NutriTrack – Assignment 3 Specification

Version: 1.0 l 14 April 2025
Assignment Name: *NutriTrack Pro: Database, Personalisation & Insights Expansion*
Unit: FIT2081 – Mobile App Development
Due Date: Week 11, Friday 11:55PM
Assessment Weight: 30% (App: 80%, Interview: 20%)

1. Overview

In this final phase of the NutriTrack journey, your task is to **extend your existing app** from Assignment 1 into a **fully-featured nutrition insights platform**.

You will:

- Move all CSV data into a **Room database**
- Introduce **multi-user login/logout** with basic authentication
- Build the **NutriCoach screen** using a third-party Fruit API (https://www.fruityvice.com/) + GenAI*
- Implement a **Settings screen** with admin analytics + GenAI
- Ensure all data handling goes through **ViewModel** and adheres to **MVVM architecture**

* Note: we will be using Gemini API (https://aistudio.google.com but you are welcome to use any other AI API as long as you provide the teaching team with an API key and instructions for remote access as part of your submission. **Critical**: Make a private Ed post letting us know your intentions to use a non-standard AI tool before the deadline to gain approval to do this.

2. Technical Requirements

A. Database Setup (10%)

- On **first launch only**, load all patient data from the provided CSV into a local **Room database**.
- Structure:
  - `Patient` table: includes UserID, PhoneNumber, Name, Sex, and all scores
  - `FoodIntake` table: store responses from the questionnaire with a foreign key column relationship with Patient table.
- After the first run, **do not read from CSV again**.

B. Login System: Changes (10%)

- Users **claim** their account on first login by:

    - Providing valid **UserID + PhoneNumber**
    - Setting a password and adding their **name** (stored in DB)

- Subsequent logins will require only *UserID + password*.
- Add support for **Logout**. Later in the spec, we will be adding a **Settings screen** where logout is possible. Keep user logged in if they close the app and return (i.e. no login required unless user manually logs out).

3. App Features

C. Core Screens (25%)

UI: Welcome / Login (Updated)

- Add account claim flow
- Validate user against DB (not CSV)

UI: Settings Screen

- Add a new Settings screen (accessible from bottom nav)
- Show logged-in user info such as name, phone number.
- Include **Logout** button which logs out the user and takes them to the login screen.

D. NutriCoach (15%)

- This screen will be unlocked via the **"Improve my diet"** button on the Insights screen OR using the bottom bar's NutroCoach button.
- The top 50% of the screen will contain a Fruits section
- The bottom 50% of the screen will contain a GenAI section.

Features:

- Fruits Section: Connect to the **FruityVice API** to retrieve fruit-related data:
    - For example: if user's fruit score is low, allow user to type some fruit names and show them fruit facts and stats

- GenAI Section: Use **GenAI integration** to show one **motivational message** or **fun food tip**
    - May use prompt: `"Generate a short encouraging message to help someone improve their fruit intake."`

NutriCoach: HD to HD++ Marks

- **Use conditional logic** -- only show the Fruit selection option in NutriCoach if user's fruit score is not optimal (read the scoring guide in Appendix for more information on what is optimal and non-optimal Fruit HEIFA scores). If user's fruit score is optimal, just display a random image loaded from https://picsum.photos/
- **Improve GenAI response** -- improve the specificity of GenAI response by sending additional information

to the GenAI model such as all the values of the patient and their food intake result.

- **Store GenAI responses in DB** -- Each time user gets a GenAI response, it needs to be stored inside a table inside your database called `NutriCoachTips`. Add a button at the bottom of this section that opens a Modal that shows all tips shown to this user previously.

E. Clinician/Admin View - HD to HD++ marks (20%)

Within the Settings screen, implement a **button for "Admin View"** (open a clinician login screen and if they enter a special key they are allowed to see the Admin View screen). The key = "dollar-entry-apples" (without quote marks).

Features:

- Show the following stats:
  - Average HEIFA score for male users
  - Average HEIFA score for female users

- Use GenAI to display **3 interesting patterns** in the data, e.g.
  - "Most users who scored highly on vegetables also had high fruit scores"
  - "Female users had higher average fruit variation than male users"

---

4. Data & Architecture

- All data access must use **Room** via **Repository** → **ViewModel** pattern
- App should follow **MVVM architecture** as introduced in Week 9
- Use State/Flow to update UI reactively (or LiveData)

---

5. Submission Requirements

A single ZIP file containing

- ✅ Full source code

Other requirements (reminder):

- ✅ CSV must be used once only to populate Room DB
- ✅ Coding interview in your Week 12 lab

---

6. CSV Sample + API Info

- Provided CSV: includes userID, phoneNumber, sex, and all score breakdowns
- API: FruityVice API
- GenAI usage for assignment: In line with Monash policy, all GenAI usage needs to be declared.

---

7. Additional Notes

- You may **reuse your Assignment 1 code**, but a refactor into MVVM is expected.
- All networking must be done asynchronously using **Retrofit + coroutines**
- UI should adapt gracefully to different screen sizes