Name: Ujjwal Baranwal
Course: CSE 535 Mobile Computing

Context Monitoring App

Abstract

This project involves developing an Android application that monitors vital signs such as heart rate and breathing rate using in-built android sensors. The app measures heart rate with the phone's rear camera and flash, and breathing rate with the accelerometer. In addition, it captures and maintains symptom data in a local database. The integration of these features aims to offer users a convenient and non-invasive method to track their vital signs, potentially aiding in early detection of health issues. This paper discusses the technical approach, design choices, implications & limitations, and the potential impact of this application on personal health management.

Introduction

The rapid advancement of mobile technology has paved the way for innovative applications that leverage the inbuilt sensors of smartphones to monitor and improve health outcomes. This app aims to design and implement a context-aware Android application that captures vital signs such as heart rate and respiratory rate, along with symptom data, using the in-built android's sensors. The primary objective is to provide users with a comprehensive tool for health monitoring, which can be particularly beneficial for individuals with chronic conditions or those who require regular health check-ups.

The project involves developing an Android application that can measure heart rate using the smartphone's back camera and flash, and respiratory rate using the accelerometer or orientation sensor. The application will also collect symptom data and store all the information in a local database, ensuring that users have easy access to their health records.

The heart rate measurement is achieved by capturing a 45-second video of the user's index finger placed on the smartphone's camera lens with the flash enabled. The variation in red coloration in the video is used to derive the heart rate. For respiratory rate measurement, the user is asked to lay down with the smartphone placed on their chest for 45 seconds, during which the accelerometer or orientation sensor data is used to compute the respiratory rate.

The application also includes a feature for logging symptoms, where users can select symptoms from a predefined list and rate them on a scale of 1 to 5. This data, along with the vital signs, is stored in a local database using RoomDB or RealmDB. The project would emphasizes the importance of data storage and user feedback mechanisms, which are crucial for developing advanced mobile computing applications.

Technical Approach

Data Collection

Heart Rate

The *captureVideo* method is used to record a video and calculate the heart rate from that video. When the method is called, it starts a video recording session with *videoCapture* and hides the "Next" button while showing a "Calculating..." status. The method uses the device camera, enabling the torch (flashlight) for better lighting, and handles the start/stop of the recording session. If an ongoing recording exists, it stops the session; otherwise, it creates a new recording with appropriate media storage options.

Once the recording finalizes (after 45 seconds countdown), the recorded video is processed to calculate the heart rate. The *heartRateCalculator* function is invoked with the video's URI to extract the heart rate data from the video content. When the heart rate is calculated, it is displayed on the screen, the status is reset to "Health App," and the data is stored in the Intent to be sent to the Symptoms Activity.

Respiratory Rate

The respiratory rate is calculated by collecting accelerometer data for 45 seconds using the accelerometer sensor. During this time, the X, Y, and Z axis values are continuously accumulated in separate arrays. A countdown timer runs in parallel, and when the 45-second period elapses, the sensor is unregistered, and the collected data is passed to a function called *respiratoryRateCalculator*. This function processes the accelerometer data and calculates the respiratory rate. Finally, the calculated rate is displayed on the user interface, and the data is stored in the Intent to be sent to the Symptoms Activity.

Symptoms

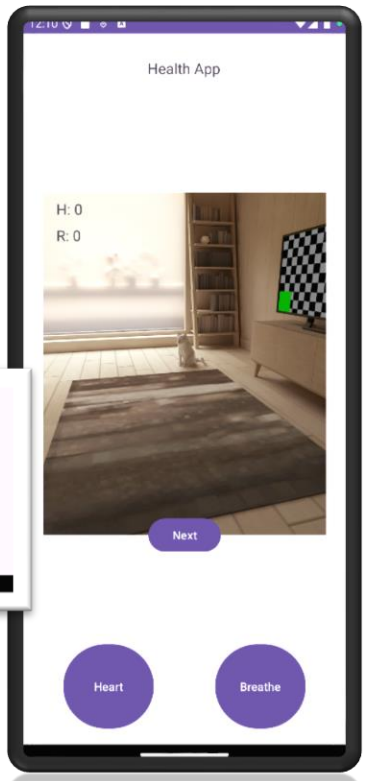
To register all the symptoms, the Heart Rate and the Respiratory Rate are imported from the MainActivity, and 10 other symptoms are measured based on user's compliance. These are then paired together and stored in a table in a database (Room DB)

	id	heartRate	respiratoryR...	nausea	headache
1	1	0.0	0.0	4	2
2	2	0.0	0.0	1	2
3	3	72.5	18.0	4	2
4	4	72.5	18.0	5	0

Design Choices

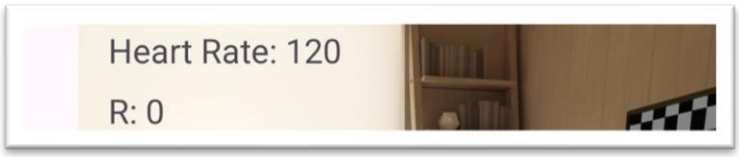
Home Page

The home page (initial screen after the app is opened and the permissions are granted) is designed in a way that allows users to easily record their heart rate and respiratory rate and displays the results on the top left of the preview view. The Next button is hidden programmatically when the user is calculating the heart or the respiratory rate as it can potentially disrupt the flow of the program.



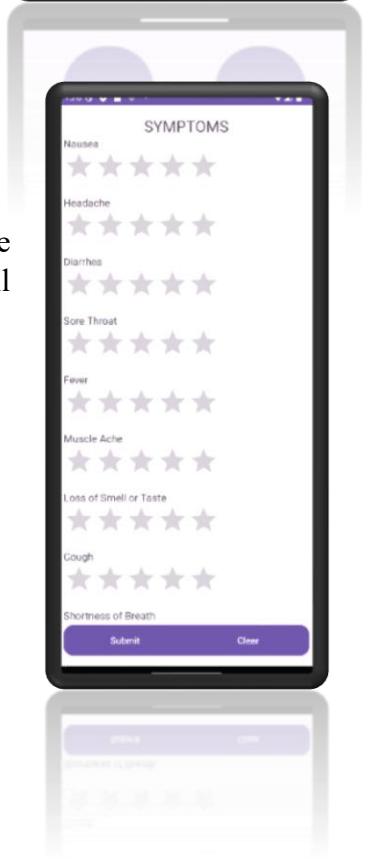
Once the button is pressed, it starts a counter of 45 seconds to record data and then sends the data to respective helper functions to calculate relevant heart rate and respiratory rate.

.



Symptoms Page

The symptoms page allows the user to scroll and mark the symptoms as they are experiencing it. It gives multiple options from a severity of 0 to 5. The submit button would upload the collected data to the Database and the Clear button would clear all selected symptoms.



Database

	id	heartRate	respiratoryR...	nausea	headache	diarrhea
1	1	0.0	0.0	4	2	5
2	2	0.0	0.0	1	2	3
3	3	72.5	18.0	4	2	3
4	4	72.5	18.0	5	0	0

Implications & Limitations

Implications

1. Health Monitoring: The app provides a convenient way for users to monitor their vital signs, such as heart rate and respiratory rate, using their smartphones. This can be particularly beneficial for individuals with chronic health conditions who need regular monitoring.
2. Data-Driven Insights: By collecting and storing symptom data along with vital signs, the app can help users and healthcare providers gain insights into the user's health trends over time. This can lead to more informed decisions regarding treatment and lifestyle changes.
3. Accessibility: The app leverages the inbuilt sensors of smartphones, making health monitoring accessible to a broader audience without the need for specialized medical equipment. This democratizes health monitoring and can be especially useful in remote or underserved areas.
4. Research and Development: The data collected by the app can be valuable for research purposes. Researchers can use the anonymized data to study patterns and correlations between symptoms and vital signs, potentially leading to new discoveries in health and wellness.

Limitations

1. Accuracy: The accuracy of the heart rate and respiratory rate measurements may be affected by various factors, such as the user's movement, lighting conditions, and the quality of the smartphone's sensors. This could lead to less reliable data compared to medical-grade equipment.
2. User Compliance: The effectiveness of the app relies on user compliance. Users need to follow the instructions correctly for accurate measurements. Any deviation from the prescribed method can result in inaccurate data.
3. Data Privacy: Storing sensitive health data on a smartphone raises concerns about data privacy and security. Ensuring that the data is securely stored and transmitted is crucial to protect user privacy.
4. Limited Scope: The app focuses on a specific set of vital signs and symptoms. It may not cover all the health parameters that a user or healthcare provider might need. Additionally, the app's functionality is limited to the capabilities of the smartphone's sensors.

Suggestions for Future Work

To improve the app, I'd look at using machine learning algorithms that can flag potential health risks based on collected data. Implementing real-time data analysis and cloud storage can help to improve accessibility and scalability. Furthermore, expanding the app's vital sign and symptom tracking capabilities can give a comprehensive health monitoring solution. Including user feedback systems and personalized health recommendations can help to engage users and promote wellness. Finally, because the app contains sensitive health information, data privacy and security must be maintained.

Links

GitHub: PipKcK/Context-Monitoring-App-535 (github.com)

YouTube: https://youtu.be/ajHBjoUeZhU

References

Milazzo, J., Bagade, P., Banerjee, A., & Gupta, S. K. S. (2013). bHealthy: A physiological feedback-based mobile wellness application suite. **Wireless Health 2013**, 1-3 November, Baltimore, MD, USA. ACM.

Banerjee, A., Verma, S., Bagade, P., & Gupta, S. K. S. (2012). Health-Dev: Model based development of pervasive health monitoring systems. **2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks**. Arizona State University, Tempe, Arizona. https://doi.org/10.1109/BSN.2012.33

CameraX Overview: Android Developers. (2024, January 5). CameraX overview. Android Developers. Retrieved from https://developer.android.com/media/camera/camerax

Room Persistence Library: Android Developers. (2024). Save data in a local database using Room. Android Developers. Retrieved from https://developer.android.com/training/data-storage/room/

Lackner, P. (2023, July 19). *Content Providers - Android Basics 2023*. YouTube. https://youtu.be/IVHZpTyVOxU

Lackner, P. (2023, March 15). *The FULL Beginner Guide for Room in Android | Local Database Tutorial for Android*. YouTube. https://youtu.be/bOd3wO0uFr8

OpenAI. (2024). *ChatGPT* [Large language model]. https://chat.openai.com/chat