Software Engineering Department
ORT Braude College

Capstone Project Phase 1 – 61998

Heart Rate Monitoring using Machine Learning

24-1-R-8

By:
Ali Mohsen
Mohammad Khamaisy

Supervisor
Zeev Frenkel

https://github.com/MiNEWGIT/FinalProject

2024

Table of content

Abstract. In contemporary society, there's a noticeable rise in individuals monitoring their heart rate for health insights, driven by increased awareness of cardiovascular health and technological advancements in wearables and health apps. This trend reflects a shift towards proactive self-care and a desire for personalized health management. However, current tools for automatically detecting abnormalities in heart rate data are limited. This research aims to address this gap by comparing automated machine learning techniques for identifying anomalies in heart rate time series from consumer wearables. The findings suggest promising methods for robust anomaly detection, which could be integrated into existing health monitoring apps to provide real-time alerts and improve outcomes through early intervention.

1. Introduction

Cardiovascular diseases (CVDs) are the leading cause of death globally, taking an estimated 17.9 million lives each year [17]. CVDs refer to a range of conditions affecting the heart and blood vessels, including heart attacks, strokes, and hypertension. They result from a combination of genetic factors and lifestyle choices such as diet, exercise, and smoking [17]. CVDs pose a substantial global health threat, emphasizing the need for effective prevention and management measures [17].

Early detection of cardiac abnormalities can greatly improve patient outcomes through preventative care and early intervention. The rapid adoption of smartwatches and mobile health apps provides a unique opportunity to analyze heart rate data at scale for signs of arrhythmias and other potential issues. However, the automated algorithms in most consumer devices today provide limited capabilities for detecting abnormalities from heart rate data. Our project aims to develop and evaluate machine learning techniques to more reliably identify anomalous patterns in heart rate signals gathered by smart devices.

Many existing heart rate monitoring apps and devices fail to account for individual variations and instead rely on generic thresholds or population-level data, potentially compromising their accuracy and personalization capabilities [19]. Some proprietary research devices use more advanced electrocardiogram sensors (EKG): (An electrocardiogram is a diagnostic test that records the electrical activity of the heart over a period of time) and artificial intelligence (AI) based algorithms [20]. Unfortunately, these are not accessible in many cases, there is an unmet need for accurate, automated detection of putative abnormalities that could be widely deployed through consumer smart device apps and wearables.

Within our project we develop an application (app) aimed to monitor the real time process of heart rate data. The app is designed to cater to individuals of all ages and health statuses, offering features aimed at promoting overall wellness. Beyond being solely for patients or older demographics, it provides tools for monitoring heart rate during exercise, tracking daily activity levels, and receiving personalized insights for maintaining a healthy lifestyle. By encouraging proactive engagement in health management, the app empowers users to take control of their well-being, regardless of age or medical condition.

Our project will leverage advanced machine learning algorithms and technologies for accurate and personalized anomaly detection. Deep learning techniques like Long Short-Term Memory (LSTM) [21] networks and Convolutional Neural Networks (CNNs) [22] will be employed to learn and model complex patterns in heart rate data. Unsupervised anomaly detection algorithms, such as Isolation Forests [23] and One-Class Support Vector Machines (OC-SVMs) [24], will be used to identify deviations from expected heart rate patterns without relying on labeled data. These algorithms will adapt to individual users' heart rate profiles, enabling personalized anomaly detection. The application will integrate with consumer wearable devices like SmartWatch or SmartPhones for real-time heart rate data collection and preprocessing [25]. The trained machine learning models will analyze the data, detect potential anomalies, and provide users with timely alerts and insights. By combining cutting-edge machine learning algorithms, personalized modeling, and integration with consumer wearables, the project aims to deliver an accurate and accessible solution for automated detection of cardiac abnormalities, empowering users to proactively manage their cardiovascular health.

2. Related Works

2.1 Heart rate applications and machine learning

Heart rate monitoring has become increasingly popular across various domains, including health, wellness, and performance optimization, with applications extending to stress management, fitness tracking, and medical screening. Consumer technology advancements have made heart rate monitoring accessible through smartphone applications. Studies support the efficacy of heart rate monitoring in optimizing cardiovascular fitness and detecting irregular heart rhythms such as atrial fibrillation, prompting early medical intervention [17]. Additionally, heart rate variability derived from monitoring serves as a valuable indicator of stress levels and overall wellness [12]. Leveraging machine learning algorithms in Heart rate applications enhances accuracy and predictive capabilities in heart rate monitoring, showcasing innovative approaches to improving healthcare outcomes through technology integration [13]. Some advantages and disadvantages of existing technologies are summarized in Table 1.

Advantages	Disadvantages
The heart rate monitoring apps can enhance healthcare delivery by handling multiple queries and requirements at a time, improving efficiency and speed	Some apps claim to measure heart rate and other vital signs using the smartphone's camera, but the accuracy of these measurements can be questionable
The use of heart rate monitoring apps can	The battery life and charging requirements of

significantly reduce the costs associated with healthcare consultations	wearable devices used for heart rate monitoring can be a limitation for continuous use
The apps are accessible anytime, anywhere, and can provide continuous heart rate monitoring through wearable devices	The wireless communication technologies used in wearable devices can potentially pose health risks due to the emission of radio waves
The apps can motivate users to live a healthier lifestyle by tracking their heart rate and providing personalized insights	For medical applications, heart rate monitoring apps may need to undergo rigorous testing and validation to ensure compliance with relevant healthcare regulations

Table 1. The main advantages and disadvantages of existing technologies for heart rate monitoring.

Our app aims to excel in several key ways. Firstly, it employs machine learning algorithms to personalize monitoring and anomaly detection, learning the user's baseline heart rate patterns and detecting deviations accurately using advanced techniques like Isolation Forests and OC-SVMs [24]. Additionally, it provides comprehensive data tracking and visualization features, allowing users to understand their heart rate trends and identify correlations with other factors. These combined features make the described app a personalized and comprehensive solution compared to other existing heart rate apps.

2.2 Algorithms used in the existing applications

The applications mentioned above leverage sophisticated algorithms, predominantly machine learning (ML) techniques, to interpret and analyze heart rate data. These algorithms are designed to detect patterns, anomalies, and correlations within the data, enabling accurate identification of various cardiovascular conditions. Various algorithms are employed within these applications for heart rate monitoring: Deep Neural Networks [26], such as convolutional neural networks, analyze raw heart rate data to detect patterns; Recurrent Neural Networks (RNNs) [27] like LSTMs interpret sequential heart rate data; Ensemble methods like Random Forests classify heart rate data effectively; Support Vector Machines (SVMs) [28] categorize heart beats as normal or abnormal; K-Nearest Neighbors (KNN) classify heart rate samples based on similarity; Clustering Algorithms like K-means segment heart rate data; Regression models track trends in heart rate over time; Signal Processing techniques process heart rate signals; Neural Style Transfer synthesizes realistic heart rate examples; and Active Learning minimizes labeling effort by dynamically selecting useful training samples. These algorithms collectively enable accurate interpretation and analysis of heart rate data for various health monitoring purposes.

In the next sections We will provide a short review of these algorithms and how we use them in our app such as Time Series Forecasting and Anomaly Detection.

3. Background

In this section we provide a short review of Electrocardiography (EKG), popular heart rate monitoring applications, their algorithms and acuracy.

3.1 Electrocardiography (EKG)

The electrocardiogram (EKG or ECG) is a vital diagnostic tool used to measure and record the heart's electrical activity by placing electrodes on the patient's chest, arms, and legs. This machine swiftly detects heart abnormalities such as arrhythmias and heart attacks with high accuracy, exceeding 90% sensitivity and specificity. Widely employed in healthcare settings, EKGs play a pivotal role in clinical evaluations, aiding treatment decisions and monitoring patients with known heart conditions. During each heartbeat, a healthy heart exhibits an orderly progression of depolarization waves triggered by cells in the sinoatrial node, spreading throughout the atrium and ventricles. Typically, more than two pairs of electrodes, known as leads, are used to detect these signals, with 12 electrodes (commonly used across the chest, arms, and legs). ECG interpretation involves analyzing the graph generated by the ECG machine [6,9,10],(see Figs. 1 and 2).

The baseline, or isoelectric line, serves as a reference point. The QRS complex, comprising the Q, R, and S waves, represents a single event, with the Q wave as a downward deflection, followed by the upward R wave and downward S wave (see Fig. 1). Modern ECG monitors digitize these signals using filters for processing (Fig.2).

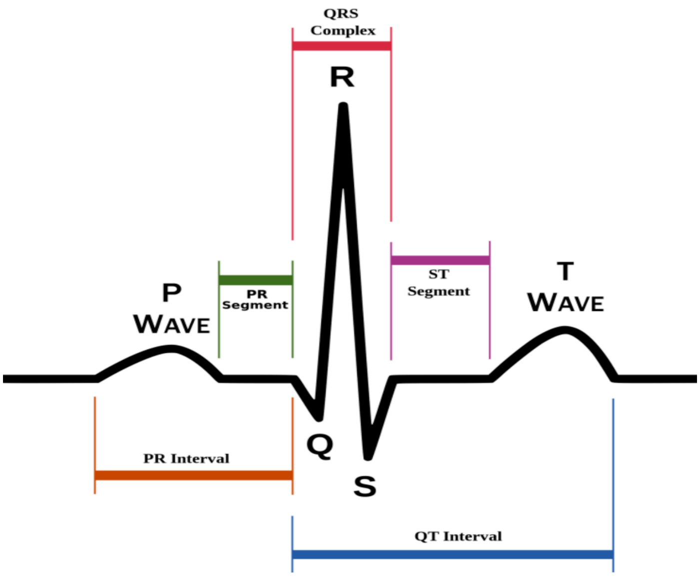


Figure 1. Typical ECG signal. Time and voltage are plotted on the x-axis and y-axis, respectively. The letters used in the ECG signal description do not have abbreviations in medical terminology (taken from The ECG Signal Compression Using an Efficient Algorithm Based on the DWT. (2016).

International Journal of Advanced Computer Science and Applications, 7(3). https://doi.org/10.14569/IJACSA.2016.070325).

• The P wave represents the atrial contractions.

• The QRS complex represents the ventricular contractions. The R peak indicates a heartbeat.

• The T wave is the last common wave in an ECG. This electrical signal is produced when the ventricles are repolarizing

3.2 Cardiovascular diseases

Also known as (CVDs) are a group of disorders involving the heart and blood vessels, including conditions such as coronary heart disease, stroke, and heart failure. According to the World Health Organization, CVDs are the leading cause of death globally, responsible for an estimated 17.9 million deaths annually . These diseases often result from a combination of risk factors, including unhealthy diet, physical inactivity, tobacco use, and genetic predisposition.

Heart rate and heart rate variability can serve as important indicators of cardiovascular health and potential CVD risk. Abnormal heart rate patterns, such as arrhythmias (irregular heartbeats) or consistently elevated resting heart rates, may signal underlying cardiovascular issues or increased risk of adverse events like heart attacks or strokes. Monitoring and analyzing heart rate data can aid in the early detection of such abnormalities, enabling timely medical intervention and treatment. Additionally, studies have shown that higher resting heart rates are associated with an increased risk of cardiovascular events and mortality . Therefore, tracking and interpreting heart rate patterns through wearable devices or mobile applications can provide valuable insights into an individual's cardiovascular health and risk of developing or experiencing complications from CVDs [33].

For example, people with coronary artery disease or heart failure may have an elevated resting heart rate, as the heart has to work harder to pump blood effectively. Conversely, those with certain types of arrhythmias, such as bradycardia (slow heart rate), may have a lower-than-normal resting heart rate [34].

During physical activity or exercise, individuals with cardiovascular diseases may also experience an abnormal heart rate response, either failing to increase adequately or exhibiting an exaggerated increase compared to healthy individuals [34].

3.3 The accuracy of smartphone-based heart rate monitoring apps

The accuracy of smartphone-based heart rate monitoring apps has garnered attention in recent years as consumers increasingly turn to these apps for health and fitness tracking. Studies assessing the accuracy of these apps have yielded mixed results. One study in the European Journal of Preventive Cardiology revealed significant disparities in accuracy among four randomly selected apps, with non-contact apps performing less reliably than contact-based ones [18]. Another study, published in the International Journal of Exercise Science, found a strong correlation between a Polar heart rate monitor and an electrocardiogram (ECG) during exercise, but not with smartphone-based apps like Runtastic Heart Rate Monitor and Instant Heart Rate [3]. Similarly, a study in the ResearchGate journal indicated that while some apps accurately measured heart rate at rest, accuracy declined during physical activity. These findings underscore the need for further research to identify factors contributing to app inaccuracy, suggesting caution in relying solely on smartphone apps for medical purposes and advocating for the use of clinically validated devices for accurate heart rate monitoring.

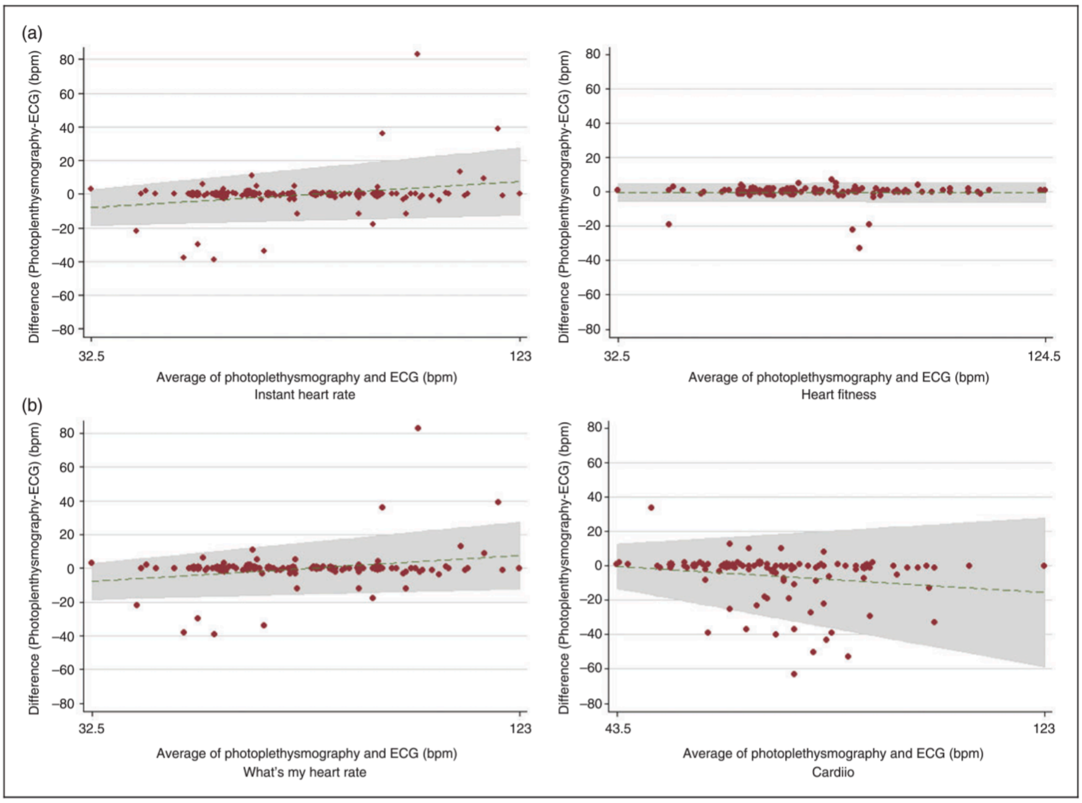


Figure 2. The Bland-Altman plots show considerable differences between the tested apps. Non-contact PPG measurements performed significantly worse compared to contact measurements. Both non-contact PPG based apps performed significantly worse at higher heart rates. They also have a tendency to underestimate higher heart rates. (a) Contact photoplethysmography; (b) non-contact photoplethysmography. (taken from European Journal of Preventive Cardiology [18]).

3.4 The Evolution of Machine Learning Applications

Machine learning is a subfield of artificial intelligence that involves developing computer systems capable of learning and improving from experience without being explicitly programmed. The history of machine learning can be traced back to the 1950s, with the introduction of the perceptron algorithm by Frank Rosenblatt . Over the decades, machine learning has evolved significantly, with key developments including the rise of neural networks and backpropagation in the 1980s , and the resurgence of deep learning in the late 2000s [13].

Today, machine learning algorithms are widely used in various applications, from image and speech recognition to natural language processing, predictive analytics, and autonomous systems. The core principle behind machine learning is the ability of algorithms to learn from data and make predictions or decisions without being explicitly programmed. Machine learning models are trained on large datasets, using techniques such as supervised learning, unsupervised learning, and reinforcement learning, to identify patterns and relationships within the data. These models can then be applied to new, unseen data to make predictions or decisions, continuously improving their performance through iterative learning [13].

The widespread adoption of machine learning is driven by its ability to tackle complex problems, automate decision-making, and generate insights from vast amounts of data. Machine learning is used in a wide range of industries, including healthcare, finance, e-commerce, transportation, and cybersecurity, to improve decision-making, optimize processes, and enhance user experiences.

3.5 Popular Machine Learning Software Tools

Popular machine learning software tools like TensorFlow, PyTorch, and Scikit-learn offer diverse capabilities for constructing and deploying machine learning models. TensorFlow, developed by Google, excels in deep learning applications, supporting a wide range of algorithms and finding extensive use in image recognition and natural language processing. PyTorch, developed by Facebook's AI Research lab, prioritizes flexibility and ease of use, particularly in computer vision and natural language processing research. Scikit-learn, a Python library, provides a user-friendly interface and a broad array of supervised and unsupervised learning algorithms, widely applied across industries like finance and healthcare.

In the field of heart rate monitoring, some applications utilize machine learning techniques to analyze heart rate and physiological data obtained through photoplethysmography (PPG), accelerometers, and electrocardiograms (ECG). However, the provided information about specific applications like Cardiio, Empatica E4, and VivaLNK Vital Scout using techniques such as convolutional neural networks, support vector machines, and decision trees could not be verified from the given search results [29-32].

(The information for 3.3,3.4 and 3.5 sections are a combination mainly from references 29-32).

3.6 Machine learning algorithms to be used in our application

Machine learning algorithms are increasingly being used in heart rate monitoring apps and devices to improve accuracy and enable advanced features. Now we are going to provide a general overview of various machine learning algorithms and techniques. Subsequently,in the next sections we will connect these concepts and elucidate how we intend to integrate and leverage these algorithms synergistically to construct our innovative heart rate monitoring application.(the pseudo-code for all the algorithms presented in this book appears in the project folder in GitHub (https://github.com/MiNEWGIT/FinalProject)).

1- Time Series Forecasting

Time series forecasting algorithms aim to predict future values based on past observations in a time-ordered sequence like heart rate. Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNNs) are popular algorithms for time series forecasting due to their ability to capture temporal dependencies in data. One of the commonly used algorithms for time series forecasting is Autoregressive Integrated Moving Average (ARIMA). Let's break down how ARIMA works with equations:

- Autoregression (AR):

Autoregression refers to predicting a future value in the time series based on past values. It assumes that the linear combination of past observations can predict the future value.The autoregressive (AR) component of ARIMA is represented by the equation:

$$Xt = c + \sum_{i=1}^{p} \phi i Xt - i + \varepsilon t$$

Xt is the value of the time series at time t, c is a constant, p is the order of the autoregressive component, indicating how many past values are used for prediction, ϕi are the coefficients of the autoregressive terms, εt is the error term at time.

- Moving Average (MA):

Moving average refers to predicting the future value of the time series based on the linear combination of past forecast errors.The moving average (MA) component of ARIMA is represented by the equation:

$$Xt = \mu + \varepsilon t + \sum_{i=1}^{q} \theta i * \varepsilon t - i$$

Xt is the value of the time series at time t, q is the order of the moving average component, indicating how many past forecast errors are used for prediction, θi are the coefficients of the moving average terms, εt is the error term at time t

- Integration (I):

Integration refers to differencing the time series to make it stationary, i.e., removing trends and seasonality.The integration (I) component of ARIMA is represented by the differencing operator ∇
▽ Xt = Xt - Xt-1
Combining these components, the ARIMA model of order (p, d, q) is given by:

$$(1 - \sum_{i=1}^{p} \phi i L^{i})(1 - L^{d})Xt = (1 + \sum_{i=1}^{q} \theta i\, L^{i})\varepsilon i$$

L is the lag operator, which shifts the time series backward by one time step, d is the order of differencing, indicating how many times differencing is required to make the time series stationary, εt is the error term at time t, assumed to be white noise.

2- Anomaly Detection

Anomaly detection algorithms aim to identify data points that deviate significantly from the norm in a dataset. One common approach is through statistical methods, particularly Gaussian Mixture Models (GMMs). Let's delve into how GMM-based anomaly detection works with equations:

Gaussian Mixture Models (GMMs):

GMMs assume that the data is generated by a mixture of several Gaussian distributions. Anomalies are identified as data points with low probability of being generated by any of these distributions.The probability density function (PDF) of a GMM is given by:

$$\mathrm{f(x)}= \sum_{k=1}^{K} \pi k.\, N(x|\mu k, \Sigma k)$$

where: f(x) is the PDF of the GMM. πk is the weight of the k-th Gaussian component, representing the probability of selecting that component. N(x|μk, Σk) is the Gaussian distribution with mean and covariance Σk. K is the number of Gaussian components.

- Anomaly Score Calculation:

Anomaly detection involves calculating an anomaly score for each data point, indicating its deviation from the expected distribution. A common metric for this is the Mahalanobis distance, which measures the distance of a data point from the centroid of the distribution in units of standard deviation.The Mahalanobis distance
D for a data point x with respect to a Gaussian distribution is given by:

$$D(x) = \sqrt{(x - \mu)^{T}\Sigma^{-1}(x - \mu)}$$

where:
x is the data point.μ is the mean vector of the Gaussian distribution.Σ is the covariance matrix of the Gaussian distribution.

An anomaly score can be calculated based on the Mahalanobis distance, considering the inverse of the distance as a measure of anomaly:

Anomaly Score = $\frac{1}{D(x)}$

- Thresholding:

Anomalies are identified by comparing the anomaly scores to a predefined threshold. Data points with anomaly scores below this threshold are considered anomalies.

4. Expected Achievements and Outcomes

1- Algorithm Development: Develop and refine machine learning algorithms tailored for analyzing heart rate data collected from smartphone-based monitoring applications. These algorithms should demonstrate improved accuracy, reliability, and robustness compared to existing methods.

2- Validation of Accuracy: Conduct rigorous validation studies to assess the accuracy and reliability of the developed algorithms in detecting abnormalities and problematic patterns in heart rate data. This includes comparing the performance of the algorithms against established standards such as electrocardiography (EKG) and clinical diagnoses.

3- Real-World Applicability: Demonstrate the practical utility and real-world viability of the developed algorithms by integrating them into existing smartphone-based heart rate monitoring applications. Evaluate their performance in diverse user populations and scenarios to ensure broad applicability and effectiveness.

4- User Impact and Health Outcomes: Evaluate the impact of automated anomaly detection in heart rate monitoring on user behavior, health outcomes, and healthcare resource utilization. Assess whether early intervention enabled by automated alerts leads to improved detection of cardiovascular issues and better health outcomes for users.

5- Guidelines and Best Practices: Develop guidelines and best practices for the use of machine learning-based anomaly detection in heart rate monitoring applications. This includes recommendations for data collection, algorithm training, validation, and implementation to ensure optimal performance and user safety.

6- Knowledge Contribution: Contribute new knowledge to the field of personalized health monitoring and preventive healthcare engagement through publications in peer-reviewed journals, conference presentations, and engagement with relevant stakeholders. Share insights and lessons learned to advance the broader research and healthcare community.

The outcomes represent a holistic advancement in smartphone-based heart rate monitoring. Initially, efforts focus on refining machine learning algorithms to enhance accuracy and reliability, while simultaneously bolstering early detection capabilities to potentially prevent adverse health outcomes. Rigorous validation studies ensure the effectiveness and reliability of these algorithms, paving the way for their integration into smartphone apps, thereby facilitating improved health monitoring. This integration fosters user awareness and behavior regarding cardiovascular health, resulting in better health outcomes. Moreover, clinical integration ensures the seamless adoption of machine learning into practice. Finally, knowledge dissemination contributes to broader understanding and guides future research directions. Achieving these outcomes holds the promise of significantly improving the early detection and management of cardiovascular issues, ultimately benefiting users, healthcare providers, and society as a whole.

- Criteria for Success

The criteria for success in the research include:

1- Developing machine learning algorithms with high accuracy in detecting heart rate abnormalities.
2- Conducting validation studies that confirm algorithm effectiveness and reliability.
3- Integrating algorithms seamlessly into smartphone-based monitoring apps.
4- Demonstrating positive impacts on user health outcomes and behavior.
5- Collaborating with healthcare providers for clinical integration and guideline development.

5. Research and Engineering Process

Until now, our investigation has been focused on gathering information regarding heart diseases that are closely associated with heart rate, such as Cardiovascular disease. We delved into professional methods for monitoring heart rate effectiveness, such as Electrocardiography (ECG). Having examined the theoretical underpinnings of these algorithms, we are now poised to delve into hardware and software perspectives of the process, aiming to gain a comprehensive understanding of how these technologies operate in practice, and how we are going to apply these algorithms in our research and to build our application.

5.1 Hardware

In this section, we will outline the essential hardware components and devices required to develop and deploy our innovative heart rate monitoring application. We will provide details on the specific hardware specifications and their roles within the overall system architecture.

1- Smartphone with a camera:The app uses the smartphone's camera to detect changes in blood volume in the user's fingertip, which is the basis for measuring heart rate.

2- Smartphone with a flash:The search results mention that the user should place their fingertip on the camera and the rest of their finger on the flash to get the best results. (see how this works in section 5.3.2).

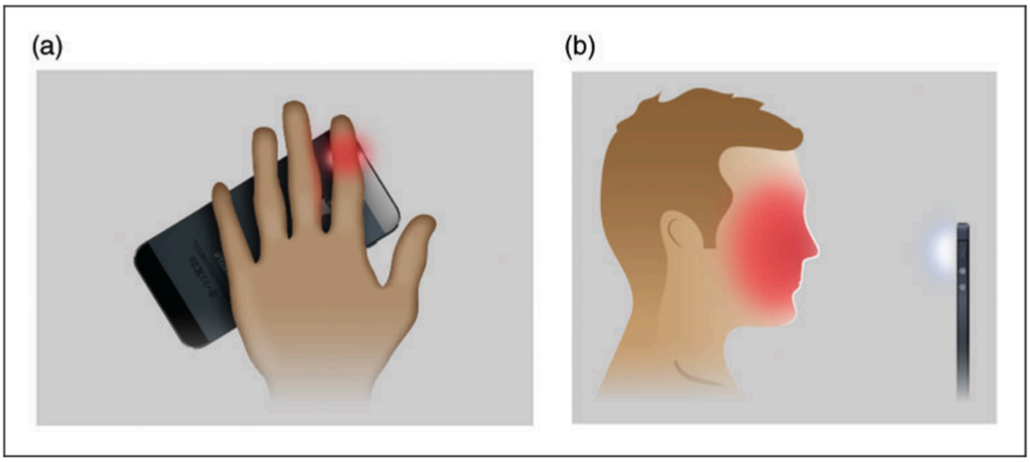


Figure 3. Two different concepts of measuring heart rate by PPG are known: In contact PPG, the subject places a finger on the built-in camera of the phone directly on the skin and the built-in flash provides the necessary light source in the visible range for reflection by blood cells (Figure 3a). In non-contact PPG, the camera is used in the classical way by holding the camera in front of the patient's face without the need for direct skin contact (Figure 3b). (a) Contact photoplethysmography; (b) non-contact photoplethysmography. (taken from European Journal of Preventive Cardiology [18]).

3- Smartwatch: These devices typically utilize photoplethysmography (PPG) sensors to measure heart rate, leveraging LED lights and photodetectors to detect changes in blood volume in the skin. Some advanced models also incorporate electrocardiogram (ECG) sensors for more precise monitoring of the heart's electrical activity, facilitating the detection of conditions like atrial fibrillation. Additionally, integrating accelerometer and gyroscope sensors enhances accuracy, particularly during physical activity, by filtering out noise and artifacts in heart rate data. Seamless connectivity options, ample onboard memory and processing power, long-lasting rechargeable batteries, and convenient charging methods further augment the functionality and usability of these devices, empowering users to better monitor and manage their cardiovascular health.

Figure 4. A heart rate measurement using a smartwatch, taken from Google Images.

5.2 Software

In this section, we will delve into the essential software components and tools necessary for the development and implementation of our cutting-edge heart rate monitoring application. We will discuss the specific software requirements, libraries, and frameworks that will form the backbone of our application's functionality.

1- Photoplethysmography (PPG) algorithm:The app uses a PPG algorithm to detect the changes in blood volume in the user's fingertip and convert that into a heart rate measurement.

2- Heart rate zone detection:The app can classify the user's heart rate into different zones, such as "Recovery Zone", "Fat-Burning Zone", "Target Heart Rate Zone", and "High Intensity Zone".

3- Data storage and visualization:The app allows users to save their heart rate history and view it in the form of color-coded bar graphs.

4- User interface and experience:The app should have a simple and intuitive user interface, as mentioned in the search results.

5- Python and Jupyter Notebook:The search results mention using Python and Jupyter Notebook to implement the machine learning algorithms.

6- Machine Learning Libraries:The search results mention using libraries like NumPy, SciPy, Scikit-learn, TensorFlow, and PyTorch for implementing the machine learning algorithms.

7- Sensor Data Processing Libraries:The search results also mention using libraries for processing sensor data, such as PPG and accelerometer data.

8- Specific Algorithms:The search results suggest using algorithms like LSTM, Random Forests, Support Vector Machines, and ARIMA for heart rate prediction and anomaly detection.

9- Development Environment:Depending on your preference, you may want to set up an Integrated Development Environment (IDE) like PyCharm or Visual Studio Code to write and run your code.

5.3 Methodology and Development Process

5.3.1 Heart rate monitoring app description

Our heart rate monitoring app will initially use a published research database containing diverse heart rate data, establishing baselines for normal and abnormal heart rate behaviors. Upon installation, the app will use this database to provide general heart rate alerts according to the initial database.

The app integrates with wearable devices equipped with PPG sensors, accelerometers, and ECG sensors to continuously capture heart rate and physiological data, which is securely transmitted to the cloud for processing and analysis. Advanced machine learning techniques, including CNNs, RNNs, and ensemble methods, will analyze this data to identify patterns, detect anomalies, and provide real-time cardiovascular health insights.

As users interact with the app and contribute their data, the app will adapt and personalize its algorithms and database to better suit each user's unique physiological characteristics. This personalized learning process will refine the app's monitoring and interpretation capabilities.

The app's evolving database will incorporate users' historical data, annotations, and relevant medical information, enabling it to track long-term trends and provide tailored recommendations and alerts. The user interface will display real-time heart rate measurements, visualizations, and notifications for any detected abnormalities.

Several key considerations emerge for the developing process in order to build an application to monitor heart rate.

5.3.2 Sensor Technology

1- Photoplethysmography (PPG): Photoplethysmography, is a non-invasive optical technique widely used to measure various physiological parameters like pulse rate, blood oxygen saturation, blood pressure, respiration rate, and left ventricular ejection time. PPG operates by detecting changes in blood volume in the skin's microvascular bed, utilizing light absorption or reflection caused by these blood volume fluctuations.

PPG can be performed using two methods: transmission mode or reflection mode. In the transmission mode, a light-emitting diode (LED) generates low-intensity infrared light that passes through the skin, and a photodiode on the opposite side detects the non-absorbed light transmitted through the tissue. In the reflection mode, the LED and photodiode are placed on the same side of the skin, and the photodiode detects the light reflected back from the tissue (see Figure. 5).

Both transmission and reflection PPG methods measure changes in blood volume, although they cannot directly quantify the exact amount of blood. The PPG signal represents these variations in blood volume, which are caused by the pulsatile nature of arterial blood flow. By analyzing the peaks in the PPG waveform, heart rate can be calculated based on the changes in blood volume.

While PPG offers a non-invasive and convenient method for continuous and real-time heart rate monitoring, suitable for integration into wearable devices, it faces challenges such as motion artifacts and low-perfusion conditions, which can potentially impact accuracy. Despite these limitations, PPG remains a popular choice for heart rate monitoring in various health and fitness applications due to its simplicity and versatility [35].

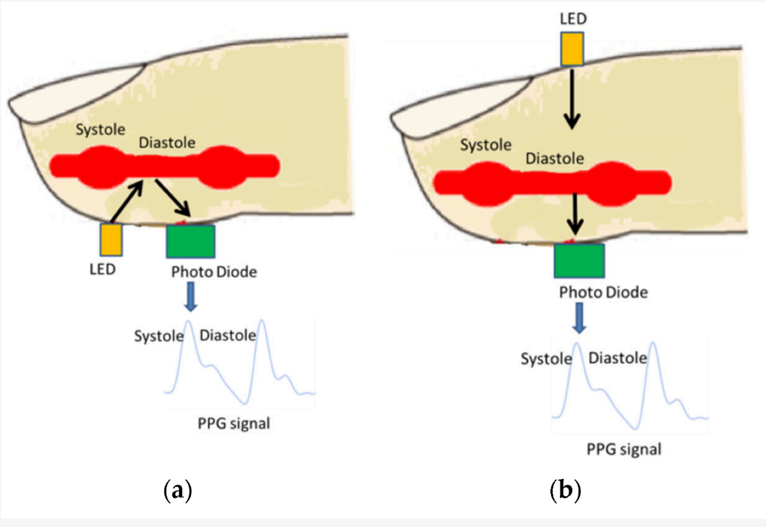


Figure 5. Schematic illustration of the photoplethysmography. The LED illuminates the skin and the non absorbed light will be detected by the photodiode. (a) For the reflected method, the LED and on the same side. (b) For the transmitted method, the LED and photodiode are on the opposite side [35].

2- Wearable devices, such as smartwatches and fitness trackers, increasingly utilize PPG sensors to measure heart rate. These sensors typically consist of a LED and a photodetector, detecting changes in reflected light from the skin due to blood volume variations. The resulting PPG waveform provides heart rate data, with each pulse representing a heartbeat. While PPG sensors offer simplicity, compactness, and power efficiency, their accuracy may be impacted by factors like motion artifacts and environmental conditions. Continuous efforts are made to enhance PPG sensor reliability through advancements in design and signal processing. Overall, wearable devices integrate PPG sensors to provide convenient and portable heart rate monitoring solutions.

The key takeaway is that the choice of sensor technology, whether it's PPG, ECG, or a combination of sensors in wearable devices, can significantly impact the accuracy of heart rate measurements. Smartphone-based PPG apps tend to have the most variability in accuracy, while wearable devices and ECG-based solutions generally provide more reliable and accurate heart rate data, especially for medical applications. Careful validation and testing are crucial to ensure the accuracy of any heart rate monitoring technology, regardless of the underlying sensor approach.

Smartwatches use green LEDs and photodiode sensors to measure heart rate by detecting changes in blood flow in the wrist. The green LEDs flash rapidly to capture blood flow variations with each heartbeat. For routine monitoring. Accuracy can be affected if the watch is worn loosely or if the skin has poor perfusion, particularly in cold conditions or during exercise. Additionally, third-party applications may experience reduced accuracy (see Figure. 6).

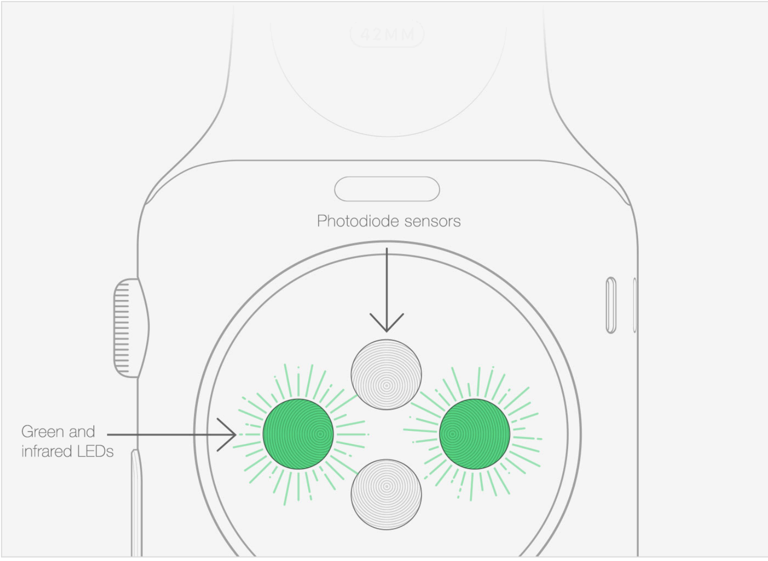


Figure 6. How do Smart Watches measure the heartbeat on the wrist, taken from https://www.quora.com/How-do-Smart-Watches-measure-the-heart-beat-on-the-wrist

5.3.3 DataBase

The app aims to provide instant heart rate monitoring and anomaly detection without prior training by utilizing a database of diverse heart rate patterns across demographics, activity levels, and health conditions. This database can be taken from research studies or curating data from various sources to represent a wide range of heart rate profiles, such as patterns during different activities, age groups, genders, and health conditions, a good example of a database to be used immediately after the installation will be the table presented in "How Much Activity Do Youth Get? A Quantitative Review of Heart-Rate Measured Activity" [36], the table show heart rate in various activities, also we can use the table presented in "Heart rate variability in healthy subjects is related to age and gender" [37] the table show heart rate among different ages and genders. The data can be organized into clusters, allowing the app to initially match the user's data with the most relevant group in the database and issue alerts based on deviations from expected patterns.
To manage this database, a NoSQL database like MongoDB can be used, with libraries like PyMongo in Python for interacting with the database. The heart rate data can be categorized into clusters based on factors like age, gender, activity levels, and health conditions, using formulas like Tanaka et al. 's (2001) for estimating maximum heart rate based on age.
As the app collects user data, it will store it with timestamps in the same database, gradually updating and refining the database to reflect personalized heart rate patterns. Over time, the app will transition from relying on the initial database to constructing personalized models tailored to individual users, improving anomaly detection and personalization. Data visualization tools like Matplotlib or Seaborn can be used to analyze and visualize the user's heart rate trends and patterns.

5.3.4 Data Processing and Analysis

Alongside data storage, the integration of machine learning techniques marks a pivotal step forward. Two prominent algorithms, Anomaly Detection and Long Short-Term Memory (LSTM), stand out for their potential in enhancing heart rate monitoring applications. Anomaly Detection algorithms, such as Isolation Forests or One-Class Support Vector Machines (OC-SVMs), excel in identifying heart rate values deviating significantly from a user's learned normal patterns. By pinpointing abnormalities, these algorithms enable timely alerts for potential heart rate irregularities. Conversely, LSTM algorithms specialize in time series forecasting, adept at learning a user's heart rate patterns over time. This capability enables the app to predict expected heart rates at various intervals, facilitating early detection of deviations from regular heart rate patterns. Both algorithms complement each other, offering a comprehensive approach to monitor heart rate and promptly alert users to potential cardiovascular concerns, thereby enhancing the app's effectiveness in promoting heart health awareness and proactive intervention.
Anomaly Detection algorithms, like Isolation Forests, work by isolating anomalies from the rest of the data by recursively partitioning the feature space. The algorithm calculates an anomaly score based on the average path length required to isolate a data point, with anomalies typically having shorter path lengths. One-Class Support Vector Machines (OC-SVMs) aim to learn a decision boundary that encompasses the majority of the normal data points, treating any points outside this boundary as anomalies.
LSTM algorithms, a variant of Recurrent Neural Networks (RNNs), are well-suited for time series forecasting tasks due to their ability to capture long-term dependencies in sequential data. The LSTM architecture includes gates (input, output, and forget gates) that regulate the flow of information, allowing the network to selectively remember or forget past information. The LSTM equation for a single cell can be represented as:

$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ # Forget gate
$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ # Input gate
$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ # Output gate
$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$ # Candidate cell state
$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$ # Cell state
$h_t = o_t * \tanh(c_t)$ # Hidden state

By combining Anomaly Detection algorithms for identifying abnormal heart rate patterns and LSTM algorithms for time series forecasting and pattern recognition, our app will provide a comprehensive solution for monitoring heart rate and detecting irregularities.

5.4 Sending notifications

The functionality of sending notifications is integral to the effectiveness of the heart rate monitoring app, serving two crucial purposes: prompting users to measure their heart rate at scheduled intervals for consistent monitoring and alerting users in the event of abnormal heart rate detection. The app leverages machine learning algorithms, such as Long Short-Term Memory (LSTM) networks and Anomaly Detection techniques, to analyze heart rate patterns and identify potential irregularities. For scheduled measurements, the app uses user-defined preferences or machine learning-based predictions to determine optimal times for prompting the user. LSTM networks, trained on historical heart rate data and activity patterns, forecast periods when measurements would be most informative, enabling intelligent reminders at appropriate intervals. In the event of abnormal heart rate detection, the app employs Anomaly Detection algorithms, like Isolation Forests or One-Class Support Vector Machines (OC-SVMs), to identify values deviating significantly from the user's normal patterns. These algorithms continuously monitor incoming data, and upon detecting an anomaly, the app promptly notifies the user. Notifications are tailored to user preferences, offering varying levels of urgency, customizable alert sounds, and additional contextual information, such as the user's location or activity at the time of detection. This personalized approach ensures notifications are meaningful and actionable, empowering users to stay proactive about their cardiovascular health. By incorporating machine learning algorithms for intelligent scheduling and anomaly detection, the app's notification system provides timely reminders for heart rate measurements and crucial alerts regarding potential abnormalities, encouraging adherence to monitoring routines and facilitating prompt action in response to fluctuations, ultimately promoting better heart health management and well-being.

6. Product

At the heart of our app's functionality is the seamless integration of machine learning algorithms, heart rate measurement capabilities, and a dynamic database system. Leveraging wearable devices with PPG, accelerometers, and ECG sensors, the app continuously captures user heart rate and physiological data. Advanced machine learning techniques such as CNNs for feature extraction and RNNs like LSTM networks for time series forecasting and pattern recognition process and analyze this data, adapting to the user's unique heart rate patterns for accurate predictions and anomaly detection. Processed data and machine learning outputs are securely stored in a cloud-based database, evolving with each interaction and new data point. Anomaly Detection algorithms monitor heart rate data, alerting users to deviations from normal patterns, while user feedback and medical information further refine the algorithms and database. This collaborative approach ensures the app continuously improves, offering a personalized solution for monitoring cardiovascular health, early issue detection, and proactive healthcare management. In this section, we delve into the functionality, user interface, diagrams, and architecture of our product. Through detailed explanations and visual representations, we aim to elucidate the intricate workings and design principles underpinning our innovative solution

6.1 Requirements

Functional:

1	The app shall get from the user some personal information
2	The app shall Collect the user's heart rate data at various times throughout the day
3	The app shall Store the collected heart rate data in a secure and privacy-compliant manner
4	The app should be able to learn the user's normal heart rate patterns over the course of the first week
5	The app shall be able connect with phone/smartwatch hardware such as Camera
6	The app shall employ anomaly detection algorithms to identify heart rate values that deviate significantly from the user's learned normal patterns
7	When an abnormal heart rate pattern is detected, the app shall trigger an alert or notification to the user, providing details about the detected anomaly
8	The app shall continuously update the machine learning models as new heart rate data is collected, allowing the system to adapt to changes in the user's heart rate patterns over time

Table 2. The essential features and functionalities that the heart rate monitoring app must possess to fulfill its intended purpose and meet user expectations.

Non-functional:

1	The app should have a friendly - easy to use interface
2	The app shall have a high level of accuracy in detecting abnormal heart rate patterns
3	The app shall have a responsive and intuitive user interface, with a maximum response time of 2 seconds for displaying updated heart rate data
4	The app shall be compatible with various mobile platforms (e.g., iOS, Android) and support a range of wearable devices and sensors
5	The app shall be scalable and able to handle an increasing number of users and data without compromising performance
6	The app shall be maintainable and easily updatable to incorporate new features or improvements

7	The app shall be reliable and available for continuous heart rate monitoring, with minimal downtime or interruptions
8	The app shall be portable and able to run on different hardware configurations or environments

Table 3. The qualitative attributes, constraints, and performance criteria to ensure optimal user experience, reliability, and overall system quality.

6.2 Architecture overview

The heart rate monitoring app's architecture comprises essential modules: Data Acquisition, Preprocessing, User Profiling, Anomaly Detection, Alerts, User Interface, and Data Storage. Data Acquisition gathers heart rate data, Preprocessing cleans and normalizes it, while User Profiling establishes baseline patterns. Anomaly Detection monitors deviations using methods like Isolation Forests, triggering Alerts. The User Interface displays data, and Data Storage ensures secure management. Optionally, Cloud Integration enables scalability. This architecture ensures efficient personalized heart rate monitoring and anomaly detection in the app (see Figure 7).

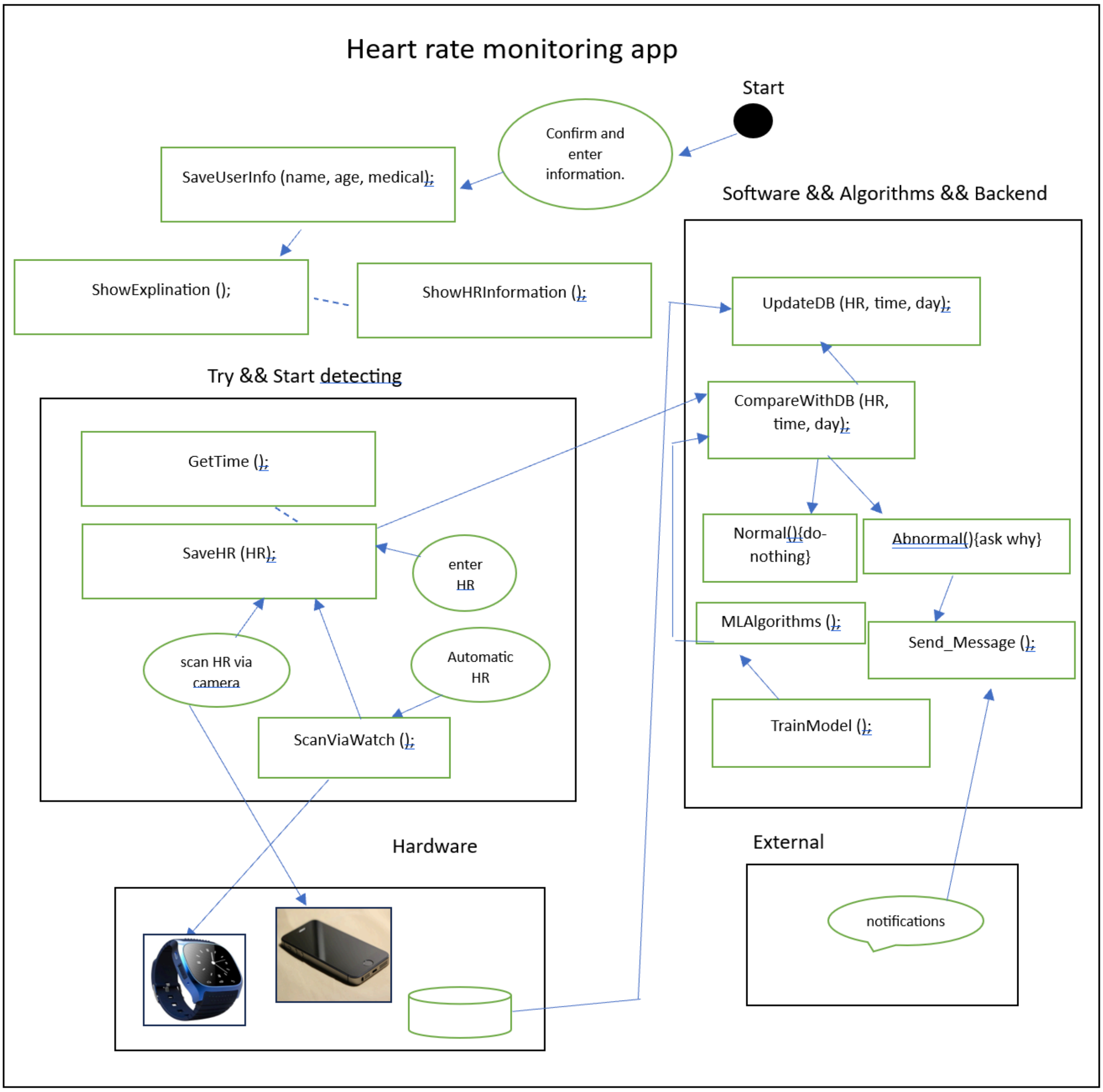


Figure 7. The architectural diagram illustrates the comprehensive system design, depicting the interconnected components, data flows, and underlying technologies that collectively enable the seamless operation of the heart rate monitoring application.

6.3 Interface

Now, we will introduce the proposed interface for the app. In the initial screen will see a welcome screen and a condition agreement (see Figure 8), and in the next screen the user will encounter terms and conditions (see Figure 9), then the user will enter some personal information (see Figure 10). In the next screen the user will see a recommendation to install the app on his smartwatch (see Figure 11), then the user will see a congratulations screen (see Figure 12), the user can also see some information about heart rate (see Figure 13 and 14).Next the app will offer the user to enter manually his heart rate (if he is using an external device, see Figure 15) or simply use his phone camera (see Figure 16). If the measured heart rate is abnormal the app will tell the user and give him the option to explain this abnormal heart rate (see Figure 17 and 18), accordingly the app will show the user a suited message (see Figures 19-23).

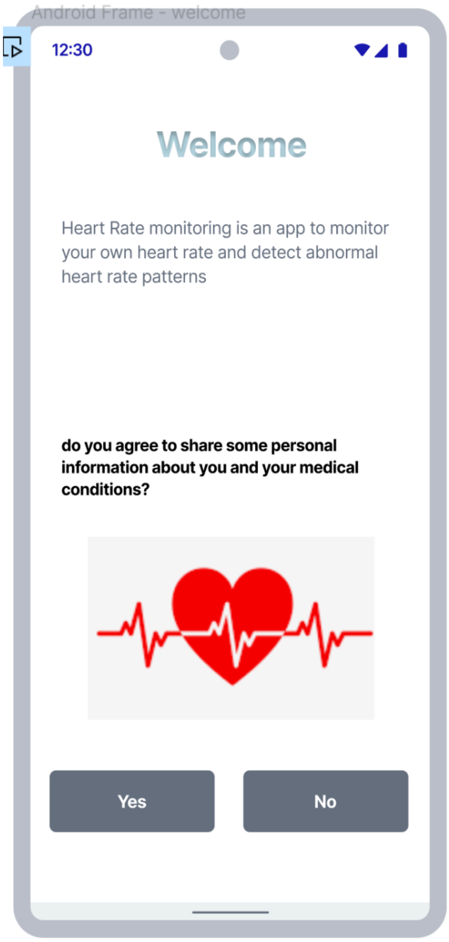


Figure 8. The initial welcome screen.

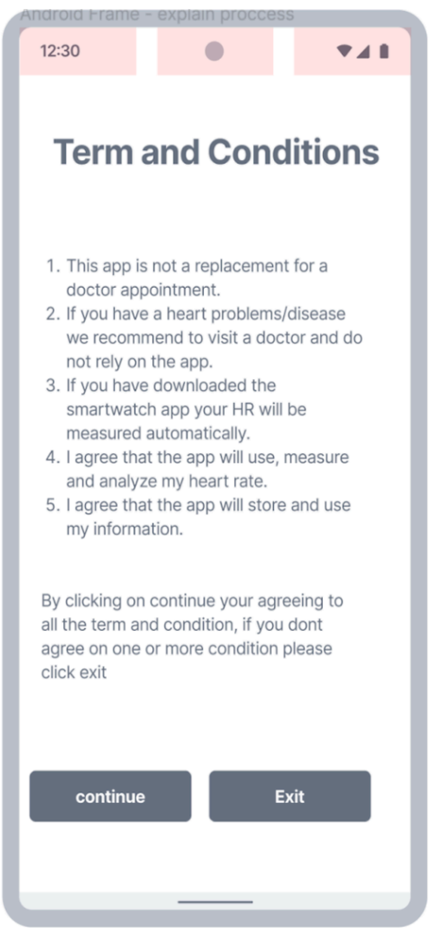


Figure 9. Terms and conditions.

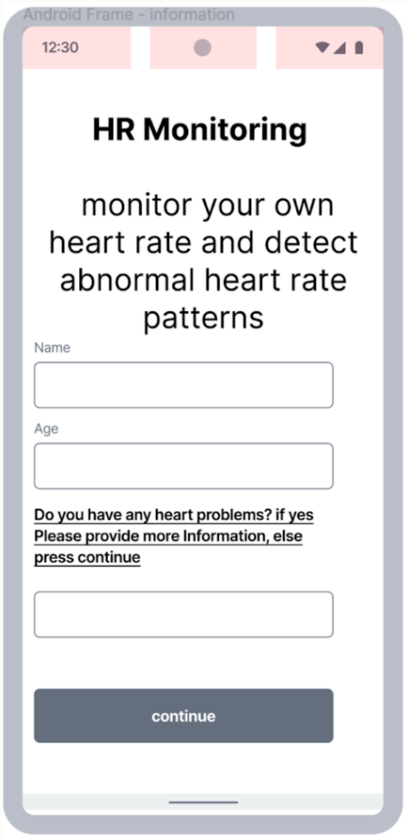


Figure 10. Personal information.

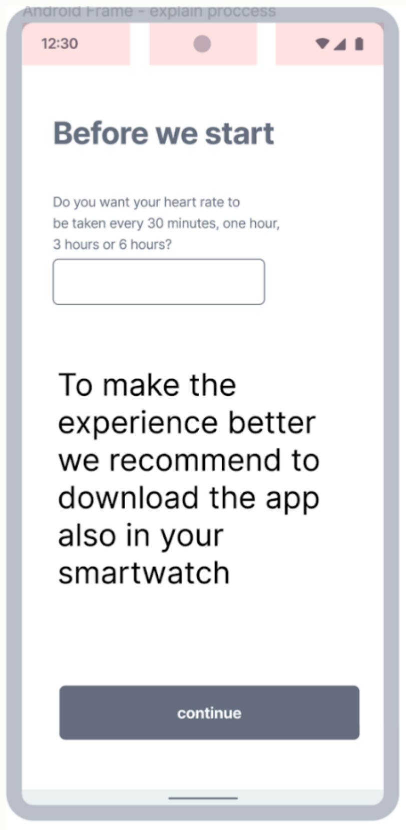


Figure 11.Recommendation.

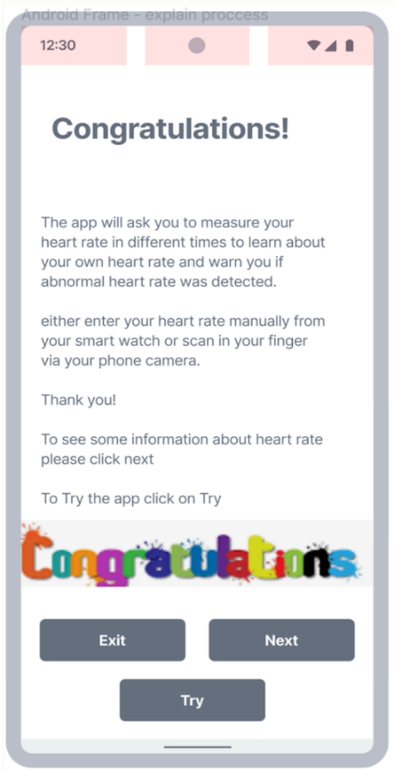


Figure 12. Congratulations.

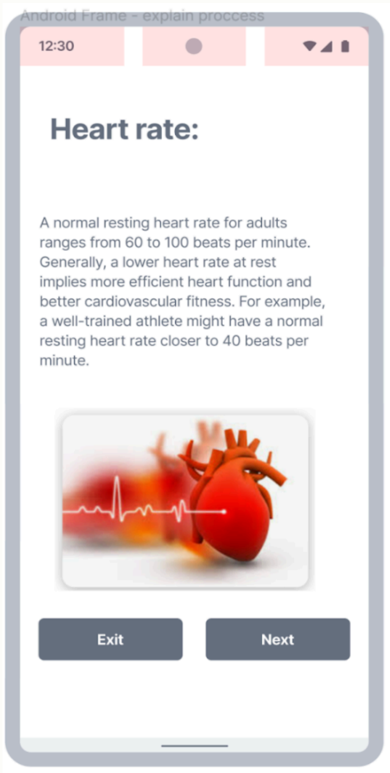


Figure 13. Heart rate information 1.

Figure 14. Heart rate information 2.

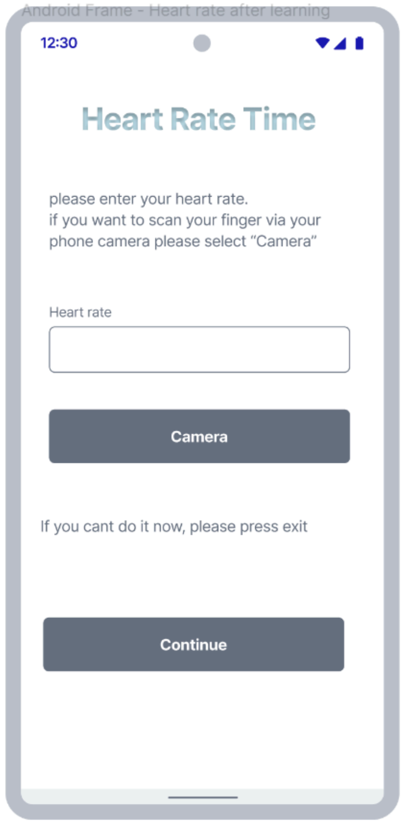


Figure 15. Enter Heart rate.

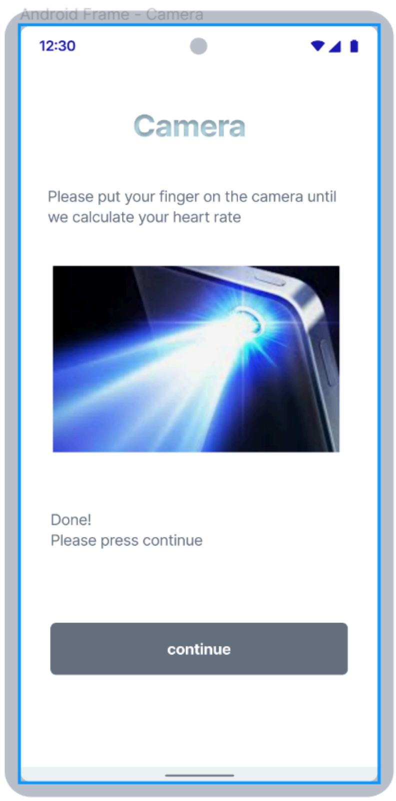


Figure 16. Measure heart rate via camera.

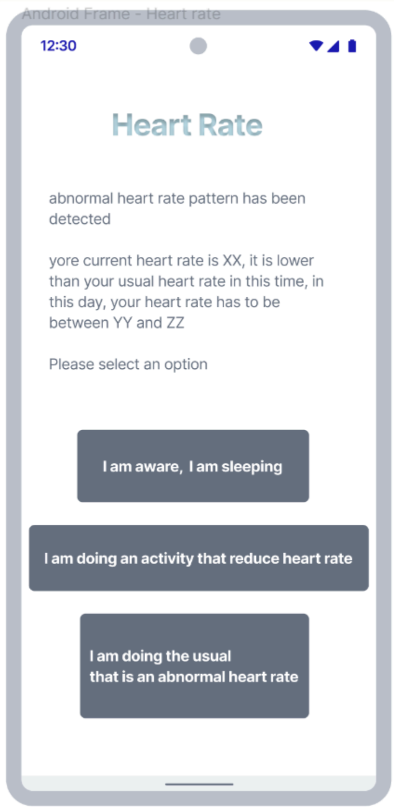


Figure 17. Low heart rate.

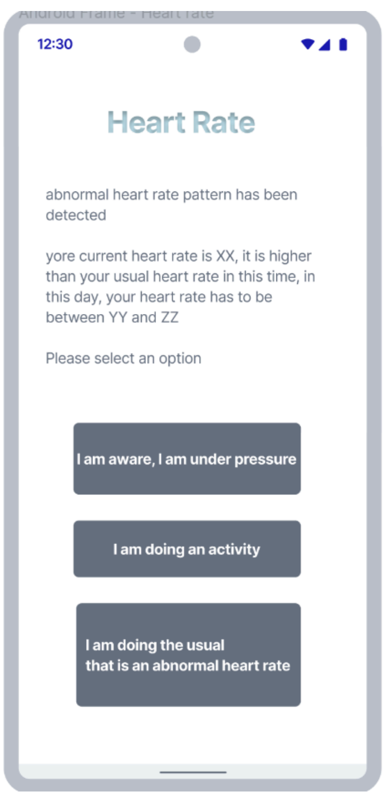


Figure 18. High heart rate.

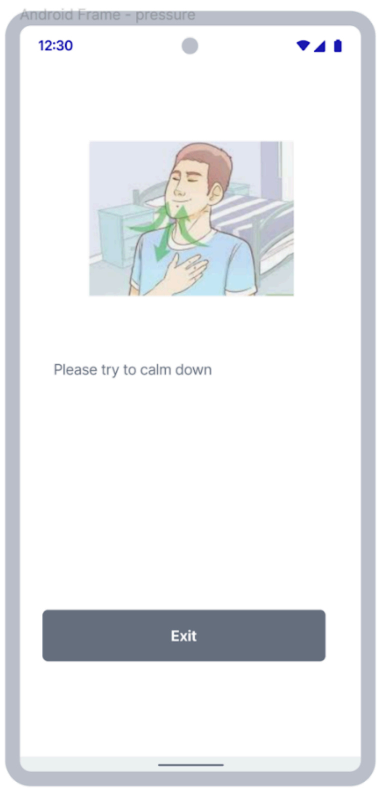


Figure 19. Recommendation: calm down.

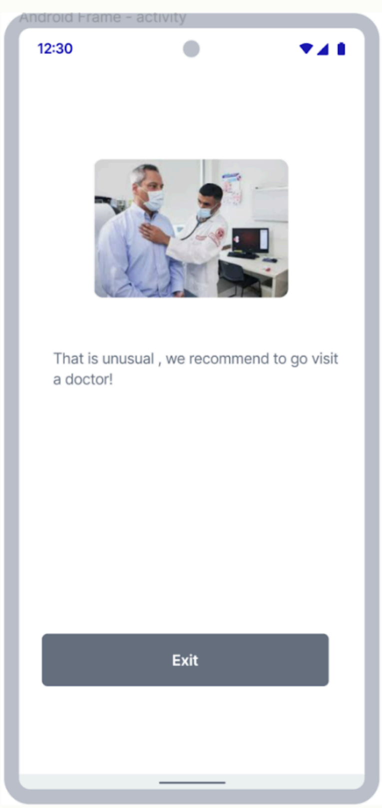


Figure 20. Abnormal heart rate detected.

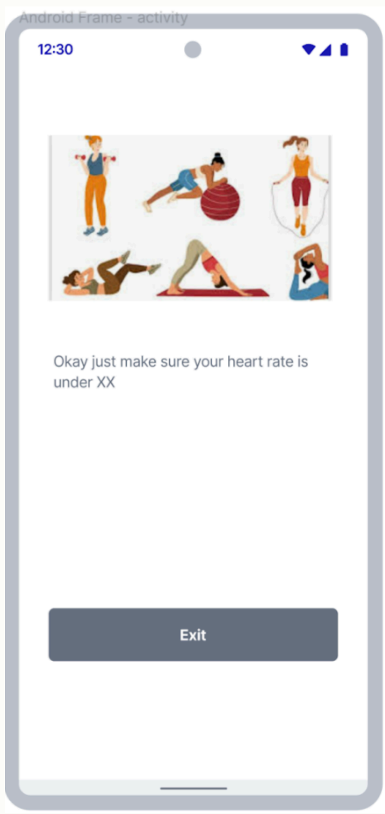
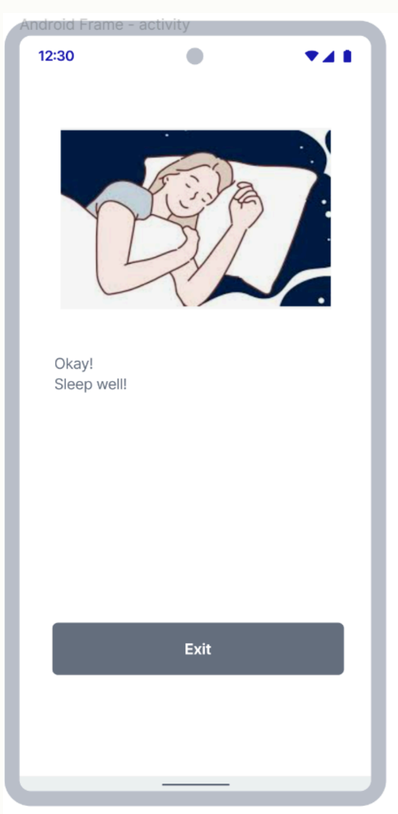


Figure 21. Exercise.



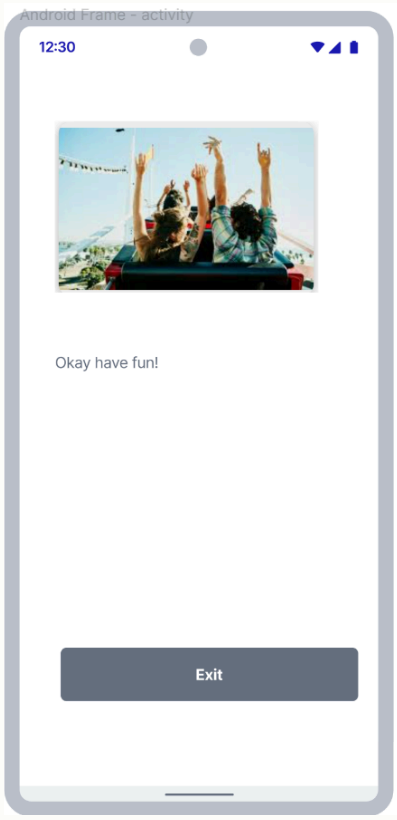


Figure 22. Sleeping. Figure 23. Activity.

6.4 Use case

In the following use case diagram, we illustrate the interactions and relationships between users and the system in the context of Heart Rate monitoring application.

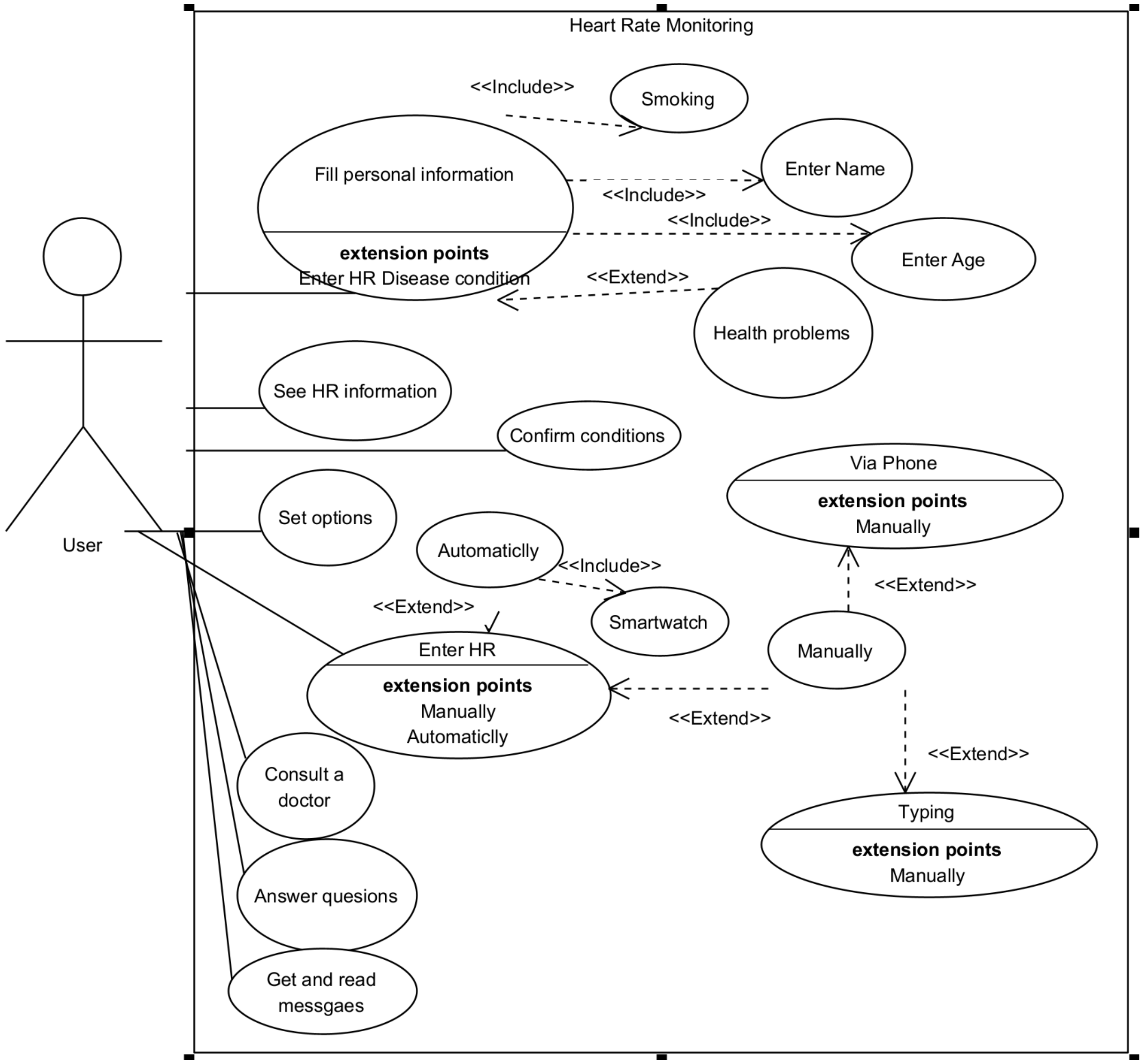


Figure 24. Use Case Diagram: Heart Rate Monitoring Application Depicting the Interactions and Functionalities for the user.

6.5 Activity Diagram

In the following activity diagram, we depict the sequence of activities and flow of control within the Heart Rate monitoring application. This diagram provides a visual representation of the steps involved in accomplishing a particular task or process.

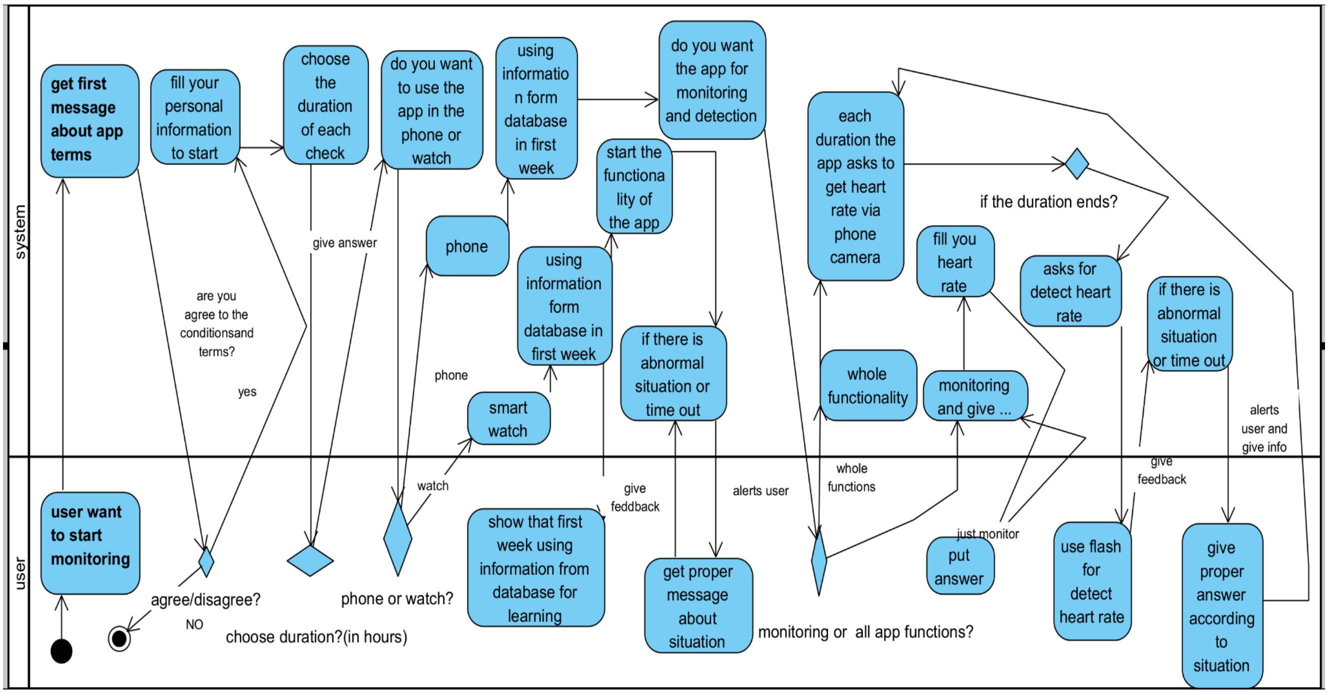


Figure 25. Activity Diagram: Heart Rate Monitoring Application Workflow Illustrating the Sequence of Activities and Data Flow for Comprehensive Heart rate monitoring.

7. Verification and Evaluation

In this section we will detail the testing procedures used to validate our heart rate monitoring application. Through rigorous testing phases such as unit testing, integration testing, end-to-end testing, performance testing, and accuracy testing, we assess the app's functionality, reliability, and accuracy. The table below summarizes each testing phase and its outcomes, providing assurance of the app's readiness for deployment and effectiveness in meeting user needs for cardiovascular health monitoring.

Testing Type	Process	Expected results
Unit Testing	Components Tested: Individual components such as data preprocessing, machine learning algorithms, user interface elements, and data visualization features are tested in isolation.	Unit tests for data preprocessing ensure handling of missing values, outliers, and data normalization. For machine learning algorithms, validation ensures accurate heart rate prediction. User interface tests confirm responsive elements like buttons and menus, while data visualization tests validate accurate display of heart rate data through charts and graphs.
code demonstrating the use of pytest and unittest frameworks for testing in Python	Testing Tools: PyTest or unittest frameworks in Python are used to automate unit tests.	Test Cases: Each component has specific test cases designed to verify its functionality and handle different scenarios correctly.
Integration Testing	Integration Points: Test the integration of different components within the app, ensuring they work together seamlessly.	Component Integration: Integration tests confirm that different components of the app communicate effectively and share data as expected. Data Flow: Tests validate that data flows smoothly between data preprocessing, machine learning algorithms, and user interface components. External Libraries: Integration tests ensure that external libraries or APIs are properly integrated and function correctly within the app.

integration testing for verifying the correct data flow between different components of the application.	Data Flow: Verify that data flows correctly between components and interactions produce the expected outcomes.	Components ensure seamless data transfer, preventing loss or corruption. Outputs are accurately received and utilized downstream. Data transformations are correct, yielding expected formats. External libraries and APIs appropriately handle and process data as required.
End-to-End Testing	User Scenarios: Simulate common user scenarios, such as launching the app, recording heart rate data, analyzing data using machine learning algorithms, and displaying results. User Interactions: Test how users interact with the app from start to finish, including navigation and task completion.	User Scenarios: End-to-end tests confirm that the app performs as expected for common user scenarios, such as starting a heart rate measurement session and viewing results. User Interactions: Testing reveals that users can navigate the app easily and complete tasks without encountering significant obstacles.
Performance Testing	Performance Metrics: Assess app responsiveness, speed, and resource usage.	Responsiveness: Performance tests show that the app responds quickly to user interactions, with minimal latency. Resource Usage: Measurements indicate that the app consumes resources efficiently, with reasonable memory usage and CPU utilization.
Load Testing	Workload: Test app performance under different workloads, such as processing large datasets or running complex machine learning algorithms.	Workload Handling: The app performs well under different workloads, demonstrating consistent performance even with large datasets or complex algorithms. Measurements: Record response times, memory usage, and CPU utilization.

Accuracy Testing	Comparison: Compare heart rate measurements obtained from the app with ground truth values from reliable sources.	Comparison: Accuracy tests reveal that heart rate measurements obtained from the app closely match ground truth values from reliable sources. Metrics: Calculated metrics demonstrate high sensitivity, specificity, accuracy, and precision, indicating that the app accurately predicts heart rate and detects abnormalities.

Table 4. The comprehensive testing approach, encompassing unit, integration, end-to-end, performance, and accuracy testing, ensures the application's robustness, reliability, and adherence to functional and non-functional requirements.

8. References

[1] Achten, J., & Jeukendrup, A. E. (2003). Heart rate monitoring. *Sports Medicine, 33*(7), 517-538.

[2] Behar, J. A., et al. (2021). Physiological measurements from smartwatches are associated with atrial fibrillation. *Heart Rhythm, 18*(12), 2125-2127.

[3] Bouts, A. M., et al. (2018). The accuracy and validity of iOS-based heart rate apps during moderate to high intensity exercise. *International Journal of Exercise Science, 11*(7), 533-540.

[4] Estes, N. A. M., et al. (2018). 2018 ACC/AHA/HRS guideline on the evaluation and management of patients with bradycardia and cardiac conduction delay: a report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines and the Heart Rhythm Society. *Journal of the American College of Cardiology, 74*(7), e51-e156.

[5] Gaziano, T., et al. (2006). Cardiovascular disease. In: *Disease control priorities in developing countries* (2nd ed.). Washington, DC: The International Bank for Reconstruction and Development / The World Bank.

[6] Kligfield, P., et al. (2007). Recommendations for the standardization and interpretation of the electrocardiogram: part I: the electrocardiogram and its technology: a scientific statement from the American Heart Association Electrocardiography and Arrhythmias Committee, Council on Clinical Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society. *Circulation, 115*(10), 1306-1324.

[7] Losa-Iglesias, M. E., et al. (2016). Reliability and concurrent validity of a peripheral pulse oximeter and health-app system for the quantification of heart rate in healthy adults. *Health Informatics Journal, 22*(2), 151-159.

[8] Monkaresi, H., et al. (2014). A machine learning approach to improve contactless heart rate monitoring using a webcam. *IEEE Journal of Biomedical and Health Informatics, 18*(4), 1153-1160.

[9] Rautaharju, P. M., et al. (2009). AHA/ACCF/HRS recommendations for the standardization and interpretation of the electrocardiogram: part IV: the ST segment, T and U waves, and the QT interval. *Journal of the American College of Cardiology, 53*(11), 982-991.

[10] Salerno, S. M., et al. (2003). Competency in interpretation of 12-lead electrocardiograms: a summary and appraisal of published evidence. *Annals of Internal Medicine, 138*(9), 751-760.

[11] Seshadri, D. R., et al. (2019). Wearable sensors for detection of atrial fibrillation. *Frontiers in Physiology, 10*, 1351.

[12] Shaffer, F., & Ginsberg, J. P. (2017). An overview of heart rate variability metrics and norms. *Frontiers in Public Health, 5*, 258.

[13] TechTarget. (2023). History and evolution of machine learning: A timeline. Retrieved from https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History

[14] Thygesen, K., et al. (2018). Fourth universal definition of myocardial infarction (2018). *European Heart Journal, 40*(3), 237-269.

[15] Vashist, S. K., et al. (2014). Commercial smartphone-based devices and smart applications for personalized healthcare monitoring and management. *Diagnostics, 4*(3), 104-128.

[16] Wallen, M. P., et al. (2016). Accuracy of heart rate watches: Implications for weight management. *PLoS ONE, 11*(5), e0154420.

[17] World Health Organization. (n.d.). Cardiovascular diseases (CVDs). Retrieved from https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

[18] Wyss, C. (2017). Accuracy of heart rate apps varies. *European Journal of Preventive Cardiology, 24*(4), 399-400.

[19] Losa-Iglesias, M. E., Becerro-de-Bengoa-Vallejo, R., & Becerro-de-Bengoa-Losa, K. R. (2016). Reliability and concurrent validity of a peripheral pulse oximeter and health-app system for the quantification of heart rate in healthy adults. *Health Informatics Journal, 22*(2), 151-159.

[20] Seshadri, D. R., Davies, E. V., Harlow, E. R., Hsu, J. J., Stratton, S. V., Virudachalam, S., & Silva, J. N. (2020). Wearable sensors for COVID-19: A call to action to harness our digital age for remote patient monitoring and virtual assessments. *Frontiers in Digital Health, 2*, 8.

[21] Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., & Acharya, U. R. (2018). Deep learning for healthcare applications based on time series data. *KES Studies in Computational Intelligence, 1*(1), 1-18.

[22] Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., & Ng, A. Y. (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv:1707.01836.

[23] Ren, H., Ranks, B., Pazin, N., Silva, J. N., & Kuang, D. (2020). Anomaly detection with conditional anomaly detection networks. arXiv preprint arXiv:2009.07612.

[24] Yin, J., Yang, L., Xiao, Y., Jiang, X., Zhou, Y., & Shen, W. (2019). Spectro-temporal feature extraction and anomaly detection for machine condition monitoring using unsupervised generative adversarial variational autoencoders. *Neurocomputing, 345*, 99-108.

[25] Seshadri, D. R., Davies, E. V., Harlow, E. R., Hsu, J. J., Stratton, S. V., Virudachalam, S., & Silva, J. N. (2020). Wearable sensors for COVID-19: A call to action to harness our digital age for remote patient monitoring and virtual assessments. *Frontiers in Digital Health, 2*, 8.

[26] Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., & Ng, A. Y. (2017). Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv:1707.01836.

[27] Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., & Acharya, U. R. (2018). Deep learning for healthcare applications based on time series data. *KES Studies in Computational Intelligence, 1*(1), 1-18.

[28] Yin, J., Yang, L., Xiao, Y., Jiang, X., Zhou, Y., & Shen, W. (2019). Spectro-temporal feature extraction and anomaly detection for machine condition monitoring using unsupervised generative adversarial variational autoencoders. *Neurocomputing, 345*, 99-108.

[29] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386-408.

[30] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.

[31] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*.

[32] TechTarget. (2023). History and evolution of machine learning: A timeline. Retrieved from https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History.

[33] World Health Organization. (n.d.). Cardiovascular diseases (CVDs). Retrieved from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[34] Estes, N. A. M., et al. (2018). 2018 ACC/AHA/HRS guideline on the evaluation and management of patients with bradycardia and cardiac conduction delay: a report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines and the Heart Rhythm Society. *Journal of the American College of Cardiology, 74*(7), e51-e156.

[35] Liu, S.-H., Li, R.-X., Wang, J.-J., Chen, W., & Su, C.-H. (2020). Classification of photoplethysmographic signal quality with deep convolution neural networks for accurate measurement of cardiac stroke volume. *Sensors, 20*(13), 3721.

[36] Cerny, L., & Roemmich, J. N. (2001). How Much Activity Do Youth Get? A Quantitative Review of Heart-Rate Measured Activity. Pediatrics, 108(3), e44. https://doi.org/10.1542/peds.108.3.e44

[37] Jensen-Urstad, K., Storck, N., Bouvier, F., Ericson, M., et al. (1997). Heart rate variability in healthy subjects is related to age and gender. Acta Physiologica Scandinavica, 160, 235-241.