**Figure 7-1 – Possible System Entities**

Product: Elevate

Team: 46

Date: 02/22/2017

|  |  |  |
| --- | --- | --- |
| **Type** | **Name** | **Description or Notes** |
| screen | homeScreen | This screen is when users are presented when they launch the app. Shows recent locations and location search bar. |
| screen | dataScreen | When user swipes right from homeScreen, user will see this screen, which shows a small amount of data about them at bottom of screen. If a location was entered or hill selected, this screen will be showing the directions. |
| screen | hillScreen | When user swipes left from homeScreen, this screen will show users hills around their general location. |
| screen | hilldataScreen | On hillScreen, if user selects a hill, this screen will provide data about the selected hill (elevation, distance, travel time). |
| screen | indepthScreen | On dataScreen, if user selects in-depth view, this screen will provide them with more detailed information. |
| database | gpsJSON | Storage location for coordinates from helmet / mobile application |
| file | collJSON | Storage location for collision data from helmet |
| code | arduino | Code for Arduino in order to provide helmet functionality |

**Figure 7-2 – Template for Detailed Design for a Screen**

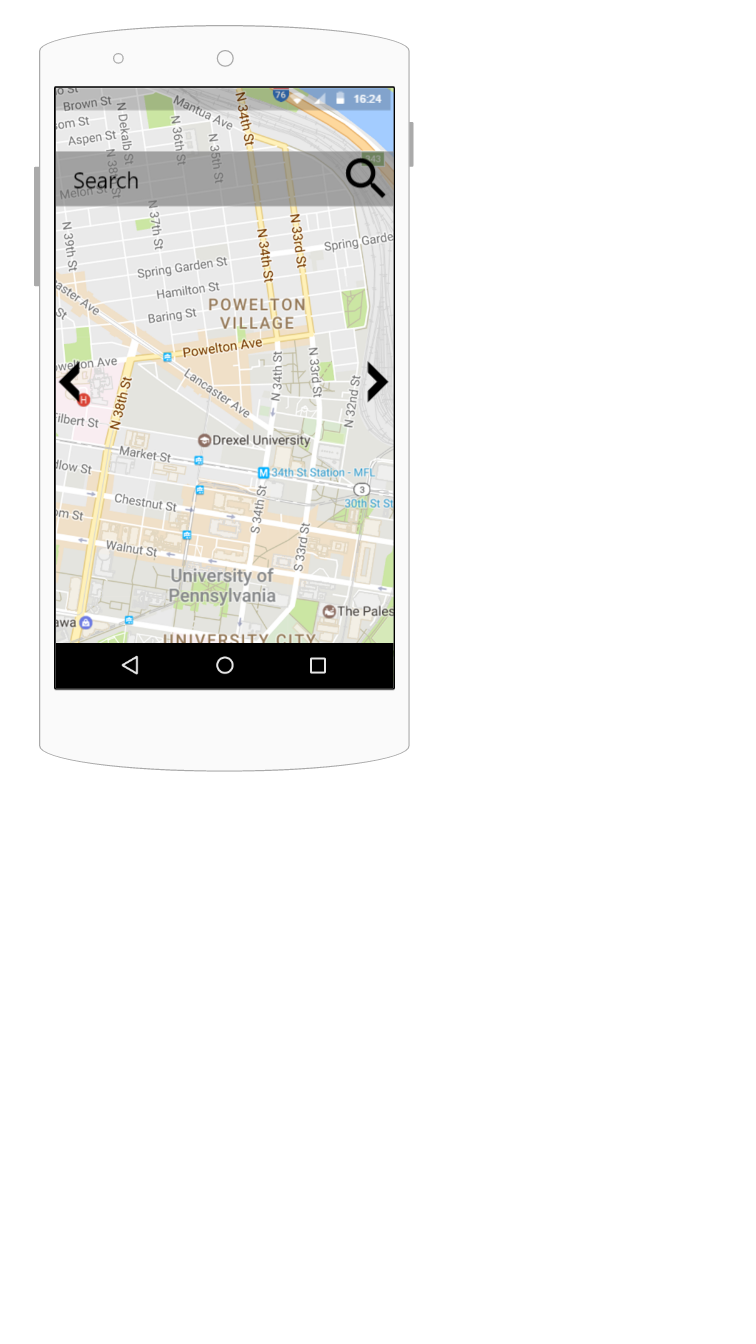
**Name:** homeScreen

**Type:** Screen

**Purpose:** This screen meets requirement 6.

**Description:** This screen contains a search bar, used to enter location to get directions, and option to see a list of recently visited locations.

**Layout:**



**Name:** hillScreen

**Type:** Screen

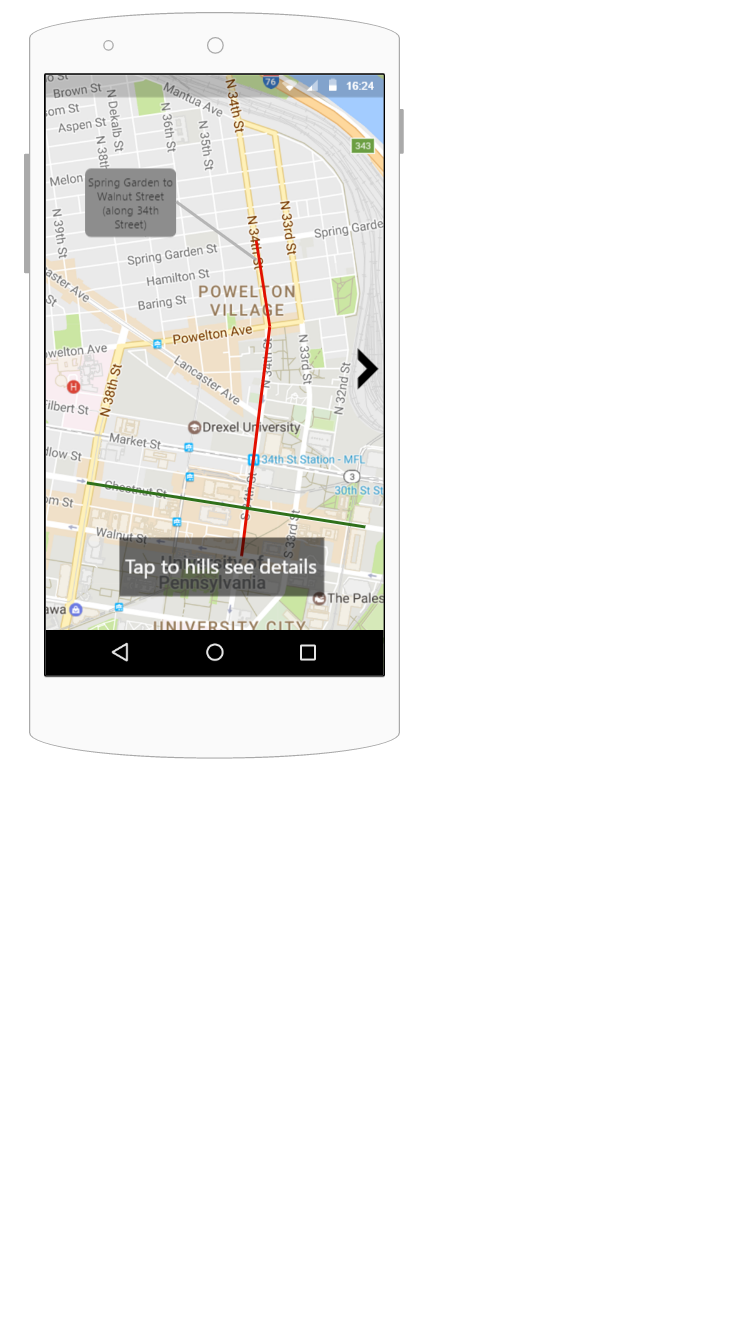
**Purpose:** This screen meets requirement 6.

**Description:** This screen contains an interactive map.

Hills with considerable elevation change will be highlighted.

If a user clicks on a hill, the hilldataScreen will be shown.

**Layout:**



**Name:** dataScreen

**Type:** Screen

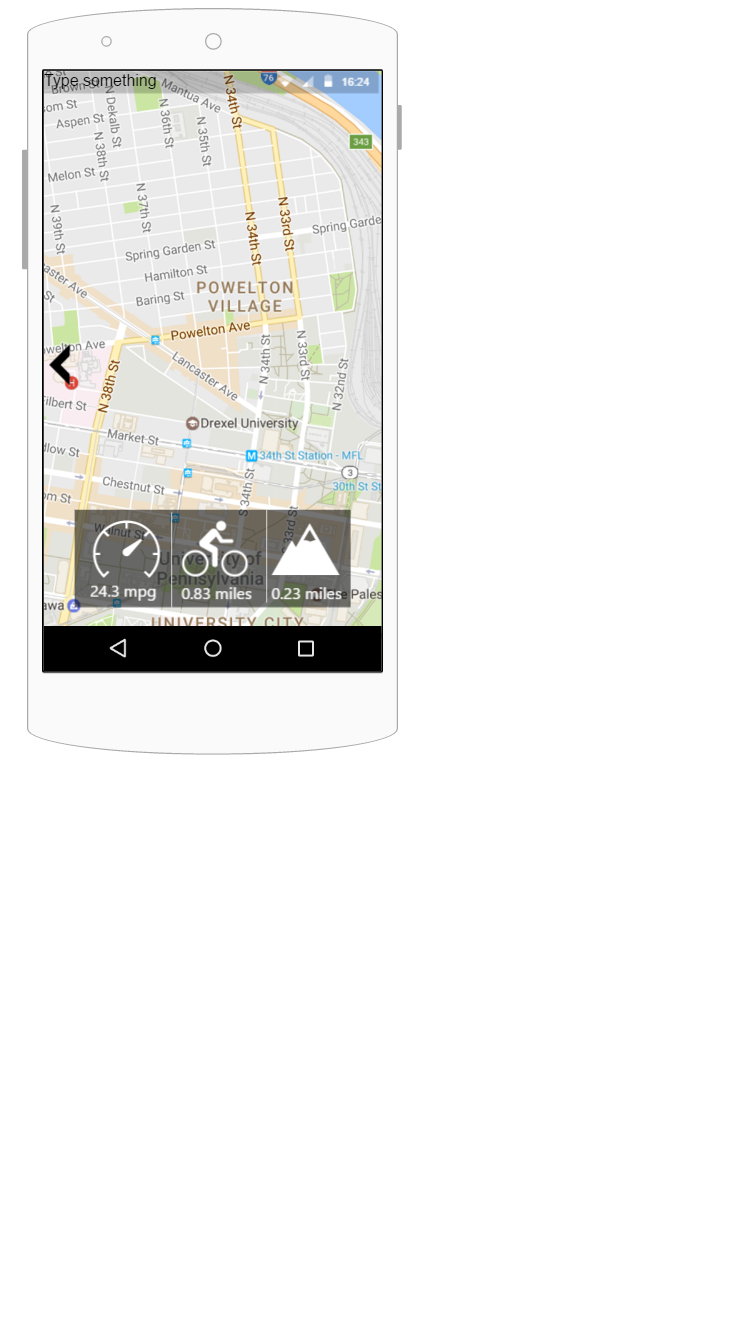
**Purpose:** This screen meets requirement 6.

**Description:** This screen is the screen where directions are shown if a hill is selected or location entered.

The screen also will always show the user some data about them.

(distance traveled, current speed, elevation)

**Layout:**



**Name:** indepthScreen

**Type:** Screen

**Purpose:** This screen meets requirement 6.

**Description:** This screen shows the user more in-depth information about their ride including :

(distance traveled, time traveling, max speed, current speed, current elevation,

net elevation change)

**Layout:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Figure 7-3 – Template for Detailed Design for a Database Table**

**Name**: gpsJSON

**Type**: Database Table

**Purpose**: This table is needed to meet requirements 5 and 7.

**Description**: Figure 1 shows the contents for this table. This table provides a way to store data sent from our system in order to manipulate the object and its’ properties simply and effectively. One row of this table represents an JSON objects’ properties.

**Table Contents:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Element Name** | **Data Type** | **Usage** | **Notes** |
| latitude | JSON | gmapAPI | Latitude coordinate of the object (data sent) |
| longitude | JSON | gmapAPI | Longitude coordinate of the object |
| elevation | JSON | gmapAPI | Elevation measurement of the object |
| distance | JSON | gmapAPI | Distance reading of the object |

**Figure 1 - gpsJSON Database Table**

**Figure 7-4 – Template for Detailed Design for a Code Function**

**Name**: void loop()

**Type**: Function

**Purpose**: This function is needed to meet requirements 2 and 4.

**Parameters**: No parameters are used to call this function. Given below is variables within the program that are manipulated.

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Notes** |
| distanceSensor | pin | Data exchange stream between our ultrasonic module and Arduino |
| piezoBuzzer | pin | Data exchange between our buzzer and the Arduino |
| collisionSensor | pin | Data exchange stream between our collision sensor and the Arduino |
| gpsSensor | pin | Data exchange stream between our GPS sensor and the Arduino |
| motion | int | Variable used by ultrasonic module to determine range of detection for the sensor |
| t\_interval | int | Variable used to determine how long our GPS will record coordinates |
| voltage | int | Variable used to determine how much voltage we will send our modules upon detection of motion / collision. |
| lat | double | Variable used to store our latitude coordinates from GPS sensor |
| lon | double | Variable used to store our longitude coordinates from GPS sensor |

**Return Type**: Does not return a data type, but rather alerts users via piezo buzzer alarm

**Processing:** The function, upon the Arduino receiving power, will initialize the modules by sending voltage to the GND/5V pins. The function will declare data objects associated with the sensors using distanceSensor, piezoBuzzer, collisonSensor, and gpsSensor. The function will constantly record over the previously declared *t\_interval* GPS Coordinates into *lat* and *lon* for storage. The ultrasonic object, when motion is detected within the range of the previously declared *motion* variable, will send an amount of voltage across its’ data pin (distanceSensor). The function will then send that voltage to the piezoBuzzer in order to alarm the user. If collision is detected, the collision sensor will send an amount of voltage through its’ data pin (collisionSensor) in order to be logged to a file within the Arduino processor. The function will constantly check motion and collision, hence the void loop() declaration.

**Figure 7-5 – Team Capability Assessment**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Matt Horger** | **Kevin Tayah** | **Jake Rauchen** |
| **Arduino Hardware** | 3 | 2 | 2 |
| **Hardware Programming** | 3 | 3 | 3 |
| **Database Structure** | 2 | 2 | 2 |
| **Data Transfer** | 2 | 2 | 2 |
| **Mobile Application** | 2 | 3 | 3 |

\*\* The table values represent an assessment of team member capabilities. The values are:

1 – No knowledge or not much relative to the needs of this project

2 – Enough knowledge to accomplish part but not all of this project

3 – Knowledge probably sufficient for this project