

Project Name: FitnessTrack with Meal Planner Application

Authors: Leosbell Galvez, Wesley Del Orbe, Sabrina Valdivia

Course: CIS 4951, Capstone II, Spring 2026

Instructor: Masoud Sadjadi

Date: 01/12/2026 - 04/17/2026

Version: 5.0

Contents

1. Project Summary.....	3
2. Problem Definition	4
3. System Overview.....	5
4. Architecture & Designing.....	6
5. Implementation Details	8
6. Application Screenshots	11
7. Testing & Evaluation.....	12
8. Results.....	13
9. Challenges & Lessons Learned	13
10. Future Work.....	14
11. Conclusion	14
12. Important Documentation and References.....	15

1. Project Summary

This project introduces development of a Fitness Tracking Application with Meal Planning features designed to allow users to keep track of exercises for their convenience and generate their own personalized meal plans based on their input and dietary preferences. This project integrates the Spoonacular API to retrieve meals and nutrition information for users to see. Users are allowed to input their weight, height, and calorie goal allowing the system to return meal plans for daily or weekly use with attached relevant information. In addition, exercise logs are stored in a cloud database that is hosted by back4app, ensuring safe data handling for every user to track across sessions and devices.

This application showcases principles of API integration, parsing data, and designing user interfaces. The final system will successfully generate meal plans under various conditions, though the API itself came up with some constraints and limitations when in use. This documentation will completely outline the full design, implementation, and monitoring of the system.

2. Problem Definition

Meal Planning is an extremely tedious task for many, whether it is time-consuming or inefficient at times. It becomes even more difficult for individuals who have restrictions on their diet, such as vegetarian, vegan, ketogenic diets, or other restrictions that may impose meal-planning. A lot of options exist but are generally isolated from other useful features such as exercise logging, which is a crucial aspect to many individuals' fitness journey.

This project aims to:

- Provide secure user authentication
- Automate the meal-planning process
- Allow for useful storing of exercise logs of a user
- Reduce the effort of users in the decision-making process
- Combining nutrition and fitness into a single system
- Improve the users awareness of their general health habits

Target Users:

- Students
- Gym-goers
- Busy individuals
- Individuals with dietary restrictions

3. System Overview

This system is a client-based application communicating to an external API for meal plan generation and utilizing a backend database for exercise logging.

This is a hybrid application of three core components:

- Authentication System (user accounts)
- Meal Planning generation Module which is API-based
- Exercise Tracking/Logging Module which utilizes cloud database per user

High-Level Flow:

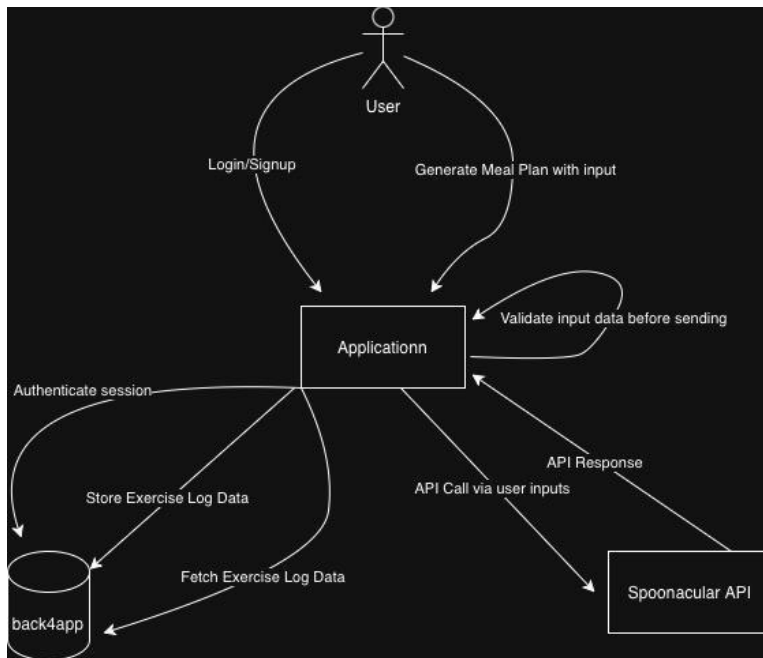
- User signs up or logs in
- Authentication is validated for the user session
- Exercise logs are stored in back4app under the user's unique ID
- Exercise History View has full exercise logging history (data fetched from database)
- Meal Plan Data is fetched from the Spoonacular API based on user preferences

4. Architecture & Designing

This system follows a hybrid architecture utilizing an API and a backend database:

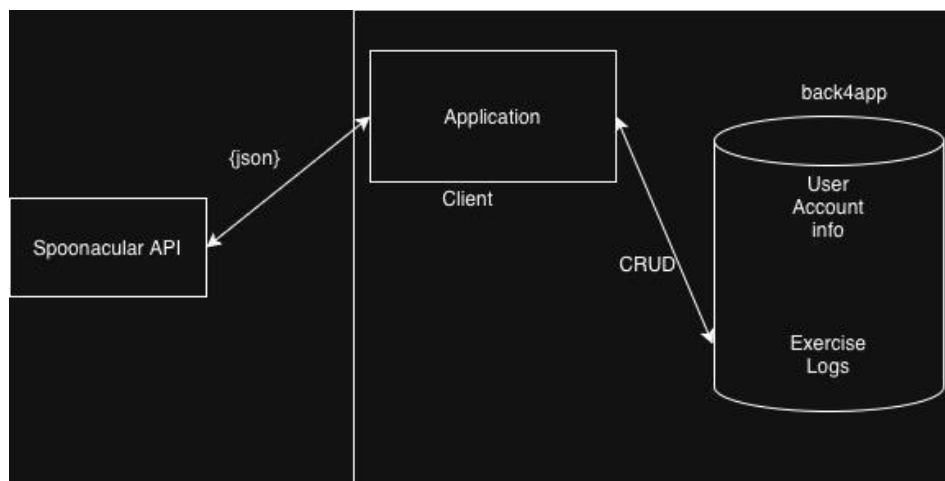
- Authentication Layer: Responsible for login, signup, and current session state
- Meal Module: Calls Spoonacular API, no meal-plan data is stored.
- Exercise Module: Responsible for storing user-specific logs in back4app.
- Application: Responsible for the combination of meal results with exercise data

4.1 System Architecture



4.2 Data Flow

- User will login and establish a session
- Meal request will trigger an actual real-time API call to Spoonacular API for meal planning.
- API will return a fresh meal plan.
- Exercise logs ARE saved to back4app database under specific user ID.
- The application essentially merges meal plan data with stored exercise history.



4.3 Design Decisions

- Stateless meal generation chosen to ensure that nutrition data stays up to date in case of a Spoonacular API change.
- The decision to avoid storing meal plans also allows to reduce potential redundancy and storage overhead.
- Used back4app database for persistent exercise tracking
- Implemented user authentication to ensure unique and secure data separation

5. Implementation Details

5.1 Technologies Chosen and Used

- Language: Swift
- Framework: SwiftUI
- Authentication: User login system
- API: Spoonacular
- Database: Back4App (exercise log handling)

5.2 Key Features

User Authentication System

- Provide secure login, signup, and logout functionality.
- Each session is uniquely associated with a user account
- Ensures personalized access to app data.

Meal Plan Generation (Stateless Aspect)

- Generates dynamic meal plans upon each request
- Utilizes Spoonacular API
- No meal data is stored to avoid redundancy, system overhead, and to ensure nutritional data stays up to date.

Exercise Logging

- Users log workouts (name, sets, reps, optional weight)
- This information is stored in Back4App

- Each entry is linked to the authenticated user

Personalized Application

- Displays the real-time meal plans generated
- Displays the exercise log history with edit/delete functionality
- Combines both into the application

5.3 Data Model Examples

```

1 import Foundation
2 import SwiftUI
3 import ParseSwift
4
5 struct User: ParseUser {
6     var objectId: String?
7     var createdAt: Date?
8     var updatedAt: Date?
9     var ACL: ParseACL?
10    var originalData: Data?
11
12    var username: String?
13    var email: String?
14    var emailVerified: Bool?
15    var password: String?
16    var authData: [String : [String : String]?]?
17 }

```

```

1 import Foundation
2 import SwiftUI
3 import ParseSwift
4
5
6 struct ExerciseLog: ParseObject {
7     var objectId: String?
8     var createdAt: Date?
9     var updatedAt: Date?
10    var ACL: ParseACL?
11    var originalData: Data?
12
13
14    var workoutName: String?
15    var sets: Int?
16    var reps: Int?
17    var weight: Double?
18    var dayOfWeek: String?
19    var date: Date?
20    var userId: String?
21 }

```

```

1 import Foundation
2 import ParseSwift
3 import SwiftUI
4
5
6 struct MealPlanResponse: Codable {
7     let meals: [Meal]?
8     let nutrients: Nutrients?
9     let week: [String: DayMeals]?
10
11     struct Meal: Codable, Identifiable {
12         let id: Int
13         let title: String
14         let readyInMinutes: Int?
15         let servings: Int?
16         let sourceUrl: String?
17     }
18
19     struct Nutrients: Codable {
20         let calories: Double
21         let protein: Double
22         let fat: Double
23         let carbohydrates: Double
24     }
25
26     struct DayMeals: Codable {
27         let meals: [Meal]
28         let nutrients: Nutrients
29     }
30 }

```

5.4 File Structure

/project

 /Authentication

 LoginView.swift

 /ExerciseViews

 EditWorkoutView.swift

 WorkoutHistoryView.swift

 WorkoutLogView.swift

 WorkoutRowView.swift

 /GeneralViews

 IndividualTabsView.swift

 HomeView.swift

 UpdateLogView.swift

 /MealViews

 MealPlanGeneratorView.swift

 MealPlanResultView.swift

 MealView.swift

 NutrientSub.swift

 NutrientSummaryView.swift

 /Models

 ExerciseLog.swift

 FallingDumbbell.swift

 MealPlanResponse.swift

 ParseUser.swift

 ContentView.swift

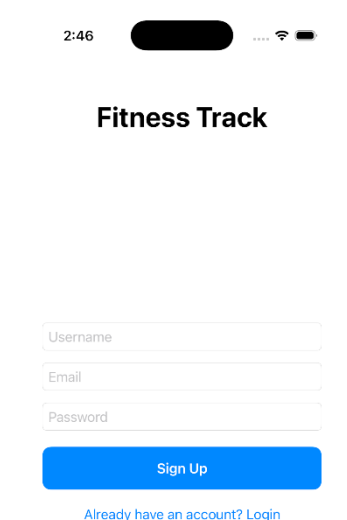
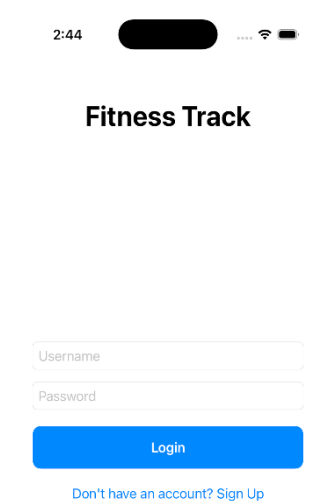
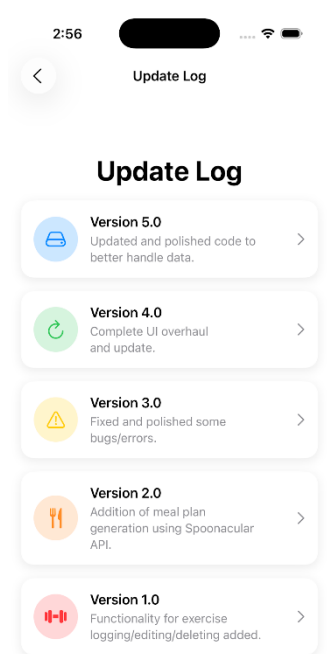
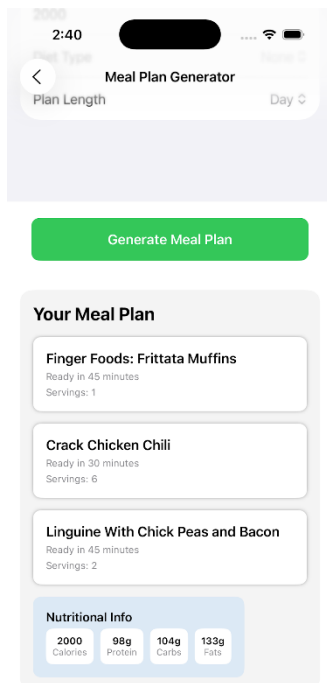
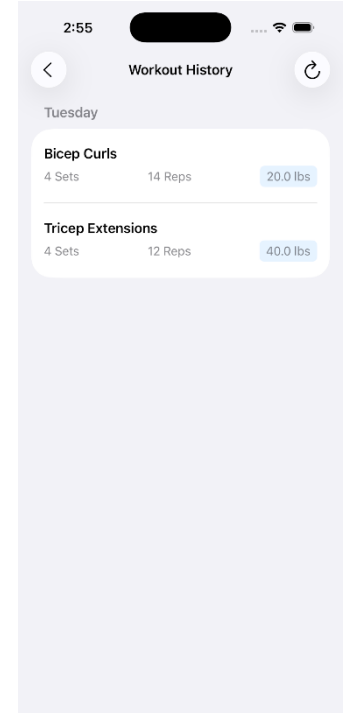
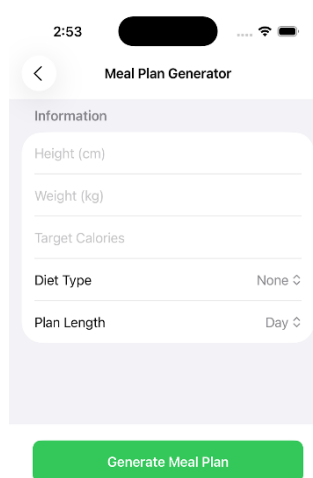
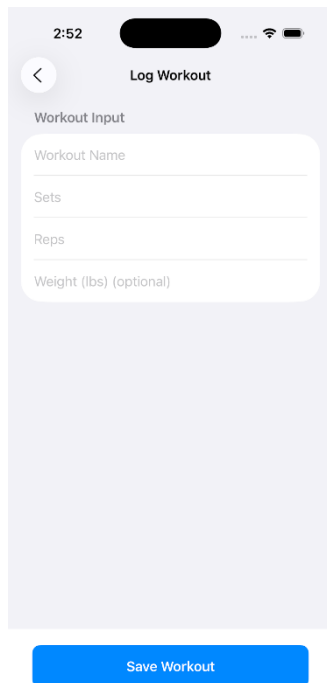
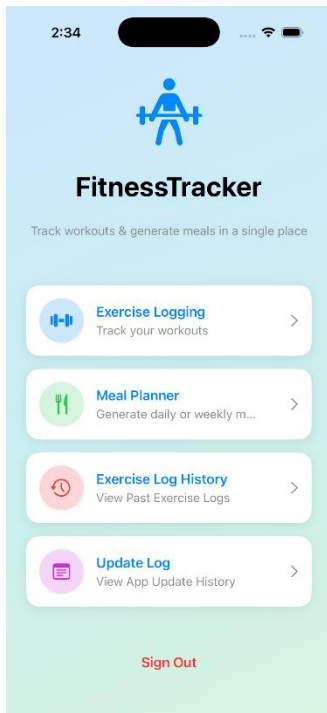
 FitnessTrackApp.swift

 Assets

Package Dependencies

 ParseSwift 4.14.2

6. Application Screenshots



These are actual UI elements in the application showcase here.

7. Testing & Evaluation

Testing Methods Deployed:

- Verified the authentication flow for login/logout/signup.
- Tested the real-time generation of meals via Spoonacular API.
- Verified that meal plans were not stored.
- Tested the persistence and accuracy of exercise logs stored in back4app database.
- Ensured isolated and unique data across accounts.

Example Test Cases

Test Case	Input	Expected Result	Actual Result
User Login	Valid Credentials/login information	Session is uniquely created	Success
Generate Meal	Gluten-free diet restriction, user input	Fresh API response with a meal plan generated	Success
Regenerate Meal after Refresh	Same exact request	New unique meal plan generated	Success
Log exercises	4 sets of 12 reps of 60lb bicep curls	Stored in back4app correctly	Success
Switch users	Login to a brand-new account.	Different unique data shown	Success
Display exercise logs	Open log history tab.	See accurate logs that were input by the user categorized by day/date.	Success
Delete exercise log	Delete an exercise log by sliding left on history tab	Should completely delete all associated data from the back4app database	Success
Edit exercise log	Edit an exercise log by sliding left on history tab	Should edit all data that was re-inputted and correctly store it in the back4app database	Success

8. Results

This system provides a multi-user health application that incorporates real-time meal generation and accurate exercise tracking. These meal plans are created dynamically requesting from the Spoonacular API and are not stored, allowing users to always receive fresh and new meal plans upon request.

Exercise logs are reliably and safely stored and retrieved from Back4App, allowing for long-term fitness tracking on multiple sessions and devices and aiding users on their fitness journey. User authentication allows for all data to remain private and properly isolated and unique.

9. Challenges & Lessons Learned

Challenges:

- Designing a stateless meal generation system in Swift.
- Ensuring secure user authentication accurately, safely, and efficiently.
- Handling API + database operations.
- Managing real-time API calls in an efficient manner.
- Dealing with navigational views incorporated into the functionality.

Lessons Learned:

- Stateless design allows for freshness and simplicity for stuff not necessary to be saved.
- Cloud databases are preferable for accurate and persistent data specific to a user.
- Authentication is extremely important for multi-user systems.

10. Future Work

- Incorporate biometric authentication like Face ID and Touch ID.
- Add more nutritional sources beyond just Spoonacular API.
- Add an optional feature for meal history saving. (If requested by user in settings)
- Add customization to the UI, including dark-mode, images, etc.
- Allow for more in detail exercise logging, including treadmill/run/walk tracking.

11. Conclusion

This project has demonstrated fully functional multi-user fitness/health managing systems which combines meal-plan generation and suggestions with accurate exercise tracking. It leverages Back4App for a cloud-based exercise logging storage, and Spoonacular API for a stateless meal planning generation feature that provides new and fresh entries.

12. Important Documentation and References

- [Spoonacular API Documentation](#) (providing all meal plan API responses)
- [Back4App Documentation](#)
- [Apple/SwiftUI development resources](#).