Contributions:

Nicole (25%):

- I set up Firebase for meals collection, and then created the meal logging feature that is currently integrated with DeepSeek and Firebase. I set up a navigation interface with buttons for each meal category (Breakfast, Lunch, Dinner, Snack) that navigates the user to the meal log screen. In the meal log screen, there is an initial prompt to add a meal if no entries exist yet for the day. However, if there are existing meals logged under this Firebase user for today, then I fetch and display the food items for that category in a table view, sorted by entry time. The meal entries are generated programmatically using UITableViews, with each cell displaying the food name, quantity, and calories. When the user enters a food item, I format the input into a strict JSON prompt and call the DeepSeek API to get nutritional information. After processing the API response (around 10 seconds) and parsing the nutritional data, I upload the complete meal log to Firebase Firestore and refresh the UI. All user meal logs are stored in Firebase Firestore associated with each user.
- I set up a UICalendarView with custom styling and Auto Layout constraints in MealMentor, and implemented UICalendarViewDelegate for future customization.
    - LogDisplayViewController.swift
    - LogEntryViewController.swift
    - LogEntry.storyboard
    - CalendarPageViewController.swift
    - Meal.swift
    - Food.swift

Bowen (25%):

- I set up the UI and functionality for the welcome, login, registration, onboarding, profile, and settings pages. I also set up the firebase authentication functionality for login/registration. For the login/registration, specific functionality includes creating an account and logging in. If the user doesn't log out, the application will automatically log in to their account and bring them to the home page. If the user has just created a new account, the registration page will segue directly to the onboarding pages(NOT to the login screen) where MealMentor will ask for specific information such as weight, gender, and height. The onboarding pages will also ask for user specific goals, allergies, etc.. The user can choose whether to answer any of the prompts in the onboarding pages. All this information will be stored in Firebase under the specific userid of the user's new account. The settings page is a page under the UITabController of the HomePage that will have functionality for changing the password, verbose chat responses in the chat page, and dark mode. The profile page is also a page under the UITabController of the HomePage that displays the user's personal information. The profile page also allows the user to log out and change their dietary restrictions, nutrition focuses, goals, and allergies.
    - Main.storyboard(login/registration)
    - LoginViewController.swift
    - OnboardingPage1ViewController.swift

    - OnboardingPage2ViewController.swift
    - ProfilePage.storyboard
    - ProfilePageViewController.swift
    - RegisterViewController.swift
    - SettingsPage.storyboard
    - SettingsViewController.swift
        - Empty Cocotouch swift file for now, will be useful in later phases
    - UserAccountInterface.swift
    - ChangePasswordViewController.swift
    - DietaryRestrictionsViewController.swift
    - WelcomePageViewController.swift

Gina (25%):

- I created the home page view controller and UI. This consists of a scrollable view, a weekly calendar collection view for displaying tracked days, a table view for displaying today's meals, and bar graph visualizations for nutrition statistics. In making the weekly calendar, I learned how to utilize Swift's Calendar and Date classes and also UICollectionView. In making the visualizations, I learned how to use the DGCharts and TinyConstraints frameworks.
- I created the visualizations page view controller and UI, which just consists of a segmented control (to navigate between nutrition categories) and a scrollable view for now.
- I created the tab bar controller and set up most of the navigation between our storyboards via segues and storyboard references (to log entry page, calendar page, settings page, profile page).
    - HomePage.storyboard
    - HomePageViewController.swift
    - VisualizationsPageViewController.swift
    - ThisWeekDayCell.swift
    - BarChartXAxisWeekdayValueFormatter.swift
    - BarChartYValueUnitValueFormatter.swift

Huyen (25%):

- I set up Firebase and Deepseek, and then created the AI chat feature that is currently integrated with Deepseek and Firebase. I set up the navigation bar to include a 'Chat' button that navigates the user to the chat page. In the chat page, there is an initial screen with sample prompts and an input bar. However, if there exists a chat history under this firebase user, then I don't show this screen and instead populate the chat with the chat messages in the last 24 hours in a scroll area, sorted by the time. The chat messages are created programmatically using a UIScrollView and a UIStackView. When the keyboard opens and closes, the chat bar will move up and down to fit the keyboard. The user can click the submit button or click return on the keyboard to chat to the AI. For every submission, I call the Deepseek API to produce a response. I also had to figure out how to store and hide the API key. I had to handle the response and upload this to firebase and populate it in the UI. I save all user chat history in Firebase firestore.
    - ChatPage.storyboard

    - ChatPageViewController.swift
    - ChatBubble.swift
    - ChatMessage.swift
    - AppDelegate.swift (for firebase setup)
    - HomePage.storyboard (navigation to my storyboard)

Differences:

- Planned to start the AI Chat feature in beta however we decided that the AI and the logging would both be utilizing Deepseek API. Therefore, since Huyen had more free time to start the project earlier, she decided to work on the Chat AI to call the Deepseek API so that Nicole could refer to this code for the logging.
- We also didn't include the nutrition visualizations in our initial proposal, but in developing our design and following instructor feedback, we decided to make this more of a priority in our app.
- Some very minor UI changes such as adding logos to the welcome page. This was because as we went along through the project we came up with new UI ideas to make the app look nicer. Additionally, added user properties in the onboarding page to allow the user to input their first and last name. We also got rid of the username property as we are using email to login, so in the settings page we didn't need a "change username" option. Also got rid of the first and last name in the properties of the profile page since we already display the first and last name at the top. Did not implement the category selectable functionality for the nutrition category on the onboarding page, and the dark mode and verbose chat responses toggles on the settings page since I decided to focus more on firebase auth and storing user properties in firestore. This was more important to integrate the other parts of the app.