Bamboo

Sprint 1 Planning Document

Team 2

Ruchira Goel, Ramya Mandyam Anandampullai, Linnea Lindstrom, Hajera Zemy

Overview

During this sprint, our main goals are to set up the different components of our project (the frontend, backend and database), allow users to set up an account, login and log out, customize certain settings, enter daily diet input, and process and store the nutritional information for these inputs.

We plan to first meet to set up and connect the frontend, backend, and database, in order for everyone to learn the basic structure of our app and become familiar with the system we will be working on.

Scrum Master

Ruchira Goel

Meeting Plan

- Tuesday at 11:30 am
- Thursday at 10:30 am with Project Coordinator
- Sunday at 3:30 pm

Risks & Challenges

Many of us are unfamiliar with most of the technologies that we are going to use, which will pose some challenge at the start. Some of our time for Sprint 1 will go into learning how to properly use and navigate through them before we can work on our assigned tasks.

We also found it very difficult to estimate the time that these different tasks would require. Because some of these tasks are very new to us we don’t have any previous experience to base our time estimations on. We tried to get as accurate a number as possible, but felt that it would need a lot of research to really understand how intensive coding each part would be.

Because it might end up taking much longer than we estimated for certain tasks, or maybe less time than we thought for some, we may not have actually assigned ourselves an appropriate amount of work for the time that we have in this sprint. While we’ve made the most educated decisions we can at this time, this may still cause problems with scheduling remaining tasks during the later sprints.

It was also difficult to predict how much some of the tasks depend on another task, due to which we might have to work on such tasks together, which might lead to us spending some additional time on those tasks.

Current Sprint Detail

Set Up - Setting up various technologies and components of the project

#	Task Description	Estimated Time	Owner
1	Set up React Native frontend	1 hr	Ruchira, Ramya, Linnea, Hajera
2	Set up MongoDB database	1 hr	Ruchira, Ramya, Linnea, Hajera
3	Set up Spring Boot backend	1 hr	Ruchira, Ramya, Linnea, Hajera
4	Link React Native project to Spring Boot, and Spring Boot to MongoDB and verify that these components can communicate with each other	2 hrs	Ruchira, Ramya, Linnea, Hajera
5	Define User objects and related methods in Spring Boot	5 hrs	Ruchira, Ramya, Linnea, Hajera

Acceptance Criteria:

1. Given that the frontend is set up correctly, if any member of the team wishes to access and make changes to the front end, they should be able to do so.
2. Given that the database is set up correctly, if any data stored in it is accessed by a member of the team, then the data should be correctly fetched.
3. Given that the components of the project are set up and linked correctly, if a member of the team tries to connect some aspect of the backend with the frontend, they should be able to do so without trouble.

User Story 1 - Sign up for a Bamboo account

#	Task Description	Estimated Time	Owner
1	Create UI to display signup page	4 hrs	Ramya

2	Provide inputs for email, password, and confirm password	2 hrs	Ramya
3	Enforce length for password	2 hrs	Hajera
4	Create a user object with unique ID and store the email and password (hashed) in the database	4 hrs	Ramya
5	Debug and test account creation with unit tests	5 hrs	Ramya

Acceptance Criteria:

1. Given that the email validation functions correctly, if a user tries to sign up with an email address that is already linked to a user object, there should be an error message.
2. Given that this error message is implemented correctly, it should include a link to the login page.
3. Given that the password validation functions correctly, if a user enters a password of less than 8 characters an error message should be displayed.
4. Given that the validation checks are implemented efficiently, a new user with no errors in email and password should be able to enter this signup information (email, password, and confirm password) and be directed to the next page within 1 minute.

User Story 2 - Set up my account profile with health characteristics specific to me

#	Task Description	Estimated Time	Owner
1	Create the UI for the user to enter health characteristics	4 hrs	Ruchira
2	Validate user entries and store the information in the database	1.5 hrs	Ruchira
3	Debug and test account profile setup with unit tests	2 hrs	Ruchira

Acceptance Criteria:

1. Given that the input validation is implemented correctly, if the user enters an invalid entry (e.g. a string for age), then an error message is displayed prompting the user to enter valid information
2. Given that the information entered by the user is valid, the user will be directed to the appropriate page (tentatively home screen).
3. Given that the UI works correctly, when the user has entered personal account information and taps the next button, then the health characteristics input page should show up.

User Story 3 - Be able to choose my system of measurement (metric or imperial)

#	Task Description	Estimated Time	Owner
1	Add option to change system of measurement during initial sign up	2 hrs	Ruchira
2	Add an option to the settings UI to select either metric or imperial system of measurement	2 hrs	Ruchira
3	Softcode the units of measure to display on the UI	0.5 hrs	Linnea
4	Test to make sure the system of measure can be changed and the units will display differently	2 hrs	Linnea, Ruchira

Acceptance Criteria:

1. Given that the functionality to change units is implemented correctly, when the user changes the system of measurement, it should reflect on all data sets that are displayed in the app.
2. Given that the option to change the system of measurement is added, when the user goes to settings they can see and change their option.
3. Given that the functionality to change units is implemented correctly, when the user logs in for the first time they will be asked which system they want to use.

User Story 4 - Be able to view my personal information

#	Task Description	Estimated Time	Owner
1	Create UI Page that displays user profile	4 hrs	Linnea
2	Create a navigation menu in the home screen and display the Account Profile option	4 hrs	Linnea
3	Debug and test profile viewing with unit tests	4 hrs	Linnea

Acceptance Criteria:

1. Given that the menu for the homepage is implemented correctly, when a user clicks on the menu button, a menu displaying the Account Profile option will show up.
2. Given that the account profile navigation is implemented correctly, when the user taps on the Account Profile option, account profile page shows up, containing profile details.
3. Given that the user profile UI is implemented well, when the user has a profile that does not fit in a single page, scrolling should be allowed.

User Story 5 - Be able to login and manage account information

#	Task Description	Estimated Time	Owner
1	Create UI to display login page	4 hrs	Hajera
2	Add UI to accommodate for changing password and name under settings	4 hrs	Ruchira
3	Validate changed user entries and store the information in the database	2 hrs	Hajera
4	Debug and test changed account profile with unit tests	2 hrs	Hajera

Acceptance Criteria:

1. Given the login page is properly implemented, when I input my email and password correctly, then I will be logged in and shown the home page.
2. Given the settings section is correctly implemented, I can successfully change my password and/or name.
3. Given the login page is properly implemented, when I input my email or password incorrectly, an error message will be displayed and I will remain on the login page.

User Story 6 - Be able to access my account on various mobile devices

#	Task Description	Estimated Time	Owner
1	Test all implemented user stories on Android devices	0.5 hr	Ruchira, Ramya, Linnea, Hajera
2	Test all implemented user stories on iOS devices	0.5 hr	Ruchira, Ramya, Linnea, Hajera
3	Access an account on one device, then log in to the account on another device	0.5 hr	Ruchira, Ramya, Linnea, Hajera

Acceptance Criteria:

1. Given the app functionality and other user stories are implemented properly, all aspects of the app implemented so far will work on an Android device.

2. Given the app functionality and other user stories are implemented properly, all aspects of the app implemented so far will work on an iOS device.
3. Given the app is cross-platform, I can login to my account and access my data on different devices.

User Story 7 - Be able to log out of my account

#	Task Description	Estimated Time	Owner
1	Create a menu option to log the user out	1 hr	Hajera
2	Create a dialog box prompting confirmation of logout and navigate them to the login page	1 hr	Hajera
3	Debug and test logout with unit tests	2 hrs	Hajera

Acceptance Criteria:

1. Given that the menu for the homepage is implemented correctly, when a user clicks on the menu button, a menu displaying the Log Out option will show up.
2. Given that the user chooses to log out, the user will be redirected to the login page
3. Given that the log out functions correctly, when the user clicks the logout button, the user should not be able to access the account data without logging back in.

Note: While completing the tasks required for the above user story, we would also be completing one of our non-functional user stories, which is listed below.

- To ensure a secure platform, a user will not be able to access the app without being logged in.

User Story 8 - Be able to enter daily diet and meal information

#	Task Description	Estimated Time	Owner
1	Create UI for entering daily meal & nutrition information manually	4 hrs	Hajera
2	Create UI for entering daily meal information by a link to a recipe online	6 hrs	Ramya
3	Define Meal and Nutrition classes and their related methods	5 hrs	Hajera
4	Validate user entries	1.5 hrs	Hajera
5	Test and debug to make sure daily input works correctly	3 hrs	Hajera, Ramya

Acceptance Criteria:

1. Given that the input is correctly implemented, when the user tries to add meal inputs, then they should be able to enter a set of ingredients with quantities as the input.
2. Given that the user entry validation is correctly implemented, if a user enters an invalid input (for eg. link to a page that does not contain a recipe, text that does not include food items or quantities) an error message should be displayed.
3. Given that the creation and storage of Meal and Nutrition objects is correctly implemented, if there is an error in user input, no meal information should be stored.

User Story 9 - Use daily diet inputs to get nutritional information

#	Task Description	Estimated Time	Owner
1	Set up and link Spoonacular API to backend	4 hrs	Ruchira, Ramya, Linnea, Hajera
2	Make API calls to calculate Nutritional information from user inputs	5 hrs	Ramya
3	Create the appropriate Meal and Nutrition object and store them in the database	1 hr	Hajera
4	Debug and test parsing nutrition information from inputs with unit tests	4 hrs each	Ramya, Hajera

Acceptance Criteria:

1. Given that the input and output for the API calls are correct and properly formatted, the nutritional information for the entered meals should be accurate.
2. Given that the meal information is stored correctly, it will show up correctly in the database.
3. Given that the API is set up and used correctly, when the user enters meal information in the form of a link to a recipe, then the nutrition information for that must be fetched and stored correctly.
4. Given that the API is set up and used correctly, when the user enters meal information in the form of a list of ingredients with quantities, then the nutritional information must be fetched and stored accordingly.

User Story 10 - Enter daily exercise information such as number of hours for each activity

#	Task Description	Estimated Time	Owner

1	Create UI for entering daily exercise information	4 hrs	Linnea
2	Validate user entries and store the information in the database	1.5 hrs	Linnea
3	Define Exercise Routine class and related methods	1.5 hrs	Linnea, Ruchira
4	Define Activity class and related methods	1.5 hrs	Linnea, Ruchira
5	Create the appropriate Activity object and store them in the database	1 hrs	Linnea
6	Test entering exercise related information with unit tests	2 hrs	Linnea

Acceptance Criteria:

1. Given that the daily input is implemented correctly, if the user enters information such as calories burned or type of activity, it must be stored to the database correctly.
2. Given that the input validation is implemented correctly, if the user enters an invalid entry, then an error message is displayed prompting the user to enter valid information.
3. Given that the creation and storage of the Activity object is correctly implemented, if there is an error in user input, no activity information should be stored.

User Story 11 - Be able to complete daily input in a short period of time

#	Task Description	Estimated Time	Owner
1	Develop test cases for typical daily input for diet and exercise	4 hrs	Ruchira
2	Test inputs and record time	2 hrs	Ruchira
3	Optimizing so that each input takes less than 1.5 minutes	2 hrs	Ruchira

Acceptance Criteria:

1. Given that this functionality works, when the user gives daily input, the input time for each input (like a single meal or exercise routine) will take a maximum of only 1.5 minutes.
2. Given that the user gives daily input, the input entries should fit in a single page without scrolling, which would require for there to be fewer number of entries
3. Given that our test cases are designed effectively, they should include different types of meal input (such as from an online recipe, from a user's own recipe, etc.), and exercise input.

User Story 12 - Ensure the security of the users' data and passwords

#	Task Description	Estimated Time	Owner
1	Enforce password length of 8 characters minimum	1 hr	Hajera
2	Hash the password, likely using bcrypt	4 hrs	Hajera
3	Test and debug to ensure password security	3 hrs	Hajera

Acceptance Criteria:

1. Given the password length is enforced, I will be shown an error message if I attempt to set a password of less than 8 characters upon account creation.
2. Given the password length is enforced, I will be shown an error message if I attempt to change my password to less than 8 characters in settings.
3. Given that the password hashing is implemented correctly, the user's actual password should never be visible or directly accessible to the developers.
4. Given that the user objects and their related data (daily inputs, goals, etc) are stored in the database correctly, users should only be able to access and modify data relevant to them.
5. Given that the password is hashed, the passwords will be protected against multiple types of attacks.

Note: While completing the tasks required for the above user story, we would also be completing one of our non-functional user stories, which is listed below.

- Passwords will be encrypted while handling requests and in the database so that they are not visible to developers and other users

User story 13 - Be able to enter information for previous days if I forgot to

#	Task Description	Estimated Time	Owner
1	Add UI to select a date for the daily input for exercise and meals	3 hrs	Ruchira
2	Get the current date and have it autofilled as the date	1 hr	Ruchira
3	Debug and test to make sure data entries are saved under the right dates	3 hrs	Ruchira

Acceptance Criteria:

1. Given that the UI for the date is implemented correctly, when the user loads the daily input page the current date should be autofilled.

2. Given that the UI for date selection is implemented correctly, the user should be able to navigate a calendar view to select the appropriate date.
3. Given that the storing of daily input is implemented correctly, the user should be able to enter information for an earlier date without overwriting any data already stored for that date.
4. Given that the storing of daily input is implemented correctly, the user should be able to enter information for an earlier date for which there is no data already stored.

User Story 14 - Be able to update my personal information as needed

#	Task Description	Estimated Time	Owner
1	Add the option to edit profile on the Account Profile menu UI	2 hrs	Linnea
2	Add validated input for the user to change their profile information	1.5 hrs	Linnea
3	Update database data upon changing personal information and saving	2 hrs	Linnea
4	Debug and test profile viewing with unit tests	1.5 hrs	Linnea

Acceptance Criteria:

1. Given the account profile menu had a properly implemented edit option, the user can input and update their personal information.
2. Given the save button on the edit view is correctly implemented, the changed information will be saved for the user in the database.
3. Given the user's personal information is successfully changed, the updated information will be correctly displayed on their profile and used in calculations.
4. Given the user's personal information is successfully changed, the updated information will be correctly used in calculations.

Remaining Backlog

Legend

🟩 current sprint

Functional Requirements

Account & Personal info

As a user, I would like to...

1. Sign up for a Bamboo account
2. Set up my account profile with health characteristics specific to me
3. Be able to log in and manage account information
4. Be able to recover my account if I forget the password
5. Be able to access my account on various mobile devices
6. Be able to log out of my account
7. Be able to delete my account and all personal data from the application if needed
8. Be able to view my personal information
9. Be able to update my personal information as needed

Daily Inputs

As a user, I would like to...

1. Enter daily exercise information such as number of hours for each activity
2. Be able to enter daily diet and meal information
3. Enter daily sleep information (if time allows)
4. Be able to enter information for previous days if I forgot to
5. Scan product barcodes for easy input of meal information (if time allows)
6. Choose menu items from nearby restaurants for easy input of meal information (if time allows)
7. Add regular workout routines or common meals to a saved list, to save time inputting the same information again on a later date
8. Be able to complete daily input in a short period of time

Goals

As a user, I would like to...

1. Set goals for minutes/hours of daily or weekly exercise
2. Set goals for daily calorie intake
3. Set dietary goals regarding certain nutrients, such as the amount of sodium, or fat in a daily diet

4. Set goals for going to sleep at a specific time (if time allows)
5. Set goals for waking up at a specific time (if time allows)
6. Set goals for getting a certain amount of sleep per night (if time allows)
7. View my current goals
8. Edit and update my current goals
9. Track my progress with my goals

Analytics & Recommendations

As a user, I would like to...

1. Check analytics over time
2. View a graphical representation of exercise over time
3. View a graphical representation of calories consumed over days
4. View weekly/monthly reports containing diet and exercise data and goal progress
5. Find recommended meals that are in line with my set goals
6. Find recommended meals with specific tags, such as high protein or low sodium
7. Set dietary restrictions and allergy information for meal recommendations
8. Find recommended exercise routines to speed up progress towards set goals

Notifications & Settings

As a user, I would like to...

1. Be able to choose my system of measurement (metric or imperial)
2. Receive notifications as reminders to input my exercise or dietary information
3. Receive reminders to exercise on days I consistently forget
4. Receive reminders of my goals based on other behaviour patterns (if time allows)
5. Receive notifications with positive encouragement, such as how long of a streak I'm on of achieving goals
6. Edit notifications settings as needed

Development

As a developer, I would like to...

1. Improve exception handling and produce proper messages for errors
2. Have thorough test cases that test functionality and possible edge cases
3. Ensure the security of the users' data and passwords
4. Use daily diet inputs to get nutritional information

Non-functional Requirements

Usability

1. Minimize manual input from the user
2. Simple and intuitive user interface so that all users find it easy to navigate through it
3. Cross-platform application to ensure that people can access it from all mobile devices
4. Track different types of data like exercise and diet, instead of only one aspect of lifestyle

Security

1. To ensure a secure platform, a user will not be able to access the app without being logged in
2. Passwords will be encrypted while handling requests and in the database so that they are not visible to developers and other users
3. Users will be provided with a choice on whether they want to allow location tracking

Response Time & Scalability

1. Application launch time, i.e. time that is required for the UI of the application to load, should be less than 10 seconds
2. Any response to a user action should be completed in less than 1 second
3. For any responses that take longer than 10 seconds, the application will show a loading screen or a progress bar that shows how long the action will take so that users will know what to expect
4. Aim to have Bamboo be able to handle at least 100 simultaneous requests