ANDROID BASSED SMART HEART MONITORING SYSTEM

A PROJECT REPORT

Submitted by

AKASH J **[710119205002]**

HARSHAVARDHAN J **[710119205004]**

PARTHASARATHY B **[710119205701]**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

ADITHYA INSTITUTE OF TECHNOLOGY

COIMBATORE 641107

ANNA UNIVERSITY: CHENNAI 600025

JUNE 2022

ANNA UNIVERSITY : CHENNAI 600025

BONAFIDE CERTIFICATE

This project report **"ANDROID BASED SMART HEART MONITORING SYSTEM "** is the bonafide work of **"AKASH J, HARSHAVARDHAN J , PARTHASARATHY B"** who carried out the project work under my supervision.

SIGNATURE
Dr.S.M. NANDHAGOPAL M.E, Ph.D.,
HEAD OF THE DEPARTMENT
Computer Science and Engineering
Adithya Institute of Technology
Kurumbapalayam
Coimbatore-641107

SIGNATURE
Dr.V.GANESHKARTHIKEYAN M.E, Ph.D.,
SUPERVISOR
Associate professor
Information Technology
Adithya Institute of Technology
Kurumbapalayam
Coimbatore-641107

Submitted for the Anna University Viva-Voce examination held on _____

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We are pleased to present the "**ANDROID BASED SMART HEART MONITORING SYSTEM**" project and take the opportunity to express our profound gratitude to all of those people who helpedus in the completion of this project.

We would like to express our deep sense of gratitude to the management of Adithya Institute of Technology, **Er.C.SUKUMARAN, Chairman, Mr.S.PRAVEEN KUMAR** for providing us with all facilities to carry out thisproject.

we would like to express our gratitude to our Principal Madam, **Dr.D.SOMASUNDARESWARI,** Adithya Institute of Technology, for providing us all facilities to carry out this project successfully.

We would like to express our profound thanks to our Head of the Department, **Dr.S.M.NANDHAGOPAL ,Computer Science and Engineering,** Adithya Institute of Technology, for her valuable suggestions and guidance throughout the course of the project.

We also express our sincere thanks **Mr.V.GANESH KARTHIKEYAN, Associate professor ,** Department of Information Technology. We also thank all the **Faculty Members** of our department for their help in making this project a successful one.

Finally, we take this opportunity to extend our deep appreciation to ourfamily and friends, for all they meant to us during the crucial times of the completion of our project.

ABSTRACT

Electrocardiogram (ECG) is a very powerful medical device, especially for heart related issues. It helps to understand how the heart works, and gives a helping hand in diagnosis of heart related problems, which is becoming very important in modern era. In this paper, we are proposing a portable android based device which is capable to store ECG signal through wireless medium for a long time for predicting the heart related artifacts and also to act as a prior warning system for the user. We are going to implement using Smart Watch to make it portable and wireless. And we are using a cloud storage to record the real-time data which can be used to diagnose a disease or arrhythmia. It can display the real time ECG in the smart phone to the user, by the help of an android app made using android studio.

TABLE OF CONTENT

44

CHAPTER 1

INTRODUCTION

OBJECTIVE

The main objective of this project is to create a working model to predict heart disease using Mobile Computing and Cloud Storage. We will use here Android App, Android Wear OS, and Cloud Computing Algorithms to utilize the data-set available on users ECG data to predict heart Alignments and Notify the User.

MOTIVATION OF THE PROJECT

Heart is one of your body's most important organs. Essentially a pump, the heart is a muscle made up of four chambers separated by valves and divided into two halves. Each half contains one chamber called an atrium and one called a ventricle. The atria (plural for atrium) collect blood, and the ventricles contract to push blood out of the heart. The right half of the heart pumps oxygen-poor blood (blood that has a low amount of oxygen) to the lungs where blood cells can obtain more oxygen. Then, the newly oxygenated blood travels from the lungs into the left atrium and the left ventricle. The left ventricle pumps the newly oxygen-rich blood to the organs and tissues of the body. This oxygen provides your body with energy and is essential to keep your body healthy.

The general term used to cover malfunctions of the heart is Heart Disease, or sometimes Cardiac Disease ("Cardiac" is a Latin term for the heart). Though there are multiple forms of heart disease, our discussion focuses on the two most common: Heart Attack and Heart Failure. This document is designed to teach you about heart attacks and heart failure: what causes these diseases, what forms these diseases take, and what can be done to treat these diseases when they occur. As both of these diseases are to some extent avoidable, we have also provided a discussion of preventative steps you can take to decrease your chances of having to deal with heart disease, or to minimize the negative effects of existing heart disease.

Please note that though this information is as accurate as possible, it is no substitute for a qualified physician's advice. Consult with your doctor before making changes to any treatment regimen you may be prescribed, and before beginning any program of exercise or other significant lifestyle change, especially if you have a known heart problem or are a middle-aged or older adult. There is no substitute for your doctor's advice.

Although heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. We start our discussion of heart disease by describing these common risk factors, and then move on to cover specific conditions.

There are many factors that can increase your risk of getting heart disease. Some of these factors are out of your control, but many of them can be avoided by choosing to live a healthy lifestyle. Some of the risk factors you cannot control are:

- **Gender**:
- **Age**:
- **Family History**:
- **Obesity**:
- **High Cholesterol**:
- **Smoking**:
- **High Blood Pressure**:
- **Diabetes**:
- **Other Factors**:

SCOPE OF THE PROJECT

In recent years, Android Software’s have generated a significant influence and commitment in various health related solutions. More heart problems have been increasing in the society Motivated by this, the authors have used Android Programming and Sensor for detection and proper diagnosis of Heart patients. The

main goal of this project is to predict heart alignment using the old ECG data of the User which have been stored in the cloud. Old user data will be compared with the current data and find the anomaly and notify the user.

PROPOSED SYSTEM

By using the Smart Watch interface User's pulse is monitored until the user is wearing. ECG data is stored by date and time wise in the cloud under user's ID. Once 15 days the old data will be compared with the new ECG data to find any rise in heart alignment. If there is any anomaly in the comparison user is notified via the Android app and the Result will be shown. Using this the user can get Treatment and find the correct heart problem in the initial stage so the treatment can be started.

CHAPTER 2

LITERATURE REVIEW

1) "Real-Time Signal Quality- Aware ECG Telemetry System for IoT-Based Health Care Monitoring"

U. Satija, B. Ramkumar and M. Sabarimalai Manikandan *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 815-823, June 2017.

In this paper, we propose a novel signal quality-aware Internet of Things (IoT)-enabled electrocardiogram (ECG) telemetry system for continuous cardiac health monitoring applications. The proposed quality-aware ECG monitoring system consists of three modules: 1) ECG signal sensing module; 2) automated signal quality assessment (SQA) module; and 3) signal-quality aware (SQAw) ECG analysis and transmission module. The main objectives of this paper are: design and development of a light-weight ECG SQA method for automatically classifying the acquired ECG signal into acceptable or unacceptable class and real-time implementation of proposed IoT-enabled ECG monitoring framework using ECG sensors, Arduino, Android phone, Bluetooth, and cloud server. The proposed framework is tested and validated using the ECG signals taken from the MIT-BIH arrhythmia and Physionet challenge databases and the real-time recorded ECG signals under different physical activities. Experimental results show that the proposed SQA method achieves promising results in identifying the unacceptable quality of ECG signals and outperforms existing methods based on the morphological and RR interval features and machine learning approaches. This paper further shows that the transmission of acceptable quality of ECG signals can significantly improve the battery lifetime of IoT-enabled devices. The proposed quality-aware IoT paradigm has great potential for assessing clinical acceptability of ECG signals in improvement of accuracy and reliability of unsupervised diagnosis system

2) "Wireless body sensor network and ECG Android application for eHealth",

Khalaf and R. Abdoola *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, pp. 1-4, 2017.

A wireless Body Sensor Network (WBSN) with ECG android monitor is presented that is capable of monitoring and displaying the ECG waveform and Heart Rate of an individual in real-time. The system uses the Wi-Fi 802.11 standard for wireless transmission of the data to an Android based mobile phone. The developed android smart-phone application displays the ECG information as well as the heart rate and the GPS data. The location is obtained using the GPS available on most smart-phones. The hardware component presented is small and modular to allow the system to fit on a wearable patch.

3) "An Energy Efficient ECG Signal Processor Detecting Cardiovascular Diseases on Smartphone",

S. K. Jain and B. Bhaumik, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 2, pp. 314-323, April 2017

A novel disease diagnostic algorithm for ECG signal processing based on forward search is implemented in Application Specific Integrated Circuit (ASIC) for cardiovascular disease diagnosis on smartphone. An ASIC is fabricated using 130-nm CMOS low leakage process technology. The area of our PQRST ASIC is 1.21 mm^2 . The energy dissipation of PQRST ASIC is 96 pJ with a supply voltage of 0.9 V. The outputs from the ASIC are fed to an Android application that generates diagnostic report and can be sent to a cardiologist via email. The ASIC and Android application are verified for the detection of bundle branch block, hypertrophy, arrhythmia and myocardial infarction using Physionet PTB diagnostic ECG database. The failed detection rate is 0.69%, 0.69%, 0.34% and 1.72% for bundle branch block, hypertrophy, arrhythmia and myocardial infarction respectively. The AV block is detected in all the three patients in the Physionet St. Petersburg arrhythmia database. Our proposed ASIC together with our Android application is the most suitable for an energy efficient wearable cardiovascular disease

detection system.

.

4) "Design and implementation of low cost ECG monitoring system for the patient using smartphone"

M. A. Ahamed, M. K. Hasan and M. S. Alam, *2015 International Conference on Electrical & Electronic Engineering (ICEEE)*, pp. 261-264, 2015.

Electrocardiographic (ECG) equipment plays a vital role for diagnosis of cardiac disease. However, the cost of this equipment is huge and the operation is too much complex which cannot offer better services to a large population in developing countries like Bangladesh. In this paper, we have designed and implemented a low cost portable single channel ECG monitoring system using smartphone having android operating system and Arduino. This manuscript also demonstrates the use of Android smartphone for processing and visualizing ECG signal. Our designed system is battery powered and it gives wireless feature. This system can also be used with desktop computer or laptop having either Windows, Linux or Mac OS. For this purpose a software is developed. An Android application is developed using Processing IDE, which requires Android version 2.3 and API level of 10. This application does not need USB host API. For this reason, around 98% Android smartphone in the market can be used for this system.

5) "Bond graph model for the right atrium of heart"

L. Moin, V. Uddin, A. Y. Memon and S. H. Zaidi, *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 181-185, 2016.

The discussion on the mathematical modeling of atrium is not still deliberated and most of the literature and research based on the model and functioning of ventricles of heart. In this paper we tried to formulate the model for right atrium using Bond Graph technique. The Bond Graph is a unified modeling approach to portray multi domain-energy systems. Cardiovascular system is a very complex system which includes interaction of several

subsystems like the heart and the circulatory system. Several energy domains describe the CVS activity; such as mechanical, hydraulic, chemical and electrical events. To take into account all these characteristics, the bond graph seems to provide a unified approach. In this research paper the right atrium is modeled for ejection phase using bond graph unified modeling. In this paper the unified Bond Graph model of the right atrium is presented, simulated and validated. The results obtained using MATLAB tool are compared with CVS physiological data present in Simbiosys (a software for simulating biological systems) and also with the CVS Wiggers diagram of heart cycle. The model takes into account a simplified description of the right atrium which is close to the medical investigation promoting the apperception and the dialogue between engineers and physiologists

6) "Electrocardiographic Interpretation of Cardiac Muscle and Coronary Blood Flow Abnormalities: Vectorial Analysis"

A. Guyton and J. Hall , Guyton and Hall Textbook of Medical Physiology, Philadelphia, Pennsylvania, PA, USA:Saunders Elsevier, pp. 129- 141, 2011

Earlier, it was pointed out that the T wave is normally positive in all the standard bipolar limb leads and that this is caused by repolarization of the apex and outer surfaces of the ventricles ahead of the intraventricular surfaces. That is, the T wave becomes abnormal when the normal sequence of repolarization does not occur. Several factors can change this sequence of repolarization.Effect of Slow Conduction of the Depolarization Wave on the Characteristics of the T Wave

Referring back to Figure 12–14, note that the QRS complex is considerably prolonged. The reason for this prolongation is *delayed conduction in the left ventricle* resulting from left bundle branch block. This causes the left ventricle to become depolarized about 0.08 second after depolarization of the right ventricle, which gives a strong mean QRS vector *to the left.* However, the refractory periods of the right and left ventricular muscle masses are not greatly different from each other. Therefore, the right ventricle begins to

repolarize long before the left ventricle; this causes strong positivity in the right ventricle and negativity in the left ventricle at the time that the T wave is devel-oping. In other words, the mean axis of the T wave is now deviated *to the right,* which is opposite the mean electrical axis of the QRS complex in the same elec-trocardiogram. Thus, when conduction of the depolar-ization impulse through the ventricles is greatly delayed, the T wave is almost always of opposite polar-ity to that of the QRS complex.

7) "The Normal Electrocardiogram"

Guyton and J. Hall Guyton and Hall Textbook ofMedical Physiology, Philadelphia, Pennsyl-vania, PA, USA:Saunders Elsevier, pp. 121- 124, 2011.

A brief explanation is needed about several features of the 12th edition. Although many of the chapters have been revised to include new principles of physiology, the text length has been closely monitored to limit the book size so that it can be used effectively in physiology courses for medical students and health care professionals. Many of the figures have also been redrawn and are in full color. New references have been chosen primarily for their presentation of physiologic principles, for the quality of their own references, and for their easy accessibility. The selected bibliography at the end of the chapters lists papers mainly from recently published scientific journals that can be freely accessed from the PubMed internet site. Use of these references, as well as cross-references from them, can give the student almost complete coverage of the entire field of physiology. The effort to be as concise as possible has, unfortunately, necessitated a more simplified and dogmatic presentation of many physiologic principles than I normally would have desired. However, the bibliography can be used to learn more about the controversies and unanswered questions that remain in understanding the complex functions of the human body in health and disease. Another feature is that the print is set in two sizes. The material in large print constitutes the fundamental physiologic information that students will require in virtually all of their medical activities and studies. The material in small print is of several different kinds .first, anatomic, chemical, and other

information that is needed for immediate discussion but that most students will learn in more detail in other courses; second, physiologic information of special importance to certain fields of clinical medicine; and, third, information that will be of value to those students who may wish to study particular physiologic mechanisms more deeply.

I wish to express sincere thanks to many persons who have helped to prepare this book, including my colleagues in the Department of Physiology and Biophysics at the University of Mississippi Medical Center who provided valuable suggestions. The members of our faculty and a brief description of the research and educational activities of the department can be found at the web site: http:// physiology.umc.edu/. I am also grateful to Stephanie Lucas and Courtney Horton Graham for their excellent secretarial services, to Michael Schenk and Walter (Kyle) Cunningham for their expert artwork, and to William Schmitt, Rebecca Gruliow, Frank Morales, and the entire Elsevier Saunders team for continued editorial and production excellence. Finally, I owe an enormous debt to Arthur Guyton for the great privilege of contributing to the Textbook of Medical Physiology, for an exciting career in physiology, for his friendship, and for the inspiration that he provided to all who knew him.

8) "Machine Learning for Real-Time Heart Disease Prediction"

D. Bertsimas, L. Mingardi and B. Stellato, *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 9, pp. 3627- 3637, Sept. 2021.

Heart-related anomalies are among the most common causes of death worldwide. Patients are often asymptomatic until a fatal event happens, and even when they are under observation, trained personnel is needed in order to identify a heart anomaly. In the last decades, there has been increasing evidence of how Machine Learning can be leveraged to detect such anomalies, thanks to the availability of Electrocardiograms (ECG) in digital format. New developments in technology have allowed to exploit such data to build models able to analyze the patterns in the occurrence of heart beats, and spot anomalies from them. In this work, we propose a novel methodology to extract ECG-related features and predict the type of ECG recorded in real time (less than 30 milliseconds). Our models

leverage a collection of almost 40 thousand ECGs labeled by expert cardiologists across different hospitals and countries, and are able to detect 7 types of signals: Normal, AF, Tachycardia, Bradycardia, Arrhythmia, Other or Noisy. We exploit the XGBoost algorithm, a leading machine learning method, to train models achieving out of sample F1 Scores in the range 0.93 - 0.99. To our knowledge, this is the first work reporting high performance across hospitals,

9) "Cardiac Arrhythmias and Their Electrocar-diographic Interpretation"

Guyton and J. Hall, Guyton and Hall Textbook of Medical Physiology, Philadelphia, Pennsylvania, PA, USA:Saunders Elsevier, pp. 143-153, 2011.

Electrocardiogram (ECG) is a very powerful medical device, especially for heart related issues. It helps to understand how the heart works, and gives a helping hand in diagnosis of heart related problems, which is becoming very important in modern era. In this paper, we are proposing a portable android based device which is capable to store ECG signal through wireless medium for a long time for predicting the heart related artifacts and also to act as a prior warning system for the user. We are going to implement using Arduino Uno and HC-05 for bluetooth communication to make it portable and wireless. And we are using a SD card to record the real-time data which can be used to diagnose a disease or arrhythmia. It can display the real time ECG in the smart phone to the user, by the help of an android app made using MIT app inventor.

10) "A review of Arduino board's Lilypad's & Arduino shields"

Nayyar and V. Puri,, *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, vol. 2016, pp. 1485-1492.

The word “Open Source” is everywhere with Linux Technology and GNU foundation. In addition to open source software's and operating systems, Open Source Hardware is also progressing and becoming center point of attraction for researchers across the nook and

corner of the world. The most widely adopted Open Source hardware available right now is “Arduino”. Arduino has various products like boards, Lilypad's and shields. The aim of this research paper is to explore the world of Arduino technology in terms of Boards, Lilypad's and Shields covering in depth regarding-Technical Specifications, features and real-world applications.

Arduino technology has enabled various manufactures and research enthusiasts to come out with their own customized boards and shields as per their research requirements and area of implementations. Arduino Open Source community is also providing platform for researchers to come up with innovative research applications and market ready products in terms of Home Automation, Robotics, Wireless Connectivity, Drones and many others.

10) "Performance Evaluation of Supervised Machine Learning Algorithms in Prediction of Heart Disease",

P. Sujatha and K. Mahalakshmi, *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pp. 1-7, 2020.

Big challenge in health care industry is to record and analyze the massive amount of information about patients. Innovations in technologies made revolution in the healthcare industries.

In recent years the data analytics developed as promising tool for problem solving and decision making in healthcare professions. Data analytics process the data automatically to make healthcare system more dynamic and robust. It systematically uses and analyses the data of health care for better treatment with low costs. The chief applications of Machine learning in healthcare are the detection and diagnosis of diseases. The heart is the chief organ of human body.

Heart disease increases the mortality rate in the world. Around 90% of heart diseases are preventable. Machine learning plays a remarkable role in the health care industry in prediction of heart disease. In this research paper, the presence of heart disease is predicted by employing Decision Tree, Naïve Bayes, Random Forest, Support Vector

Machine, K-Nearest Neighbor and logistic Regression algorithms. The performance of the algorithms was analyzed using parameters such as Accuracy, Precision, AUC and F1-score.

From the experimental result, it is found that the Random Forest is more accurate for predicting the heart disease with accuracy of 83.52% compared with other supervised machine learning algorithms. The F1- Score, AUC and precision score of Random forest classifiers are 84.21%, 88.24% and 88.89% respectively.

11) "Prediction of Heart Disease Using Machine Learning",

A. Gavhane, G. Kokkula, I. Pandya and K. Devadkar *2018 Second International Conference on Electronics Communication and Aerospace Technology (ICECA)*, pp. 1275-1278, 2018

with the rampant increase in the heart stroke rates at juvenile ages, we need to put a system in place to be able to detect the symptoms of a heart stroke at an early stage and thus prevent it. It is impractical for a common man to frequently undergo costly tests like the ECG and thus there needs to be a system in place which is handy and at the same time reliable, in predicting the chances of a heart disease. Thus we propose to develop an application which can predict the vulnerability of a heart disease given basic symptoms like age, sex, pulse rate etc. The machine learning algorithm neural networks has proven to be the most accurate and reliable algorithm and hence used in the proposed system.

CHAPTER 3

SOFTWARE REQUIRMENTS SPECIFICATION

Android Studio:

- Android studio 2021.2.1 (Chipmunk)

Libraries

- Flutter UI
- Wear OS
- ChannelClient
- DataClient

Operating System

- Windows 10

Hardware Requirements Specification

- Laptop with basic hardware
- Android
- Smart Watch

CHAPTER 4

SYSTEM ANALYSIS

SYSTEM REVIEW

This survey is done to comprehend the need and prerequisite of the general population who has Heart Problems, and to do as such, we went through different sites and applications and looked for the fundamental data. Based on these data, we madean audit that helped us get new thoughts and make different arrangements for ourtask. We reached the decision that there is a need of such application and felt thatthere is a decent extent of progress in this field too.

TECHNOLOGY USED

ANDROID STUDIO

Android Studio is the official IDE for android application development.It works based on **IntelliJ IDEA**, You can download the latest version of android studio from Android Studio 2.2 Download, If you are new to installing Android Studio on windows,you will find a file, which is named as *android-studio-bundle-143.3101438-windows.exe*.So just download and run on windows machine according to android studio wizard guideline.

If you are installing Android Studio on Mac or Linux, You can download the latest version from Android Studio Mac Download,or Android Studio Linux Download, check the instructions provided along with the downloaded file for Mac OS and Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 8.1 operating system.

- It is available for download on Windows, macOS and Linux based operating systems.

It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

- Android Studio was announced on May 16, 2003 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.
- On 7 May 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported in android app development using Android Studio.

ANDROID STUDIO INSTALLATION

Step 1: Get the Android Studio executable or zip file.

Step 2: Download Android Studio.

Step 3: After the downloading has finished, open the file and run it. It will prompt the welcome dialogue box. Click on next.

Step 4: It will start the installation, and once it is completed, it will be displays like Installation complete. Click on next.

Step 5: Once "Finish" is clicked, it will ask whether the previous settings needs to be imported [if android studio had been installed earlier], or not. It is better to choose the 'Don't import settings option. Click the OK button.

Step 6: This will start the Android Studio. Meanwhile it will be finding the available SDK components.

Step 7: After it has found the SDK components, it will redirect to the welcome dialogue box. Click on next.

- Choose standard and click on next.

- Now choose the theme, whether light theme or the dark one. The light one is called the IntelliJ theme whereas the dark theme is called Darcula. Choose as required.
- Click on next button.

Step 8: Now it is time to download the SDK components. Click on finish. It has started downloading the components. The Android Studio has been successfully configured. Now it's time to launch and build apps. Click on the finish button to launch it.

Step 9: Click on 'Start new android project' to build a new app.

Libraries

- Flutter UI

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia,and the web from a single codebase. The first version of Flutter was known as "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, the last major release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first stable version of the framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

In Android , the View is the foundation of everything that shows up on the screen. Buttons, toolbars, and inputs, everything is a View. In Flutter, the rough equivalent to a View is a Widget. Widgets don't map exactly to Android views, but while you're getting acquainted with how Flutter works you can think of them as "the way you declare and construct UI".

However, these have a few differences to a View. To start, widgets have a different lifespan: they are immutable and only exist until they need to be changed. Whenever widgets or their state change, Flutter's framework creates a new tree of widget instances. In comparison, an Android view is drawn once and does not redraw until invalidate is called.

Flutter's widgets are lightweight, in part due to their immutability. Because they aren't views themselves , and aren't directly drawing anything, but rather are a description of the UI and its semantics that get "inflated" into actual view objects under the hood.

Flutter includes the Material Components library. These are widgets that implement the Material Design guidelines. Material Design is a flexible design system optimized for all platforms, including iOS.

But Flutter is flexible and expressive enough to implement any design language. For example, on iOS, you can use the Cupertino widgets to produce an interface that looks like Apple's iOS design.

On May 6, 2020, the Dart software development kit (SDK) version 2.8 and Flutter 1.17.0 were released, adding support for the Metal API which improves performance on iOS devices by approximately 50%, as well as new Material widgets and network tracking development tools.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with a new CanvasKit renderer and web specific widgets, early-access desktop application support for Windows, macOS, and Linux and improved Add-to-App APIs. This release also utilized Dart 2.0 that featured sound null-safety, which caused many breaking changes and issues with many external packages; however, the Flutter team included instructions and tools to

mitigate these issues.

On September 8th, 2021, Dart 2.14 and Flutter 2.5 were released by Google. The update brought improvements to the Android full-screen mode and the latest version of Google's Material Design called Material You. Dart received two new updates, standardizing lint conditions and marking support for Apple Silicon as stable

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS. The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

- Wear OS

Wear OS (also known simply as Wear and formerly Android Wear) is a version of Google's Android operating system designed for smartwatches and other wearables. By pairing with mobile phones running Android version 6.0 "Marshmallow" or newer, or iOS version 10.0 or newer with limited support from Google's pairing application, Wear OS integrates Google Assistant technology and mobile notifications into a smartwatch form factor.

Wear OS supports Bluetooth, NFC, Wi-Fi, 3G, and LTE connectivity, as well as a range of features and applications. Watch face styles include round, square and rectangular. Hardware manufacturing partners include Asus, Broadcom, Fossil, HTC, Intel, LG, MediaTek,

Imagination Technologies, Motorola, New Balance, Qualcomm, Samsung, Huawei, Skagen, Polar, TAG Heuer, Suunto, and Mobvoi.

The platform was announced on March 18, 2014, along with the release of a developer preview. At the same time, companies such as Motorola, Samsung, LG, HTC and Asus were announced as partners. On June 25, 2014, at Google I/O, the Samsung Gear Live and LG G Watch were launched, along with further details about Android Wear. The LG G Watch is the first Android Wear smartwatch to be released and shipped. Motorola's Moto 360 was released on September 5, 2014.

Features

Wear OS can synchronize notifications from a paired device, and supports voice control with the "OK Google" hotword along with gesture-based input. Wear OS integrates with Google services such as the Google Assistant and Google mobile services (including Gmail, Google Maps, and Google Pay), as well as third-party watch apps from Play Store.

From the watch face, the user can swipe up to access their notifications, down to access a quick settings panel, from the left to view their personalized Google feed, and the right to view Google Fit.[39] Via Google Fit and similar applications, Wear OS supports ride and run tracking, and devices containing heart rate sensors can perform a reading on-demand, or at intervals throughout the day. The watch can control media being played on streamed on paired devices.

ANDROID APP

Android app we develop utilizes Flutter UI. In general, developing a mobile application is a complex and challenging task. There are many frameworks available to develop a mobile application. Android provides a native framework based on Java language and iOS provides a native framework based on Objective- C / Swift language.

However, to develop an application supporting both the OSs, we need to code in two different languages using two different frameworks. To help overcome this complexity, there exists mobile frameworks supporting both OS. These frameworks range from simple HTML based hybrid mobile application framework (which uses HTML for User Interface and JavaScript for application logic) to complex language specific framework (which do the heavy lifting of converting code to native code). Irrespective of their simplicity or complexity, these frameworks always have many disadvantages, one of the main drawback being their slow performance.

In this scenario, Flutter – a simple and high performance framework based on Dart language, provides high performance by rendering the UI directly in the operating system's canvas rather than through native framework.

Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environment and designing the application using widgets is as simple as designing HTML.

To be specific, Flutter application is itself a widget. Flutter widgets also supports animations and gestures. The application logic is based on reactive programming. Widget may optionally have a state. By changing the state of the widget, Flutter will automatically (reactive programming) compare the widget's state (old and

new) and render the widget with only the necessary changes instead of re-rendering the whole widget.

We shall discuss the complete architecture in the coming chapters.

SMART WATCH SYNC

Wear OS apps run directly on a watch, giving you access to hardware such assensors and the GPU. Wearable apps are similar to other apps that use the Android SDK, but differ in design and functionality.

A Wear OS app should work independently of a phone app, allowing users the greatest flexibility in their choice of phones.

smartwatch is a wearable computer in the form of a watch; modern smartwatches provide a local touchscreen interface for daily use, while an associated smartphone app provides for management and telemetry (such as long-term biomonitoring). While early models could perform basic tasks, such as calculations, digital time telling, translations, and game-playing, 2010s smartwatches have more general functionality closer to smartphones, including mobile apps, a mobile operating system and WiFi/Bluetooth connectivity.

Some smartwatches function as portable media players, with FM radio and playback of digital audio and video files via a Bluetooth headset. Some models, called watch phones (or vice versa), have mobile cellular functionality like making calls.

Many smartwatch smartphone models manufactured in the 2010s are completely functional as standalone products. Some are used in sports, the GPS tracking unit being used to record historical data. For example, after a workout, data can be uploaded onto a computer or

online to create a log of activities for analysis or sharing. Some watches can serve as full GPS watches, displaying maps and current coordinates, and recording tracks. Users can "mark" their current location and then edit the entry's name and coordinates, which enables navigation to those new coordinates.

As companies add competitive products into the market, media space is becoming a desired commodity on smartwatches. With Apple, Sony, Samsung, and Motorola introducing their smartwatch models, 15 percent of tech consumers use wearable technologies. The functionality of these watches attracts tech consumers with buying power, which has attracted advertisers.

Mobile advertising on wearable devices was expected to increase heavily by 2017 as advanced hypertargeting modules were introduced to the devices. In order for an advertisement to be effective on a smartwatch, companies have stated that each ad must be able to create an experience native to the smartwatch itself.

A smartwatch is a wearable computer in the form of a watch; modern smartwatches provide a local touchscreen interface for daily use, while an associated smartphone app provides for management and telemetry (such as long-term biomonitoring). While early models could perform basic tasks, such as calculations, digital time telling, translations, and game-playing, 2010s smartwatches have more general functionality closer to smartphones, including mobile apps, a mobile operating system and WiFi/Bluetooth connectivity. Some smartwatches function as portable media players, with FM radio and playback of digital audio and video files via a Bluetooth headset. Some models, called watch phones (or vice versa), have mobile cellular functionality like making calls.

Many smartwatch smartphone models manufactured in the 2010s are completely functional as standalone products. Some are used in sports, the GPS tracking unit being used to record historical data. For example, after a workout, data can be uploaded onto a computer or online to create a log of activities for analysis or sharing. Some watches can serve as full GPS

watches, displaying maps and current coordinates, and recording tracks. Users can "mark" their current location and then edit the entry's name and coordinates, which enables navigation to those new coordinates. As companies add competitive products into the market, media space is becoming a desired commodity on smartwatches. With Apple, Sony, Samsung, and Motorola introducing their smartwatch models, 15 percent of tech consumers use wearable technologies. The functionality of these watches attracts tech consumers with buying power, which has attracted advertisers. Mobile advertising on wearable devices was expected to increase heavily by 2017 as advanced hypertargeting modules were introduced to the devices. In order for an advertisement to be effective on a smartwatch, companies have stated that each ad must be able to create an experience native to the smartwatch itself.

Wear OS is a version of Google's Android operating system designed for smartwatches and other wearables. By pairing with mobile phones running Android version 6.0 "Marshmallow" or newer, or iOS version 10.0 or newer with limited support from Google's pairing application, Wear OS integrates Google Assistant technology and mobile notifications into a smartwatch form factor.

Wear OS supports Bluetooth, NFC, Wi-Fi, 3G, and LTE connectivity, as well as a range of features and applications. Watch face styles include round, square and rectangular. Hardware manufacturing partners include Asus, Broadcom, Fossil, HTC, Intel, LG, MediaTek, Imagination Technologies, Motorola, New Balance, Qualcomm, Samsung, Huawei, Skagen, Polar, TAG Heuer, Suunto, and Mobvoi.

In the first six months of availability, Canalys estimates that over 720,000 Android Wear smartwatches were shipped. As of 15 March 2018, Wear OS had between 10 and 50 million application installations. Wear OS was estimated to account for 10% of the smartwatch market in2015.

CLOUD STORAGE

Cloud storage is a digital storage solution that utilizes multiple servers (typically across multiple locations) to safely store files (such as site backups). In the past few years, cloud storage has grown in popularity and become a direct challenger to local storage, mainly down to the benefits it provides:

1. **Security.** Your site backups will be located off-site and across multiple servers. This means your backup is better protected from data loss or hacking than if it were stored on a local server.
2. **Accessibility.** As it's accessible online, you (and your team) can access site backups whenever you need them, regardless of your location. This isn't possible with local storage!
3. **No maintenance required.** Given that cloud servers are maintained by a separate company, you won't need to hire knowledgeable IT staff to maintain the server. This will save you thousands of dollars in the long run.

Cloud storage is a model of computer data storage in which the digital data is stored in logical pools, said to be on "the cloud". The physical storage spans multiple servers (sometimes in multiple locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment secured, protected, and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data.

Cloud storage services may be accessed through a colocated cloud computing service, a web service application programming interface (API) or by applications that use the API, such as cloud desktop storage, a cloud storage gateway or Web-based content management systems.

Architecture

Cloud storage is based on highly virtualized infrastructure and is like broader cloud computing in terms of interfaces, near-instant elasticity and scalability, multi-tenancy, and metered resources. Cloud storage services can be used from an off-premises service (Amazon S3) or deployed on-premises (ViON Capacity Services).

There are three types of cloud storage: a hosted object storage service, file storage, and block storage. Each of these cloud storage types offer their own unique advantages.

Examples of object storage services that can be hosted and deployed with cloud storage characteristics include Amazon S3, Oracle Cloud Storage and Microsoft Azure Storage, object storage software like Openstack Swift, object storage systems like EMC Atmos, EMC ECS and Hitachi Content Platform, and distributed storage research projects like OceanStore and VISION Cloud.

Examples of file storage services include Amazon Elastic File System (EFS) and Qumulo Core, used for applications that need access to shared files and require a file system. This storage is often supported with a Network Attached Storage (NAS) server, used for large content repositories, development environments, media stores, or user home directories.

A block storage service like Amazon Elastic Block Store (EBS) is used for other enterprise applications like databases and often require dedicated, low latency storage for each host. This is comparable in certain respects to direct attached storage (DAS) or a storage area network (SAN).

Cloud storage is

1. Made up of many distributed resources, but still acts as one, either in a federated or a cooperative storage cloud architecture
2. Highly fault tolerant through redundancy and distribution of data
3. Highly durable through the creation of versioned copies
4. Typically eventually consistent with regard to data replicas

Advantages

1. Companies need only pay for the storage they actually use, typically an average of consumption during a month, quarter, or year. This does not mean that cloud storage is less expensive, only that it incurs operating expenses rather than capital expenses.

2. Businesses using cloud storage can cut their energy consumption by up to 70% making them a more green business.

3. Organizations can choose between off-premises and on-premises cloud storage options, or a mixture of the two options, depending on relevant decision criteria that is complementary to initial direct cost savings potential; for instance, continuity of operations (COOP), disaster recovery (DR), security (PII, HIPAA, SARBOX, IA/CND), and records retention laws, regulations, and policies.

4. Storage availability and data protection is intrinsic to object storage architecture, so depending on the application, the additional technology, effort and cost to add availability and protection can be eliminated.

5. Storage maintenance tasks, such as purchasing additional storage capacity, are offloaded to the responsibility of a service provider.

6. Cloud storage provides users with immediate access to a broad range of resources and applications hosted in the infrastructure of another organization via a web service interface.

7. Cloud storage can be used for copying virtual machine images from the cloud to on-premises locations or to import a virtual machine image from an on-premises location to the cloud image library. In addition, cloud storage can be used to move virtual machine images between user accounts or between data centers.

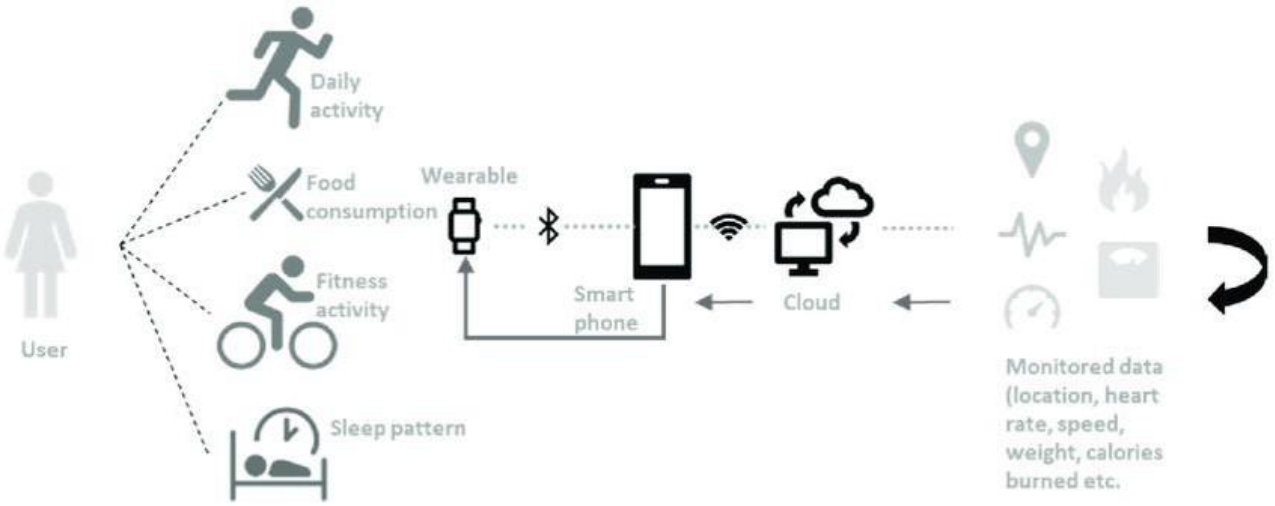
8. Cloud storage can be used as natural disaster proof backup, as normally there are 2 or 3 different backup servers located in different places around the globe.

9. Cloud storage can be mapped as a local drive with the WebDAV protocol. It can function as a central file server for organizations with multiple office locations.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM DIAGRAM



CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 HEART DISEASE PREDICTION

Heart attack is one of the leading causes of human death worldwide. Every year, about 610,000 people die of heart attack in the United States alone—that is one in every four deaths—but there are well understood early symptoms of heart attack that could be used to greatly help in saving many lives and minimizing damages by detecting and reporting at an early stage.

On the other hand, every year, about2.35 million people get injured or disabled from road accidents. Unexpectedly, many of these fatal accidents happen due to the heart attack of drivers that leads to the loss of control of the vehicle.

The current work proposes the development of a wearable system for real-time detection and warning of heart attacks in drivers, which could be enormously helpful in reducing road accidents.The system consists of two subsystems that communicate wirelessly using Bluetooth technology, namely, a wearable sensor subsystem and an intelligent heart attack detection and warning subsystem.

The sensor subsystem records the electrical activity of the heart from the chest area to produce electrocardiogram (ECG) trace and send that to the other portable decision-making subsystem where the symptoms of heart attack are detected. We evaluated the performance of dry electrodes and different electrode configurations and measured overall power consumption of the system.

Linear classification and several machine algorithms were trained and tested for real-time application. It was observed that the linear classification algorithm was not able to detect heart attack in noisy data, whereasthe support vector machine (SVM) algorithm with polynomial kernel with extended time–frequency features using extended modified B-distribution (EMBD) showed highest accuracy and was able to detect 97.4% and 96.3% of ST-elevation myocardial infarction (STEMI) and non-ST-elevation MI. (NSTEMI), respectively.

The proposed system can therefore help in reducing the loss of lives from the growing number of heart alignments all over the world..

Among all fatal disease, heart attacks diseases are considered as the most prevalent. Medical practitioners conduct different surveys on heart diseases and gather information of heart patients, their symptoms and disease progression.

Increasingly are reported about patients with common diseases who have typical symptoms. In this fast moving world people want to live a very luxurious life so they work like a machine in order to earn lot of money and live a comfortable life therefore in this race they forget to take care of themselves, because of this there food habits change their entire lifestyle change, in this type of lifestyle they are more tensed they have blood pressure, sugar at a very young age and they don't give enough rest for themselves and eat what they get and they even don't bother about the quality of the food if sick the go for their own medication as a result of all these small negligence it leads to a major threat that is the heart disease The term 'heart disease' includes the diverse diseases that affect heart

. The number of people suffering from heart disease is on the rise (health topics, 2010). The report from world health organization shows us a large number of people that die every year due to the heart disease all over the world. Heart disease is also stated as one of the greatest killers in Africa. Data mining has been used in a variety of applications such as marketing, customer relationship management, engineering, and medicine analysis, expert prediction, web mining and mobile computing. Of late, data mining has been applied successfully in healthcare fraud and detecting abuse cases.

Heart disease can be managed effectively with a combination of lifestyle changes, medicine and, in some cases, surgery. With the right treatment, the symptoms of heart disease can be reduced and the functioning of the heart improved.

The predicted results can be used to prevent and thus reduce cost for surgical treatment and other expensive. The overall objective of my work will be to predict

accurately with few tests and attributes the presence of heart disease. Attributes considered form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with few attributes and faster efficiency the risk of having heart disease. Decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the data set and databases.

This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. Data mining holds great potential for the healthcare industry to enable health systems to systematically use data and analytics to identify inefficiencies and best practices that improve care and reduce costs. According to (Wurz & Takala, 2006) the opportunities to improve care and reduce costs concurrently could apply to as much as 30% of overall healthcare spending. The successful application of data mining in highly visible fields like e-business, marketing and rail has led to its application in other industries and sectors.

PROGRAMS AND FEATURES

1. A personalized experience

When it's about tracking the user's daily activity and further evaluating the user's fitness, the application needs to have user's personal information like, age, height, gender, and weight.

These will help the application in ascertaining the accurate details.

With these, allow users to customize the application as per their need. You may include essential features like:

- Planning personalized exercise routines
- Saving the statistics he/she gets
- Setting goals
- Comparing the performance/progress

2. Customized Diet Plans

Customized diet plans are tailored as per your specific energy consumption needs. Also, a customized diet is designed just for you. So if you have any preferences, food intolerances, specific requirements, or food allergies, they are all taken into consideration. You will be more likely to stick to a plan that includes the kinds of foods you enjoy.

With customized diet plans, you can effortlessly avoid the pitfalls that come with simply browsing through the market with no obvious intention or purpose.

That's why the shopping list and recipes that come along with a customized diet plan are so helpful. They save you time by making the planning and shopping experience easier and more purposeful for you.

3. Nutrition Tracking

Nutrition is the core of every fitness activity. There are many nutrition tracking apps available in the market that allow your users to control their weight by providing them with the data of burned calories, controlling water balance and motivating the

users to encourage healthy food habits.

Personal goal setting is one of the basic features of fitness apps. If a person is facing trouble with the time sticking to healthy nutrition, the app will help to create grocery shopping lists and even collect healthy food recipes.

4. Push notification system & reminders

Anyone who is working out regularly would never want to miss a session. However, keeping up with our busy lives; sometimes, it does slips out of our minds. In such cases, a push notification from the app can be very useful to remind us of our workout session.

This is the very reason why many Fitness app development companies are investing a lot of capital and time in the notification system. However, they must keep one thing in mind: Push Notifications work as a double-edged sword.

5. Goal Setting

Fitness apps are aimed at helping users to bring measurable results from their daily workouts. The user sets the desired results themselves, so the process of goal setting should be simple and clear.

Besides this, fitness apps also focus on sports activities and nutritional goal settings. For effective utilisation of fitness apps, the user needs to select a daily task, input goal value, or you can decide a specific date to achieve the goal. Furthermore, the fitness apps show the task performance statistics.

6. Deep heart monitoring

For many years doctors, athletes, and astronauts are using heart rate monitors to measure stress and other health metrics. Nowadays, a majority of fitness apps use optical heart rate monitors.

7. Progress Tracking

You cannot able to achieve your fitness goals properly unless you do not measure its

progress. Repetitions, sets, calories, hours, kilometres, kilograms, pounds – everything can be counted.

The app owners can track their progress in the aforesaid measurable units, become motivated by this information, and continue using the app to achieve more. The most important thing here is that fitness apps do not track unnecessary amounts of data to avoid data overload. A graphical overview of the metrics provides the entire fitness routine summary to the user at their fingertips, including the route map with the colour codes showing the most active segments.

8. Sleep tracker

This is must add feature looking at today's busy lifestyles. The feature tracks the time user didn't get the sound sleep. This implies the moments he is active during sleep

NOTIFICATION

Once the heart alignment is identified in the cloud then the user is notified using Firebase Module used for Notifications. For Initial or small hearth alignments the user is identified directly via message or android app notification. If the heart problem is usage were the user cannot access like myocardial infarction the users relative will receive notification.

REPORT GENERATION

If the user needs a report of his conditions then he/she can access the data via the User Interface. The time and date of the heart problem will be produced as a ECG which can be downloaded as PDF. This report can be given to the heart specialist and the problem can be predicted at the initial stage. By using this future problems can be reduced.

CHAPTER 7

CONCLUSION AND FURTURE WORK

The objective of this thesis was to develop a compact ECG monitoring device. All the steps for the development of this project have been described in this thesis.

We have discussed every aspect of the project from analog front end design to Android phone application which is the most significant part of the project. Struggle for design of analog front end has shown new ways of signal processing.

We have tried to present a clear ECG signal keeping in view all bandwidth and noise requirements. The advancement in mobile and wireless health care solutions is contributing in different aspects of our lives that range from diagnosis to treatment of different diseases like cardiovascular disease.

Android applications are also a part of diverse solutions that are offered as healthcare solutions together with a variety of health monitoring devices..

The result of the project has achieved part of the objectives that were set at the beginning. The bachelor's thesis was an opportunity to extend and discover new skills in Mobile application development. The final result of the application can also be adapted as a reference for Bluetooth application development in the android platform.

The Paper lays out a foundation for future developments that are focused on wireless health solutions in the android platform. Some of the features that can be included are integrated SMS (texting) functions, options for saving the data on the device or data streaming function that can work with a control server in a health center. These features will help.

clinicians to monitor patients efficiently and therefore improve service. It is also important to include more options like zooming functionality, which will

improve the usability of the app. In addition, the application can be customized to offer more business opportunities. For example; it is possible to add GPS tracker features that can be used during exercises like cycling and running.

CHAPTER 8

APPENDIX

Android App – Main Activity

```
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class MainActivity extends AppCompatActivity {
   BottomNavigationView bottomNavigationView;
   int code = 1;
   String table;
   private boolean exit = false;
   private void initViews() {
      bottomNavigationView = findViewById(R.id.bottomNavigation);
   }

   private void bottomNav() {


getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, new Goal()).commit();
      bottomNavigationView.setOnItemSelectedListener(item -> {
         Fragment selectedFragment = null;
         switch (item.getItemId()) {
            case R.id.navGoal:
               selectedFragment = new Goal();
```

```
                break;
            case R.id.navProfile:
                selectedFragment = new Profile();
                break;
            case R.id.navWorkout:
                selectedFragment = new Workout();
                break;
            case R.id.navTrack:
                selectedFragment = new Track();
                break;
        }

getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, selectedFragment).commit();

        return true;
    });

}

private void bottomNavClick() {
    bottomNavigationView.setOnItemSelectedListener(item -> {


        Fragment selectedFragment = null;
        switch (item.getItemId()) {
            case R.id.navGoal:
                selectedFragment = new Goal();
                break;
            case R.id.navProfile:
                selectedFragment = new Profile();
                break;
            case R.id.navWorkout:
                selectedFragment = new Workout();
                break;
            case R.id.navTrack:
                selectedFragment = new Track();
                break;
        }

//          replaceFragment(selectedFragment);
```

```
getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, selectedFragment).commit();

            return true;
        });
    }

    private void fragmentNavigation() {
        switch (code) {
            case 2:
                bottomNavigationView.setSelectedItemId(R.id.navTrack);

                Bundle bundle = new Bundle();
                bundle.putString("tableName", table);

                Track track = new Track();
                track.setArguments(bundle);


getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, track).commit();
                break;
            case 3:
                bottomNavigationView.setSelectedItemId(R.id.navWorkout);

getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, new Workout()).commit();
                break;
            case 4:
                bottomNavigationView.setSelectedItemId(R.id.navProfile);

getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, new Profile()).commit();
                break;
            default:
                bottomNavigationView.setSelectedItemId(R.id.navGoal);

getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContai
nerView, new Goal()).commit();
                break;
        }
    }
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        Intent intent = getIntent();
        table = intent.getStringExtra("tableName");
        code = intent.getIntExtra("code", 0);


        initViews();

        bottomNavClick();
        fragmentNavigation();

    }

    @Override
    public void onBackPressed() {

        if (exit) {
            super.onBackPressed();
        }

        exit = true;
        Toast.makeText(this, "Tap/Slide again to exit",
Toast.LENGTH_SHORT).show();

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                exit = false;
            }
        }, 2000);
    }
}
```

USER DATA

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.material.button.MaterialButton;
import com.google.android.material.button.MaterialButtonToggleGroup;
import com.google.android.material.textfield.TextInputLayout;

public class PersonDetails extends AppCompatActivity {

   private EditText editTextAge, editTextWeight, editTextHeight;
   private MaterialButton btnNext;
   private MaterialButtonToggleGroup maleFemaleToggle;
   private String gender = "gender";
   public void initViews() {
      editTextAge = findViewById(R.id.editTextAge);
      editTextWeight = findViewById(R.id.editTextWeight);
      editTextHeight = findViewById(R.id.editTextHeight);
      btnNext = findViewById(R.id.btnNext);
      maleFemaleToggle = findViewById(R.id.maleFemaleToggle);
   }

   public void maleFemaleToggle() {
      maleFemaleToggle.addOnButtonCheckedListener(new
MaterialButtonToggleGroup.OnButtonCheckedListener() {
         @Override
         public void onButtonChecked(MaterialButtonToggleGroup group, int
checkedId, boolean isChecked) {
            if (isChecked) {
               switch (checkedId) {
                  case R.id.toggleMale:
```

```
                    gender = "Male";
                    break;
                case R.id.toggleFemale:
                    gender = "Female";
                    break;
            }
        }
    }
});
}

public void onBtnProceedClick() {

    Boolean notOkay = editTextAge.getText().toString().trim().equals("")
            || editTextHeight.getText().toString().trim().equals("")
            || editTextWeight.getText().toString().trim().equals("")
            || gender.equals("gender");

        if (notOkay) {
            Toast.makeText(PersonDetails.this, "Error! field empty",
Toast.LENGTH_SHORT).show();

        } else {
            int age = Integer.parseInt(editTextAge.getText().toString());
            float weight = Float.parseFloat(editTextWeight.getText().toString());
            float height = Float.parseFloat(editTextHeight.getText().toString());

            SharedPreferences personData = getSharedPreferences("personData",
MODE_PRIVATE);
            SharedPreferences.Editor editor = personData.edit();


            editor.putInt("age", age);
            editor.putFloat("weight", weight);
            editor.putFloat("height", height);
            editor.putString("gender", gender);
            editor.apply();

            Intent intent = new Intent(PersonDetails.this, PersonGoal.class);
            startActivity(intent);
        }
}
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_person_details);

        initViews();
        maleFemaleToggle();

        btnNext.setOnClickListener(v -> {
            onBtnProceedClick();
        });
    }
}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.blueserial"
android:versionCode="1"
android:versionName="1.0" >

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

<uses-sdk
android:minSdkVersion="10"
android:targetSdkVersion="19" />

<application
android:allowBackup="true"
android:icon="@drawable/logo"
android:label="@string/app_name"
android:name="com.blueserial.MyApplication"
android:theme="@style/AppTheme" >
<activity
android:name="com.blueserial.MainActivity"
android:label="@string/app_name"
android:configChanges="orientation"
```

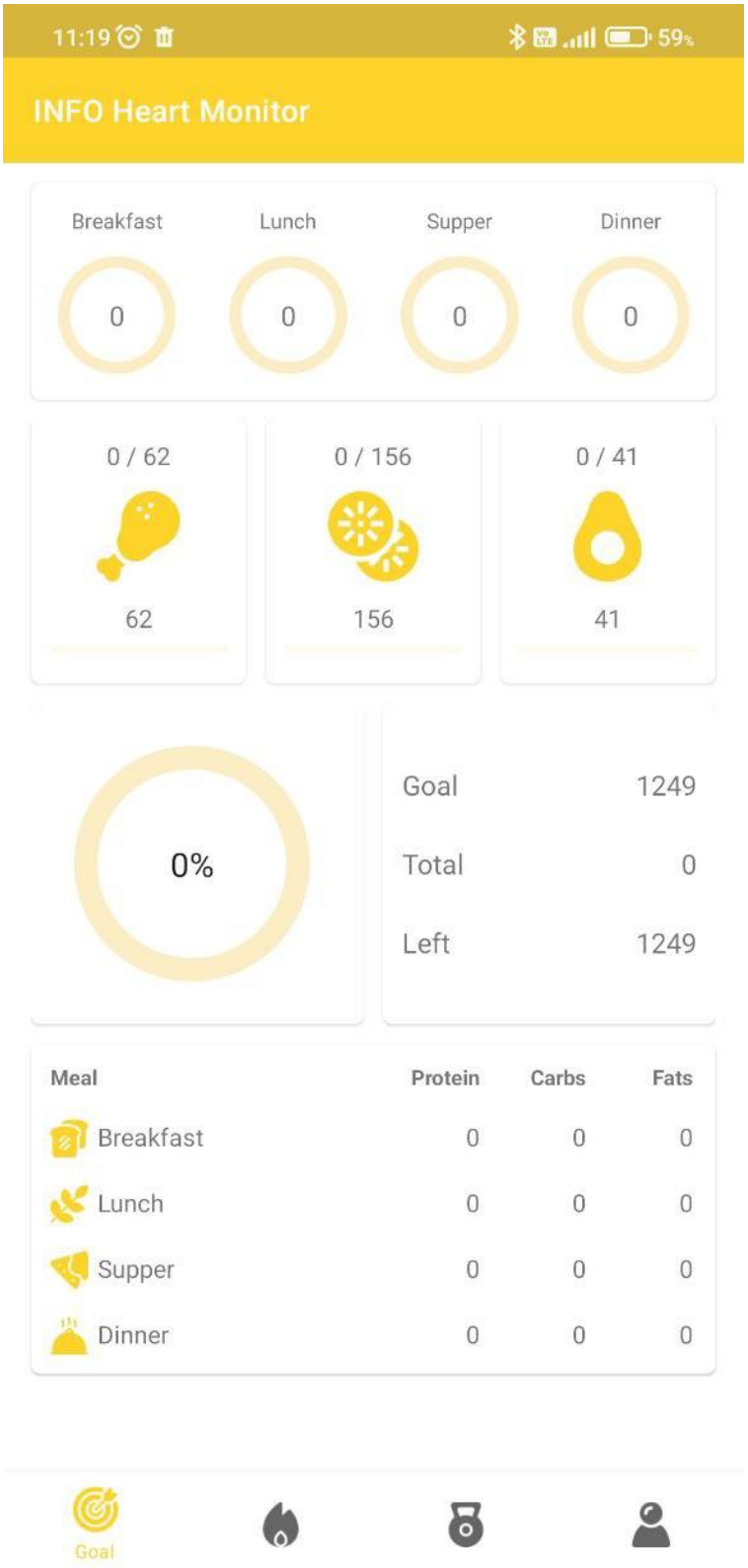
```
android:windowSoftInputMode="adjustResize|stateHidden" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
</intent-filter>
</activity>
<activity
android:name="com.blueserial.Homescreen"
android:configChanges="keyboardHidden|orientation"
android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity android:name="PreferencesActivity"></activity>
</application>

</manifest>
```

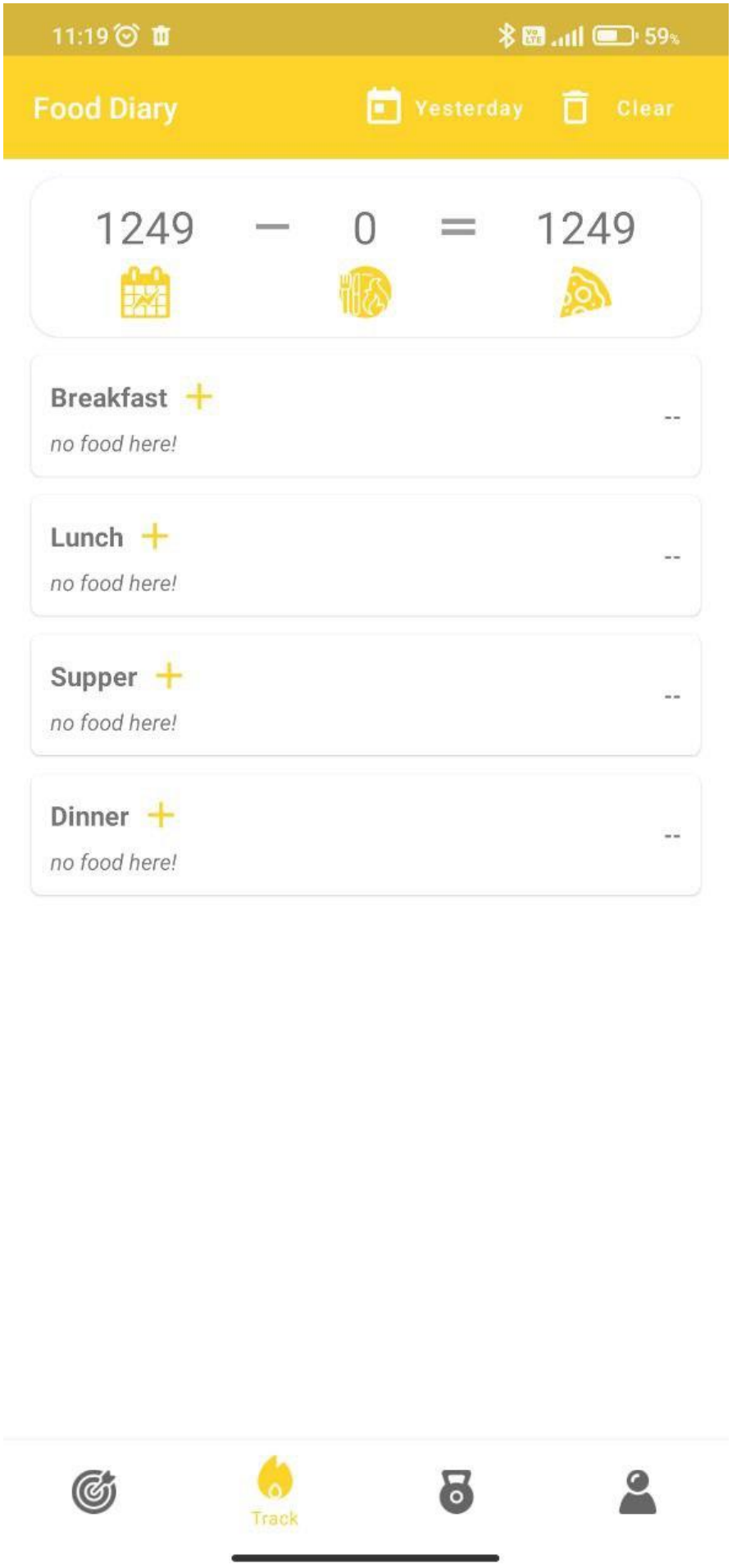
activity_main.xml:

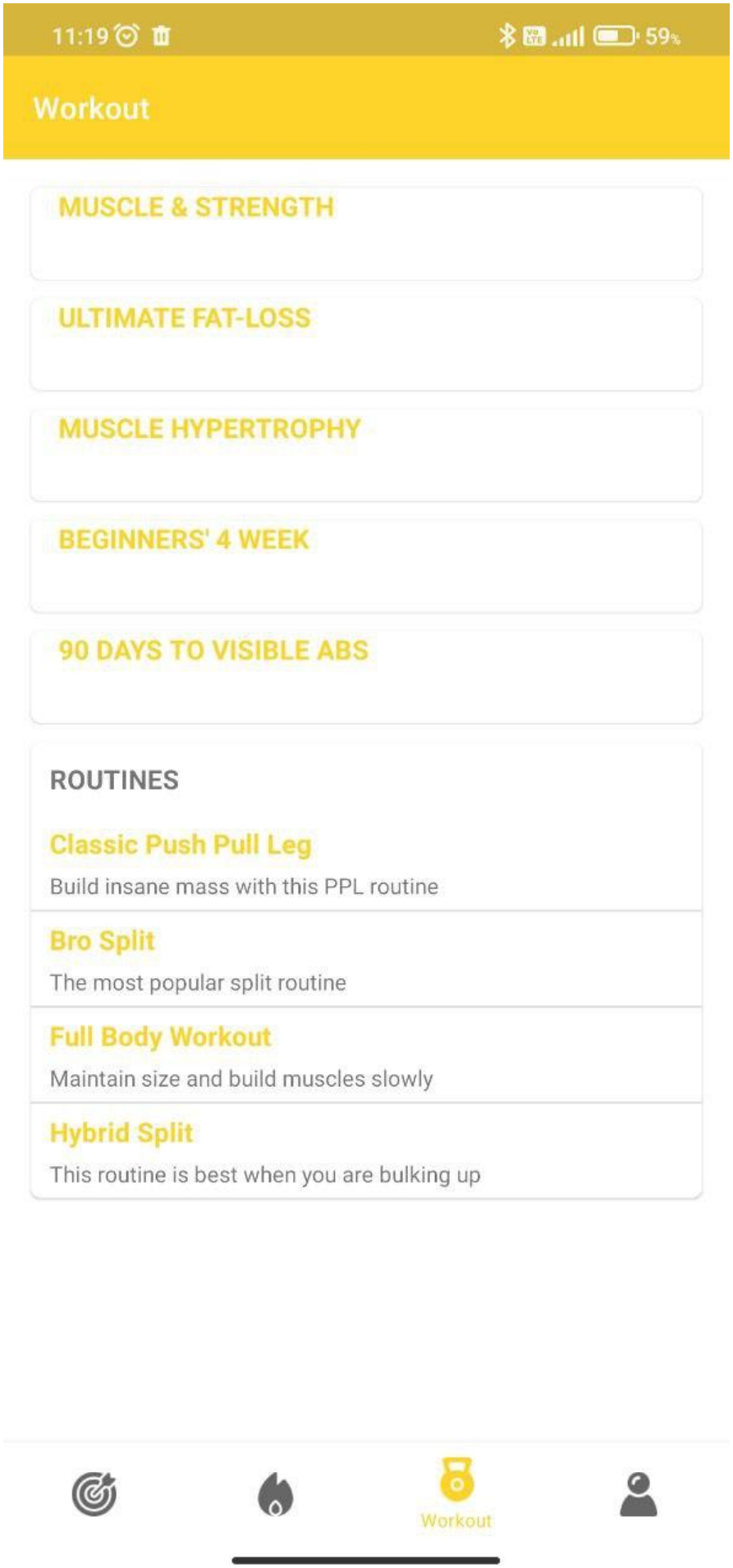
```
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools" android:id="@+id/drawer_layout"
android:layout_width="match_parent" android:layout_height="match_parent"
tools:context=".MainActivity">

<!-- As the main content view, the view below consumes the entire
space available using match_parent in both dimensions. -->
<FrameLayoutandroid:id="@+id/container" android:layout_width="match_parent"
android:layout_height="match_parent" />

<!--android:layout_gravity="start" tells DrawerLayout to treat
this as a sliding drawer on the left side for left-to-right
languages and on the right side for right-to-left languages.
        If you're not building against API 17 or higher, use
android:layout_gravity="left" instead. -->
```

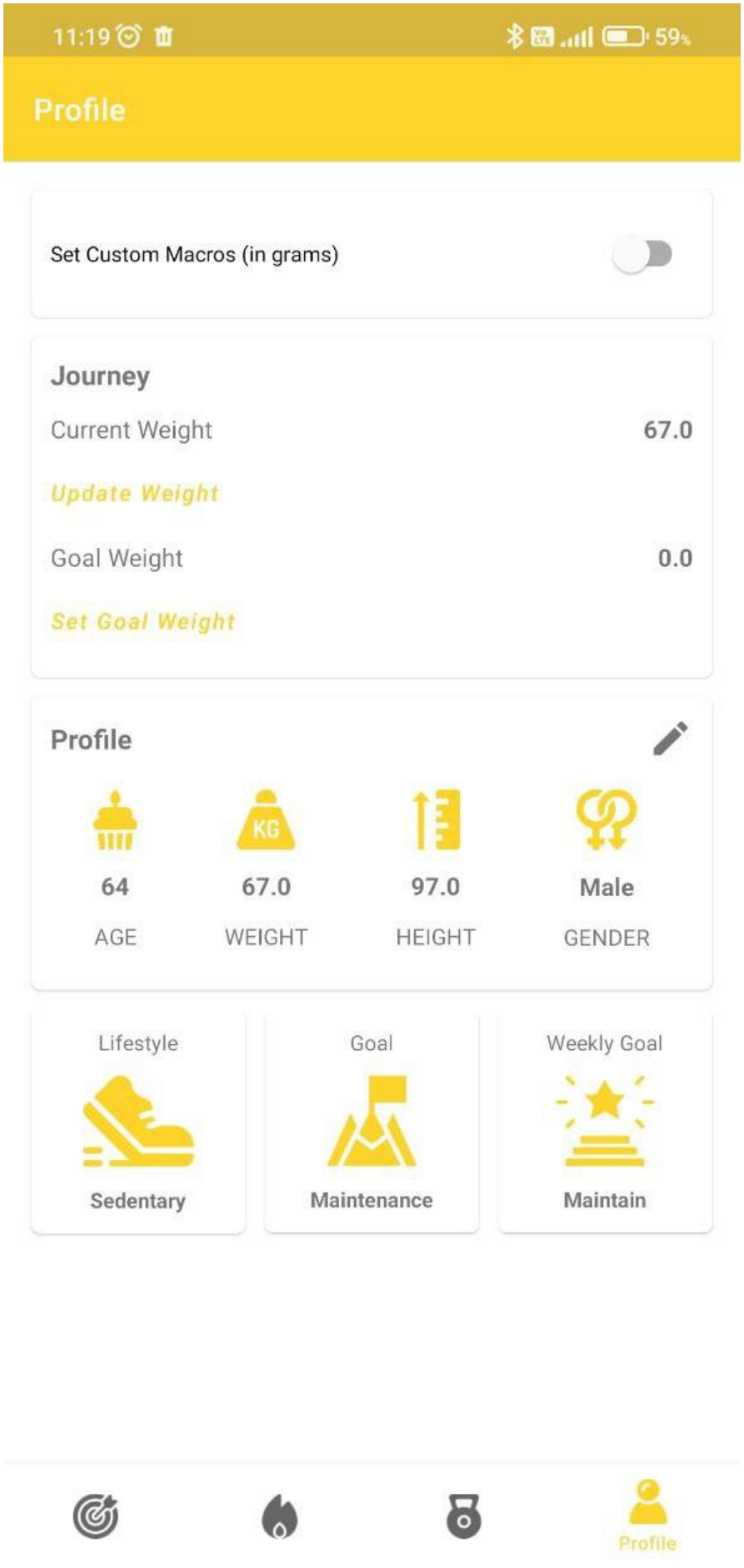
```
<!-- The drawer is given a fixed width in dp and extends the full height of
the container. -->
<fragment android:id="@+id/navigation_drawer"
android:layout_width="@dimen/navigation_drawer_width"
android:layout_height="match_parent"
android:layout_gravity="start"
android:name="com.adithya.healthcare.ecgmonitor.NavigationDrawerFragment"
tools:layout="@layout/fragment_navigation_drawer" />
<fragment android:id="@+id/devices_fragment"
android:layout_width="match_parent" android:layout_height="match_parent"
android:layout_gravity="start"
android:name="com.adithya.healthcare.ecgmonitor.DevicesFragment"
tools:layout="@layout/fragment_devices" />

</android.support.v4.widget.DrawerLayout>
```

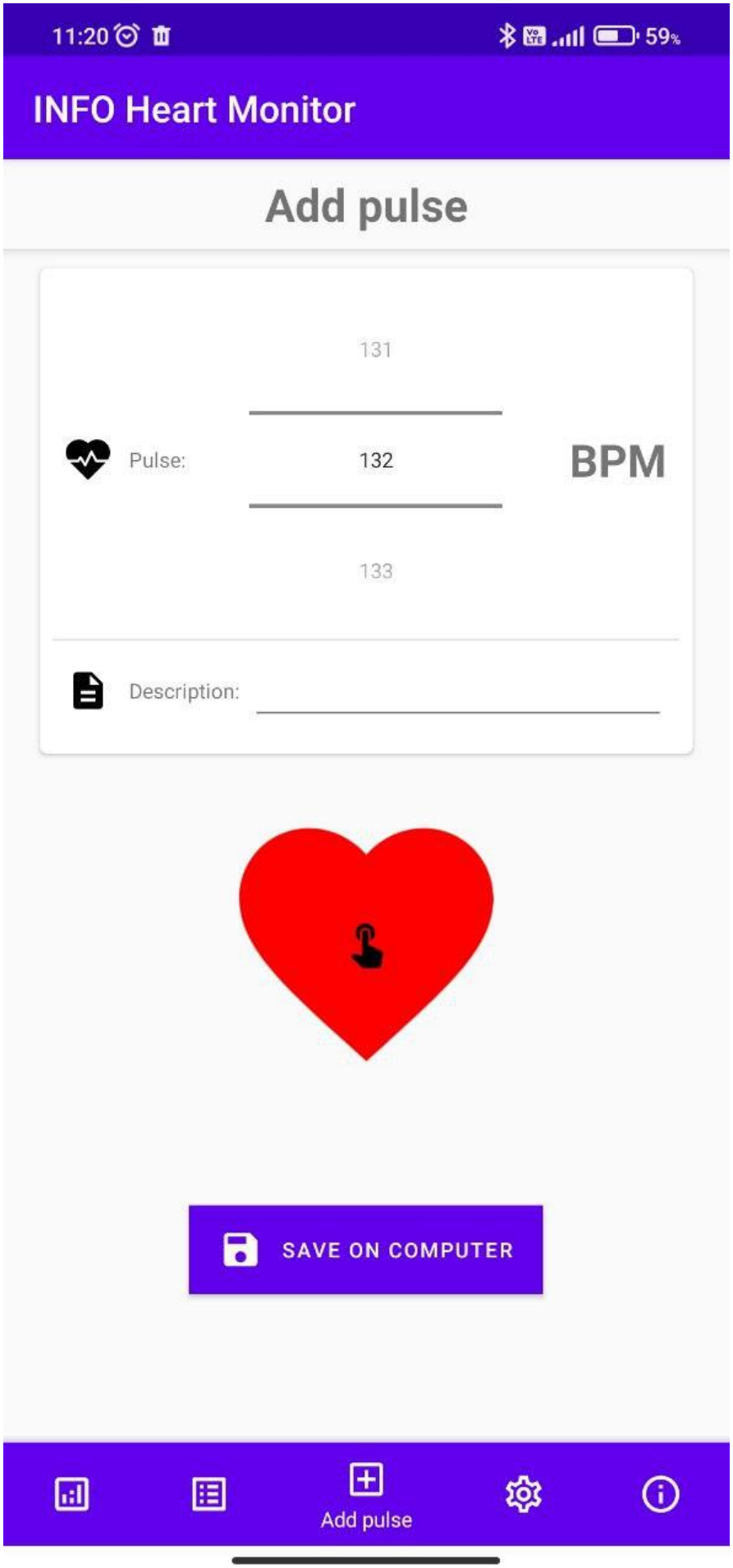
APP SCREENSHOT



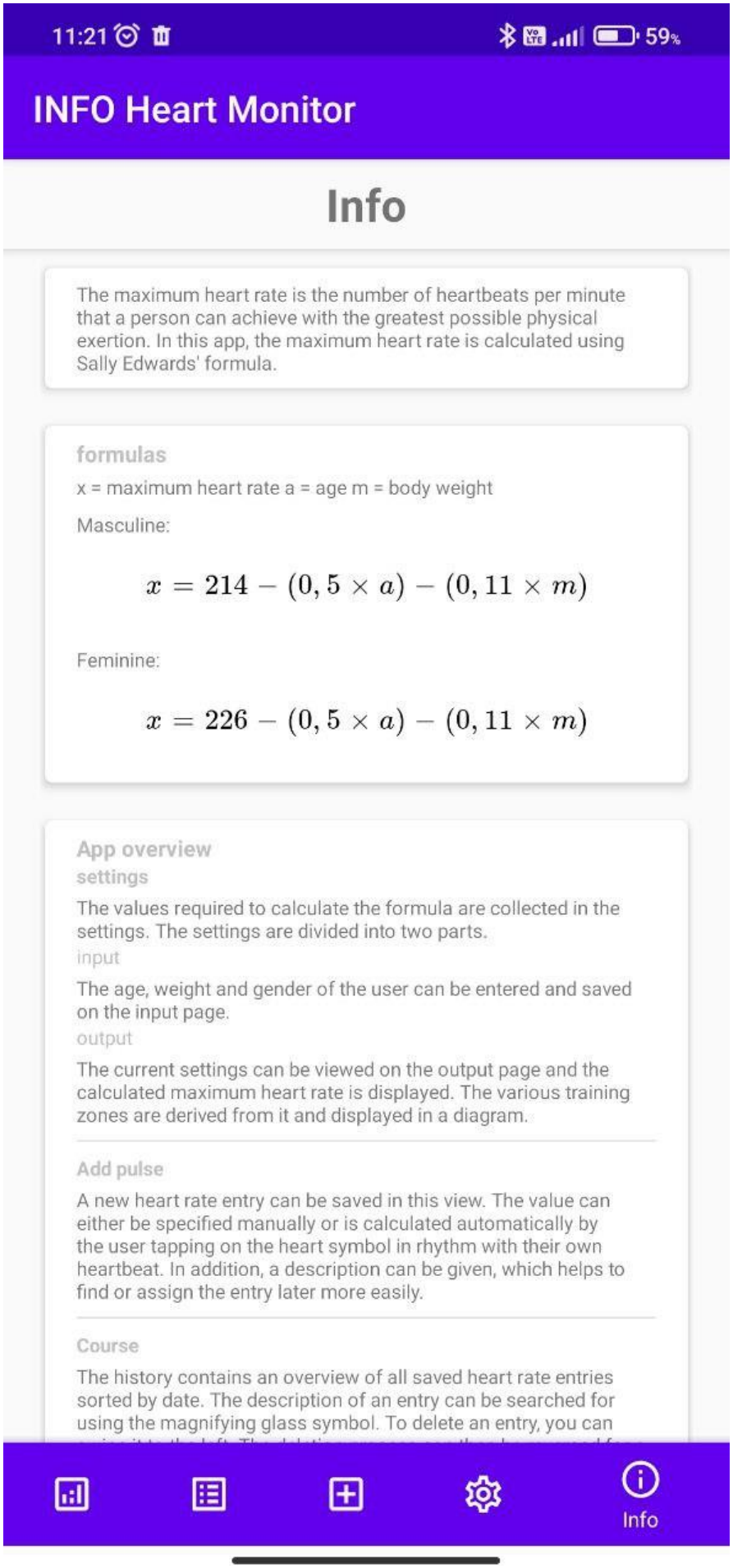

11:19
59%
Food Diary
Yesterday
Clear
1249
−
0
=
1249
Breakfast
no food here!
--
Lunch
no food here!
--
Supper
no food here!
--
Dinner
no food here!
--
Track


11:19
59%
Workout
MUSCLE & STRENGTH
ULTIMATE FAT-LOSS
MUSCLE HYPERTROPHY
BEGINNERS' 4 WEEK
90 DAYS TO VISIBLE ABS
ROUTINES
Classic Push Pull Leg
Build insane mass with this PPL routine
Bro Split
The most popular split routine
Full Body Workout
Maintain size and build muscles slowly
Hybrid Split
This routine is best when you are bulking up
Workout


11:19
59%
Profile
Set Custom Macros (in grams)
Journey
Current Weight
67.0
Update Weight
Goal Weight
0.0
Set Goal Weight
Profile
KG
64
AGE
67.0
WEIGHT
97.0
HEIGHT
Male
GENDER
Lifestyle
Sedentary
Goal
Maintenance
Weekly Goal
Maintain
Profile


11:20
59%
INFO Heart Monitor
Add pulse
131
Pulse:
132
BPM
133
Description:
SAVE ON COMPUTER
Add pulse


11:20
59%
INFO Heart Monitor
settings
INPUT
RESULT
29
Age:
30
31
59
Weight:
KG
60
61
Gender:
männlich
SAVE ON COMPUTER
settings


11:21
59%
INFO Heart Monitor
Info
The maximum heart rate is the number of heartbeats per minute that a person can achieve with the greatest possible physical exertion. In this app, the maximum heart rate is calculated using Sally Edwards' formula.
formulas
x = maximum heart rate a = age m = body weight
Masculine:
$$x = 214 - (0,5 \times a) - (0,11 \times m)$$
Feminine:
$$x = 226 - (0,5 \times a) - (0,11 \times m)$$
App overview
settings
The values required to calculate the formula are collected in the settings. The settings are divided into two parts.
input
The age, weight and gender of the user can be entered and saved on the input page.
output
The current settings can be viewed on the output page and the calculated maximum heart rate is displayed. The various training zones are derived from it and displayed in a diagram.
Add pulse
A new heart rate entry can be saved in this view. The value can either be specified manually or is calculated automatically by the user tapping on the heart symbol in rhythm with their own heartbeat. In addition, a description can be given, which helps to find or assign the entry later more easily.
Course
The history contains an overview of all saved heart rate entries sorted by date. The description of an entry can be searched for using the magnifying glass symbol. To delete an entry, you can
Info

11:21
59%
INFO Heart Monitor
Info
shared as an image in other apps.
Value range of the parameters
Age must be GREATER than 0 and in whole number years.
Body weight must be GREATER than 0 and in kilograms.
Male or female must be selected as gender.
interpretation
An interpretation of the result can be determined using the additionally displayed tables.
50-60% of maximum heart rate is the health zone.
60-70% of maximum heart rate is the fat burning zone.
70-80% of maximum heart rate is the aerobic zone.
80-90% of maximum heart rate is the anaerobic zone.
90-100% of maximum heart rate is the warning zone.
110.0 %
100.0 %
90.0 %
80.0 %
70.0 %
60.0 %
50.0 %
40.0 %
30.0 %
20.0 %
100
warning zone
90
anaerobic zone
80
aerobic zone
70
fat burning zone
60
health zone
50
no zone
Info

CHAPTER 9

REFERENCES

1. U. Satija, B. Ramkumar and M. Sabarimalai Manikandan, "Real-Time Signal Quality-Aware ECG Telemetry System for IoT-Based Health Care Monitoring", *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 815-823, June 2017.

2. A. Khalaf and R. Abdoola, "Wireless body sensor network and ECG Android application for eHealth", *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, pp. 1-4, 2017.

3. S. K. Jain and B. Bhaumik, "An Energy Efficient ECG Signal Processor Detecting Cardiovascular Diseases on Smartphone", *IEEE Transactions on Biomedical Circuits and Systems*, vol. 11, no. 2, pp. 314-323, April 2017.

4. M. A. Ahamed, M. K. Hasan and M. S. Alam, "Design and implementation of low cost ECG monitoring system for the patient using smartphone", *2015 International Conference on Electrical & Electronic Engineering (ICEEE)*, pp. 261-264, 2015.

5. L. Moin, V. Uddin, A. Y. Memon and S. H. Zaidi, "Bond graph model for the right atrium of heart", *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 181-185, 2016.

6. A. Guyton and J. Hall, "Electrocardiographic Interpretation of Cardiac Muscle and Coronary Blood Flow Abnormalities: Vectorial Analysis" in Guyton and Hall Textbook of Medical Physiology, Philadelphia, Pennsylvania, PA, USA:Saunders Elsevier, pp. 129-141, 2011.

7. A. Guyton and J. Hall, "The Normal Electrocardiogram" in Guyton and Hall Textbook ofMedical Physiology, Philadelphia, Pennsyl-vania, PA, USA:Saunders Elsevier, pp. 121-124, 2011.

8. D. Bertsimas, L. Mingardi and B. Stellato, "Machine Learning for Real-Time Heart Disease Prediction", *IEEE Journal of Biomedical and Health*

Informatics, vol. 25, no. 9, pp. 3627-3637, Sept. 2021.

9. A. Guyton and J. Hall, "Cardiac Arrhythmias and Their Electrocardiographic Interpretation" in Guyton and Hall Textbook of Medical Physiology, Philadelphia, Pennsylvania, PA, USA:Saunders Elsevier, pp. 143-153, 2011.

10. A. Nayyar and V. Puri, "A review of Arduino board's Lilypad's & Arduino shields", *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, vol. 2016, pp. 1485-1492.

11. P. Sujatha and K. Mahalakshmi, "Performance Evaluation of Supervised Machine Learning Algorithms in Prediction of Heart Disease", *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pp. 1-7, 2020.

12. A. Gavhane, G. Kokkula, I. Pandya and K. Devadkar, "Prediction of Heart Disease Using Machine Learning", *2018 Second International Conference on Electronics Communication and Aerospace Technology (ICECA)*, pp. 1275-1278, 2018.

13. L. Zhang, K. S. Kumar, H. He, C. Cai, X. He, H. Gao, et al., "Fully organic compliant dry electrodes self-adhesive to skin for long-term motion-robust epidermal biopotential monitoring", *Nat. Commun. 11*, pp. 1-13, 2020.