

# Final Report

## Problem Statement:

In the past decade, people have been suffering from many health-related issues such as blood pressure, heart attack, and diabetes. A critical part of our measuring is Heart rate. Heart rate is central to this process because the function of the heart is directly related to heart and stroke volume (which means "Cardiac output"). In this project, we aim to measure heart rate and respiratory rate through any android phone. Usually, the normal heart rate is 70 to 120 beats per minute. If it's slower than 70 means bradycardia (slow heart) and if it's high 120 means Tachycardia (fast rate) so it's important to recognize your heart rate daily. For instance, during the pandemic COVID19, everyone is monitoring heart rate. If that is available on the phone it shows how well your heart is pumping blood, reducing the chances of virus-exploring to contact. Similarly, we aim to measure the Respiratory rate per minute which is usually 15 to 30 per minute. (It shows how well your body is working to deliver Oxygen). For respiratory rate put your phone on the chest position to calculate respiratory which shows accurate results. For example 22.655.

In order to achieve the stated goal, we used java to design the application. Specifically, to measure the heart rate and pulse rate we used computer vision techniques. To find time-based differences in the spatial location and/or pressure of ridges on a finger, which variations originate from time-based variations in blood flow, a time series of fingerprint pictures will be studied. If the time-based fluctuations are suggestive of a pulse in one of the person's digits, it may be determined from the analysis what the pulse rate is based on the variations.

This technique does not require data from fingerprint scanners (like Mi or Samsung phone features requirements) or heart rate sensors or any external sensors which are used in smartwatches, which are the alternative options. As our method requires minimum resources to find the heart rate and respiratory rate in order to track human health, we find it more promising.

## Step by Step build/installation instructions:

### Prerequisite:

1. Android Studio is installed and has emulators configured.

### Steps to run the code in Android Studio:

1. Download the folder "Dhake\_smita\_IWS\_final\_project".
2. Open android studio
3. From the file menu select Open

4. In the open file or Project wizard, Select the build.gradle file under “Dhake\_smita\_IWS\_Final\_project\MyApplication\Myapplication”.
5. Click ok.
6. This will load the Gradle project in android studio.
7. Once the project is loaded, select the emulator and click the green triangle to or click (Shift + F10)
8. After the build is done and APK is installed on the emulator, the app will launch on its own.

### Steps to run the application on an android phone:

Download the “[app-debug.apk](#)” on the phone. Give permission to take pictures and record video and the app is ready to use.

### Detailed instructions on how to use the app “COMP 5130: HR Tracker”

1. Download the “[app-debug.apk](#)” on your android phone. You will be asked to give permission to the application to take pictures and record videos. You will be given three options (As shown in Figure 1).
  - a. While using the app
  - b. Only this time
  - c. Don't allowYou can choose either a or b to move ahead with the application.

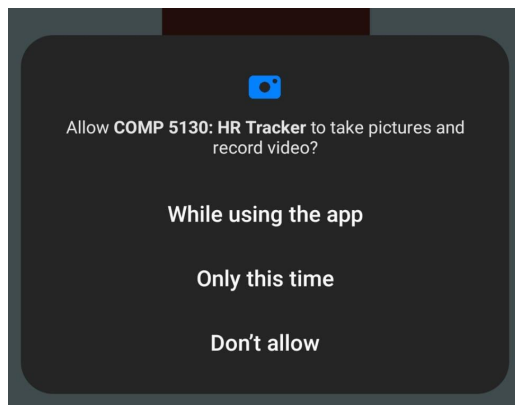
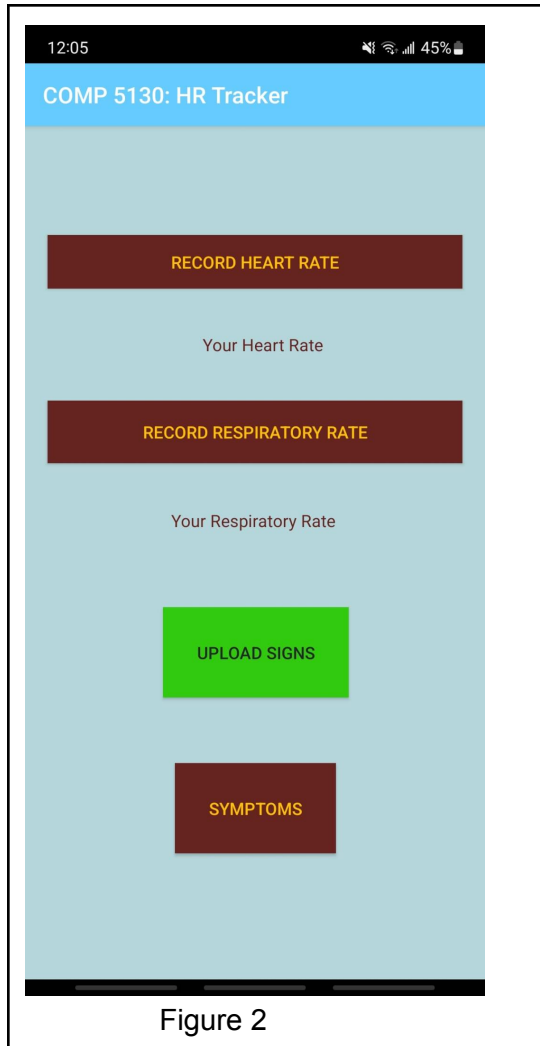


Figure 1

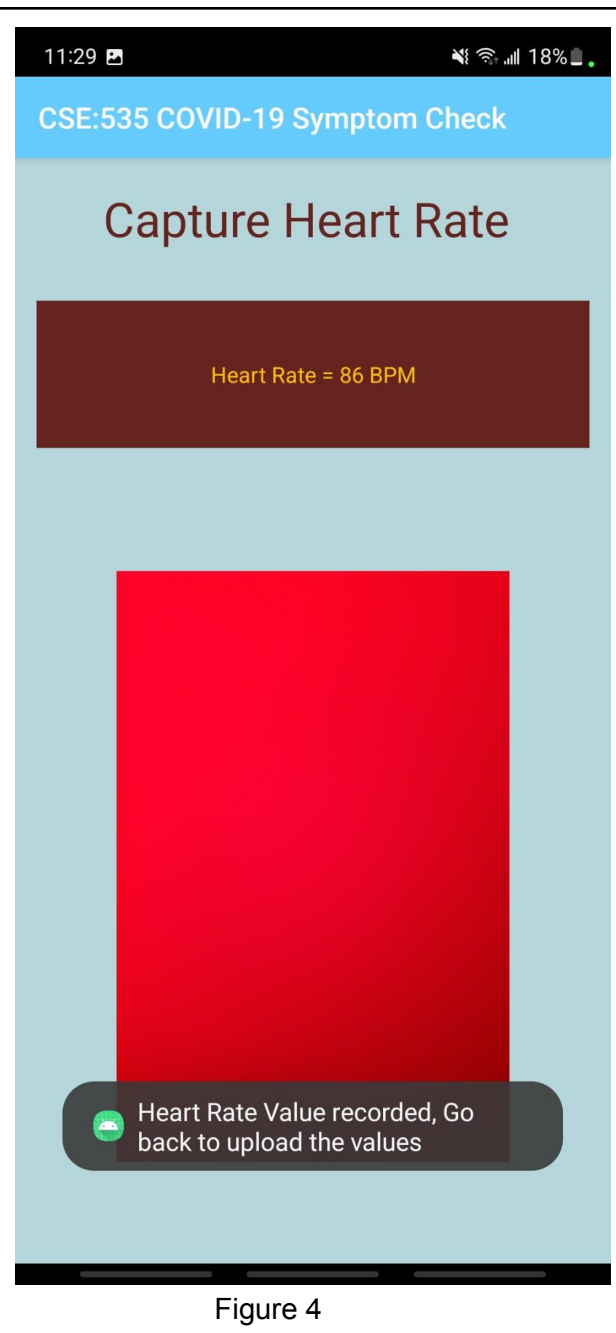
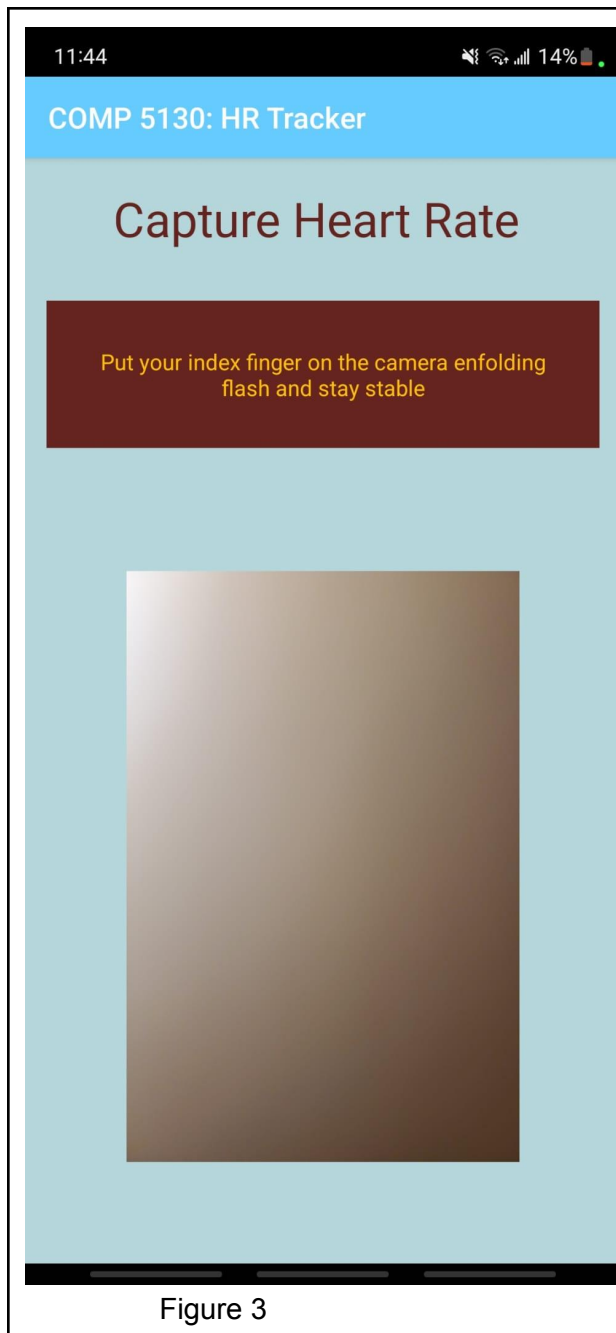
2. Once you are done with step 1, you will be directed to the home page of the application (as shown in Figure 2). You will be able to see 4 buttons in the application.
  - a. **RECORD HEART RATE:** This button will lead to the subsequent page which will help in recording the heart rate data using the phone camera. The resultant data will be shown beside the text “Your Heart Rate”.
  - b. **RECORD RESPIRATORY RATE:** This button will lead to the subsequent page which will help in recording the heart rate data using the phone camera. The resultant data will be shown beside the text “Your Respiratory Rate”.

- c. **SYMPTOMS:** This button will lead to the subsequent page where they can upload any symptoms like fever, cough, etc.
- d. **UPLOAD SIGNS:** You can store the collected data by clicking on this button.



3. By clicking on “RECORD HEART RATE” you will be directed to the page shown in Figure 3. Once the user clicks on this button, the user will be redirected to a new page and the flashlight of the phone will be enabled. On the new page, they will see the instructions for measuring the heart rate which is as follows: “Put your index finger on the camera enfolding flash and stay stable.”

After 45 minutes, the results will be shown saying Heart rate=86 as shown in figure 4.



4. By clicking on "RECORD RESPIRATORY RATE", you will be seeing the message "Your respiratory rate is being recorded". It will take 45 seconds to measure the data.

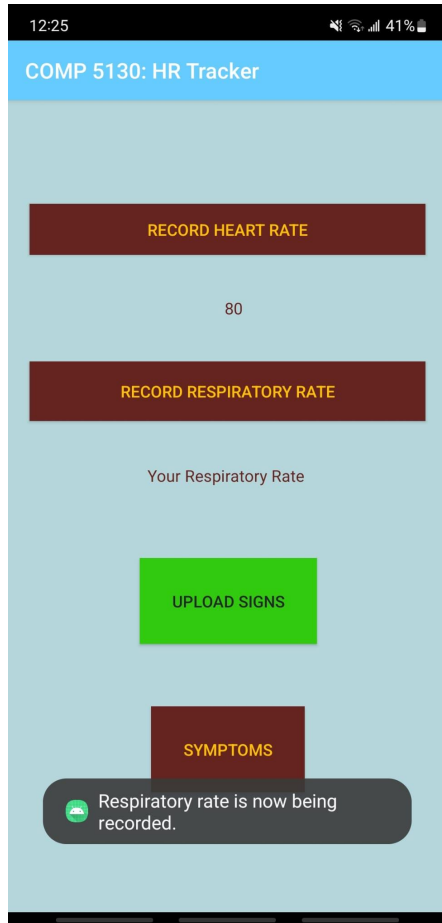


Figure 4

5. By clicking on the SYMPTOMS button, you will be directed to the page shown in Figure 5. On this page, you will be able to select any symptoms like fever, cough, etc. You can rate the severity of each symptom on a scale of 1-5. You can press the stars buttons to do so.

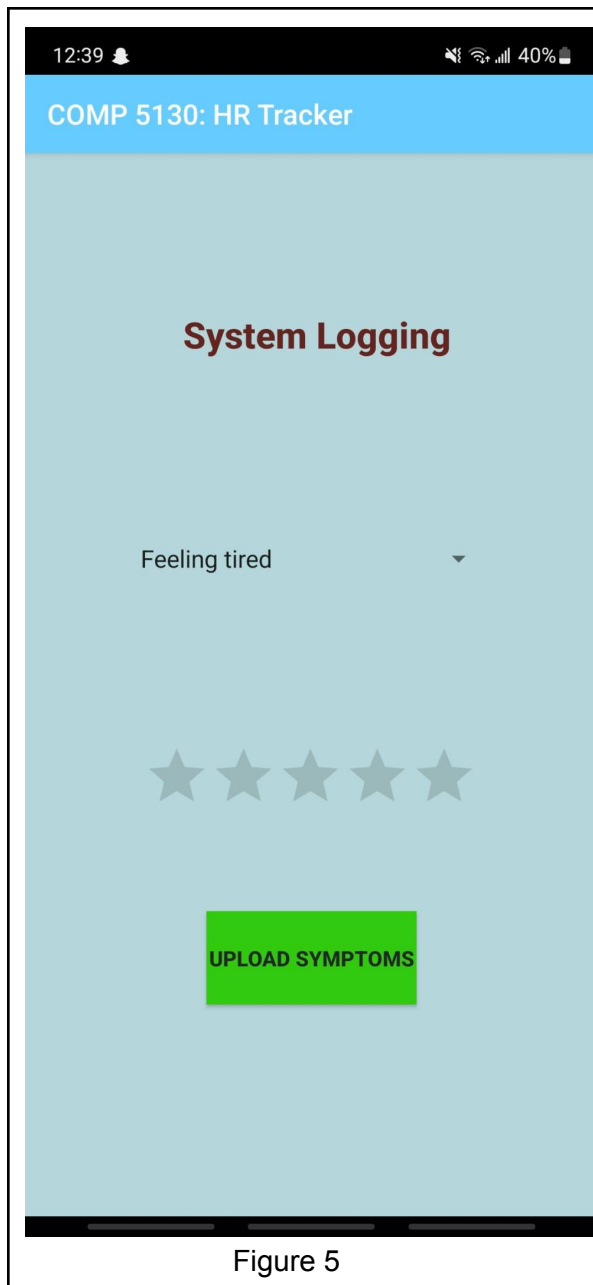


Figure 5

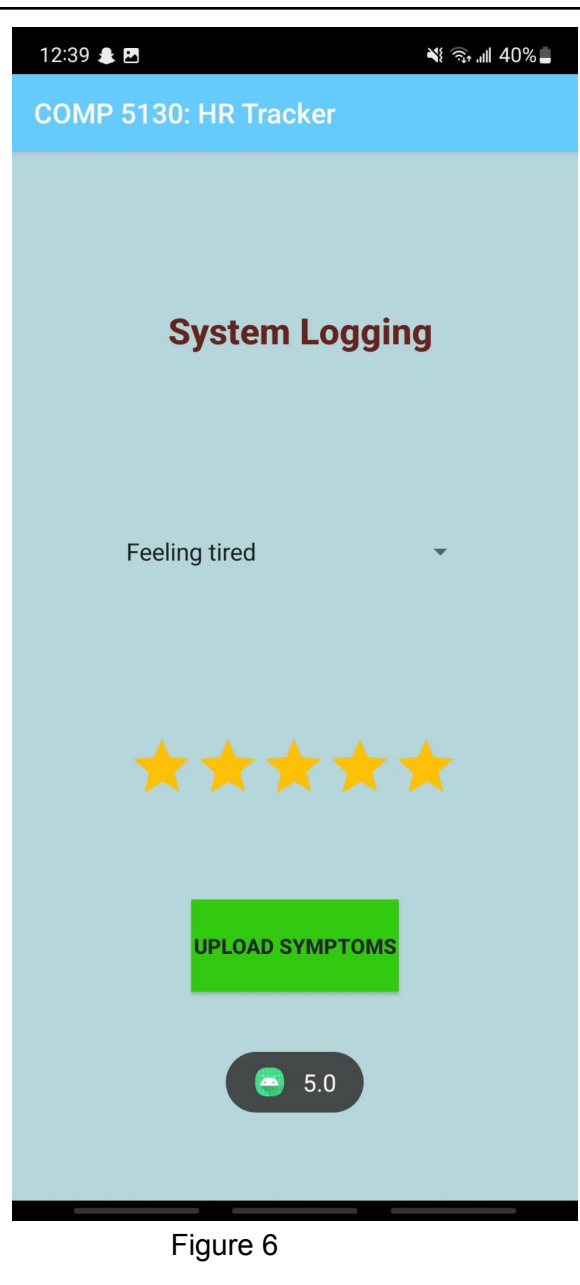


Figure 6

6. When you click on UPLOAD SIGNS, the data will be stored in the database,

**Heart Rate functionality (Developed by Maitry Trivedi):**

I developed the functionality for measuring the heart rate using the phone camera. In order to work with the camera, I used `onSurfaceTextureUpdated()` method where I got the bitmap from

the camera using `textureView.getBitmap()`. At every pulse, there will be a difference in the pixels in the image captured. This data will be used to calculate the heartbeats. Moreover, `Cameradevice.statecallback()` that function I used to callback function is used for receiving updates and defining the camera updates. In addition, To open the device I used `CameraDevice` camera. The calculation for beats per minute is shown in the function `calculateBeatsPerMinute()` in `HeartrateView.java`.

In order to build this functionality, I referred to the following resources [1] and [2].

### Respiratory rate functionality (Developed by Smita Dhake):

In this project, I(Smita) developed specifically the Respiratory rate functionality. The accelerometer is a good sensor to use if you are monitoring device motion. Nowadays Each android powered phone accelerometer is used; it requires very less power compared to other sensors. Motion sensors are used for any movement such as shake and rotation. The user will be asked to lie down and place the smartphone on their chest for a period of 45 seconds. The respiratory rate will be computed from the accelerometer or orientation data. The respiratory rate will be computed from the accelerometer or orientation data.

Accelerometers have been increasingly used in recent years to derive respiration rates. The respiratory rate is measured by counting the number of times your chest or abdomen rises over the course of one minute. In order to implement this feature, the accelerometer sensor was retrieved from the sensor manager. Then, to retrieve the run-time values from the sensor, the sensor was registered through the event listener. The value after 45 seconds will be shown on the screen. For Motion movement, I refer to this link [3].

### Database:( Developed by Maitry and Smita)

Initially, we both(Maitry, Smita) worked on creating the database schema for our application. We figured out how many attributes we need to keep for the database table. We also decided what datatype is required for each attribute. The schema is shown below:

**Tables (2)**

Name	Type	Schema
<b>User_Info</b>		CREATE TABLE User_Info (_id INTEGER PRIMARY KEY, heart_rate TEXT, respiratory_rate TEXT, feeling_tired INT, shortness_of_breath INT, cough INT, loss_of_smell_or_taste INT, muscle_ache INT, fever INT, nausea INT, headache INT, diarrhea INT, soar_throat INT)
_id	INTEGER	"_id" INTEGER
heart_rate	TEXT	"heart_rate" TEXT
respiratory_rate	TEXT	"respiratory_rate" TEXT
feeling_tired	INT	"feeling_tired" INT
shortness_of_breath	INT	"shortness_of_breath" INT
cough	INT	"cough" INT
loss_of_smell_or_taste	INT	"loss_of_smell_or_taste" INT
muscle_ache	INT	"muscle_ache" INT
fever	INT	"fever" INT
nausea	INT	"nausea" INT
headache	INT	"headache" INT
diarrhea	INT	"diarrhea" INT
soar_throat	INT	"soar_throat" INT
<b>android_metadata</b>		CREATE TABLE android_metadata (locale TEXT)
locale	TEXT	"locale" TEXT

We created the database in the file `UserInfoDatabase.java` using the helper class `SQLiteOpenHelper` to manage database creation and version management. All the data regarding the heart rate, respiratory rate, and all symptoms ratings are stored in the database

as shown in the figure. This class makes it easy for ContentProvider implementations to defer opening and upgrading the database, to avoid blocking application startup with long-running database upgrades.

DB Browser for SQLite - C:\Users\trive\Downloads\CSE535\_Assignment1-2\MC-Assignment1\Trivedi-MaitryRonakbhai-IWS\_Final\MaitrySmita.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: User\_Info Filter in any column

_id	heart_rate	respiratory_rate	feeling_tired	ortness_of_brez	cough	_of_smell_or_tx	muscle_ache	fever	nausea	headache	diarrhea	soar_throat
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	194	22.666666	4	0	0	0	0	4	0	0	0	0
2	2145	10.666667	0	0	0	0	0	0	0	0	0	4
3	3115	22.666666	0	4	0	0	0	0	0	0	0	0

The database is stored in the file named MaitrySmita.db

In order to create the database schema we used data types This data type we get from from the website [4]. We Read the description from this website and used it.

#### References:

1. <https://developer.android.com/reference/android/hardware/camera2/CameraDevice.StateCallback>
2. <https://stackoverflow.com/questions/30705322/advice-on-android-textureview-and-bitmap-rendering>
3. [https://developer.android.com/guide/topics/sensors/sensors\\_motion](https://developer.android.com/guide/topics/sensors/sensors_motion)
4. [https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)