The goal of this project is to develop an app that geared towards user-friendliness & helping them achieve their goals via a weight tracking application. This allows the user to monitor their daily weight while working in confines set & imposed by the user. Unlike existing weight-tracking solutions, this app will prioritize security, multi-user support, and customizable progress tracking, catering to a diverse range of users with different health and fitness objectives.

Our app offers to provide users a seamless system for logging their weight, tracking progress, & receiving motivational updates. It will have features to provide secure user authentication, allowing multiple users to sign into a single device by username & password. Each login will greet the users a motivational message, last login summary, of recent entry information, & progress to target. The app will offer a weight history, a trends sections, which will display information in a graphical style. Users can set future weight goals in allotted time limits & receive daily reminders. There should also be some community features like a message board that allows users to share the progress and uplift each other or compete.

For competitive analysis, two apps, *My Fitness Pal* and *Simple Weight Tracker*, were evaluated. Both apps display a summary table upon launch, showing metrics such as current weight, weight change, weekly trends, monthly changes, and total weight change. *Simple Weight Tracker* offers additional features like goal prediction tracking time spent on the user journey, and monitoring BMI. However, neither app supports multiple users or provides data protection. Furthermore, most existing apps, including the two analyzed, primarily cater to users aiming to lose weight. Our app will not make such assumptions, accommodating users with both weight loss and weight gain goals. All features will be designed to work equally well for both user groups with any bias expectations.

Our app will be designed with simplicity & useability in mind, making sure to adhere to the Android Design & Quality Guidelines. When first loading app, users will be directed to a login screen where they will enter their credentials. Once logged in, they will be taken to the dashboard, which will display an overview of their progress, including the most recent weight entry, the time remaining for their goal, and reminders. Using the dashboard, users can navigate to the weight entry screen, where they can log their current weight, which will update the database. A history & trends section allows users to view their past weight entries in graphical formats. The goal management screen will provide options to set new weight goals, define target deadlines, and enable reminders. A settings screen allows users to customize preferences, notifications, and security settings. Notifications will play a crucial role in the app, reminding users to update their weight according to a predefined schedule and alerting them when they achieve their goal. These notifications will comply with the VX-S1 standard in the Android Design and Quality Guidelines. Additionally, all buttons, graphics, and text will follow the VX-A1, -A2, and -A3 standards.

The app will adopt a Model-View-Controller (MVC) architecture. The *Model* layer will include classes such as *Users.java*, *Goals.java*, and *Records.java*, which will define user attributes (e.g., name, admin status, password), goal attributes (e.g., target weight, set date, expiration date, username), and weight record attributes (e.g., username, date, time, weight), respectively. The *View* layer will consist of XML files like *main\_activity.xml*, which will define our user interface, including the login screen and other potential screens or fragments. The *Controller* layer, comprising classes like *RecordsActivity.java*, *GoalsActivity.java*, and *DatabaseDAO.java*, will handle the logic connecting the model and view layers, as well as manage database interactions, likely through a singleton pattern.

Development of the app, we should incorporate implementation units & integration testing as our priority. By following best practices, we will have comprehensive test coverage & prevent regression when adding in new features. The integration testing will improve our app’s reliability & simplify overhead cost of maintaining the app.