

## System and Unit Test Report

Product: Melody Tracks

Team Name: Melody Tracks Team

Date: 7/23/20

- Sprint 1:
  - User Story #1: "As a user, I want a music player because it is essential to our program"
    - Scenario:
      1. Use music files other than .wav for input into bpm analyzer
      2. Compare the bpm value output by the analyzer to well known values for songs
      3. Values should match up regardless of file type
- Sprint 2:
  - User Story #5: "As a user, I want the App to know what BPM I'm jogging at, because I want to know."
    - Scenario:
      1. Set a large difference between the bpm value and the footstep cadence value of the user
      2. Select smart mode from selection screen
      3. User should hear successful time stretching even with the large difference between bpm and cadence
  - User Story #6: "As a user I want this audio thing to work correctly because it's giving me a headache."
    - Scenario: Try different genres of music to see if there are any noticeable levels of audio artifacting in relation to the timbre of the waveform.
- Sprint 3:
  - User Story 2: "As a developer, I need to know the BPM of the songs in order to manipulate the audio to match the pace of the user"
    - Scenario:
      1. The app starts and the BPM for the songs are calculated.
      2. The list of songs are displayed on the main screen.
  - User Story 3: "As a user, I want the app to change the BPM of a song to match my current pace because it provides motivation during exercise"
    - Scenario:
      1. The user starts the app.

2. The user puts the app in "Smart" mode.
  3. The user presses the "Start Run!" button.
  4. The user begins to run and the song's BPM changes to the user's pace.
- User Story 4: "As a user, I want to be able to choose whether to jog at a steady or dynamic pace"
    - Scenario:
      1. The user starts the app.
      2. The user puts the app in "Smart" mode or "Fixed" mode.
      3. The user presses the "Start Run!" button.
      4. The user begins to run and the song's BPM changes to the user's pace, if the user has selected "Smart" Mode. Otherwise, the app is in "Fixed" Mode.
  - User Story 5: "As a user, I want to navigate different songs I can play and see my jog process"
    - Scenario:
      1. The user starts the app.
      2. The user puts the app in "Smart" mode or "Fixed" mode.
      3. The user presses the "Start Run!" button.
      4. The user can pull up the Bottom View and skip forwards or backwards.
      5. The user can also see his progress on the map.
  - User Story 7: "As a developer, I need the different UI pages complete so that backend work can link to the UI"
    - Scenario:
      1. The developer starts the app.
      2. The developer progresses through the various screens and functionalities.
      3. The app does not crash.
- Unit Testing:
    - Daniel: Tested each screen to ensure that screen transitions do not crash the app. Each button on each screen was also tested.
    - Daniel: Tested the app after each manual merge to ensure that the merge was successful.

- John A.: Ran a CMPedometer instance in an isolated test app where values were simply printed to the screen to verify if they looked reasonable. Tested by simply walking around to ensure reasonable values were being displayed, then tried running and changing pace quickly to make sure it updates in a timely manner.
- John A.: Ran a CoreLocation instance with mapKit in an isolated test app to make sure user location is properly displayed, and user location history is reflected on the map.