**COEN 6311 SOFTWARE ENGINEERING-WINTER 2025**

**Sprint Goal**

**Team Goal Statement:**

The team aims to design and develop CUFitness, a personalized fitness and nutrition platform tailored for the Concordia University community, empowering users to achieve their health and fitness goals through intuitive tools, community engagement, and seamless integration with SoftGrid.

Definition of Done (DoD):

The CUFitness platform by SoftGrid is considered "done" when:

1. **Functionality**: All key features (user onboarding, personalized workout plans, meal tracking, progress tracking, community engagement, and integration with SoftGrid) are implemented and fully functional.
2. **Usability**: The app provides a seamless, intuitive user experience across all devices, with no major usability issues identified through testing.
3. **Security**: User data is securely handled, encrypted, and protected according to industry best practices.
4. **Performance**: The app meets defined performance benchmarks, with fast load times and smooth interaction, even under high traffic.
5. **Quality Assurance**: All features have passed manual and automated tests, ensuring minimal bugs and errors.
6. **User Documentation**: Help guides and FAQs are available within the app for user support.
7. **Deployment**: The app is fully deployed and accessible for the target users at Concordia University.
8. **Feedback Integration**: Initial user feedback has been collected and addressed, with key improvements made based on real-world usage.

**User Story Estimation:**

**1. Discuss Each Story in Detail**

We'll review and clarify the stories, identifying any uncertainties. After that, we'll estimate the effort required for each story.

**Epic: User Onboarding and Account Setup**

User Story 1.1: Create an account and set up my profile.

* Details: Users should be able to create a new account, input basic information (name, email, password), and choose their fitness preferences (goal type, fitness level, etc.).
* Uncertainties: None, standard account creation process.

User Story 1.2: Input fitness level, goals, and preferences for personalized recommendations.

* Details: After account creation, users will input their fitness goals (e.g., weight loss, muscle gain), fitness level (e.g., beginner, intermediate), and preferences (e.g., workout time, equipment available).
* Uncertainties: Need to define if fitness level and goals will be preset options or user-defined.

**Epic: Personalized Workout Plans**

User Story 2.1: Receive a customized workout plan based on my fitness level and goals.

* Details: The system will generate a personalized workout plan tailored to the user's inputted fitness level and goals.
* Uncertainties: Which algorithm will be used to generate plans (machine learning or simple rule-based recommendations)?

User Story 2.2: Adjust my workout plan if I miss a session or hit a milestone.

* Details: The app will allow users to adjust their workout plan if they miss a session or reach a goal (e.g., increase intensity after hitting a milestone).
* Uncertainties: How will milestones be tracked, and how will the system offer adjustments?

**Epic: Nutrition and Meal Tracking**

User Story 3.1: Log meals and track calories.

* Details: Users will be able to input meals and track their daily caloric intake, using either a search tool or barcode scanner.
* Uncertainties: Will we integrate third-party databases for nutritional info, or will users manually input data?

User Story 3.2: Get meal recommendations based on my dietary preferences and goals.

* Details: Based on user preferences and fitness goals, the app will suggest meals with detailed nutrition breakdown.
* Uncertainties: Will the meal suggestions be dynamically generated from a database or manually curated?

**Epic: Social and Community Engagement**

User Story 4.1: Join fitness challenges and compete with others.

* Details: Users can join fitness challenges (e.g., “Run 10K this month”) and compete against others.
* Uncertainties: What kind of challenges will we offer, and how will progress be tracked (via GPS, manual input)?

User Story 4.2: Share fitness progress with friends and social groups.

* Details: Users can share their progress, workouts, and achievements with their friends and groups within CUFitness.
* Uncertainties: Which social media platforms will we integrate with, or will it be an in-app social feature?

**Epic: Progress Tracking and Insights**

User Story 5.1: Track progress through a dashboard.

* Details: Users can view their fitness progress (e.g., weight, calories burned, workouts completed) in a graphical dashboard.
* Uncertainties: What metrics will be most useful, and how will data be presented visually?

User Story 5.2: Set fitness goals and get progress reports.

* Details: Users can set goals (e.g., “Lose 5kg in 3 months”) and track progress toward those goals through regular reports.
* Uncertainties: Will the system provide weekly or monthly reports, and what kind of notifications will accompany progress tracking?

**2. Identify the Easiest, Hardest, and Middle Stories**

Easiest (1 Point):

* User Story 1.1: Create an account and set up my profile.
  + Simple account creation and basic profile setup.

Hardest (8 Points):

* User Story 2.1: Receive a customized workout plan based on my fitness level and goals.
  + This requires personalization logic, possibly integrating machine learning or complex algorithms, to generate customized plans.

Middle (3 Points):

* User Story 5.1: Track progress through a dashboard.
  + Requires design and data presentation but is relatively straightforward compared to personalized workout plan logic.

**3. Estimate the Effort for Each Story**

Story Points Assignment:

* User Story 1.1: 1 point (easy, basic functionality)
* User Story 1.2: 3 points (a bit more complex but still manageable)
* User Story 2.1: 8 points (customized workout plans require significant effort)
* User Story 2.2: 5 points (adjusting plans requires some logic and dynamic features)
* User Story 3.1: 3 points (meal logging can use existing APIs or barcode scanning)
* User Story 3.2: 5 points (meal recommendations require dynamic, goal-based suggestions)
* User Story 4.1: 5 points (fitness challenges will need tracking and social components)
* User Story 4.2: 3 points (sharing progress is more about UI and integration)
* User Story 5.1: 3 points (progress tracking dashboard with basic graphs)
* User Story 5.2: 5 points (setting goals and tracking reports requires integration of data and notifications)

**4. Find the Median Number of Story Points**

Once we have the individual estimates from all team members, we would share and compare estimates for each user story. After that:

* Calculate the median value of the points (the middle value when all estimates are ordered from lowest to highest).
* If any story has more than 3 points and is too large, we will break it down into smaller tasks.

**5. Decide Which Stories to Keep**

* Keep: User stories that are clear and well-defined (e.g., creating an account, tracking progress, meal logging).
* Break Down: User stories with higher points (e.g., generating customized workout plans, adjusting plans based on milestones) into smaller tasks to ensure they can be completed in iterations.

### **Acceptance Criteria for Each User Story:**

**Epic 1: User Onboarding and Account Setup**

**User Story 1.1: Create an account and set up my profile**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * The user can create an account using an email address and password, or by logging in with SoftGrid credentials.
   * The system should validate the email format and password strength (min 8 characters, mix of letters and numbers).
   * After account creation, the user is prompted to enter basic information (name, fitness goals, fitness level).
   * The profile data is saved securely in the database.
   * The user receives a confirmation email after registration.
2. **Performance**:
   * The registration and profile setup process should not take more than 3 seconds for the user to complete.
3. **Security**:
   * User data, including passwords, is stored securely using hashing techniques.
   * Ensure HTTPS is used for all data transactions.
4. **Error Handling**:
   * Display an error message if the email is already in use or invalid.
   * If the registration fails (due to network or server issues), a friendly error message is shown to the user.
   * If the password doesn't meet security criteria, display a clear message explaining why it’s invalid.
5. **Clarifications**:
   * The user should be able to proceed with basic setup or skip adding fitness preferences for later.
   * Profile data can be updated after account creation.

**User Story 1.2: Input fitness level, goals, and preferences for personalized recommendations**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * The user is prompted to enter fitness goals (e.g., weight loss, muscle gain) and select their fitness level (e.g., beginner, intermediate).
   * The user can select workout preferences (e.g., time of day, workout duration, equipment available).
   * Fitness data is saved and used to generate customized recommendations.
2. **Performance**:
   * The fitness preferences input screen should load within 2 seconds.
3. **Security**:
   * Ensure user preferences and fitness data are securely stored in the database.
4. **Error Handling**:
   * If the user skips any field, show a prompt to complete it or confirm the user wants to skip.
   * If invalid data is entered (e.g., selecting multiple conflicting preferences), display an error message.
5. **Clarifications**:
   * The fitness data should be editable at any time in the profile section.

**Epic 2: Personalized Workout Plans**

**User Story 2.1: Receive a customized workout plan based on my fitness level and goals**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * The app must generate a workout plan tailored to the user’s fitness level and goals.
   * The workout plan must display exercises, duration, frequency, and intensity.
   * The user can see the workout plan in a clear, organized format (e.g., calendar view or list view).
2. **Performance**:
   * The workout plan should be generated in less than 5 seconds after the user enters their goals and preferences.
3. **Security**:
   * The workout data should be stored securely and associated with the user’s account.
4. **Error Handling**:
   * If there’s an issue generating the workout plan (e.g., server error), the system should notify the user with a friendly message and retry after a few seconds.
5. **Clarifications**:
   * The user should be able to view, edit, or skip workouts from their plan.
   * The system should adjust the plan based on missed sessions or progress.

**User Story 2.2: Adjust my workout plan if I miss a session or hit a milestone**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * The app automatically adjusts the workout plan if the user misses a session (e.g., reschedules the session to the next available day).
   * If the user hits a milestone (e.g., completes 30 days of workouts), the system recommends a higher intensity workout or additional exercises.
2. **Performance**:
   * Changes to the workout plan (due to missed sessions or milestones) should be reflected within 3 seconds.
3. **Security**:
   * Any updates to the workout plan should be securely stored and reflected in the user’s account data.
4. **Error Handling**:
   * If the system cannot adjust the workout plan due to an error (e.g., network issue), an error message should notify the user and allow them to retry.
5. **Clarifications**:
   * The user must be notified when the plan is updated due to missed sessions or milestones.
   * The adjustment must be appropriate to the user’s progress and preferences.

**Epic 3: Nutrition and Meal Tracking**

**User Story 3.1: Log meals and track calories**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * Users can log their meals manually by searching for food items or scanning barcodes.
   * The system should calculate and display the total calories and macronutrients (carbs, proteins, fats) for each meal.
   * The user can edit or delete meal entries.
2. **Performance**:
   * The meal logging page should load within 2 seconds and submit meal data within 1 second.
3. **Security**:
   * Meal logs and nutritional data are stored securely and linked to the user's account.
4. **Error Handling**:
   * If a food item is not found during the search, display a message allowing the user to manually enter the data.
   * If barcode scanning fails, provide an option to enter the data manually.
5. **Clarifications**:
   * The app should display nutritional information based on the food database (which may be third-party integrated).

**User Story 3.2: Get meal recommendations based on my dietary preferences and goals**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * The system recommends meals based on the user’s dietary preferences (e.g., vegetarian, gluten-free) and fitness goals.
   * Each meal recommendation must include nutritional breakdown (calories, macronutrients, portion sizes).
   * Users can save meal recommendations to their daily meal plan.
2. **Performance**:
   * Meal recommendations should be generated and displayed within 5 seconds after input.
3. **Security**:
   * Dietary preferences and meal logs are securely stored and associated with the user's account.
4. **Error Handling**:
   * If no matching meals are found based on the user's preferences, the app should notify the user and suggest broader meal categories.
5. **Clarifications**:
   * The meal recommendations must be dynamic, updating based on the user’s changing goals and preferences.

**Epic 4: Social and Community Engagement**

**User Story 4.1: Join fitness challenges and compete with others**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * Users can view available fitness challenges (e.g., “Run 10K this month”) and join them with a single click.
   * Progress toward challenge completion is tracked (e.g., distance run, days exercised).
2. **Performance**:
   * Challenge data (progress, leaderboard) should load within 3 seconds.
3. **Security**:
   * User participation and challenge data are stored securely.
4. **Error Handling**:
   * If there is a problem with tracking challenge progress (e.g., GPS not available), display an error and allow manual entry.
5. **Clarifications**:
   * The user should be able to leave a challenge at any time and still track their progress for future challenges.

**Epic 5: Progress Tracking and Insights**

**User Story 5.1: Track progress through a dashboard**

**Acceptance Criteria:**

1. **Functional Requirements**:
   * Users can view their fitness progress (workouts completed, calories burned, meals logged) in a graphical dashboard.
   * Data should be presented clearly in charts (e.g., bar, line) or progress bars.
2. **Performance**:
   * The dashboard should load within 2 seconds and update in real-time as the user logs workouts or meals.
3. **Security**:
   * All progress data is stored securely.
4. **Error Handling**:
   * If data is unavailable or fails to load, show an error message explaining the issue and prompt the user to refresh.
5. **Clarifications**:
   * The user should be able to filter progress by time (e.g., daily, weekly, monthly).

These Acceptance Criteria define clear requirements, performance standards, and error-handling procedures for each story, ensuring that the team has a comprehensive understanding of what is expected for the feature to be considered done.