

myCalorie Tracker Project

OMIS 651- Business Systems Analysis & Design

Northern Illinois University

Date: December 7th, 2020

By:

(Group 5)

Raj Kumar

Behrooz Alipoor

Sukhada Kanetkar

Table of contents:

Summary	2
Introduction to the current system and challenges	3
Current System	3
Challenges in the existing system:	4
Proposed solution	5
Software development approach	6
Requirements	7
Functional requirements:	7
Non-functional requirements:	8
Five-Factor Analysis	8
Use Cases	9
Use Case 1	9
Use Case 2	10
Use Case 3	12
Use Case 4	13
System acquisition strategy	14
UML diagrams	15
Use Case diagram	15
Class diagram	16
State-machine diagram:	17
Sequence diagram:	18
UI designs	19
Test plan	23
Test Cases	25
Test case 1:	25
Test Case 2:	25
Test Case 3:	26
Test Case 4:	27
Transition strategy	27
Conclusion:	28

Summary

The 'myCalorieTracker' is a mobile application for fitness programs that allow users to track their calorie intake. With this application, users can keep a record of their daily food consumption and align themselves with their fitness goals. Users can access different functions to get more insights into their health, diet, and physical activities.

With an increasing demand for health awareness, people are getting more concerned about their fitness. People have started using smart gadgets to keep them aligned with their health goals. To provide the level of accuracy and predictive health monitoring, the company has decided to upgrade its system to support a new device called 'Smart Scanner'. This device is invented by us and can provide accurate food calorie information by simply scanning and weighing the food item. The company has decided to develop a smartphone app that can support this gadget. With this newly developed system, users can easily pair the smart scanner with their smartphones and always maintain their food diet precision. The new application will also support some additional features to support a wide variety of users.

With the help of our customer feedback and surveys, we initiated the product requirement document to capture popular customer needs. A project plan was created to best support all the phases of the project. We also identified a development approach for the software development team for optimum performance. We defined the use cases and figured out the system acquisition strategy. UML and UI diagrams were developed. In order to test the software, we

created the test plan and defined the test cases. The application will be thoroughly tested by executing the test cases, and then a bug report will be submitted.

Introduction to current system and challenges

Current System

The current system allows users to select food items based on a fixed value to generate a diet plan for a certain number of calories. The application uses a database where each type of food is assigned with a fixed number of calories for a fixed quantity. So, the application calculates the number of calories by using information that is stored in the database and provides the output to the users.

Users can find the calories information on the web application by entering the food details, for example, food category, name, etc. The application is beneficial for people from different areas like health, medical, sports, etc.

To use the application, the user first needs to register and create an account. Once the user created an account, they can search the calories information by entering the food details manually. Users can view this information and save it for future use.

The application also provides some standard diet plans that customers may follow as per requirement. Customers calculate calories and nutrients they intake manually by looking at the chart. The application allows users to save these calories and nutrients data in the application manually, and they can log this information daily to keep track of calorie intake.

The application uses a database to store information related to customers' data. When the customer inputs values, the application searches this information in the database. Furthermore, the output is displayed to the customers.

Challenges in the existing system

There is a device called a smart scanner that calculates the calories and nutrients directly by scanning the food, but the current system is not compatible with the device to automate the calories tracking process. Every time a user has to manually enter the food information into the web application to find the food components. The company is facing challenges like lack of automated processes, connectivity, custom diet plan, and lack of accuracy.

The current system's challenge is that the system is semi-manual, and it does not automatically calculate the food components. Some user processes are not automated. The customer needs to enter data manually to see the result. Furthermore, the user needs to log data manually to keep track of daily calorie intake. This is a time-consuming process. Furthermore, the company has noticed that they are unable to attract more customers.

The existing system has limitations on connecting with other applications. User profiles and data cannot be linked with other applications. For example, a user cannot link or share their calorie information with other health applications. The current system does not allow the creation of custom diet plans as per user requirements. For example, customers cannot create diet plans using their personal information like age, gender, health condition, etc. The system just provides some standard diet plans.

The system measures food components from the standard values, which are stored in the database. It does not actually weigh or measure the quantity of the food being consumed. So, the customers are unable to find an accurate number of calories required for their diet plan.

To overcome all these challenges, the company has decided to find a solution that can provide accurate measurements, smart connectivity, customizable diet plans, and user-friendly automated processes.

Proposed solution

The company has found a new solution that overcomes all the challenges described in the previous section. The new proposed system is also compatible with the device called a smart scanner.

With the proposed new system, we are developing a new mobile application using the smart scanner. Through the system, the users would find the exact chemical compositions and nutrients in the food. With the help of the smart scanner and an electronic weighing scale, a user would scan and weigh the food, respectively.

The proposed mobile application works as a user interface. The microprocessor present in the smart scanner and the weighing scale would process the information and send it to the connected application on the phone through Bluetooth. The application would then match the information available in the database library over the cloud and display the results. Using an algorithm, the application would be able to process and calculate the composition's exact

distribution. The users would view the chemical composition, total calories of the food, and the nutrient's breakup such as protein, carbohydrates, fats, vitamins, etc.

Through the new system, users would be able to customize their diet plan as well. If the user authorizes, the system automatically tracks and integrates all the daily calorie intake, which is scanned. The new system would be able to identify most food by itself. Customers can also link their information with the other applications available on the mobile.

Software development approach

We will be following the agile scrum methodology to develop the software because it has several benefits over traditional approaches. In the scrum framework approach, the development team works simultaneously rather than sequentially. We are following this approach because it helps release high quality and usable products to users and customers. It is also a flexible process; any changes are accepted during the implementation. Requirements are changeable. There are continuous iterations, and each iteration consists of two to four weeks of sprints. It provides more significant customer satisfaction because they get high-quality, and productive portions of the completed product quicker. Customers can try and provide feedback for it. This is a critical stage to ensure overall project success.

Our scrum team contains a product owner who has gathered all the requirements and is responsible for the product backlog, meanwhile, a scrum master is responsible for the sprint planning, and a development team is responsible for implementing the product. The timeline for each sprint is decided as four to five weeks long. We are using the JIRA tool for requirements tracking, test case, issue reporting, and use cases writing. Moreover, everyone is following the JIRA tool to log their work.

A product backlog is a list of all the features implemented in each sprint by the development team. In sprint planning, the scrum team decided the product backlog that contains the features like Login screen, Home screen, Bluetooth connectivity, Profile screen, and Custom diet plan. These modules are planned to implement in different sprints. The first Sprint will include the implementation of the login screen and the daily sprint meetings are conducted to keep track of work status.

Further, the work status is updated in the JIRA tool. We will be conducting a sprint review meeting and sprint retrospective at the end of each sprint. The sprint review is about the assessment of the product being developed during the Sprint. Also, the product owner may release the module or feature in the sprint review meeting. The sprint retrospective is about the feedback of the Sprint process and people. During the sprint retrospective, the scrum team discusses the sprint process, figures out what went right and wrong, as well as areas of improvement needed for the next Sprint.

Requirements

- The user should be able to log in
- The user should be able to update personal information
- The user should be able to select and customize their goal plan
- The user should be able to add the scanned items to the system
- The user should be able to view the requested calorie information

Functional requirements:

- The system should be able to authenticate and allow users to login
- The system should allow users to update their profiles.

- The system should allow users to search and add food items.
- The system should allow users to scan using a smart scanner and add items
- The system should be able to retrieve calorie information from the database
- The system should be able to display the retrieved calorie and other nutrient information
- The system should allow users to view the past tracking history for up to 1 year

Non-functional requirements:

- The system should be able to respond and fulfill the user's request within 10 seconds.
- The system should be able to handle 1 million users.
- The system should be able to handle 200,000 users at its peak time.
- The system should have security features protecting user personal information and data.
- The system should be developed for iOS and Android devices.
- A successful account created email should be sent out to the user when a new account is created.

Five-Factor Analysis

Factor	Approach
Size	Development: Small Integration: Large

Criticality	Low criticality
Dynamism	Dynamic
Personnel	Development: High Integration & Maintenance : Low
Culture	Development: Chaos Integration: In order

Use Cases

Use Case 1

Use Case Name: Login		ID: UC-1	Priority: High
Actor: User			
Description: The user logs in to the system. If the user doesn't have an account then they create an account. The system satisfies the request by logging in			
Trigger: The user enters the account by entering the credentials. Type: External			
Preconditions: 1. The user has opened the application			
Normal Course: <ol style="list-style-type: none"> 1. The user sees the login page. 2. The system requires user credentials. 3. The user enters the credentials. 4. The system acknowledges the credentials. 5. The system welcomes the user on the homepage. 		Information for Steps: <p>-> log in request</p> <p><- Credential request</p> <p>-> Enter credential</p> <p><- System acknowledgment</p> <p>-> User log in</p>	

Alternative Course: <div><div><div>1. The user logs in to the system but is denied access.</div><div>2. The user receives denial acknowledgment.</div><div>3. System asks the user to re-enter credentials.</div><div>4. The user re-enters the credentials.</div><div>5. The system acknowledges the credentials.</div><div>6. The homepage is visible to the user.</div></div><div><div><- Deny Notification</div><div><- Credential request</div><div>-> User re-enter credential</div><div><- System acknowledgment</div></div></div>			
Postconditions: <div><div>1. The user is logged in and it on the dashboard screen.</div><div>2. Users can update information.</div></div>			
Exceptions: E1 <div><div>1. The user is not able to login due to technical issues.</div><div>2. The system asks the user to exit.</div><div>3. The user selects the exit option.</div><div>4. The system terminates the use case</div></div>			
Summary Inputs	Source	Outputs	Destination
log in request Credential request Enter credential System acknowledgment User login	User Database system User Database system Database System	Deny notice User denied Credential request User re-enter credential System acknowledgment	Database System User Database system User Database System

Use Case 2

Use Case Name: Request calorie info	ID: UC-02 Priority: High
Actor: User and System Database	
Description: The user processes the request calorie info after selecting the goal plan.	

Trigger: The user requires the calorie info. Type: External			
Preconditions: 1. The user identity is authenticated. 2. The user has selected the goal plan.			
Normal Course: 1. The user already logged in by entering the credentials. 2. The user selects the goal plan. 3. The user requests calorie info. 4. The system verifies the request. 5. The system processes the request 6. The system acknowledges the request.		Information for Steps: -> User credential -> Goal plan selection -> Info request <- Request verification <- Request processing <- acknowledgment	
Alternative Course: 1. The user selects a different goal plan. 2. The user requests different info. 1. The system verifies the request. 2. The system processes the request. 3. The system acknowledges the request.		 -> Goal plan selection -> Info request <- Request verification <- Request processing <- Acknowledgment	
Postconditions: 1. The system checks the calories information in the database			
Exceptions: E-2: 1. The requested object information is not available 2. The system will display a message “The request for calorie info is not available now”. 3. The system will ask the user to specify different info or exit. 4. The user selects the to exit 5. The system terminates the use case			
Summary Inputs	Source	Outputs	Destination

User credential Goal plan selection Info request Request verification Request processing Request acknowledgment	User User User System Database System Database System Database	Goal plan selection Info request Request verification Request processing Request acknowledgment	User System database System database System database
--	---	---	---

Use Case 3

Use Case Name: Scanning object		ID: UC-03	Priority: High
Actor: User and Scanner			
Description: The user scans the object and the system satisfies the request by scanning the object			
Trigger: The user selects the scan button, and the scanner completes the scanning procedure. Type: External			
Preconditions: <ol style="list-style-type: none"> 1. The user needs to know the calorie information on the food object. 2. The scanner can scan one object at a time unless it is a homogeneous mixture 3. The scanner is ready to scan the object and it is plugged into a power source and connected to the mobile device 			
Normal Course: <ol style="list-style-type: none"> 1. The user scans the object. 2. The system identifies the object. 3. The user weights the object. 4. The system processes the scan 5. The system matches the object into the system database. 6. The system acknowledges the object in the system database. 		Information for Steps: <ul style="list-style-type: none"> -> Scanning -> Object identification -> Scanning <- Retrieving <- Acknowledgment 	

Alternative Course: 1. The user enters the object weight or quantity manually. 2. The user selects the menu 3. The user registers the object into the database. 4. The system sends an acknowledgment 5. The system displays the message: information will be available soon". 6. The system terminates the use case.		 -> Object measurement -> Object identification -> Object registration -< Acknowledgement	
Postconditions: 1. The system is able to retrieve the information for requested item 2. The user can move on to the view report screen			
Exceptions: E-3: 1. The scanner fails to scan due to technical problems. 2. The system displays an error message and ask the user to continue or exit 3. The selects to continue 4. The system displays the home screen 2b. The user selects to exit 3b. The system terminates the use case.			
Summary Inputs	Source	Outputs	Destination
scanning process object identification scanning process retrieving process acknowledgment process	User Sys. database Sys. database Sys. database Sys. database	object measurement object identification object registration	User User User

Use Case 4

Use Case Name: Calculate data	ID: UC-04 Priority: High
Actor: User and System Database	

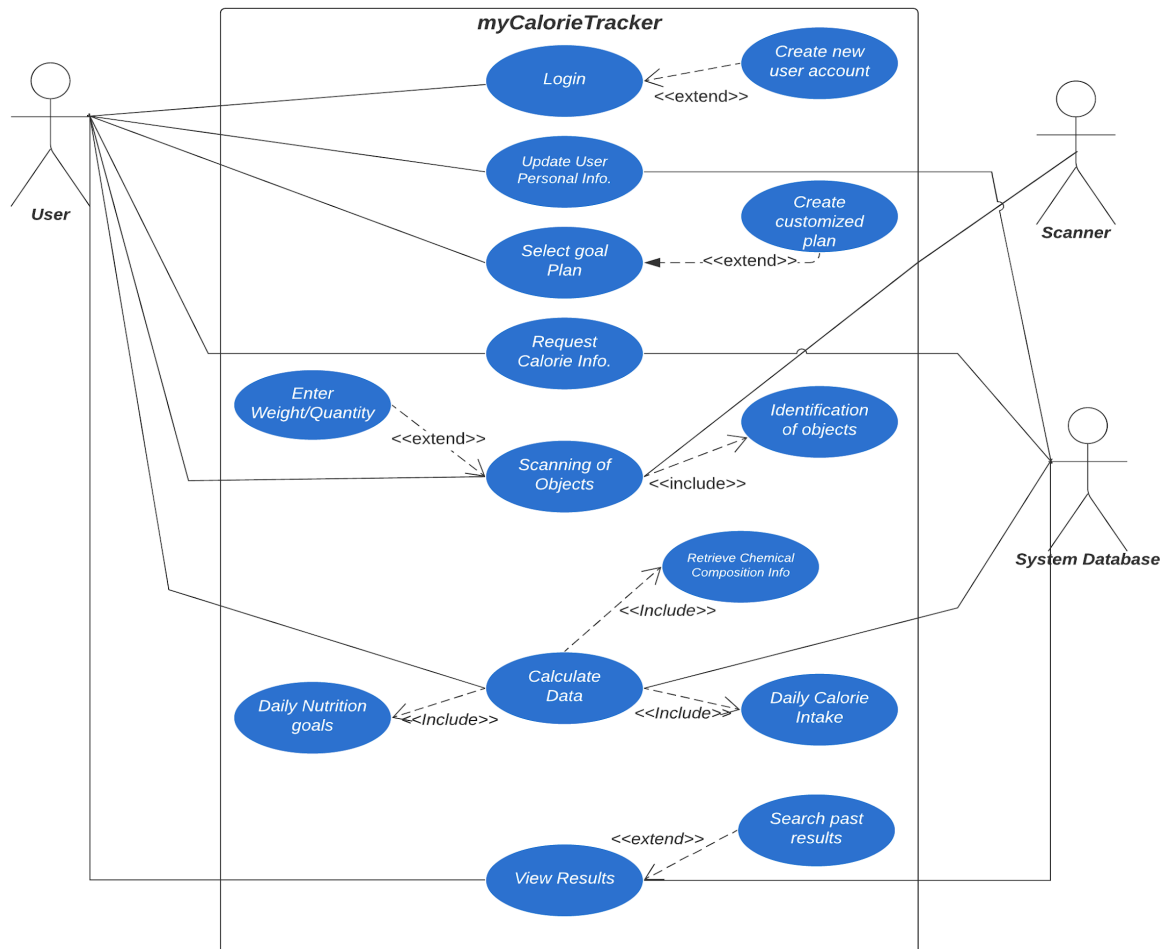
Description: After scanning objects, the system database calculates the data and displays the result.			
Trigger: The user has requested the calorie and other nutrient information Type: External			
Preconditions: 1. The user has scanned or selected the food items.			
Normal Course: 1. The system processes the data calculation. 2. The system completes the data calculation. 3. The system displays the data calculation result. 4. The user views the data calculation result.		Information for Steps: -<-process acknowledgement -<- Display result process -<- View result process	
Alternative Course: 1. The User postpones the data calculation process. 2. The system stores the data calculation result. 3. The user views the data calculation result later.		-<- Delayed process -> Store process -<- View result process	
Postconditions: 1. The user can view the calculated result. 2. The system notifies the user of the calculated data result.			
Exceptions: E-4: 1. The data calculation process fails due to a failure with the system database. 2. The system displays the error message and asks to exit. 3. The user selects to exit. 4. The system terminates the use case.			
Summary Inputs	Source	Outputs	Destination
Calculation process Completion of the calculation process	System Database System Database	Delayed process Store process	User System Database
Display result process View result process	System Database User	View result process	System Database

System acquisition strategy

We'll be using an in-house custom development strategy. This strategy works for us because our business needs are unique and we don't want our core business needs to be exposed. Not only that, in-house development allows us to be a little flexible on our requirements which may modify later in the development stage according to our needs. The other reason for having this strategy is we have in-house technical employees and developers to develop the system. Our team has experience designing other systems. One last reason to go with in house strategy is that our time frame is pretty flexible.

UML diagrams

Use Case diagram

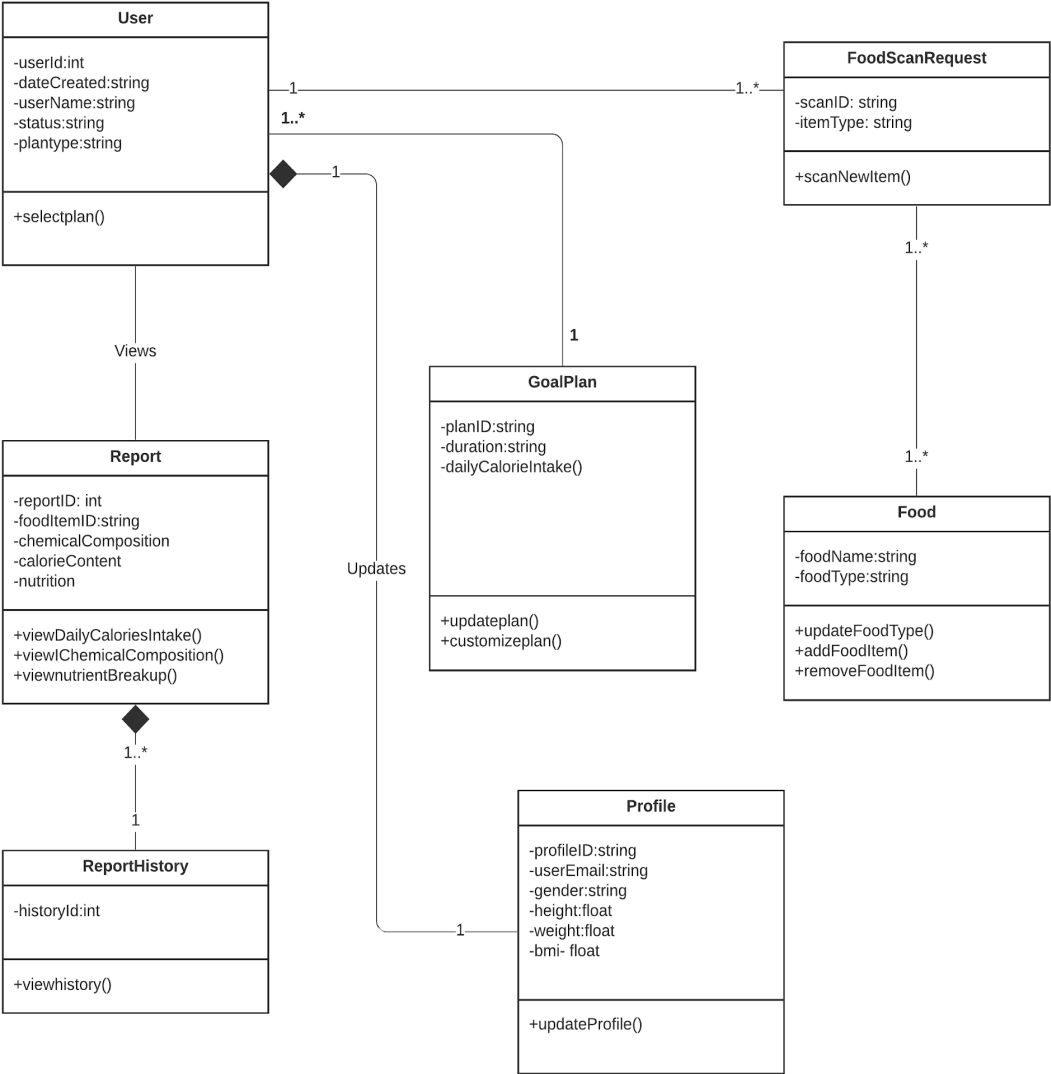


User: User logs in and updates personal information. The user also selects the goal plan and request calorie information. The user scans the objects(food) and triggers the calculation and processing of the data. The user views the result.

Scanner: The scanning of the object is initiated by the user and completed by the scanner by scanning and sending the data to the system.

System Database: The database verifies and sends the required calories request and other necessary information requested by the system.

Class diagram



selectplan() function allows users to select their goal plan.

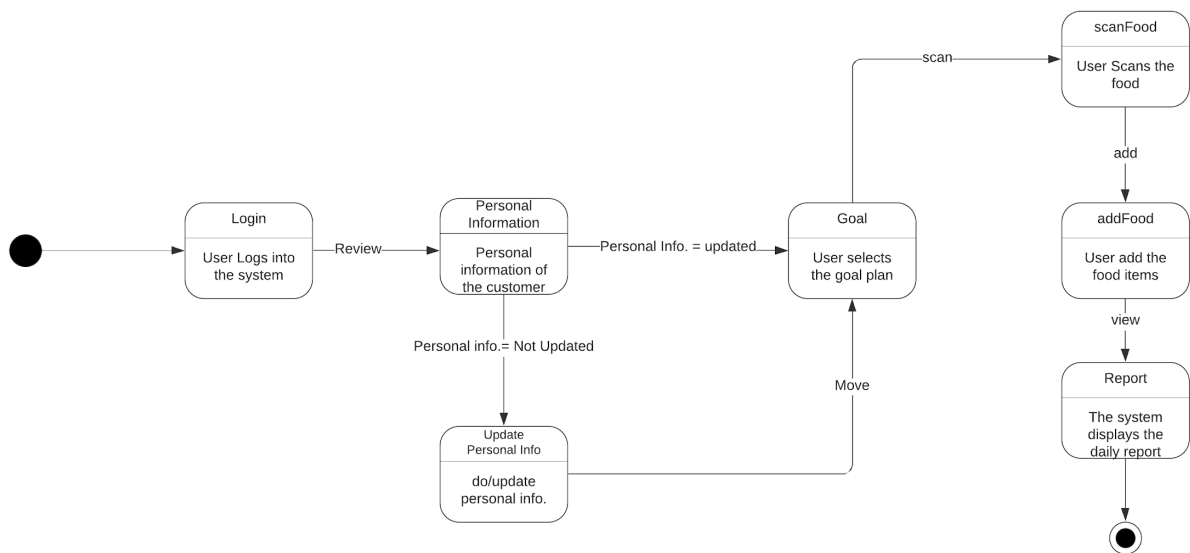
updateProfile() allows users and the system to update the profile and personal information of the system.

scanNewItems() allows new items to be scanned for addition.

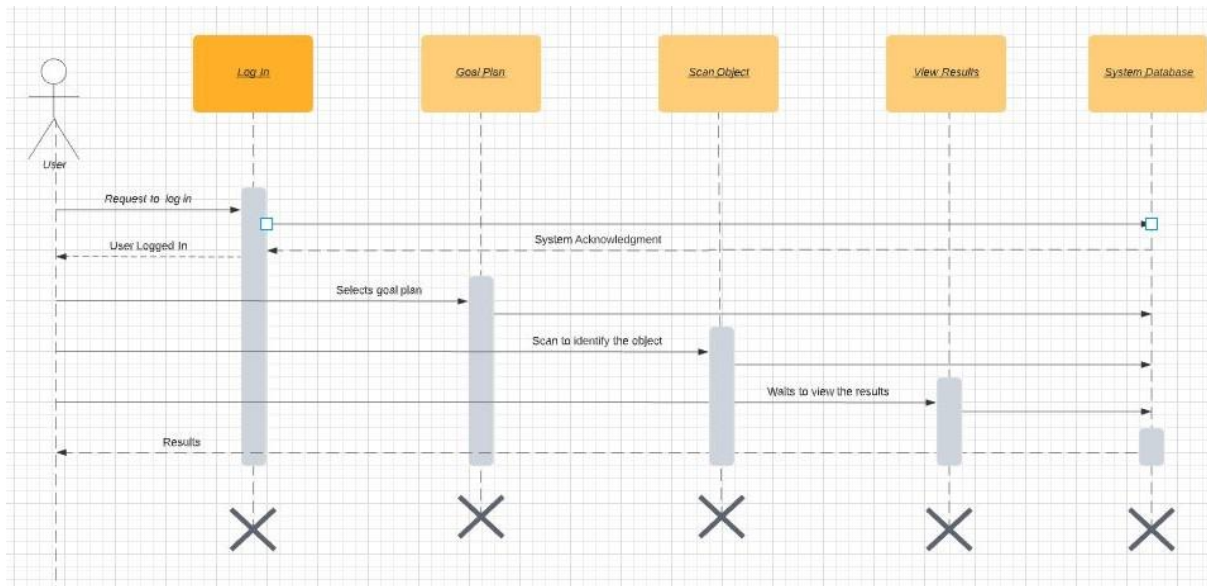
viewDailyCalorieIntake() provides all the calories information .

viewHistory() allows users to view the past history of calories.

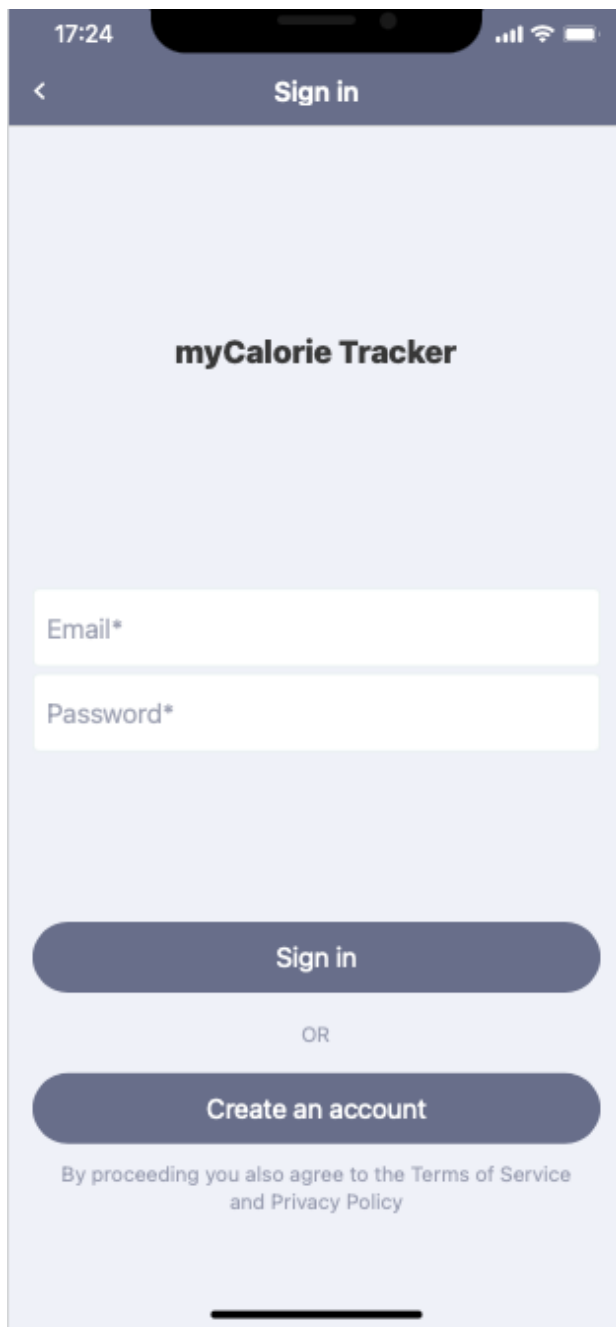
State-machine diagram:



Sequence diagram:



UI designs



17:24

< Sign in

myCalorie Tracker

Email*

Password*

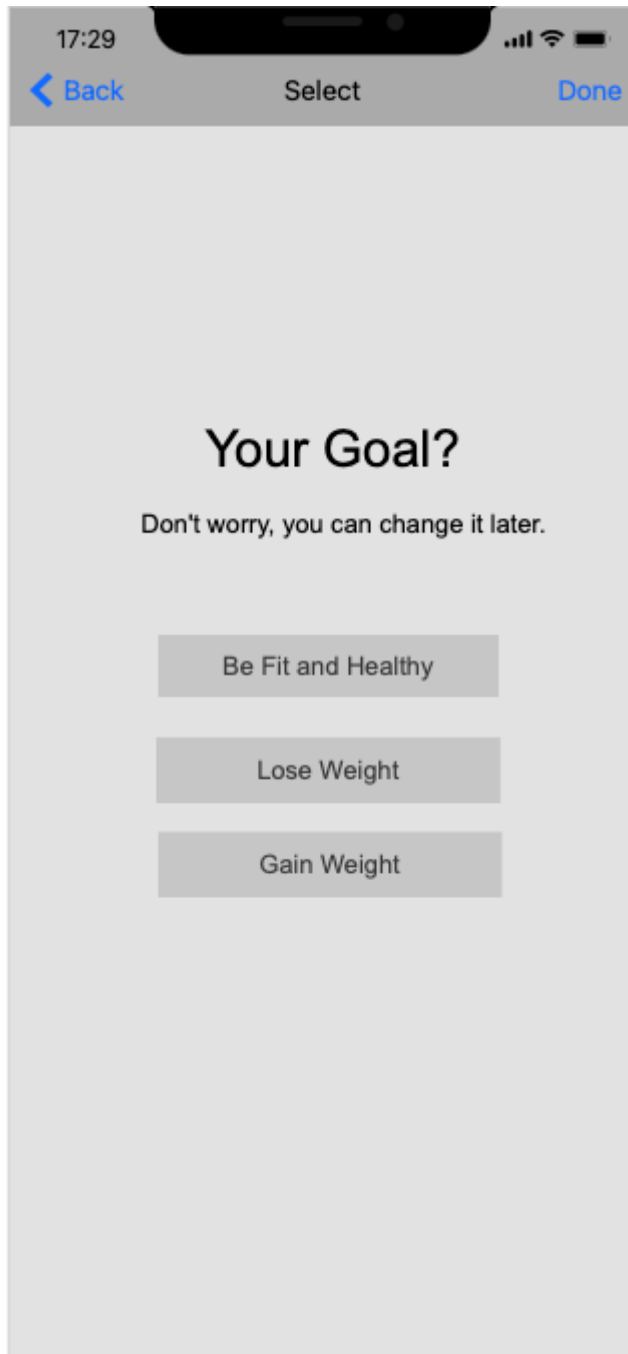
Sign in

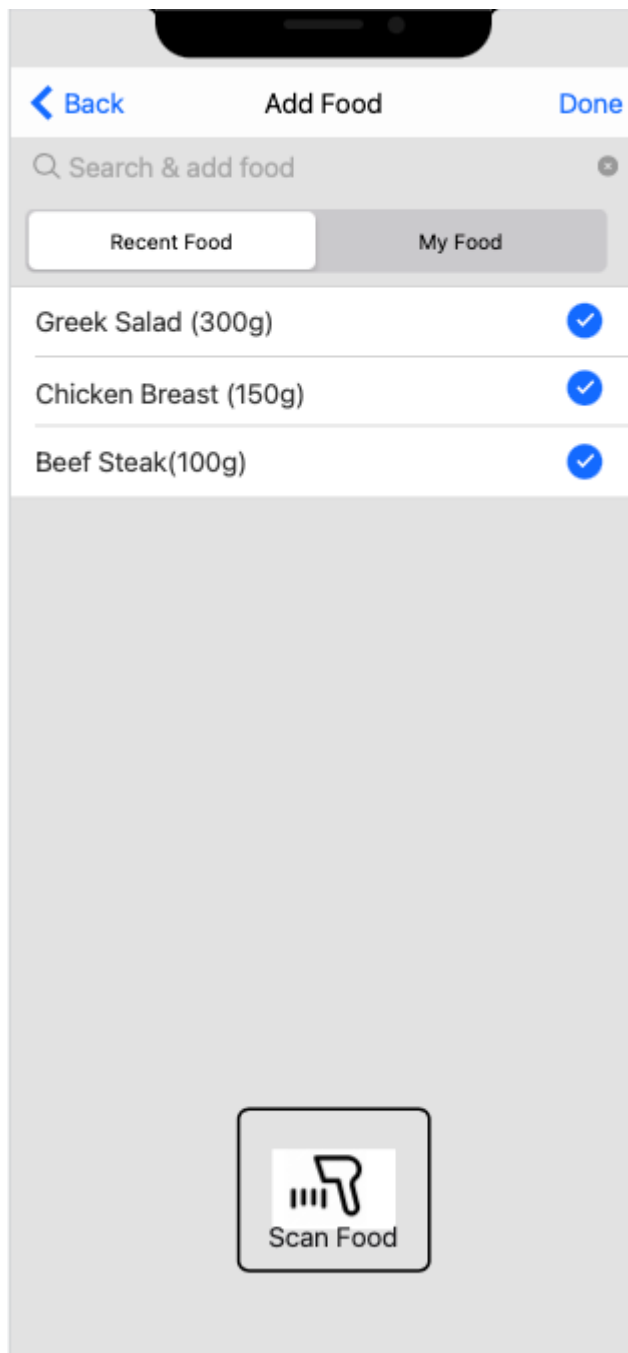
OR

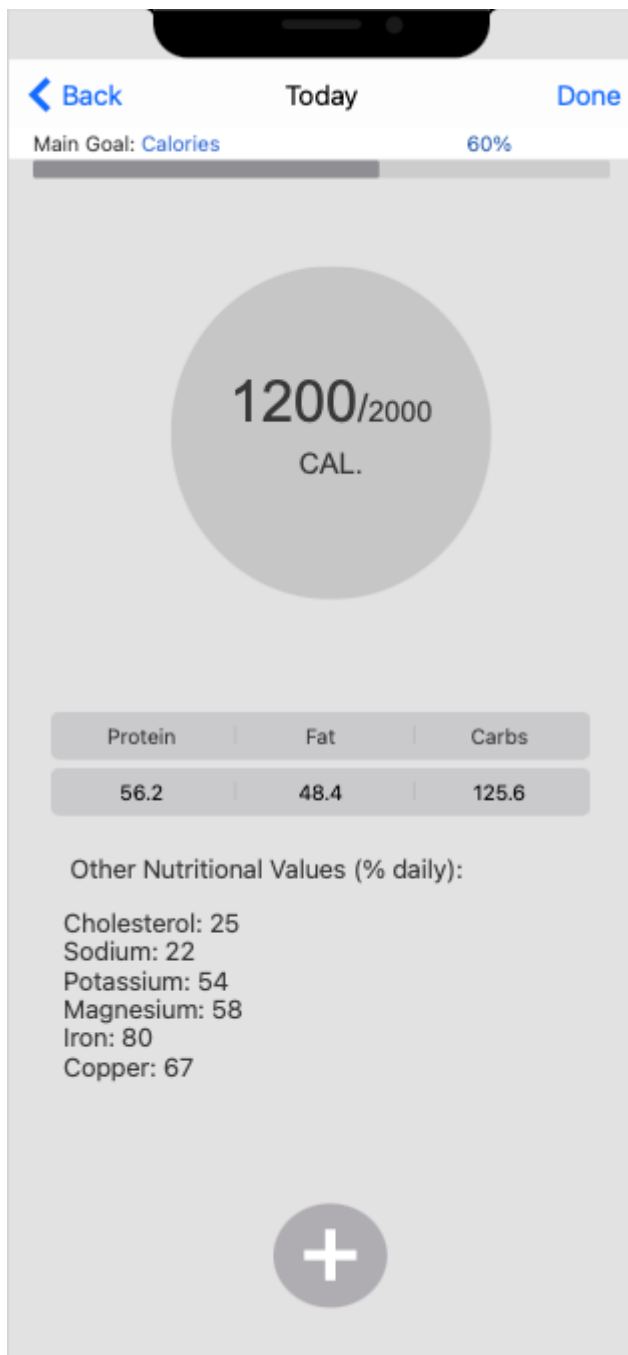
Create an account

By proceeding you also agree to the Terms of Service and Privacy Policy

This is a mobile app sign-in screen. At the top, there is a status bar showing the time 17:24, signal strength, Wi-Fi, and battery. Below this is a dark blue header with a back arrow and the text 'Sign in'. The main content area has a light blue background. It features the app name 'myCalorie Tracker' in bold. Below the app name are two white input fields with rounded corners, labeled 'Email*' and 'Password*'. Under these fields is a dark blue rounded button labeled 'Sign in'. Below the button is the text 'OR' in a small, light gray font. Below 'OR' is another dark blue rounded button labeled 'Create an account'. At the bottom, there is a line of small text: 'By proceeding you also agree to the Terms of Service and Privacy Policy'. A black horizontal line is visible at the very bottom of the screen, likely representing the home indicator bar.







1.

Test plan

Test Plan Identifier:

myCalorieTracker_TestPlan_0001

References:

The list of documents that are used to create the test plan are as follows:

Requirement Specification Document, Use Case Documents, Test Strategy

Introduction:

This document aims to test the functionality of the 'myCalorieTracker' to assure that the system meets all functional and non-functional requirements, satisfies the use case scenarios, and maintains the product's quality.

Test Items:

Testing should be done on the following items:

- iOS mobiles - iOS is minimum 10.0 and above.
- Android mobiles - OS 7.0 onwards.

Features To Be Tested:

The features to be tested are Login screen, Home screen, Results generated, and Bluetooth Connectivity.

Features Not To Be Tested:

The feature which is not tested in this release is creating custom diet plans.

Approach:

We are following the Agile Methodology in this project.

Pass/Fail Criteria:

All the application's critical functionality should work as proposed, and the pass percentage of test cases should be more than 96%, and there should not be any critical bugs.

Suspension Criteria:

If any critical functionalities are not working or the system experiences defects while login in or a Bluetooth connectivity issue, testing should suspend.

Test Deliverables:

The list of all test documents needs to be delivered is as follows:

Test Cases, Bug Report

Entry Criteria:

- The requirements document should be available.
- Complete understanding of the application flow is required.

Exit Criteria:

- Test case documents should be reviewed and approved.
- All critical Test Cases should be passed
- Validate and fix all the high priority bugs

Environmental Needs:

Smart scanner devices, databases, mobile devices, and food are needed for a test environment.

Responsibilities:

Test Lead should prepare a test plan. Testers should write test cases and execute the test cases.

Schedule:

The time required to perform each task should mention as follows:

Test cases document – 40-man-hours

Test cases execution – 100-man-hours

Defect report – 30-man-hours

Approvals:

The project manager should agree on the completion of the Test cases document and Bug report.

Test Cases

Test case 1:

Project Name	myCalorieTracker		Created by	Team 5 - Sukhada Kanetkar				
Project Module	Login		Date of creation	12/5/20				
Reference Document	Test Plan		Review date	12/7/20				
Test Case ID	Test case Scenario	Test Case	Pre-requisite	Test Case Steps	Test Data	Expected Output	Actual Output	Test Case Result
TC_login_001	Verify the user login	Enter valid username and valid password	User needs a valid account to login	1. Enter a valid username 2. Enter correct password. 3. Click on login button.	<valid_username> <valid_password>	User should be login successfully		
TC_login_002	Verify the user login	Enter valid username and invalid password	User needs a valid account to login	1. Enter a valid username 2. Enter incorrect password. 3. Click on login button.	<valid_username> <invalid_password>	User should not login and error message should be displayed "Incorrect username/password"		
TC_login_003	Verify the user login	Enter invalid username and valid password	User needs a valid account to login	1. Enter a invalid username 2. Enter correct password. 3. Click on login button.	<invalid_username> <valid_password>	User should not login and error message should be displayed "Incorrect username/password"		
TC_login_004	Verify the user login	Enter invalid username and invalid password	User needs a valid account to login	1. Enter a invalid username 2. Enter incorrect password. 3. Click on login button.	<invalid_username> <invalid_password>	User should not login and error message should be displayed "Incorrect username/password"		

Test Case 2

Project Name	myCalorieTracker		Created by	Team 5 - Sukhada Kanetkar				
Project Module	Personal Information		Date of creation	12/5/20				
Reference Document	Requirements, Test Plan		Review date	12/7/20				

Test Case ID	Test case Scenario	Test Case	Pre-requisite	Test Case Steps	Test Data	Expected Output	Actual Output	Test Case Result
TC_PersInfo_001	Verify personal information	Enter valid personal information	1. Login with valid username and password 2. On main window, click on profile icon. 3. On Profile screen, click on "Personal Information"	1. Enter valid email_id. 2. Enter all mandatory fields Name, Age, Gender. 3. Enter health condition if required. 4. Click on "Save" button to save the information.	<valid email_id> <Name, Age, Gender> <Health disease>	Information should be save successfully		
TC_PersInfo_002	Verify personal information	Enter valid personal information	1. Login with valid username and password 2. On main window, click on profile icon. 3. On Profile screen, click on "Personal Information"	1. Enter valid email_id. 2. Enter all mandatory fields Name, DOB, Age, Gender. 3. Enter health condition if required. 4. Click on "Cancel" button to save the information.	<valid email_id> <Name, DOB, Age, Gender> <Health disease>	User should be navigate to previous screen, Profile screen.		
TC_PersInfo_003	Verify mandatory fields	Keep the mandatory fields blank.	1. Login with valid username and password 2. On main window, click on profile icon. 3. On Profile screen, click on "Personal Information"	1. Enter valid email_id. 2. Do not enter anything in mandatory fields. 3. Enter health condition if required. 4. Click on "Save" button to save the information.	<valid email_id> <Health disease>	Error message should be display "Enter all mandatory (*) fields"		
TC_PersInfo_004	Verify updating the personal information	Update some personal information	1. Login with valid username and password 2. On main window, click on profile icon. 3. Personal information needs to filled already. 4. On Profile screen, click on "Update Personal Information".	1. Change email_id. 2. Change health condition. 3. Click on "Update Information" button	<valid email_id> <Health disease>	Information should be update successfully		

Test Case 3:

Project Name	myCalorieTracker		Created by	Team 5 - Sukhada Kanetkar				
Project Module	Bluetooth_Connectivity		Date of creation	12/5/20				
Reference Document	Requirements, Test Plan		Review date	12/7/20				

Test Case ID	Test case Scenario	Test Case	Pre-requisite	Test Case Steps	Test Data	Expected Output	Actual Output	Test Case Result
TC_BluetoothConn_001	Verify the bluetooth connectivity between smart scanner and myCalorieTracker	Connect the mobile to smart scanner through bluetooth	1. Needs to install application on mobile. 2. Needs to login myCalorieTracker application. 3. Device needs to be paired	1. On application home screen, Click on connect device.	<Smart scanner bluetooth >	Smart scanner bluetooth option should be available		
TC_BluetoothConn_002	Verify the bluetooth connectivity between smart scanner and myCalorieTracker	Connect the mobile to smart scanner through bluetooth	1. Needs to install application on mobile. 2. Needs to login myCalorieTracker application	1. On application home screen, Click on connect device. 2. ApplicationTurn on the bluetooth option on mobile. 3. Click on Smart scanner to connect.	<Smart scanner bluetooth >	Smart scanner and mobile should be connect successfully		

Test Case 4:

Project Name	myCalorieTracker		Created by	Team 5 - Sukhada Kanetkar				
Project Module	Measure Components		Date of creation	12/5/20				
Reference Document	Requirements, Test Plan		Review date	12/7/20				
Test Case ID	Test case Scenario	Test Case	Pre-requisite	Test Case Steps	Test Data	Expected Output	Actual Output	Test Case Result
TC_MeasureComponents_001	Verify the food components measurement	Scan the food with smart scanner	1. Needs to install application on mobile. 2. Needs to login myCalorieTracker application. 3. Connect the smart scanner and application via bluetooth.	1. Scan and weigh the food with smart scanner.	<food - Apple>	Food components i.e. calorie count, nutrients from the apple should be display on the mobile application.		
TC_MeasureComponents_002	Verify the food components quantity is showing accurately.	Scan the food with smart scanner	1. Needs to install application on mobile. 2. Needs to login myCalorieTracker application. 3. Connect the smart scanner and application via bluetooth.	1. Smart and weigh the food with scan scanner. 2. Verify the resulted values with the values present in the database manually	<food - Apple>	The resulted output values should be match with the values calculated from the database using algorithm table		
TC_MeasureComponents_003	Verify the daily calorie intake data saving automatically in the application	Scan the food with smart scanner	1. Needs to install application on mobile. 2. Needs to login myCalorieTracker application. 3. Connect the smart scanner and application via bluetooth.	1. Scan and weigh the food with scan scanner. 2. On home screen, select "Automatically save result" checkbox option.	Day 1: <food - Apple> <food - Bread > Day 2: <food - milk >	The number of calories measured for each food and for every day should be save automatically and correctly		

Transition strategy

The “myCalorieTracker” application will have short term and long-term business strategy depending on the technology advancement in the market, but the conversion strategy will be implemented through the path that has low risk, medium cost, and takes short time.

Meanwhile, the policies, measurement, and standard procedures will be adapted during the process of the application.

Conversion style:

- Direct style:** Since we start the system from scratch, therefore we will follow the direct style where the risk is high, the cost is low, and it takes short time to figure out.
- Pilot location:** The conversion location has low risk, medium cost, and medium time consumption as well.

- c. **Whole system module.** The conversion module is a high risk, has a medium cost and the time is short.

Technical Issue:

- a. **Hardware:** The application server will be installed and managed overcloud.
- b. **Software:** Server installation and computer software for the system and iOS or OS 7.0 onward mobile applications are required for users.
- c. **Data:** The system database will store the data after it is processed by the actors.

Conclusion:

The system would help users to stay healthy by allowing them to track their calorie intake.

The system is also integrating a smart scanner to help users keep track of their diet.

The introduction of smart scanners has created a new business need and we believe that it is going to benefit the users. Solving the business need has given us a business opportunity to solve a problem and would subsequently increase our earnings and profit.