

# Personal Unit Test Report

**Product Name: SlugDex**

**Team Name: The Slug Surveyors**

**Team Member Name: Kyle Lim**

**Revision Date: 11/29/2022**

## Files Tested

settingsPage.dart

- Test Cases:
  - Page scales the settings layout according to width and height constraints of device
    1. Create an emulator instance with a 16:9 aspect ratio
    2. Create 2 more emulator instances with wider and thinner aspect ratios
    3. Debug the app on each emulator and bring up the settings page
    4. Visually inspect the layout for satisfaction
    5. Use the widget inspector tool to view specific pixel dimensions and overlapping areas
  - Changes to settings persist throughout the app
    1. Open the app and navigate to the settings page
    2. Toggle the developer mode slider
    3. Navigate away from the screen, or even close the app
    4. Check the program variable in debugger and see that it has been updated
  - The 'Logout' option works
    1. Open the app and log in to any account
    2. Navigate to the settings screen and tap the logout button
    3. Ensure that the screen promptly changes to the auth page
    4. Check the program variables in the debugger and see that the current user is null
  - The 'Delete account' option works
    1. Open the app and create a new account
    2. In the firebase console, confirm that the new account is now in the authentication table
    3. Navigate to the settings screen and tap the 'delete account' button
    4. Confirm the deletion in the dialog
    5. Ensure that the screen promptly changes to the auth page
    6. From the firebase console, ensure that the new account no longer exists in the authentication table
  - Pressing "Custom Map" navigates to the prototype artist's rendition of campus
    1. From the settings page, press "Custom Map"
    2. Ensure the testImageMap screen is loaded

3. Press the back button and ensure that the settings page is reloaded
- Pressing “Edit Profile” navigates to the appropriate page
  1. From the settings page, press “Edit Profile”
  2. Ensure the editProfilePage screen is loaded
  3. Press the back button and ensure that the settings page is reloaded
- Profile Pic and Username automatically update
  1. Open the app and log in to any account
  2. Open the firebase console and view the userData table
  3. Under the account, change the associated username
  4. In the firebase storage portal, change the associated profile image
  5. Navigate to the settings page and ensure that the displayed image and name correspond to those in the database

settingsTools.dart (contains resources for the above page)

- Test Cases:

- n/a

LiveMapScreen.dart

- Test Cases:

- Populates the map with circles for each undiscovered entry
  1. Initialize the entry list to all be undiscovered
  2. Open the app and ensure the number of circles = the number of entries
  3. Manually mark an entry as discovered
  4. Rebuild and ensure the number of circles present is one less
- Tapping a circle triggers a discovery sequence only when in range
  1. Open the app and set the emulators location outside of any circles
  2. Tap on a circle and ensure nothing happens
  3. Set the emulator’s location to inside a circle
  4. Tap on that circle and ensure that the discovery animation is played for that specific entry
- Circle radius and color matches each entry
  1. Initialize the entry list to all be undiscovered
  2. Initialize the rarities all to Mythical (0)
  3. Ensure all entries are the same color and match what color they should be for Mythical
  4. Change the first entries rarity to Legendary, Rare, Uncommon, and Common
  5. Ensure the color of the circle updates correctly upon rebuild
  6. If that circle updates correctly the assumption can be made that the others will update correctly as well
- Sliding panel properly scales and moves button elements

1. Open 2 emulators of different screen heights and build the app
2. Move the sliding drawer up and down
3. Ensure that the buttons animate up and down with the movement of the drawer.
4. Ensure that the buttons remain clickable in all states
- Pressing the GPS re-centering button centers the user location
  1. Open an emulator and build the app
  2. Set the emulator's location to a specific point
  3. Navigate to the live map screen and move the view away from the user location
  4. Press the re-center button and ensure that the view correctly animates to be centered on the user location
  5. Repeat as needed by changing the gps location of the emulator
- Pressing the floating profile button navigates to the profile page
  1. From the settings page, press floating avatar image
  2. Ensure the profilePage screen is loaded
  3. Press the back button and ensure that the live map screen is reloaded

profilePage.dart

- Test Cases
  - Sliding Panel scales UI elements
  - Amount discovered reflects user data
    1. Initialize the entry list to all be undiscovered
    2. Navigate to the profile page and ensure the number of discovered is '0 / 25'
    3. Manually mark an entry as discovered
    4. Rebuild and ensure the number of discovered is now '1/25'
  - Pressing the settings button navigates to the appropriate page
    1. From the profile page, press the gear icon
    2. Ensure the settingsPage screen is loaded
    3. Press the back button and ensure that the profilePage is reloaded
  - Profile Pic and Username automatically update
    6. Open the app and log in to any account
    7. Open the firebase console and view the userData table
    8. Under the account, change the associated username
    9. In the firebase storage portal, change the associated profile image
    10. Navigate to the profile page and ensure that the displayed image and name correspond to those in the database

editProfilePage.dart

- Test Cases:

- Edit username and profile image buttons
- Profile Pic and Username automatically update
  11. Open the app and log in to any account
  12. Open the firebase console and view the userData table
  13. Under the account, change the associated username
  14. In the firebase storage portal, change the associated profile image
  15. Navigate to the profile page and ensure that the displayed image and name correspond to those in the database

#### DexEntryPage.dart

- Test Cases:
  - Heading and AppBar accommodate sliding panel
    1. Navigate to the live map screen
    2. Ensure that the sliding drawer is easily draggable
    3. Scroll through the Dex Entry Page in the drawer
    4. Ensure that all 25 elements are fully viewable and clickable
    5. Ensure that the draggable 'handle' is always on top of the content

#### testImageMap.dart

- Test Cases:
  - Populate markers on map for each entry
    1. Navigate to the "Custom Map" screen from the settings page
    2. Ensure the number of markers = the number of entries (25)
  - Marker locations correspond to real-world locations
    1. Open a side-by-side window of google maps with the saved pins of each entry
    2. Open the app to the test image map page
    3. Visually inspect the accuracy of each pin on the map with google maps
    4. Note in which direction the markers appear to be skewed
    5. In the same axis, choose an identifier on each edge of the custom map to act as bounds
    6. Count the number of pixels from the edge to those identifiers
    7. Add from zero on one edge to get the lower bound, and subtract from the width or height, respectively, to get the upper bound
    8. On google maps, find the same identifiers and record the exact coordinates
    9. Set the upper and lower bounds from google maps coordinates to pixel coordinates in the code for linear conversion
    10. Rebuild the app and repeat this process as necessary
  - Clicking on a marker displays some information about the entry
    1. Navigate to the "Custom Map" screen from the settings page

2. Tap on a marker and ensure that the associated name of the entry appears on screen

### **Additional Testing**

All UI Elements:

- General Test Cases for each screen:
  - Ensure that all navigation buttons lead to the correct page, and back button inputs correctly load the previous context
  - Using emulators with different aspect-ratios, ensure the static layout scales appropriately so that no content is clipped or inaccessible
  - Using emulators with different aspect-ratios, ensure the dynamic elements that move or animate do not create clipped or inaccessible content
  - Using the firebase console, ensure that elements displaying account-specific data update when the database is changed
- Appearance Testing:
  - Get peer validation on the ease of use and attractiveness of various screens