# Altering Heartbeat Using AI Selected Music
# Design Specification Document

Team Members:

May Wandyez, Jeremy John, Ahmad Shah

# Table of Contents

# 1. Introduction

## 1.1 Purpose of the Design Specification Document

This document is intended for the development team and project manager. Its purpose is to define the details of the architecture and system design for the desktop application for Altering Heartbeat Using AI Selected Music such that the development team could construct the envisioned product.

# 2. General Overview and Design Guidelines

## 2.1 Assumptions/Constraints/Standards

It is assumed that the desktop application will be functional on any Windows 11 machine.

There is a constraint of 10,000 songs within the music directory, that the only songs that may be added are mp3 files, and that the desktop application is only compatible with ANT+ heart rate monitoring devices.

# 3. Architecture Design

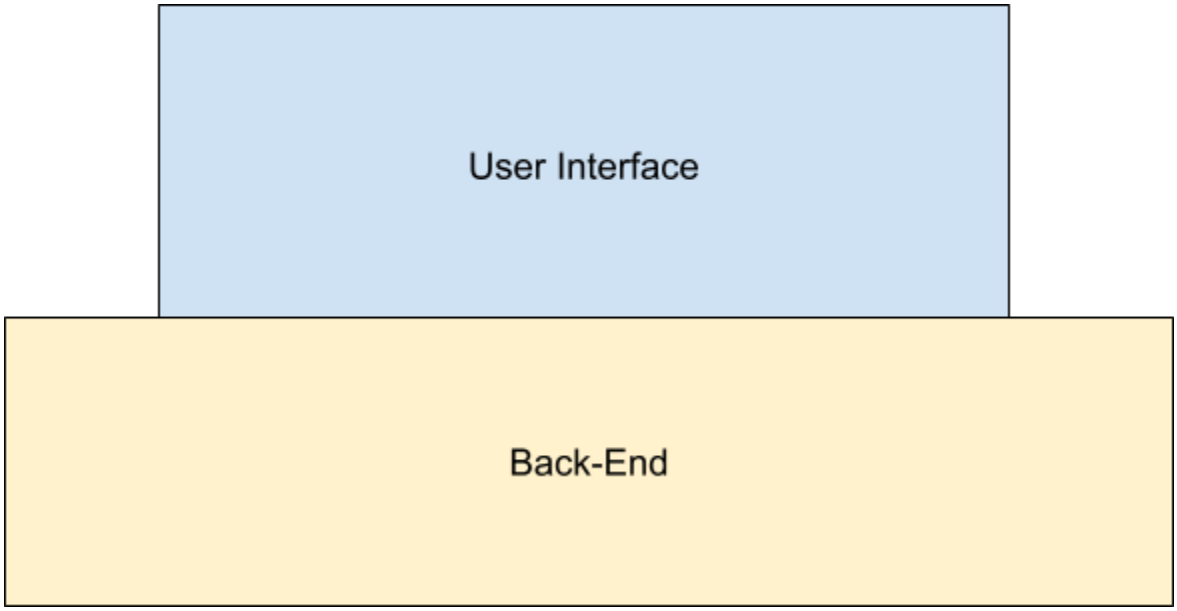
## 3.1 Hardware Architecture

The hardware consists of three main components. The Garmin Vivosmart 4 heart rate monitor, the Garmin ANT+ USB Stick, and the Windows 11 Desktop. The Garmin Vivosmart 4 captures the user's current heart rate, and broadcasts it using ANT+ wireless communications protocol. The Garmin ANT+ USB stick receives the ANT+ broadcast from the heart rate monitor and allows the Windows 11 Desktop to use it for the desktop application. The Windows 11 Desktop runs the desktop application, and requires the user's current heart rate. The Windows 11 Desktop also allows the user to interact with the desktop application's front end.


| Hardware: | Connection Type: | Connects To: | Purpose: |
|---|---|---|---|
| Windows 11 Desktop | Not Applicable. | Garmin USB ANT+ Stick | Runs the desktop application. |
| Garmin USB ANT+ Stick | USB Type A Port. | Windows 11 desktop computer, Garmin Vivosmart 4. | Receives ANT+ wireless communications containing heart rate information from wearer, allowing Windows 11 Desktop to access this information. |
| Garmin Vivosmart 4 | Wireless. | Garmin USB ANT+ Stick | Records user heart rate, and broadcasts it via ANT+ wireless communications. |

## 3.2 Software Architecture

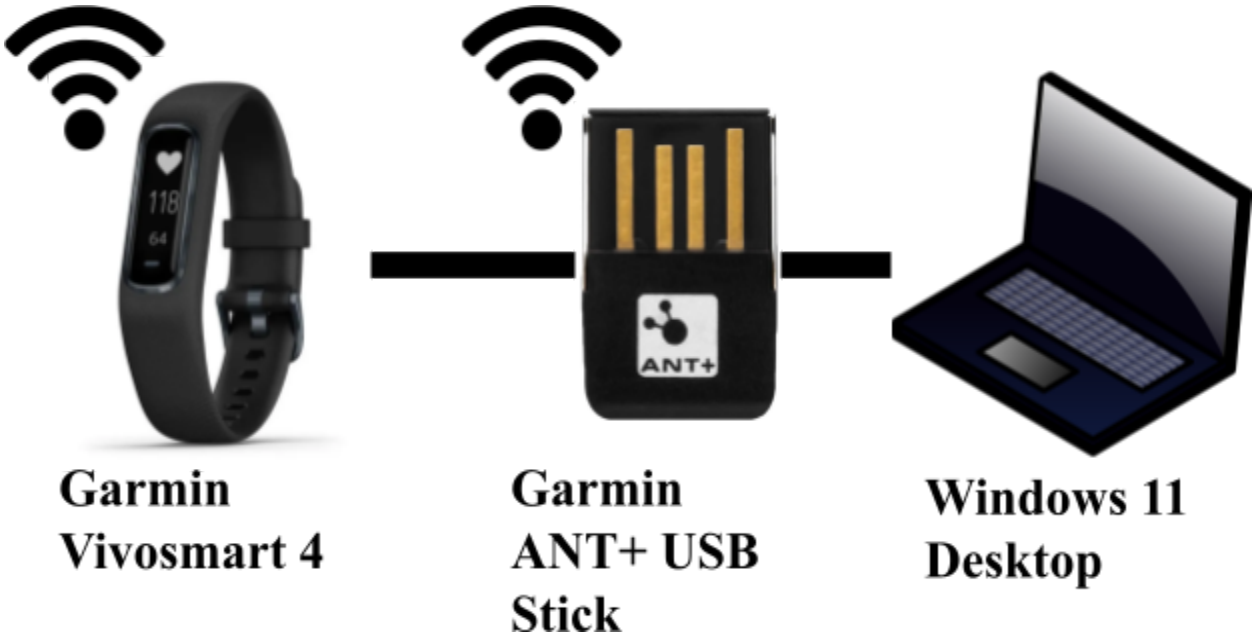
The software architecture of the desktop application is monolithic architecture. A single script handles both front and back end operations, with the back end operations being based on multithreading for concurrent operations. This means that the application can be envisioned as a User Interface connected to a back-end.




## 3.3 Security Architecture


There is no necessary security architecture for the design of the desktop application. This is due to the fact that no sensitive data is saved on the desktop application. Security is further unnecessary because the desktop application does not connect to the internet. In addition, it is not possible to encrypt wireless communications as ANT+ wireless communications used by Garmin Heart Rate Monitors are deliberately unencrypted to ensure interoperability of ANT+ wireless communication devices.

## 3.4 Communication Architecture




The Garmin Vivosmart 4 connects to the Garmin ANT+ USB stick using Garmin's Adaptive Network Topology ("ANT") wireless communications protocol. The ANT+ USB Stick's direct USB A connection into a laptop allows the laptop to connect to the Garmin Vivosmart 4 Heart Rate Monitor via the Garmin ANT+ USB Stick.

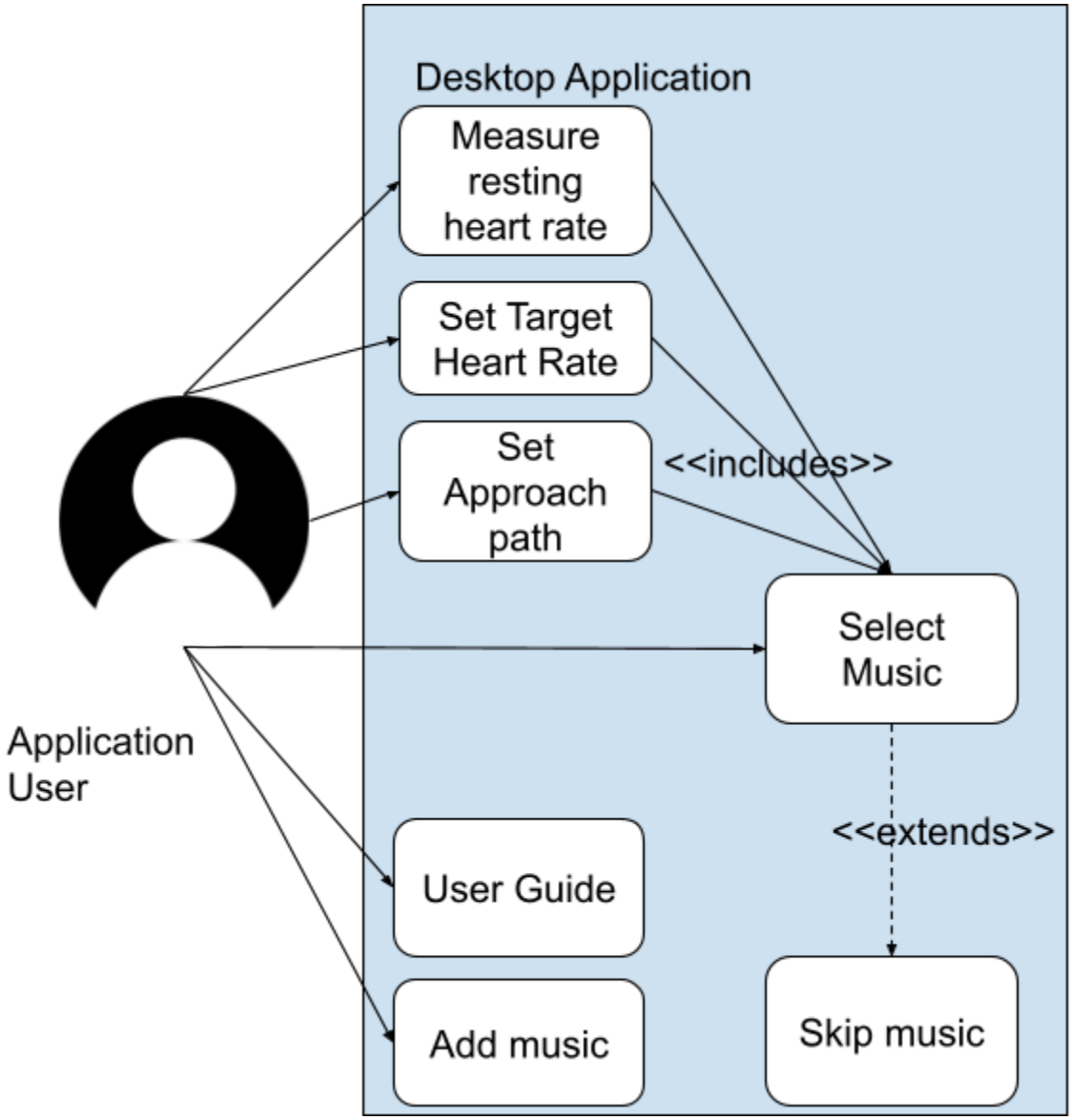
## 3.5 Performance

The performance architecture for the desktop application consists of the desktop application itself, and an independent thread that continuously reads the user's heart rate.

# 4. System Design

## 4.1 Use Cases

Below is the use case diagram, which illustrates how the application use interacts with the use cases of the system.




| | |
|---|---|
| **ID:** | UC-1 |
| **Use Case Name:** | Target Heart Rate Input |
| **Description:** | This use case will describe an application user who is selecting the target heart rate that they wish to achieve through listening to music selected by a machine learning model. To minimize the risk of the user choosing a target heart rate that is unsafe or the user inputting a wrong value, the input will only be able to be received by tapping a button for increase and one button for decrease for the target heart rate. |
| **Actors:** | Application User |

| | |
|---|---|
| **Trigger:** | This is triggered by the user selecting their target heart rate within the home page of the application. |
| **Preconditions:** | 1.) The desktop application is open. |
| **Postconditions:** | 1.) The selected target heart rate is saved as a local variable within webapp.py<br>2.) The selected target heart rate is displayed on screen on the home page. |
| **Normal Flow:** | 1.) User enters the application.<br>2.) The user selects a new target heart rate. |
| **Alternative Flow:** | 1.) The user enters the application<br>2.) The user is satisfied with the default target heart rate, and does not change it. |
| **Exceptions:** | 1.) Heart Rate Out of Bounds Exception - The user attempts to select a target heart rate that is above 140 beats per minute or below 40 beats per minute, when this exception is triggered the target heart rate will automatically round to maximum or minimum of 140 beats per minute or 40 beats per minute, depending on which is closer to the user's attempted selection. |
| **Assumptions:** | None. |

| | |
|---|---|
| **ID:** | UC-2 |
| **Use Case Name:** | Resting Heart Rate Input |
| **Description:** | This use case describes how the Application User's resting heart rate is measured by the desktop application. |
| **Actors:** | Application User |
| **Trigger:** | This is triggered by the user opening the application, or the user selecting the 'select resting heart rate' under advanced features. |
| **Preconditions:** | 1.) The desktop application is open. |
| **Postconditions:** | 2.) The selected target heart rate is saved as a local variable within webapp.py<br>3.) The selected target heart rate is displayed on screen on the home page. |

| | |
|---|---|
| **Normal Flow:** | 1.) Application User enters the application.<br>2.) Application attempts to connect to Application User's Garmin Vivosmart 4 to read heart rate.<br>3.) A popup window prompts the user to stay still, relax, and take deep breaths for the purpose of holding their heart rate steady..<br>4.) Once the user's heart rate stabilizes, the user's resting heart rate is displayed on screen. The popup window closes. |
| **Alternative Flows:** | 1.) User clicks on the 'advanced settings' button on the home page.<br>2.) User clicks on the 'Set Resting Heart Rate' button.<br>3.) User manually types in their resting heart rate. |
| **Exceptions:** | 1.) Connection Timeout Exception - The desktop application has failed to connect to the user's heart rate monitor for a total of one minute.<br>2.) Unstable Heart Rate Timeout Exception - The user has failed to stabilize their heart rate within a one minute period. |
| **Assumptions:** | The user is wearing their Garmin Vivosmart 4 ANT+ wireless heart rate monitor and has set it to broadcast mode. |

| | |
|---|---|
| **ID:** | UC-3 |
| **Use Case Name:** | Select Music |
| **Description:** | This use case describes what happens when the user starts the application by selecting the 'start' button. |
| **Actors:** | Application User |
| **Trigger:** | This is triggered by the user opening the application, or the user selecting the 'select resting heart rate' under advanced features. |
| **Preconditions:** | 1.) The user has selected a valid target heart rate.<br>2.) The user has retrieved a valid resting heart rate. |
| **Postconditions:** | 1.) The selected song is displayed.<br>2.) The selected song begins playing in a music widget.<br>3.) A skip button appears next to the music widget. |
| **Normal Flow:** | 1.) The user selects the large 'start' button on the home page.<br>2.) Current user heart rate is recorded from ANT+ enabled wireless heart rate monitor connection.<br>3.) The desktop application selects a song by generating |

| | |
|---|---|
| | predictions for change in heart rate for each song within the music directory of the desktop application using a machine learning model. A choice from among these predictions is selected using the 'approach path'.<br>4.) The desktop application displays the selected song, and begins automatically playing it within a music widget embedded in the HTML which enables pausing and playing of the selected song. |
| **Alternative Flows:** | |
| **Exceptions:** | 1.) No Valid Music Exception - There is no valid music within the music directory that is predicted to achieve a change in the user's heart rate towards their target heart rate.<br>2.) No Resting Heart Rate Exception - The user has not determined a value for their resting heart rate.<br>3.) Connection Timeout Exception - The desktop application is unable to connect to the user's ANT+ enabled wireless heart rate monitor for a period of 1 minute or more. |
| **Assumptions:** | The user is wearing their Garmin Vivosmart 4 ANT+ wireless heart rate monitor and has set it to broadcast mode. |

| | |
|---|---|
| **ID:** | UC-4 |
| **Use Case Name:** | Skip Music |
| **Description:** | This use case describes when a user decides to skip the currently playing song in favor of another song. |
| **Actors:** | Application User |
| **Trigger:** | This is triggered by the user hitting the 'skip' button. |
| **Preconditions:** | 1.) The user has selected a valid target heart rate.<br>2.) The user has retrieved a valid resting heart rate.<br>3.) A song has already been selected.<br>4.) The music widget has appeared for the user. |
| **Postconditions:** | 1.) The newly selected song is displayed within a music widget that automatically begins playing. |
| **Normal Flow:** | 1.) The user selects the 'skip' button on the home page next to the music playing widget. |

| | |
|---|---|
| | 2.) The currently selected song is no longer displayed. .<br>3.) The desktop application selects a song by generating predictions for change in heart rate for each song within the music directory of the desktop application.<br>4.) The desktop application displays the selected song, and begins automatically playing it. . |
| **Alternative Flows:** | |
| **Exceptions:** | 1.) No Valid Music Exception - There is no valid music within the music directory that is predicted to achieve a change in the user's heart rate towards their target heart rate.<br>2.) No Resting Heart Rate Exception - The user has not determined a value for their resting heart rate.<br>3.) Connection Timeout Exception - The desktop application is unable to connect to the user's ANT+ enabled wireless heart rate monitor for a period of 1 minute or more. |
| **Assumptions:** | The user is wearing their Garmin Vivosmart 4 ANT+ wireless heart rate monitor and has set it to broadcast mode. |

| | |
|---|---|
| **ID:** | UC-5 |
| **Use Case Name:** | Select Approach Path |
| **Description:** | This use case describes what happens when the user delves into the advanced settings to change the way that music is selected. |
| **Actors:** | Application User |
| **Trigger:** | The application user selects an approach path from the approach path dropdown menu. |
| **Preconditions:** | None. |
| **Postconditions:** | 1.) The selected approach path is displayed. |
| **Normal Flow:** | 1.) The user clicks the 'advanced features' button on the home page.<br>2.) The user selects the Approach Path from the dropdown options.<br>3.) The selected approach path is displayed. |
| **Alternative** | None. |

| | |
|---|---|
| **Flows:** | |
| **Exceptions:** | None. |
| **Assumptions:** | None. |

| | |
|---|---|
| **ID:** | UC-6 |
| **Use Case Name:** | User Guide |
| **Description:** | This use case describes what happens when the user is present on the home page, and wishes to open the friendly user guide to understand how the desktop application works. |
| **Actors:** | Application User |
| **Trigger:** | The application selects the ‘User Guide’ button in the navigation bar. |
| **Preconditions:** | 1.) The user is present on the home page. |
| **Postconditions:** | 1.) The User Guide page is shown. |
| **Normal Flow:** | 1.) User clicks on the ‘User Guide’ button in the navigation bar’<br>2.) The ‘User Guide Page’ is displayed.<br>3.) User reads the ‘User Guide’ page and becomes informed on how to use the desktop application.<br>4.) User clicks on the ‘Home’ button in the navigation bar.<br>5.) The ‘Home Page’ is displayed. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Assumptions:** | None. |

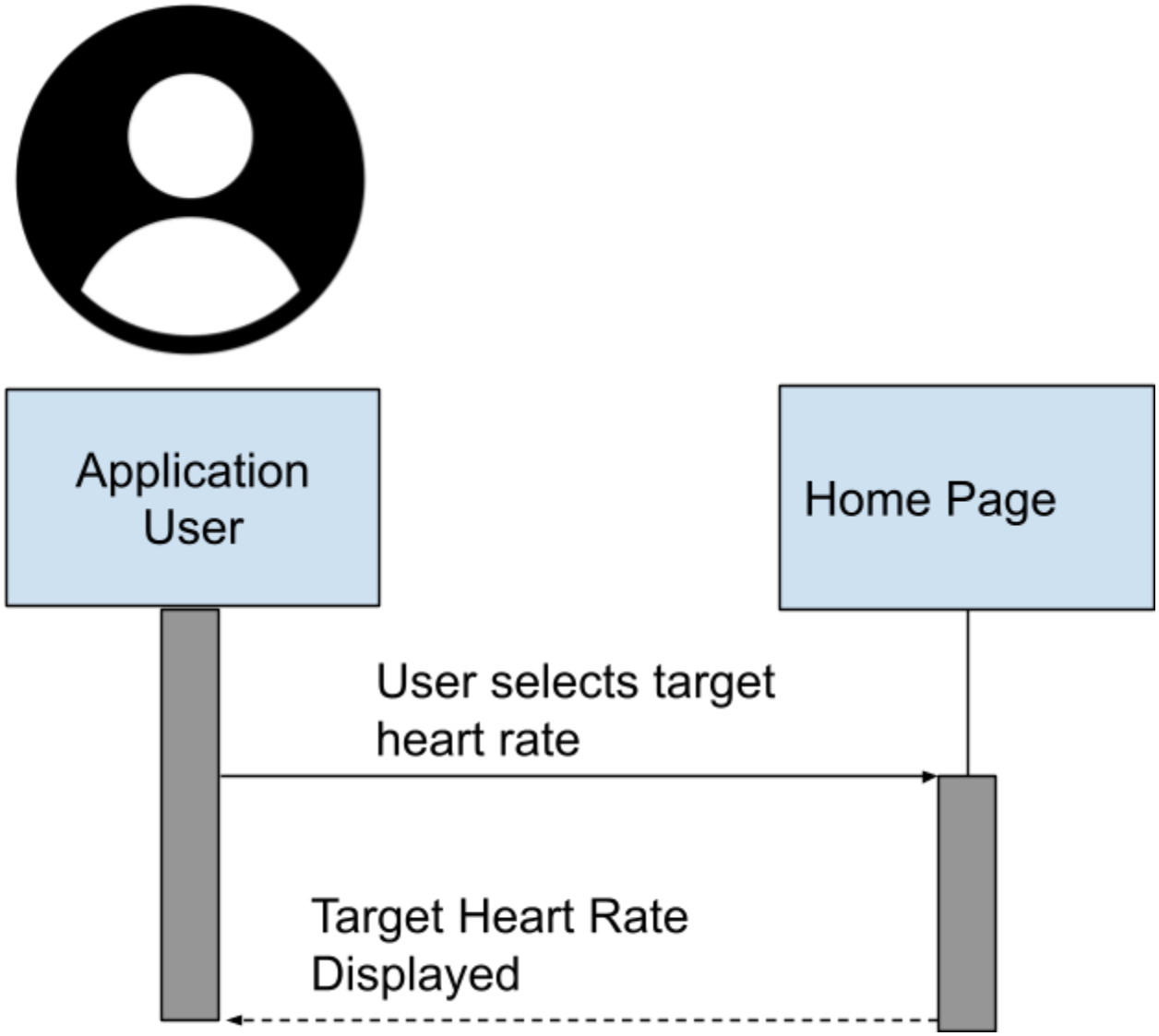
| | |
|---|---|
| **ID:** | UC-7 |
| **Use Case Name:** | Adding Music |

| | |
|---|---|
| **Description:** | This use case describes how the Application User would go about adding more music to the desktop application. |
| **Actors:** | Application User |
| **Trigger:** | The application user clicks and drags a file into the home page. |
| **Preconditions:** | 1.) The user is present on the home page. |
| **Postconditions:** | 1.) The selected file is added to the music directory<br>2.) The selected music file characteristics as well as the new filepath are added to the CSV. |
| **Normal Flow:** | 1.)User clicks and drags a .mp3 file from elsewhere on the computer and hovers it over the home page of the desktop application.<br>2.) The file is moved to the music directory of the desktop application.<br>3.) The file characteristics for the file are added to the music characteristics CSV. |
| **Alternative Flows:** | None. |
| **Exceptions:** | Wrong File Type Exception - Any other file than a .mp3 file is not compatible. |
| **Assumptions:** | None. |

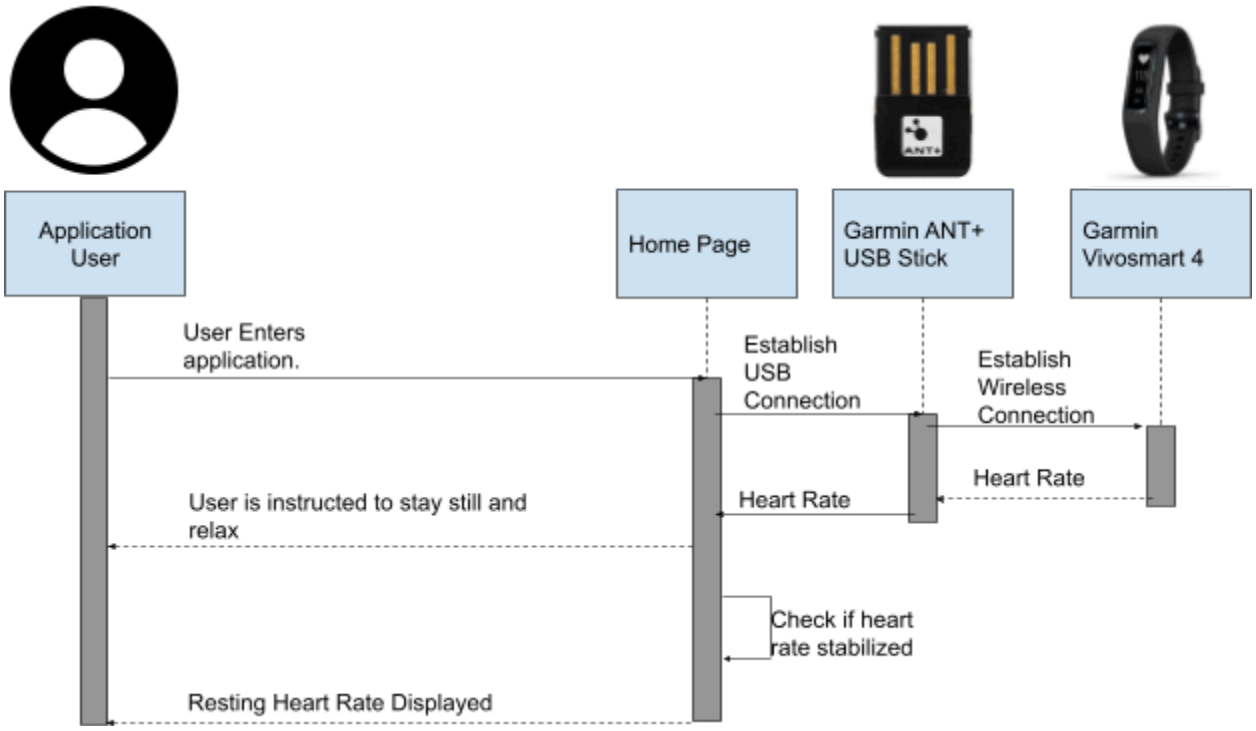
| | |
|---|---|
| **ID:** | UC-8 |
| **Use Case Name:** | Advanced Features |
| **Description:** | This use case describes how the user can click a button to show or hide the advanced features. |
| **Actors:** | Application User |
| **Trigger:** | The application user selects the ‘Advanced Features’ button. |
| **Preconditions:** | 1.) The user is present on the home page. |
| **Postconditions:** | 1.) Advanced features are displayed. |
| **Normal Flow:** | 1.)User clicks the advanced features button. |

| | |
|---|---|
| | 2.)The user is able to see the advanced features. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Assumptions:** | None. |

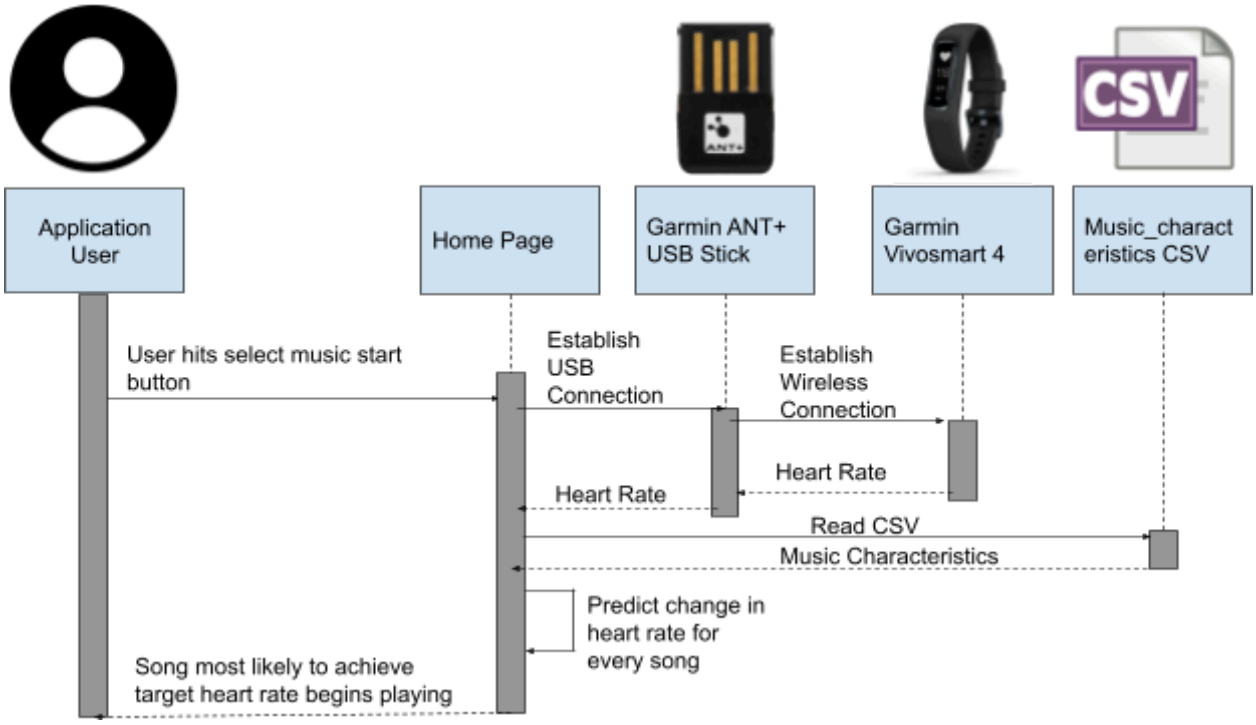
## 4.2 Sequence-Diagram



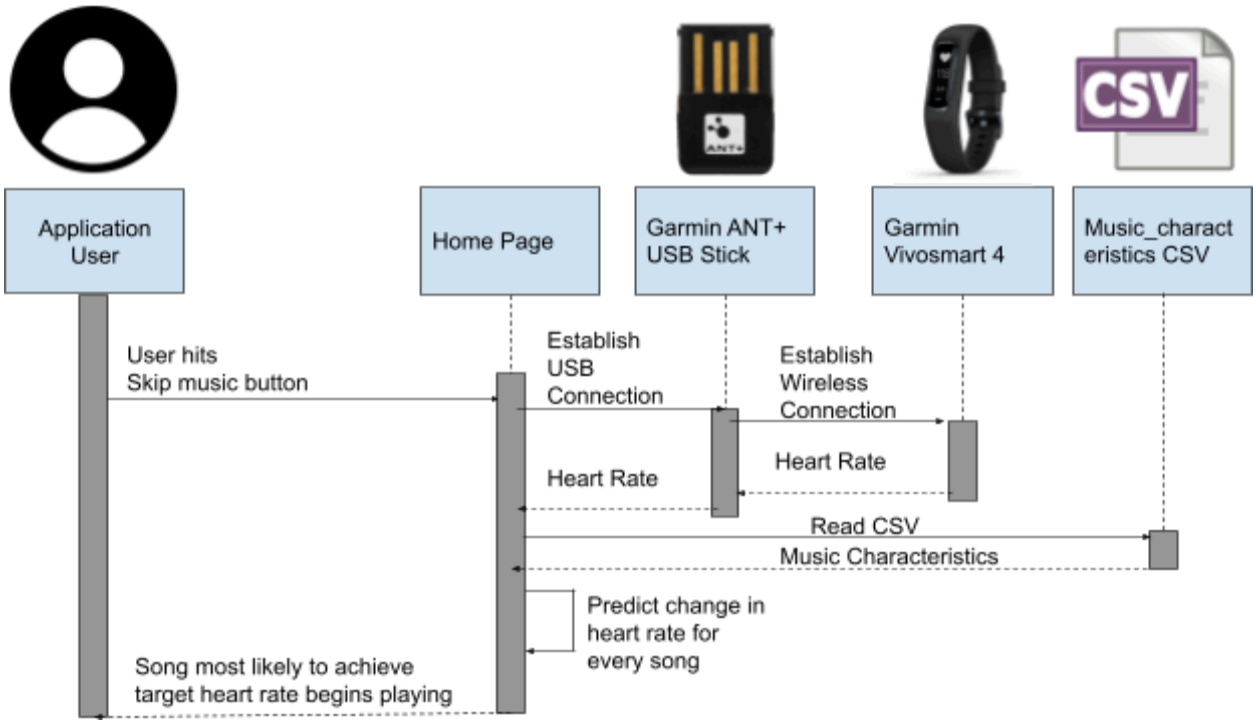

UC-1 Diagram



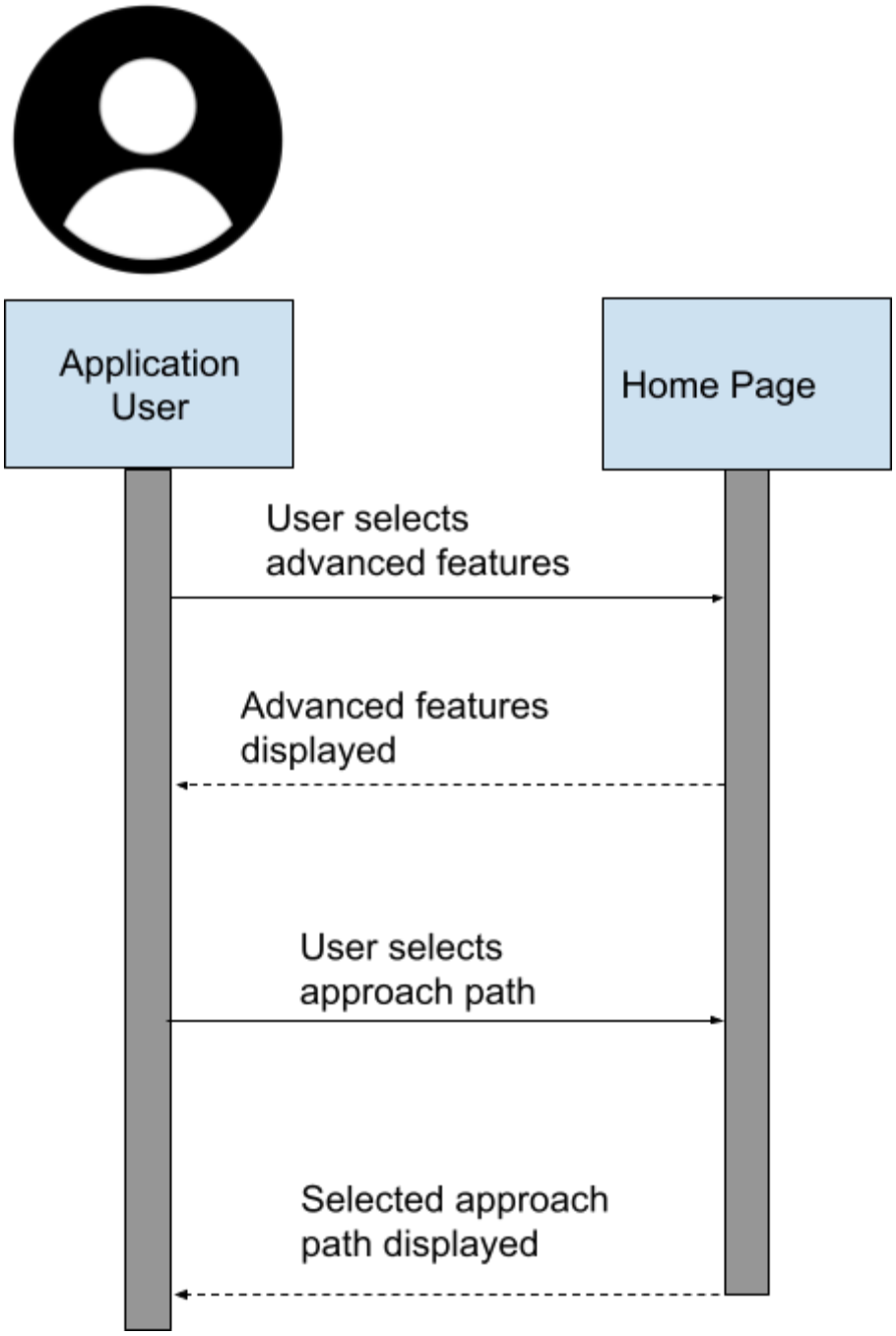

UC-2 Sequence Diagram



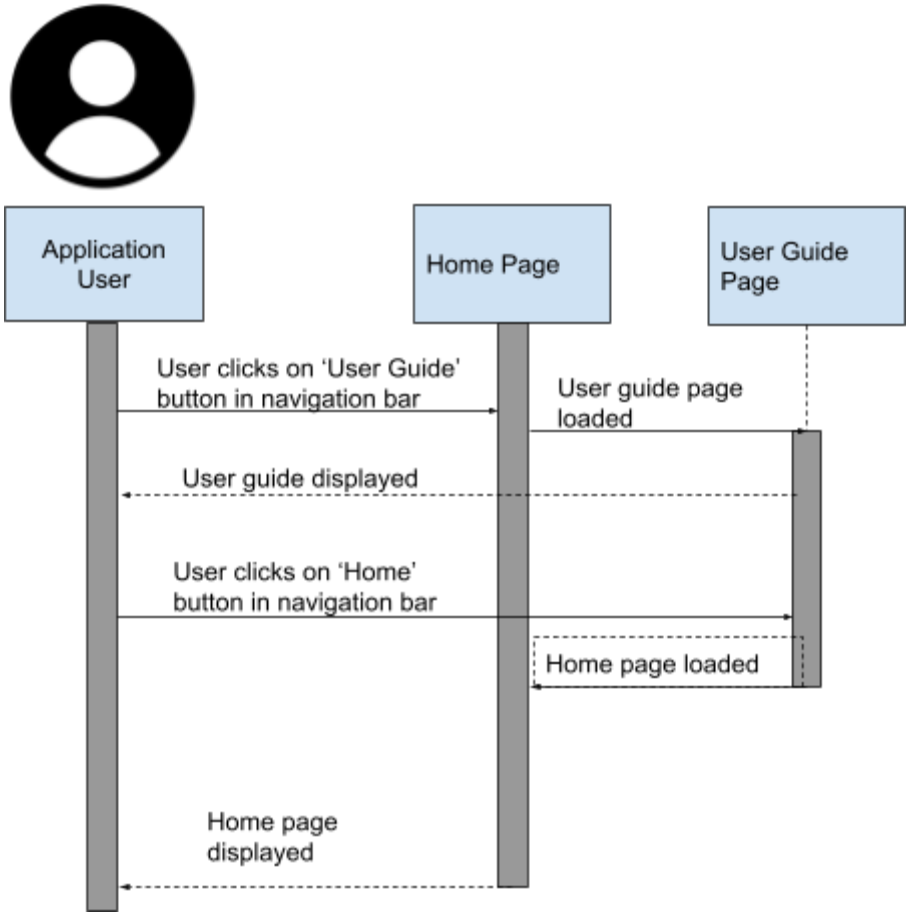

UC-3 Diagram



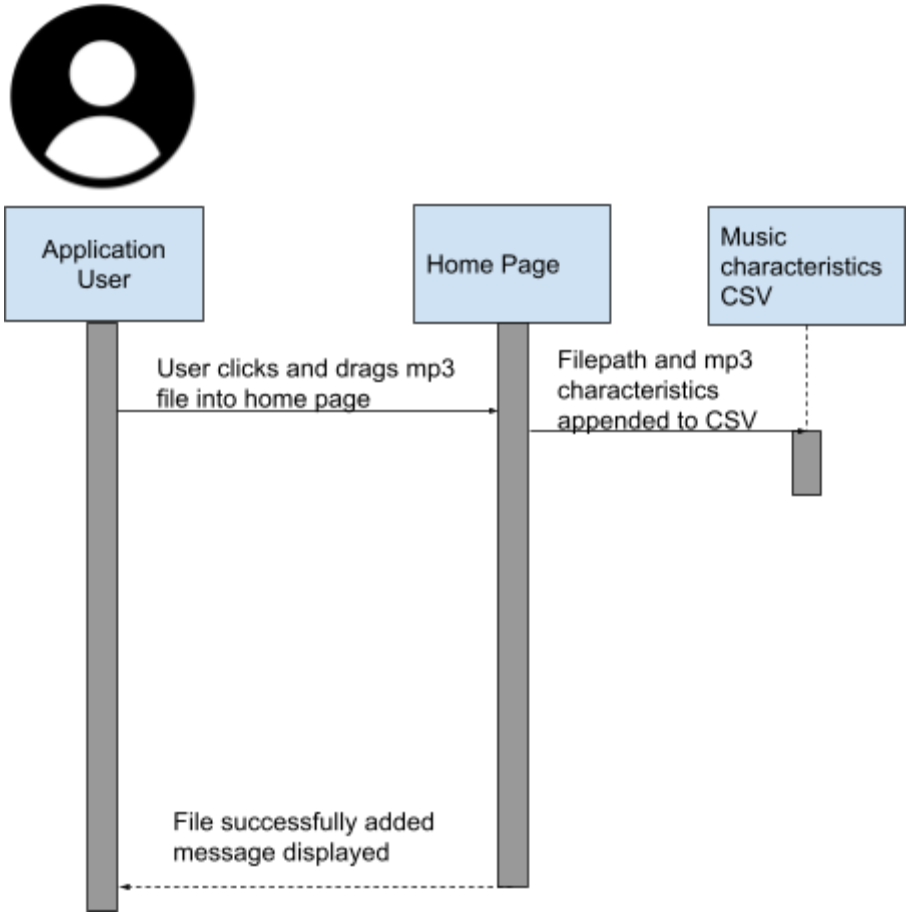

UC-4 Sequence Diagram



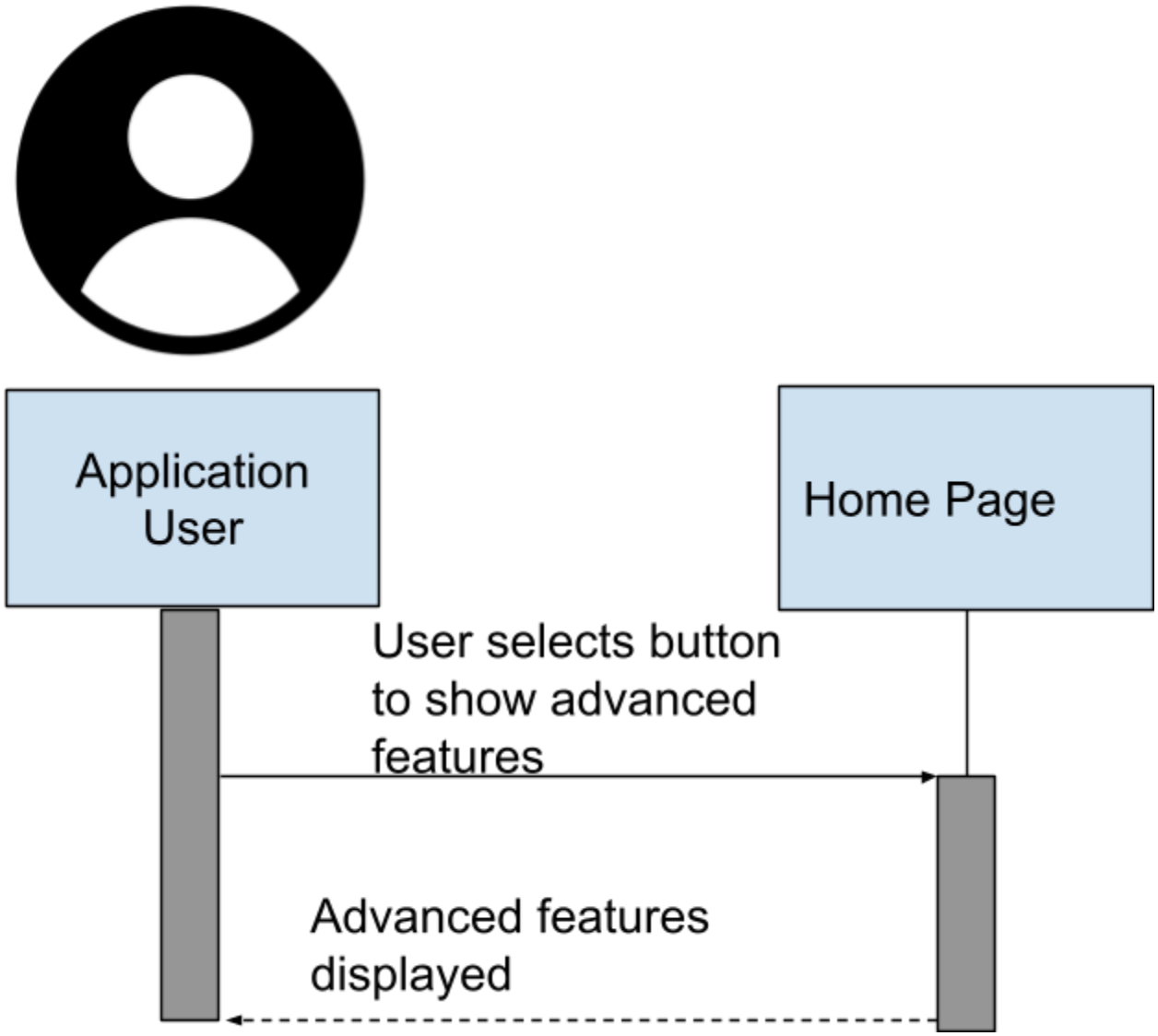

UC-5 Sequence Diagram




UC-6 Sequence Diagram

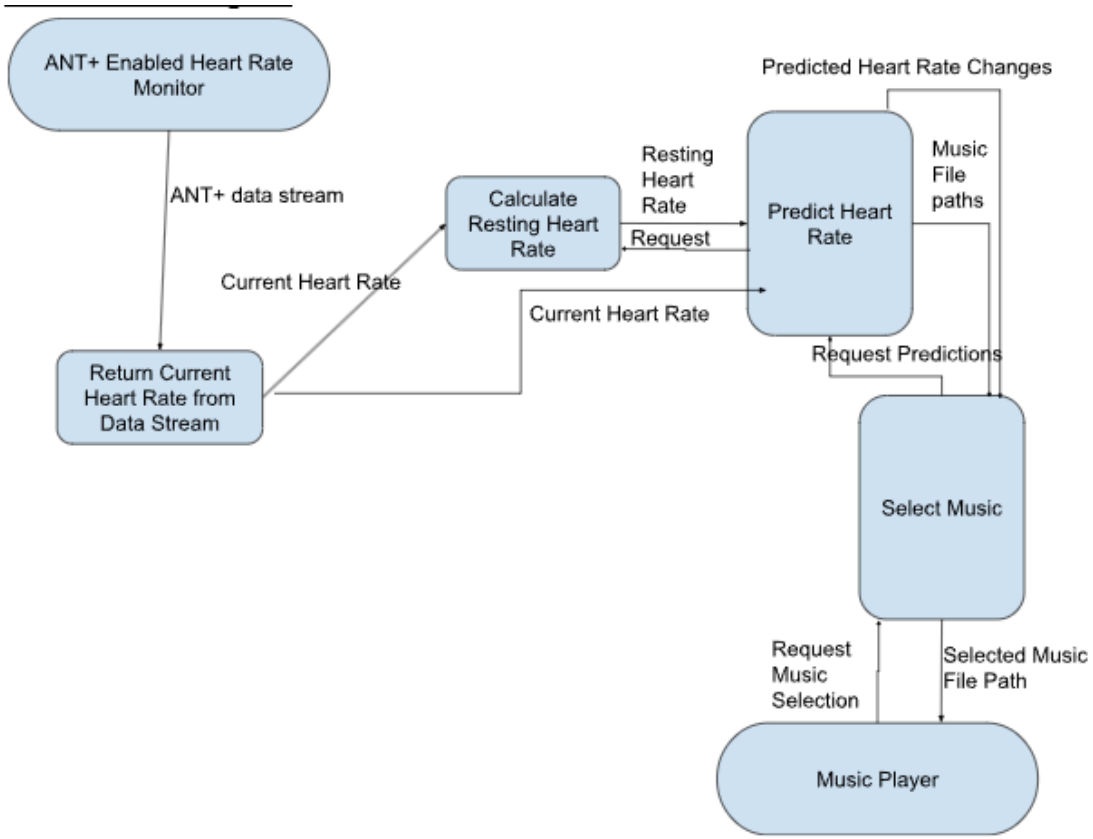



UC-7 Sequence Diagram


Application User
Home Page
User selects button to show advanced features
Advanced features displayed

UC- 8 Diagram

## 4.3 Data Flow Diagram




## 4.4 Database Diagram

The database consists of a single table for the purposes of allowing predictions to be rapidly generated from the saved characteristics of a song. The database is set up so that the characteristics of a music file are calculated as each music file is added to the database.

| **music_characteristics** |
|---|
| Song filepath |
| Tempo of Song (bpm) |
| Tempo of First 30 Seconds of Song (bpm) |
| Tempo of Last 30 Seconds of Song (bpm) |
| Length of Song (seconds) |

| |
|---|
| Average Pitch of First 30 Seconds of Song (hz) |
| Average Pitch of Last 30 Seconds of Song (hz) |
| Average Pitch of Entire Song (hz) |

## 4.5 Class Diagram

The class diagram for the application consists of the following:

| **webapp** |
|---|
| +restingHeartRate: Float<br>+approachPath: String<br>+currentHeartRate: List |
| +getCharacteristics()<br>+generateCSV()<br>+appendCSV()<br>+startHeartRateFeed()<br>+getCurrentHeartRate()<br>+getRestingHeartRate()<br>+selectMusic() |

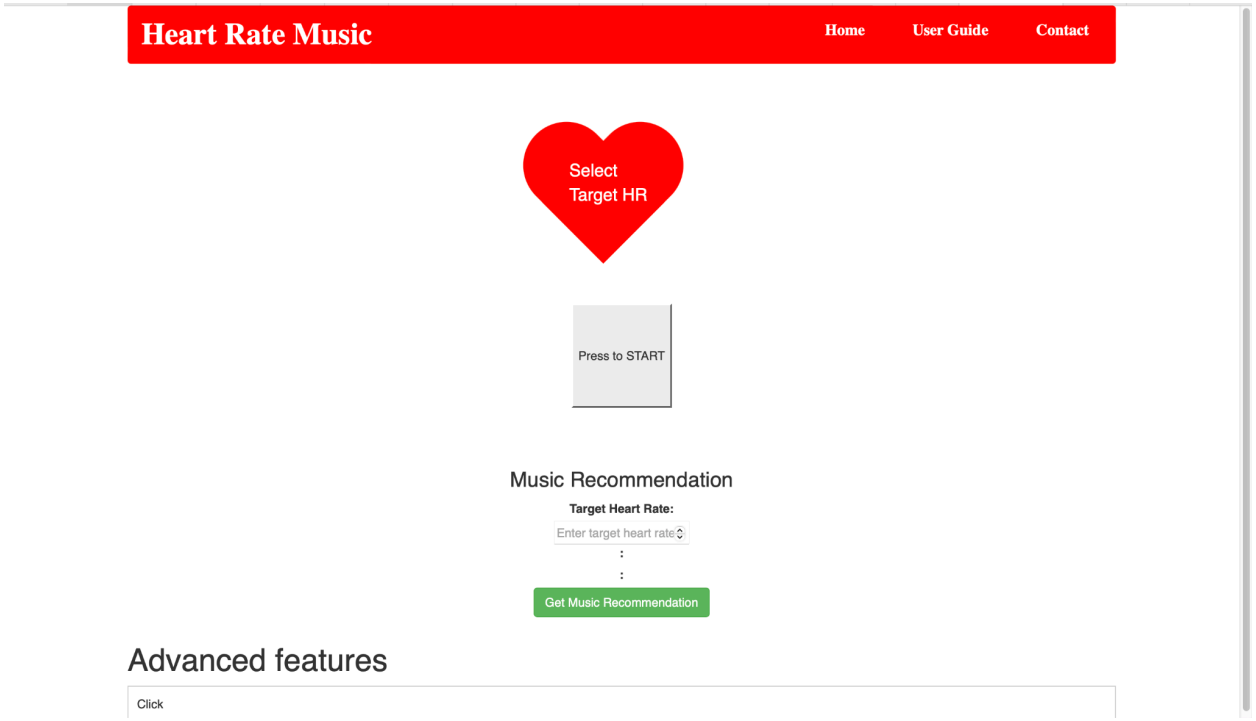
## 4.6 Application Program Interfaces

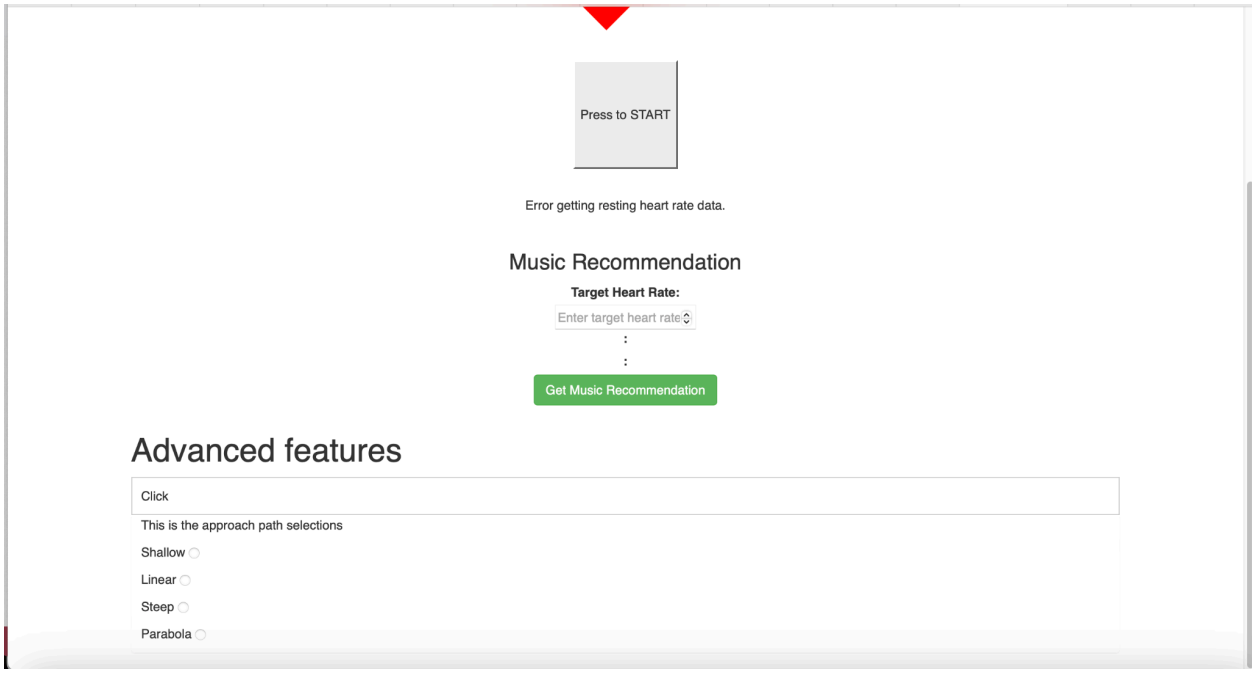
The desktop application is self contained, no APIs are used to communicate with other applications.

## 4.7 User Design Interface

The user design interface consists of a single home page, with an optional user guide available in the navigation bar that includes a tutorial. The home page contains an ‘advanced features’ feature to contain features once the user is comfortable with using the application. On opening the home page, the user’s resting heart rate is measured automatically. On the home page the user is prompted to select their target heart rate that they wish to achieve. The start button begins

selecting music for the user using the machine learning model to generate predictions. The advanced features  tab allows the user to select their desired approach path, as well as manually input their resting heart rate.



Demonstration of the Dropdown Menu