**CS-360 Project One App Development Proposal**

**Option 3: Weight-Tracking App**

Ulises Almaguer

1. **Project Goals:**
   1. The mobile application scenario that I have selected is Option 3 Weight-Tracking application. The main goal is to be a personal health companion, helping users track their weight, set goals, and see their progress. It will be a simple yet effective digital logbook for weight management. Users can log in, see their weight history on a grid, easily add new weights, and set their target. The app notifies them when they hit their goal.

**App Features:**

* + 1. Database with three tables daily weights, user logins, and goal weight.
    2. Login/Signup screen for secure access.
    3. Daily Weight Grid screen to display all recorded weights and dates.
    4. Mechanism to add daily weight and set a goal weight.
    5. Notification system to alert users when they reach their goal.

1. **Application Users:**
   1. This app will primarily target people who are actively managing their weight and. I see three main user types who could potentially use this app or at least will find it beneficial.
      1. The first person that I see using the app will be those users that are dedicated to tracking their daily weight to see trends and to stay accountable in their weight loss journey.
      2. The second type of users are those who set goals for motivation and are focused on hitting specific targets
      3. The third base of users are those who just care about their health and want to monitor the weight fluctuations and just in general be health conscious.
2. **UI Design:**
   1. **Welcome / Login & Signup Screen:**
      1. Purpose: Secure entry point for all users.
      2. Features: Welcome message, email/password inputs, "Login" button, "Create Account" link, "Forgot Password?" link.
      3. User Centered Design: Streamlined login/signup, clear guidance.
   2. Daily Weight Tracker screen
      1. Purpose: Main hub for viewing history and adding new weight.
      2. Features: Large display of current and goal weight, visual progress indicator "Add Weight" Floating Action Button (FAB), scrollable weight history grid, bottom navigation (Home, Goals, Profile).
      3. User-Centered Design: Prioritizes key info, makes primary action (add weight) easily accessible, displays scannable history.
   3. Add Weight Screen:
      1. Purpose: Quick and accurate daily weight input.
      2. Features: Numeric weight input, units selector, date/time picker (optional), "Save" and "Cancel" buttons.
      3. User-Centered Design: Single-task focus, clear inputs for simplicity.
   4. Set Goal Screen:
      1. Purpose: Define or update target weight.
      2. Features: Goal weight input, current goal display, "Save Goal" and "Cancel" buttons.
      3. User-Centered Design: Simple fields for a single, important task.
   5. Notifications:
      1. Purpose: Timely reminders and celebratory alerts.
      2. Features: Goal achievement notification (congratulations message).
      3. User-Centered Design: Crucial for motivation and positive reinforcement.
3. **Code Design:**
   1. User Authentication (Login/Signup Screen):
      1. Code: An Authentication Module handles registerUser() and loginUser() functions.
      2. UI: EditTexts take user email/password. "Login" and "Create Account" buttons trigger calls to the Auth Module, sending this data. Error messages are displayed in a TextView.
      3. Data Flow: UI (input) → Code → Database (store/verify) → UI (feedback).
   2. Daily Weight Tracking:
      1. Code: A Weight Data Manager handles addDailyWeight(), getDailyWeights(), and getLatestWeight(), interacting with the database.
      2. UI: Home Screen: TextView displays currentWeightText (from getLatestWeight()). A RecyclerView displays weightGrid (from getDailyWeights()). A FAB triggers navigation.
      3. Add Weight Screen: EditText takes weightInput. Date/Time pickers take input. "Save" button triggers addDailyWeight().
      4. Data Flow: UI (input) --->Code --->Database (store) --->Code (retrieve) --->UI (display).
   3. Goal Weight Management (Home & Set Goal Screens):
      1. Code: The Weight Data Manager also handles setGoalWeight() and getGoalWeight().
   4. UI: Home Screen: TextView displays goalWeightText (from getGoalWeight()).
      1. Set Goal Screen: EditText takes goalInput. "Save Goal" button triggers setGoalWeight().
      2. Data Flow: UI (input) ---> Code Database (store) --->Code (retrieve) --->UI (display).
   5. 4. Goal Achievement Notification:
      1. Code: The Weight Data Manager has checkGoalAchievement() logic, and a Background Service/Notification Module sends alerts.
      2. UI: Appears as a system-level notification.
      3. Data Flow: Database (weights/goal) --->Code (check) --->System Notification.