# PetFindr Health Tracker Prototype Documentation

Landing Screen:

A screenshot of a cell phone

Description automatically generated with medium confidence

The landing screen includes a readout of the current Total Steps and Goal at the top of the screen. When Total Steps exceeds the Goal the numbers turn green and a green cheer message appears at the bottom of the screen. Four buttons are included as a navigation menu. The TinyDB1 non visible component is required to retrieve the Total Step and Goal values from other pages. The other non-visible components are placeholders for a later prototype.

The blocks view of the code for page 1 is shown below:

Timeline

Description automatically generated

Workout Page

A screenshot of a cell phone

Description automatically generated with low confidence

The workout page uses the pedometer non visible component to count steps and the TinyDB1 component to store them. Due to limitations in MIT app inventor the app cannot be continuously running in the background, so the counter is set up as a stopwatch before we move to more advanced coding tools. Pressing start begins the count, tapping stop working out pauses it and holding the button zeros the counter on this page and the landing page. Menu options to the landing page and goals page are included.

The blocks view of the code for the workout page is included below:

Timeline

Description automatically generated

Goals Page:

Table

Description automatically generated with medium confidence

The goals page uses the TinyDB1 component to store the user entered goal.

The block view of the code for the goals page is included below:

Diagram

Description automatically generated with medium confidence

History Page:

Graphical user interface, table

Description automatically generated

The History page simulates at later function where the app will record the continuous data it receives and generate graphs to present the information to the user. Due to limitation with the MIT app inventor this is simulated by having the user enter test step and goal data and then adjusting the size of the following bars based on the step data to create the graph. The bars change colour if the step in that position in the array exceeds the corresponding goal. The TinyDB1 is used here to store the step data to be used as calories out information on the Calories page.

The block view of the code for the history page is shown below:

Timeline

Description automatically generated

Timeline

Description automatically generated

Calories Page:

A picture containing graphical user interface

Description automatically generated

The Calories page uses TinyDB1 to populate a data for calories burned based on the test step data entered by the user on the History page. The user enters test values for calories eaten per day to create the calories eaten. If calories burned exceeds calories eaten the relevant bars turn green. The barcode scanner non visible component is included as a place holder for a later version of the prototype that will receive the calories eaten by scanning pet food barcodes.

The block view of the code for the Calories page is included below:

Graphical user interface

Description automatically generated with medium confidence

Timeline

Description automatically generated

Timeline

Description automatically generated

Heart-Temp Screen:

Calendar

Description automatically generated with medium confidence

The Heart-Temp screen simulates a function that will later receive data from sensors on the PetFindr collar. Current temp and heartrate are static values and the safe range for temp and heartrate are entered by the user. If the current value falls outside of the safe range a red warning message for the exceeded parameter is displayed. Later prototypes will include a push notification in this scenario. The thermometer and Accelerometer sensors are used as placeholders for the sensors on the collar.

The block view of the code for the Heart-Temp screen is included below:

Timeline

Description automatically generated