Health Monitoring Application

Group #2

Report #1, Part 1 & Part 2 & Part 3

Project URL : https://github.com/vpranathy/Health-Monitor

https://yuyangchen0122.github.io/Health-Monitoring

Softenggroup2healthmonitor-env.aabwmssgtz.us-east-1.elasticbeanstalk.com

Aniket Anilkumar

Yuyang Chen

Divyaprakash Dhurandhar

Zihao Ding

Malay Shah

Pranathy Veldandi

Table of Contents

0. Individual Contribution Breakdown

Task	Aniket	Yuyang	Divyaprakash	Zihao	Malay	Pranathy	Total
Customer State of Requirements	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
Glossary of Terms	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
System Requirements	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
Functional Requirement Specifications	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
User Interface Specifications	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
Domain Analysis	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
Plan of Work	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
Project Management	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %
References	16.67 %	16.67%	16.67%	16.67 %	16.67 %	16.67%	100.00 %

1) Customer Statement of Requirements

1.1 Problem

It is a well known fact that a healthy lifestyle depends on three main factors - diet, sleep and exercise. While diet may be dependent on additional factors like income and availability of different produce, sleep and exercise can and should be controllable. Yet, this is not the case due to the fast paced lives people lead and factors like stress, fatigue and lack of motivation acting as barriers.

Lack of education about proper fitness is a widespread problem. Many people in the country would like to exercise and stay in shape, but only a small subset of those people know how to monitor their health in a way that allows them to stay fit. There are several methods out there which people can use to get the proper information; tools such as fitness blogs, the President's Council on Fitness, Sports, and Nutrition, and the classic visit to the doctor's office are all excellent examples. However, many people don't know about those methods or choose not to utilize them, and they do their body a disservice by performing exercises that could be detrimental to their health. With information like this readily available to exercisers, it can be hard to find correct information. And even if one does find correct information, he must check to see if that information applies to a person with his body shape and size. The general problem of finding correct exercise information is that there is no set standard; there is no "one size fits all" set of guidelines which one can follow to have an effective workout. Everybody's body responds differently to different exercises, so the best that the medical community can do is to provide a set of recommendations for people of the most average body type. While this set of recommendations is good in the general, they will never tailor to the needs of one's body and workout. Also, people only have a limited amount of time on their hands to exercise. This is due to the fact their schedule can limit them.

Of all the different metrics for measuring the quality of one's fitness, heart rate is the most important factor in determining whether a workout was effective. Monitoring one's heart rate is useful because it determines whether the exerciser is performing his exercise safely as well as successfully. Experts recommend that one's target heart rate during exercise should be between 60-85% percent of the maximum heart rate, and that anything higher than 85% increases cardiovascular and orthopedic risk to the exerciser. Naturally, the target heart rate varies for people of different ages, so one should always take this into account before starting a fitness regimen. Heart rate is a significant, if not the most important, factor in determining whether a workout was done correctly and effectively, and it must be monitored closely in order to prevent injury.

Unfortunately, there are people who don't know how to correctly monitor their heart rate, and they mistakenly create a certain fitness plan based on wrong information and end up not optimizing their workout. They go to the gym, run on the treadmill at a

light pace, and consider that enough to maintain their health. They do not check their heart rate and make sure they are in the safe region of activity. This critical lack of measurement affects the entire workout. For an exercise to be effective, one must maintain a heart rate that is within the target range for an extended period of time. If not, the exerciser either puts himself at risk of injury or completes a workout that does very little to improve his fitness. Some use exhaustion and soreness after a workout as a judge of an effective workout. Although these methods do give an indication as to how effective the exercise was, they do not provide an insightful and accurate description of one's health. As a result, these people continue bad habits and routines that hinder their progress to stay fit; in fact, they may not be even making progress. A solution to the problem of uninformed exercise must have three main components; it must include all relevant medical data such as heart rate information, create a fitness plan that fits relatively well to the client's body, and provide the client with feedback about the effectiveness of his workout. Once all these components come together, the client will be able to correctly monitor his health during exercise and get the most out of his workout.

Another aspect of a healthy life that is hard to maintain is a good night sleep. Good sleep..

..can eliminate fatigue and resume physical strength, protect the brain, restore energy, enhance immunity, anti-aging and promote longevity. However, many people are suffering insomnia and other sleep related diseases. People are unable to have uninterrupted deep sleep due to stress or unhealthy lifestyle. A good sleep can be quantified and detected based on ECG and heartbeats.When people have a good sleep, their recorded heartbeats are stable without abnormal pulses.

Scientifically there are five stages of sleep:

Stage 1: Beginning of sleep, relatively light stage of sleep, can be considered as a transition

period between awake and sleep. It lasts about 5 ~ 10 mins.

Stage 2: It lasts about 20 mins, heart rate begins to decrease.

Stage 3: Transition period between light sleep to very deep sleep.

Stage 4: Stage 4 is a deep sleep that lasts approximately 30 mins, bedwetting and sleepwalking

usually happen at the end of stage 4 sleep.

Stage 5: Most dreaming occurs during this stage, known as Rapid Eye Movement(REM) sleep.

During this stage, body system become more active, and heart rate is supposed to rise up. On an average, we enter the REM stage approximately 90 mins after falling asleep. The lasting time of REM stage might get longer with each sleep cycle, up to an hour as sleep progresses. In this modern day and age, an ideal sleep cycle has become a myth

due to increased pressures and stress levels. People do not have the required resources to analyse and improve their sleep cycle. As a whole, healthy lifestyles and activities in general are hard to maintain.

1.2 Solution

It has now been well established that heartbeat rate is an essential indicator of an effective workout and a good sleep cycle and in general, an indicator of a healthy lifestyle. Our proposed solution is a web application and a mobile application integrated with a pulse sensor. This will allow us to record and visualize user data and give the user information on how to improve his/her lifestyle. The proposed application would consist of various different aspects, from basic exercise statistics to the use of music to regulate heartbeat to provide better workouts and sleep cycles. Future work includes expansions of the application, including implementations of machine learning to let the app be unique to each user. To generate the huge amount of data required for it, the user would have to sleep with the pulse sensor on their finger and the database connected to our application would thus get their heart rate fluctuations over their entire sleep period. After a while, the data acquired would be enough to perform machine learning on it to give helpful suggestions to improve sleep cycles or play soothing music to help the user relax.

1.2.1 Music

Music and healing once went hand in hand. The Chinese character for medicine includes the character for music. In ancient Greece, music was used to ease stress, promote sleep, and soothe pain. Native Americans and Africans used singing and chanting as part of their healing rituals. Music can powerfully evoke and modulate emotions and moods, along with changes in heart activity, blood pressure (BP), and breathing. Heart rate can be influenced by factors like age, fitness, having a cardiovascular disease, emotions, body-size etc. The heart rate also varies according to the activity the person is involved in. Heart rate acts as a gauge to judge how well the person is involved in that activity. Hence, heart rate should be modulated to get the best out of the activity. Different music tempo can be used to modulate the heartbeat of the person. For example, if the person wants to workout, he can reach optimum heart rate by listening to music with high tempo. Similarly, the tempo varies with the activities like meditating, studying and sleeping.

1.2.2 Database

In this project, we will mainly use the AWS RDS database. First of all, this system will allow new user to sign up a new account, and the account information will be stored directly in database. Whenever existing users log in to the system, the system will detect whether their username and password can match the data to the database. If they match, the system will allow users to log in to the system; otherwise, users will not be allowed to log in.

Users will want to monitor their personal health status, so our project will allow the user to view his heartbeat data graph directly on his phone. This eliminates the inconvenience of having the user log in to a personal account on a website to view his data, because everything he needs will be on the phone itself. But of course, they are allowed to view the data on our web-based system. All the data collected from Arduino will be stored on the database, and the system will perform the necessary database calls to retrieve that data. That data will be processed and formatted into different graphs that will display the correlation between music and heart rate.

1.2.3 System Architecture Diagram

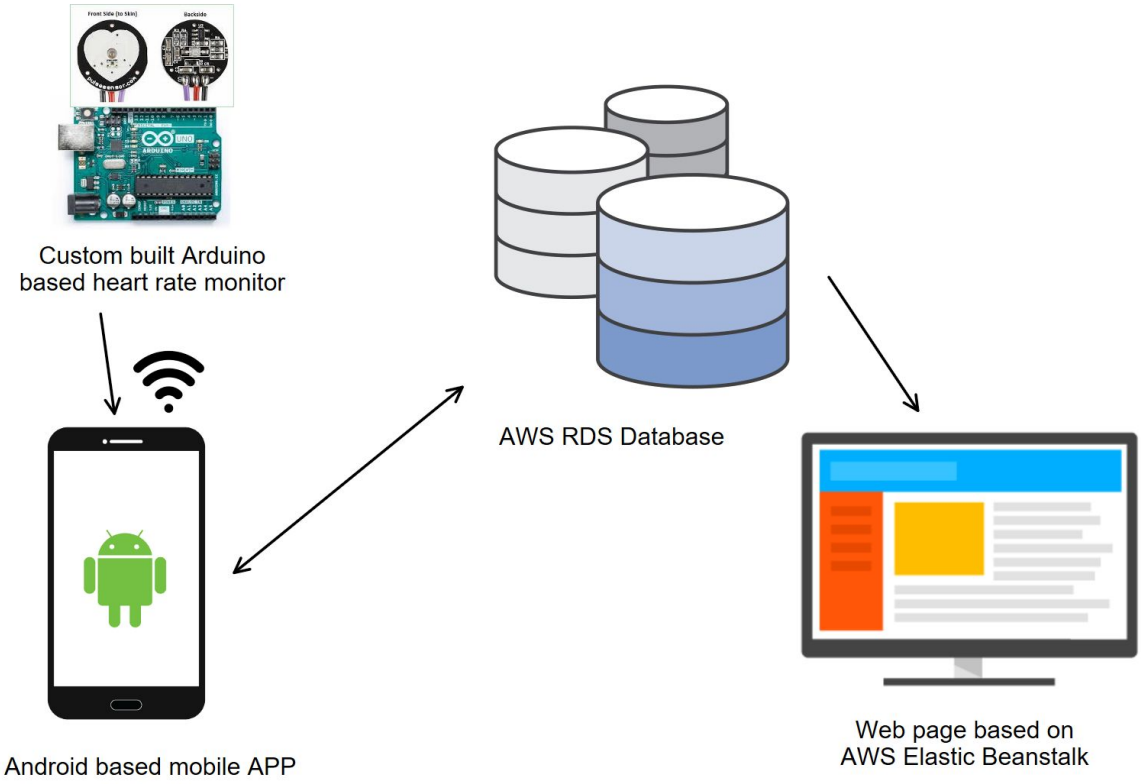


Fig: System Architecture

The mobile application is the core part of our system since it needs to upload the data from the heart rate monitor and present both visual and music feedback to the user. The mobile application part uses the Model-View-Controller architecture. The Arduino heart rate monitor collects the heart rate of the user and uploads the information to the Database via Android smartphone, those data will be processed with AWS Lambda. The graphs and other feedback produced by the AWS Lambda will send back to the APP to let app to play proper music and let the user get to know about their current health status.

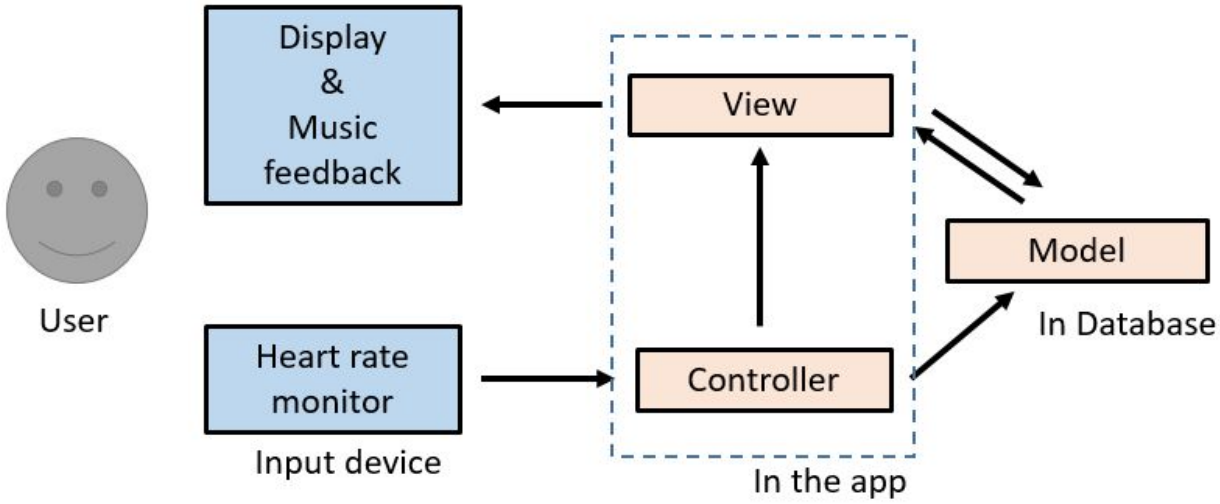


Fig: MVC Architecture for the Mobile APP

The webpage version of the application uses the client/server architectural style, which the data and code to run the application is stored on the server and clients can access their data using all kinds of web browsers. The webpage of the APP is mainly used to display user's histories. It may also provide customized health suggestion according to the user's data.

1.2.4 Product Usage

- The heart rate monitor should only be worn by the user on their finger and it will be linked to an Arduino board. It should be worn while it is in use - only while the user is exercising or the user is sleeping. Although it is safe to wear the heart rate monitor during other times, there will be no benefit unless the application is currently running.

- Users may choose to use the Heart Rate Monitor while not sleeping or exercising if they wish to adjust their heart rate for alternate reasons (possibly for playing video games or preparing for an exam).

- The user will run the android application, and then input a target heart rate. The software will then choose a song based on your current heart rate and begin to either raise or lower it. Once the target heart rate is obtained within a certain tolerance, the software will work to maintain this heart rate rather than increasing/decreasing it.

- Music will be selected from the user's own personal music library (which should be stored on the flash memory of the Android device) to either increase or decrease the user's heart-rate. Music will be played by our software.

- The software will select and play music according to the user's current heart rate in real-time as it receives information from the connected heart rate monitor.

- Music will be delivered through the headphone jack on the Android device or through any bluetooth device.

- Receive information on the songs that are listened to in relation to their usage of the Android device. (What songs were listened to, which songs were the most effective at changing their heart rate, etc.)

1.2.5 Product Ownership (tentative)

Our team will be divided into three smaller sub-teams of three, two and one individuals each, the pairings listed below. Each sub-team will be responsible for hardware, software and web development and provide a brief description of their work on a shared Google drive folder. They will also include the necessary UML diagrams and charts. Every week we will meet twice for 1-3 hours. During the meeting, we will have a specific agenda that primarily involves the week's progress and upcoming deliverable. Our discussion will probably be centered along the following questions: 1) What did you work on this past week? 2) What do you plan on working on next week? 3) Are there any changes that need to be made to the project? Every week, a different team member will take the lead for the next deliverable to ensure that everything is on time.

- Aniket, Pranathy and Malay will develop the Android application which will connect to the Arduino heart rate monitor and provide the user interface as well as all the other user features.

- Yuyang will work on a database that receives, stores, and processes the data from the Android device. The database will export the data to a CSV file, he will use python to do the data visualization based on the CSV file, it will provide users a more intuitive data analysis. He will also work on the web development to synchronize all information between web client and mobile client, making it easier for users to use the system.

- Divyaprakash and Zihao will be responsible for designing and programming the Heart Rate Monitor using the Arduino board as well as configuring it to connect via bluetooth to the android smartphone.

1.2.6 Devices and Specifications

- Heart Rate monitor: Arduino based custom built monitor
- Smartphone: Running Android 5.0+

2) Glossary of Terms

- **Arduino** - An open-source platform used for building electronic and software projects
- **API -** An application program interface (API) is a set of routines, protocols, and tools used for building the software applications
- **AWS** - Amazon web services (AWS) provides cloud computing services
- **AWS RDS Database -** A collection of information that is organized so that it can be easily accessed, managed, and updated
- **Beats per Minute (BPM)** - BPM is the amount of times that the heart beats given one minute of time
- **Encrypted -** A document is encrypted if it can not be accessed by the public and hence only allows specific users to read the data
- **Fetch -** A process in which a computer retrieves the program from its memory to carry out an action that is instructed by the user
- **Histogram** - A bar graph which shows how often something appears within a certain range

- **Interface -** A device or program enabling a user to communicate with the computer
- **Keystrokes -** A single depression of a key on a keyboard in order to measure work
- **Loading Screen -** Provides a plethora of fun facts to keep users engaged between transitions and educate them about general health
- **Module -** A number of distinct but interrelated units from which a program could build up
- **Physical Data -** All data pertaining to the user's step count, heart rate, and type of workout
- **Profile -** A unique set of information pertaining to a particular user, including but not limited to their avatar, lifetime points, and username
- **Query** - A question sent to the database for the purpose of extracting data
- **Sidebar-** A menu of all the modules displayed on the left side of the screen for easy access
- **Smartphone** - A smartphone is a handheld personal computer. It possesses extensive computer capabilities, including high-speed access to the Internet using both Wifi and mobile broadband.
- **System Admin -** Approves community module posts, group and challenge creation, and login
- **UI -** Stands for User Interface. Refers to the general appearance and functionality of the modules, and listens for the user's keystrokes
- **User Effort Estimation -** Refers to the number of keystrokes or mouse clicks needed to navigate and get to the actual context where the user needs to enter the data
- **Visitor -** An outsider who does not yet have an account that is able to create a new account from the "Sign Up" page on the default page of the app.
- **Widget -** An application, or a component of an application, that enables a user to perform a function or access a service

3) System Requirements

3.1 Enumerated Functional Requirements

ID	Priority Weight	Requirement
REQ-1	5	System should play the correct music according to the customer's current activity
REQ-2	4	System should retrieve data from arduino sensor
REQ-3	4	System should transfer data to mobile application
REQ-4	4	System should store data into database
REQ-5	5	System should transfer data to web application
REQ-6	3	System should let the user visualize their data by graphs
REQ-7	2	The user should be able to rank the songs
REQ-8	1	The user should have the ability to change the song if their activity changes
REQ-9	1	The user should have the ability to pause the song if their activity stops

3.2 Non-Functional Requirements

ID	Priority Weight	Requirement
REQ-10	5	The Android Application will be user friendly and require very little user navigation
REQ-11	4	The system shall run constantly, requiring little maintenance
REQ-12	4	The system shall present all information in an orderly manner
REQ-13	3	The system shall recover easily
REQ-14	3	The system will allow multiple users online
REQ-15	3	The system will prevent users from modifying existing system data

3.3 On-Screen Appearance Requirements

This section contains mock-ups of the Android and web application's user interface. This user interface is subject to change. The images presented below do contain essential information for the user. This information will be functionally developed by the team. The input displays the application name, the start button, and the previous session. There is a study mode, sleep mode, and work mode. The minimum heart rate, maximum heart rate, and music being played is also listed within the application.

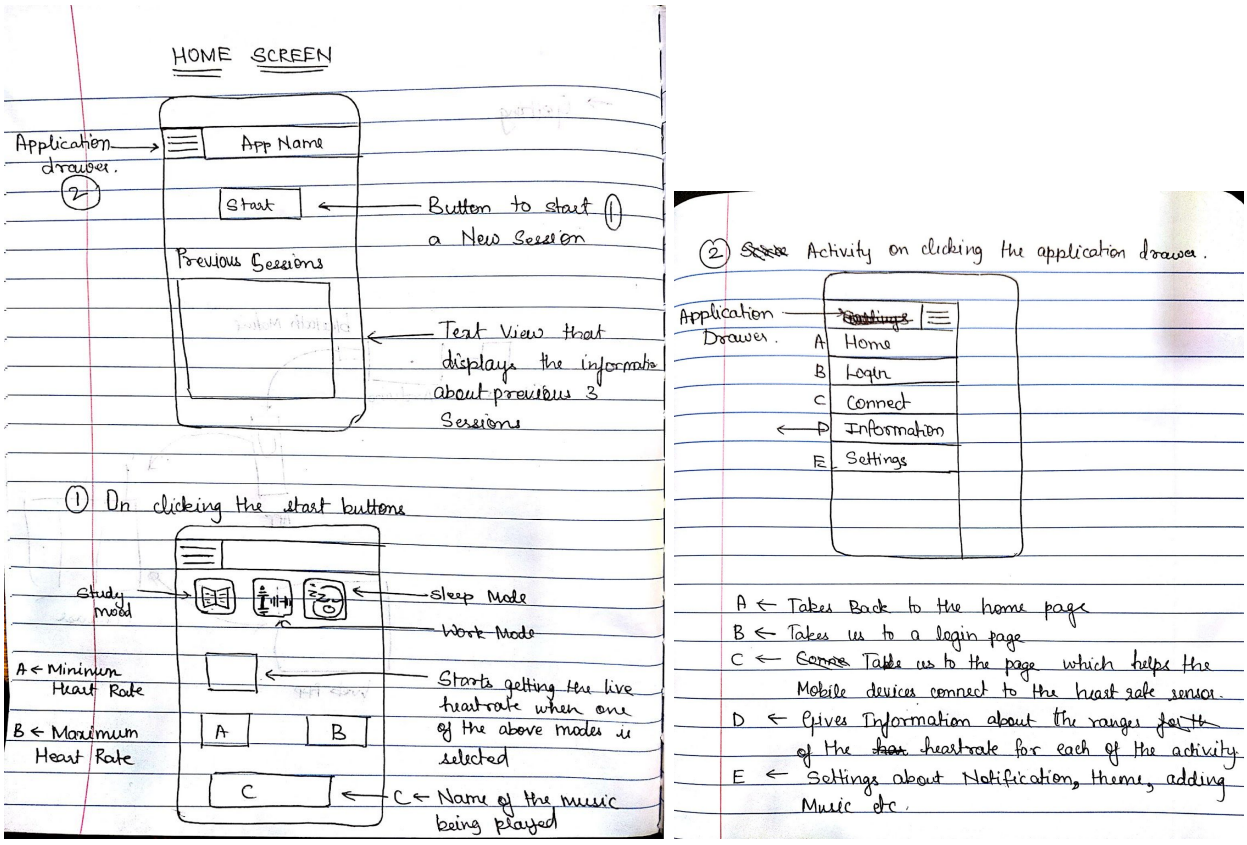


Fig: A basic sketch showing the UI of the Android App

Web-based Dashboard page

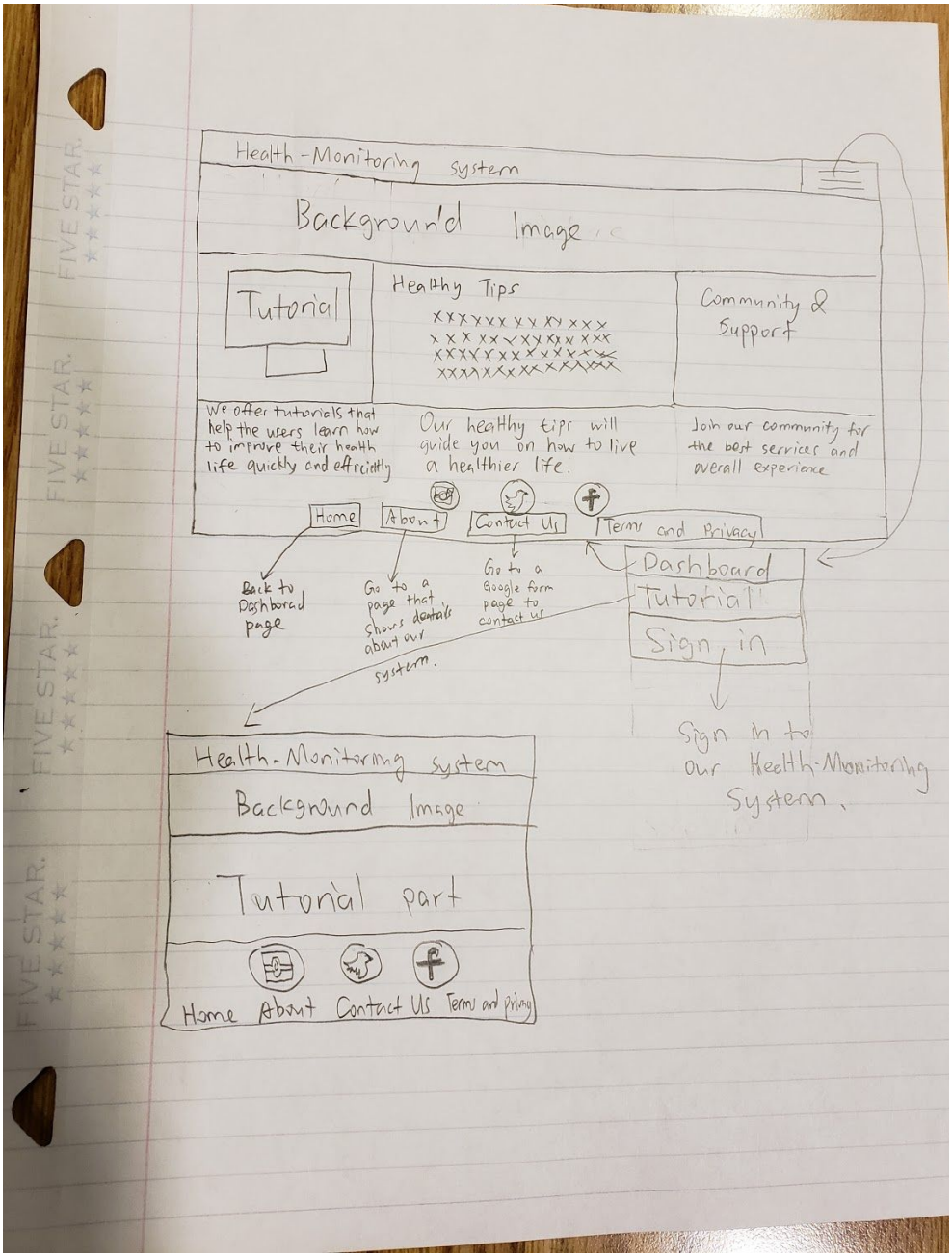


Fig: A basic sketch of the UI of Web Application for data visualisation

The Webpage will be used for data visualization. Graphs depict the heart rate of the user during an activity. Another graph will display the mean of the heart rate for the last n times the same activity was conducted.

Web-based Login page

Each user will have to login with their created username and password to access their data and health suggestions.

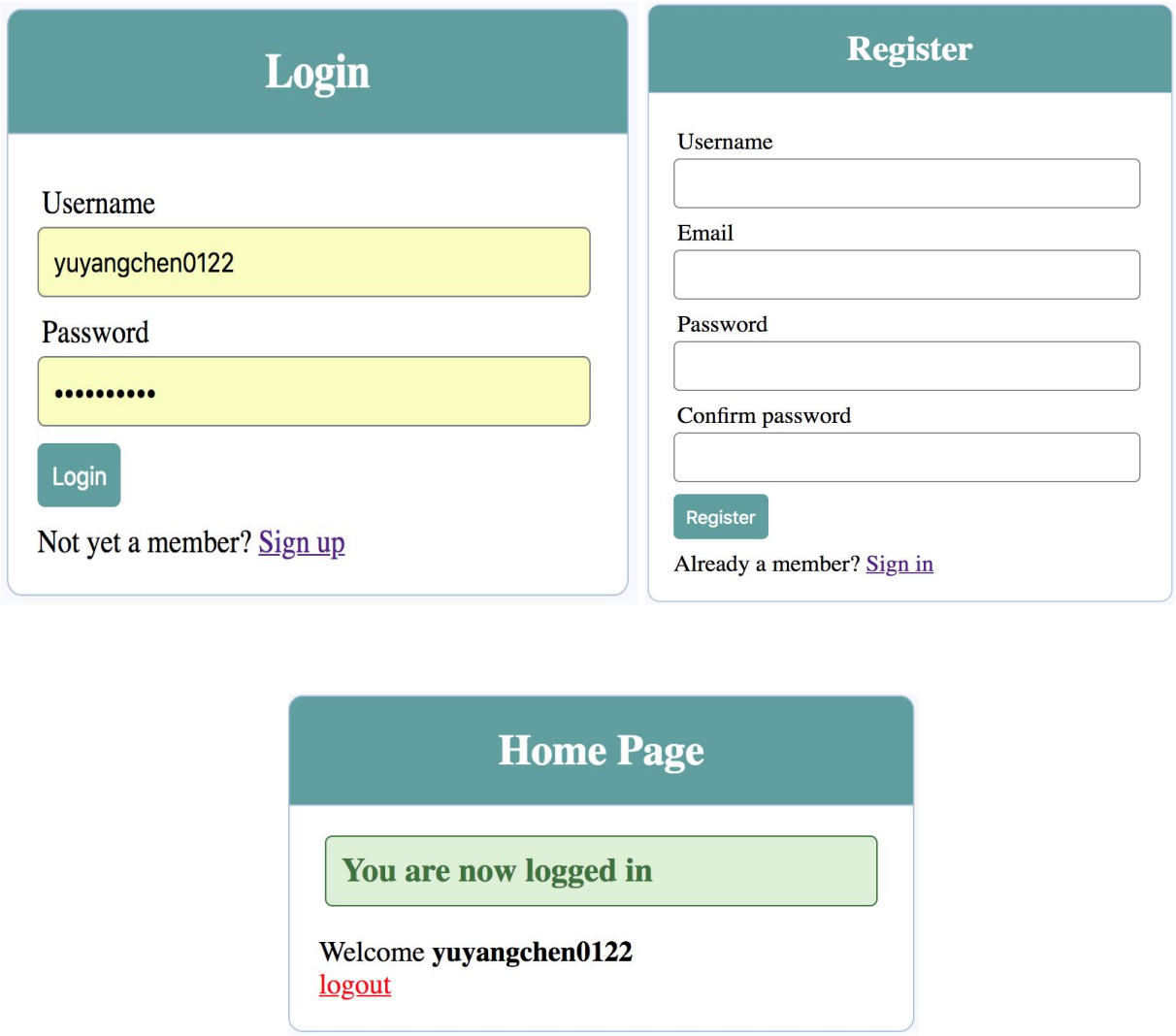


Fig: User login page for web application

4. Functional Requirement Specification

4.1 Stakeholders

Stakeholders include individuals and organizations which are interested in the completion and use of a given product. The amount of stakeholders and different types of stakeholders relies on the versatility and ease-of-use of the product in question. Due to this software's very simple interface and design, stakeholders may include users of all ages and multiple types of organizations who are interested in obtaining easier sleep or a more energetic workout. Examples of potential stakeholders include:

- **Administrator:** The administrator will look into the system to ensure the smooth functioning of the online leaderboard, rewards and the feedbacks from every-user.
- **Cardiologists**:The cardiologists will use this application to get the rest heart-rate and comment on the health of the heart based on the physical structure of the patient.
- **Fitness-instructor**: The instructor can monitor the heart-rate of the of the students during their fitness activity and suggest changes in the warm-up regime to get the heart-rate to the optimal level.
- **Developer:** The developers of the application can work on the data to develop machine learning algorithms which can help to improve the music suggestions, advice changes in daily activities and consult sleep hours.

4.2 Actors and Goals

Actors can be defined as people or devices that will directly interact with the product, and can also be loosely labeled as either "initiators" or "participators". These actors will have a specific goal with the given product, which is what the actors are attempting to achieve by interacting with the system. Actors and their respective goals are:

Actor	***Actor's Goal***
User(initiator)	To increase heart rate for exercising
User(initiator)	To decrease heart rate for sleeping
User(initiator)	To analyze health information from given graphs

Doctor/Fitness instructor	To consult the changes that should be brought to improve health of the heart
DataBase	A repository of the heart-rate data from previous activities.
Music Suggestion Algorithm	An algorithm that asks users music preference and suggests music from the same genre

This product is one which only requires the interaction of one human actor, the user of the product. While there is the potential for other humans to interact with the user's health information which is produced, only the user himself is considered an actor. The heart monitoring device is a participating actor worn by the user to monitor information and relay the information via Bluetooth back to the smartphone which is running the application. The one exception is that our heart monitoring device may be an initiating actor and notify the user if his/her heart rate is abnormally high or low. In this case, the user would be the participating actor.

4.3 Use Cases

i. Casual Description

UC#1 Collecting Health Data from Arduino
The user data shall be collected and then transferred to the database which will be linked to the web and mobile application.

UC#2 Displaying Heart Rate Data
The heart rate data will be displayed through the web and mobile application.

UC#3 Visualization of Data through a Graphical Representation
The data shall be presented to the user through graphs and be distinguishable through a legend.

UC#4 Music Selection & Playback
The system shall recommend music to the user dependent on their activity and playback that music accordingly.

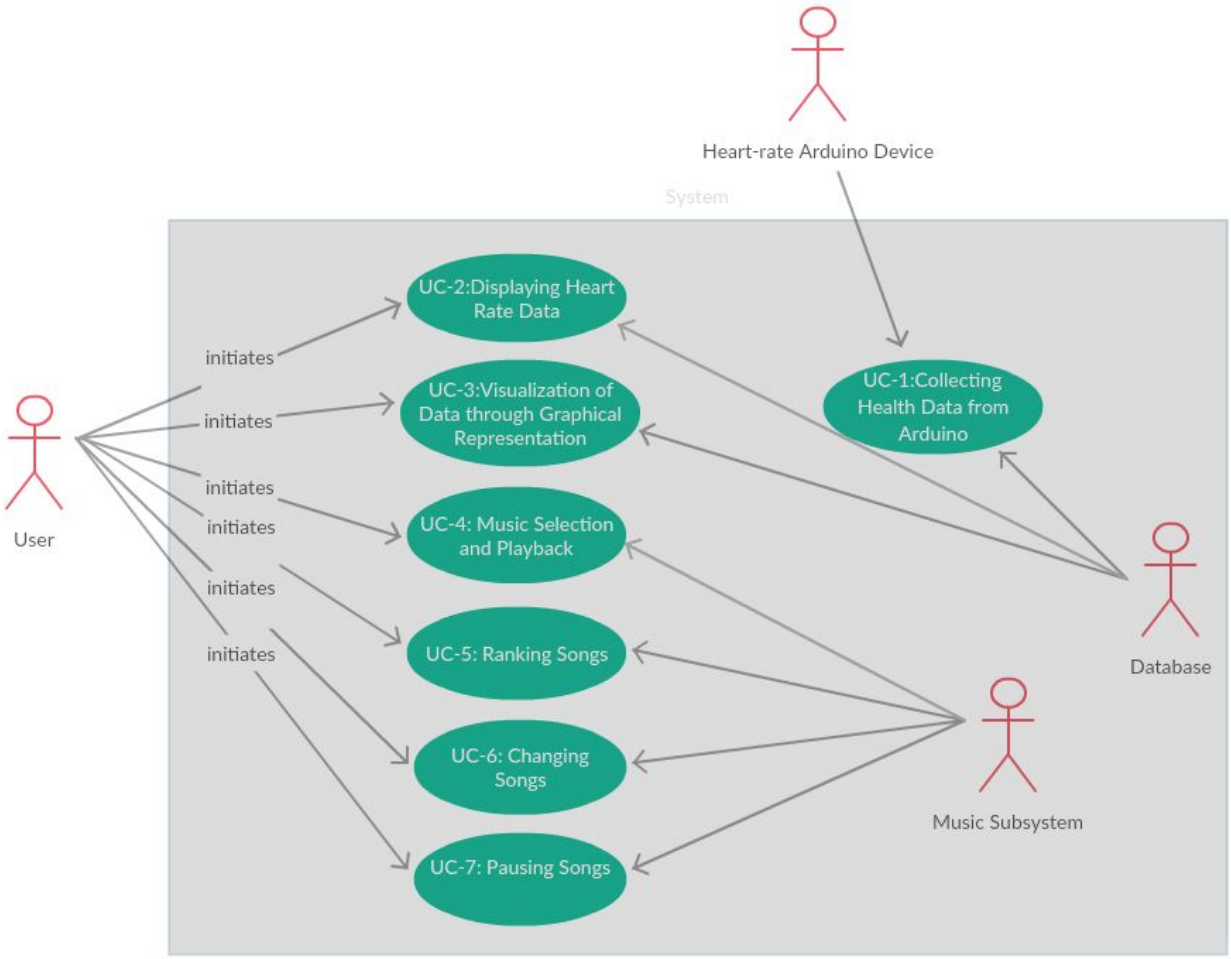
UC#5 Ranking Songs
Users will give their feedback about the songs after every activity. This feedback will be

in the form of points out 10. This pointing system will help the application to rank the songs.

UC#6 Changing Song
Users will have the ability to change the song within the selected genre to their liking. This will help the user reach their desired heart rate.

UC#7 Pausing Song
Users will have the ability to pause the song if they choose to stop the activity or take a break from the activity they are currently performing.

ii. Use Case Diagram



iii. Traceability Matrix

The Traceability Matrix allows the reader to cross the function and non-functional requirements described earlier with the use cases. This will demonstrate which use cases fulfill each requirement.

	UC#1	UC#2	UC#3	UC#4	UC#5	UC#6	UC#7
REQ-1				X			
REQ-2	X						
REQ-3	X	X	X				
REQ-4	X	X					
REQ-5	X	X					
REQ-6			X				
REQ-7					X		
REQ-8						X	
REQ-9							X

iv. Fully Dressed Descriptions

Use Case UC-1:	Collecting Health Data from Arduino
Related Requirements:	REQ-2,REQ-3,REQ-4, & REQ-5
Initiating Actor:	System
Actor's Goal:	To collect the data and transfer it to the database
Participating Actors:	
User Preconditions:	The system collects the data and transfers it to the database.
Postcondition:	User will be able to view data.
Failed End Condition:	Unable to collect data please try again.

Flow of Events for Main Success Scenario:
-> 1.Users performs activities and to generate data.
<- 2.System collects the data.
-> 3.System waits to transfer data.
<- 4.System transfers the data to the database.

Flow of Events for Extensions:
4(a) No data acquired.
<- 1.System would display a error page saying the data cannot be acquired due to insufficient data.

Use Case UC-2:	Displaying Heart Rate Data
Related Requirements:	REQ-3, REQ-4 & REQ-5
Initiating Actor:	Users
Actor's Goal:	To display users' heart rate data
Participating Actors:	
System Preconditions:	The website displays at the Dashboard Page.
Postcondition:	Users get their health index dashboard by month or day.
Failed End Condition:	Unable to show the health index please try again

Flow of Events for Main Success Scenario:
-> 1. The system reads data from the database.
<- 2.System connect to the mobile applications.
-> 3.System waits to connect.
<- 4.System displays the heart rate data on the Dashboard Page.

Flow of Events for Extensions:
4(a) No data acquired.
<- 1.System would display a error page saying the data cannot be acquired from database.

Use Case UC-3:	Visualization of Data through a Graphical Representation
Related Requirements:	REQ-3, REQ-6
Initiating Actor:	System
Actor's Goal:	To display graphs that shows the relationship between heart rate and other factors to users, to explain how these factors affect heart rate
Participating Actors:	
User Preconditions:	The website displays at the Dashboard Page.

Postcondition:	Users get their Heart rate analysis chart by month or day.
Failed End Condition:	Unable to show the Heart rate analysis please try again

Flow of Events for Main Success Scenario:
-> 1. The system reads data from the database
<- 2.System waits to transfer data.
-> 3.The system generates heart rate data analysis graphs.
<- 4.System waits to transfer graphs to Web page, which will be displayed to users.

Flow of Events for Extensions:
3(a) Heart rate analysis chart generation failed
<- 1.System would display a error page saying Heart rate analysis graph generation failure based on reading data error

Use Case UC-4:	Music Selection & Playback
Related Requirements:	REQ-1
Initiating Actor:	System
Actor's Goal:	To select the music of their choice and play it at their choice.
Participating Actors:	
User Preconditions:	The application shall allow the user to play music.
Postcondition:	The user will play their music for their current activity.
Failed End Condition:	Unable to play music for the user.

Flow of Events for Main Success Scenario:
-> 1. The system allows the user to select their song choice
<- 2.System waits for users choice
-> 3.The system plays the users choice
<- 4.User continues their activity

Flow of Events for Extensions:
3(a) Failure of Music Selection

<- 1.System would display a error page saying music selection failure based on the song list of the user.

Use Case UC-5:	Ranking Songs
Related Requirements:	REQ-7
Initiating Actor:	User
Actor's Goal:	To pick their favorite songs for their favorite activities.
Participating Actors:	
System Preconditions:	The user has ranked songs within the application..
Postcondition:	The ranked songs will be available for use
Failed End Condition:	The user will not be able to play their ranked songs.

Flow of Events for Main Success Scenario:
-> 1. The system allows the user to select their ranked song choice
<- 2.System waits for users choice
-> 3.The system plays the ranked song
<- 4.User continues their activity

Flow of Events for Extensions:
3(a) Failure of Ranked Music Selection
<- 1.System would display a error page saying music selection failure based on the song list of the user.

Use Case UC-6:	Changing Songs
Related Requirements:	REQ-8
Initiating Actor:	User
Actor's Goal:	To change the music of their choice and play it at their choice.
Participating Actors:	

System Preconditions: This application allows the user to change the music being played

Postcondition: The application changes the music that the user listened to before to broadcast the new music that the user selected.

Failed End Condition: The application cannot change the music that the user has previously listened to and continues to play the music that the user wants to change.

Flow of Events for Main Success Scenario:
-> 1. The system allows the user to change their song choice
<- 2.System waits for users choice
-> 3.The system stop playing the previous song and change to play the user's new choice
<- 4.User continues their activity

Flow of Events for Extensions: System would display a error page saying music selection failure based on the song list of the user.

Use Case UC-7: Pausing Songs

Related Requirements: REQ-9

Initiating Actor: User

Actor's Goal: To pause the music that is playing
Participating Actors:

System Preconditions: This application allows the user to pause the music being played

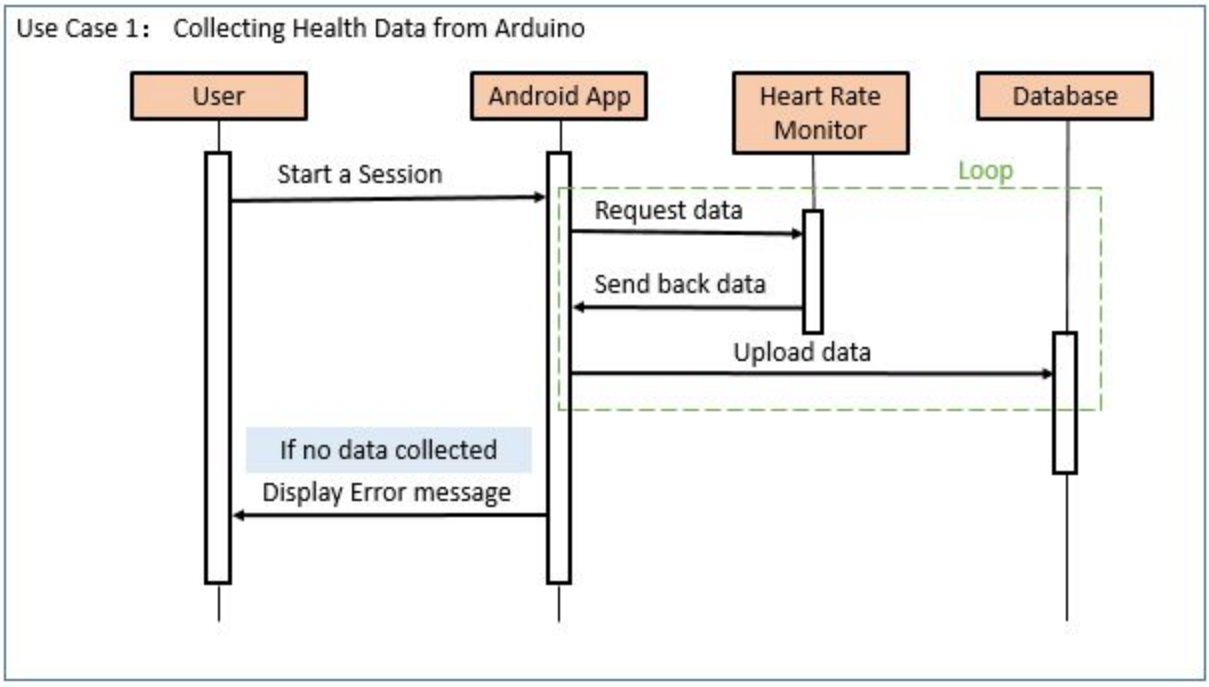
Postcondition: The application suspends the music being played and does not continue to play any music

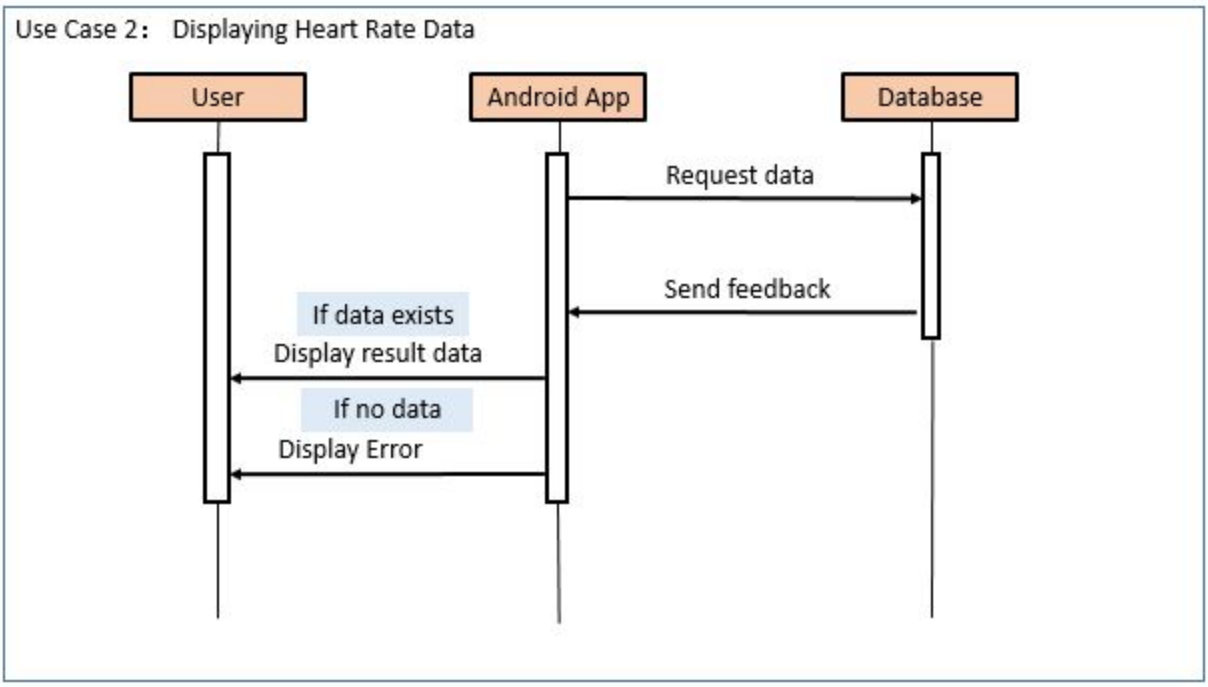
Failed End Condition: The application cannot pause music being played and continue to play music

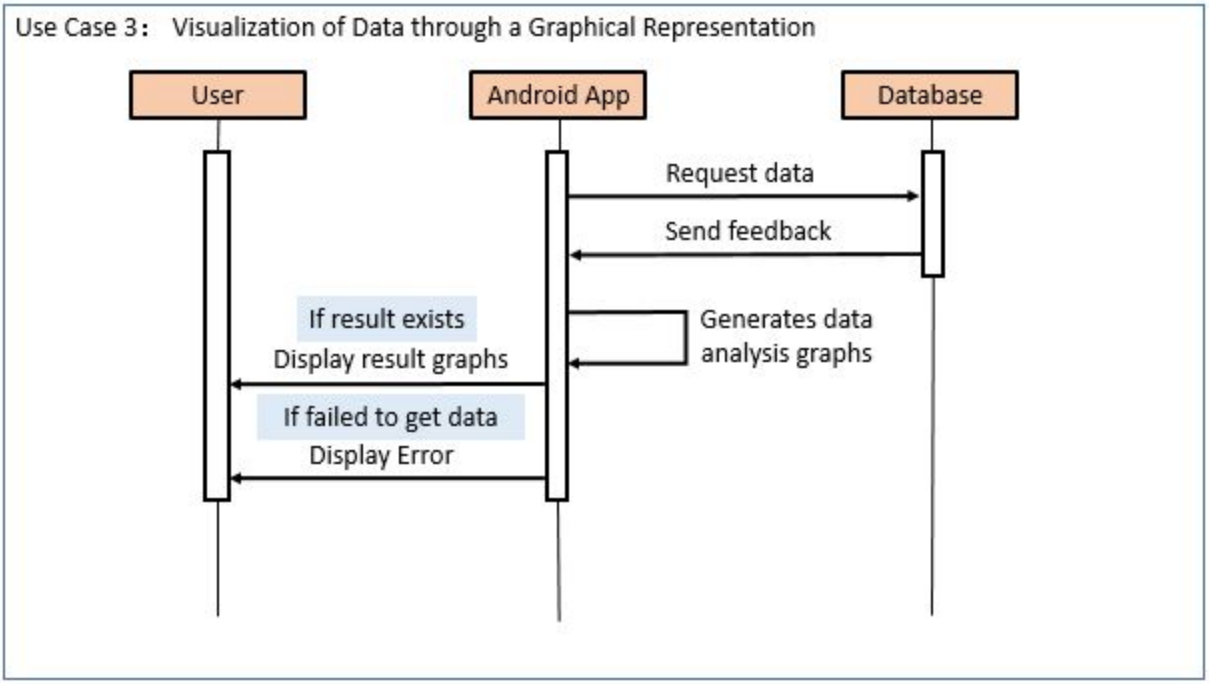
Flow of Events for Main Success Scenario:
-> 1. The system allows the user to pause the song being played
 <- 2.System waits for users choice
-> 3.The system stop playing the music
<- 4.User continues their activity

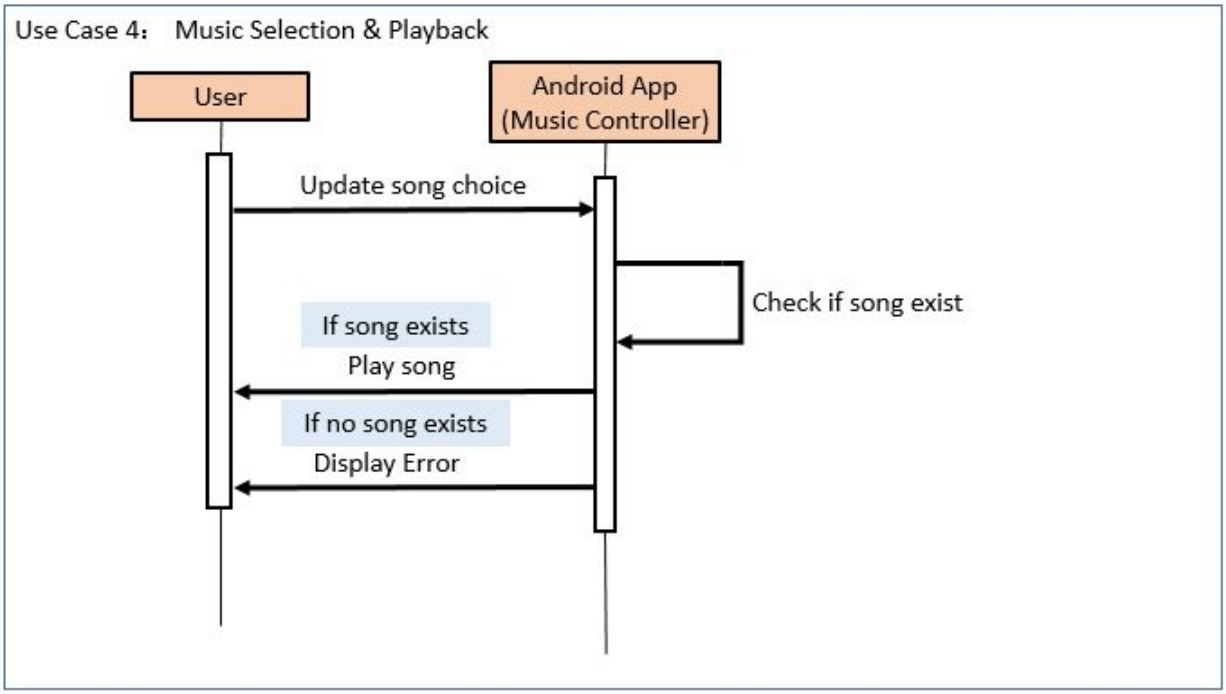
Flow of Events for Extensions: System would display a error page saying music pausing failure based on the error of system.

v. System Sequence Diagrams




Use Case 2: Displaying Heart Rate Data
User
Android App
Database
Request data
Send feedback
If data exists
Display result data
If no data
Display Error


Use Case 3: Visualization of Data through a Graphical Representation
User
Android App
Database
Request data
Send feedback
If result exists
Display result graphs
Generates data
analysis graphs
If failed to get data
Display Error

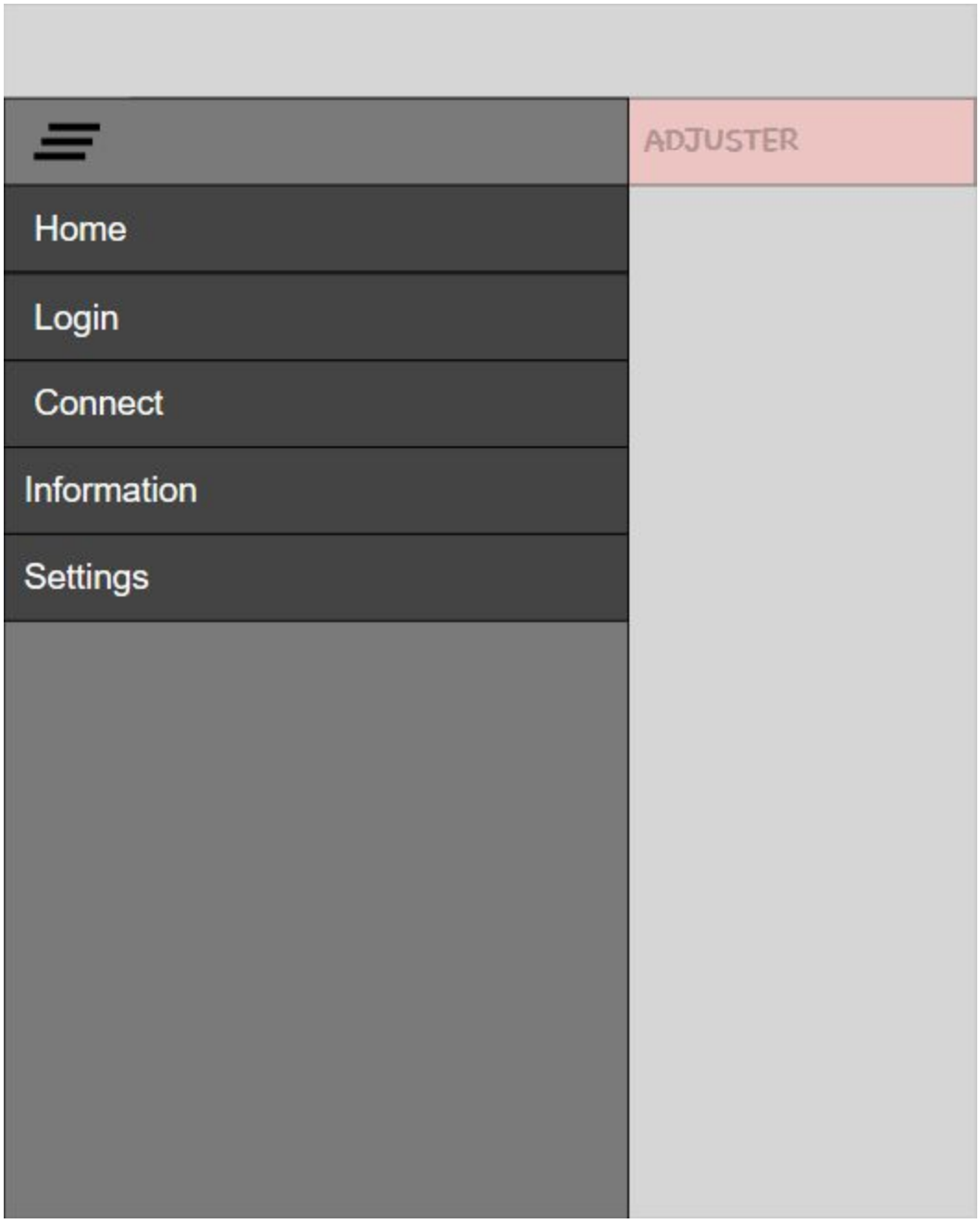

Use Case 4: Music Selection & Playback
User
Android App
(Music Controller)
Update song choice
Check if song exist
If song exists
Play song
If no song exists
Display Error

5. User Interface Specification

5.1 Preliminary Design

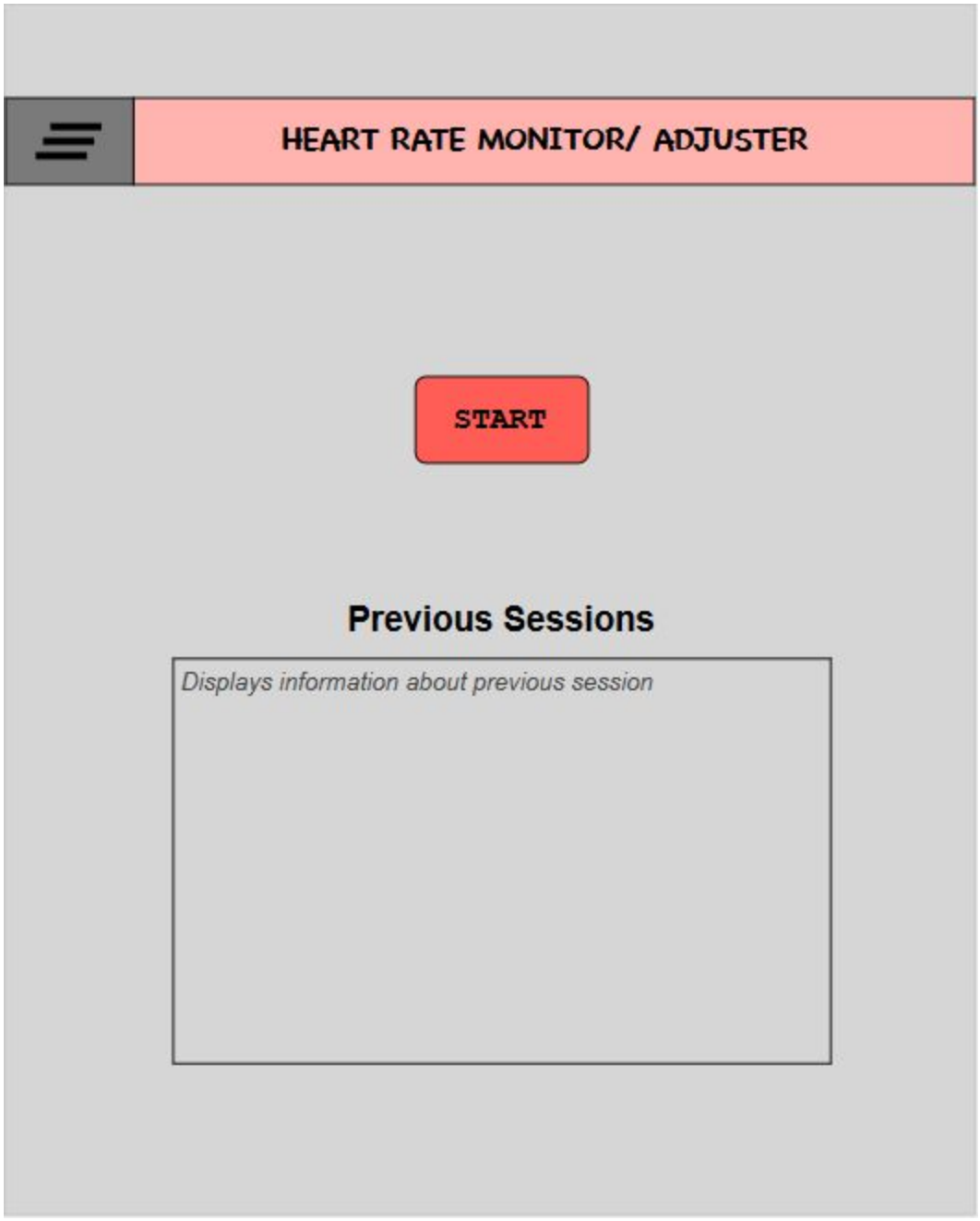
UC#1 Collecting Health Data from Arduino (2 clicks)

1) The user has to click on icon at the top left corner of the screen
2) The user must then click on Connect to connect to Arduino via bluetooth

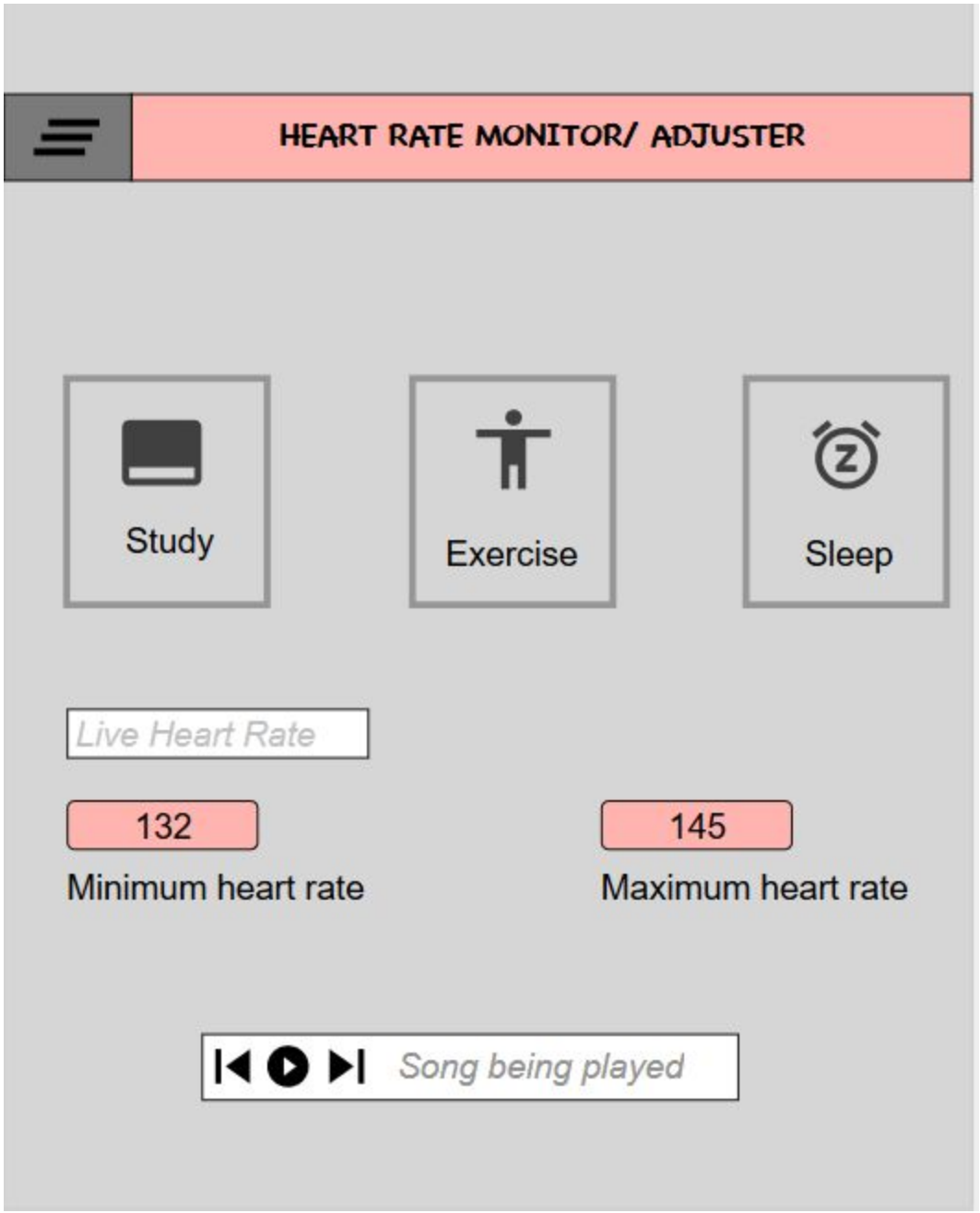


UC#2 Displaying Heart Rate Data (2 clicks)

1) The user must click on START on the home page in the app (1 click)



2) Then he/she must choose the mode they are using the app in (1 click)



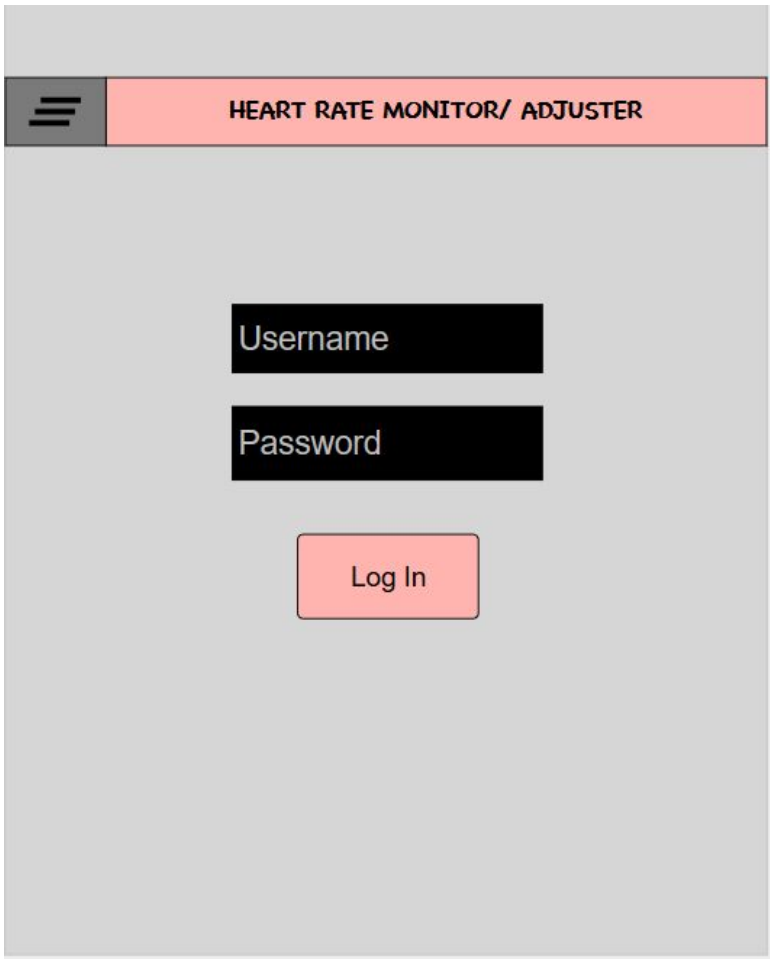
3) Live heart rate will then be displayed in a textView.

UC#3 Visualization of Data through a Graphical Representation

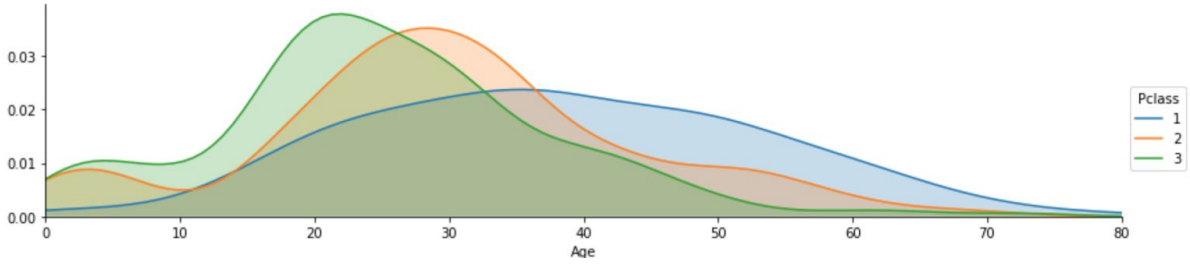
1) The user has to click on icon at the top left corner of the screen. (1 click)

2) Then click on Login from the menu bar. (1 click)

3) The user must then enter their username and password and click Log In. (12-20 keystrokes and 1 click)

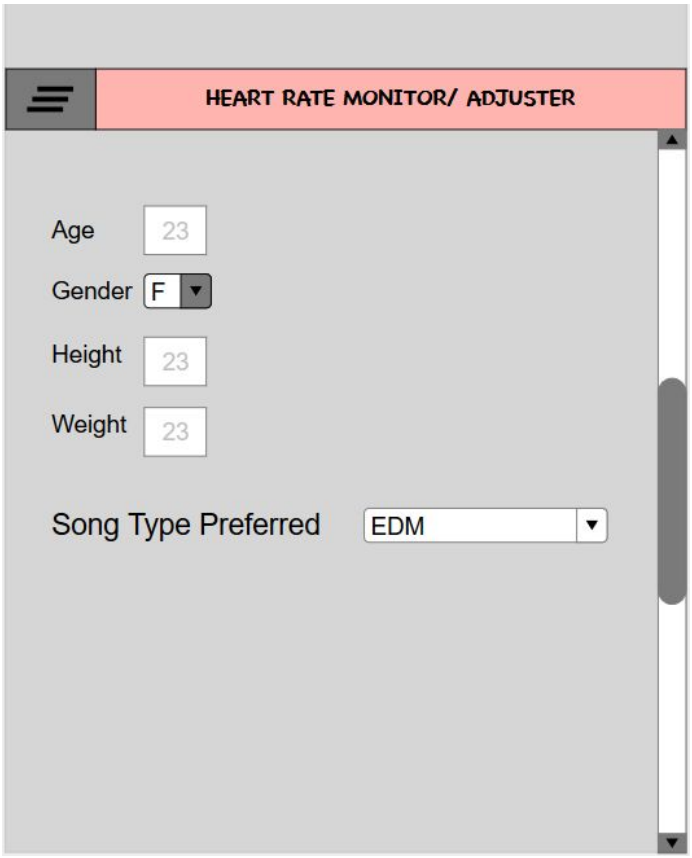


4)The user is then linked to their database and can view their Heart beat Data in a graphical representation. This is a similar graph and the actual graphs has to wait until the data is stored into the database, so it cannot be generated at this moment.



UC#4 Music Selection & Playback and UC#5 Ranking Songs (3 clicks)

1) The user has to click on icon at the top left corner of the screen. (1 click)
2) Then click on Settings from the menu bar. (1 click)
3) The user can then click on Music Type preferred from the scroll bar to play a song and to rank favourite songs. (1 click)



UC#6 Changing Song and UC#7 Pausing Song: (3 clicks)

1) The user gets to the app page that displays live heart rate data by following UC#2 (2 clicks)
2) The bottom of the screen then has the music player being used to adjust heart rate. By clicking the Play/Pause button, music can be toggled on/off (1 click) and by pressing next/prev, the songs can be changed (1 click).

5.2 User Effort Estimation:

1) The User is greeted with a message which appreciates the last successful session with the details with a button which prompts “Continue”. On clicking this button the home activity opens up. (1- click)
2) On the home screen the user has 3 options:
    a) **Start Button:** On one click the application switches to the activity screen where user can select the available activity options(1-click) and start the activity using the play button(1-click).So a total of 3 clicks to start the activity.(3-Clicks)
    b) **History Button:** On just one click the application switches to an activity which displays the previous activities with their respective time period and the mean of the heart-rate within that time period. (1-click).
3) The application drawer has one-click options like Home, settings, information and login.

Of the above mentioned user estimation, roughly 85% of the click actions are for user interface Navigation and the rest is clerical data entry which varies from one user to the other.

6. Domain Analysis

a. Domain Model

i. **Concept definitions:**To analyse the domain model, we first derive domain model concepts and corresponding responsibilities from the formerly defined system use cases. The table below lists all the domain model concepts and corresponding responsibilities.

Responsibility	Type	Concept
Pairing/Communicating with the Heart Rate Sensor	D	HR Manager
Retrieve the sensed data	D	Log Retriever
Musical Playback	D	Music Player
Logging tracks as they are played	D	Tracker
Queueing next track	D	Music Queue
Listen for user input	D	General UI
Recommend Target Heart Rate	D	Range suggester
Setting the Users Heart Rate	D	Rest Setter
Alert the user in case of danger	D	User Alerter
Graphically displaying User's info	D	User UI
Displaying music Information	D	Music UI
Displaying Current Activity Heart Rate Data	D	Workout VIew
Displaying previous Activity Heart Rate Data	D	History View
Graphically displaying relationship between heart rate data, workout data and music data	D	Data analysis View
User data store for Login pairing	K	User Store

Data Store for Activities	K	Activity Store
Data Store for music Metadata	K	Metadata Store
Data Store for Music Files	K	Music Store

ii. Association definitions: Some of the concepts defined above as domain concepts have to work in certain pattern to finish some target. The table below gives the corresponding association definitions based on the defined domain concepts.

Concept Pair	**Association Description**	**Use Case**
User Info↔User data store	System pairing user's login information to allow user to login to system from user info store	data retrieval
music playbacker ↔ metadata store	music playbacker retrieves information about the current track from metadata store	data retrieval
history view ↔ workout store	history view retrieves data about previous workouts from the workout store	data retrieval
track logger ↔ music playbacker	tracks played by music playbacker are logged by track logger	data logging
music playbacker ↔ track queuer	music playbacker retrieves the next track from the track queuer	data retrieval
music playbacker ↔ playback view	playback view displays information based on the data in music playbacker	human data interface
rest setter ↔ HR manager	HRM manager retrieves user's current heart rate to set as resting	human data interface
user alerter ↔ HRM manager	HRM manager retrieves user's current heart rate and activates user alerter if in dangerous levels	human data interface
hrm manager ↔ general UI	general UI pairs and reports hrm status based on hrm manager	human data interface

heart beat view ↔ hrm manager	retrieves and displays heart rate data from the Heart-rate manager	human data interface
log retriever ↔ workout store	log retriever fetches logs from the workout store and used to display the graphs	data logging
music playbacker ↔ music store	music playbacker plays songs from the music store	data retrieval
Data analysis↔music store, workout store and activity store	Retrieves music data, workout data and heart rate data to do the data visualization for analyzing data.	data retrieval

iii. Attribute definitions

Concepts	Attributes	Attribute Definition
User Info	Data logging	Data logging has to with the storage or retrieval of logged data or the logging of data.
HRM Manager		
Log retriever		
Track logger		
Data Analysis		
Music Playbacker	Human data Interface	Human data interfaces deal with the interaction between the user and the data
Track queuer		
Rest setter		
Peak calculator		
General UI		
User Alerter		
Playback View		
Heart beat view		

Workout view		
History View		
User Data Store	Data storage	Data storage deals with the storage of data.
Workout Store		
Metadata Store		
Music Store		

Iv. Traceability matrix

	HRM	Log Retriever	Music Player	Tracker	Music Queue	General UI	Rangesuggester	Rest Setter	User Alerter	User UI	Music UI	Workout VIew	History View	Data analysis View	User Store	Activity Store	Metadata Store	Music Store
UC-1	X		X			X				X					X	X	X	
UC-2		X		X		X		X		X		X	X			X	X	
UC-3		X			X	X	X		X	X		X	X	X		X	X	
UC-4				X			X			X	X						X	
UC-5	X		X			X					X				X			X
UC-6	X		X			X					X				X			X
UC-7	X		X			X					X				X			X

6. Part B: System Operation Contracts

System operation contracts for the operations of the fully-dressed use cases.

Collecting Health Data from Arduino

1. Pre Condition: The system collects the data and transfers it to the database.
2. Post Condition: User will be able to view data.

Displaying Heart Rate Data

1. Pre Condition: The website displays at the Dashboard Page.
2. Post Condition: Users get their health index dashboard by month or day.

Visualization of Data through a Graphical Representation

1. Pre Condition: The website displays at the Dashboard Page.
2. Post Condition: Users get their Heart rate analysis chart by month or day.

Music Selection & Playback

1. Pre Condition: The application shall allow the user to play music.
2. Post Condition: The user will play their music for their current activity.

Ranking Songs

1. Pre Condition: The user has ranked songs within the application.
2. Post Condition: The ranked songs will be available for use.

Changing Songs

1. Pre Condition: The application allows the user to change the music being played.
2. Post Condition: The application changes the music that the user listened to before to broadcast the new music that user selected.

Pausing Songs

1. Pre Condition: This application allows the user to pause the music being played.
2. Post Condition: The application suspends the music being played and does not continue to play any music.

7. Plan of Work

Although we are a team of 5 students working on a project, we have divided ourselves into 3 sub-teams of three, two and one individuals. Each sub-team will be responsible for hardware, software and web development.

The software team will be responsible for the development of the Android application and the user interface. Different activities for the android application consists of the home screen activity, connection screen activity, select mode screen activity, login screen activity, settings screen activity and the information screen activity. After completion of report 3 i.e October 7, we plan on getting the activities working in one week and then preparing the layouts in 2 weeks. By the end of the 2nd week, we plan on getting the software requirements to connect the application to the arduino device.

The hardware team will be responsible for the development and testing of the heart-rate monitor. The heart-rate monitor consists of a pulse sensor, a bluetooth module and an arduino board. Each of these components is independant and requires separate testing. We achieved the preliminary testing of the pulse sensor by October 5. We hope to get the bluetooth module working by October 14. Complete integration with the android app is expected to be done after the 1st demo i.e 31st October.

The web-based database development consists of using the Amazon Relational Database Service. Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. We already have the AWS RDS database which is based on the MySQL. Our database is called “fall2018softenggroup2health”. We have created the user's table using MySQL's commands. This table would store the user's login information. When they set up a new account, the login name, password and email information are all gonna stored in the database. When the existing user logs into our system, the system will automatically match the login information entered by the user with the stored user login information in the database. If so, users will be allowed to log in.

We will create a new table to store user’s heart rate information, but we need to finish setting up Arduino and wait for the android app to be developed. We expect to create this new table set up by November. We will extract the data from the database and export them to a file in CSV format, through which we can use python for data visualization analysis. All these tasks are expected to be complete by mid-November.

We expect to complete the full fledged working of the arduino heart rate device before the first demo i.e October 31, android app integration by the first week of November and database integration by mid November leaving at least two weeks for additional testing and debugging before the final submission.

8. References

[1] Stavros Asimakopoulos , Grigorios Asimakopoulos , and Frank Spillers, Motivation and User Engagement in Fitness Tracking: Heuristics for Mobile Healthcare Wearables
[2] Stefan Koelsch and Lutz Jancke, Music and the heart, European Heart Journal (2015) 36, 3043–3048
[3] Using music to tune the heart URL: https://www.health.harvard.edu/newsletter_article/using-music-to-tune-the-heart
[4] Heart rate change from awake stage to sleep stage URL: http://www.livestrong.com/article/105256-normal-heart-rate-sleeping/
[5] Masao Yaso, Atsuo Nuruki, Seiichi Tsujimura, and Kazutomo Yunokuchi, Detection of REM sleep by heart rate, Proceedings of The First International Workshop on Kansei
[6] Previous project: http://www.ece.rutgers.edu/~marsic/books/SE/projects/HealthMonitor/2014-g12-report3.pdf