Altering Heartbeat Using AI Selected Music
Test Plan

Team Members:
May Wandyez, Jeremy John, Ahmad Shah

Table of Contents

1.0 Introduction:

Welcome to the testing plan document. The testing plan document is intended for internal use for the team, as this project does not have an external client. The purpose of the testing plan document is to lay out the methodology for testing the application, this includes what needs to be done and what the general objectives are (test objectives), how it will be tested (test cases), and when it needs to be tested (testing schedule). In general, this project will consist of testing only the finished product, tested as a whole (system test).

2.0 Test Objectives:

The objectives of this test plan are to validate the functional requirements of the desktop application, to verify that the application as whole is responsive with minimal lag in performance, to ensure that the application is functional under circumstances previously not intended, and to ensure that the desktop application does not store protected health information.

2.1 Functional Requirement Testing Objectives:

Functional Requirement:	Title:	Objective:
FR 3.2.1	Retrieve Heart Rate from ANT+ Wireless Connection	● Confirm that a connection has been established between the ANT+ USB device and the ANT+ enabled wireless heart rate monitor ● Confirm that wireless data stream is received by desktop application. ● Confirm that wireless data stream is converted into a usable list of heart rate values.
FR 3.2.2	Establish Resting Heart Rate	● Confirm that user's resting heart rate is calculated from a continuous stream of the user's heart rate.
FR 3.2.3	User Selects Target Heart Rate	● Confirm that the user is capable of selecting their target heart rate ● Confirm that the user is limited in the input of their target heart rate to a range of between 40-150 beats per minute.
FR 3.2.4	Get Characteristics Of Music In Desktop Application Music Folder	● Confirm that a CSV file titled 'music_characteristics.csv' is generated with each row consisting of the following for each song present within the music directory: song length, average pitch, pitch of the first 30 seconds of the song, pitch of the last 30 seconds of the song, tempo of the song, tempo of the first 30 seconds of the song, tempo of the last 30 seconds of the song.

FR 3.2.5	User Selects Approach Path	● Confirm that the user is able to select from among the following approach paths using buttons available on the home page of the application: ”Shallow”, “Linear”, “Steep”, “Fastest” , “Parabola”, and “Rollercoaster”
FR 3.2.6	Select Music Feature	● Confirm that the predictive model generates predictions for every song present in music_characteristics.csv ● Confirm that the select music function receives the target heart rate selected by the user. ● Confirm that the select music function receives the approach path selected by the user ● Confirm that the select music function receives the current heart rate of the user. ● Confirm that the select music function receives the resting heart rate of the user. ● Confirm that a song is selected that has a prediction that gets the user from their current heart rate to the target heart rate.
FR 3.2.8	Navigation bar	● Confirm that the user can navigate between the user guide page and the home page using a navigation bar.
FR 3.2.9	User Guide Page	● Confirm that the user guide page includes step by step instructions of the purpose and use of every feature in the desktop application. ● Confirm that a video containing basic instructions on how to use the desktop application is present.
FR 3.2.10	Advanced Features Page	● Confirm that the advanced features are hidden until the user clicks a button to view them. ● Confirm that the user is able to hide the advanced features after showing them. ● Confirm that the advanced features includes an input box for resting heart rate. ● Confirm that the advanced features page includes
FR 3.2.11	Home Page	● Confirm that there is a home page present with the ability to select target heart rate ● Ensure that the user is able to select a start button to enable the user to begin selecting music to hit their target heart rate.

FR 3.2.12	Upload File	● Ensure that the user has the capability to upload files to the music folder via a button on the home page. ● Ensure that the user has the capability to upload files to the music folder via a drag and drop on the home page. ● Ensure that when a file is uploaded, it is present in the music folder. ● Ensure that when a file is uploaded, music_characteristics.csv updates.

2.2 Performance, Security, Reliability, Testing Objectives

Performance Requirement	Title	Objective:
NFR 3.3.1	Performance	Desktop application selects a song within 10 seconds. Desktop application loads within less than 5 seconds.
NFR 3.2.2	Safety	Users should not be able to select a resting heart rate or target heart rate exceeding 150 beats per minute, or below 40 beats per minute.
NFR 3.3.3	Availability	Desktop application should function in absence of internet connection
NFR 3.3.4	Security	Desktop application should save heart rate data.
NFR 3.3.5	Maintainability	The desktop application should be compatible with the most up to date version of web browsers.
NFR 3.3.6	Portability	The desktop application should be able to run on any computer running Windows 11

3.0 Test Strategy:

The primary test strategy will rely upon Black-Box testing. That is to say that the testers will not have to be aware of the inner workings of the desktop application, only its results. This strategy was decided upon as only the end result of the application matters, and if a process is not producing its desired result then it is likely that there is more than just one thing that needs changing.

Testing will be performed on a mixture of unit level and system level at the same time. The reasoning behind this unique approach is that some of the system level testing only performs one function within itself (such as inputting the target heart rate, or selecting the approach path). This means that the only real system-wide test that relies upon other features is the select music feature. The test planning has been structured so that testing of the select music feature will occur last, thus, ensuring that all the features that the select music function relies upon will be tested first.

4.0 Test Scope:

Testing scope will be limited only to those features already completed. Testing scope will not include testing of individual documents, or creation of 'test scripts'. This will ensure that testing is focused on the final product only.

5.0 Test Deliverables:

Testing reports are intended to be self-contained reports, which both include who performed the test, when, on what test case, the instructions for that test case, screenshots confirming the preconditions were met, screenshots of each step, a screenshot of the final result, an explanation of whether that final result met the pass/fail criteria, and a determination of whether the testing was passed or failed. The sections the user is intended to fill out are located in the white boxes. The testing report for each test case must be drafted as follows:

5.1 Test Report Template

Test Report	
Test ID:	**Ex. TC-1-1**
Attempt Number:	**Ex. 01**
Date:	**Ex. 7/8/2024**
Test Performed by:	**Ex. May Wandyez**
Pass/Fail	**Ex .Pass**
Test Case Description:	

ID:	**TC-1-1**
Item to Test:	UC-1, FR 3.2.3
Description:	Test of target heart rate default value feature.
Pre-Conditions:	No target heart rate has been entered by the user.
Test Steps:	1.) Open the application. 2.) Observe the ‘target heart rate’ input box.
Expected Results:	The value for target heart rate is set at the default value of 70.
Priority:	MEDIUM
Pass Criteria:	The value for the default target heart rate is visible, the value for the default target heart rate is 70.
Fail Criteria:	The value for the default target heart rate is not visible. The value for the default target heart rate is not 70.
Screenshot Confirming Preconditions met:	
Screenshots of Each Step:	
Screenshot of Final Result:	
Explain Why This Result Matches the Pass/Fail Criteria:	The pass result criteria requires the default heart rate to be visible, and the default heart rate is visible. The pass result criteria requires the default target heart rate to be 70, the displayed default target heart rate is 70.
Is this a Pass or Fail?	Ex.Pass.
If fail, please note the parameters that	

created the failure.	

The test plan for functional requirements is broken down by use case, with test plans dedicated to each feature necessary to perform each use case. The testing for non-functional requirements is broken down by non-functional requirement.

5.2 Functional Test Cases

5.2.1 UC-1: Target Heart Rate

ID:	**TC-1-1**
Item to Test:	Target Heart Rate Input
Description:	Test of target heart rate default value feature.
Pre-Conditions:	No target heart rate has been entered by user.
Test Steps:	1.) Open the application. 2.) Observe the ‘target heart rate’ input box.
Expected Results:	The value for target heart rate is set at the default value of 70.
Priority:	MEDIUM
Pass Criteria:	The value for the default target heart rate is visible, the value for the default target heart rate is 70.
Fail Criteria:	The value for the default target heart rate is not visible. The value for the default target heart rate is not 70.

ID:	**TC-1-2**
Item to Test:	Target Heart Rate Input

Description:	Test of target heart rate range limitation feature.
Pre-Conditions:	No target heart rate has been entered by the user.
Test steps:	1.) Open the application 2.) Attempt to input a value above 150 in the target heart rate box. 3.) Attempt to input a value below 40 in the target heart rate box.
Expected Results:	The user is unable to set the target heart rate to a value higher than 150 or lower than 140. When the user attempts to input a value higher than 150, the value will not increase above 150 and will remain at 150. If the user attempts to input a value below 40, the value will not decrease below 40 and the value will remain at 40.
Priority:	MEDIUM
Pass Criteria:	The value for the default target heart rate is visible, the value for the default target heart rate is 70.
Fail Criteria:	User is able to input a value above 150 or below 40.

5.2.2 UC-2: Resting Heart Rate Input

ID:	**TC-2-1**
Item to Test:	Resting Heart Rate Input
Description:	Test of resting heart rate default value feature.
Pre-Conditions:	The application is started, no resting heart rate has been entered by user.
Test steps:	1.) Open the application. 2.) Click on the ‘advanced features’ tab to reveal the resting heart rate input. 3.) Observe the current value in the resting heart rate input.
Expected Results:	The value within the resting heart rate input should be 70.
Priority:	MEDIUM
Pass Criteria:	The value for the default resting heart rate is visible, the value for the default resting heart rate is 70.
Fail Criteria:	The resting heart rate value is not visible, or is some number other

	than 70.

ID:	**TC-2-2**
Item to Test:	Resting Heart Rate Input
Description:	Test of resting heart rate range limitation feature
Pre-Conditions:	The application is started, no resting heart rate has been entered by user.
Test steps:	1.) Open the application. 2.) Click the 'advanced features' button to reveal the resting heart rate input. 3.) Attempt to input a value greater than 150. 4.) Attempt to input a value less than 40.
Expected Results:	It should be impossible to enter a value greater than 150, or less than 40 within the resting heart rate input box. If the user attempts to input a value greater than 150, the value will automatically adjust to 150. If the user attempts to input a value lower than 40, the value will automatically adjust to 40.
Priority:	MEDIUM
Pass Criteria:	User is unable to adjust resting heart rate above 150, or below 40.
Fail Criteria:	User is able to adjust the resting heart rate to above 150, or below 40.

ID:	**TC-2-3**
Item to Test:	Get Resting Heart Rate Button
Description:	Test of 'get resting heart rate' feature.
Pre-Conditions:	The application is started, no resting heart rate has been entered by user. The user is wearing their Garmin Vivosmart 4.
Test steps:	1.) Enable 'broadcast mode' on your Garmin Vivosmart 4. 2.) Start the application. 3.) Click the 'advanced features' button to reveal the resting heart rate input, and the 'get resting heart rate' button. 4.) Click the resting heart rate button.

Expected Results:	The default value within the resting heart rate input box is replaced with some other value.
Priority:	MEDIUM
Pass Criteria:	The resting heart rate in the resting heart rate input box is replaced with some other value than 70.
Fail Criteria:	The value within the resting heart rate input box does not change.

5.2.3 UC-3: Select Music

ID:	**TC-3-1**
Item to Test:	Start Button
Description:	Test of 'select music feature
Pre-Conditions:	The application is started. The user is wearing their Garmin Vivosmart 4.
Test steps:	1.) Enable 'broadcast mode' on your Garmin Vivosmart 4. 2.) Open the application. 3.) Set the target heart rate to 150. 4.) Click the 'start' button. 5.) Note which song begins playing. 6.) Close the application. 7.) Open the application. 8.) Set the target heart rate to 40. 9.) Click the 'start' button. 10.) Note which song begins playing.
Expected Results:	Different songs should appear on screen and begin playing automatically.
Priority:	HIGH
Pass Criteria:	The songs that are playing are different. The tempo of the first song (to raise your heart rate) should be high, the tempo of the second song that should be playing should be low.
Fail Criteria:	No song begins playing or the same song plays in both instances. Music widget fails to appear.

ID:	**TC-3-2**
Item to Test:	Machine Learning Model
Description:	Test of ‘select music feature’s underlying machine learning model for accuracy.
Pre-Conditions:	Ensure ‘HR.csv’ is present in the same folder as ‘FutureHeartBeatCompiler’
Test steps:	1.) Open FutureHeartBeatCompiler.py 2.) Click ‘run’
Expected Results:	The terminal log for FutureHeartBeatCompiler.py should note the accuracy of the model, this reported value should exceed 95%.
Priority:	HIGH
Pass Criteria:	Accuracy of model exceeds 95%
Fail Criteria:	Accuracy of model is below 95%

5.2.4 UC-4: Skip Music

ID:	**TC-4-1**
Item to Test:	Skip Music Button
Description:	Test of skip music feature
Pre-Conditions:	The application is started. The user is wearing their Garmin Vivosmart 4.
Test steps:	1.) Enable broadcast mode on your Garmin Vivosmart 4. 2.) Open the application. 3.) Click the ‘start’ button, a song should appear. 4.) Click the ‘skip’ button that appears alongside the music widget.
Expected Results:	A song should start playing, once the skip button is hit, the first song will stop playing, and a new song will start playing.
Priority:	HIGH
Pass Criteria:	A different song begins playing when the skip button is hit. Different

	song begins playing within 3 seconds of button click.
Fail Criteria:	The same song continues to play or skip button does not appear. Song does not begin playing within 3 seconds of button click.

5.2.5 UC-5: Select Approach Path

ID:	**TC-5-1**
Item to Test:	Approach Path Buttons
Description:	Test of select approach path feature.
Pre-Conditions:	The application is started. The user is wearing their Garmin Vivosmart 4.
Test steps:	1.) Enable the broadcast mode on your Garmin Vivosmart 4. 2.) Open the application. 3.) Select the 'advanced features' button on the home page of the application. 4.) Click the 'rollercoaster' approach path. 5.) Click the 'get resting heart rate' button to set the resting heart rate input to your current resting heart rate. 6.) Set the target heart rate to your current resting heart rate. 7.) Click the start button. 8.) Once a song appears, click the skip button. 9.) Repeat clicking the skip button, a song of lower tempo than your current resting heart rate should eventually be chosen.
Expected Results:	A song of lower resting heart rate should be chosen.
Priority:	LOW
Pass Criteria:	A song of lower tempo than your current resting heart rate should be chosen.
Fail Criteria:	No song plays at all.

5.2.6 UC-6: User Guide

ID:	**TC-6-1**

Item to Test:	User Guide
Description:	Test of user guide.
Pre-Conditions:	None.
Test steps:	1.) Open the application. 2.) Click the ‘User Guide’ button in the navigation bar at the top of the screen. 3.) View the user guide, click the video containing the user guide explaining to the user the features of the application. 4.) Click the home page button in the navigation bar.
Expected Results:	The user guide appears when the user guide button is clicked. The video appears on the user guide page, the video loads and plays when started by the user.
Priority:	MEDIUM
Pass Criteria:	The user guide page appears when the user guide button is clicked, the video appears on the user guide page and is capable of being played, the home page appears again when clicked. Each page loads within 3 seconds.
Fail Criteria:	User cannot access user guide page, user is unable to return to home page, the video fails to load. Pages take longer than 3 seconds to load.

5.2.7 UC-7: Adding Music

ID:	**TC-7-1**
Item to Test:	Add Music Button.
Description:	Test of adding music to the application
Pre-Conditions:	Have 2 MP3 files over 1 minute in length available.
Test steps:	1.) Delete all contents of the ‘music’ folder where the application is stored. 2.) Delete ‘music_characteristics.csv’ in the folder where the application is stored. 3.) Open the application 4.) Click on the ‘add file’ button on the home page. 5.) Select the first mp3 file when prompted. 6.) Success message should appear indicating that the file has been

	added. 7.) Click and drag the second mp3 file over the add file button. 8.) Success message should appear indicating the file has been added. 9.) Open the music folder, the two music files you added should both be present. 10.) Open music_characteristics.csv
Expected Results:	Success message appears when file added via drag and drop, and when added via manual file selection. Added files are present in the music folder. ‘music_characteristcs.csv’ successfully generated, with one row dedicated to each song added.
Priority:	MEDIUM
Pass Criteria:	File successfully added message appears after drag and drop, success message appears after manual file selection. New music_characterstics.csv generated.
Fail Criteria:	No file successfully added message appears. No new music_characteristics.csv generated, no values of added files present inside music_characteristics.csv. Added files are not present in music folder.

5.2.8 UC-8: Advanced Features

ID:	**TC-8-1**
Item to Test:	Advanced Features Button
Description:	Test of advanced features button.
Pre-Conditions:	None.
Test steps:	1.) Open the application. 2.) Click the ‘advanced features’ button. 3.) The approach path buttons and resting heart rate buttons should appear.
Expected Results:	Approach path buttons and resting heart rate buttons should appear.
Priority:	HIGH

Pass Criteria:	Approach path buttons and resting heart rate input appear.
Fail Criteria:	Approach path buttons and resting heart rate input do not appear.

5.3 Non-Functional Testing.

ID:	**TC-NF-1-1**
Item to Test:	Performance Requirements
Description:	Test of performance requirements by determining the time it takes for the start button to load music.
Pre-Conditions:	None.
Test steps:	1.) Set your Garmin Vivosmart 4 to broadcast mode. 2.) Open the desktop application. 3.) Set a timer, be prepared to click it at the same time as the start button. 4.) Click the start button. 5.) When the music loads, measure the time it took for the song to load.
Expected Results:	The music should begin in less than 5 seconds after the start button is selected.
Priority:	HIGH
Pass Criteria:	The music begins playing is less than 5 seconds after the star button is selected
Fail Criteria:	The music begins playing after longer than 5 seconds.

ID:	**TC-NF-1-2**
Item to Test:	Performance Requirements
Description:	Test of performance requirements for time to boot up the desktop application.
Pre-Conditions:	None.
Test steps:	1.) Set a timer, be prepared to start it when you start the desktop application.

	2.) Start the desktop application 3.) When the desktop application screen loads, stop the timer.
Expected Results:	The desktop application should load in less than 5 seconds.
Priority:	HIGH
Pass Criteria:	The desktop application loads in less than 5 seconds.
Fail Criteria:	The desktop application takes greater than 5 seconds to load.

ID:	**TC-NF-2-1**
Item to Test:	Target Heart Rate Input
Description:	Test of target heart rate range limitation feature for safety.
Pre-Conditions:	The application is started, no target heart rate has been entered by user.
Test steps:	1.) Open the application 2.) Attempt to input a value above 150 in the target heart rate box. 3.) Attempt to input a value below 40 in the target heart rate box.
Expected Results:	The user is unable to set the target heart rate to a value higher than 150, or lower than 140. When the user attempts to input a value higher than 150 the value will not increase above 150, and will remain at 150. If the user attempts to input a value below 40, the value will not decrease below 40, and the value will remain at 40.
Priority:	MEDIUM
Pass Criteria:	The value for the default target heart rate is visible, the value for the default target heart rate is 70.
Fail Criteria:	User is able to input a value above 150 or below 40.

ID:	**TC-NF-2-2**
Item to Test:	Resting Heart Rate Input
Description:	Test of resting heart rate range limitation feature for safety
Pre-Conditions:	The application is started, no resting heart rate has been entered by

	user.
Test steps:	1.) Open the application. 2.) Click the ‘advanced features’ button to reveal the resting heart rate input. 3.) Attempt to input a value greater than 150. 4.) Attempt to input a value less than 40.
Expected Results:	It should be impossible to enter a value greater than 150, or less than 40 within the resting heart rate input box. If the user attempts to input a value greater than 150, the value will automatically adjust to 150. If the user attempts to input a value lower than 40, the value will automatically adjust to 40.
Priority:	MEDIUM
Pass Criteria:	User is unable to adjust resting heart rate above 150, or below 40.
Fail Criteria:	User is able to adjust the resting heart rate to above 150, or below 40.

ID:	**TC-NF-3**
Item to Test:	Availability Requirements
Description:	Test of availability without internet.
Pre-Conditions:	None.
Test steps:	1.) Disconnect your computer from wifi. 2.) Set the Garmin Vivosmart 4 to broadcast mode. 3.) Open the desktop application. 4.) Click the start button.
Expected Results:	A music file should load and begin playing.
Priority:	LOW
Pass Criteria:	Music file is still selected without internet connection.
Fail Criteria:	Music file is not selected without internet connection. .

ID:	**TC-NF-4**
Item to Test:	Security Requirements
Description:	Testing of security, which is to ensure that no heart rate data is measured from user use of the application.
Pre-Conditions:	None.
Test steps:	1.) Open the application. 2.) Click the ‘advanced features’ button. 3.) The approach path buttons and resting heart rate buttons should appear.
Expected Results:	There should be no heart rate data permanently stored within the music characteristics.csv file
Priority:	HIGH
Pass Criteria:	No heart rate data is saved.
Fail Criteria:	Heart rate data is saved.

ID:	**TC-NF-5**
Item to Test:	Maintainability Requirement
Description:	Testing of maintainability requirement, ensuring that the desktop application is compatible with most up to date software.
Pre-Conditions:	None.
Test steps:	1.) Download the latest version of google Chrome. 2.) Set google chrome as your default browser. 3.) Set your Garmin Vivosmart 4 to broadcast mode. 4.) Start the desktop application, the HTML file should be loaded in google chrome. 5.) Select the start button.
Expected Results:	A music file should be selected and begin playing.
Priority:	LOW

Pass Criteria:	Desktop application loads and core features of playing music is operational
Fail Criteria:	Desktop application does not load.

ID:	**TC-NF-6**
Item to Test:	Portability Requirement.
Description:	Testing of portability feature
Pre-Conditions:	Possession of computer running Windows 11 operating system.
Test steps:	1.) Set your Garmin Vivosmart 4 to broadcast mode 2.) Open the desktop application 3.) Select the start button.
Expected Results:	A music file should be selected and begin playing.
Priority:	LOW
Pass Criteria:	Desktop application loads and core features of playing music is operational.
Fail Criteria:	Desktop application does not load.

5.4 Traceability Matrix.

Functional Requirement:	**Use Case:**	**Test Case:**
FR 3.2.1: Retrieve Heart Rate from ANT+ Wireless Connection	UC-3: Select Music	TC-3-1
		TC-3-2
FR 3.2.2: Establish Resting Heart Rate	UC-2: Resting Heart Rate Input	TC-2-1
		TC-2-2
		TC-2-3
FR 3.2.3: User Selects Target	UC-1: Target Heart Rate Input	TC-1-1

Heart Rate		TC-1-2
FR 3.2.4: Get Characteristics of Music In Desktop Application Music Folder	UC-7: Adding Music	TC-7-1
FR 3.2.5: USer Selects Approach Path	UC-5: Select Approach path	TC-5-1
FR 3.2.6: Select Music Feature	UC-3: Select Music	TC-3-1
		TC-3-2
	UC-4: Skip music	TC-4-1
FR 3.2.8: Navigation Bar	UC-6: User Guide	TC-6-1
FR 3.2.9: User Guide Page		
FR 3.2.10: Advanced Features Page	UC-8: Advanced Features	TC-8-1
FR 3.2.11: Home Page	UC-1: Target Heart Rate Input	TC-1-1
		TC-1-2
	UC-2: Resting Heart Rate Input	TC-2-1
		TC-2-2
		TC-2-3
	UC-3:Select Music	TC-3-1
		TC-3-2
	UC-4 Skip Music	TC-4-1
	UC-5: Select Approach path	TC-5-1
	UC-7: Adding Music	TC-7-1
	UC-8: Advanced Features	TC-8-1

FR 3.2.12: Upload File	UC-7: Adding Music	TC-7-1

6.0 Test Schedule:

Deadline:	Tests To Complete:
7/12/2024	TC-1-1, TC-1-2, TC-8-1, TC-2-1, TC-2-2, TC-6-1
7/15/2024	TC-2-3 TC-7-1,
7/17/2024	TC-3-1, TC-3-2
7/19/2024	TC-4-1, TC-5-1, TC-NF-1-1, TC NF -1-2, TC-NF-2-1, TC NF-2-2, TC-NF-3, TC-NF-4, TC-NF5, TC-NF-6

7.0 Test Environment:

The test environment consists of the following hardware, the following dependencies, and the following operating system.

Testing Environment		
Technology:	Type:	Purpose/Installation Guide:
Garmin Vivosmart 4	Hardware	The Garmin Vivosmart 4 is the ANT+ compatible heart rate monitor. The current heart rate of the user is necessary in order to predict the future heart rate of the user. Your Garmin Vivosmart 4 will need to be in broadcast mode in order for its data to be received, instructions on how to set the Garmin Vivosmart 4 to broadcast mode can be found in the following link: https://www8.garmin.com/manuals/webhelp/vivosmart4/EN-US/GUID-3629BD02-A0FB-4F18-B1A4-600C2643C835.html
Garmin USB ANT Stick	Hardware	The Garmin USB ANT stick acts as a radio receiver for ANT+ wireless communications. This USB stick is necessary to receive the heart rate from the Garmin Vivosmart 4. Plug in the ANT+ USB stick to your computer.

		You will have to install the drivers manually, the driver are located in the 'drivers' folder that came with the download for the project. Here is a guide on how to manually locate and install drivers: https://support.microsoft.com/en-us/windows/update-drivers-manually-in-windows-ec62f46c-ff14-c91d-eead-d7126dc1f7b6
Windows 11	Operating System	Windows 11 operating system has been identified as compatible with ANT+ communications drivers which are necessary for communicating between the application and the heart rate monitor. The testing environment must be conducted on a Windows 11 computer.
python 3.12	Dependency	python serves the general backend for this project. The testing environment must have this version of python installed. Python can be downloaded from this link: https://wiki.python.org/moin/BeginnersGuide/Download
openANT 1.3.1	Dependency	openANT is a python implementation of ANT+ wireless communications protocol - allows python programs to communicate with ANT+ heart rate monitor devices. openANT must be installed in the testing environment. openANT can be downloaded from this link: https://github.com/Tigge/openant/tree/master/openant Alternatively, it can be installed with the following command: pip install openant
pyusb 1.2.1	Dependency	pyUSB is a requirement for openANT and allows python to communicate to USB devices. Pyusb can be downloaded from the following link: https://pypi.org/project/pyusb/ Pyusb can also be installed with the following command: pip install pyusb
librosa 0.10	Dependency	librosa is a python package for audio analysis. librosa allows audio files to be broken down to their base characteristics (tempo, pitch, length). It is necessary to break audio files down to their base characteristics in order for the machine learning model to be able to identify how a music file will impact heart rate. Librosa can be installed from the following link:

		https://librosa.org/doc/latest/install.html Librosa can also be installed using the following command: pip install librosa
PyTorch 2.3	Dependency	PyTorch is a python library for construction of machine learning models. PyTorch will be used to create a machine learning model that will predict heart rate based on music characteristics and current user heart rate. Pytorch can be installed from the following link: https://pytorch.org/ Pytorch can also be installed using the following command: pip install torch
Flask 3.0.3	Dependency	Flask allows communication between a HTML front end and a python based backend. Flask can be downloaded from the following link: https://flask.palletsprojects.com/en/3.0.x/installation/ Flask can also be installed using the following command: pip install Flask
Flask-Bootstrap 3.3.7.0	Dependency	Allows integration of the popular 'bootstrap' framework into Flask. This allows the frontend to be more responsive and integrate javascript features. Flask-Bootstrap can be downloaded from the following link: https://bootstrap-flask.readthedocs.io/en/stable/basic/#installation Flask-Bootstrap can also be installed using the following command: pip install bootstrap-flask

8.0 Test Entry and Exit Criteria:

The testing plan will begin execution once all features are completed for a 'test-ready' state. This means that every feature has been tested individually, and as a unit by at least one team member, and is now ready for testing by all team members. Testing will exit once test reports are completed for all functional and non-functional requirements, and two-thirds of all team members agree that each test report shows a pass.

9.0 Test Pass and Fail Criteria:

For the system to pass the testing plan, each test report must be reviewed by two-thirds of all team members who must affirm that it passes testing - if this quorum is not reached then testing is regarded as a failure.

10.0 Test Suspension and Resumption Criteria:

Testing will be suspended if an error occurs that requires code alterations that extend beyond the feature being tested. This is necessary as any alterations that extend beyond a particular feature could cascade into new errors in previously functional features. Testing will resume once a fix has been implemented and pushed to the shared repository.

11.0 Test Design and Execution:

All testing will be performed manually. This is because most of the testing involved is in regards to human-computer interaction - measuring heart rates, determining preferences etc.

Test execution will involve the user starting with a copy of the report template. The user will then follow the test case step by step, taking screenshots along the way for each step to note how they completed their steps. The user will include a screenshot of the end result, comparing it to the expected result, and rating the test as a failure or success based on the result. This design allows users to replicate the exact process that led to the failure or success by following the screenshots for each step.

If a report indicates a failure, a team member will be assigned to recreate the failure on their end. If the defect can be replicated a team member will be assigned by the team lead to fix what caused the failure. The overall structure of the test plan is such that tests that rely upon other features are scheduled to occur after the tests that occur first.

12.0 Test Data & Defect Management:

Testers are instructed to provide their own mp3 files as necessary to test the desktop application. This is necessary as much music that we think of as ‘catchy’ enough to cause the user to synchronize their heartbeat is protected by copyright. Copyright free music sufficient to test the features of the project is provided in the central repository.

Defect management will be notified by having the team member involved alert all other team members to the defects. The team member must note the situation where the defect occurred, and how to replicate it. After the defect is replicated by another team member, a team member will be assigned by the team leader to resolve the defect.

13.0 Risk Analysis:

There is a remote risk that during testing that a user may alter their heart rate to a dangerously high level. The testing plan has been meticulously constructed so that testing of safeguards for limitation on heart rate change are tested first, and that testing those safeguards does not require the tester's heart rate to be altered. Even so the concern remains that sustained heart rates near the acceptable bounds could cause strain on the heart similar to how a high-stress job puts strain on the heart.

14.0 Roles and Responsibilities:

Team Member:	**Assigned Tests:**	**Additional Responsibilities:**
May Wandyez	TC3-2, TC-4-1, TC-5-1, TC-NF-4, TC-NF-5.	Responsible for creating test-case template for standardization of results, ensuring other team members turn in test cases on time.
Jeremy John	TC-1-1,TC-1-2, TC-2-1, TC-2-2, TC-6-1, TC-8-1, TC-NF-2-1, TC-NF-2-2.	Responsible for front end performance and remediation of performance issues. Responsible for creating and ensuring that there is a clear on screen response for every interaction with the front end.
Ahmad Shah	TC-2-3, TC-3-1, TC-7-1, TC-NF-1-1, TC-NF-1-2, TC-NF-3, TC-NF-6.	Responsible for ensuring that data collected from the heart rate monitor is properly received and saved.