Software Requirements Specification

For

HealthTrack Fitness App

Submitted by

Team 1

IS663 - Professor Vassilka Kirova
NJIT
Longchao Da, Winona Patrick, Rohan Chopra, Khushali Sheth, Rutva Gandhi, Hao Mei, Vinoothna Chowdary Yarlagadda
Version: 0.1.0
Date Submitted: 2023-03-08

Table of Contents

1 Introduction

1.1 Purpose

This document serves as a comprehensive Software Requirements Specification (SRS) for our product, HealthTrack. The goal of this section is to provide a detailed and thorough understanding of the requirements for the development of this product.

In Section 1, we will provide an overview of the project, including its purpose, scope, and relevant definitions, acronyms, and abbreviations.

In Section 2, we will dive deeper into the product details. This section will cover the product perspective, including its relationship to other systems and technologies, as well as the functional and non-functional requirements for the product. We will also discuss the customer characteristics and constraints, as well as any dependencies that may impact the development of the product.

In Section 3, we will present the specific requirements for the product. This section will be broken down into three sub-sections: external interface requirements, functional requirements, and non-functional requirements. We will provide a detailed description of each requirement, including any relevant technical specifications and constraints.

Finally, we will discuss about current stage possible open issues in a separate section. This section will identify any potential challenges or concerns that may arise during the development of the product and outline possible solutions to address them.

Overall, this SRS aims to provide a comprehensive understanding of the requirements for the development of HealthTrack. By outlining the product's purpose, scope, and features, as well as its specific functional and non-functional requirements, we aim to ensure that the development team has a clear understanding of the project's goals and can develop a product that meets the needs of our customers.

1.2 Scope

HealthTrack is a subscription-based mobile fitness app, priced at $9.99 USD a month. The HealthTrack app was designed to maximize a customer’s productivity towards their training and weight loss goals. The system has several features that help a customer work towards a healthier body and lifestyle. It can calculate the maximum calories a customer can eat daily in order to achieve their weight loss goal, track their daily calorie intake, track how many calories they burn daily, and send them push notifications to remind them of their daily goals. There will also be in-app home workout videos led by expert trainers. HealthTrack will have community features as well. Global challenges will be posted regularly to motivate customers and encourage community engagement. Leaders of each challenge will earn FitPoints, HealthTrack’s in-app currency, that can be used to purchase real items from the in-app shop. They can also use their own real money to purchase items.

There will be a forum feature that provides educational articles written by in-house experts and customers can also make their own posts to ask questions or share advice. The app will have an intuitive and sleek UI/UX design that is simple to navigate and use.

1.3 Definitions, Acronyms, and Abbreviations

There are no definitions, acronyms, or abbreviations at this time.

1.4 References

[1] statista information on the market dominance of other companies that are doing similar products related to exercise and health APPs. https://www.statista.com/study/74030/wearables-fitbit-in-the-united-states-brand-report/

[2] Annual report of APPLE

https://d18rn0p25nwr6d.cloudfront.net/CIK-0000320193/7b5717ca-6222-48e6-801c-9ea28feeef86.pdf

[3] Doe J. Recommended practice for software requirements specifications (ieee)[J]. IEEE, New York, 2011.

[4] Boehm B W. Software engineering[J]. IEEE Trans. Computers, 1976, 25(12): 1226-1241.

1.5 Overview

In this document, we aim to provide a comprehensive Software Requirements Specification that covers all aspects of our product. To achieve this, we have divided the document into two main sections: Overall Description and Specific Requirements.

The Overall Description section is intended to provide a high-level overview of the product, which will help stakeholders understand the scope of the project. In this section, we will briefly list six key items that define our product, including its purpose, features, and target audience.

Moving on to the Specific Requirements section, we will delve into the details of the product's functional and non-functional requirements. These requirements have been grouped into four categories: External Interfaces, Functional Requirements, Non-Functional (quality) Requirements, and Logical Database Requirements. Each category will contain a detailed list of requirements that must be met to ensure the success of the project.

In addition to the above categories, we have included some optional descriptions that will provide further insight into the product. These descriptions cover a range of topics, including customer interface design, performance metrics, and system architecture.

Overall, this Software Requirements Specification is designed to be a comprehensive and detailed document that will guide the development of our product. By following the requirements outlined in this document, we aim to create a high-quality software product that meets the needs of our customers.

2 Overall Description

2.1 Product Perspective

In a bid to foster a culture of wellness and healthy living among individuals, we have designed the HealthTrack APP, an innovative cross-platform solution that caters to the needs of Android and IOS customers. Our cutting-edge product is primarily focused on tracking customers' daily calorie intake for each meal, while providing a scientific monitoring mechanism based on individual health metrics such as weight, height, age, and other vital parameters.

Our state-of-the-art data tracking system is aimed at providing customers with real-time information on their daily dietary requirements, enabling them to make informed decisions on their meal choices.

Our ultimate goal is to leverage our advanced technology and system to enable more people to adopt healthier lifestyles. The HealthTrack APP offers a seamless and intuitive customer experience, making it easier for customers to stay on track with their health and fitness goals.

We firmly believe that the HealthTrack APP is a game-changer in the health and wellness industry, and we are committed to continually updating and improving our product to ensure that it remains at the cutting edge of the industry. Our mission is to empower individuals to take charge of their health, one meal, and one workout at a time.

With the HealthTrack APP, individuals can monitor their progress, track their daily caloric intake, and receive personalized meal plans that are tailored to their unique needs. Our product is designed to provide a holistic approach to health and wellness, making it an essential tool for individuals who are serious about living healthier and happier lives.

We are confident that the HealthTrack APP will revolutionize the health and wellness industry and help millions of people worldwide to achieve their health and fitness goals.

2.2 Product Functions:

- The app will calculate the maximum amount of calories customers can consume daily to safely reach their weight goal as fast as possible.
- customers will log what they eat and the app will track the calories.
- App provides workout videos led by experienced trainers in a variety of workout categories (ex: treadmill, weights, yoga, etc.).
- Each workout video has an estimated amount of calories that will be burned. The app tracks the amount of calories burned based on which workout videos were watched.

- customers can manually input an estimate of how many calories they burned in the day if they exercised **without using the app's videos**. (all the possible exercise ways)
- customers are awarded badges for goals they complete.
- Community features:
    - Challenges are posted regularly. Leaders in each challenge earn FitPoints that can be used to buy items in the in-app shop.
    - Forum for customers to interact with each other, ask questions, post recipes, and more.
    - Transformation page, where customers can share their body transformations with the app community or keep them private.
- Education by on-staff nutritional experts will be posted regularly.
- customers will get notification reminders to exercise and input their food calories for the day.

customers are expected to input:

- Their weight loss goal
- How many pounds they want to lose per week
- Weight
- Age
- Height
- Gender

Based on this data, the app will calculate the maximum amount of calories customers can consume daily to safely reach their weight goal as fast as possible. They can also use the food log feature, which tracks the food they eat and how many calories they contain. customers can watch the in-app home workout videos, which automatically contribute to the calories burned tracker. Or, if they exercise without using in-app videos, they can add an estimate of how many calories they burned manually. There are community features for customers to feel connected and motivated as well. Challenges will be posted regularly, and customers on top of the leaderboard will earn FitPoints, which can be used to buy items in the in-app shop. There will be a forum for customers to post questions, share advice or recipes, and more. There will be a Transformation Story page where customers can also share pictures of their transformations with the app community if they choose to. They can also keep their pictures private.

2.3 User Characteristics

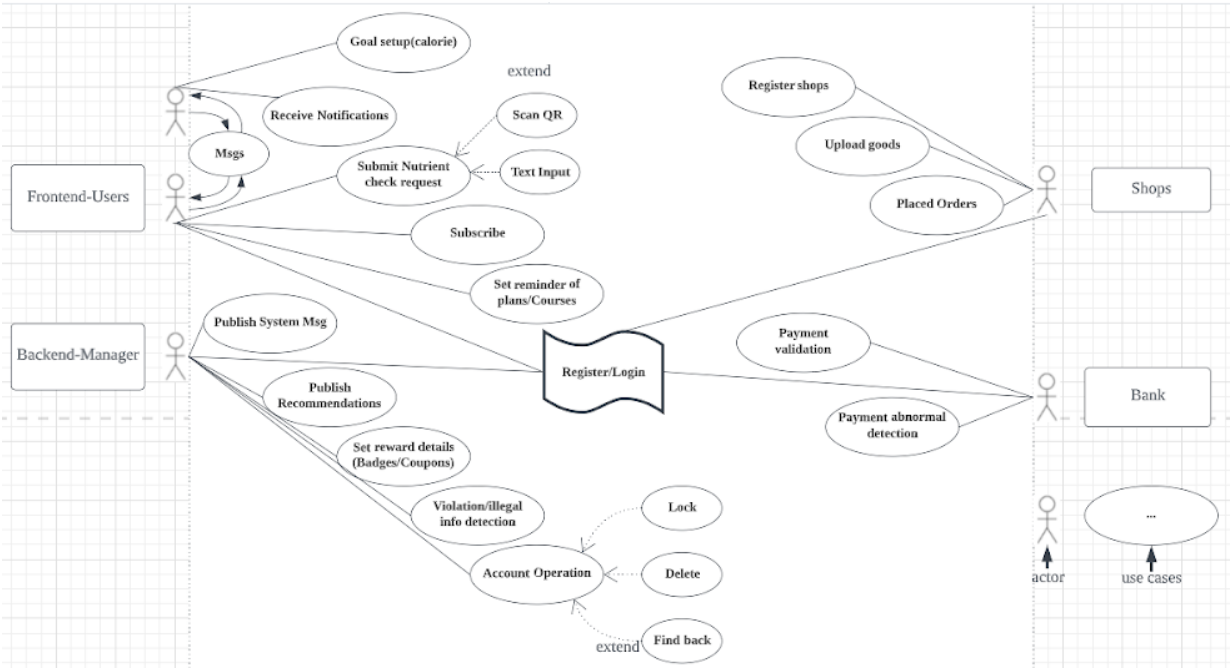
Users (actors) of the app:

- Customers: consumers who download and use HealthTrack.
- Backend managers: Develop and maintain HealthTrack’s databases.
- Bank: HealthTrack will integrate a financial API, like Plaid, to enable in-app purchases.

- Shop: HealthTrack will partner with an athletic retail company to give customers the ability to win items with FitPoints (credits earned in-app) or purchase items using money.
- Expert: We will have experts on staff who write articles for our app.

Objectives/activities of customers (actors) of HealthTrack:

- Customers should be able to:
    - Create an account and securely log in
    - Securely pay for HealthTrack's monthly subscription plan
    - Purchase items in-app from integrated shop partner
    - Submit all necessary personal data (height, weight, age, gender, weight goal) and receive daily calorie intake goal
    - Submit daily calorie burn goal and earn badges when goal is achieved
    - Log daily calorie intake either manually or search for food items stored in app database
    - Watch in-app workout videos
    - Post and comment on in-app customer forum
    - Read in-app articles
- Backend managers will:
    - Conduct customer account management
    - Protect data by developing data security and restoration policies, procedures, and controls
    - Create, maintain, test, and debug backend
- Shop will:
    - Upload their items for sale
    - Manage placed orders
- Bank will:
    - Verify bank accounts
    - Conduct payment validation
    - Ensure secure transactions
- Expert will:
    - Write articles and upload them to the app using their own interface that we provide them with.



2.4 Constraints

Some features in the HealthTrack app will only be accessible by customers who are paid subscribers. If customers click on features like Custom Meal Plan, Custom Exercise Plan, or Barcode Scan (for calorie tracking), if the customer is not a subscriber, a popup will appear that constrains them from accessing the feature. The popup will state that this is a locked feature and they have to subscribe first. There will be a list of subscription benefits and a button that says “Subscribe” if the customer decides to subscribe. If they do not want to subscribe they can click the exit button and close the popup, but they will not be able to access these features.

2.5 Assumptions and Dependencies

2.5.1 Assumptions

It is assumed that required networks will be available and functioning as normal. It is assumed that the customer has access to the Internet in order to download and use the application.

2.5.2 Dependencies

In order for the in-app shop feature to work, the financial institution API must be fully functional, secure, and available at all times. customers must have an email address in order to register for the application.

2.6 Allocation of Requirements

For detailed allocation of requirements, please refer to our PBL

3 Specific Requirements

3.1 External Interfaces

Communications Interface: Wi-Fi or cellular data is required to use the app. The app will use HTTP for communication with servers and other web services. The app will utilize the networking hardware of the customer's device through network communications services provided by the Android or iOS operating systems.

Software Interface:

- **Frontend user interface:** This is the app's user interface that is accessible by customers.
- **Backend**: All customer information will be stored in a relational database. There will be an interface for backend managers to manage accounts and view inputted customer information like weight loss goal, height, age, etc.
- **Shop interface**: HealthTrack will partner with an athletic retail company and integrate its products into our in-app shop. We will design an interface for the retail partner so they can upload items from their database and handle orders.
- **Expert's interface**: We will create an interface for our on-staff experts so they can post the articles they have written for our app.
- **Bank:** The app will connect to an external financial institution's API for in-app transactions.

Hardware Interface: The app works on Android and iOS mobile devices. No other hardware is required. The app will be available for download on the App Store and Google Play Store.

3.1.1 Data Interface

Customer Personal Data: The customer Personal Data contains customer's personal information like age, weight, weight loss goal, gender, and height.

Exercise Related Data: The customers will be using our app to conduct daily exercise by following the instructions provided from tutorial videos and recording exercise information. The video type that they watch, video length, how long does one's exercise last, etc.

Calorie Data: The app will collect the customer's calorie intake for each food item they consume in the day, as well as the amount of calories they burned by exercising. It will also collect what their daily calorie burn goal is.

Orders Related Data: This data is generated from the customer's order action in the in-app shop. Once they place an order, the information will be recorded and logged into the database, like the order number, order time, order cost, the linked product, responsible shop, etc.

Subscription Data: The subscription data generated from the subscribing action, once a customer subscribes to our app, the order will be placed and the app will send a request to the financial institution API's side, to validate the account and perform the transaction. Subscription data will then be saved into the database, including but not limited to: subscription date, subscription length, etc.

3.1.2 Customer Interface

Opening page: Upon opening the app, there will be an opening page with the HealthTrack title and logo. If the customer is not already logged in, there will also be an action button to register (redirects customer to registration page) and an action button to login (redirects customer to login page). The customer will click the register button if they do not already have an account. If the customer does have an account, they can click the login action button. If the customer is already logged in, just the title and logo will be displayed for a brief moment, before redirecting to the homepage.

Registration page: This page is where the customer will create their account. There will be four input boxes. One for the customer's name, one for their email, one for a password, and one for them to retype the password. There will be a "create account" action button for them to click once they input all the information. If the account does not already exist and the passwords match, the button will redirect them to the first-time-customer survey page. If not, the form will not go through and the page will flash an error message, example: "Passwords do not match".

There will also be a "Already have an account? Log in" action button if the customer already has an account and did not intend to register for a new account. This button will direct them to the login page.

Login page: This page is where customers will log in. There will be two input boxes, one for their email and one for their password. There will be a "Log in" action button that will direct them to the home page if their account is validated in the authentication process. If

the information is incorrect, the form will not go through and the page will flash an error message, example: “Your email or password is incorrect”.

There will also be a “Don’t have an account? Sign up” action button if the customer does not have an account and did not intend to be on the login page. The button will direct them to the registration page.

Net calorie goal survey pages: There will be two survey pages where customers will input information to calculate their maximum daily calorie intake in order to achieve their weight loss goal. The third page will display the calculation result.

The first survey page will have five input boxes. One to input their current weight, one to input their goal weight, one for their height, one for their age, and one for their gender. Once all of the information is entered, the customer can click on the “continue” action button, which will direct them to the next survey page.

The second survey page will only have one question: “What is your weekly weight loss goal?” There will be four options: 0.5 pounds per week, 1 pound per week, 1.5 pounds per week, and 2 pounds per week. Each option will have its own button, and the customer can click the button that has the weight loss goal they choose. Once they are certain of their choice, they can click the “continue button” to see the next page.

The third page will display the calculated daily net calorie goal based on the customer’s input. There will be a “continue” page that directs them to the calorie burn goal setting page.

Daily calorie burn goal page: This is the page where customers will set up a calorie burn goal based on how active they are or how active they would like to be. For example, if a customer wants to burn at least 300 calories daily by exercising, they would set that goal here. There are plus and minus buttons for customers to increase or decrease their goal in increments of 10. Once they are done setting up the goal, they can press “continue”, which will direct them to the home page.

Bottom tab bar/navigation bar: There will be a bottom tab bar/ navigation bar on the bottom of EVERY PAGE from this point (so, not including Opening, Registration, Login, and Survey pages). This navigation bar will have a tab for the Home page, Journal page, Exercise page, Forum page, and Shop page.

Home page: The top of the page will display the following statistics: maximum calorie intake number, how many calories have left for the day, and how many calories they have

burned from exercising in the day. This will provide them a quick snapshot of their exercise progress and how many calories they have left in order to avoid exceeding their calorie goal.

Under these statistics, there will be a section for recently uploaded recipes by our nutrition experts. This is a paid feature, so customers will only be able to access these recipes if they are a subscriber. If the customer is subscribed, they can click on the video graphic to be taken to the workout video. Or they can click "see more videos" to be taken to the Exercise page. If the customer is not a subscriber, a popup will appear that says that this is a locked feature. This popup will contain subscription benefit information and pricing. There will be a "subscribe to unlock" button that customers can click if they want to subscribe, or there will be an exit button if they do not.

There will be a settings icon on the top right hand corner of the home page. Clicking on this icon will take customers to a settings page.

Settings page: On this page, customers can customize their push notification settings based on what kind of reminders they would like to receive. They can also change their daily calorie burn goal here. There will also be a Privacy Center section where customers can read HealthTrack's Terms and Privacy Policy.

Forum page: This page is to provide education and community engagement for our customers. There will be an "Education" section with articles written by our nutrition and exercise experts. Clicking on the graphic or title of an article will direct the customer to that article. Or, customers can click the "see more articles" action button to be taken to the Education section of the Forum page. Only the recent four articles will be shown, customers will click "See more articles" to be dedicated to the full Education page.

Under the Education section there will be action buttons to "write a new post" or "view my posts". If customers click the "write a new post button", a page will popup up for them to write their post. They will then click "post" to post it. If the customer clicks "view my posts", they will be taken to a page that lists all of their previous posts. They can click on the post to read the entire post and read any responses from community members. There will be a back button so they can go back when they are done and see the rest of their posts.

Under these buttons, there will be a list of forum post previews. Consumers can sort the posts by "popular" or "recently posted" by clicking the "Sort by" dropdown and selecting an option. They can click on the forum post to be directed to a dedicated page for the post and view any responses. There will be two action buttons to comment on the post or upvote the post. There will be a back button so they can go back when they are done and see other posts.

Exercise page: This page is where all workout videos will be stored. At the top of this page, if there are any challenges being hosted by the app, there will be a section dedicated to it. They can click this section to be taken to a page that displays all challenge info and allows them to upload their challenge submission, view other community submissions, and view the leaderboard.

Back on the main Exercise page, under the Challenge section, there will be a button for each exercise category, example: weights, treadmill, etc. Consumers can click on any of these categories to be taken to a dedicated page that lists each video within that category. Consumers can click on a video to start it. Once the video has started, there will be a pause/play button and exit button.

Also on the main Exercise page, there will be an "Awards" section under the exercise category buttons. This will display all the badges a customer has earned for achieving certain goals, example: doubling their calorie burn goal.

In-app shop page: This page will feature all items sold by our retail partner. The main page will show new items, previews of each item category, and there will be a "cart" icon in the top right. Consumers can click "see more" next to each category to be directed to a page with that item. Example, if they click "see more" next to the t-shirt section, they will be redirected to a page that features all available t-shirts. They can click on an item to be taken to its dedicated product information page. On this page, there will be a picture of the item, product information, and an "add to cart" button. There will be two prices displayed, one in USD and one in FitPoints. If they do not have enough FitPoints to purchase the item, they will have to purchase it with USD through their bank account.

Consumers can click "add to cart" if they want to purchase the item. They can then click the "cart" icon on the top right, where the items in their cart will be listed and they can click the "purchase" button. Pressing this button will lead them to a page that asks for their address and payment information, which will be validated by the Financial API we use.

Journal page: This page is where customers will enter the calorie information for all the food they eat throughout the day and calories burned. The top of the page will display how many calories they have left for the day and how many calories they burned. There will be sections for Breakfast, Lunch, Dinners, Snacks, Exercise, and Water. Under each section there will be an "add" button.

If it is a food section, the "add button" will take them to a page that has two buttons: "scan barcode" and "enter calories". Scanning a barcode and accessing our food database is a premium feature. If the customer is not a subscriber, a popup will appear that says that this

is a locked feature. This popup will contain subscription benefit information and pricing. There will be a “subscribe to unlock” button that customers can click if they want to subscribe, or there will be an exit button if they do not. If they are a subscriber already, they will be able to scan the barcode with their camera and the calories of that item will be added to their calorie tracker automatically. Entering the calories of their food manually is free. Consumers can click the “enter calories” button and input the number of calories in their food item. This number will be added to their calorie tracker.

If it is the exercise section, the “add” button will take them to a page where they can select the type of exercise they did, the length of the workout, and how many calories they estimate they burned. This calorie burn number will be added to their calories burned tracker.

3.2 Functional Requirements

Use Case 1: <Register>

CHARACTERISTIC INFORMATION
Goal in Context: <The new customer is trying to register a new account in the Health Track 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: <Customer has the app installed and does not have an account already.>
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 clicks the “Get started” button on opening screen of the app
<step 2> System displays the registration page
<step 3> Customer inputs their name, email, and password and clicks “create account” button
<step 4> System conducts the information validation process (account/password check). System saves the customer’s information into the database if the account passes validation requirements. System allows the customer to move forward and access the app.

Use Case 2: <Input nutrients>

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 navigates to the Journal page.

<step 2> System displays the Journal page

<step 3> Customer clicks "add" under the Breakfast section of the Journal page.

<step 4> System displays a page with two ways to enter nutrients: manually or by scanning a barcode.

Variations:

1. Customer selects the manual entry option.
    a. **<step 1>** Customer clicks "enter manually" button
    b. **<step 2>** System displays the manual entry page
    c. **<step 3>** Customer inputs an estimation of the amount of calories in their meal and selects "save"
    d. **<step 4>**System logs the item in the customer's Journal history. System adds the food's calorie count to the customer's current calorie intake, to update how many calories the customer has eaten today. System displays the updated Journal page which now has the food item they scanned listed and the customer's updated calorie intake. System displays the updated Journal page which now has the food item they scanned listed.
2. Customer selects the barcode scan option
    a. **<step 1>** Customer clicks "scan barcode"
    b. **<step 2>** System displays the barcode scan page.
    c. **<step 3>** Customer takes a picture of the barcode on their food item.
    d. **<step 4>** System analyzes barcode in the picture and searches for it in our food database in the backend. System finds a match for the barcode and fetches the product information, including name and calorie count. System logs the item in the customer's Journal history. System adds the food's calorie count to the customer's current calorie intake, to update how many calories the customer has eaten today. System displays the updated Journal page which now has the food item they scanned listed and the customer's updated calorie intake. System displays the updated Journal page which now has the food item they scanned listed.

Use Case 3: <Customer earns a badge>

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 user 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. 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. 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 4: <Custom meal plan>

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 nutrition 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. The plan will be displayed on the frontend 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>

Use Case 5: <Set up reminders>

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 >

Use Case 6: <Post to the Forum.>

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: <Primary Task >
Primary Actor: <customer>
Channel to Primary Actor: <Frontend user interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 7: <Subscribe>

CHARACTERISTIC INFORMATION
Goal in Context: <TThe customer subscribes for a premium account. >

Scope: <The frontend is required for the customer to subscribe and the backend is required to store that they have a premium subscription..>
Level: <primary Task >
Primary Actor: <customer>
Channel to Primary Actor: <Frontend user interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 8: <Manage articles>

CHARACTERISTIC INFORMATION
Goal in Context: <On-staff nutrition or exercise experts have their own interface in which they create/read/update/delete (CRUD) the educational articles they have written for our app. >
Scope: <The frontend is required for the expert to post their article.>
Level: <Primary Task >
Primary Actor: <Experts>
Channel to Primary Actor: <Expert's interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 9: <Manage goods>

CHARACTERISTIC INFORMATION
Goal in Context: <The retailer we partner with has its own interface for it to create/read/update/delete (CRUD) their goods for our in-app shop>
Scope: <The frontend is required for the expert to post the goods. The retailer's database which stores the goods' information is needed.>
Level: <Primary Task >
Primary Actor: <Shops>
Channel to Primary Actor: <Shop's interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 10: <Placed Orders>

CHARACTERISTIC INFORMATION
Goal in Context: <The retailer we partner accesses the orders our customers have placed on their interface. >
Scope: <The frontend is required for them to view the placed orders.>
Level: <Primary Task >

Primary Actor: <Shop>
Channel to Primary Actor: <Shop's interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 11: <Validates payment>

CHARACTERISTIC INFORMATION
Goal in Context: <The bank must validate payment made by the customer in the app. >
Scope: <The frontend is required for the expert to post their article.>
Level: <Primary Task >
Primary Actor: <Bank>
Channel to Primary Actor: <Bank's interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 12: <Publish system message>

CHARACTERISTIC INFORMATION
Goal in Context: <The backend manager publishes messages related to the system. For example, if the system will be down for maintenance, the backend manager will publish this message to the app. >
Scope: <The frontend interface will be used to write and publish the messages.>
Level: <Primary Task >
Primary Actor: <Backend Manager>
Channel to Primary Actor: <Backend interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 13: <Publish recommendations>

CHARACTERISTIC INFORMATION
Goal in Context: <The backend manager is given exercise and nutrition recommendations to provide to customers, and must publish them to the app >
Scope: <The frontend interface will be used to publish the messages.>
Level: <Primary Task >
Primary Actor: <Backend Manager>
Channel to Primary Actor: <Backend interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 14: <Set reward details (badges/FitPoints)>

CHARACTERISTIC INFORMATION
Goal in Context: <The backend manager is given a list of requirements for how customers can earn badges and FitPoints. The app will have special challenges in which customers can earn badges or FitPoints. For example, if the app hosts a challenge to burn 1000 calories during American Heart Month, the backend manager will have to implement this calorie number requirement within the app so the badge and/or FitPoints can be earned when they complete the challenge.>
Scope: <The backend interface will be used to set the details.>
Level: <Primary Task >
Primary Actor: <Backend Manager>
Channel to Primary Actor: <Backend interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 15: <Violation/Illegal info policies>

CHARACTERISTIC INFORMATION
Goal in Context: <Backend managers must upload policies for violations and illegal information practices. These policies will be provided to them after being drafted by legal staff and they have to publish them to the app and make sure they are enforced.>
Scope: <The backend interface will be used to publish the policies.>
Level: <Primary Task >
Primary Actor: <Backend Manager>
Channel to Primary Actor: <Backend interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 16: <Account operation>

CHARACTERISTIC INFORMATION
Goal in Context: <Backend managers must conduct account operation practices like locking an account, deleting an account, or unblocking accounts. If an account violates our legal policies, the account will be locked or potentially deleted. If customers want to appeal the account lock, the backend manager will conduct a review to see if the account should be unblocked.>
Scope: <The backend interface and user database will be used.>
Level: <Primary Task >
Primary Actor: <Backend Manager>
Channel to Primary Actor: <Backend interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 17: <Order Products>

CHARACTERISTIC INFORMATION
Goal in Context: <The frontend user could have the options to order Products on the online shops. They will be able to browse the product information in the shop and decide to add to the shopping cart and purchase items.>
Scope: <The Customer could save products to the cart and purchase, the order will be created>
Level: <Primary Task >
Primary Actor: <Customer>
Channel to Primary Actor: <Customer interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

Use Case 18: <Register Shops>

CHARACTERISTIC INFORMATION
Goal in Context: <The prospective shop owners could request to register for an online shop after they send a registration request, the system will arrange the request to a waiting list and proceed the automatical verification on the shop's enligibility>
Scope: <The Shop's interface and shop database will be used.>
Level: <Primary Task >
Primary Actor: < Shops>
Channel to Primary Actor: <Shop's interface>
Supporting Actors: <none>
Channel to Secondary Actors: <none>

3.3 Non-Functional (quality) Requirements

- **Performance:** The load time for each screen should take no longer than 4 seconds when the total number of simultaneous customers is greater than 50,000
- **Availability:** The system must be available to customers at all times.
- **Security:**
    - System grants access to accounts when customers enter the correct customer name and password.
    - Passwords must be hashed.
    - After 3 login attempts, the security system locks the account to protect the customer's information. To unlock an account, customers must do a password reset.
    - The system must be resilient to any kind of attacks.
- **Compliance:** The system must comply with all applicable laws and regulations.
- **Maintenance:** The system must be easy to maintain and update.
- **Durability:** Uploaded data and photos should not get lost.
- **Latency:** Photos should be fetched with minimal latency.

3.4 Logical Database Requirements

A basic relational database structure design as follow (note: due to the limited implementation time, we just provide a sample table with possible main key, foreign key to connect the relations between tables, and basic data type description, like Varchar/Decimal/Int)

1. **User Table**

```
CREATE TABLE customers (
user_id INT PRIMARY KEY,
first_name VARCHAR(50),
last_name VARCHAR(50),
email VARCHAR(100),
password VARCHAR(100),
gender ENUM('Male', 'Female', 'Other'),
age INT,
height DECIMAL(5,2),
weight DECIMAL(5,2),
goal_weight DECIMAL(5,2)
max_daily_calories DECIMAL(5,2)
```

reminder_settings Boolean(True/False)

login_session Boolean(True/False) #description: To remember the login status,

calorie_burn_goal DECIMAL(5,2),

);

This table will be used for the registration, login, survey, and calorie goal setting features. When a customer registers for an account, their email, name, userID, and password will be stored in this database. Their input for the survey, including age, height, weight, gender, and goal weight will be stored here as well. This will be used when calculating their maximum daily calorie intake, which will also be saved in this database. The customer will also set a daily calorie burn goal, which will be stored here as well, and will be displayed on the frontend on the Home page and Exercise page. The reminder settings information is for when customers set their preference for which reminders they want to receive.

2. **Meal Table**

CREATE TABLE meals (

meal_id INT PRIMARY KEY,

customer_id INT,

meal_name VARCHAR(100),

meal_description VARCHAR(255),

calories INT,

date DATE,

FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

);

This database stores information inputted on the Journal page. Consumers will input information about the food they eat throughout the day, and it will be saved here. This data will be fetched when customers want to look at their meal history on the Journal page.

3. **Exercise Table**

CREATE TABLE exercises (

exercise_id INT PRIMARY KEY,

customer_id INT,

exercise_name VARCHAR(100),

duration INT,

calories_burned INT,

date DATE,

FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

);

This is where the customer's exercise history will be saved. The customer can input what exercises they did, how long the workout session was, and how many calories they burned. The date will be set automatically. This log will be displayed on the frontend on the Exercise page.

4. **Personalized Exercise Plan Table**

CREATE TABLE exercise_plans (

plan_id INT PRIMARY KEY,

customer_id INT,

plan_name VARCHAR(100),

plan_description VARCHAR(255),

FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

);

This is for the premium personalized exercise plan feature. The plan will be curated by our experts and saved in this database. The name and descriptions will be displayed on the frontend on the Exercise page.

5. **Nutrition Table**

CREATE TABLE nutrition (

food_id INT PRIMARY KEY,

recommended_food VARCHAR(100)

food_to_avoid_info VARCHAR(200), # what should be avoided to intake together

FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

);

This will be used for the premium feature of custom meal planning. There will be a custom meal plan page with meals curated by our nutrition experts, and this data will be fetched for that page.

6. **Product Calorie Record Table**

```
CREATE TABLE product_calorie_record (
product_id INT PRIMARY KEY,
product_ code VARCHAR(500), # the info provided after scanning QR code
nutrition_gradients VARCHAR(500), # what it may contain and percentage/unit
);
```

This is for the premium barcode scanning feature. This database will store thousands of food items, and when customers scan the barcode on their food item, it will be in this database so the calories can automatically be added to their Journal calorie tracker. This data will be displayed in their Journal on the frontend.

3.5 Design Constraints

Software constraint: Software must be available on both iOS and Android.

Battery constraint: Battery life is a very limiting factor in mobile devices. System must be optimized to minimize its power footprint while still considering performance.

Memory constraint: Small payload should be utilized because compared to a desktop, mobile devices have small memory caches, which slows loading speeds down as the information needs to get reloaded.

Customer Interface (UI) constraint: Mobile devices are small. UI design must be appropriate for smaller screen sizes and should be adaptable to different orientations.

Network constraint: Mobile networks are slow compared to desktop speeds. Small payloads must be utilized, because the bigger the page and graphics, the longer it takes to load.

3.6 Standards Compliance

No particular standards compliance requirements have been identified

4 Other Requirements

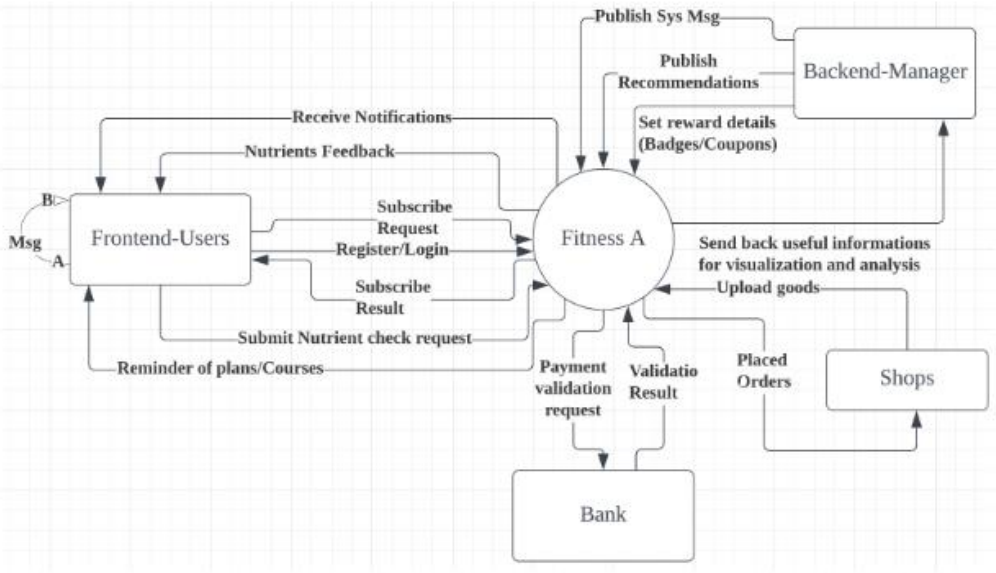
No other requirements have been identified.

5 Open Issues

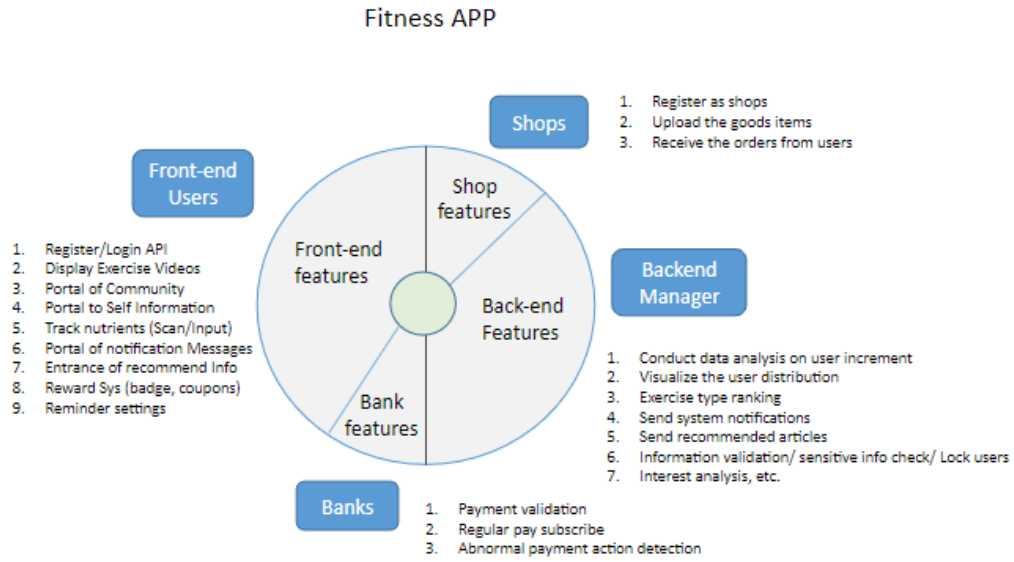
- No wearable exercise product connected (hard to record the detailed exercise information, hard to guarantee the exercise effect).
- Hard to get enough data to use as a training set for specialized customer recommendation models.
- We don’t have a testing group for version testing before the real official version1 release, which can not promise the system stability.

Appendix A: Supporting Analysis Information

1. High Level Context Diagram



2. Informal Overview Context Diagram



3. System Use Case Model