

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 Lab2 Deliverables

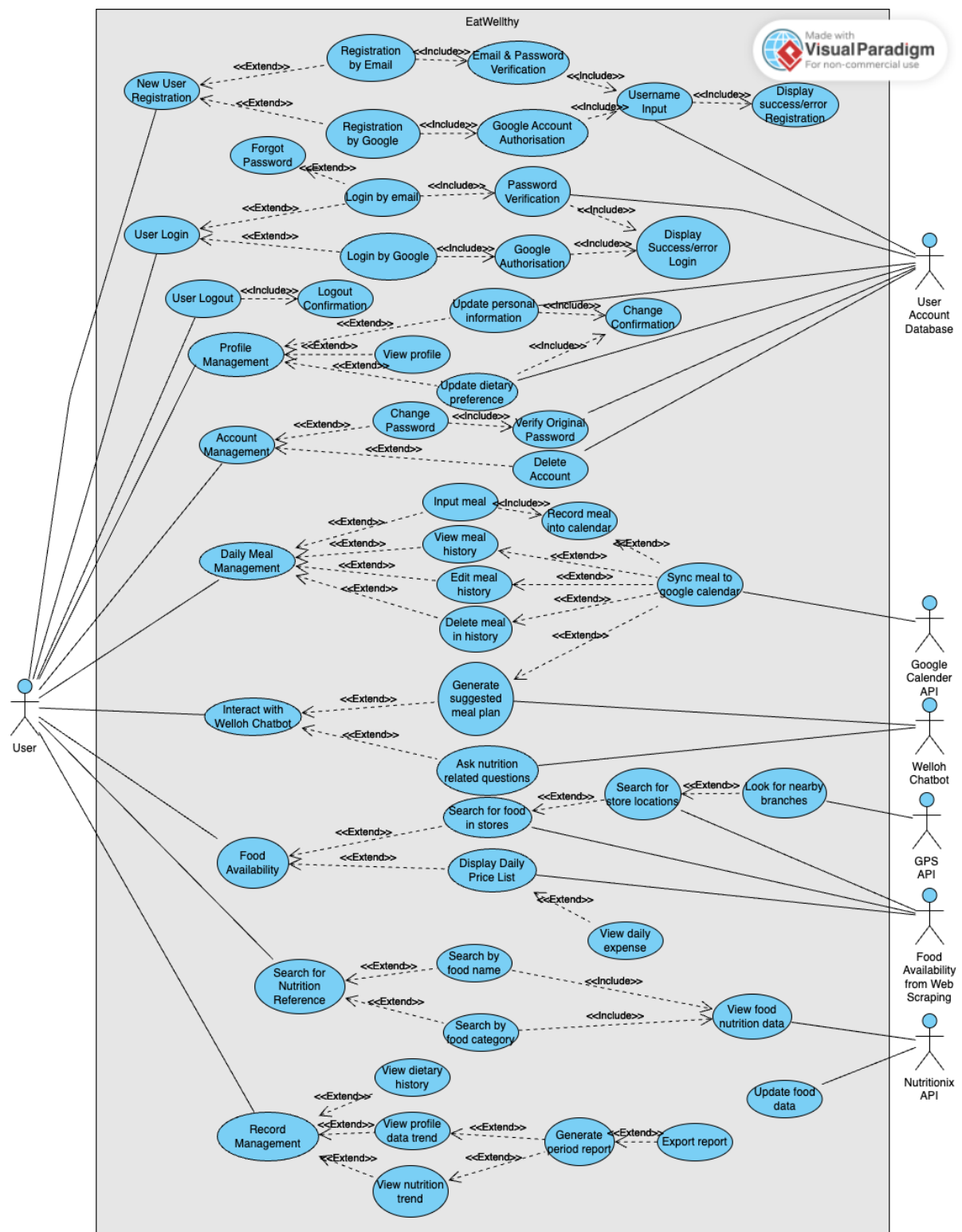
Lab index	SDAA	
Group number	31	
Group members	Liu Xiaotao	U2320836F
	Low Jo Yi, Nicole	U2321370D
	Mahi Pandey	U2321382F
	Mehta Rishika	U2323133H
	Zhang Yichi	U2320736J
	Zhao Qixian	U2321752L

Contents

Use Case Model and Use Case Descriptions.....	4
I. For Functional requirement #1 (Registration)	5
U0101 New User Registration using email.....	5
U0102 New User Registration using Google Account	7
U0103 User Login via Email.....	8
U0104 User Login via Google Account	10
U0105 Reset Forgotten Password.....	11
U0106 Change Account Password	12
II. For Functional Requirement #3 (Profile Management)	13
U0201 View User Profile.....	13
U0202 Update User Profile Information	14
U0203 Manage Dietary Preferences	15
U0204 Update Security Settings	16
U0205 Delete User Account.....	17
III. For Functional Requirement #4 (Nutrition Calculation)	19
U0301 Log Daily Meal.....	19
U0302 Edit Logged Meal	21
U0303 Delete Logged Meal	23
U0304 View Meal History	25
U0305 Generate Personalized Diet Plan	27
U0306 Sync Diet Plan with Google Calendar	30
IV. For Functional Requirement #8 (Welloh Chatbot)	32
U0401 Ask Nutritional Questions.....	32
U0402 Receive Nutritional Alerts.....	34
U0403 Query Nutritional Information via Chatbot.....	37
V. For Functional Requirement # (Nutrition Reference Sheet)	40
U0501 Search for Food Items.....	40
U0502 Update Food Database.....	41
U0503 Request store availability data.....	43
U0504 Manage User data	44
U0505 Backup and Restore	46
VI. For Functional Requirement #7 (Meal Planner).....	47
U0601 View Dietary History	47
U0602 Analyze Nutritional Intake Trends.....	48
U0603 Generate User Reports.....	50

U0604 Export Nutritional Data	51
Class Diagram	53
Key Boundary Diagram	53
Sequence diagram 1-3	54
Dialog Map	58

Use Case Model and Use Case Descriptions



Use Case Descriptions

I. For Functional requirement #1 (Registration)

U0101 New User Registration using email

Use Case ID:	U0101		
Use Case Name:	New User Registration using email		
Created By:	Zhao Qixian	Last Updated By:	Mahi Pandey
Date Created:	29/8/2024	Date Last Updated:	04/09/2024

Actor:	New User
Description:	It enables new users to create an account within EatWellthy's system database. To create an account, users are required to provide their Username, Email, and Password. New Users will be assigned a unique UserID.
Preconditions:	<ol style="list-style-type: none">1. The selected Username must be unique.2. Passwords must adhere to the following requirements:<ul style="list-style-type: none">• Must contain at least 3 out of these 4 conditions: small letter character, capital letter character, number, special character(e.g., @, #, !).• Must be a minimum of 8 characters in length.3. The supplied Email must not be associated with an existing user account.
Postconditions:	<p>Upon a successful registration, the system seamlessly executes the following actions:</p> <ul style="list-style-type: none">• Instantly dispatches a confirmation link to the user's provided Email address, ensuring account verification.• Redirection to the main page of EatWellthy occurs seamlessly.• The user is assigned a unique UserID• The user is now equipped to access the app's features, using their newly created Username and Password, or their Email and Password.• The user's UserID, email, Username and encrypted Password is stored in the database.
Priority:	High
Frequency of Use:	One-off
Flow of Events:	<ol style="list-style-type: none">1. The new user opens EatWellthy and selects the "New User Register" option.2. The system presents the user with a registration form, soliciting the following details:<ul style="list-style-type: none">• Name• Username (I want to be addressed as)

	<ul style="list-style-type: none"> Email Password <p>3. The user fills in the necessary information and clicks "Submit".</p> <p>4. The system conducts immediate validation on the user's input, verifying compliance with the preconditions mentioned earlier.</p> <ul style="list-style-type: none"> If any precondition is not met, the system displays precise error messages, guiding the user to correct their input. If the input is valid, the registration process advances, and the user is directed to email verification.
Alternative Flows:	<p>1. Validation Errors</p> <ul style="list-style-type: none"> If user inputs don't meet preconditions (e.g., invalid password, duplicate Username, existing Email), the system displays error messages for correction. The user corrects input and resubmits for validation. Process continues when all input is valid. <p>2. Password Reset</p> <ul style="list-style-type: none"> Users can initiate a password reset if forgotten or facing login issues. The system sends a reset link to the user's Email. User clicks the link to set a new password. Users can log in with a new password. <p>3. Confirmation Link Resend:</p> <ul style="list-style-type: none"> Users can request a resend of the confirmation link every 60 seconds. The system sends a new confirmation link to their Email. User clicks the link to confirm Email and activate the account
Exceptions:	<p>1. Email Not Received:</p> <ol style="list-style-type: none"> If the user does not receive the confirmation email, they may request a resend. If the issue persists, the system advises the user to check spam/junk folders or contact support.
Includes:	<p>2. Email & Password Verification</p> <p>3. Username Input</p> <p>4. Display Success/Error Registration</p>
Special Requirements:	<p>5. The system must ensure that passwords are stored</p>

	securely using encryption. 6. The system should handle email delivery promptly and securely.
Assumptions:	7. It is assumed that the user has a valid and accessible email address. 8. It is assumed that the user understands and complies with the password requirements.
Notes and Issues:	

U0102 New User Registration using Google Account

Use Case ID:	U0102		
Use Case Name:	New User Registration using Google Account		
Created By:	Zhao Qixian	Last Updated By:	Mahi Pandey
Date Created:	29/8/2024	Date Last Updated:	04/09/2024

Actor:	New User
Description:	This use case allows new users to register within the EatWellthy system using their existing Google account. By opting for Google-based registration, users can bypass the manual entry of Email, and Password. The system will automatically retrieve the user's Google Email and assign a unique UserID within the EatWellthy database.
Preconditions:	The user must possess an active Google account. The Google account must not already be linked to an existing EatWellthy user account.
Postconditions:	Upon successful registration, the system performs the following actions: 9. The user is assigned a unique UserID within the system. 10. The user is automatically logged in and redirected to the main page of EatWellthy. 11. The user's Google Email and associated UserID are stored in the database. 12. The user is equipped to access the app's features using their Google account credentials.
Priority:	Medium
Frequency of Use:	One-off
Flow of Events:	The new user opens EatWellthy and selects the "Register with Google" option. The system prompts the user to sign in to their Google account.

	<p>The user selects or enters their Google account credentials and authorises access.</p> <p>The system retrieves the user's Google Email and validates it against the EatWellthy database.</p> <ul style="list-style-type: none"> • If the Google Email is already associated with an existing user, an error message is displayed. • If the Google Email is valid and unique, the registration process continues. <p>The system assigns a unique UserID to the new user and stores the Google Email and UserID in the database.</p> <p>The user is redirected to the main page of EatWellthy and is automatically logged in.</p>
Alternative Flows:	<p>13. Google Account Authorization Failure:</p> <ul style="list-style-type: none"> • If the user is unable to authorise their Google account (e.g., incorrect credentials, network issues), the system displays an appropriate error message. • The user may retry authorization or choose another registration method. <p>14. Existing Google Email:</p> <ul style="list-style-type: none"> • If the Google Email is already linked to an EatWellthy account, the system notifies the user and offers the option to log in instead.
Exceptions:	If the Google API is unavailable, the system should inform the user and suggest trying again later or using an alternate registration method.
Includes:	<p>15. Google Account Authorisation</p> <p>16. Username Input</p> <p>17. Display Success/Error Registration</p>
Special Requirements:	The system must adhere to Google's OAuth 2.0 standards for secure authentication and data retrieval.
Assumptions:	It is assumed that the user has a Google account and can access it for registration.
Notes and Issues:	

U0103 User Login via Email

Use Case ID:	U0103		
Use Case Name:	User Login via Email		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	01/09/2024	Date Last Updated:	02/09/2024

Actor:	Registered User
--------	-----------------

Description:	This use case enables registered users to log into the EatWellthy system using their email and password. Successful login grants access to the app's features.
Preconditions:	<ul style="list-style-type: none"> 18. The user must have a valid registered account. 19. The user must provide a valid email and password.
Postconditions:	<ul style="list-style-type: none"> 20. The user is successfully logged into the system. 21. The user's session is authenticated and maintained 22. The user is redirected to the EatWellthy main page
Priority:	High
Frequency of Use:	Every time user wishes to login, Multiple Times
Flow of Events:	<ul style="list-style-type: none"> 23. The user opens the EatWellthy app and selects the Login option. 24. The system presents the login form, requesting Email and Password. 25. The user enters their Email and Password and clicks "Login" 26. The system validates the credentials. <ul style="list-style-type: none"> • If valid, the system logs in the user and redirects them to the main page. • If invalid, the system displays an error message.
Alternative Flows:	<ul style="list-style-type: none"> 27. Invalid Credentials <ul style="list-style-type: none"> • The system informs the user of incorrect Email or Password and prompts re-entry. 28. Password Reset <ul style="list-style-type: none"> • If the user forgets their password, they can select "Forgot Password" to initiate a reset process.
Exceptions:	<ul style="list-style-type: none"> 29. If the system is down or the database is unreachable, an error message is displayed, and the user is asked to try again later.
Includes:	<ul style="list-style-type: none"> 30. Password Verification 31. Display Success/Error Login
Special Requirements:	The system must ensure secure transmission of login credentials.
Assumptions:	It is assumed that the user has registered and verified their email address.
Notes and Issues:	Consider implementing CAPTCHA to avoid bots.

U0104 User Login via Google Account

Use Case ID:	U0104		
Use Case Name:	User Login via Google Account		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Registered User
Description:	This use case allows users to log into the EatWellthy system using their Google Account. Upon successful login, users are granted access to the website's features.
Preconditions:	<ul style="list-style-type: none">32. The user must have a Google account33. The Google account must be linked to an EatWellthy user profile.
Postconditions:	<ul style="list-style-type: none">34. The user is successfully logged into the system.35. The user session is authenticated and maintained.36. The user is redirected to the EatWellthy main page.
Priority:	Medium
Frequency of Use:	Every time user wishes to login, Multiple Times
Flow of Events:	<ul style="list-style-type: none">37. The user selects "Login with Google".38. System redirects the user to Google's login page.39. The user enters their Google credentials and authorises access.40. System retrieves the user's Google Account.<ul style="list-style-type: none">• If valid, the user is logged in and redirected to the main page.• If invalid, an error message is displayed.
Alternative Flows:	<ul style="list-style-type: none">41. Google Authorization Failure<ul style="list-style-type: none">• If the user fails to log into their Google Account, the system displays an error message.• The user can retry or choose another login method42. Account Does Not Exist<ul style="list-style-type: none">• If the user does not have a pre-existing account linked to their Google Account, the system displays an error message.• The user can retry or choose another login method
Exceptions:	<ul style="list-style-type: none">43. If Google API is unavailable, the user is informed and asked to try again later or use email login

	method.
Includes:	44. Google Authorisation 45. Display Success/Error Login
Special Requirements:	System must comply with Google's OAuth 2.0 standards.
Assumptions:	The user has a valid and accessible Google account.
Notes and Issues:	

U0105 Reset Forgotten Password

Use Case ID:	U0105		
Use Case Name:	Reset Forgotten Password		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Registered User
Description:	This use case enables users who have forgotten their password to reset it using their registered email address.
Preconditions:	46. The user must have a registered account with EatWellthy. 47. The user must have access to the registered email address.
Postconditions:	48. The user successfully resets their password. 49. The user is prompted to log in with the new password.
Priority:	Medium
Frequency of Use:	Infrequent, as needed
Flow of Events:	50. The user selects "Forgot Password" on the login page. 51. The system prompts the user to enter their registered email. 52. The user enters their email and clicks "Submit". 53. The system verifies the email and sends a password reset link. 54. The user receives the email and clicks the reset link. 55. The system prompts the user to enter and confirm a new password. 56. The user submits the new password, and the

	system updates their account. 57. The user is prompted to log in with the new password.
Alternative Flows:	58. Invalid Email <ul style="list-style-type: none"> If the entered email is not registered, the system informs the user and prompts for re-entry.
Exceptions:	If the system cannot send the reset email, the user is informed of a delay and asked to try again later.
Includes:	
Special Requirements:	The system must ensure secure handling of password reset requests.
Assumptions:	The user can access their registered email account.
Notes and Issues:	Consider implementing a rate limit for password reset requests to prevent abuse.

U0106 Change Account Password

Use Case ID:	U0106		
Use Case Name:	Change Account Password		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Registered User
Description:	This use case allows registered users to change their account password while logged into the system.
Preconditions:	59. The user must be logged into their account. 60. The user must know their current password.
Postconditions:	61. The user's password is successfully changed. 62. The system logs the user out and prompts them to log in again with the new password.
Priority:	Medium
Frequency of Use:	Infrequent, as needed
Flow of Events:	63. The user navigates to the "Account Settings" page. 64. The system presents the option to change the password. 65. The user enters their current password and the new password, then confirms the new password. 66. The system validates the current password and

	<p>checks the new password against security requirements.</p> <p>67. If valid, the system updates the password and logs the user out.</p> <p>68. The user is prompted to log in with new password.</p>
Alternative Flows:	<p>69. Incorrect Current Password</p> <ul style="list-style-type: none"> If the current password is incorrect, the system informs the user and allows re-entry. <p>70. Password Strength Failure</p> <ul style="list-style-type: none"> If the new password does not meet security requirements, the system prompts the user to enter a stronger password. <p>71. New Password Equivalent to Current Password</p> <ul style="list-style-type: none"> If the new password is the same as the current password, the system prompts the user to enter a different password.
Exceptions:	If the system is unable to update the password due to technical issues, an error message is displayed, and the user is asked to try again later.
Includes:	72. Verify Original Password
Special Requirements:	The system must ensure secure handling and storage of passwords.
Assumptions:	The user has access to their current password.
Notes and Issues:	Provide feedback on the strength of the password.

II. For Functional Requirement #3 (Profile Management)

U0201 View User Profile

Use Case ID:	U0201		
Use Case Name:	View User Profile		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Logged-in & Registered User
Description:	This use case allows users to view their profile information within the EatWellthy app, including personal details, dietary preferences and activity history.

Preconditions:	73. The user must be logged into their EatWellthy account.
Postconditions:	74. The user's profile information is displayed on the screen
Priority:	Low
Frequency of Use:	Occasional
Flow of Events:	<p>75. The user selects "View Profile" from the account menu.</p> <p>76. The system retrieves the user's profile data from the database.</p> <p>77. The system displays the profile information to the user.</p>
Alternative Flows:	NIL
Exceptions:	If the system is unable to retrieve the profile data, an error message is displayed.
Includes:	
Special Requirements:	
Assumptions:	It is assumed that the user has profile information stored in the database.
Notes and Issues:	

U0202 Update User Profile Information

Use Case ID:	U0202		
Use Case Name:	Update User Profile Information		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Logged-in & Registered User
Description:	This use case allows users to update their profile information, such as name, email, dietary preferences, and other personal details within the EatWellthy website.
Preconditions:	78. The user must be logged into their EatWellthy account.
Postconditions:	79. The user's profile information is successfully updated in the database.
Priority:	Medium

Frequency of Use:	Occasionally, as needed
Flow of Events:	<p>80. The user navigates to "Edit Profile" from their account menu.</p> <p>81. The system displays the current profile information in editable fields.</p> <p>82. The user updates the desired fields and clicks "Save".</p> <p>83. The system validates the new information.</p> <ul style="list-style-type: none"> • If valid, the system updates the profile in the database and confirms the update. • If invalid, the system prompts the user to correct the information.
Alternative Flows:	<p>84. Invalid Input:</p> <ul style="list-style-type: none"> • If any new information is invalid (e.g., incorrectly formatted email), the system informs the user and prevents saving until corrected.
Exceptions:	If the system fails to update the profile due to technical issues, an error message is displayed, and the user is asked to try again later.
Includes:	85. Change Confirmation
Special Requirements:	The system must ensure that updates are secure and reflect immediately across the platform.
Assumptions:	It is assumed that the user has valid profile information that they want updated.
Notes and Issues:	

U0203 Manage Dietary Preferences

Use Case ID:	U0203		
Use Case Name:	Manage Dietary Preferences		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Logged-in & registered user
Description:	This use case allows users to specify or update their dietary preferences within the EatWellthy app, which will be used to tailor meal recommendations.
Preconditions:	86. The user must be logged into their EatWellthy account.

Postconditions:	87. The user's dietary preferences are successfully updated in the system.
Priority:	High
Frequency of Use:	Occasional, as needed
Flow of Events:	<p>88. The user navigates to "Dietary Preferences" from their profile settings.</p> <p>89. The system displays the current dietary preferences.</p> <p>90. The user selects or modifies their dietary preferences and clicks "Save".</p> <p>91. The system updates the preferences in the database and confirms the update.</p>
Alternative Flows:	<p>92. Invalid Preference</p> <ul style="list-style-type: none"> If any preference selection is invalid, the system prompts the user to correct it before saving.
Exceptions:	If the system is unable to update dietary preferences due to technical issues, an error message is displayed, and the user is asked to try again later.
Includes:	93. Change Confirmation
Special Requirements:	
Assumptions:	It is assumed that the user has dietary preferences that they wish to manage.
Notes and Issues:	

U0204 Update Security Settings

Use Case ID:	U0204		
Use Case Name:	Update Security Settings		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Logged-in User
Description:	This use case allows users to update their security settings within the EatWellthy app, such as enabling two-factor authentication or changing security questions.
Preconditions:	94. The user must be logged into their EatWellthy account.

Postconditions:	95. The user's security settings are successfully updated and take effect immediately.
Priority:	Low
Frequency of Use:	Occasional, as needed
Flow of Events:	<p>96. The user navigates to "Security Settings" from their account menu.</p> <p>97. The system displays the current security settings.</p> <p>98. The user updates the desired security settings (e.g., enabling two-factor authentication).</p> <p>99. The system validates the changes and updates the settings in the database.</p> <p>100. The system confirms the update and informs the user of the changes.</p>
Alternative Flows:	<p>101. Invalid Security Settings</p> <ul style="list-style-type: none"> If any security setting is invalid or fails validation (e.g., weak security question answer), the system informs the user and prevents saving until corrected.
Exceptions:	If the system fails to update the security settings due to technical issues, an error message is displayed, and the user is asked to try again later.
Includes:	102. Verify Password
Special Requirements:	The system must ensure that security updates are handled with the highest level of protection.
Assumptions:	It is assumed that the user has security settings that they wish to manage.
Notes and Issues:	Can consider implementing an email alert system to notify users if their security information is updated.

U0205 Delete User Account

Use Case ID:	U0205		
Use Case Name:	Delete User Account		
Created By:	Mahi Pandey	Last Updated By:	Mahi Pandey
Date Created:	02/09/2024	Date Last Updated:	02/09/2024

Actor:	Logged-in & Registered User
Description:	This use case allows users to permanently delete their EatWellthy account. Deleting an account removes all user data from the system and makes the account irretrievable.

Preconditions:	<p>103. The user must be logged into their EatWellthy account.</p> <p>104. The user must confirm their intention to permanently delete the account.</p> <p>105. The system must inform the user of the consequences of deletion, including the irretrievable loss of data.</p>
Postconditions:	<p>106. The user's account and all associated data are permanently removed from the system.</p> <p>107. The user is logged out and cannot recover the account or data.</p>
Priority:	Medium
Frequency of Use:	Infrequent, as needed
Flow of Events:	<p>108. The user navigates to "Account Settings" and selects the "Delete Account" option.</p> <p>109. The system prompts the user to confirm their decision to permanently delete the account.</p> <p>110. The user confirms the deletion by providing their password for security verification.</p> <p>111. The system warns the user about the permanent nature of account deletion and the loss of all data.</p> <p>112. The user confirms the deletion.</p> <p>113. The system permanently removes the user's account and all associated data from the database.</p> <p>114. The user is logged out and redirected to the homepage with a confirmation of account deletion.</p>
Alternative Flows:	<p>115. Cancellation of Deletion</p> <ul style="list-style-type: none"> • If the user decides not to delete the account after being prompted for confirmation, they can cancel the process, and no changes are made to the account. <p>116. Data Export Option</p> <ul style="list-style-type: none"> • Before confirming deletion, the system offers the user the option to export their data. If the user chooses to export, the system generates a downloadable file with the user's data.
Exceptions:	If the system fails to delete the account due to technical issues, an error message is displayed, and the user is advised to try again later or contact support.
Includes:	

Special Requirements:	<p>117. The system must ensure that the deletion process is secure and irreversible.</p> <p>118. The system should comply with relevant data protection regulations to avoid legal disputes.</p>
Assumptions:	It is assumed that the user understands the implications of account deletion, including the permanent loss of access and data.
Notes and Issues:	

III. For Functional Requirement #4 (Nutrition Calculation)

U0301 Log Daily Meal

Use Case ID:	U0301		
Use Case Name:	Log Daily Meal		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	1/9/2024	Date Last Updated:	1/9/2024

Actor:	Current User
Description:	This use case enables users to log their daily meals into the EatWellthy system. Users can input details such as meal type (breakfast, lunch, dinner, snacks), food items, portion sizes, and time of consumption. The system will store this information.
Preconditions:	<ol style="list-style-type: none"> The user must be logged into the EatWellthy system with a valid account. The system must have access to the user's profile to store meal logs.
Postconditions:	<ul style="list-style-type: none"> The meal information is successfully stored in the user's profile in the database. The system updates the user's nutritional summary for the day, reflecting the logged meal. Users can view, edit, or delete the meal entry at any time. The system may provide feedback or suggestions based on the nutritional content of the logged meals.
Priority:	High
Frequency of Use:	Daily
Flow of Events:	<p>119. The user accesses the EatWellthy application and navigates to the "Log Daily Meal" section.</p>

	<p>120. The system presents the user with a form to input meal details:</p> <ul style="list-style-type: none"> • Meal type (e.g., breakfast, lunch, dinner, snacks) • Food items consumed • Portion sizes • Time of consumption <p>121. The user fills in the necessary information and clicks "Submit."</p> <p>122. The system validates the input:</p> <ul style="list-style-type: none"> • Ensures that all required fields are filled. • Verifies that the portion sizes are reasonable (e.g., not negative). <p>123. If the input is valid, the system stores the meal information in the user's profile.</p> <p>124. The system updates the user's daily nutritional summary with the new data.</p> <p>125. The system displays a confirmation message to the user, indicating that the meal has been successfully logged.</p>
Alternative Flows:	<p>1. Validation Errors:</p> <ul style="list-style-type: none"> • If any required fields are missing or contain invalid data, the system displays an error message. • The user corrects the input and resubmits the form. • The process continues once the input is valid. <p>2. Editing a Meal Log:</p> <ul style="list-style-type: none"> • The user can navigate to their meal history and select a previously logged meal. • The system displays the meal details, allowing the user to make changes. • The user submits the changes, and the system updates the meal information in the database and adjusts the nutritional summary accordingly. <p>3. Deleting a Meal Log:</p> <ul style="list-style-type: none"> • The user selects a meal from their meal history that they wish to delete. • The system prompts the user to confirm the deletion. • Upon confirmation, the system removes the meal from the user's profile and adjusts the

	nutritional summary
Exceptions:	<p>Network Failure:</p> <ul style="list-style-type: none"> If there is a network issue, the system will prompt the user to retry or save the meal log locally until the network is restored. <p>System Unavailability:</p> <ul style="list-style-type: none"> If the system is down for maintenance or encounters an error, the user will receive a notification and will be unable to log meals until the issue is resolved.
Includes:	Confirm message
Special Requirements:	
Assumptions:	
Notes and Issues:	

U0302 Edit Logged Meal

Use Case ID:	U302		
Use Case Name:	Edit Logged Meal		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	1/9/2024	Date Last Updated:	1/9/2024

Actor:	Current User
Description:	This use case allows users to edit a previously logged meal in the EatWellthy system. Users can modify details such as meal type, food items, portion sizes, and time of consumption. The system updates the user's nutritional summary to reflect the changes.
Preconditions:	<p>126. The user must be logged into the EatWellthy system with a valid account.</p> <p>127. The user must have previously logged a meal that they wish to edit.</p> <p>128. The system must have access to the user's meal history.</p>
Postconditions:	<p>129. The updated meal information is successfully stored in the user's profile in the</p>

	<p>database.</p> <p>130. The system recalculates and updates the user's daily nutritional summary based on the changes.</p> <p>131. The user can view the updated meal entry and the adjusted nutritional information.</p>
Priority:	
Frequency of Use:	As needed
Flow of Events:	<p>132. The user accesses the EatWellthy application and navigates to their meal history.</p> <p>133. The system presents a list of previously logged meals, organised by date and time.</p> <p>134. The user selects the meal they wish to edit.</p> <p>135. The system displays the meal details, including:</p> <ul style="list-style-type: none"> • Meal type (e.g., breakfast, lunch, dinner, snacks) • Food items consumed • Portion sizes • Time of consumption <p>136. The user makes the necessary changes to the meal details.</p> <p>137. The user clicks "Save" or "Update" to submit the changes.</p> <p>138. The system validates the new input:</p> <ul style="list-style-type: none"> • Ensures all required fields are filled. • Verifies that the portion sizes are reasonable (e.g., not negative). <p>139. If the input is valid, the system updates the meal information in the user's profile.</p> <p>140. The system recalculates the user's daily nutritional summary to reflect the updated meal data.</p> <p>141. The system displays a confirmation message, indicating that the meal has been successfully updated.</p>
Alternative Flows:	<p>142. Validation Errors:</p> <ul style="list-style-type: none"> • If any required fields are missing or contain invalid data, the system displays an error message. • The user corrects the input and resubmits the form. • The process continues once the input is valid. <p>143. Cancelling an Edit:</p> <ul style="list-style-type: none"> • The user may choose to cancel the edit at any time before saving.

	<ul style="list-style-type: none"> The system discards any changes made and returns to the meal history view without altering the original meal log. <p>144. Undoing Changes:</p> <ul style="list-style-type: none"> After saving an edit, the user may realize they made a mistake.
Exceptions:	<p>145. Network Failure:</p> <ul style="list-style-type: none"> If there is a network issue, the system will prompt the user to retry or save the meal log locally until the network is restored. <p>146. System Unavailability:</p> <ul style="list-style-type: none"> If the system is down for maintenance or encounters an error, the user will receive a notification and will be unable to log meals until the issue is resolved.
Includes:	Confirmation message
Special Requirements:	
Assumptions:	
Notes and Issues:	

U0303 Delete Logged Meal

Use Case ID:	U303		
Use Case Name:	Delete Logged Meal		
Created By:	LiuXiaotao	Last Updated By:	Liu Xiaotao
Date Created:	1/9/2024	Date Last Updated:	1/9/2024

Actor:	Current User
Description:	This use case allows users to delete a previously logged meal from the EatWellthy system. Once deleted, the meal information is removed from the user's profile, and the system updates the user's daily nutritional summary to reflect the deletion.
Preconditions:	<p>147. The user must be logged into the EatWellthy system with a valid account.</p> <p>148. The user must have at least one previously logged meal in their history.</p> <p>149. The system must have access to the user's meal history.</p>
Postconditions:	150. The selected meal is permanently

	<p>removed from the user's profile in the database.</p> <p>151. The system recalculates and updates the user's daily nutritional summary based on the deletion.</p> <p>152. The user can view their updated meal history and nutritional information.</p>
Priority:	
Frequency of Use:	As needed
Flow of Events:	<p>153. The user accesses the EatWellthy application and navigates to their meal history.</p> <p>154. The system presents a list of previously logged meals, organized by date and time.</p> <p>155. The user selects the meal they wish to delete.</p> <p>156. The system displays the meal details, including:</p> <ul style="list-style-type: none"> • Meal type (e.g., breakfast, lunch, dinner, snacks) • Food items consumed • Portion sizes • Time of consumption <p>157. The user clicks the "Delete" button to remove the selected meal.</p> <p>158. The system prompts the user to confirm the deletion to prevent accidental removal.</p> <p>159. The user confirms the deletion by selecting "Yes" or "Confirm".</p> <p>160. The system validates the request and proceeds to delete the meal from the user's profile.</p> <p>161. The system recalculates the user's daily nutritional summary to reflect the removal of the meal data.</p> <p>162. The system displays a confirmation message, indicating that the meal has been successfully deleted.</p> <p>163. The user's meal history is updated to exclude the deleted meal.</p>
Alternative Flows:	<p>164. Cancelling the Deletion:</p> <ul style="list-style-type: none"> • If the user decides not to delete the meal, they can select "No" or "Cancel" at the confirmation prompt. • The system aborts the deletion process and returns to the meal history view with no changes made. <p>165. Restoring a Deleted Meal:</p> <ul style="list-style-type: none"> • If the user accidentally deletes a meal, they may have the option to restore it within a limited time

	<p>(e.g., 5 minutes) by selecting an "Undo" option.</p> <ul style="list-style-type: none"> The system restores the deleted meal details and updates the nutritional summary accordingly. <p>166. Multiple Deletions:</p> <ul style="list-style-type: none"> The user can select and delete multiple meals at once. The system provides a batch delete option, and the user confirms the deletion of all selected meals. The system then updates the nutritional summary based on the collective deletion.
Exceptions:	<p>167. Network Failure:</p> <ul style="list-style-type: none"> If there is a network issue, the system will prompt the user to retry or save the meal log locally until the network is restored. <p>168. System Unavailability:</p> <ul style="list-style-type: none"> If the system is down for maintenance or encounters an error, the user will receive a notification and will be unable to log meals until the issue is resolved.
Includes:	
Special Requirements:	The system should ask for confirmation before the data is deleted
Assumptions:	
Notes and Issues:	

U0304 View Meal History

Use Case ID:	U304		
Use Case Name:	View Meal History		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	1/9/2024	Date Last Updated:	1/9/2024

Actor:	Current User
Description:	This use case allows users to view their logged meal history within the EatWellthy system. Users can access details of past meals, including the type of meal, food items consumed, portion sizes, and nutritional information.
Preconditions:	<p>169. The user must be logged into the EatWellthy system with a valid account.</p> <p>170. The user must have logged at least one</p>

	<p>meal in the system.</p> <p>171. The system must have access to the user's meal history data.</p>
Postconditions:	<p>172. The user must be logged into the EatWellthy system with a valid account.</p> <p>173. The user must have logged at least one meal in the system.</p> <p>174. The system must have access to the user's meal history data.</p>
Priority:	
Frequency of Use:	As needed
Flow of Events:	<p>175. The user accesses the EatWellthy application and navigates to the "Meal History" section.</p> <p>176. The system retrieves the user's logged meal data from the database.</p> <p>177. The system presents the meal history in a chronological list, displaying the following for each meal:</p> <ul style="list-style-type: none"> • Date and time of the meal • Meal type (e.g., breakfast, lunch, dinner, snacks) • Summary of food items consumed • Total calories and key nutritional information (e.g., protein, carbs, fats) <p>178. The user selects a specific meal entry to view more detailed information.</p> <p>179. The system displays the full details of the selected meal, including:</p> <ul style="list-style-type: none"> • Individual food items with portion sizes • Detailed nutritional information for each food item (e.g., calories, macronutrients) • Any notes or additional information entered by the user <p>180. The user can navigate back to the overall meal history or select another meal to view.</p> <p>181. The user can use filtering options (e.g., by date, meal type) to refine the meal history display.</p> <p>182. The system allows the user to search for specific meals by keywords or dates.</p> <p>183. The user can exit the meal history section and return to the main dashboard or another section of the application.</p>
Alternative Flows:	<p>184. No Logged Meals:</p>

	<ul style="list-style-type: none"> • If the user has not logged any meals, the system displays a message indicating that no meal history is available. • The user is prompted to log their first meal. <p>185. Filtering and Sorting:</p> <ul style="list-style-type: none"> • The user can apply filters to view specific types of meals (e.g., only breakfasts) or sort the history by criteria such as date, calories, or specific nutrients. • The system adjusts the display to show only the filtered or sorted meals. <p>186. Searching Meal History:</p> <ul style="list-style-type: none"> • The user enters a search term (e.g., "salad," "chicken") in the search bar. • The system displays meals that match the search criteria, highlighting the relevant entries.
Exceptions:	<p>187. Data Retrieval Failure:</p> <ul style="list-style-type: none"> • If the system fails to retrieve meal history due to a network or server issue, the user is notified with an error message. • The system may offer an option to retry or advise the user to check back later. <p>188. Incomplete Data:</p> <ul style="list-style-type: none"> • If any part of the meal history data is incomplete or corrupted, the system displays a warning and attempts to recover or display what is available. • The user is informed about the incomplete data and is given options to refresh or contact support.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	Avoid the situation that the system loaded the data of another user

U0305 Generate Personalized Diet Plan

Use Case ID:	U305		
Use Case Name:	Generate Personalised Diet Plan		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	2/9/2024	Date Last Updated:	2/9/2024

Actor:	Current User
Description:	This use case allows users to generate a personalized diet plan tailored to their individual health goals, dietary preferences, and nutritional needs. The system takes into

	account various user inputs, such as age, gender, weight, height, activity level, and specific dietary restrictions or preferences, to create a customized diet plan.
Preconditions:	<p>189. The user must be logged into the EatWellthy application with a valid account.</p> <p>190. The user must provide relevant health and dietary information within the application.</p> <p>191. The system must have access to a comprehensive nutrition database to generate accurate diet plans.</p>
Postconditions:	<p>192. The user receives a personalized diet plan that aligns with their health goals and dietary preferences.</p> <p>193. The generated diet plan is stored in the user's profile for easy access and future reference.</p> <p>194. The user can follow the diet plan and track their progress within the EatWellthy application.</p>
Priority:	High
Frequency of Use:	Daily
Flow of Events:	<p>195. The user logs into the EatWellthy application and navigates to the "Generate Diet Plan" section.</p> <p>196. The system prompts the user to provide or update the following information:</p> <ul style="list-style-type: none"> • Age, Gender, Weight, Height • Activity level (e.g., sedentary, lightly active, moderately active, very active) • Health goals (e.g., weight loss, muscle gain, maintenance) • Dietary preferences (e.g., vegetarian, vegan, low-carb, gluten-free) • Any food allergies or restrictions <p>197. The user enters the required information and confirms their input.</p> <p>198. The system validates the user's input for completeness and accuracy. If any required information is missing or incorrect, the system prompts the user to correct it.</p> <p>199. Upon successful validation, the system accesses its nutrition database and calculate the user's daily caloric needs and macronutrient distribution (e.g., protein, fats, carbohydrates).</p> <p>200. The system generates a personalized diet plan, including:</p> <ul style="list-style-type: none"> • Suggested daily calorie intake • Recommended meals and snacks

	<ul style="list-style-type: none"> • Portion sizes for each meal • Nutritional breakdown (e.g., calories, macronutrients, vitamins, minerals) for each meal • Meal timing recommendations (e.g., breakfast, lunch, dinner, snacks) <p>201. The system displays the personalized diet plan to user, allowing them to review and confirm the plan.</p> <p>202. The user can either accept the diet plan as-is or make adjustments (e.g., swap meals, modify portion sizes).</p> <p>203. Once satisfied, the user saves the personalized diet plan to their profile.</p> <p>204. The system stores the diet plan in the user's profile and makes it accessible from the dashboard for daily reference and meal logging.</p>
Alternative Flows:	<p>205. Plan Regeneration</p> <ul style="list-style-type: none"> • If the user is not satisfied with the initial diet plan, they can regenerate the plan with adjusted inputs (e.g., changing the health goal or activity level). • The system generates a new plan based on the updated inputs.
Exceptions:	<p>206. Database Access Issues:</p> <ul style="list-style-type: none"> • If the system cannot access the nutrition database (e.g., due to a network issue), it displays an error message and suggests the user try again later. • The user is informed that their current data will be saved and can be used to generate the diet plan once the issue is resolved. <p>207. Inconsistent or Conflicting Inputs:</p> <ul style="list-style-type: none"> • If the user's inputs are inconsistent or conflict with each other (e.g., selecting both high-carb and low-carb preferences), the system alerts the user and requests clarification. • The user can modify their inputs to resolve the conflict and proceed with generating the diet plan.
Includes:	
Special Requirements:	User should be able to copy the generated plan and paste it in their personal diet site
Assumptions:	
Notes and Issues:	

U0306 Sync Diet Plan with Google Calendar

Use Case ID:	U306		
Use Case Name:	Sync Diet Plan with Google Calendar		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	1/9/2024	Date Last Updated:	1/9/2024

Actor:	Current User
Description:	This use case allows users to sync their diet plan from EatWellthy with their Google Calendar. By doing this, users can have meal reminders, diet plan events, and nutritional goals automatically appear in their Google Calendar.
Preconditions:	<ul style="list-style-type: none">208. The user must be logged into the EatWellthy system with a valid account.209. The user must have a diet plan created within the EatWellthy application.210. The user must have an active Google account and must be signed into Google services.211. The user must authorise the EatWellthy app to access and manage their Google Calendar.
Postconditions:	<ul style="list-style-type: none">212. The user's diet plan is successfully synced with their Google Calendar.213. The system automatically creates calendar events for each meal or diet-related activity, based on the user's diet plan.214. Users receive notifications on their devices as per the scheduled times in the Google Calendar.
Priority:	
Frequency of Use:	Frequently (every time the plan is generated)
Flow of Events:	<ul style="list-style-type: none">215. The user logs into the EatWellthy application and navigates to the "Diet Plan" section.216. The user selects the option to sync their diet plan with Google Calendar.217. The system prompts the user to sign in to their Google account, if they haven't already.218. The system requests the necessary permissions to access and manage the user's Google Calendar.219. The user grants permission, allowing the EatWellthy app to access their Google Calendar.220. The system retrieves the user's diet plan and begins syncing the data with the Google

	<p>Calendar.</p> <p>221. For each meal or diet-related event in plan, system:</p> <ul style="list-style-type: none"> • Creates a corresponding event in Google Calendar. • Sets the event time and date based on the meal schedule. • Includes details in the event description, such as the meal type, food items, and nutritional goals. <p>222. The system confirms the successful sync and notifies the user that their diet plan is now available in Google Calendar.</p> <p>223. The user can now view, edit, or receive reminders for their diet plan directly from Google Calendar.</p>
Alternative Flows:	<p>224. Authorization Failure:</p> <ul style="list-style-type: none"> • If the user denies the authorization request, the system displays a message explaining that syncing cannot proceed without the necessary permissions. • The user is given the option to retry the authorization process. <p>225. Sync Failure:</p> <ul style="list-style-type: none"> • If the system encounters an issue while syncing (e.g., network issues, Google API errors), the system displays an error message. • The user is given options to retry the sync or to troubleshoot the issue. <p>226. Partial Sync:</p> <ul style="list-style-type: none"> • If only part of the diet plan syncs successfully (e.g., some events were not created), the system notifies the user of the partial success. • The system provides details on which events were not synced and offers options to retry syncing only those events.
Exceptions:	<p>227. Network Issues:</p> <ul style="list-style-type: none"> • If there is a network connectivity problem during the sync, the system alerts the user and pauses the syncing process until the connection is restored. • The user can manually retry the sync after resolving any network issues. <p>228. Google Calendar Quota Limits:</p> <ul style="list-style-type: none"> • If the user's Google Calendar has reached its event creation limit, the system notifies the user and cancels the sync process.

	<ul style="list-style-type: none"> The user is advised to clear up space in their Google Calendar or contact Google support for further assistance. <p>229. Calendar Event Conflicts:</p> <ul style="list-style-type: none"> If there are conflicting events in the Google Calendar during the sync (e.g., overlapping times with existing events), the system warns the user. The user can choose to skip the conflicting events or manually resolve the conflicts in Google Calendar.
Includes:	
Special Requirements:	The system should access and interact with Google calendar successfully.
Assumptions:	We assume that the user have a Google account and the system is able to access it.
Notes and Issues:	

IV. For Functional Requirement #8 (Welloh Chatbot)

U0401 Ask Nutritional Questions

Use Case ID:	U401		
Use Case Name:	Ask Nutritional Questions		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	2/9/2024	Date Last Updated:	2/9/2024

Actor:	Current User
Description:	This use case allows users to interact with the Welloh ChatBot to ask various nutritional questions. The Welloh ChatBot is powered by GPT-based technology, fine-tuned to provide accurate, relevant, and personalised nutritional information. Users can inquire about dietary advice, nutritional content of foods, meal suggestions, and other related topics. The chatbot is designed to deliver responses quickly and efficiently.
Preconditions:	<p>230. The user must be logged into the EatWellthy application.</p> <p>231. The Welloh ChatBot must be connected to the internet and have access to the latest nutritional databases and GPT-based model.</p> <p>232. The ChatBot's underlying GPT model must be optimised for performance, ensuring quick and relevant responses.</p>

Postconditions:	<p>233. The user receives an accurate and relevant response to their nutritional question</p> <p>234. The system logs the interaction for future reference and potential improvement of the ChatBot's responses.</p> <p>235. The user can save the response or add suggested meals directly to their meal plan within the app.</p>
Priority:	High
Frequency of Use:	As needed
Flow of Events:	<p>236. The user opens the EatWellthy application and navigates to the "Welloh ChatBot" feature.</p> <p>237. The user types a nutritional question into the ChatBot interface. Examples of questions include:</p> <ul style="list-style-type: none"> • "What are the health benefits of eating avocados?" • "How many calories are in a medium-sized apple?" • "Can you suggest a high-protein breakfast?" <p>238. The system receives the user's input and processes the question through the Welloh ChatBot's GPT-based model.</p> <p>239. The ChatBot analyzes the question, retrieving relevant information from its nutritional database and applying natural language processing to generate a clear and concise response.</p> <p>240. The system displays the response to the user within 2 seconds of receiving the question. The response may include:</p> <ul style="list-style-type: none"> • Nutritional information (e.g., calorie content, macronutrient breakdown) • Health benefits and potential risks of certain foods • Personalized meal suggestions based on the user's profile and dietary preferences <p>241. The user reviews the response and can either ask follow-up questions, save the response, or incorporate suggestions into their meal plan.</p> <p>242. The system logs the interaction for future reference, potentially improving future responses through machine learning.</p>
Alternative Flows:	<p>243. Unavailable Information:</p> <ul style="list-style-type: none"> • If the ChatBot cannot find relevant data for the

	<p>user's query, it informs the user and offers to search for related information or suggest alternative questions.</p> <ul style="list-style-type: none"> The user can either modify their question or accept the ChatBot's suggestions. <p>244. Technical Issues:</p> <ul style="list-style-type: none"> If the ChatBot experiences technical issues (e.g., connectivity problems, server downtime), it informs the user and suggests trying again later. The system may offer offline alternatives, such as browsing the FAQ section or accessing preloaded nutritional guides.
Exceptions:	<p>245. Excessive Response Time:</p> <ul style="list-style-type: none"> If the ChatBot takes longer than 2 seconds to generate a response, the system displays a message indicating that the response is delayed. The user is given the option to wait or rephrase the question for a potentially faster response.
Includes:	
Special Requirements:	
Assumptions:	It assume the system can have a stable GPT service
Notes and Issues:	May need some prompt techniques for an accurate response.

U0402 Receive Nutritional Alerts

Use Case ID:	U402		
Use Case Name:	Receive Nutritional Alerts		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	2/9/2024	Date Last Updated:	2/9/2024

Actor:	Current User
Description:	This use case allows users to receive alerts and notifications related to their nutritional intake. These alerts may include reminders to log meals, notifications of nutritional goals being met or missed, and suggestions for adjustments based on their diet plan.
Preconditions:	<p>246. The user logs into the EatWellthy application and navigates to the settings menu.</p> <p>247. The user ensures that notifications are enabled and that the necessary permissions are granted for the app to send alerts to their device.</p> <p>248. The user sets their preferences for receiving alerts, such as frequency (e.g., daily,</p>

	<p>weekly), type of alerts (e.g., meal reminders, goal tracking), and preferred times for receiving notifications.</p> <p>249. As the user logs meals or updates their diet plan, the system continuously monitors their nutritional intake and progress towards their goals.</p> <p>250. When the system detects a condition that triggers an alert (e.g., a meal is missed, a nutritional goal is met or missed), it generates a notification.</p> <p>251. The notification includes relevant information, such as:</p> <ul style="list-style-type: none"> • A reminder to log a meal if it hasn't been recorded within a certain time frame. • An alert that a nutritional goal has been met or is at risk of being missed, with suggestions on how to stay on track. • Tips or suggestions for adjusting the diet to improve nutritional balance. <p>252. The system sends the notification to the user's device, ensuring it arrives at the preferred time set by the user.</p> <p>253. The user receives the alert on their device, reviews the information, and takes appropriate action, such as logging a meal or adjusting their diet plan.</p>
Postconditions:	<p>254. Notification Settings Disabled:</p> <ul style="list-style-type: none"> • If the user has not enabled notifications, the system displays a message within the app, encouraging the user to enable notifications to receive important alerts. • The user can navigate to the settings to enable notifications if desired. <p>255. Missed Alerts:</p> <ul style="list-style-type: none"> • If the user misses an alert (e.g., due to being offline), the system logs the alert and resends it when the user is back online or next opens the app. • The user can view missed alerts in a dedicated "Notification History" section within the app. <p>256. Custom Alert Frequency:</p> <ul style="list-style-type: none"> • If the user sets a custom frequency for receiving alerts (e.g., only once a day), the system respects this setting and consolidates multiple alerts into a single notification. • The user receives a summary alert at their preferred time, covering all relevant information.

Priority:	Middle
Frequency of Use:	As needed
Flow of Events:	<p>257. The user logs into the EatWellthy application and navigates to the settings menu.</p> <p>258. The user ensures that notifications are enabled and that the necessary permissions are granted for the app to send alerts to their device.</p> <p>259. The user sets their preferences for receiving alerts, such as frequency (e.g., daily, weekly), type of alerts (e.g., meal reminders, goal tracking), and preferred times for receiving notifications.</p> <p>260. As the user logs meals or updates their diet plan, the system continuously monitors their nutritional intake and progress towards their goals.</p> <p>261. When the system detects a condition that triggers an alert (e.g., a meal is missed, a nutritional goal is met or missed), it generates a notification.</p> <p>262. The notification includes relevant information, such as:</p> <ul style="list-style-type: none"> • A reminder to log a meal if it hasn't been recorded within a certain time frame. • An alert that a nutritional goal has been met or is at risk of being missed, with suggestions on how to stay on track. • Tips or suggestions for adjusting the diet to improve nutritional balance. <p>263. The system sends the notification to the user's device, ensuring it arrives at the preferred time set by the user.</p> <p>264. The user receives the alert on their device, reviews the information, and takes appropriate action, such as logging a meal or adjusting their diet plan.</p>
Alternative Flows:	<p>265. Notification Settings Disabled:</p> <ul style="list-style-type: none"> • If the user has not enabled notifications, the system displays a message within the app, encouraging the user to enable notifications to receive important alerts. • The user can navigate to the settings to enable notifications if desired. <p>266. Missed Alerts:</p> <ul style="list-style-type: none"> • If the user misses an alert (e.g., due to being offline), the system logs the alert and resends it when the user is back online or next opens the app.

	<ul style="list-style-type: none"> The user can view missed alerts in a dedicated "Notification History" section within the app. <p>267. Custom Alert Frequency:</p> <ul style="list-style-type: none"> If the user sets a custom frequency for receiving alerts (e.g., only once a day), the system respects this setting and consolidates multiple alerts into a single notification. The user receives a summary alert at their preferred time, covering all relevant information
Exceptions:	<p>268. Device Notification Issues:</p> <ul style="list-style-type: none"> If the device is unable to receive notifications due to technical issues (e.g., Do Not Disturb mode, network problems), the system logs the alert and attempts to resend it once the issue is resolved. The user can view a history of missed alerts within the app. <p>269. Nutritional Data Sync Delays:</p> <ul style="list-style-type: none"> If there is a delay in syncing the user's nutritional data (e.g., due to a slow network), the system may send alerts later than expected. The system prioritizes timely delivery and informs the user if there is a delay in processing their nutritional data. <p>270. Inaccurate Alerts:</p> <ul style="list-style-type: none"> If an alert is generated based on incorrect or incomplete data (e.g., user input errors), the system allows the user to dismiss or correct the alert. The system may prompt the user to review their logged meals or update their diet plan to ensure accuracy.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	The alert should consider the users' feeling i.e. a gentle expression is needed

U0403 Query Nutritional Information via Chatbot

Use Case ID:	U403		
Use Case Name:	Query Nutritional Information via Chatbot		
Created By:	Liu Xiaotao	Last Updated By:	Liu Xiaotao
Date Created:	2/9/2024	Date Last Updated:	2/9/2024

Actor:	Current User
Description:	This use case allows users to query specific nutritional information via the Welloh ChatBot. The ChatBot, powered by a GPT-based model optimised for enhanced performance, provides detailed nutritional data on various foods, ingredients, and meals. Users can inquire about calorie counts, macronutrient breakdowns, vitamin content, and more, receiving tailored responses based on their dietary preferences and needs.
Preconditions:	<p>271. The user must be logged into the EatWellthy application.</p> <p>272. The Welloh ChatBot must be operational, with access to the latest nutritional databases and the GPT-based model.</p> <p>273. The ChatBot's system must be up-to-date, ensuring that the nutritional information provided is accurate and reliable.</p>
Postconditions:	<p>274. The user receives precise nutritional information based on their query.</p> <p>275. The system logs the interaction for future reference, helping to improve the accuracy and relevance of future responses.</p> <p>276. The user has the option to save the queried information or add related items to their meal plan.</p>
Priority:	Medium
Frequency of Use:	As needed
Flow of Events:	<p>277. The user opens the EatWellthy application and navigates to the "Welloh ChatBot" feature.</p> <p>278. The user types a specific query regarding nutritional information into the ChatBot interface. Examples of queries include:</p> <ul style="list-style-type: none"> • "How many grams of protein are in a chicken breast?" • "What is the fat content of 100 grams of almonds?" • "How much vitamin C is in an orange?" <p>279. The system receives the user's query and processes it through the Welloh ChatBot's GPT-based model.</p> <p>280. The ChatBot analyzes the query, retrieves relevant nutritional data from its database, and generates a response.</p> <p>281. The system displays the nutritional information to the user within 2 seconds of receiving the query. The response may include:</p> <ul style="list-style-type: none"> • Calorie content

	<ul style="list-style-type: none"> • Macronutrient breakdown (proteins, fats, carbohydrates) • Micronutrient details (vitamins, minerals) • Additional health-related information or advice <p>282. The user reviews the information and can choose to:</p> <ul style="list-style-type: none"> • Save the data for future reference • Add the item to their daily meal log • Ask follow-up questions for more details <p>283. The system logs the interaction for future analysis and improvement of the ChatBot's responses.</p>
Alternative Flows:	<p>284. Unavailable Information:</p> <ul style="list-style-type: none"> • If the ChatBot cannot find specific nutritional data for the user's query, it informs the user and offers alternatives (e.g., similar foods or items). • The user can choose to refine their query or select from the suggested alternatives.
Exceptions:	<p>285. Excessive Response Time:</p> <ul style="list-style-type: none"> • If the ChatBot takes longer than 2 seconds to generate a response, the system displays a message indicating that the response is delayed. • The user is given the option to wait or rephrase the question for a potentially faster response.
Includes:	
Special Requirements:	
Assumptions:	It assume the system can have a stable GPT service
Notes and Issues:	May need some prompt techniques for an accurate response.

V. For Functional Requirement # (Nutrition Reference Sheet)

U0501 Search for Food Items

Use Case ID:	U0501		
Use Case Name:	Search for Food Items		
Created By:	Mehta Rishika	Last Updated By:	Mehta Rishika
Date Created:	29/8/2024	Date Last Updated:	29/8/2024

Actor:	Logged in User
Description:	This use case allows users who are logged into EatWellthy to be able to search for food items and their nutritional and caloric value. This can be done by passing the data of the meal plan into the EatWellthy system. The system will retrieve the desired information from the food database.
Preconditions:	<p>286. The user must be logged in.</p> <p>287. The user must provide valid input.</p>
Postconditions:	<p>288. Upon successful request, the system performs the following actions:</p> <ul style="list-style-type: none"> The system retrieves the data from the food database The system displays the caloric intake of the user based on the data it retrieves. The user is equipped to access the app's features using their Google account credentials.
Priority:	
Frequency of Use:	
Flow of Events:	<p>289. The user will log in to EatWellthy using their credentials</p> <p>290. The user clicks on "Nutrition Calculator"</p> <p>291. The system displays a page where the user can enter the meal they want to check the caloric value for.</p> <ul style="list-style-type: none"> If the user enters an invalid input, the system displays the message: "Input is invalid, please enter a correct value." and the user is redirected to "Nutrition Calculator" again. If the user inputs a valid input, the system fetches the necessary data from the database and gives the nutritional and caloric value of the given meal. <p>292. The system displays the nutritional and caloric value.</p>

	293. The user can check the nutritional content of other food items.
Alternative Flows:	294. Valid Input But No Data Found If the meal data entered is valid but not found in the database, the system displays: "No data available for the entered meal. Please try another meal."
Exceptions:	295. Database Unavailable <ul style="list-style-type: none"> If the food database is down or unreachable, the system displays: "Service is currently unavailable. Please try again later."
Includes:	
Special Requirements:	
Assumptions:	It is assumed that User has already registered and possesses an account to access the features of the website. The food database is up-to-date and contains accurate nutritional information.
Notes and Issues:	

U0502 Update Food Database

Use Case ID:	U0502		
Use Case Name:	Update Food database		
Created By:	Mehta Rishika	Last Updated By:	Mehta Rishika
Date Created:	29/8/2024	Date Last Updated:	29/8/2024

Actor:	System Integration Service
Description:	The system updates the food database dynamically in response to updates from the API to ensure that the application has the most current and accurate food and nutritional information available.
Preconditions:	The system must have an active and functional API connection for retrieving updates. The food database must be capable of being updated dynamically without manual intervention. The API provides data in a format compatible with the database schema.
Postconditions:	296. The food database is updated with the latest information from the API. 297. Users can access and view the most current nutritional information through the

	<p>application.</p> <p>298. Any changes are logged for tracking and auditing purposes.</p>
Priority:	
Frequency of Use:	As updates from the API are received.
Flow of Events:	<p>299. Trigger Update - The API sends a notification or data update to the system.</p> <p>300. Retrieve Data - The system connects to the API and retrieves the updated food data.</p> <p>301. Process Data - The system processes the data, performing validation and transformation as needed.</p> <p>302. Update Database - The system updates the food database with the new or modified information.</p>
Alternative Flows:	<p>303. Data Retrieval Issue</p> <ul style="list-style-type: none"> • If the API cannot be reached or data retrieval fails, the system logs the error and retries the operation or alerts the administrator. <p>304. Data Processing Error</p> <ul style="list-style-type: none"> • If the data is invalid or processing fails, the system logs the issue and may either retry or notify the administrator for manual review.
Exceptions:	<p>305. If the API is down or returns an error, the system handles the exception by implementing a retry mechanism or notifying support personnel.</p> <p>306. If an issue occurs during the database update (e.g., connectivity issues or schema mismatches), the system logs the error and attempts to roll back changes if possible.</p>
Includes:	
Special Requirements:	
Assumptions:	<p>307. The API is reliable and delivers data in a timely and accurate manner.</p> <p>308. The database schema is designed to accommodate dynamic updates and changes.</p>
Notes and Issues:	

U0503 Request store availability data

Use Case ID:	U0503		
Use Case Name:	Request Store Availability Data		
Created By:	Mehta Rishika	Last Updated By:	Mehta Rishika
Date Created:	29/8/2024	Date Last Updated:	29/8/2024

Actor:	System Integration Service / Web Crawler
Description:	The system requests and retrieves store availability data from various supermarkets in Singapore to provide users with up-to-date information on product availability and pricing. This data helps users make cost-effective food choices by comparing prices and availability across different stores.
Preconditions:	<p>309. The system must have access to a reliable web crawler or API that provides store availability data.</p> <p>310. The store availability data source must be operational and accessible.</p> <p>311. The system must be able to handle and parse data from multiple sources.</p>
Postconditions:	<p>312. Store availability data is retrieved and updated in the system's database.</p> <p>313. Users can view current availability and prices of food items from different stores.</p> <p>314. Any discrepancies or issues in data retrieval are logged and addressed.</p>
Priority:	
Frequency of Use:	Daily at 12 pm
Flow of Events:	<p>315. The system initiates a request for store availability data on a scheduled basis at 12 pm.</p> <p>316. The system uses a web crawler or API to fetch data from participating supermarkets.</p> <p>317. The system processes and parses the retrieved data, extracting relevant information such as product availability and pricing.</p> <p>318. The system updates its internal database with the new store availability data.</p> <p>319. Users can access the updated information through the application's interface.</p>
Alternative Flows:	<p>320. If the data source is unavailable or returns</p>

	<p>an error, the system logs the failure and may retry the request or notify the administrator.</p> <p>321. If the data format is incorrect or parsing fails, the system logs the error and may attempt to correct the issue or request data again.</p>
Exceptions:	<p>322. If the store availability data source is down, the system handles the exception by notifying support and possibly using cached data if available.</p>
Includes:	
Special Requirements:	
Assumptions:	<p>323. The web crawler or API providing store data is reliable and regularly updated.</p> <p>324. The system's database schema supports the integration of store availability and pricing data.</p>
Notes and Issues:	

U0504 Manage User data

Use Case ID:	U0504		
Use Case Name:	Manage User data		
Created By:	Mehta Rishika	Last Updated By:	Mehta Rishika
Date Created:	29/8/2024	Date Last Updated:	29/8/2024

Actor:	System Service
Description:	The system autonomously manages user data within the EatWellthy application. This includes creating, updating, retrieving, and deleting user profiles based on user actions and system requirements. The system ensures that user data is accurately maintained and secure.
Preconditions:	<p>325. The system must have access to the user database.</p> <p>326. User profiles must exist in the database for updates or deletions.</p> <p>327. The system must ensure that data management actions comply with security and</p>

	privacy regulations.
Postconditions:	<p>328. User profiles are accurately created, updated, or deleted as required.</p> <p>329. The database reflects the most current user information.</p> <p>330. The integrity and confidentiality of user data are preserved.</p>
Priority:	
Frequency of Use:	Depends on user actions and needs
Flow of Events:	<p>331. The system detects a request to create, update, retrieve, or delete a user profile.</p> <p>332. The system verifies the user's identity and permissions automatically if applicable.</p> <p>333. Data Management Actions -</p> <ul style="list-style-type: none"> • Create Profile: The system adds a new user profile to the database based on provided information upon registering. The system displays "User successfully registered" • Update Profile: The system updates an existing user profile with new data, ensuring validation and correctness. The system displays "Profile updated successfully" • Retrieve Profile: The system retrieves and displays user profile data as requested. • Delete Profile: The system removes a user profile from the database based on the request. <p>334. The system ensures that the database is updated in real time to reflect the changes made.</p>
Alternative Flows:	<p>335. If there is an issue during the profile update process, the system handles the error by logging it and notifying the user or administrator a message, "Unable to update the profile, please try again".</p> <p>336. If the deletion fails, the system logs the error and informs the user.</p>
Exceptions:	<p>337. The system denies access to data management functions if the request lacks proper authorization and logs the attempt.</p>
Includes:	<p>338. Authentication and authorization mechanisms.</p> <p>339. Data validation processes.</p>

	340. Error handling and logging functions.
Special Requirements:	
Assumptions:	<p>341. The system has the necessary access and permissions to manage user data.</p> <p>342. Data management functions are integrated with the system's security and privacy protocols.</p>
Notes and Issues:	

U0505 Backup and Restore

Use Case ID:	U0505		
Use Case Name:	Backup and Restore		
Created By:	Mehta Rishika	Last Updated By:	Mehta Rishika
Date Created:	29/8/2024	Date Last Updated:	29/8/2024

Actor:	System server
Description:	The system manages the backup and restoration of data to ensure data integrity and availability. This includes creating periodic backups of user data and other critical information, as well as restoring data from backups when necessary. This process helps in data recovery in case of system failures, data corruption, or accidental loss.
Preconditions:	<p>343. The system must have access to the database and backup storage.</p> <p>344. Backup policies and schedules must be defined and configured.</p> <p>345. Data to be backed up must be in a consistent state.</p>
Postconditions:	<p>346. Backup files are created and stored securely according to the backup schedule.</p> <p>347. Data is restored accurately from backup files as needed.</p> <p>348. The system ensures the integrity and availability of user data.</p>
Priority:	High
Frequency of Use:	Regularly scheduled (weekly) for data recovery
Flow of Events:	<p>349. The system triggers a backup process according to the predefined schedule or upon request.</p> <p>350. The system collects and compiles user</p>

	<p>data and other critical information from the database.</p> <p>351. The data is compressed and encrypted if necessary.</p> <p>352. The system stores the backup in a secure, designated storage location.</p> <p>353. Confirmation by updating the logs.</p>
Alternative Flows:	<p>354. If the backup process fails, the system logs the error and notifies the system administrator. The system may attempt to retry the backup or alert the administrator to take corrective actions.</p>
Exceptions:	<p>355. If there is not enough storage space for backups, the system notifies the administrator and may halt the backup process until space is freed up.</p> <p>356. If the data in the backup file is corrupted, the system logs the issue and may attempt to use an alternative backup or pop an error.</p>
Includes:	
Special Requirements:	
Assumptions:	<p>357. The backup and restore mechanisms are properly configured and tested.</p> <p>358. Backup storage is secure and accessible to the system.</p>
Notes and Issues:	

VI. For Functional Requirement #7 (Meal Planner)

U0601 View Dietary History

Use Case ID:	U0601		
Use Case Name:	View Dietary History		
Created By:	Zhang Yichi	Last Updated By:	2024/08/30
Date Created:	2024/08/30	Date Last Updated:	2024/08/30

Actor:	User (login)
Description:	User requests to view their dietary history in a structured

	format that shows the meals consumed over a specific period.
Preconditions:	<p>359. User must be logged in</p> <p>360. The dietary history must be recorded and stored in the system.</p> <p>361. The system should have access to the user's dietary records.</p>
Postconditions:	<p>362. The system displays the dietary history in a readable format.</p> <p>363. The dietary data reflects the user's consumption accurately.</p> <p>364. The dietary history is presented without any errors.</p>
Priority:	Medium
Frequency of Use:	Daily (Users typically check their dietary history once a day or week.)
Flow of Events:	<p>365. User logs into the system.</p> <p>366. User navigates to the dietary history section.</p> <p>367. User selects the desired time frame for viewing dietary history.</p> <p>368. The system retrieves dietary records from the database.</p> <p>369. The system displays the dietary history on the screen.</p>
Alternative Flows:	<p>370. If the system fails to retrieve dietary records, an error message is displayed to the user.</p> <p>371. If no records are found for the selected time frame, the system informs the user with a message.</p>
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	The record are stored fully without glitch
Notes and Issues:	

U0602 Analyze Nutritional Intake Trends

Use Case ID:	U0602
Use Case	Analyse Nutritional Intake Trends

Name:			
Created By:	Zhang Yichi	Last Updated By:	08/30/2024
Date Created:	08/30/2024	Date Last Updated:	08/30/2024

Actor:	User
Description:	User analyzes nutritional intake trends based on their historical dietary data.
Preconditions:	<p>372. User must be logged in.</p> <p>373. The system must have access to historical dietary data.</p> <p>374. Nutritional analysis algorithms must be implemented in the system.</p>
Postconditions:	<p>Related data sum up correctly</p> <p>375. The nutrition data of the chosen items would be draw from nutrition sheets correctly</p> <p>376. The system provides a visual and statistical analysis of the user's nutritional intake trends.</p> <p>377. The trends reflect the user's consumption habits accurately.</p>
Priority:	High
Frequency of Use:	Weekly (Users often review their nutritional trends weekly.)
Flow of Events:	<p>378. User logs into the system.</p> <p>379. User navigates to the nutritional analysis section.</p> <p>380. User selects the desired parameters for analysis (e.g., time frame, specific nutrients).</p> <p>381. The system retrieves relevant dietary data from the database.</p> <p>382. The system processes the data and generates an analysis report.</p> <p>383. The system displays the nutritional intake trends on the screen.</p>
Alternative Flows:	<p>384. If the system fails to generate an analysis report, an error message is displayed to the user.</p> <p>385. If no data is available for the selected parameters, the system informs the user accordingly.</p>

Exceptions:	386. Data inconsistencies or missing data may affect the accuracy of the analysis.
Includes:	
Special Requirements:	
Assumptions:	The record are stored fully without glitch
Notes and Issues:	

U0603 Generate User Reports

Use Case ID:	U0603		
Use Case Name:	Generate User Reports		
Created By:	Zhang Yichi	Last Updated By:	08/30/2024
Date Created:	08/30/2024	Date Last Updated:	08/30/2024

Actor:	User
Description:	Users generate detailed reports on their dietary and nutritional intake.
Preconditions:	387. User must be logged in. 388. The system should have access to all relevant user data. 389. Report generation functionalities must be implemented.
Postconditions:	390. The system generates a comprehensive report based on user data. 391. The report accurately reflects the user's dietary and nutritional intake.
Priority:	Medium
Frequency of Use:	Monthly (Users typically generate reports on a monthly basis.)
Flow of Events:	392. User logs into the system. 393. User navigates to the report generation section. 394. User selects the desired report type and parameters. 395. The system retrieves necessary data from database. 396. The system generates the report and displays it on the screen or allows download.
Alternative Flows:	397. If the system fails to generate the report,

	an error message is displayed to the user. 398. If no data is available for the selected parameters, the system informs the user.
Exceptions:	399. Data inconsistencies or missing data may affect the accuracy of the analysis.
Includes:	400. System errors during data retrieval or report generation.
Special Requirements:	
Assumptions:	The record are stored fully without glitch
Notes and Issues:	

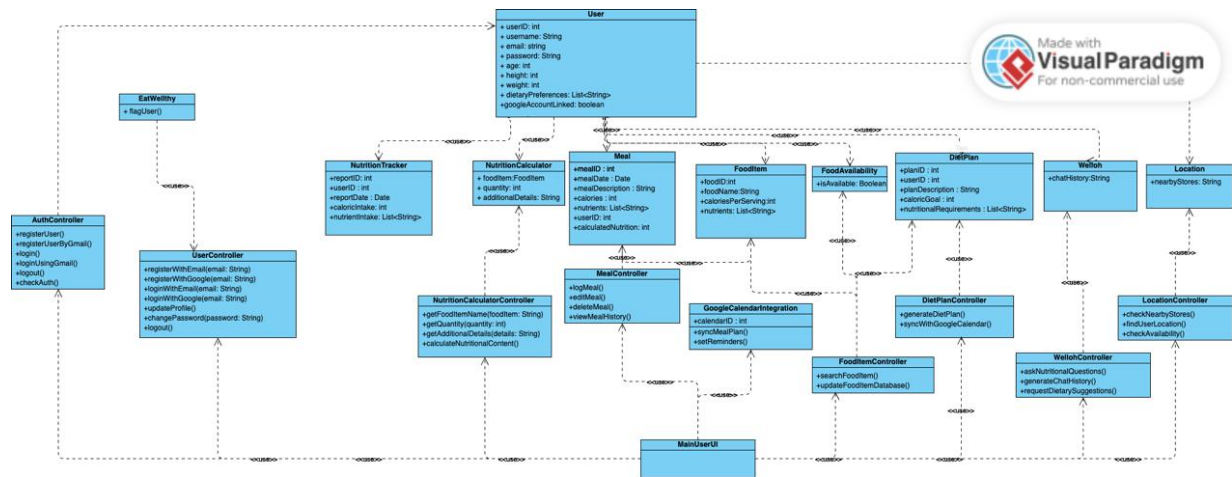
U0604 Export Nutritional Data

Use Case ID:	U0604		
Use Case Name:	Export Nutritional Data		
Created By:	Zhang Yichi	Last Updated By:	08/30/2024
Date Created:	08/30/2024	Date Last Updated:	08/30/2024

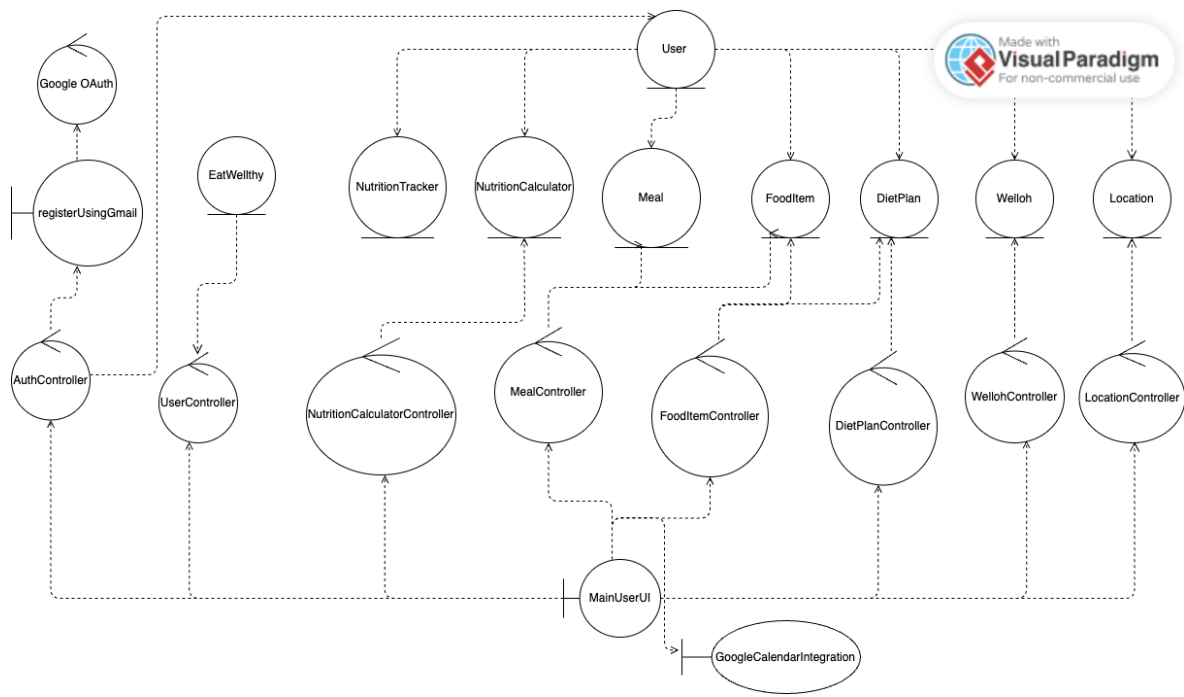
Actor:	User
Description:	Users export their nutritional data for external analysis or record-keeping.
Preconditions:	401. User must be logged in. 402. The system should have access to the user's nutritional data. 403. Export functionality must be enabled in the system.
Postconditions:	404. The system exports the nutritional data in the desired format. 405. The exported data is accurate and complete.
Priority:	Low
Frequency of Use:	Occasionally (Users export data when needed for external purposes.)

Flow of Events:	<p>406. User logs into the system.</p> <p>407. User navigates to the data export section.</p> <p>408. User selects the desired data range and format for export.</p> <p>409. The system retrieves the relevant data from the database.</p> <p>410. The system exports the data in the selected format and provides a download link.</p>
Alternative Flows:	<p>411. If the system fails to export the data, an error message is displayed to the user.</p> <p>412. If no data is available for the selected range, the system informs the user.</p>
Exceptions:	N
Includes:	
Special Requirements:	
Assumptions:	The record are stored fully without glitch
Notes and Issues:	

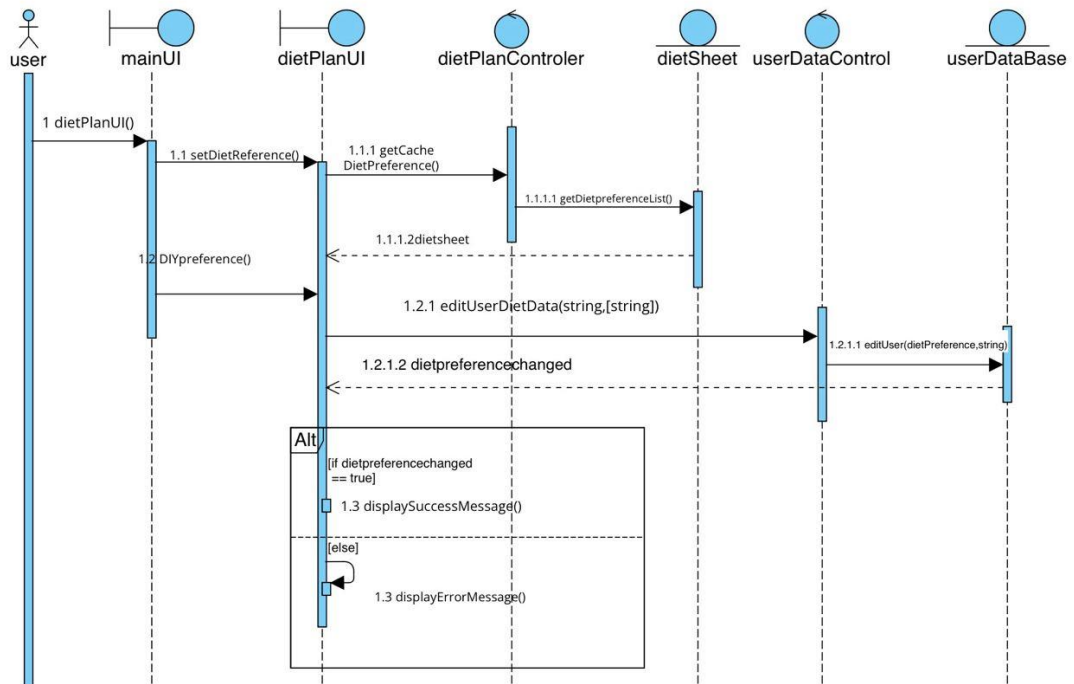
Class Diagram



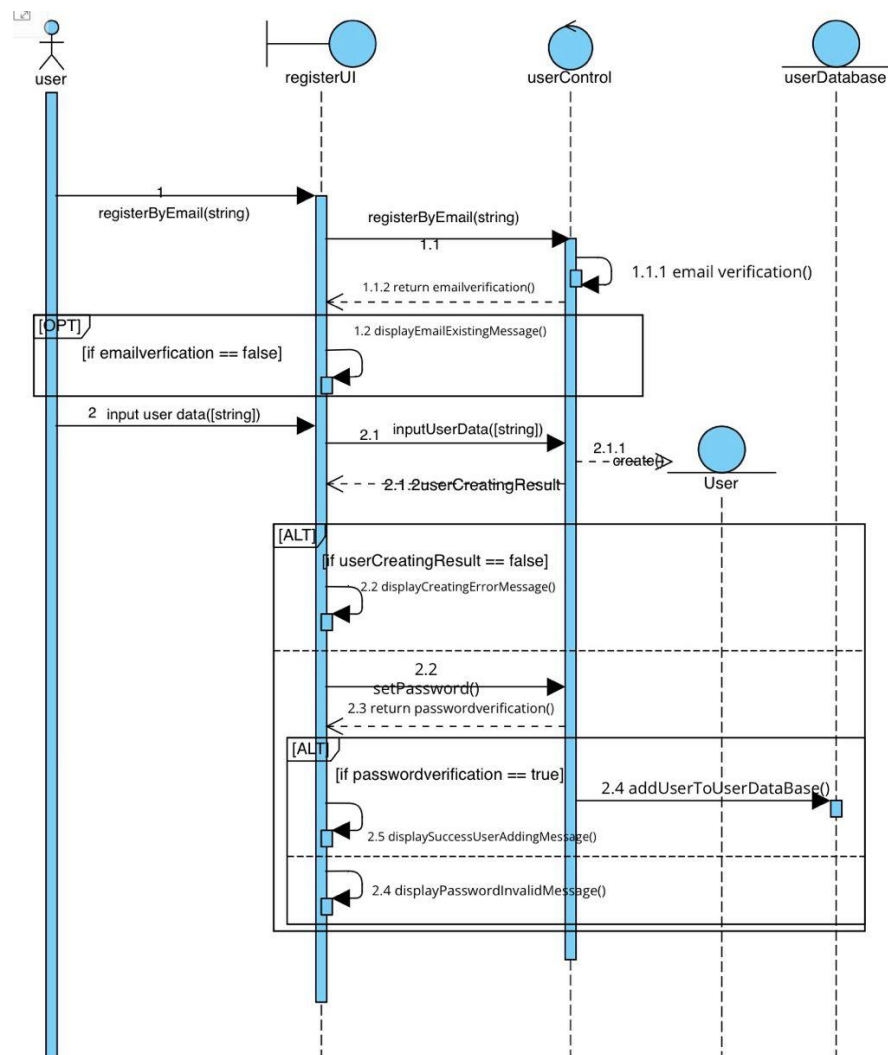
Key Boundary Diagram



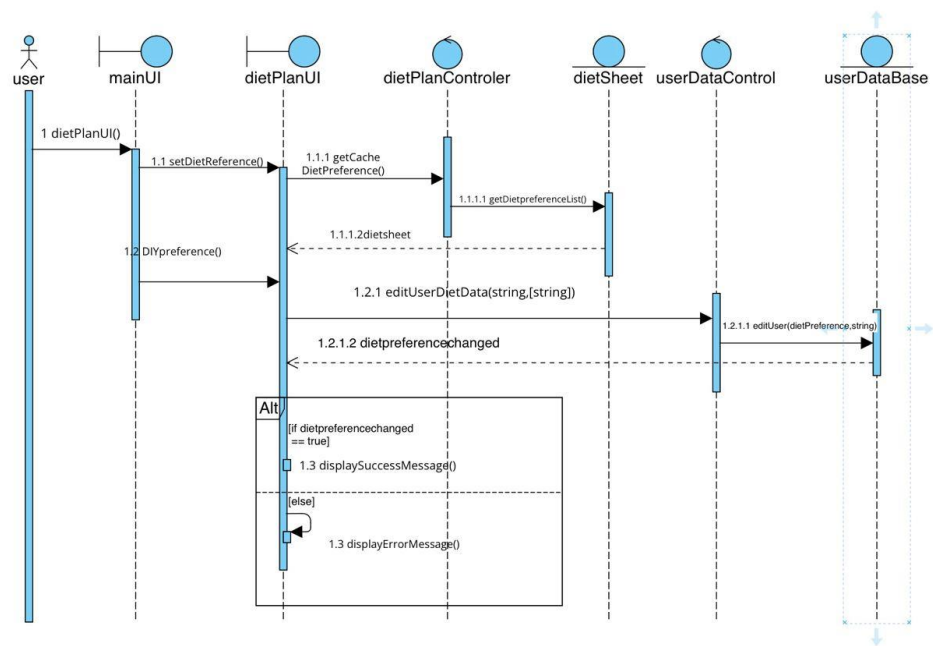
Sequence diagram 1-3



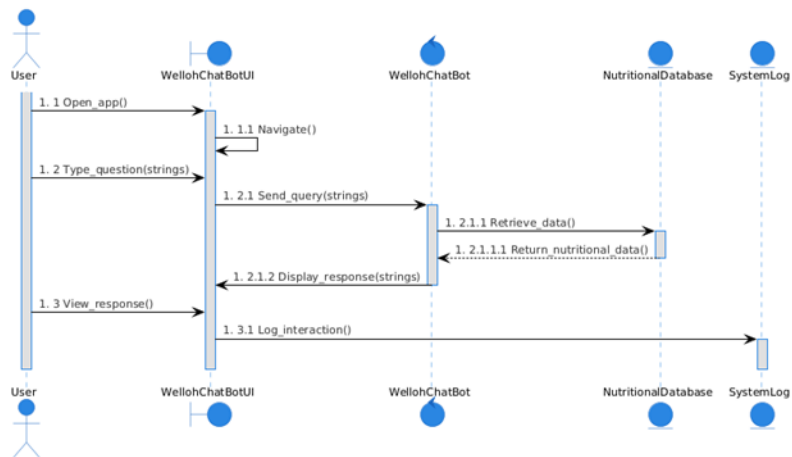
U1



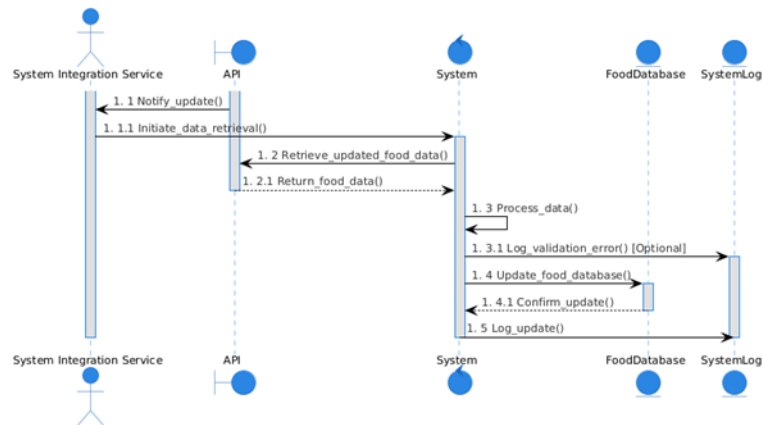
U2



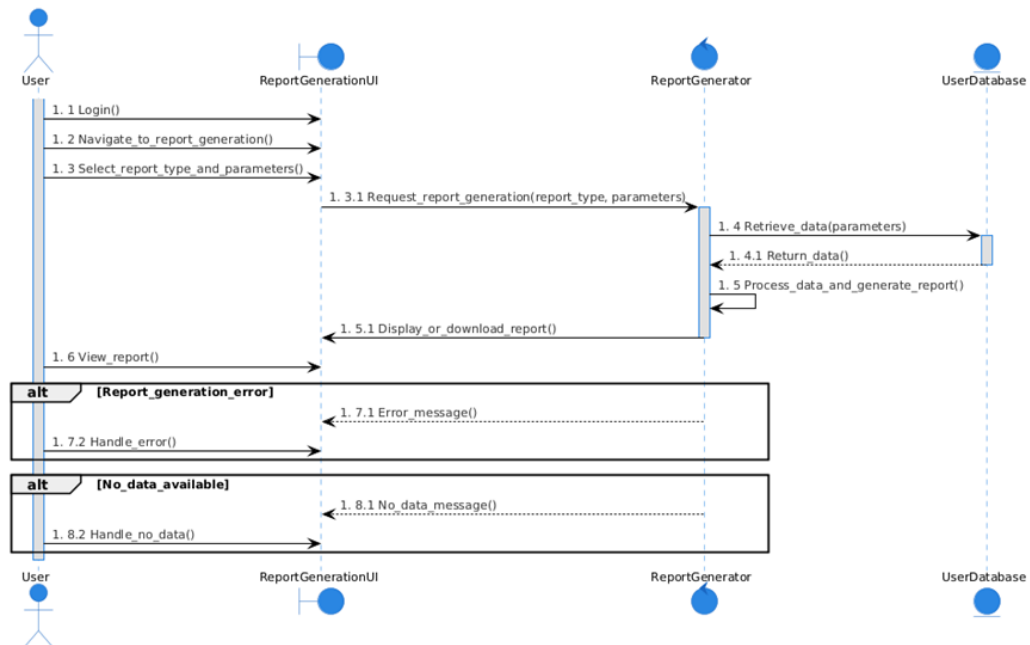
U3



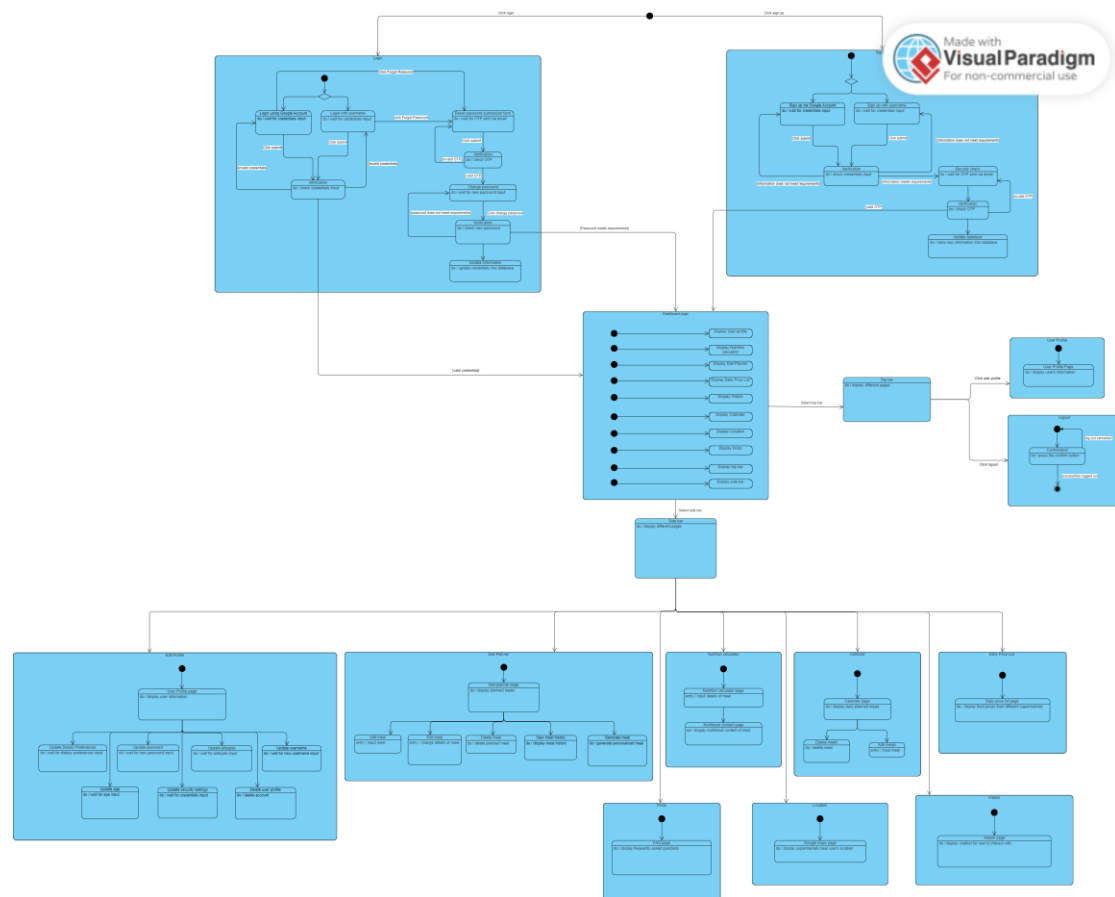
U4



U5



Dialog Map



Link to view clear diagram:

<https://online.visual-paradigm.com/share.jsp?id=333536313435312d31#diagram:workspace=bngafzqv&proj=0&id=1>