Use Case For Health Track

Use Case: <1> <customer A register a new account in Fitness APP>

CHARACTERISTIC INFORMATION
Goal in Context: <The new customer is trying to register a new account in the Fitness APP, so he/she will be able to use the functions that requires the stored information, etc. and store the customer data into the related account database>
Scope: <The logics behind the button "register", including save password, password style validation, hash encryption, existing account validation, write into the database, etc.>
Level: < Primary task >
Primary Actor: <customer>
Channel to primary actor: <database>
Supporting Actors: <None>
Channel to Secondary Actors: <None >

PRE-CONDITIONS, END-CONDITIONS, TRIGGER
Preconditions: <The Fitness APP's front end is deployed and connected to Database level process and its APIs are active waiting for call>
Success End Condition: <Successful registration>
Failed End Condition: <customer existed, Password not matching the rules, Account linked email is locked>
Trigger: <None>

SCENARIOS – EXTENSIONS - VARIATIONS
Main Scenario
<step 1> customer have the app installed.
<step 2> customer click the register signal and ask the system led to the information page
<step 3> System asks for (email/phone num) and set password
<step 4> customer input the required information
<step 5> System conducts the information validation process (account/password check)
<step 6> System feedback of register status (success/failure + reason)

Extensions: <put here there extensions, one at a time, each referring to the step of the main scenario>

2a. <lost page connection>
2a1. <System present "Ooops Page lost" with a 'go back' arrow>

2a2. <customer click 'go back' >
2a3. <System jump to the previous page >

Variations: <put here the variations that will cause eventual bifurcation in the scenario>
<step or variation # > <list of sub-variations>
<step or variation # > <list of sub-variations>

1.
   a. *<a variation of step 1>*
   b. *<a different variation of step 1>*
   c. *<a different variation of step 1>*

RELATED INFORMATION (optional)
Priority: <how critical to your system / organization>
Performance Target: <the amount of time this use case should take>
Frequency: <how often it is expected to happen>
Superordinate Use Case: <optional, name of use case that includes this one>
Subordinate Use Cases: <optional, depending on tools, links to sub.use cases>

OPEN ISSUES (optional)
<list of issues about this use cases awaiting decisions>

SCHEDULE
Due Date: <date or release of deployment>
...any other schedule / staffing information you need...

Use Case For Fitness APP

Use Case: <2> <customer A can track his/her nutrients and food consumption in Fitness APP>

CHARACTERISTIC INFORMATION

Goal in Context: <The customer just ate breakfast and wants to input the calorie count of the food she ate. She will do this by scanning the barcode on the food item she ate.>

Scope: < The frontend and backend will be used. The Journal Page will be displayed to the customer on the frontend. This is where the customer will select to add a breakfast item to their journal. The backend will be used to match the barcode the customer scanned to a barcode stored in our database. Once the barcode is matched, the calorie information of that item needs to be displayed and added to their Journal to count towards their calorie count.>

Level: < Primary task >

Primary Actor: <Customer>

Channel to primary actor: <frontend customer interface>

Supporting Actors: <None>

Channel to Secondary Actors: <None >

PRE-CONDITIONS, END-CONDITIONS, TRIGGER

Preconditions: <Customer has the app installed and is logged into the system. it is running on their device. They have eaten a food that they want to input the calories of.>

Success End Condition: <Barcode feature successfully scans and finds item in our food database. Food item is successfully logged in Journal and calorie count is added to their daily calorie intake tracker.>

Failed End Condition: <Food item is not properly logged.>

Trigger: <None>

SCENARIOS – EXTENSIONS - VARIATIONS

Main Scenario

<step 1> Customer has the app installed.

<step 2> Customer ate a food item for Breakfast and wants to input the calories of the meal into their Journal on HealthTrack.

<step 3> Customer opens the app and navigates to the Journal page.

<step 4> Customer clicks “add” under the Breakfast section of the Journal page.

<step 5> Customer clicks the “scan barcode” button.

<step 6> Customer takes a picture of the barcode on their food item.
<step 7>System analyzes barcode in the picture and searches for it in our food database in the backend.
<step 8> System finds a match for the barcode and fetches the product information, including name and calorie count.
<step 9> System logs the item in the customer’s Journal history.
<step 10> System adds the food’s calorie count to the customer’s current calorie intake, to update how many calories the customer has eaten today.
<step 11> System displays the updated Journal page which now has the food item they scanned listed and the customer’s updated calorie intake.
<step 12> System displays the updated Journal page which now has the food item they scanned listed.
<step 13> customer exits this Journal section of the app

Use Case For Fitness APP

Use Case:

<3> <customer A earns a badge for completing their Daily Calories Burned Goal>

CHARACTERISTIC INFORMATION

Goal in Context: <The customer has completed burning their selected daily calorie goal, and the system must reward her with a badge.>

Scope: <The frontend and backend systems will be used. The backend will be used to check what the customer's saved daily calorie goal is, and will also be used to save the badge in their account. The badge will be displayed on the front end and can be accessed by the customer at any time.>

Level: < Primary task >

Primary Actor: <customer>

Channel to primary actor: <Frontend>

Supporting Actors: <None>

Channel to Secondary Actors: <None >

PRE-CONDITIONS, END-CONDITIONS, TRIGGER

Preconditions: <customer has the app installed and is logged into the system. it is running on their device. The customer has burned the amount of calories they have set their Daily Calories Burned Goal to.>

Success End Condition: <Badge is awarded to customer>

Failed End Condition: <Badge is not awarded to the customer>

Trigger: <None>

SCENARIOS – EXTENSIONS - VARIATIONS

Main Scenario

<step 1> customer selects and clicks start on a workout video on the app

<step 2> The system displays the video

<step 3> customer exits the video

<step 4> System calculates calories burned and records them

<step 5> System compares to the target for a badge and assigns a badge if the goal is reached

<step 6> customer exits this workout section of the app

Use Case For Fitness APP

Use Case 4: <Customer can get a custom diet routine by inputting information about their lifestyle and body type>

CHARACTERISTIC INFORMATION

Goal in Context: <The customer wants to get a custom meal plan made by nutrition experts in order to help them lose weight. >

Scope: <Both frontend and backend are required. On the frontend, customers will input details about their body and weight goal in order to customize their plan. The backend has a database that stores nutricion specifications from nutrition experts that the plan will be based on. The backend will also be used to store the custom plan once it is made.>

Level: < Primary Task >

Primary Actor: <Customer>

Channel to Primary Actor: <Frontend user interface>

Supporting Actors: <None>

Channel to Secondary Actors: <None>

PRE-CONDITIONS, END-CONDITIONS, TRIGGER

Preconditions: <The customer has the app installed and is logged in. The customer must be a subscriber, as this is a premium feature.>

Success End Condition: <Customer receives a custom meal plan.>

Failed End Condition: <Customer does not receive a custom meal plan.>

Trigger: <None>

SCENARIOS – EXTENSIONS - VARIATIONS

Main Scenario

<step 1> customer has the app installed.

<step 2> Customer navigates to home page

<step 3> Customer selects the “custom meal plan” button

<step 4> System displays a survey that prompts the customer to input information like weight, age, food preference (vegetarian, keto, etc.)

<step 5> Customer inputs the necessary information

<step 6> System accepts information, references database of nutrition specifications, and goes through an algorithm to create a custom plan

<step 7> System displays custom plan and saves it in User table

Use Case For Fitness APP

Use Case 5: <Customer sets up Daily Reminders in HealthTrack app>

CHARACTERISTIC INFORMATION

Goal in Context: <The customer is trying to set up which daily reminders they would like to receive. These reminders are motivational ones to remind them to complete their goals or input their food calorie information for the day.>

Scope: <This requires the frontend and backend. The customer has to use the user interface to select which notifications they would like to receive using toggles. Once their selections are made, this will be saved in the backend in the user table. Their selections have to be reflected on the frontend in the settings page. >

Level: < Primary task >

Primary Actor: <customer>

Channel to primary actor: <frontend user interface>

Supporting Actors: <None>

Channel to Secondary Actors: <None >

—---

PRE-CONDITIONS, END-CONDITIONS, TRIGGER

Preconditions: <The customer has the app installed and is logged in.>

Success End Condition: <Reminder was set successfully>

Failed End Condition: <Reminders did not save>

Trigger: <None>

SCENARIOS – EXTENSIONS - VARIATIONS

Main Scenario

<step 1> customer has the app installed.

<step 2> Customer navigates to the settings page by clicking on the gear icon on the home page

<step 3> Customer clicks “Reminders” tab

<step 4> Customer toggles on the reminders they want to receive.

<step 5> Customer clicks “save” button

<step 6> System saves the customer’s settings in the user table.

Use Case For Fitness APP

Use Case 6: <Customer can post a question to the Forum page.>

CHARACTERISTIC INFORMATION
Goal in Context: <The customer has a question about his current workout regimen and wants advice from community members. He wants to post the question on the forum to clear his doubts. >
Scope: <The frontend is required for the customer to make the post and the backend is required for the forum framework.>
Level: <Secondary Task >
Primary Actor: <customer>
Channel to Primary Actor: <Frontend user interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

PRE-CONDITIONS, END-CONDITIONS, TRIGGER
Preconditions: <The customer has the app installed and is logged in. >
Success End Condition: <The user is able to write a post and upload it to the Forum page.>
Failed End Condition: <The post is not able to upload.>
Trigger: <None>

SCENARIOS – EXTENSIONS - VARIATIONS
Main Scenario
<step 1> customer has the app installed.
<step 2> Customer navigates to Forum page
<step 3> Customer clicks “write a post” button
<step 4> Customer writes out their post and click “post”
<step 5> System uploads the post and displays it on the frontend. System also saves the post in the customer's post history.

Use Case For Fitness APP

Use Case: <7> <customers can change their preference and get relevant contents from the recommendation system>

CHARACTERISTIC INFORMATION

Goal in Context: <The customer can clear his/her doubts and concerns about workout or nutrition which experts can help solve..>

Scope: <Both frontend and backend are required, need to save the queries and resolution from the experts in the database. >

Level: < Secondary Task >

Primary Actor: <customer>

Channel to Primary Actor: <Interactive,UI>

Supporting Actors: <Experts>

Channel to Secondary Actors: <Interactive, UI>

PRE-CONDITIONS, END-CONDITIONS, TRIGGER

Preconditions: <The customer should ask the question in concise manner and provide all the necessary information accurately.>

Success End Condition: <customer get satisfactory response to his/her queries.>

Failed End Condition: <No expert available to help.>

Trigger: <Upon posting queries on community channel.>

SCENARIOS – EXTENSIONS - VARIATIONS

Main Scenario

<step 1> customer has the app installed or visits the correct website link.

<step 2> customer posts question in understandable manner.

<step 3> System sends these data to the next available expert.

<step 4> Experts replies to the query with satisfactory response.

<step 5> customer can continue to ask his/her doubts on the community channel