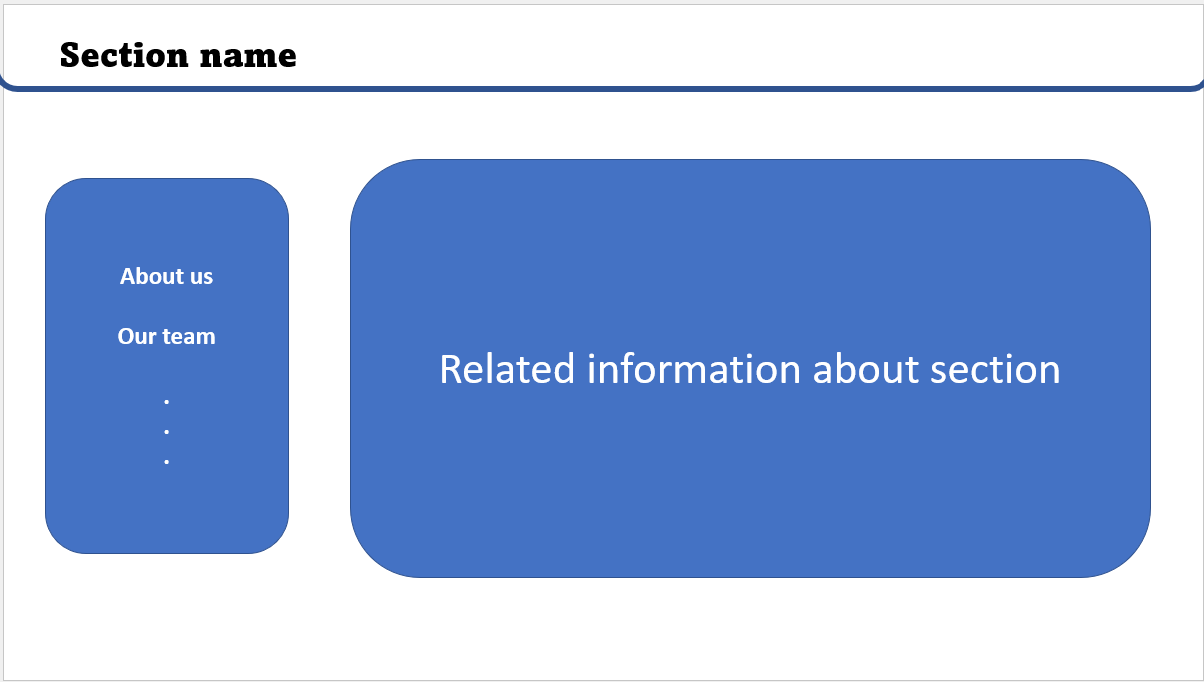
****

**UI Mockup**

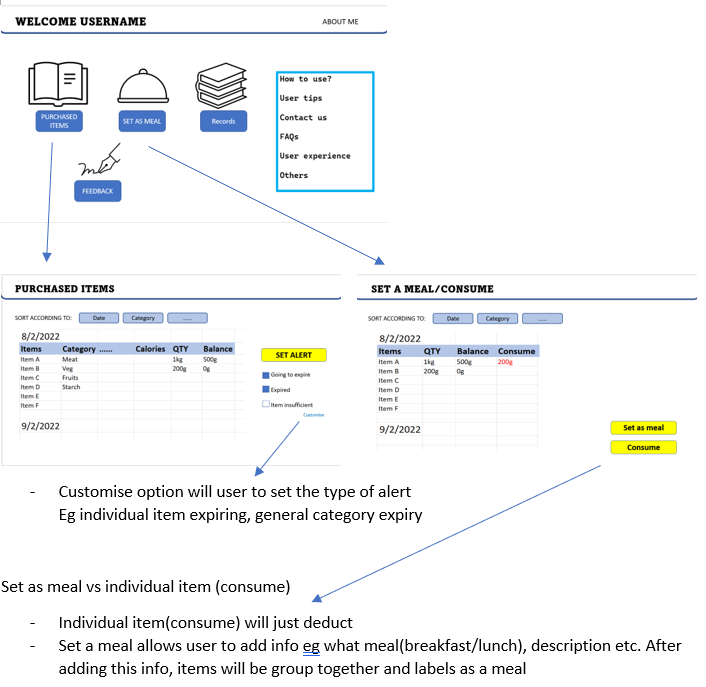


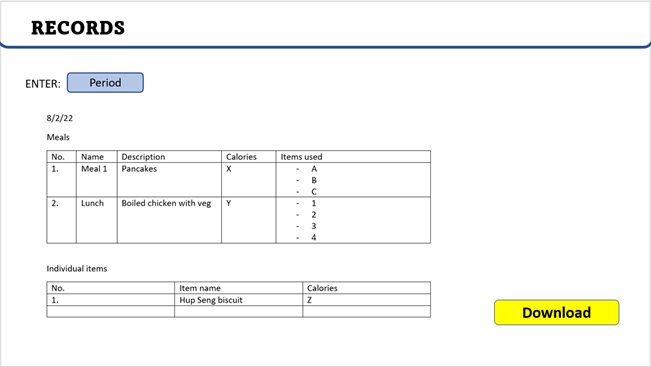
* After signing up, a verification email will be sent.
* user can then proceed to the login page to login

About us and our team options in the landing page



Main page (after login) and functions pages





* period: daily/weekly



clicking any of this option in main page will lead to:

