HealthTrack

Iteration 2 for Team 1

Software Engineering 3350
Members of **Team 1:**
Katrina Barkwell
Arden Raaen
Scott Woodcock
Khaled El-Salakawy
Wilson Tieu
Rashpal Robby Singh

Table of Contents

Vision Statement

HealthTrack: The Applet for your Diet

Summary: HealthTrack will be a diet health-tracking system that emphasizes meeting nutrition-related goals and getting enough essential nutrients in one's diet.

The HealthTrack system will be used to track and analyze the ongoing diet and health conditions of its users. Users will input goals and/or targets for various health related metrics, which can be used against real values from tracked diet to analyse and display results. The system will include the ability to track diet entries for each day, by searching for a food from a public database of foods, or adding foods to a custom list by entering details about the food item. The user will enter the number of servings consumed, and add the entry to a list of foods consumed for that day. The system will calculate and tabulate total nutrient intake for that day, and additionally create graphical representations of the intake statistics over the current week, month and year. Based on intake statistics, the system will have the ability to recommend foods to eat based on the remaining allowances for certain nutrient or diet composition limitations for the day. The system will also have the capability to alert users when their dietary restrictions have been exceeded.

The customer wishes to be able to track their "goal" nutritional values and if they have been meeting them on a weekly basis. The customer wants an interface that will bring them back to using the application and wants easy to understand, beautifully displayed statistical information that is not overwhelming to look at. In particular, the health values that the customer wishes to track are Calorie, protein, carbohydrate and fat intake. Any additional nutritional values to track (iron, sodium, etc.) will be a bonus. Thus, this application will be useful to those choosing to live a healthier lifestyle, those wishing to lose weight, and those that need to watch their specific nutrient intake for health reasons (eg. sodium, cholesterol).

The system is not a fitness tracking app, and therefore does not include workout entries or reminders of any kind. Instead a user will be able to input their age, height and weight, and the system will approximate the normal amount of calories required in a day. The user can also manually input this, if desired. HealthTrack does not need to be invasive on the user's android use-habits. In essence, the user does not want this application to "force" them to live this life style – it should be completely opt-in. That means no overriding the functions of the phone that could interfere with other applications. HealthTrack will rely on the user's input to track their diet. HealthTrack will also not perform any network activities – it will be completely usable without an internet connection.

While many competing products focus on calories consumed, and, ultimately, weight loss, this app has a greater emphasis on the medical recommendations and limitations for dietary intake. The customer requires a system that she can recommend to patients to get a better grasp of their dietary composition for diagnostic use, as well as a system easy enough for a patient to use to understand the limits of their diet. The app will help avoid over- or under-consumption for more sensitive patients.

Success will be evaluated in the form of surveys. The idea will be described to people walking through university and they will be asked if this application would help them in their lifestyle. We can send an email survey that will ask them to rate the system, so we can determine satisfaction of the app.

Big User Stories

1. Add Flexible Goals

“As a user, I want to be able to set flexible goals, for example over one week, biweekly, monthly”

Priority: Medium (changed from High in Iteration 2) Cost: 7 days Created Iteration: 0

Discussion: We have redesigned the structure behind this user story several times, clarifying with the customer what it was that they really want. At the moment, we have decided to save the information locally vs in the database to minimize the database complexity. They will be temporarily separated and range-search optimized. At the moment, we are having trouble with saving/loading the files so we are planning to complete it in Iteration 3.

2. Goal Setting

“As a user, I want to be able to set health-related goals”

Priority: High Cost: 7 days Created: Iteration 0

A. Detailed User Story: Setting Goals

Allow the user to input goals that they want to track and be notified on.

Priority: Medium (changed from high in It2) Cost: 3 days (changed from 6 days in Iteration 1)From Iteration: 0

See discussion on Add Flexible Goals. As we had to make a late design decision to change the saving-format of the Goals (vs implementing it into SQL) we were unable to finish this user story for this iteration and will focus on it in the next one. We have dropped its priority as a result.

3. Custom Foods

“As a user, I want to be able to enter custom foods and their nutritional composition for use in diet tracking.”

Priority: High Cost: 8 days Created: Iteration 0

Discussion: We have been having design-related disagreements with the format of the external database. At the moment, Robby has combined all the nutrients with the foods in one row and it has resulted in 600 column tables. We are unsure whether this will be too slow and if we should use primary-key ID lookups between tables to keep things organized, but that would definitely be a lot harder to implement. We will return to this topic in Iteration 3. For now, there is still a base-interface to enter foods

B. Detailed User Story: Enter Custom Foods

Allow the user to record information about foods/meals, including Calories, grams of proteins/carbohydrates/fats.

Priority: High Cost: 4 days From Iteration: 0

Discussion: See “Custom Foods” Big User Story discussion

C. Detailed User Story: Enter Meals

Allow the user to enter details of a meal eaten using foods from an pre-existing list, and entering serving size.

Priority: High Cost: 4 days (changed from 8 days in Iteration 1) From Iteration: 0

Discussion: See “Custom Foods” Big User Story discussion.

D. *Detailed User Story: Display Eaten Meals*

Display the records of all previously entered meals, separated temporally.

Priority: High Cost: 8 days From Iteration: 0

Discussion: See “Custom Foods” Big User Story discussion. We have an interface ready to pull from the database, once the format has been agreed on.

4. Pre-Loaded Foods

"As a user, I want to use foods that are already in the application for dietary tracking."

Priority: High Cost: 10 days (changed from 3-4 days in Iteration 2) Created: Iteration 0

Discussion: This has been by far the longest big-user-story. Using the Canadian Nutrient File has proven to be a bigger task than anticipated we will have it completed for Iteration 3. At the moment, there is a bug in its deleting of table entries to optimize the table output. There is already code set-way for generating the SQL database to be loaded into the existing application.

The way this works is CsvConverter.java will parse the provided extdb/*.csv files, extract foods and nutrients that are desired, and output an SQLite database with the java classes to hold it. This will NOT be run during application run-time but as a developer-side application, so it will not need testing so much so as the output SQL file will. That is, the output SQL database will be provided WITH the application. The processing/running of the Main method in DatabaseManager to output this file will be done before launch.

As the CsvConverter extends no activity context, a **new third-party library** was added to the project to export SQLite databases for import into the application.

5. Push Notifications

“As a user, I want push notifications to remind when I’ve made a goal or when I’ve forgotten to enter data.”

Priority: Low Cost: 3-4 days (changed from 8 days in Iteration 1) Created: Iteration 0

Discussion: We were able to successfully implement the base-structure for generating notifications. We need only to finish the GoalsAccess feature to tie this in with it.

E. Detailed User Story: Display Notifications

The application will notify the user (through the action bar) when a user has met or exceeded their daily/weekly goals.

Priority: High Cost: 4 days (changed from 10 days in Iteration 1) From Iteration: 0

Discussion: See “Push Notification” big-user-story discussion. We are currently awaiting the other user-stories to finish before tying this in with them.

6. Statistical Information

“As a user, I want to be able to see a breakdown of the foods I have eaten and recorded based on qualifications I desire to see like carbs, protein etc.”

Priority: High (changed from medium in It0) Cost: 3 days (from 10) Created: Iteration 0

Discussion: We were able to make a working interface for this. The charts are outputting successfully and the library is proving very versatile and user-friendly. We are waiting for the other user-stories to finish before connecting this feature with them.

F. Detailed User Story: User Information Submission Form

Allow the user to submit information about their weight, age, and height in a form.

Priority: Medium Cost: 3 days (changed from 8 days in Iteration 1) Created: Iteration 0

Discussion: We were able to make a decent user-submission form. We are having troubles with serializing data outside of the SQL database but will have that figured out for the next iteration

G. Detailed User Story: Determining Diet Statistics

Calculate the nutritional statistics for the diet, based on diet entries (by evaluating the daily/weekly nutritional amounts for Calories, proteins, carbohydrates, and fats).

Priority: High Cost: 6 days From Iteration: 0

Discussion: We were able to implement the classes to calculate these diet statistics (Calc*.java). We are waiting for the other user-stories to finish before connecting this feature to it.

H. Detailed User Story: Display Nutritional Information Graphically

Display the user’s statistics in graphical format (pie charts and bar graph).

Priority: High Cost: 3 days (changed from 15 days in Iteration in Iteration 1) From Iteration: 0

Discussion: We were able to make a working interface for this user story. The library proved very useful and saved us a lot of time, allowing us to implement the new features below.

7. (NEW) Ambient Music

“As a user, I would like some relaxing sounds to be played in the background as I use the app.”

Priority: High Cost: 3 days Created: Iteration 2

Discussion: We received this new Big-User-Story late into Iteration 2 from the customer but it was a small enough task to attempt implementing. Unfortunately, the sound proved very difficult to control and will need more work in the next Iteration.

8. (NEW) Button Sounds

“As a user, I would like to hear something, some sort of sound, when I click on the buttons of the app.”

Priority: High Cost: 2 days Created: Iteration 2

Discussion: We received this new Big-User-Story late into Iteration 2 from the customer. Unfortunately, the sound proved very difficult to control and will need more work in the next Iteration.

9. (NEW) Health-Related Variable Calculations

“As a user, I want the app to tell me what my body-mass-index is or how much my heart-rate should be, for example.”

Priority: High Cost: 1 days Created: Iteration 2

Discussion: We received this new Big-User-Story late into Iteration 2 from the customer but it was a small enough task to attempt implementing. We were able to successfully implement an interface for the calculations and are waiting for the other User Stories to finish before putting it to actual use.

10. (NEW) Splash Screen

“As a user, I would like to see a nice customizable pop-up screen as I run the app.”

Priority: low Cost: 1 days Created: Iteration 2

Discussion: We received this new Big-User-Story late into Iteration 2 from the customer. We did not feel it merited attention for now as the other user-stories were far more important.

Testing Decisions and Business Layer

The following business layer objects will not have corresponding tests for the provided reasons:

i) ApplicationConstants
- Being an Interface, this will not require a test as it contains only constants
- Holds all constants related to the project as well as the dependency injection for the Stub Database

ii) Calculator
- Is an interface for the other Calc*.java files; used as an abstract base class

iii) ClickSound
- Is broken
- We were having troubles disabling the sound between launches of the app. Extensive research revealed several deprecated functions that made our current technique not plausible. It will require more research and work in Iteration 3 to make work.

iv) Goals
- Intermediate class; storage for getters/setters

v) GoalsAccess
- Work in progress, not ready to test in this iteration
- Bugs with saving

vi) InitPieChart
- Intermediate class; storage for getters/setters
- Relies on Stats class, which is tested

vii) Journal
- Work in progress; not ready to test in this iteration
- Bugs with data retrieval/storage

viii) JournalEntry
- Intermediate class; storage for getters/setters

ix) NotificationDataAccess
- Work in progress; not ready to test in this iteration
- Waiting on external database to finish

x) Nutrient
- Intermediate class; storage for getters/setters

xi) Stats
- The testing is done on StatsDataAccess (which it relies on)

xii) UserDataAccess
- Not enough time to test??

xiii) External DB (Persistence Layer)
- Will not be tested until next iteration. Has bugs with deletion and may need an entire remodeling as it is currently an X foods/row by 600 columns table, which can’t be efficient but we’ll see.