**Weight Tracking App**

**Project One: CS-360**

Brandon Hobbs

January 16, 2023

The app being discussed in this report will be a simple tool that allows multiple users to track their weight goals. Upon searching the Google App store there are few other simple weight tracking apps; however, the top returns have a few limitations that this app will bypass to allow it to be competitive in this space.

“Simple Weight Tracker” and “Weight Tracker” were analyzed as a means of competitive analysis. Both greet the user with a small summary table showing actual weight, change, the weekly trend, this week’s and this’s months change, and the total weight change. “Weight Tracker” adds prediction ability (trying to predict when the goal will be reached), time spent on the user journey, and track BMI.

Neither of these apps appear to allow multiple users to track their goals and weight not do they have any protection on the data – which will be the selling point of this app over its closest rivals. Moreover, the two apps analyzed and many other available apps only target users wanting to lose weight. This app will not make that assumption.

The types of users who might use an app such as this would fall into two classes: those wanting to lose weight and those wanting to gain weight. All features should work equally well for either class of user.

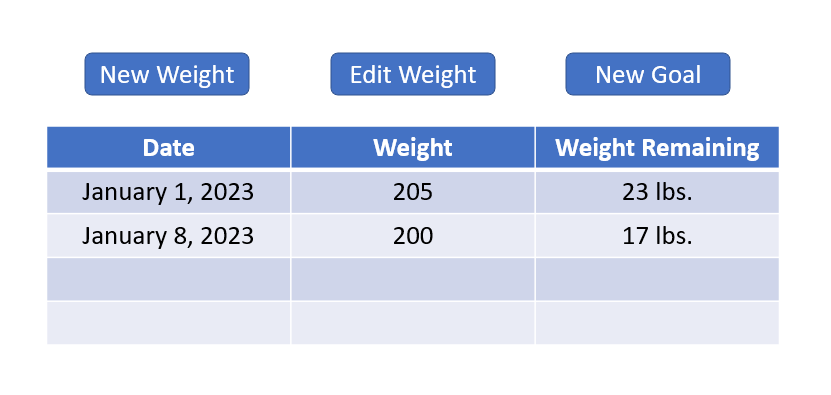
Over the course of a day, a user will have very little interaction with the app but at key moments, e.g., morning ablutions, there will be data entry, review, and editing. During these brief interactions data entry should involve as few clicks as possible. The design hypothesis is that the user will spend most time reviewing data instead of entering and editing data.

Upon opening the app, the user will be greeted by their name with a message showing pertinent data. For example, this data might include:

* Last app opening date and time
* Last time recording a weight
* Time remaining within the time-bound goal
* Delta between goal weight and last recorded weight

There would also need to be a way to dismiss this pop-up message.

After this message is closed one might expect a simple table to be present that shows a tabular view of data. Above this table could be the control buttons. An example layout is shown below:



When a user clicks “New Weight” a popup dialog will appear asking for the pertinent details, e.g., time weight is taken (may be defaulted to the current local time) and the weight.

“Edit Weight” will produce a table with all of the database entries assigned to the user who is logged in. The user will need a method to select a record and then be presented with the option to edit the weight or time. Conversely, the app may allow a user to only delete a record and then add a “new” record which contains the edited data.

“New Goal” will create a goal for the user who is logged in. This can be a weight increase or loss. There will be no option to edit or delete this goal as the action of creating a new goal will be used for editing/updating.

Because this app will allow multiple users, there will need to be a screen to administrate user accounts, i.e., add users, delete users, set admin account. At a lower priority, or even possible beyond the scope of the MVP, will be another screen that plots the weight versus the date.

As shown in VX-N1 and VX-N2 Navigation standards of the *Android Design and Quality Guidelines* the back button and swiping shall be supported to move between these screens. VX-U1 suggests that both landscape and portrait orientations should be supported, but portrait is probably the more useful so the MVP version of the app might forego landscape unless users demand that feature.

Notifications will be employed to alert a user if they fail to update their weight at a prearranged schedule. They will also be used once a user has met their goal. Notifications will follow VX-S1 in the *Android Design and Quality Guidelines*.

Lastly, VX-A1, -A2, and -A3 will be followed for any buttons, graphics, and text.

The app will follow an MVC (model-view-controller) design. A general overview of the architecture scheme is shown below.

|  |  |  |
| --- | --- | --- |
| **Model** | **View** | **Controller** |
| 1. Users.java 2. Goals.java 3. Records.java | 1. main\_activity.xml | 1. RecordsActivity.java 2. GoalsActivity.java 3. DatabaseDAO.java |

At a high level, the Users class will have the attributes necessary to describe a user, e.g., name, Admin status, password.

The Goals class will hold the attributes necessary to describe a goal, e.g., weight goal, date set, date to expire, username.

Records will have the attributes needed to describe a weight record, e.g., username creating the record, date created, time created, weight.

The main\_activity will hold the view a user sees when logging in (as shown previously). Other screens may need their own xml but these screens are not yet designed and may be fragments instead of a full activity.

The controllers will have the logic that connects the different classes to the UI. The controllers will also contain the database connection, more than likely through a singleton.

As the app is being developed creating unit tests should be considered. As a best practice, implementing unit and integration tests while work is ongoing ensures better coverage of the application. Moreover, as new features are implemented or expanded these unit tests can help to protect against any regressions.

**Citations**

Hobbs, B. (2023). CS 360 project one prep writing prompt. [Unpublished report]. SNHU.