Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2018

Project Title:	**Real-Time Blood Pressure Data Communication and Feature Extraction on Android Devices via Bluetooth**
Student:	**Paulo Alessandro Iswanto Carrasquero**
CID:	**00926196**
Course:	**4T**
Project Supervisor:	**Dr Longfang Zou** and Co-supervisor: **Prof Christopher McLeod**
Second Marker:	**Dr Esther Rodriguez Villegas**

Abstract

Heart failure is a crippling chronic condition that takes its toll on patients as hospitalization is often inevitable. Traditional care for heart failure consists of patient self-management following an acute episode, but increasing advances in technology have allowed the development of telemedicine solutions to supplement this treatment. In fact, hemodynamic monitoring of pulmonary artery pressure has recently been shown to be the most effective method of preventing patient rehospitalization to date. This project is concerned with the development of a mobile application to interact with a wireless blood pressure sensor developed at Imperial College London. The app provides patients and clinicians with a real-time view of blood pressure measurements and the different features associated with the waveform such as systolic, diastolic and mean pressures. Different algorithms for feature extraction were compared on a dataset of pulmonary blood pressure measurements from various hospitalized patients. The resulting app combines feature extraction and abnormality detection on waveforms to alert users of any irregularities and has been shown to be a promising tool for self-management of heart failure patients.

Contents

Abstract **i**

1 Introduction **1**

1.1 Motivation 1

1.2 Project Objectives 2

1.3 Report Overview 3

2 Background **4**

2.1 Introduction 4

2.2 The Cardiovascular System 4

2.2.1 Cardiac Cycle 5

2.2.2 Heart Failure 7

2.3 Self-Management in Patients with Heart Failure 9

2.4 The Rise of Telemedicine for Heart Failure Care 10

2.4.1 Noninvasive systems 11

2.4.2 Hemodynamic Monitoring Sensors 13

3 Requirements Capture **15**

3.1 Android Application 16

3.2 Comprehensive User Interface 17

3.3 Real-Time Visualization . . . 17

3.4 Two-Way Communication . . . 17

3.5 Feature Extraction . . . 18

3.6 Abnormality Detection . . . 18

3.7 Data Logging . . . 18

4 System Design **20**

4.1 Architecture Overview . . . 20

4.2 Android Design . . . 21

4.3 Hardware . . . 24

5 Charting Library Selection **25**

5.1 Library Search . . . 25

5.2 Library Performance . . . 26

6 Signal Processing **31**

6.1 Introduction . . . 31

6.2 Physiological Signal Dataset Creation . . . 32

6.3 Feature Extraction . . . 35

6.3.1 Beat Onset Detection . . . 36

6.3.2 Systolic Pressure . . . 38

6.3.3 Diastolic Pressure . . . 43

6.3.4 Mean Pressure . . . 44

6.3.5 Dicrotic Notch 45
6.3.6 Dicrotic Peak 47
6.3.7 Max Rate of Change 48
6.3.8 Heart Rate 49
6.4 Abnormality Detection 50
6.4.1 Abnormality Definition 51
6.4.2 Signal Quality Index 52
6.4.3 Results 53
6.5 Data Issues 54
6.5.1 Estimate Accuracy 55
6.5.2 Data Quality 55

7 System Implementation 58

7.1 Development Practices 58
7.2 App Architecture Overview 59
7.3 Two-Way Data Communication 60
7.3.1 Reader Simulation 60
7.3.2 Device Bluetooth Connection 61
7.4 Real-Time Visualization 64
7.5 Signal Processing 64
7.6 Database 66
7.7 Security and Privacy 67

8 Interface Design and Implementation **68**

8.1 Initial Mock 68

8.2 Graph Page 69

8.3 Menu 72

8.4 About Page 73

9 Testing **75**

9.1 Motivation and Objectives 75

9.2 Testing Overview 76

9.2.1 Unit tests 76

9.2.2 Integration tests 76

9.2.3 UI tests 76

9.3 Unit Tests 77

9.3.1 GUI 77

9.3.2 Logging 77

9.4 Integration Tests 77

9.4.1 Data Communication 77

9.5 UI Tests 78

9.5.1 Usability 78

9.5.2 Performance 78

9.5.3 Signal Processing 78

10 Results 79

10.1 Usability . . . 79

10.2 Performance . . . 80

10.3 Signal Processing . . . 81

11 Evaluation 82

12 Conclusions and Further Work 84

12.1 Conclusions . . . 84

12.2 Future Work . . . 84

13 User Guide 85

13.1 App Development . . . 85

13.2 Reader . . . 85

13.3 Signal Processing . . . 85

Bibliography 86

Appendix A Additional documents 95

Appendix B MARS Questionnaire 96

List of Tables

2.1 Normal blood pressure ranges. 6

5.1 Android Charting Libraries Comparison. 25

5.2 Charting library performance statistics for 10ms update rate. 28

5.3 Charting library performance statistics for 20ms update rate. 29

6.1 Algorithm performance across all patients for systolic pressure estimation. . . . 41

6.2 Algorithm performance across all patients for diastolic pressure estimation. . . . 44

6.3 Algorithm performance across all patients for mean pressure estimation. 45

6.4 Algorithm performance across all patients for dicrotic notch pressure estimation. 46

6.5 Algorithm performance across all patients for dicrotic peak pressure estimation. 48

6.6 Algorithm performance across all patients for max rate of pressure change estimation. 49

6.7 Algorithm performance across all patients for heart rate estimation. 50

6.8 Range of values between 10th and 90th percentiles for features in the PAP Dataset. 51

6.9 Abnormality criteria. 52

6.10 Abnormality detection in patients with primary pulmonary hypertension. 54

7.1 Reader parameters. 63

10.1 Rating Results. 79

10.2 Performance Statistics. 80

List of Figures

2.1 Heart diagram during systole and diastole phases. 5

2.2 Wigger's diagram. 6

2.3 Pulmonary artery pressure trace. 7

2.4 The benefits of telemedicine. 11

2.5 The CardioMEMS system. 13

3.1 Hemodynamic Monitoring System Diagram. 15

3.2 Project System Diagram. 16

4.1 System Architecture Diagram. 20

4.2 Android Activity Lifecycle. 22

4.3 Multiple Activity App. 23

4.4 Single Activity App. 23

5.1 GraphView performance test. 27

5.2 GameBench app (desktop above and cloud bottom). 27

5.3 Charting library performance for 10ms update rate. 28

5.4 Charting library performance for 20ms update rate. 28

6.1 Diagram of Matlab-PhysioNet Interaction. 33

6.2 Dataset Creation Flow Chart. 33

6.3 Venn Diagram of PAP data from MIMIC-III. 34

6.4 Example PAP Waveforms. 37

6.5 WABP Algorithm Flow Diagram. 37

6.6 Feature Extraction Overview. 38

6.7 Maximum Algorithm. 39

6.8 Inflexion and Zero-Point Crossing Algorithm. 39

6.9 Find Peaks Algorithm. 40

6.10 AMPD Algorithm. 40

6.11 Time complexity of max algorithm. 41

6.12 Time complexity of inflexion algorithm. 42

6.13 Time complexity of find peaks algorithm. 42

6.14 Time complexity of AMPD algorithm. 42

6.15 Minimum Algorithm. 43

6.16 End of Beat Algorithm. 43

6.17 One-Third, Two-Thirds Algorithm. 44

6.18 Integral Algorithm. 44

6.19 Straight-Line Maximum Algorithm. 45

6.20 Inflexion Algorithm. 46

6.21 Straight-Line Minimum Algorithm. 47

6.22 Maximum Algorithm. 47

6.23 jSQI Algorithm. 53

6.24 Matlab-MIMIC-III Interaction. 54

6.25 Scatter Plot of Average RMSE for Systolic Pressure for a Given Record Versus Number of Beats. 55

6.26 Machine noise. 56

6.27 Missing data. 56

6.28 Irregular Waveform. 56

7.1 Agile Methodology Diagram. 58

7.2 Android App Class Diagram. 59

7.3 Bluetooth Connection Diagram. 62

7.4 Code Porting from Matlab to Java. 66

7.5 Database schema. 66

8.1 Minimum viable product mock. 68

8.2 Graph Page. 69

8.3 Graph Plot and Menu. 70

8.4 Make Discoverable Dialog. 70

8.5 Bluetooth Connection Window. 71

8.6 Error Message. 71

8.7 Export Progress Bar. 72

8.8 Navigation Drawer and Tabbed Menu Hybrid App. 72

8.9 Navigation Drawer. 73

8.10 About Page. 74

8.11 Navigation Drawer Update. 74

9.1 The testing pyramid. 75

10.1 App performance during a typical session. 80

Chapter 1

Introduction

1.1 Motivation

Heart failure is a common occurrence that is costly both to the patient and health care system. The total medical costs of heart failure are posed to increase from $21 billion in 2012 to $70 billion in 2030 in the United States alone [1]. The condition is associated with high rehospitalization rates as up to 20% of patients are readmitted within a month of their discharge and 50% of those affected are readmitted within 6 months [2]. This places an additional burden on hospitals since health care plans implement a penalty system for readmissions according to the Hospital Readmissions Reduction Program [3]. An approximate 40% of discharged patients end up in skilled nursing facilities that provide specialized care towards rehabilitation. However, a recent study by M Manemann et al. [4] showed that patients in these facilities had a higher chance of being rehospitalized, bringing into question the effectiveness of current heart failure rehospitalization prevention measures.

Current research in the area of heart failure focuses on rehospitalization prevention, mainly due to the lack of adequate monitoring measures after hospitalization [5]. This is partly due to insufficient patient data as the best way to obtain blood pressure measurements is through invasive methods like a catheter which obtains readings solely when the patient is in the hospital. Readings are few and far between and do not take into account a patient's vital signs throughout daily life. Noninvasive telehealth systems exist, but randomized controlled trials have not correlated these systems to lower rehospitalization rates [6]. This can be attributed to the poor quality of the measured signals due to the nature of noninvasive systems. The paper by Nguyen and Squara [7] showed that the diagnostic performance of noninvasive systems is unreliable due to the heterogeneity of measurements stemming from the varying precisions and accuracies of such devices. Hence, invasive systems are needed to accurately measure patients' vital signs after discharge to prevent further rehospitalization due to heart failure.

1.2 Project Objectives

As part of the research project "Automated Blood Pressure Monitoring" funded by the Wellcome Trust and the Department of Health, Murphy et al. [8] lead by Prof. McLeod from the Institute of Biomedical Engineering at Imperial College London developed a wireless sensor based on surface acoustic wave (SAW) technology to measure pulmonary arterial pressure (PAP).

The sensor can be interrogated wirelessly by a wearable device which transmits the pressure measurements via a reader. Recent developments in heart failure treatments focus on blood pressure readings to take a proactive approach in post-hospitalization patient management. A patient implanted with the sensor could obtain continuous pulmonary artery blood pressure measurements which could help doctors provide suitable therapy based on pressure data to avoid further hospitalization in case of worsening symptoms.

The aim of this project consists of developing an Android application for mobile devices to communicate via Bluetooth with the reader to set its internal parameters and display blood pressure measurements. Furthermore, the app will extract meaningful features from the measurements such as systolic, diastolic and mean pressures. It should also perform abnormality detection to check if there are irregularities in the waveforms. However, the scope of this project is limited to flagging abnormalities to users for further medical care as opposed to diagnosing and prescribing which should be done by an experienced medical professional.

1.3 Report Overview

This report outlines the importance of heart failure management and advances in condition treatment in chapter 2. It discusses the major types of telemedicine solutions available and the importance of hemodynamic monitoring for rehospitalization prevention. Chapter 3 provides an overview of user requirements for the Android application to be developed and chapter 4 the analysis and design of the overall system. Chapter 5 discusses the solutions available for real-time graphing and chapter 6 the algorithms involved in feature extraction and abnormality detection. The implementation of said system is discussed in chapter 7 and user interface considerations detailed in chapter 8. Chapter 9 explains the tests in place to ensure that the app functions correctly and chapter 10 examines the performance of the overall system. Chapters 11 and 12 discuss the fulfillment of project objectives compared to future avenues of research. Finally, chapter 13 shows the steps involved in continuing the project.

Chapter 2

Background

2.1 Introduction

Recent years have seen the rise of the biomedical industry at the intersection of medicine and technology. With innovation at its core, the sector has adopted modern solutions to solve traditional problems with the aim of improving public health and quality of life. This chapter focuses on the evolution of treatment for heart failure and the incorporation of telehealth as a supplement to patient self-management of the condition.

2.2 The Cardiovascular System

The cardiovascular system is responsible for pumping blood through the human body to transport nutrients, food, cells and waste. It consists of:

- The heart, a muscular organ that pumps blood in repeated, rhythmic contractions.
- A network of blood vessels comprised of arteries that transport blood away from the heart, veins that transport blood to the heart and capillaries that transport blood to tissue cells to exchange substances such as nutrients or waste.

It is part of the larger circulatory system that includes the lymphatic system and circulates blood and lymph through the body.

2.2.1 Cardiac Cycle

The heart pumps blood in a rhythmic fashion made up of two phases: the systole where the heart contracts and pumps blood into circulation, and the diastole where it relaxes and allows blood to fill its chambers. The cardiac cycle is defined as one repetition of the aforementioned sequences. Its duration is on average 0.8 seconds for a healthy young adult and the heart rate can be defined as the inverse of this duration. Thus, the average heart rate for a healthy young adult is around 75 beats per minute (bpm) [9].

The heart is made up of four chambers: upper right and left atria, and lower right and left ventricles. The left and right pairs of atrium and ventricle each form a separate pumping system. The right atrium receives oxygen-poor blood from the body that passes through the right ventricle and is then pumped to the lungs through the pulmonary artery. The oxygen-rich blood reaches the left atrium, passes the left ventricle and is then pumped through the aorta to the rest of he body. The process is illustrated in figure 2.1:

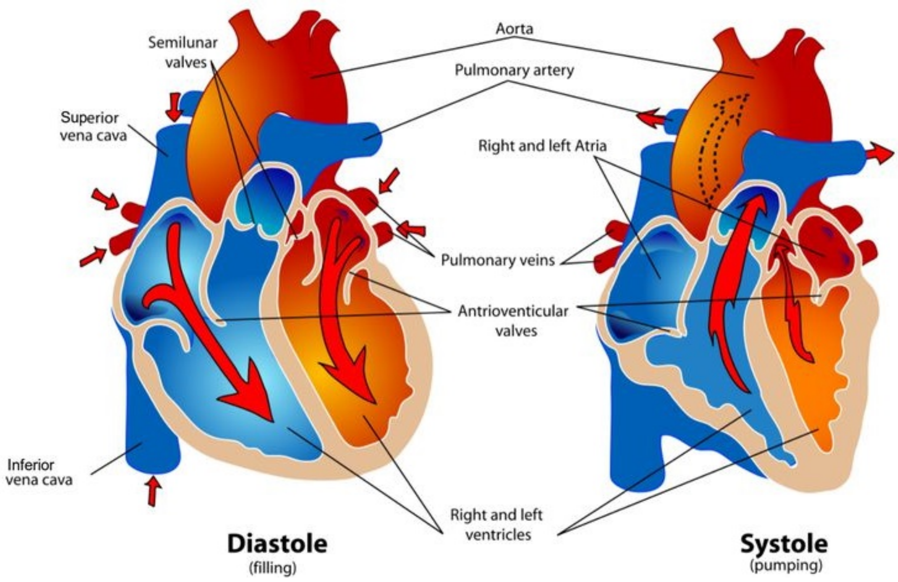

Figure 2.1: Heart diagram during systole and diastole phases [10].

The one-way flow of blood through the heart is ensured by atrioventicular valves (tricuspid and mitral valves) between each atrium and ventricle as well as semilunar valves (aortic and pulmonary valves) controlling blood flow coming out of each ventricle. Furthermore, the ventricles have thicker walls than the atria as the pressure needed to pump blood through the body or lungs is greater than the pressure needed to pass blood from atrium to ventricle. The left ventricle has a thicker wall than the right ventricle as more pressure is needed to carry blood through the body compared to the lungs. Thus, the arterial blood pressure (ABP) will be higher than the pulmonary artery pressure (PAP) as seen in table 2.1 below:

Blood Pressure	Systolic (mmHg)	Diastolic (mmHg)
Arterial	90-140	60-90
Pulmonary	15-30	4-12

Table 2.1: Normal blood pressure ranges [11].

Ranges are quoted for systolic and diastolic pressures defined as the highest pressure when the heart contracts and lowest pressure when the heart is relaxed respectively. The cardiac cycle is best explained by relating its phases to blood pressure over time as shown in figure 2.2 below:

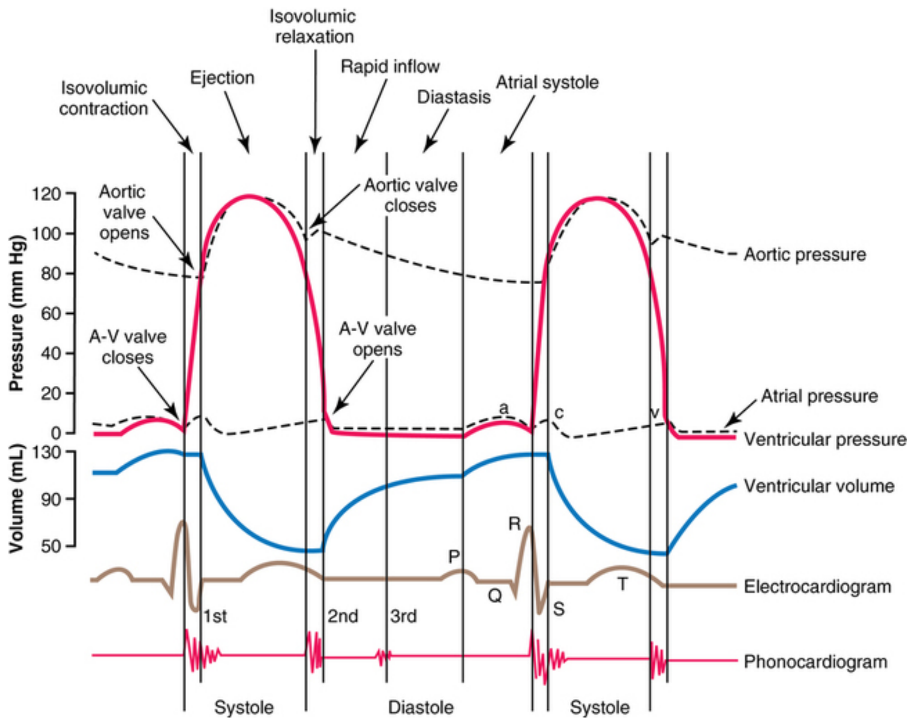

Figure 2.2: Wigger's diagram [12].

- **Isovolumic contraction**: The atrioventricular valves close as blood has flowed from the atria to the ventricles. The ventricles start to contract but the semilunar valves are closed, thus there is no change in ventricular volume and ventricular pressure increases rapidly.

- **Ejection**: The semilunar valves are open and the ventricles contract.

- **Isovolumic relaxation time**: The semilunar valves close and the ventricles begin to relax. Ventricular volume remains the same and ventricular pressure decreases.
- **Rapid inflow and diastasis (diastole)**: The atrioventricular valves open and the ventricles fill with blood from the pulmonary system or other parts of the body.
- **Atrial systole**: The atria contract allowing blood to flow from the atria to the ventricles.

The cardiac cycle can be used to explain the pulmonary artery pressure (PAP) waveform as well:

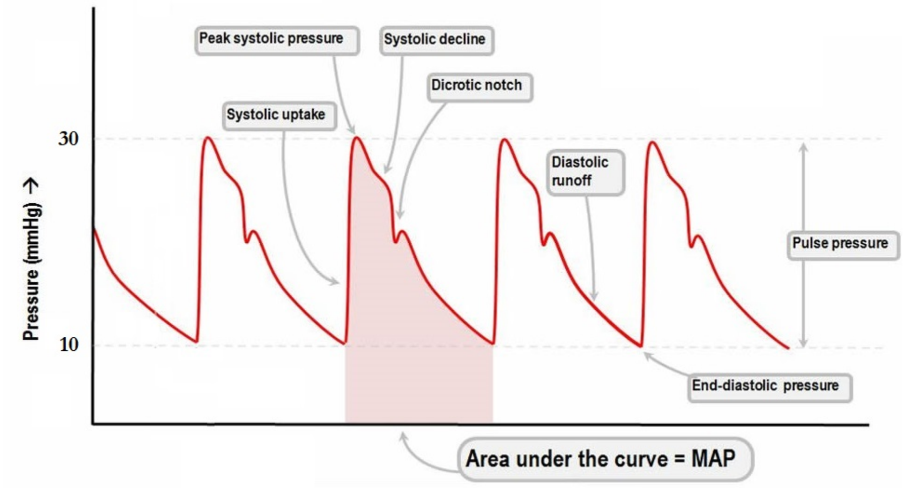

Figure 2.3: Pulmonary artery pressure trace [12].

The PAP trace is similar to the ABP except for lower pressure values due to ventricle thickness. The dicrotic notch present after peak systole represents a small decrease in pressure due to the closure of the pulmonary valve and the short reflection of the wave.

2.2.2 Heart Failure

Heart failure occurs when the heart does not pump blood around the body properly. This can happen as the heart's chamber walls thicken, meaning that the heart will not fill with enough blood due to stiffness or because the heart muscle weakens and the ventricles cannot pump blood through the body at the required pressure due to the dilation of the ventricles causing its walls to become thinner. It is a long-term condition that gets progressively worse with time although it can be controlled with proper treatment.

Heart failure is a complex condition that has many underlying risk factors and is often brought on by a compound of causes including: heart attack, high blood pressure, congenital heart conditions or heart valve disease. Its prevalence increases with age so it is all the more important to treat in an aging global population.

The prognosis for heart failure is poor since more than 40% of patients diagnosed with heart failure die within one year of hospitalization [13]. Thus, there have been recent efforts to raise awareness of the causes of heart failure to lead the public to adopt a healthier lifestyle and reduce the risk of developing the condition. The Study of Heart Failure Awareness and Perception in Europe (SHAPE) was a public education programme study that surveyed 47,985 households in different European countries. Participants were given a questionnaire and asked to identify the condition associated with the written signs and symptoms. The results showed that "only 3% identified HF from the description of breathlessness, tiredness, or swollen ankle" [14]. The study uncovered challenges related to the early diagnosis of heart failure since the general population had trouble recognizing the signs and symptoms related to the condition. This was further acknowledged by Hobbs et al. [15] who concluded that "individual symptoms [...] and signs [...] are generally weak predictors of heart failure, with poor reliability, and little agreement between clinicians on their presence or absence, even amongst specialists".

The bulk of early diagnosis of heart failure relies on the recognition of signs and symptoms by patients and clinicians. However, timely diagnosis is hampered by the variability of symptoms amongst patients and the inability of clinicians to accurately identify the condition. As a consequence, focus in the field has shifted from the prevention of heart failure to providing adequate care for patients post-hospitalization to prevent recurrent episodes [16].

2.3 Self-Management in Patients with Heart Failure

In the context of health care, self-management refers to a patient's active management of their health on a day-to-day basis. This is particularly important for heart failure patients who suffer from the long-term condition.

The early 2000s saw the introduction of chronic disease self-management (CDSM) programs that aimed to provide a comprehensive framework to improve chronic illness treatment [17]. In the context of heart failure, the objective of CDSM is to prevent symptoms, reduce health care costs and improve patient outcomes.

The American Heart Association (AHA) published widely accepted guidelines for self-care behavior in patients with heart failure including medication adherence, symptom monitoring, dietary adherence, fluid restriction and weight loss. In the paper, Riegel et al. [18] found that self care activities "significantly reduced HF hospitalizations [...] and all-cause hospitalization".

Self-management has been identified by researchers as one of the pillars in the treatment of chronic illnesses. However, unfavorable patient outcomes have been associated with poor self-management [19]. Trupp et al. [20] attributed poor post-hospitalization results to the fact that heart failure management is complex due to "the requisite polypharmacy resulting from evidence based care, dietary limitations, lifestyle modifications, and the need for frequent contact and follow-up". This was further exposed by Michalsen et al. [21] who showed that over 60% of rehospitalizations amongst heart failure patients were directly due to lack of adherence to the prescribed medical and dietary regimens. After suffering from an acute episode of heart failure, it is understandable that a patient may be overwhelmed by the adoption of sudden lifestyle changes ending up in poor self-management of the condition.

The varying degrees of self-management evidenced by patients led Ecock Connelly [22] to develop the Model of Self-Care in Chronic Illness (MSCCI) in which age, gender, socioeconomic status, symptom severity, education, comorbidity and social support form the characteristics that influence self-management behavior. Subsequent research found that education and symptom severity were the most important factors in the MSCCI model [23]. Thus, patients who are well educated and likely to report symptoms to clinicians are more likely to succeed in self-managing heart failure.

This led to the development of the patient activation concept described by Hibbard et al. [24] as "the skills, knowledge, and motivation [of patients] to participate as effective members of the care team". This shifts the focus of the patient from a passive recipient to an active member in the management of the condition as measured by the Patient Activation Measure (PAM). A latter study by Mosen et al. [25] involving 4,108 adults suffering from chronic conditions confirmed the usefulness of patient activation theory as "[patients] with higher PAM scores were more likely to perform self-management behaviors, use self-management services, and report higher medication adherence".

Shifts in the landscape of health care policy have put the patient forward as an active agent of the health care system. This places the patient at a powerful position to impact health care quality and reduce associated costs. However, active engagement is dependent on patient activation which illustrates the opportunity for patients to delegate some of the responsibilities of self-management to technology.

2.4 The Rise of Telemedicine for Heart Failure Care

The term telemedicine refers to the remote delivery of health-related services allowing long distance patient outreach and monitoring by clinicians. It is not to be confused with the mobile health trend (mHealth), a consumer-driven approach to deliver health services via smart phones with little regulation and quality control regarding application development. Telemedicine encompasses all means of telecommunications and is clinician-driven, meaning that most solutions are backed by medical experts who analyze data collected and are more suited to give medical advice.

The field of telemedicine has been researched since the last century. However, recent technological developments have allowed the proliferation of telemedicine solutions as evidenced

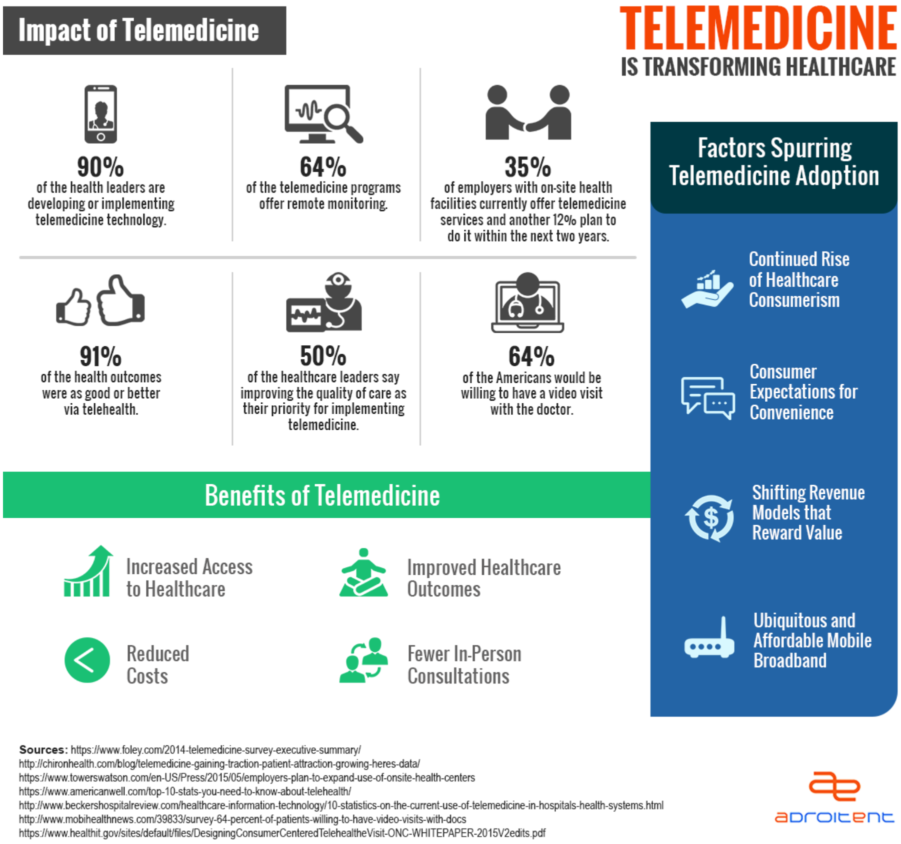

Figure 2.4: The benefits of telemedicine [26].

by the tenfold increase in articles citing the word "telemedicine" in a span of 20 years from the early 1990s to 2010 [27]. Empirical evidence has backed telemedicine solutions which have been shown to reduce mortality for patients suffering from a chronic disease by 15% to 56% depending on the illness [28].

The history of heart failure treatment using telemedicine has seen several developments in recent years. Below is a summary of the different solutions:

2.4.1 Noninvasive systems

A noninvasive system is characterized by the lack of introduction of foreign objects in the human body and is purely external. This type of system involves regular interaction between patient and clinician for collection of measurements, symptoms and signs.

Noninvasive systems do not necessarily involve the use of devices to measure a patient's physiological measurements. One of the first large scale studies investigating the usefulness of telemedicine for heart failure patients involved only telephone calls. The Telemonitoring to Improve Heart Failure Outcomes (TELE-HF) trial split 1,653 recently hospitalized patients suffering from heart failure into a control group undergoing usual care and a group receiving telemonitoring support [29]. Telemonitoring was implemented as ”telephone-based interactive voice-response system that collected daily information about symptoms and weight that was reviewed by the patients' clinicians”. Chaudhry et al. [29] concluded that telemonitoring did not improve rehospitalization or death rates within 6 months of enrollment in the trial.

Following the TELE-HF trial, another more thorough randomized clinical trial investigated the benefits of noninvasive systems collecting both qualitative and quantitative data from heart failure patients. The Better Effectiveness After Transition–Heart Failure (BEAT-HF) trial followed 1,437 recently hospitalized heart failure patients for 180 days [30]. Patients received coaching telephone calls and clinicians collected daily blood pressure, weight, symptom and heart rate measurements with electronic equipment. The centralized data was reviewed by specialized nurses who followed protocolized actions according to daily data reviews. At the end of the trial, telemonitoring did not reduce rehospitalization rates. However, quality of life was significantly higher in the group of remotely monitored patients.

This suggests that noninvasive telemedicine provides palliative care that does not reduce the risk of acute heart failure episodes in patients. This can be attributed to the fact that the signs and symptoms monitored occur late in the wake of an acute heart failure episode [15] meaning that once detected, hospitalization is hard to prevent.

Zile et al. [31] later showed that ventricular filling pressures increase in the weeks preceding hospitalization, a phenomenon called hemodynamic congestion. This finding along with previous evidence from poor performance of noninvasive systems led to the exploration of invasive methods to measure blood pressure with the rationale that a proactive approach to hemodynamic congestion would reduce symptoms and eventually, prevent hospitalization for the patient.

2.4.2 Hemodynamic Monitoring Sensors

Hemodynamic sensors measure blood pressure in different parts of the heart. A recent discovery in the field involved the development of implantable sensors for the right ventricle measuring pulmonary artery pressure. The Chronicle Offers Management to Patients with Advanced Signs and Symptoms of Heart Failure (COMPASS-HF) trial involved 274 New York Heart Association (NYHA) functional class III or IV heart failure patients implanted with a right ventricle hemodynamic sensor or receiving usual care [32]. All patients received optimal therapy albeit those with monitoring sensors received treatment including information from the measurements of the sensors. Bourge et al. [32] found that hemodynamic monitoring reduced the risk of first time hospitalization by 36%. The study further highlighted pulmonary artery pressure (PAP) as an important biomarker for heart failure. However, the study did not specify clear guidelines for PAP ranges which meant that several patients maintained high PAP values for extended periods of time.

The most remarkable breakthrough in hemodynamic monitoring remains the CardioMEMS HF System that monitors PAP wirelessly and without need of a battery.

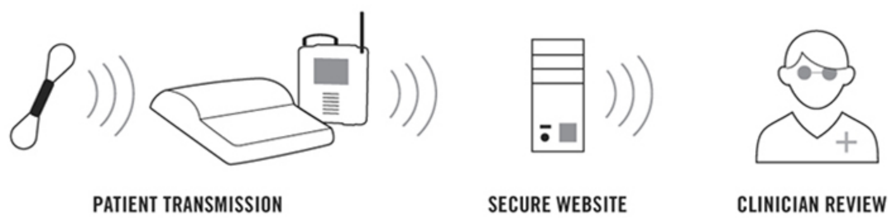

Figure 2.5: The CardioMEMS system [26].

The landmark CardioMEMS Heart Sensor Allows Monitoring of Pressure to Improve Outcomes in NYHA Class III Heart Failure Patients (CHAMPIONS) trial monitored 550 patients diagnosed with NYHA Class III heart failure. The study was more robust than COMPASS-HF as it involved clear guidelines for normal PAP ranges that fed into a treatment algorithm. The CHAMPIONS trial ended with a 37% reduction in rehospitalizations compared to the control group over a 15 moth period [33]. Furthermore, the device did not experience any failures and had a low complication rate compared to other hemodynamic sensors. This led to the U.S. Food and Drug Administration (FDA) approving the device for use in patients with NYHA Class III heart failure and a history of heart failure hospitalization within the past year.

The success of the CardioMEMS HF System spurred the development of various hemodynamic monitoring systems measuring PAP including the first implantable device using surface acoustic wave (SAW) technology developed at Imperial College London by a team of researchers from the Institute of Biomedical Engineering [8].

Chapter 3

Requirements Capture

The SAW sensor is designed to be implanted in the pulmonary artery [8] and intended to measure PAP without need of a battery. It transmits data wirelessly through excitation by an external reader to be worn by patients. The data is stored internally in the reader and sent to clinicians for review. The reader can also send the data to a mobile device via Bluetooth for visualization, logging and signal processing including feature extraction and abnormality detection. Furthermore, the reader has settable parameters that can be set through the mobile device.

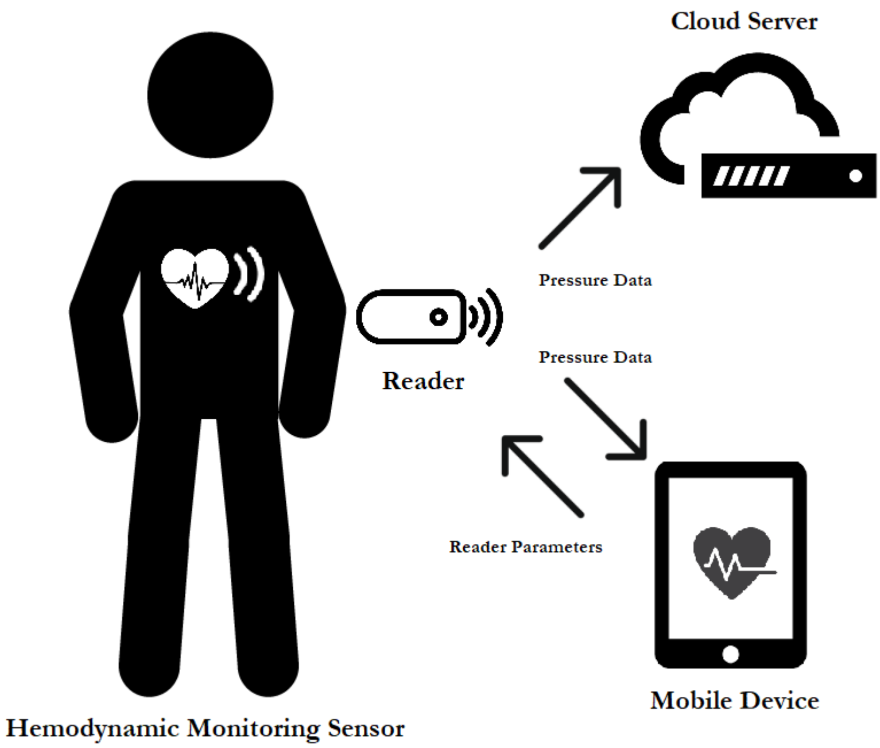

Figure 3.1: Hemodynamic Monitoring System Diagram.

The scope of this project includes the transmission of data between the reader and the mobile device through an Android app. The implanted sensor will be simulated by a laptop with PAP data and the reader by a Bluetooth development kit to transmit the data to the mobile device.

Henceforth, any mention of the reader implicitly refers to the Bluetooth development kit.

The reader is being developed simultaneously to this project so for all intents and purposes, the project is independent of the reader which will have to be simulated. However, if the reader is ready by the end of the project and time permits, it will be integrated to the system by replacing the Bluetooth development kit in the diagram below:

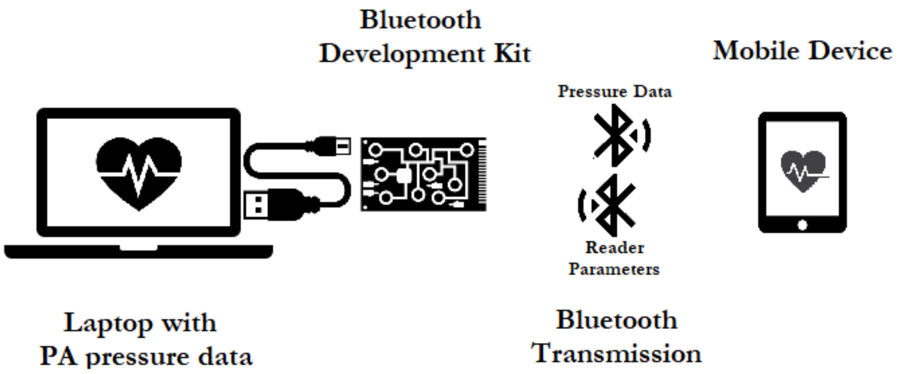

Figure 3.2: Project System Diagram.

The Android application is intended to be eventually used by patients and clinicians to visualize real-time PAP measurements and related waveform features. Below are the requirements for the mobile application:

3.1 Android Application

Android is the most popular mobile device operating system (OS) in the world, with more than 80% market share [34]. Thus, the mobile application will be developed only for Android as it has the most potential to reach a wide population.

For data transmission, Bluetooth is a mature wireless technology standard and the defacto method to transmit data over short distances. It is selected as the communication standard between the reader and mobile device for its maturity and proven reliability. More specifically, the application will require Bluetooth 4.0 that combines the high data throughput of Bluetooth 3.0 with improved stability and range [35].

- The Android application should be able to connect to devices through Bluetooth. It should be compatible with mobile devices that have Android 4.3 and above which has built-in support for Bluetooth 4.0 [36].

3.2 Comprehensive User Interface

Users should be able to intuitively browse the application and the user interface (UI) should remain consistent throughout the experience.

- The application should have an intuitive user interface and enforce common UI patterns for a good user experience (UX).

3.3 Real-Time Visualization

The app should be able to maintain a high enough frame rate for the duration of the app session to provide a smooth experience.

- The application should be able to visualize PAP measurements on a graph.
- The application should maintain a minimum median frame rate of 20 frames per second (fps) for a smooth experience [37].

3.4 Two-Way Communication

The application should be able to receive pressure data from the reader and send data back to the reader to set its internal parameters.

Furthermore, it is assumed that the reader samples the sensor's measurements at a base sampling rate of 50Hz which is the same its data transmission rate. This means that a new data sample will be sent to the app every 20ms. It is assumed that this is the maximum reader sampling rate and thus, its maximum data transmission rate.

- The application should be able to receive pressure data from the reader at a maximum data transmission rate of 50Hz or a new sample every 20ms.
- The application should be able to send data to the reader in order to set its internal parameters.

3.5 Feature Extraction

The application should display key PAP signal features to extract meaningful information from the pressure data.

- The application should display systolic, diastolic, mean, dicrotic notch, dicrotic peak pressures and maximum rate of pressure change (dp/dt) of PAP as well as heart rate.

3.6 Abnormality Detection

The application derives its usefulness from visualizing PA pressure measurements in real-time. However, flagging abnormalities gives users an added feature as they can report such incidents immediately to clinicians.

- The application should detect and flag abnormalities in PA pressure measurements.

3.7 Data Logging

All the pressure data from the sensor will be stored in the reader and periodically sent to clinicians. However, logging is necessary in the mobile device to store data obtained when the device is connected to the reader in case of a problem with the reader's communication chip.

- The application should log PAP measurements coming from the reader and enrich the raw data by logging key signal features and flagged abnormalities.

Chapter 4

System Design

4.1 Architecture Overview

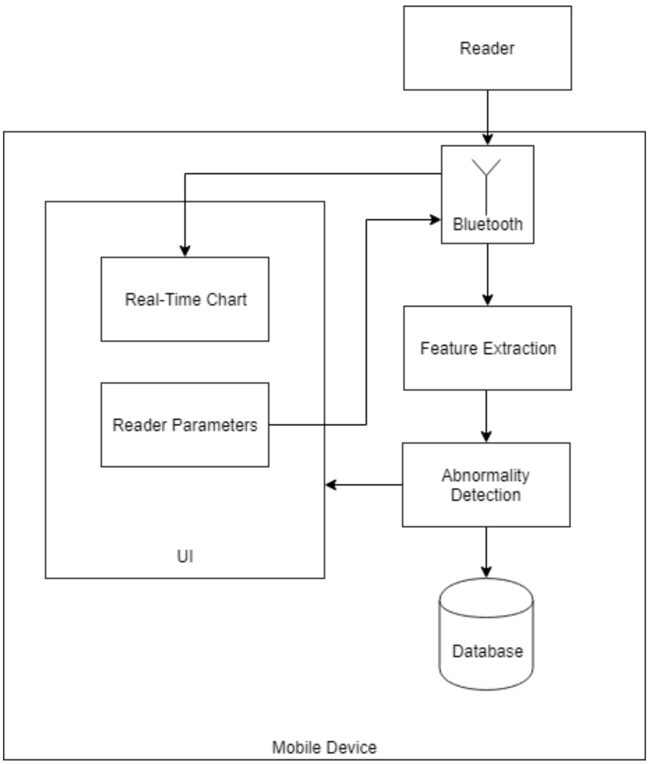

Figure 4.1: System Architecture Diagram.

The system is comprised of two parts: the reader simulation and the Android app. The Android app can be decomposed into its components:

- Bluetooth communication
- Visualization
- Signal Processing
  - Feature Extraction

  - Abnormality Detection
- Database

Figure 4.1 shows the high-level logic of the app. The reader sends data to the app which is displayed in the UI. At the same time, it is processed to extract features and detect abnormalities. In the end, all measurements are logged to a database and features/abnormalities displayed on the UI. Alternatively, users can input reader parameters in the UI which are then sent to the reader.

This architecture has the advantage of being simple as there is no complex feedback loop or dependency: all data flow is one-way. Furthermore, components are separated by their location in the app:

- **UI**:
  - Real-Time Chart
  - Reader Parameters
- **Background**:
  - Bluetooth
  - Feature Extraction
  - Abnormality Detection
  - Database

This clear separation enables a modular architecture where each component can be replaced without affecting any other components.

However, this does not take into account the features and limitations of the Android operating system which are discussed in the following section.

4.2 Android Design

The fundamental app component in any Android app is an activity. An activity is a single focused thing that a user can perform. Thus, the app could have an activity to display a real-time graph and another to input and send reader parameters. As the user navigates the app, the activity will transition between different states. Figure 4.2 shows the activity lifecyle.

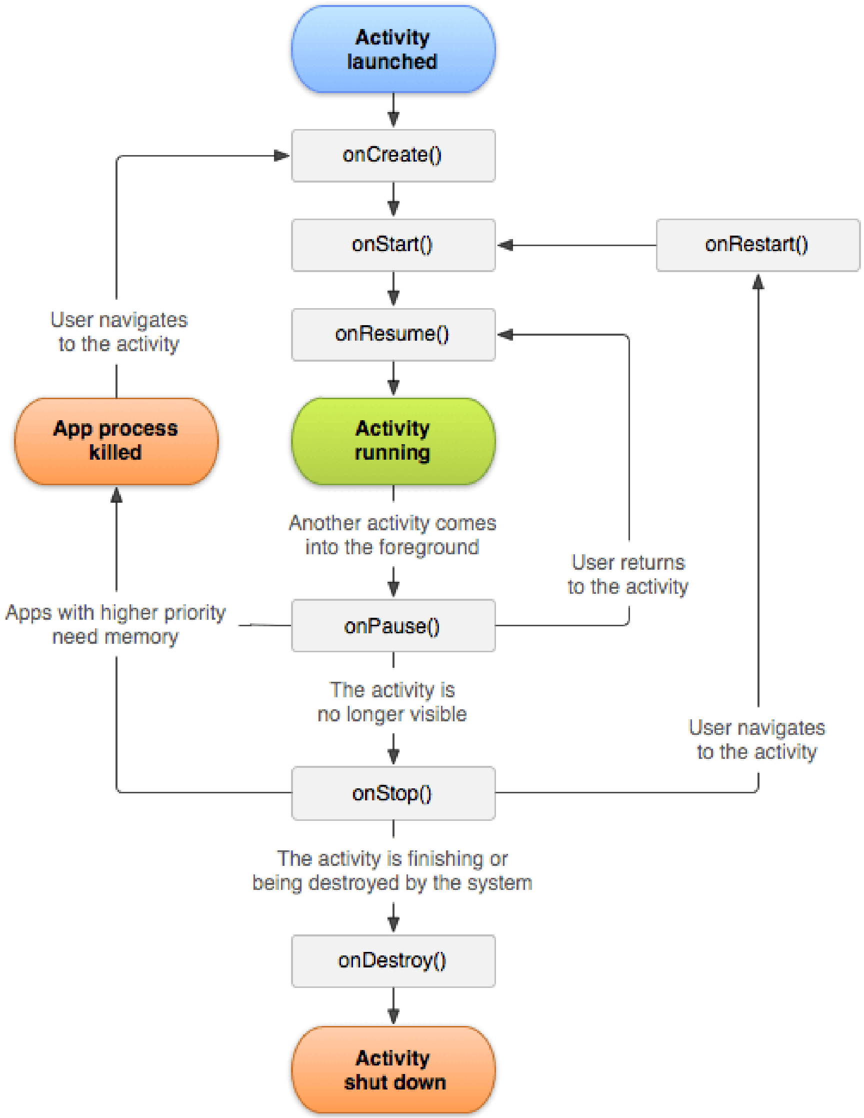

Figure 4.2: Android Activity Lifecycle.

Notably **onCreate()** is initially called when the activity is called. This is where initialization is performed for the rest of the cycle of the activity. Next **onStart()** provides a way of displaying the activity on the UI. This is where any XML code will be inflated to provide UI elements such as the graph. Another relevant method is **onDestroy()** which terminates the activity. This method is called when the device is rotated so rotating the device during an app session where data is being visualized will result in the UI being cleared and initialized. This is clearly undesirable as the user will lose the data displayed in the graph. Thus, another requirement for the app will be to keep the same orientation throughout a session, meaning that the screen will not be rotated whilst the app is running. Furthermore, as visualization is a key feature of the app, it is desired to have as much screen real estate as possible to display information. Thus, the device orientation will be locked to landscape during app sessions.

The app can be designed to be multi-activity with one activity per component in the system diagram as in figure 4.3. However, it is possible to design an app with a single activity and multiple fragments instead as in figure 4.4.

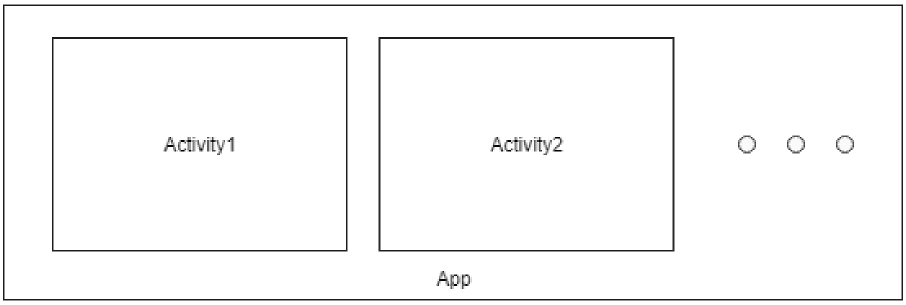

Figure 4.3: Multiple Activity App.

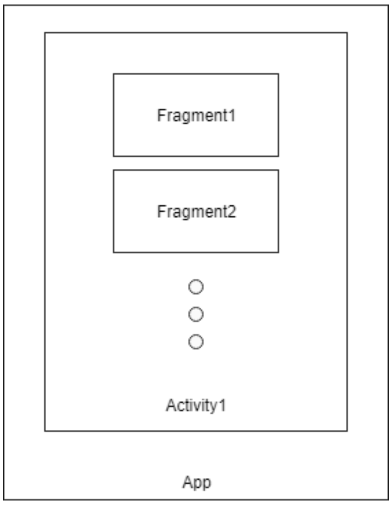

Figure 4.4: Single Activity App.

A fragment is also a single focused thing that a user can perform. However, fragments are part of activities. This is beneficial as fragments can be reused often due to their nature (such as a fragment for displaying text) and they are more lightweight than activities in terms of setup: it is possible to have fragments inside fragments. Furthermore, transactions (handoffs) between fragments is faster than between activities and common UI elements are easier to manage in a single activity app such as the action bar. Also, it is easier to manage subcomponents as the context is always the same (the context is the common namespace used by all components of a given activity).

Thus, a single activity app with multiple fragments is preferred. Multi-activity, multi-fragment apps also exist, but the system architecture does not warrant such a complex design.

4.3 Hardware

The project requires multiple pieces of hardware to fully recreate the hemodynamic monitoring system described in chapter 3. The following have been sourced in adherence to project specifications:

- A laptop to simulate the sensor.
- A Bluetooth development kit to simulate the reader. The Bluegiga DKBT Bluetooth Smart Ready Development Kit [38] was chosen as it enables serial communication through the development board and supports Bluetooth 4.0.
- A mobile device supporting Android 4.3 and above to develop the Android application. The Samsung Galaxy Tab A (10.1 inch version) was chosen as it is a mid-range tablet with a big enough screen to comfortably visualize data. Furthermore, the tablet supports Bluetooth 4.0 and Android 4.3 and above.

Chapter 5

Charting Library Selection

Chapter 4 laid out a system architecture with different components. The following chapters detail the design choices pertaining to each component starting with the real-time chart.

5.1 Library Search

One of the most important features of the app is to display blood pressure data in real-time. Thus, the charting library used plays a big role in the overall user experience. The main requirement for the Android charting library is the support of real-time line graphs.

The most important Android charting libraries were found by searching for all Android open-source charting libraries available on Github [39] and selecting those that had more than 100 stars in their repositories. They are compared below:

Charting Library	Number of Github Stars	Latest commit	Real time support
SciChart	Unknown	Unknown	Yes
MPAndroidChart	20.3k+	Less than one day ago	Yes
HelloCharts	5.4k+	Less than a year ago	No
WilliamChart	3.6k+	Less than a year ago	No
GraphView	1.8k+	Less than one day ago	Yes
AChartEngine	500+	Less than a year ago	Yes
AndroidPlot	100+	Less than a month ago	Yes

Table 5.1: Android Charting Libraries Comparison.

SciChart is not an open-source library. However, it was included as the company behind it claimed that it is ”the fastest Android Chart” [40].

Given that 1,000+ stars is a good indicator of maturity and usefulness of a repository, AChartEngine and AndroidPlot can be discarded. Furthermore, they are not being actively maintained; AChartEngine was last updated over a year ago.

HelloCharts and WilliamChart can be discarded as well since both libraries do not offer real-time support.

This leaves SciChart, MPAndroidChart and GraphView as candidates which have more than 1,000 stars on Github and are being regularly updated.

5.2 Library Performance

To test the performance of the 3 aforementioned libraries, the following experiment was conducted:

- A single activity application was produced using each library to display a graph.
- At the specified update rate, one point would be dynamically added to each of 3 data series and the x-axis adjusted so that the graph would display all of the available datasets such as in figure 5.1. Points were generated from the following equation: $x(n) = 40 * rand$ to simulate real-life usage since PA pressure varies between those values.
- The experiment was conducted for 2 minutes to gather sufficient statistics and test stability. Furthermore, FPS measurements were made using the GameBench desktop software and uploaded to the GameBench cloud for analysis.
- Experiments were conducted using GameBench's strict mode which automatically sets brightness, volume and checks if the battery is in a specified range to obtain more accurate results.

The experiment was performed for all 3 libraries. To stress the drawing engine in all instances, the tests are performed at an update rate of 10ms and 20ms.

The maximum screen refresh rate of Android devices is capped at 60fps meaning that they cannot draw a screen image faster than every 16.7ms. This means that even if the GPU can produce images at a rate faster than every 16.7ms, the device screen will only display a new

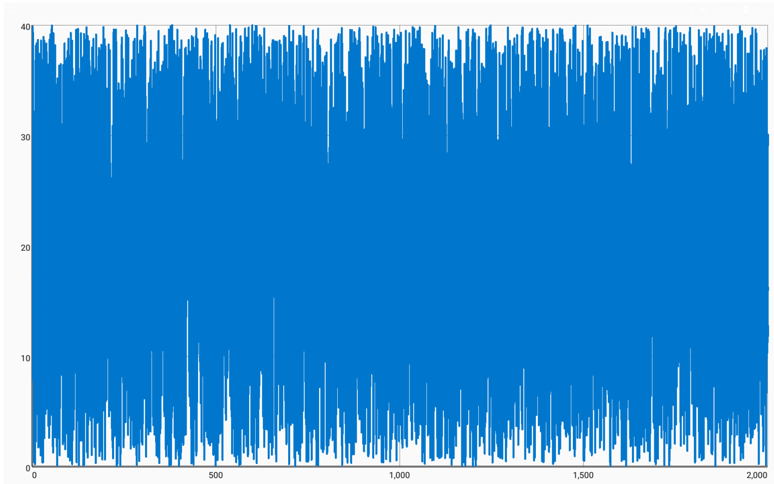

Figure 5.1: GraphView performance test.

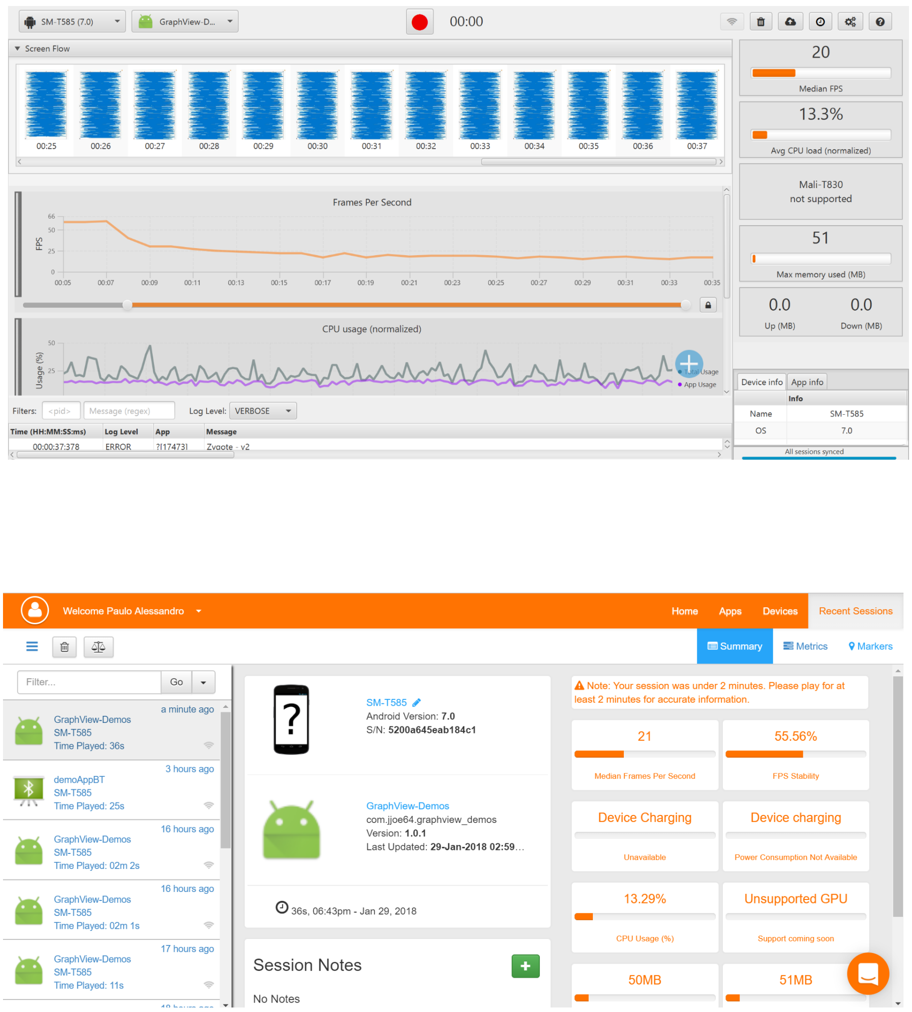

Figure 5.2: GameBench app (desktop above and cloud bottom).

image every 16.7ms. 10ms is below this rate and such an update rate tests how well the library can perform at high loads. The 20ms update rate simulates real-life conditions as the maximum reader transmission rate is 50Hz.

Below are the results in the form of an FPS graph and summarizing table for every update rate:

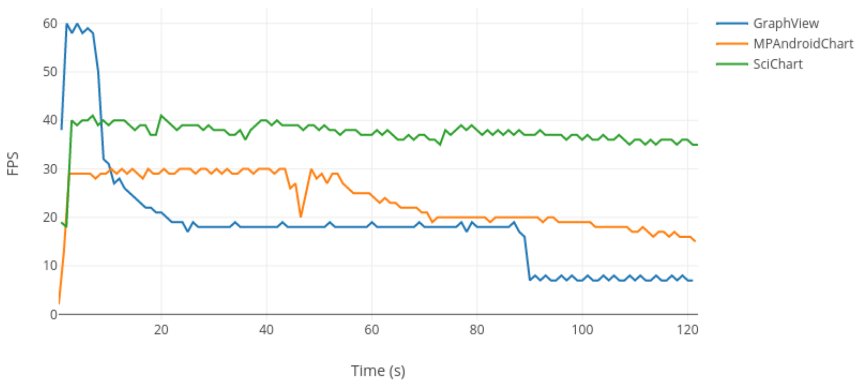

Figure 5.3: Charting library performance for 10ms update rate.

Charting Library	Median FPS (FPS)	FPS Stability (%)	Stability Index
SciChart	38	98	0.94
MPAndroidChart	23	43	0.92
GraphView	18	59	1.07

Table 5.2: Charting library performance statistics for 10ms update rate.

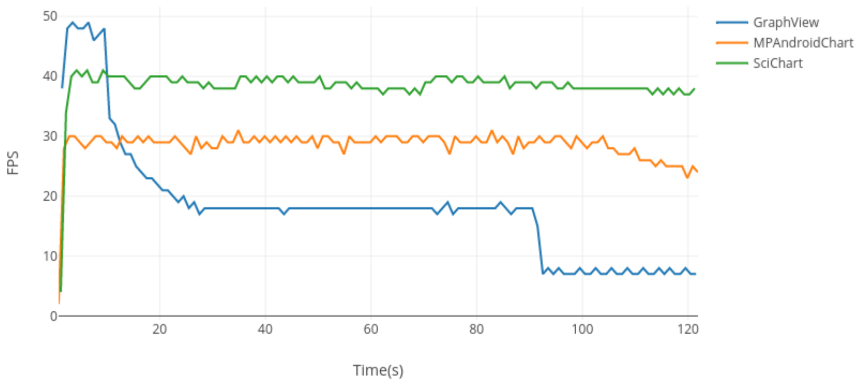

Figure 5.4: Charting library performance for 20ms update rate.

Charting Library	Median FPS (FPS)	FPS Stability (%)	Stability Index
SciChart	39	99	0.85
MPAndroidChart	29	98	1.12
GraphView	18	60	0.81

Table 5.3: Charting library performance statistics for 20ms update rate.

The metrics presented in the tables offer a different view of performance and offer second-order statistics of frame rates. Median FPS is a measure of performance.

FPS Stability is defined as the percentage of time spent within 20% of the median. It is a measure of stability as it indicates how long the app performed at a frame rate close to the median. A value of more than 50% is considered good as this means more than half the app session was spent at a frame rate close to the median.

Stability Index is defined as the average difference between pairwise samples. This is another stability measure with the added quality that it is independent of median FPS. A value of less than 1 is considered good as this means that the average FPS variation throughout the app session was less than one frame each time. The stability index is calculated as follows:

$$StabilityIndex = \frac{1}{N-1} * \sum_{i=1}^{N-1} |x_i - x_{i+1}| \tag{5.1}$$

Where N is the number of FPS samples and x_i is the ith FPS sample.

From the frame rate data recorded, it is clear SciChart is the best Android charting library since it maintains a median frame rate around 38-39fps with a stability close to 100% for both 10ms and 20ms update rates. This corroborates the company's earlier statement [40] that it is the best Android charting library.

However, SciChart is not open-source and cannot be used for this project. The next best library is MPAndroidChart with a median frame rate of 29fps at an update rate of 20ms and 98% stability. However, performance degrades at 10ms by about 6fps and stability suffers as seen in figure 5.3 where the app's frame rate decreases throughout the session.

It is worth noting that in the first 10 seconds of the session, GraphView is the best performing library with a frame rate close to 50-60fps. However, its performance severely degrades as frame rate drops below 20fps after approximately 20s and then below 10fps after a minute and a half.

In conclusion, SciChart is the best Android charting library for real-time graphs. However, it is proprietary. Thus, MPAndroidChart is chosen for this project as it is the best performing open-source Android charting library with real-time support.

Chapter 6

Signal Processing

This chapter discusses the creation of a dataset of PAP waveforms and metrics to evaluate feature extraction and abnormality detection algorithms.

6.1 Introduction

Blood pressure is a periodic waveform that exhibits several characteristics such as an initial peak during systole, valley during diastole and possibly another dicrotic valley/peak in between. Clinicians examine the waveform to assess the overall health status of patients. However, it is not always possible to process such large amounts of data, especially when measurements are taken continuously. For this reason, meaningful features are extracted from the waveform and are used as a proxy. Below are the chosen features for extraction and their relevance:

- **Systolic pressure**: Research studies have shown through epidemiological and treatment studies that systolic, diastolic and mean pressures are the most important features in risk prediction of hypertension, one of the leading causes of heart failure [41]. Thus, these features will be included in the app.
- **Diastolic pressure**: Same as systolic pressure.
- **Mean pressure**: Same as systolic pressure.
- **Heart Rate**: It is often used as a measure of general health and has been shown to be predictive of heart failure patient outcomes [42].
- **Maximum rate of pressure change**: It is a key measure of ventricular contractability [43]. This is useful for patients who have experienced heart failure or older patients as their ventricle walls will be thinner and thus, will have a slower blood ejection from the

heart.

- **Dicrotic notch pressure**: It has long been used as a measure of arterial stiffness [44]. This is another measure along with maximum rate of pressure change of ventricular anatomy degeneration.
- **Dicrotic peak pressure**: Same as dicrotic notch pressure.

6.2 Physiological Signal Dataset Creation

It is important to have a ground truth to compare against algorithm outputs for their validation. In this case, it was necessary to find a dataset of PAP measurements which also contained values for the features described previously.

PhysioNet [45] is the most well-known resource for physiological signals. It provides access to multiple collections of physiological signals including blood pressure through PhysioBank as well as open-source software to interact with the resources. PhysioBank includes many resources such as an ATM to visualize waveforms from any of its databases and the WFDB toolbox (which can be found in the appendix A in the Final-Year-Project repository) to access the databases through Matlab which was installed for this project.

Additionally, MIMIC-III [46] is "a large, publicly-available database comprising de-identified health-related data associated with approximately sixty thousand admissions of patients who stayed in critical care units". It is very useful as it is publicly available, contains data from a wide array of patients and contains high resolution data along with notes from lab results, doctor observations or even diagnoses. Thus, access was requested and the database installed for this project.

Finally, MIMIC-III offers access to physiological signals and annotations. However, the database including all the waveforms has not been released yet. This is due to data privacy issues as all data is anonymized which means that waveforms have to be associated to a given patient through a record matching process and more waveforms may become available as time goes on. The result of this matching process is hosted in PhysioBank, more specifically in the MIMIC-III

Waveform Database Matched Subset database. In this database, there are records for each patient containing both waveform and numerics data. Waveforms are physiological signals such as PAP whereas numerics are vital signs such as systolic pressure or heart rate. Thus, the MIMIC-III Waveform Database Matched Subset database was used for the dataset creation.

In summary, figure 6.1 shows the interaction between Matlab and PhysioNet during the dataset creation.

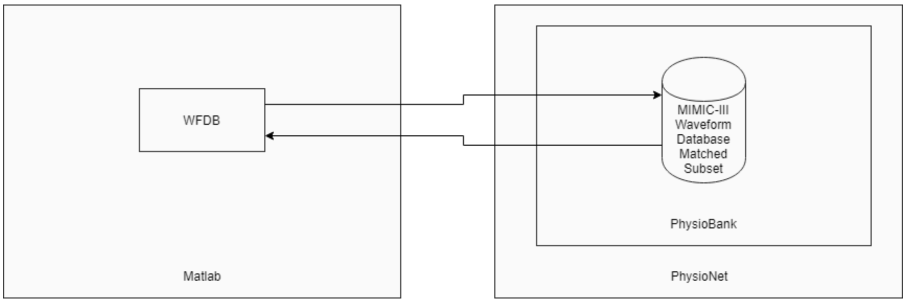

Figure 6.1: Diagram of Matlab-PhysioNet Interaction.

The PAP dataset was created using the logic in the flow chart of figure 6.2 and can be obtained from Appendix A in the repository MIMIC-PAP-Dataset or can be generated through the **getPAPDataset** script in the Final-Year-Project repository.

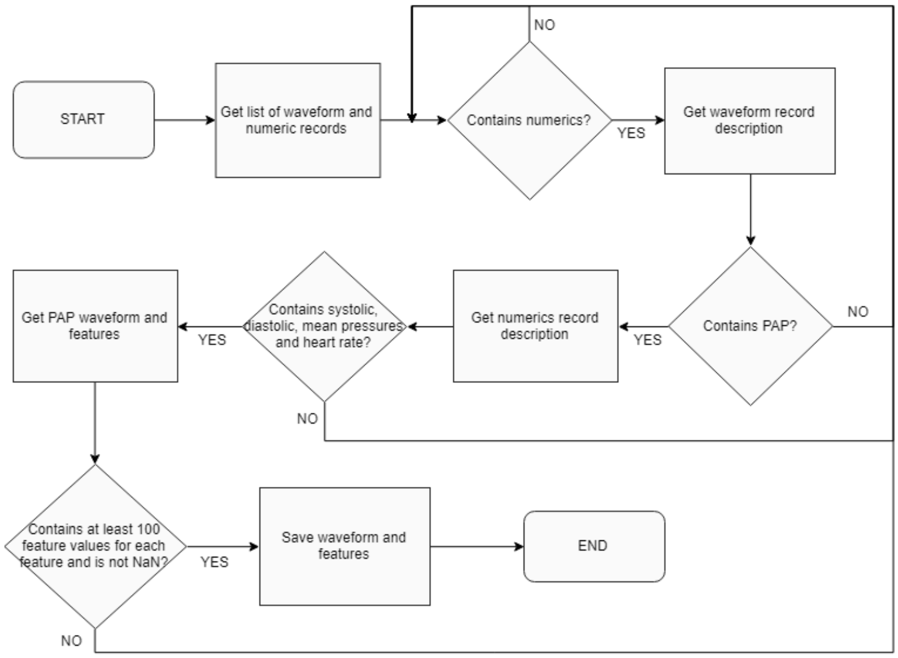

Figure 6.2: Dataset Creation Flow Chart.

The resulting PAP dataset was narrowed down to 589 records from the original 10282 from the MIMIC-III Waveform Database Matched Subset as seen in figure 6.3.

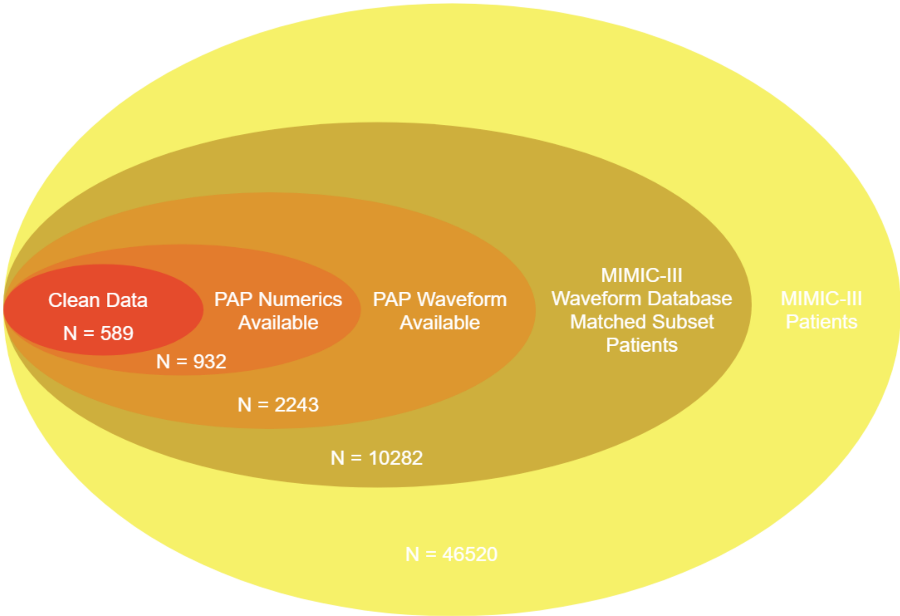

Figure 6.3: Venn Diagram of PAP data from MIMIC-III.

Furthermore, some records were discarded to obtain a final 405 records with the following features:

- Each record name is in the same format as the MIMIC-III Waveform Database Matched Subset: **pXX/pXXNNNN/pXXNNNN-YYYY-MM-DD-hh-mm** where XXNNNN is the matching MIMIC-III Clinical Database Subject_ID, and YYYY, MM, DD, hh, and mm are the surrogate year, month (01-12), and day (01-31), and the real hour (00-23) and minute (00-59).
- All PAP waveforms have a duration of 6000 seconds and were sampled at 125Hz. Also, they are not all NaN (for data quality issues as many waveforms were all NaN values).
- All features (systolic, diastolic, mean pressures and heart rate) have a duration of 6000 seconds and were sampled at 0.0167Hz (one value every minute). Thus, the 6000 second requirement means that every record will have $\frac{6000}{60} = 100$ feature values.
- All PAP waveforms yielded at least 1000 onsets using the WABP function with no more than a 10 second duration between consecutive onsets. This is to ensure the waveform quality is good enough for any further processing and that there will be no excessively

large beats due to WABP missing some onsets. This is the reason the number of final records was reduced from 589 to 405.

Noticeably, the sampling frequency of numerics is much larger than that of waveforms: 1 minute compared to 8ms. Thus, estimated feature values will have to be averaged and compared to their true value for every 60 second period.

6.3 Feature Extraction

In this section, feature extraction algorithms will be compared using the PAP dataset from the previous section. First, the average RMSE is defined as the metric to be used for algorithm validation. Below are some definitions:

x_m is the true feature value for the mth 60 second period.

$\hat{x}_b$ is the feature estimate for a given onset b.

$\hat{x}_{avg,m}$ is the averaged feature estimate for the mth 60 second period.

B_m is the number of beats for the mth 60 second period.

M_n is the number of true feature values for nth record (in this case this is always 100).

N is the number of records.

Thus, the averaged RMSE over all records for a given algorithm is defined as:

$$averageRMSE = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\frac{1}{M_n}\sum_{m=1}^{M_n}(\hat{x}_{avg,m} - x_m)^2} \tag{6.1}$$

$$\hat{x}_{avg,m} = \frac{1}{B_m}\sum_{b=1}^{B_m}\hat{x}_b \tag{6.2}$$

This metric allows a direct comparison between algorithms as the one with the lowest average RMSE can be chosen as the best. Without the outer averaging, the formula can be repurposed as the average RMSE for a given record and further, for a given m we obtain the RMSE for a given 60 second period.

The main criticism of this metric comes from the fact that a comparison is being made between a true feature value x_m and the average of estimates for beats within that 60 second period $\hat{x}_{avg,m}$. It is not possible to determine whether $\hat{x}_{avg,m}$ is a consistent estimator for x_m. However, if we assume that PAP measurements are slowly varying then $\hat{x}_{avg,m}$ will approach x_m as the number of beats used increases. This assumption is associated with short-term blood pressure variability. In a study by Staessen et al. [47], intermittent hemodynamic monitoring was conducted over a 24-hour period during which readings would be taken at different intervals. As a result, it was found that there was close agreement between a beat-by-beat analysis and sampling at intervals up to 30 minutes. The difference in pressure values between all measurement methods averaged only 1mmHg for systolic and diastolic blood pressure. Thus, it is safe to assume that PAP measurements are slowly varying in general.

The **validateAlgorithms** script which can be found in Appendix A in the repository Final-Year-Project was used to validate the following algorithms. It provides a framework for algorithm validation as results from a given algorithm are stored in the corresponding cell array (one for each feature). Thus, a new algorithm can be implemented and validated by storing its output in the corresponding cell array.

6.3.1 Beat Onset Detection

Beat onset is defined as the point at which a new beat begins as shown in figure 6.4. The detection of these onsets is critical to feature extraction as features are associated with a single beat.

As such, it was necessary to have a reliable algorithm that has been proven to work. The WABP algorithm [48] is recommended by PhysioNet as it has been tested on its databases. Zong et al.

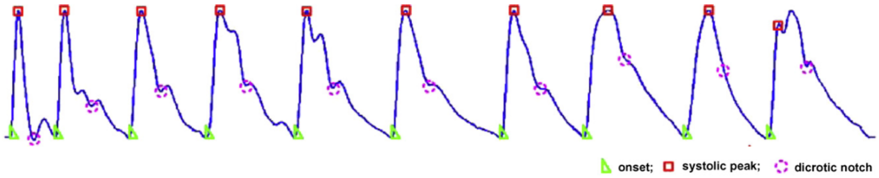

Figure 6.4: Example PAP Waveforms.

[48] achieved a 99.71% success rate in detecting onsets that were manually annotated in the MIT-BIH Polysomnographic Database. The logic of the algorithm is detailed in figure 6.5.

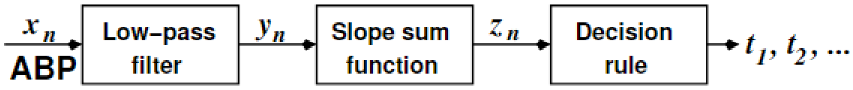

Figure 6.5: WABP Algorithm Flow Diagram [48].

First, the signal is passed through a low-pass filter with 3dB cutoff at 16Hz to remove unwanted noise. Then, the filtered signal is converted into a weighted slope sum function (SSF) in which the signal is windowed and upward slopes enhanced through the following function:

$$z_i = \sum_{k=i-w}^{i} \Delta u_k, \Delta u_k = max(y_k - y_{k-1}, 0) \tag{6.3}$$

Where w is the window length.

The SSF function enhances points where the signal is rising and suppresses those in which the signal is not. Finally, an adaptive threshold is used to determine whether SSF maxima are onsets since for example the dicrotic peak could yield a peak in the SSF function.

The WABP algorithm was tested on the PAP dataset and yielded an average of 6034 beats per record. Each record is 6000 seconds long and assuming an average beat duration of 0.8s, this results in a $\frac{6034}{\frac{6000}{0.8}} = 80.45\%$ detection rate for onsets.

Furthermore, it is possible that the algorithm may not detect some beats and there may be periods where no beat is detected. For the PAP dataset, the average number of beats with duration larger than 2 seconds was 307. Thus, less than 5.1% of beats detected will contain more than one beat.

It is noted that all of the waveforms tested were sampled at 125Hz which is high resolution compared to the maximum sampling rate of 50Hz for the reader. Thus, it is necessary to define a minimum sampling rate at which waveforms must be sampled for efficient feature extraction. The feature with smallest duration in the waveform is the dicrotic notch and peak. Since the average ejection time index (duration between dicrotic notch and peak) is 0.1s [49], the minimum sampling frequency is taken to be 10Hz for an accurate representation of the PAP waveform.

Furthermore, signals are preprocessed before feature extraction to remove unwanted noise. From this point forward, x is defined to be the filtered signal for a single beat.

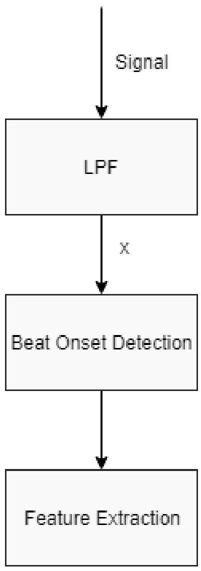

Figure 6.6: Feature Extraction Overview.

Figure 6.6 shows that before feature extraction, the signal is filtered with a Butterworth first order low pass filter with 3dB cutoff at 25Hz to remove unwanted noise.

6.3.2 Systolic Pressure

The most noticeable feature in blood pressure waveforms is the systolic peak which is defined as the maximum pressure during systole as seen in figure 6.4. Peak detection is an important part of signal processing for which many algorithms have been proposed from windowed-thresholds to transform analysis to increasingly sophisticated machine learning models. Conversely, more generic algorithms will require settable free parameters which is not desirable. It is important to keep in mind that the algorithms will run in conjunction with a real-time application so speed and simplicity will be favored over complexity to a degree. Following are the proposed methods for systolic pressure extraction.

Maximum

In this method, the systolic pressure is taken to be the maximum of the waveform.

Figure 6.7: Maximum Algorithm.

This method has the advantage of being simple as it consists of traversing signal values and selecting the maximum. However, it is not robust as noise can make the maximum value be arbitrarily large. This is in part mitigated by the low pass filter beefore feature extraction.

Inflexion and Zero-Point Crossing

In this method, the first derivative of the signal is used. At the inflexion points, the derivative changes concavity (the second derivative changes sign) and marks the highest/lowest rate of change in the vicinity. This is useful as the systolic uptake is almost often the sharpest rise in the PAP waveform and consequently, produces the largest inflexion in the first derivative. The systolic pressure is taken to be the first zero-point crossing after inflexion of the first derivative.

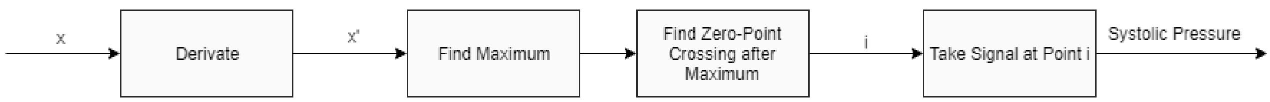

Figure 6.8: Inflexion and Zero-Point Crossing Algorithm.

This method has the property that it will select the first point in the systolic uptake where pressure stops changing as the systolic pressure. However, it means that it is possible to mistake the anacrotic peak as the systolic peak. An anacrotic notch as seen in the last beat of figure 6.4 is a variation of the PAP waveform which may be present due to aortic stenosis. Its presence means that there will be a peak before the systolic peak and the two may be confounded.

Find Peaks

This method is an enhanced version of the maximum method as the maximum is obtained from the set of points which are larger than their neighbors.

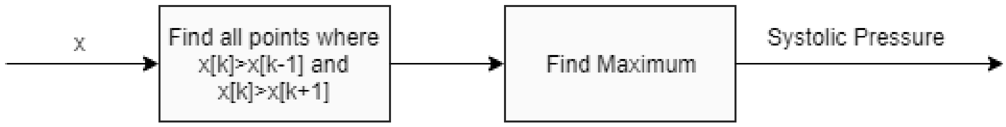

Figure 6.9: Find Peaks Algorithm.

This is beneficial as the maximum is taken across all local maxima. However, this method will fail to return peaks if the maximum is lying in a plateau as either $x[k] > x[k-1]$ or $x[k] > x[k+1]$ may return false.

AMPD

The method Authomatic Multi-Scale Peak Detection (AMPD) [50] is based on the calculation of the local maxima scalogram (LMS) M of the signal which are the scale dependent local maxima of the signal. The row-wise summation of M yields a distribution of local maxima γ which can be used to rescale the LMS by removing all elements which are not neighbors of the local maxima. Finally, the column-wise standard deviation of M is taken and columns where it is equal to 0 are defined to be the indices of the peaks.

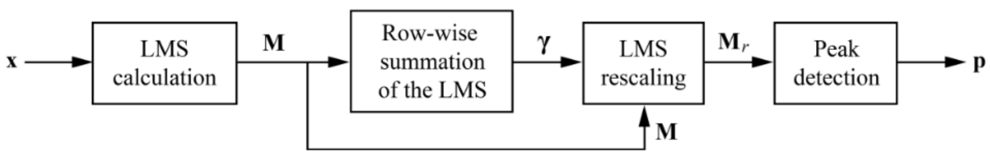

Figure 6.10: AMPD Algorithm.

This method has the advantage of being robust to noise as the computation of the LMS includes a moving window and averaging which reduces noise. However, it is significantly more complex than the previous methods which is expected to be reflected in its time complexity.

Results

Table 6.1 shows the average RMSE over the PAP dataset for the different methods of systolic pressure extraction.

Algorithm	Max	Inflexion	Find Peaks	AMPD
Average RMSE	4.6675	4.6214	4.6094	4.4203
Average Std	8.9477	8.2592	8.8485	8.8055

Table 6.1: Algorithm performance across all patients for systolic pressure estimation.

As expected, more complex methods give better results. However, the decrease in the average RMSE is minimal as it decreases by about 5% going from the maximum method to AMPD. Furthermore, the decrease in error comes at the cost of computational complexity.

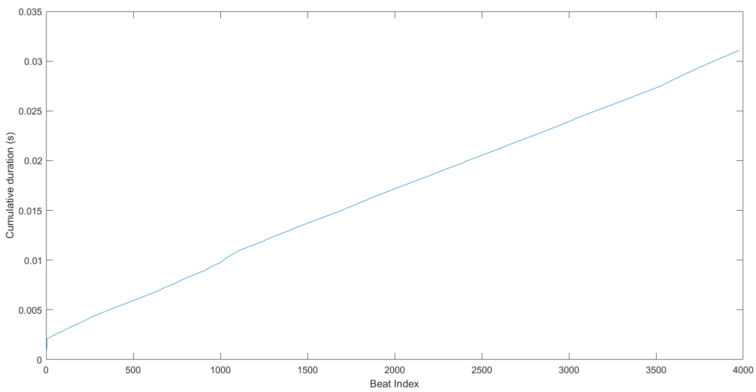

Figure 6.11: Time complexity of max algorithm.

The figures show the computational complexity of the systolic pressure extraction methods which are all linear in the number of beats. In practice, the find peaks and AMPD methods are 100 times slower than the maximum and inflexion and zero-point crossing methods. Thus, as a compromise between accuracy and complexity, the inflexion and zero-point method is chosen for systolic feature extraction.

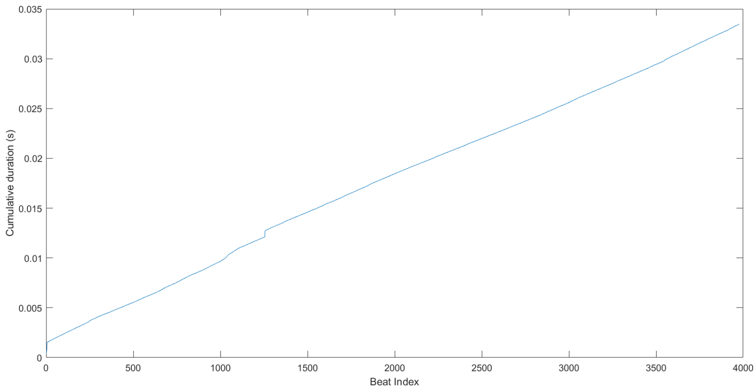

Figure 6.12: Time complexity of inflexion algorithm.

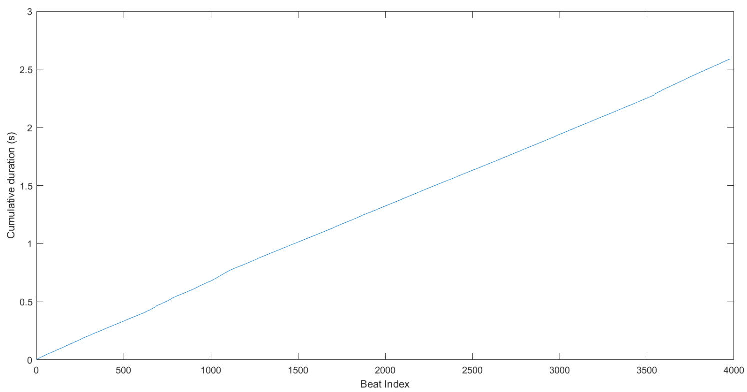

Figure 6.13: Time complexity of find peaks algorithm.

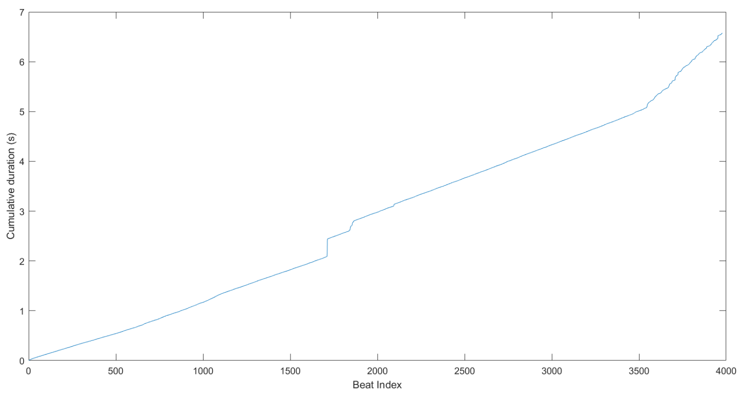

Figure 6.14: Time complexity of AMPD algorithm.

6.3.3 Diastolic Pressure

Diastolic pressure is defined as the lowest pressure in the PAP waveform. This means that it is usually at the end of a beat in the lowest valley of the signal.

Minimum

This method takes the minimum of the waveform to be the diastolic pressure. It is simple, but not robust to noise.

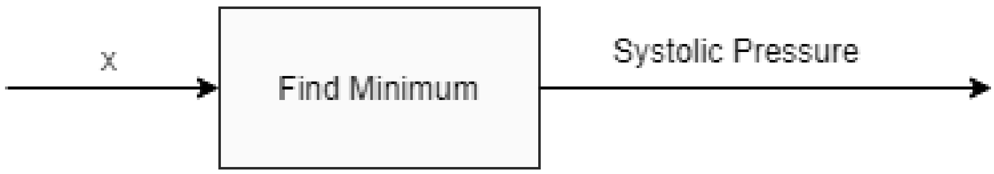

Figure 6.15: Minimum Algorithm.

End of Beat

This method takes the end of the signal to be the diastolic pressure. The end of the signal can also be viewed as the onset of the next beat which could be some point between the true diastolic pressure and the foot of the systolic rise.

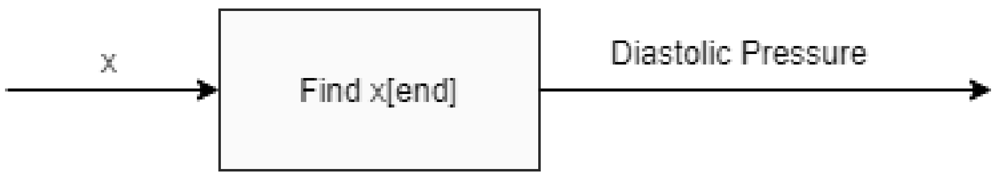

Figure 6.16: End of Beat Algorithm.

Results

Table 6.2 shows the average RMSE for the diastolic pressure extraction methods.

It shows that performance of both methods is similar with a slight edge to the end of beat method. However, notice how the average standard deviation for that method is larger meaning

Algorithm	Min	Waveform End
Average RMSE	3.1336	3.1126
Average Std	4.8786	5.0266

Table 6.2: Algorithm performance across all patients for diastolic pressure estimation.

that results are not as consistent across beats. The minimum method is chosen as it has the added property that it may select the true diastolic pressure whereas the end of beat method is limited to always selecting the last value.

6.3.4 Mean Pressure

Mean pressure is defined as the average pressure during a cardiac cycle.

One-Third, Two-Third

A common rule of thumb for mean pressure is to use a weighted average of systolic and diastolic pressures in a 1:2 ratio as it is assumed that diastole takes twice as long as systole. This method is only an approximate of the real mean pressure.

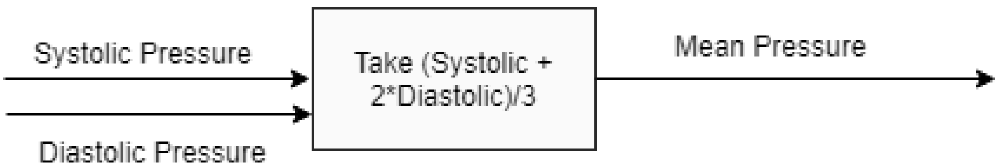

Figure 6.17: One-Third, Two-Thirds Algorithm.

Integral

This method gives the true value of the mean pressure per its definition.

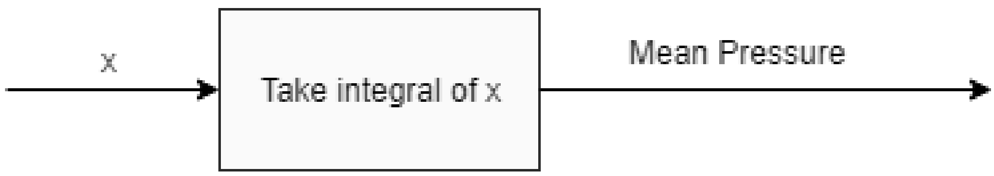

Figure 6.18: Integral Algorithm.

Results

Table 6.3 shows the average RMSE for the mean pressure extraction methods.

Algorithm	One-Third, Two-Thirds	Integral
Average RMSE	4.1686	65.0398
Average Std	7.8237	206.2547

Table 6.3: Algorithm performance across all patients for mean pressure estimation.

Interestingly, the integral method which gives the true mean pressure has a large error compared to the rule of thumb. This can be explained by the fact that the beat onset detection algorithm may not detect all beats and thus, some beats may contain more than one beat. This makes the integral bigger which results in a larger error. Thus, the rule of thumb method is used for mean pressure extraction.

6.3.5 Dicrotic Notch

The dicrotic notch is one of the minima following the systolic peak and before the dicrotic peak.

Straight-Line Maximum

This method defines the dicrotic notch as the minimum in the difference between the straight-line between systolic peak and diastolic valley and the signal.

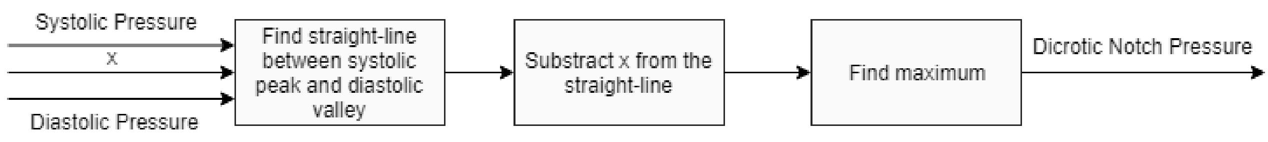

Figure 6.19: Straight-Line Maximum Algorithm.

The drawback to this method is that it requires an estimate of both systolic and diastolic pressures so it is dependent on two different estimates.

Inflexion

Similar to the systolic inflexion method, this method finds the lowest local minima after the systolic peak.

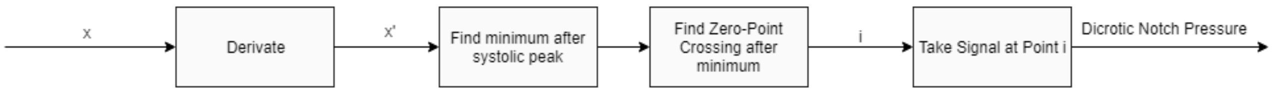

Figure 6.20: Inflexion Algorithm.

Results

The PAP dataset only includes true feature values for systolic, diastolic, mean pressures and heart rate. Thus, the dicrotic notch pressures were evaluated against the estimate diastolic pressures. This means that the resulting errors will not be absolute, but a comparative measure between algorithms as the dicrotic notch pressure is closest to diastolic pressure out of all features.

Table 6.4 shows the average RMSE for the dicrotic notch pressure extraction methods.

Algorithm	Straight-Line Maximum	Inflexion
Average RMSE	0.2587	5.4871
Average Std	0.4274	665.7516

Table 6.4: Algorithm performance across all patients for dicrotic notch pressure estimation.

Interestingly, the straight-line maximum method gives a very low error. Upon further inspection, it is noted that this method produced estimates which are almost always the same as the diastolic pressure estimates. This is due to the fact that the straight line usually has a very steep slope and the minimum in a waveform is often repeated since it is slowly-varying and sampled at 125Hz. Thus, there are usually samples before the diastolic valley which have the same value that give a sharp maximum in the method. For this reason, the inflexion method was chosen for dicrotic notch pressure extraction.

6.3.6 Dicrotic Peak

Straight-Line Minimum

This method is similar to the one for dicrotic notch pressure, but the minimum of the substraction is taken to be the dicrotic peak pressure.

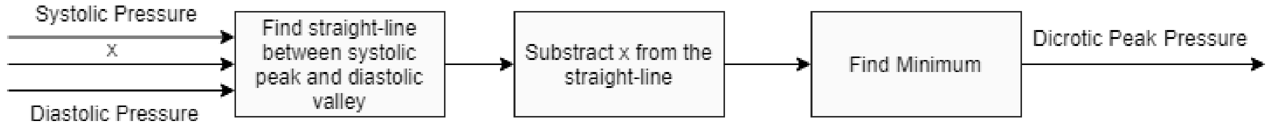

Figure 6.21: Straight-Line Minimum Algorithm.

Maximum

This method requires one less input compared to the previous and finds the maximum of the waveform after the dicrotic notch.

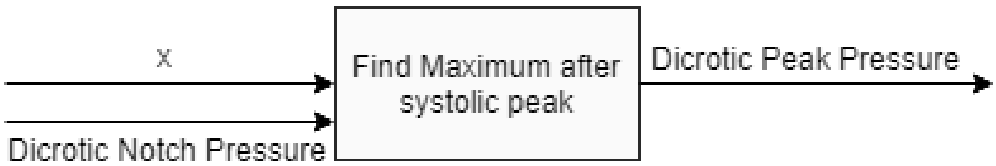

Figure 6.22: Maximum Algorithm.

Results

The PAP dataset only includes true feature values for systolic, diastolic, mean pressures and heart rate. Thus, the dicrotic peak pressures were evaluated against the estimate systolic pressures. This means that the resulting errors will not be absolute, but a comparative measure between algorithms as the dicrotic peak pressure is closest to systolic pressure out of all features.

Table 6.5 shows the average RMSE for the dicrotic peak pressure extraction methods.

The high average RMSE should be disregarded as they come from comparing estimate dicrotic peak pressure values to estimate systolic pressure values. Instead, the maximum method has

Algorithm	Straight-Line Minimum	Maximum
Average RMSE	21.3614	17.9023
Average Std	1529	1501

Table 6.5: Algorithm performance across all patients for dicrotic peak pressure estimation.

a lower error compared to the straight-line minimum method. Thus, the maximum method is chosen for dicrotic peak pressure extraction.

However, one possible explanation for the high error values is that some waveforms do not possess a dicrotic peak and instead have an incisura as seen in figure 6.4 in the penultimate beat. An incisura is the lack of dicrotic notch and peak due to ventricular anatomy degeneration which makes the waveform after systole look like a steady decline. In this case, not much can be done for dicrotic peak pressure extraction except assign it the value of the dicrotic notch pressure which is what the maximum method does.

6.3.7 Max Rate of Change

The maximum rate of pressure change (max dp/dt) is defined as the maximum derivative of the PAP waveform during systolic uptake.

Thus, it can be approximated by taking the derivative between the beat onset and the systolic peak. The estimate becomes:

$$\hat{x}_b = \frac{y_{systolicIndex} - y_1}{l_{b,1:systolicIndex}} \tag{6.4}$$

Where $y_{systolicIndex}$ is the systolic pressure estimate and $l_{b,1:systolicIndex}$ the time between beat onset and systolic peak.

However, the PAP dataset does not provide true values for this feature. Thus, we define the approximate true values as the theoretical derivative between the current 60 second period true diastolic pressure (the closest estimate we have of the systolic foot) and the next 60 second period true systolic pressure:

$$x_m = \frac{\hat{x}_{avg,m+1,systolic} - \hat{x}_{avg,m,diastolic}}{\alpha} \tag{6.5}$$

Where α is the average duration between systolic foot and systolic peak.

Table 6.6 shows the average RMSE for different values of α.

Algorithm	$\alpha = 0.02s$	$\alpha = 0.03s$	$\alpha = 0.04s$
Average RMSE	290.2043	223.6662	391.5239
Average Std	43935	37992	54271

Table 6.6: Algorithm performance across all patients for max rate of pressure change estimation.

Whilst the extraction method is correct, it yields a large error due to the fact that α should be different for every 60 second period to reflect real life variability and that the true diastolic pressure is not always the same as the systolic foot pressure. This is reflected in the high average standard deviation which shows that estimates vary a lot between records. Nevertheless, it is found that an average duration of 300ms between systolic foot and systolic index is the most accurate.

6.3.8 Heart Rate

The heart rate is defined as the number of cardiac cycles in a given period (usually per minute and in bpm).

Whole Waveform

In this method, the estimate uses the duration of the entire beat and becomes:

$$\hat{x}_b = \frac{60}{l_b} \tag{6.6}$$

Where l_b is the duration of the current beat.

Signal Until Diastole

In this method, the estimate uses the duration of the beat until the end of diastole and becomes:

$$\hat{x}_b = \frac{60}{l_{b,1:diastolicIndex}} \tag{6.7}$$

Where $l_{b,1:diastolicIndex}$ is the duration of the current beat until the end of diastole.

It is expected that this method gives a better estimate as the previous method may underestimate the heart rate due to there being more than one beat in a given beat from the onset detection algorithm

Results

Table 6.7 shows the average RMSE for the heart rate extraction methods.

Algorithm	Whole Signal	Signal Until Diastole
Average RMSE	18.2382	23.3265
Average Std	14.3717	22.0366

Table 6.7: Algorithm performance across all patients for heart rate estimation.

Interestingly, the whole signal method performs better. This is due to the fact that the first method is more consistent in its estimates as reflected by the lower average standard deviation since the signal until diastole method relies on the diastolic pressure estimate. Thus, the whole signal method is chosen for heart rate extraction.

6.4 Abnormality Detection

The PAP waveform can present abnormalities due to various factors such as aging, ventricular anatomy degeneration or any illnesses that the patient might be suffering from. Thus, it is important to detect abnormalities which can then be flagged to clinicians for further review.

This will be performed using the previously extracted features. However, it is important to highlight the difference between short-term and long-term abnormality detection. The Android app has to visualize measurements in real-time and works on a beat-by-beat basis which means it is not designed to be operating for days or months on end. Thus, it is only appropriate to perform beat-by-beat abnormality detection as any long-term anomaly would have to be detected by analyzing previous signals over a given period which is more suited to an online process such as a web dashboard and back-end for processing. This means that the app will provide feedback based on abnormalities for users to immediately flag to clinicians instead of providing long-term insight such as a steadily rising mean pressure over the past month.

6.4.1 Abnormality Definition

To perform abnormality detection it is first necessary to define what it means for a PAP waveform to be abnormal. Smetana [51] conducted a study to evaluate the impact of revising systolic and diastolic pressure guidelines on the prevalence of hypertension in the United States based on overall mortality rates. It was found that current guidelines are too strict on the definition of abnormal and proposed looser guidelines meaning that the abnormal ranges for both systolic and diastolic pressure were enlarged. However, this meant that individuals with pre-hypertension, for which no treatment is recommended as the condition is considered benign, were also classified as abnormal. Furthermore, looser guidelines would have significant socioeconomic impacts on society as more resources would have to be diverted to the care of such patients due to over-labeling. As such, it is important to consider when a PAP waveform is abnormal and the implication of false positives/negatives.

For our purposes, abnormal waveforms will be defined as those that are not normal. To find out what normal means, ranges were extracted from the PAP dataset for every feature as in table 6.8.

Feature	Systolic Pressure	Diastolic Pressure	Mean Pressure	Heart Rate
[P10; P90]	[26; 59.7]	[11.4; 30.1]	[17.5; 41.8]	[67.9; 106]

Table 6.8: Range of values between 10th and 90th percentiles for features in the PAP Dataset.

The table shows the range of values between the 10th and 90th percentile for each feature meaning that at least 80% of values will fall in that range. Thus, an abnormal waveform could be defined as one in which its features fall outside of these ranges.

However, diastolic pressure presents a lower bound for the PAP waveform values so it is only necessary to consider upper bounds for systolic and mean pressures. Furthermore, a high heart rate is usually worse than a low heart rate as it is associated with multiple heart conditions whereas it is possible to be below guidelines and still be healthy in the case of athletes for example. Thus, any abnormality detection algorithm will only consider the lower range for diastolic pressure and the upper range for the other features.

6.4.2 Signal Quality Index

Much of the literature concerning abnormality detection for blood pressure has focused on the creation of signal quality indexes (SQI). These indexes are values given on a beat-by-beat basis to determine if they are abnormal. The most widely cited paper in the corpus is by Sun et al. [52] who created a rule-based boolean index called jSQI that is false if the beat is normal and true if any of the beat features were outside specified physiological ranges. The jSQI was originally designed for ABP waveforms meaning that values were higher than for PAP waveforms. As such, the ranges were adapted to reflect the PAP waveform:

Feature	**Abnormality Criteria**
P_s	$P_s > 70$
P_d	$P_d < 25$
P_m	$P_m > 40$
HR	$HR > 100$
D_{pdt}	$D_{pdt} < 25$
P_{dn}	$\|P_{dn} - P_d\| < 1$ or $\|P_{dn} - P_s\| < 1$
P_{dp}	$\|P_{dp} - P_{dn}\| < 1$
$P_s[k] - P_s[k-1]$	$\|\Delta P_s\| > 20$
$P_d[k] - P_d[k-1]$	$\|\Delta P_d\| > 20$
$P_m[k] - P_m[k-1]$	$\|\Delta P_m\| > 15$

Table 6.9: Abnormality criteria.

Table 6.9 shows the resulting criteria for jSQI resulting from PAP dataset ranges and guidelines from the American College of Cardiology [53]. Figure 6.23 shows the jSQI algorithm flow chart.

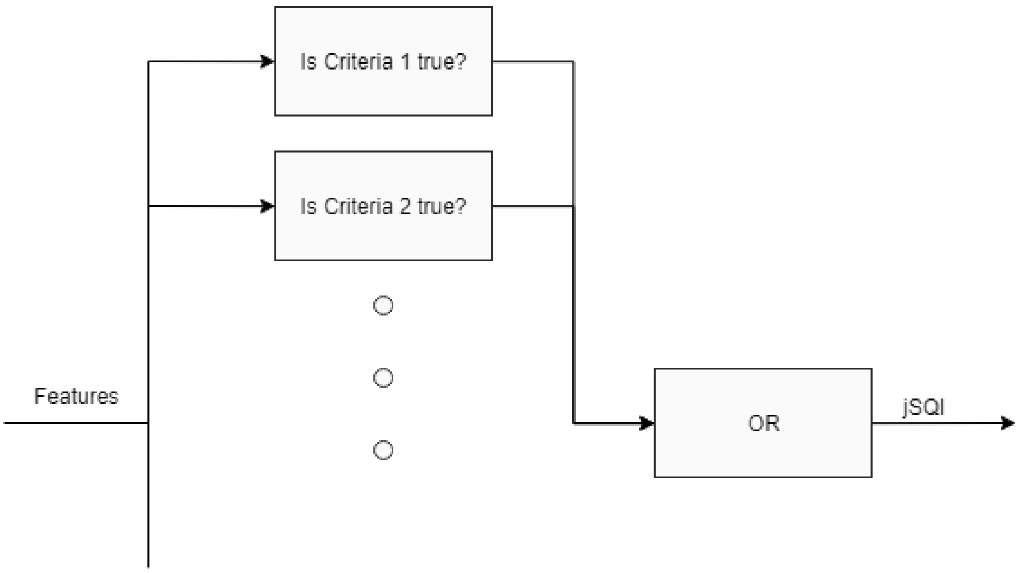

Figure 6.23: jSQI Algorithm.

This algorithm is suitable for abnormality detection as it can perform on a beat-by-beat basis, but also provides the added criteria of differences between successive beats.

6.4.3 Results

The resulting jSQI algorithm was tested to see if it could detect abnormal beats in patients with a known disease. The MIMIC-III database offers patient diagnoses by ICD9 code so it was possible to extract the list of patients with the condition by searching for code 416 associated to chronic pulmonary heart disease as in figure 6.24.

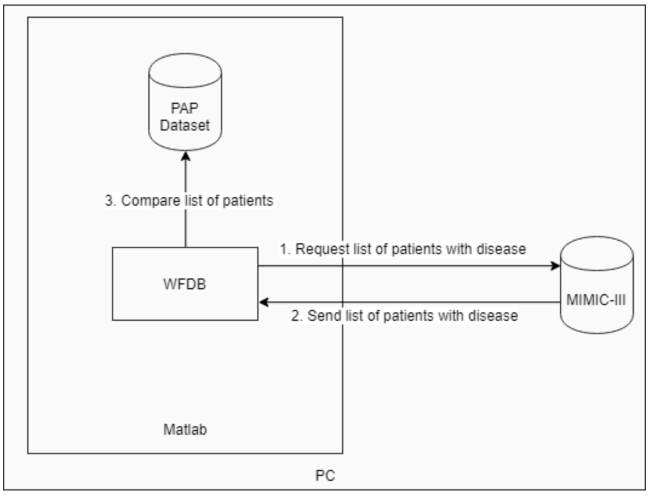

Figure 6.24: Matlab-MIMIC-III Interaction.

The list of patients was cross-referenced with the PAP dataset to find out which records belonged to patients who had the condition using the script **getPatientsWithPrimaryPulmonaryHypertension** available in Appendix A in the repository Final-Year-Project. Each of those records was inputted to the jSQI algorithm and the output recorded in table 6.10.

Patient ID	Percentage of abnormal beats
4462	99.92%
5606	89.09%
6917	100%
8548	52.28%
12104	99.39%
12581	72.15%
12849	99.95%
16873	97.56%
22766	97.96%

Table 6.10: Abnormality detection in patients with primary pulmonary hypertension.

The table shows that the algorithm is able to detect which patients are suffering from a condition which affect PAP almost all records presented more than 90% abnormal beats.

6.5 Data Issues

As with any other dataset, the PAP dataset suffers from quality issues stemming from the source of their recordings.

6.5.1 Estimate Accuracy

It was established at the beginning of the chapter that the accuracy of feature estimates depended on the number of beats in a given 60 second period to average. Figure 6.25 shows the correlation between average RMSE per record and the number of beats per 60 second period.

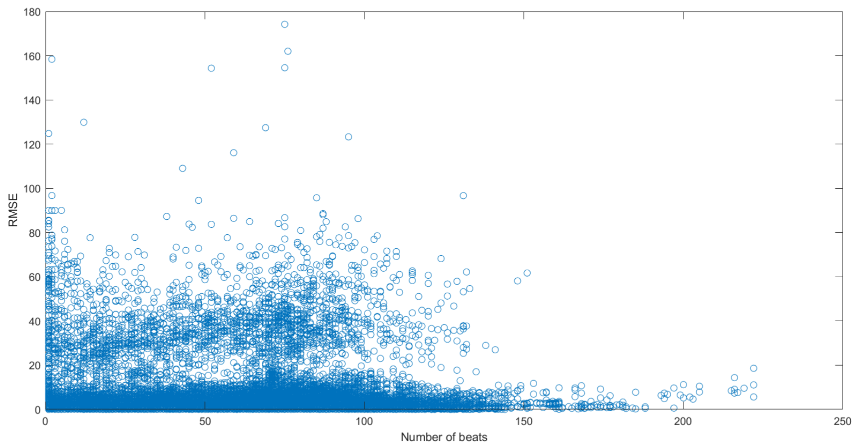

Figure 6.25: Scatter Plot of Average RMSE for Systolic Pressure for a Given Record Versus Number of Beats.

There is in fact a small negative correlation $\rho = -0.1096$ between both variables meaning that the average RMSE is weakly inversely proportional to the number of beats used. This highlights the importance of the beat onset detection algorithm in feature extraction as missing or combining beats can give misleading estimates.

6.5.2 Data Quality

Although all records in the PAP dataset passed the initial requirements, many still remain with quality issues that are not visible until the signals are plotted. Below are some examples:

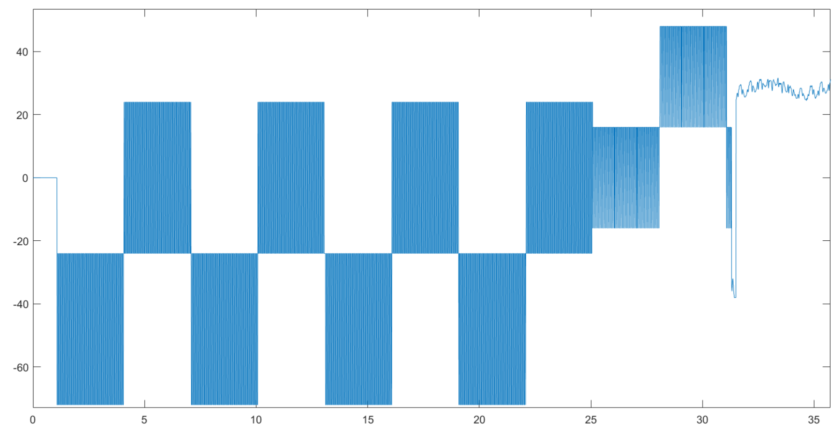

Figure 6.26: Machine noise.

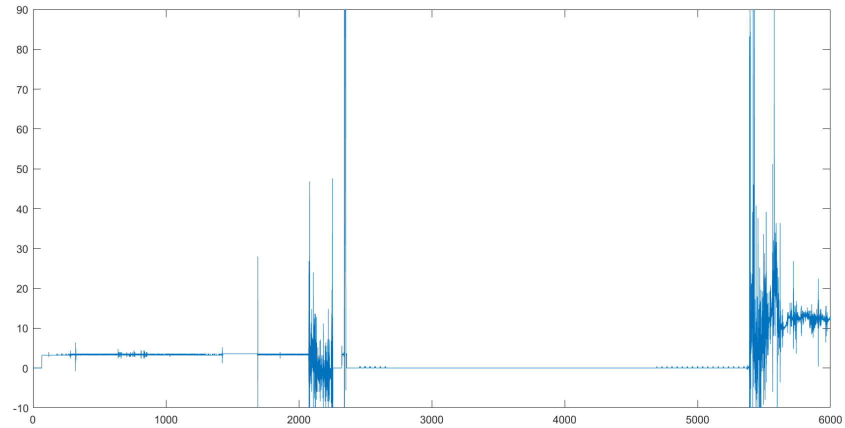

Figure 6.27: Missing data.

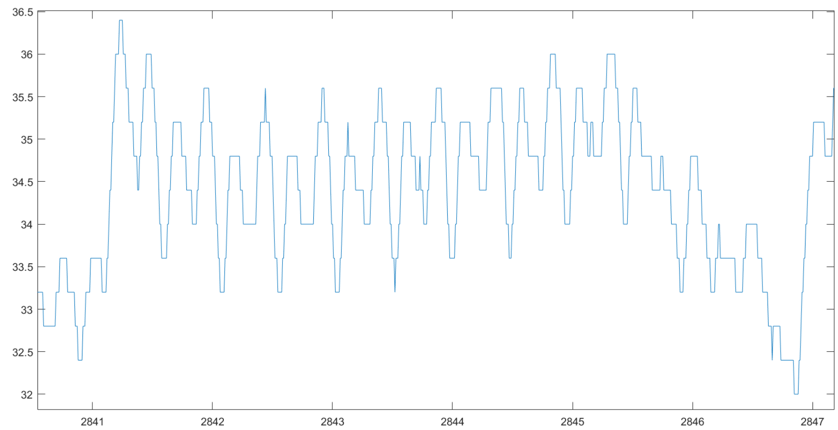

Figure 6.28: Irregular Waveform.

These data quality issues can often be categorized as follows:

- **Machine noise**: This is due to interference from another machine and often results in periodic square waves.
- **Missing data**: Values can be NaN, Inf or even 0 for an extended period of time.
- **Irregular waveforms**: This is the hardest issue to face as the signal cannot be disregarded until close inspection.

Beat onset detection again plays an important role in pruning bad data as it is assumed that any irregularities will not be contained in onsets. More safeguard measures should still be implemented to increase the accuracy of any further signal processing such as ranking records by their jSQI and removing those that fall below a threshold.

Chapter 7

System Implementation

Application development was done following the Agile approach to software development. Agile development consists of a series of short sprints lasting a few weeks at most to iteratively produce results. The approach was suitable for the project given time constraints and the fact that development was adjacent to requirement gathering. It allows for flexible planning and early feedback at the end of each sprint where a deliverable is presented to the client (in this case, the project supervisor during the weekly meeting).

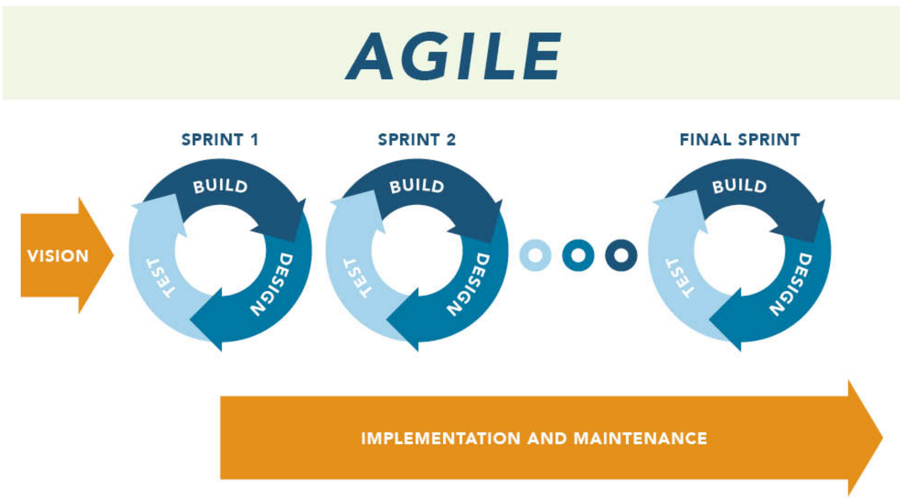

Figure 7.1: Agile Methodology Diagram [54].

7.1 Development Practices

Android Studio is the de facto IDE for development of Android apps and was used for this project. Android apps can be developed either in Java or Kotlin, a language that was recently supported by the Android OS in 2017. Java was picked for this project due to its maturity, wealth of documentation and overwhelming number of developers that know the language compared to Kotlin for future development. Care was taken to ensure usage of appropriate Java features with respect to compatible Android versions. Thus, newer Java features such as lambda expressions were replaced with legacy features such as standard methods. Also, code was written in a modular fashion to limit the number of dependencies such that each component would be easily replaceable. This often manifested in the reduction of global variables or removing unnecessary arguments in methods. Furthermore, the project was linked to a Github repository to ensure redundancy in case the local machine was lost and for version control.

Besides app performance testing, the main reason for using an Android device is the lack of Bluetooth support by Android Studio. The IDE offers support for virtual devices which can be simulated by the computer. However, the emulator does not offer wireless communication capabilities such as WiFi, NFC or Bluetooth.

7.2 App Architecture Overview

Following design guidelines present in chapter 4, figure 7.2 shows the class diagram of the Android app generated in Android Studio using the simpleUMLCE plugin.

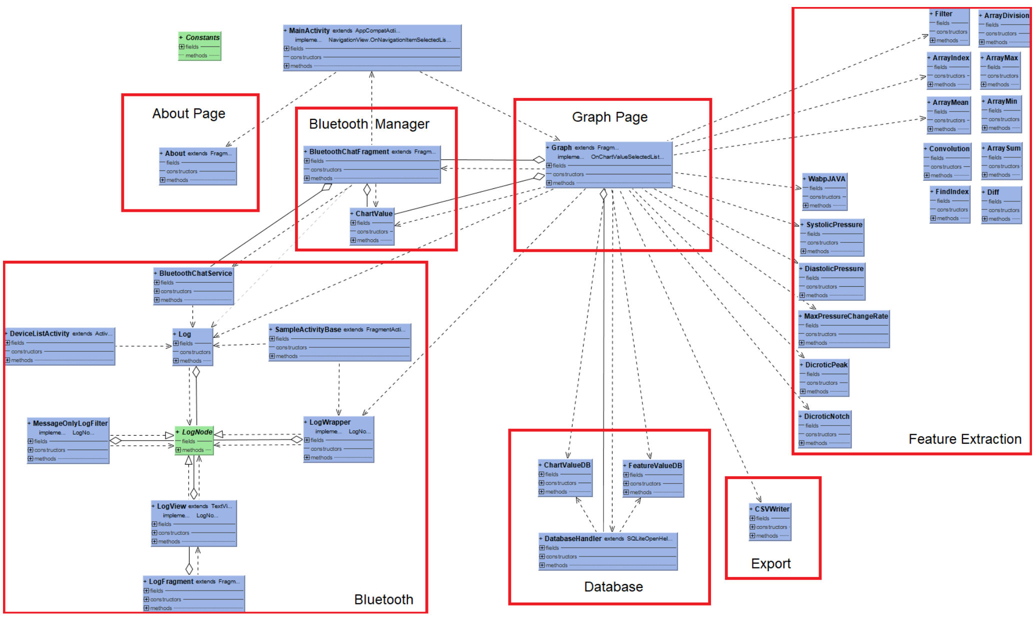

Figure 7.2: Android App Class Diagram.

As mentioned before, the app has a single activity that spans multiple fragments namely the **Graph**, **About** and **BluetoothChat** fragments. The implementations of individual features are discussed below. Also, readers are encouraged to refer back to the diagram when a class is mentioned in subsequent sections.

7.3 Two-Way Data Communication

7.3.1 Reader Simulation

The reader is a device meant to be worn by a patient that periodically excites the SAW sensor and receives blood pressure measurements. It will be simulated using a Bluetooth development kit receiving pressure data from a laptop.

PA pressure data has been sourced from the PAP dataset as explained in chapter 6. PAP measurements were obtained in MATLAB format and converted to a text file where each line corresponds to a pressure measurement.

The data format for the reader is **cXYYYYYYYYYYYYZZZZZ** where:

- X: 1 character for channel (0: active, 1: reference and 2: catheter).
- Y: 12 characters for time in ms (all integer, no decimal).
- Z: 5 characters for pressure in mmHg (3 integer, 2 decimal).
- All values are padded with 0s at the front if necessary.

Below is an example measurement:

- c000000000001035.430 which corresponds to a measurement from the sensor (active channel) taken 10ms after the start of the connection with a value of 35.43mmHg.

This data format has the advantage of limiting the number of characters used to transmit a single measurement. Because the format is known to both the reader and app in advance, no characters are wasted for decimal points or delimitations which makes data transmission more efficient.

Pressure data is then sent through the Bluetooth development board. The laptop connects to the board via its serial port so a terminal emulator can be used to write commands to the board. TeraTerm is used to communicate with the board and a script written in the Tera Term language (TTL) is used to set the data transmission rate.

Block transmission is available by combining multiple measurements in a single line of the text file. However, this is not recommended as the real-time graph might not look smooth to the user if data points are dynamically added in batches instead of individually.

7.3.2 Device Bluetooth Connection

The Android device must be able to connect to the reader via Bluetooth. This is ensured by the **BluetoothChat** fragment which is the entry point for anything Bluetooth related. The **BluetoothChat** fragment has several responsibilities:

- Checking if the device has Bluetooth support
- Checking if Bluetooth is enabled
- Interacting with the rest of the app

The Bluetooth feature is best explained by going through a cycle of connecting to a device and receiving data:

- 1: The user clicks on a UI element specifying that they want to initiate a Bluetooth connection. This request is passed down to the **BluetoothChat** fragment and down to the **DeviceListActivity** class.
- 2: The **DeviceListActivity** creates an instance of the **BluetoothAdapter** class and calls the **startDiscovery()** method. The device will start scanning. Then, **DeviceListActivity** will use an instance of the **BroadcastReceiver** class to receive a list of available devices. The list of available devices is sent to the **BluetoothChat** fragment which

sends it to the UI.

- 3: After the user has selected the preferred device for connection, the **BluetoothChat** fragment sends the preferred device to the **BluetoothChatService**. Bluetooth works in a client/server model which in this case equates to reader/device. The **BluetoothChat-Service** creates an instance of the **BluetoothServerSocket** class and opens the socket by calling the **listenUsingRfcommWithServiceRecord()** method. **BluetoothChat-Service** then initiates a connection with the preferred device by calling the **accept()** method with the device name.
- 4: **BluetoothChatService** receives or sends data by calling the **getInputStream()** or **getOutputStream()** method on the socket which can then be used to transmit data using the **read()** or **write()** methods. The results from the **read()** method are sent to the **BluetoothChat** fragment for further processing. Finally, **BluetoothChatService** can terminate the connection by calling the **close()** method on the socket.

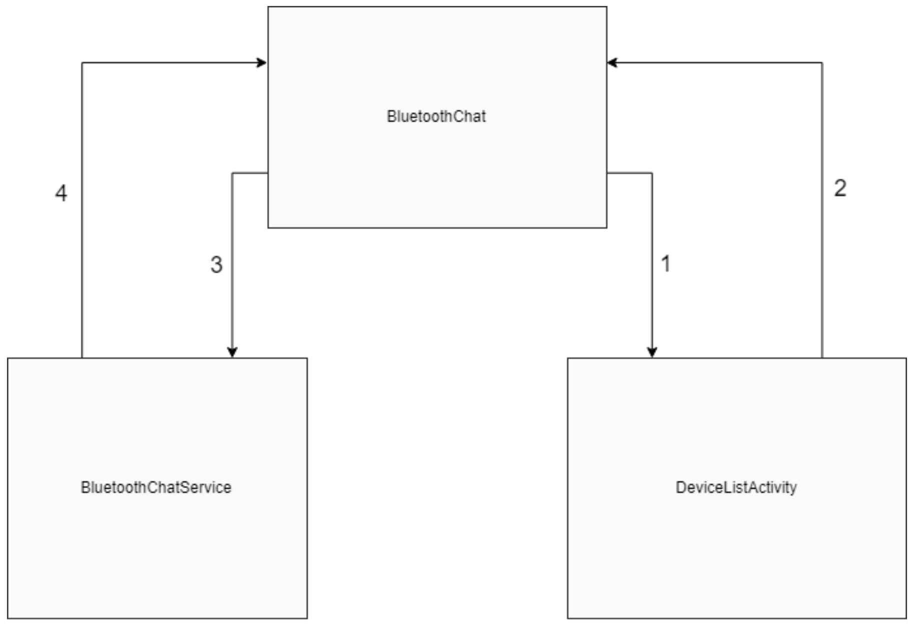

Figure 7.3: Bluetooth Connection Diagram.

Furthermore, the app provides insecure connections using the **listenUsingInsecureRfcommWith-ServiceRecord()** method in the **BluetoothChatService** instead in case the reader requires unauthenticated pairing. Also, the app provides the possibility of making the device discoverable by providing the **BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE** string to an Intent in the **BluetoothChat** fragment in case the device receives files from the reader in the future.

The app ensures correctness of the data by parsing the result of **read()** calls. It checks that strings are of a specified size and separates them into channel, time and pressure value by casting. These values are then checked to ensure they are correct. If everything is correct, time and pressure values are stored in an instance of the the **ChartValue** class.

Finally, the app must be able to send data to reader to set its internal parameters. Below are all the parameters and data ranges:

Parameter	Default	Range	Comments
Active resonator frequency	916.3 MHz	915-920	4 characters. Must be less than reference resonator frequency.
Reference resonator frequency	919.9 MHz	915-920	4 characters.
Active resonator sampling rate	20 ms	5-100	3 characters.
Reference resonator sampling rate	1 s	1s-7,200h	4 characters.
Sweeps	100	1-2047	4 characters.
Samples	64	64-1000	4 characters.
Averages	63	63-999	4 characters. Must be less than number of samples.
Tx power	511	0-1023	4 characters.
Tx time	114	0-1023	4 characters.
Rx delay	22	0-1023	4 characters.

Table 7.1: Reader parameters.

Table 7.1 is used to code the logic in the reader parameter UI elements to check that user inputs conform. The reader parameters are sent in the following format:

arfQQQQrrfRRRRarsSSSrrsTTTTsweUUUUsamVVVVaveWWWWtxpXXXXtxt YYYYrxdZZZZ

Where the strings in lower case represent the parameter keys and strings in upper case the actual values.

All reader parameters are sent at once to simplify reader logic and to be consistent with a sending format.

7.4 Real-Time Visualization

All code related to the real-time chart is contained within the **Graph** fragment. The graph is instantiated in its **onActivityCreated()** method to ensure it is ready to display in the UI. Its data points are those in the **GraphValue** class to which a listener is attached. When a new **GraphValue** instance is created or updated, it calls the **addEntry()** method which adds the point to the graph. This way points are dynamically added to the graph when they are received by the app.

7.5 Signal Processing

App Implementation

In the **Graph** fragment, the **ChartValue** class is instantiated once and updated every time new data is received by the app. Thus, it is necessary to keep a second variable with all measurements up to that point. This variable is the **pap** array consisting of double values as they may be decimal. Time values are not stored as it is possible to know the time duration between 2 **pap** values from the variable **arsr** which is set as the active resonator sampling frequency of the reader and updated every time reader parameters are sent.

Signal processing is not triggered until the **pap** buffer reaches a certain length which in this case has been set to 10 seconds to obtain several beats. Thus, signal processing begins when the condition $10 == \frac{length(pap)}{\frac{arsr}{1000}}$ as **arsr** is in ms. At this point, an **AsyncTask** is created with the **pap** array as an argument.

In Android, an **AsyncTask** is a class that allows asynchronous operations in the background whose results are published on the UI thread. So far, every piece of code detailed is set to run in the UI thread which is the main thread of the app. However, this should be reserved for actions which are directly visible to the user. Signal processing is not one of such tasks so the operation runs in the background by placing related code in the **doInBackground()** method of the **AsyncTask** and passing results back to the UI thread through the **onPostExecute()** method of the **AsyncTask**.

Below are the operations performed in **doInBackground()**:

- Feature Extraction
- Abnormality Detection

Below are the operations performed in **onPostExecute()**:

- Update UI with feature values
- Store features and abnormalities in database

After the **AsyncTask** has been called, the **pap** array is cleared so new values can be stored for the next cycle.

Algorithm Porting

The chosen feature extraction and abnormality detection algorithms had to be ported to Java from Matlab following chapter 6. One method to port code is to rewrite everything in Java. However, Matlab offers C/C++ code generation through Matlab Coder [55]. This was used to generate C++ code which was then integrated with Java code using the Java Native Interface. This was done by wrapping the C++ functions in a Java method call which could then be used in the app [56].

Porting Matlab code to C++ produced an obvious limitation as Matlab Coder required an array size for inputs and outputs which presumably makes the porting easier by using arrays instead of vectors. This meant that arrays could only be of a prespecified size which would not work with feature extraction as individual beat arrays differ in size. Thus, all algorithms were rewritten in Java including helper functions such as Diff and Convolution which are used in the WABP algorithm.

Figure 7.4: Code Porting from Matlab to Java.

7.6 Database

Logging

Data storage is performed using SQLite which is one of the native libraries supported by the Android software stack.

Below is the current database schema which includes 2 tables for the storage of pressure measurements and feature extraction/abnormality detection results:

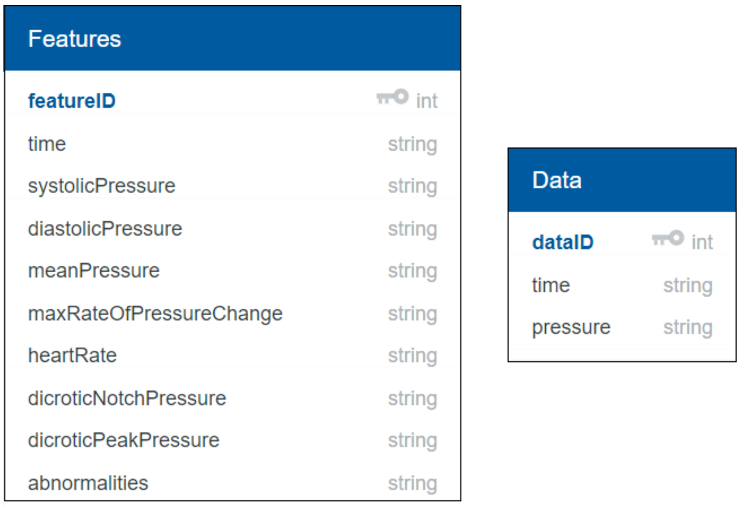

Figure 7.5: Database schema.

It is noted that the time string is stored in ISO8601 format per Android guidelines: **yyyy-MM-dd HH:mm:ss.SSS**.

Features and abnormalities are stored in a different table as these are only created once every 10 seconds. This way, an entry is created only when there are features and abnormalities to log as opposed to having everything in a single table.

Exporting

An additional feature added during the development of the app was the ability to export the database to Excel. This can be performed through an **AsyncTask** as the feature does not support any UI element.

In the **doInBackground()** call of the **AsyncTask**, 2 CSV files are created: one to store the contents of the Data table and another for the Features table. The formatting of the data is handled by the **CSVWriter** class.

7.7 Security and Privacy

Special care must be given when handling user data especially for a mobile application in the healthcare space. Throughout development, mobile app privacy guidelines [57] were followed to ensure that the app respected user privacy and was secure. Thus, the app includes a disclaimer in the **About** page detailing which user data it is using (in this case, reader measurements) and how it is using it (in this case, storing values in the database).

Furthermore, the app **sharedPreferences** (where any data that has to be persisted between sessions is stored) are stored in private mode such that only this specific application can access it.

Chapter 8

Interface Design and Implementation

This chapter pertains to the graphical design of the app and overall UX. It presents design and implementation considerations from the point of view of the user.

8.1 Initial Mock

Before starting app development, a mock was developed to illustrate the main features of the app. Below is the minimum viable product mock which defines the core features to be developed:

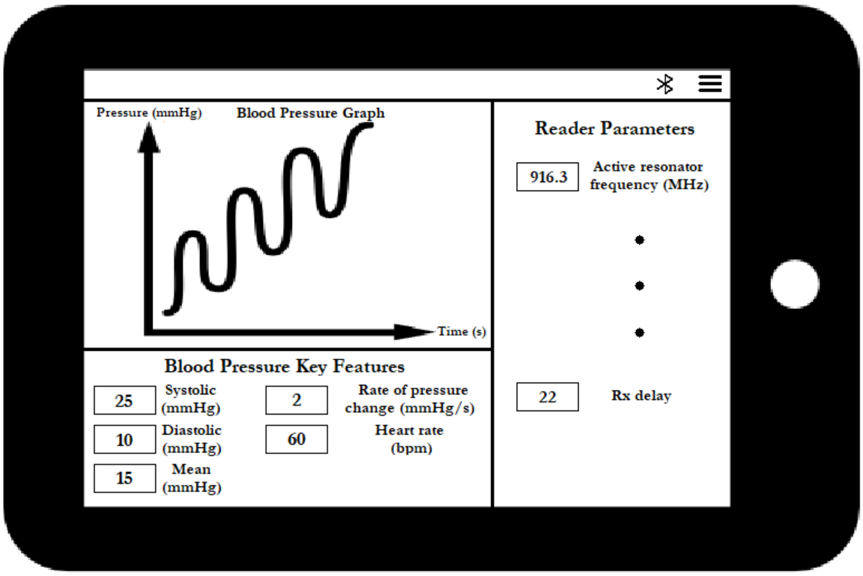

Figure 8.1: Minimum viable product mock.

As seen above, the core of the app would consist of a single activity including a line graph, editable text views to set reader parameters, text views for key pressure features and a button to initiate Bluetooth connection between the device and the reader.

8.2 Graph Page

The graph page is the main entry point of the app for the user. Most screen real estate is dedicated to the graph itself as it is important to have ample space to visualize waveforms.

At the bottom of the page are the settable reader parameters in the form of **EditText** views. These can be sent to the reader by pressing the send button in the form of a **FloatingAction-Button**. **EditText** views were chosen as a common UI element that prompts the user to take action as seen by the red highlight in the active resonator frequency field. Also, the red send button stands out from the page due to its importance.

In a larger font size, the feature values are placed directly below the graph for convenience in viewing both the graph and feature values. As they are more important that the reader parameters, the features are centered in the middle of the page and are larger in size.

Figure 8.2: Graph Page.

Figure 8.3 shows the settings menu options which include Bluetooth connection options discussed in chapter 7 as in figure 8.4, the option to clear the chart and to export data.

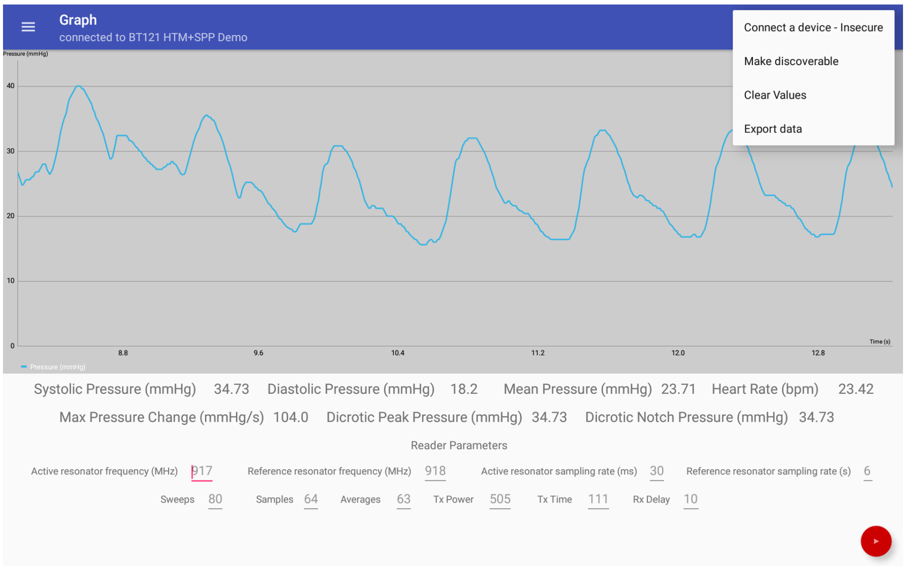

Figure 8.3: Graph Plot and Menu.

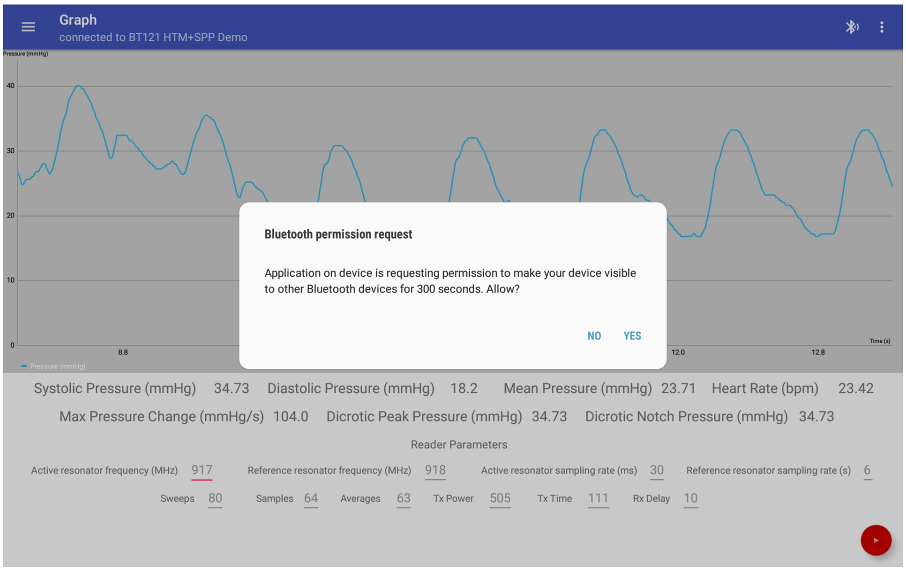

Figure 8.4: Make Discoverable Dialog.

Users can also initiate a Bluetooth connection by clicking on the Bluetooth icon in the action bar next to the settings icon. As initiating a Bluetooth is one of the main actions for users, the icon is visible and clickable from the main page which opens a window to show available devices as in figure 8.5. When a Bluetooth connection is initiated, the reader first sends its internal parameters to the app which updates its view to match the reader values.

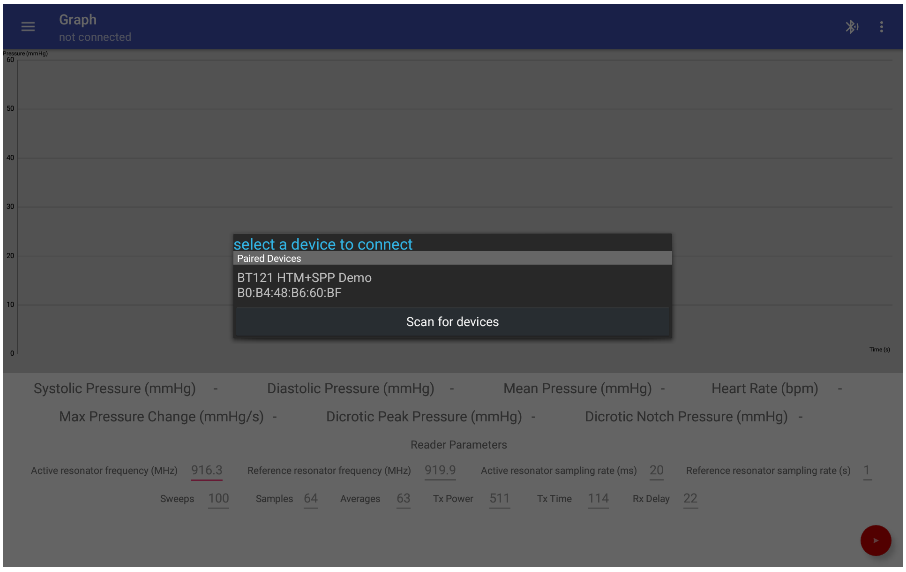

Figure 8.5: Bluetooth Connection Window.

It is important to provide visual feedback when a user performs an operation. Figure 8.6 shows the outcome of trying to send erroneous parameters to the reader. The app displays both a toast to show that the operation failed and an error message next to the visual element that caused the error. This way, users can quickly identify the error and correct it.

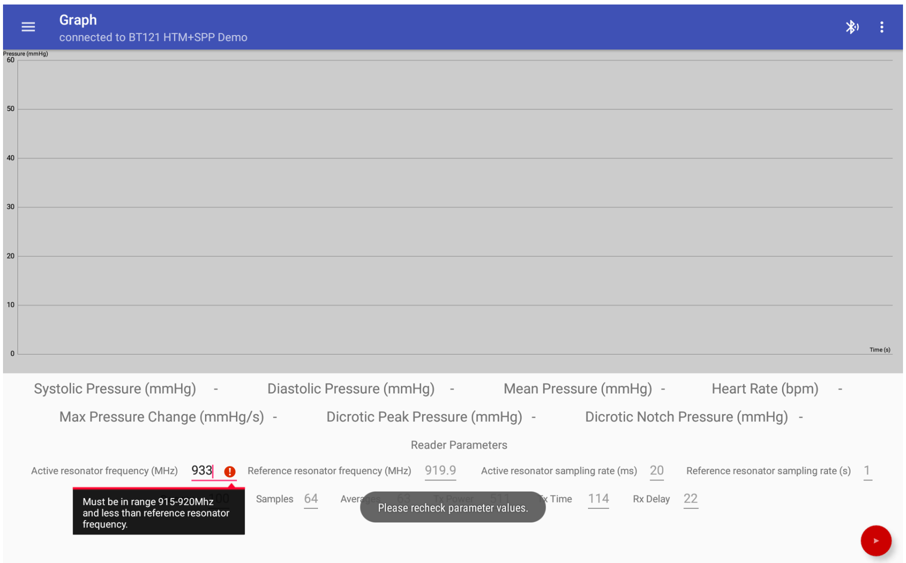

Figure 8.6: Error Message.

Finally, users can export the database through the option in the settings menu. This opens a progress bar that informs the user of the progress of the operation. This is useful for long operations as users may want to know how long is left until the end of said operation.

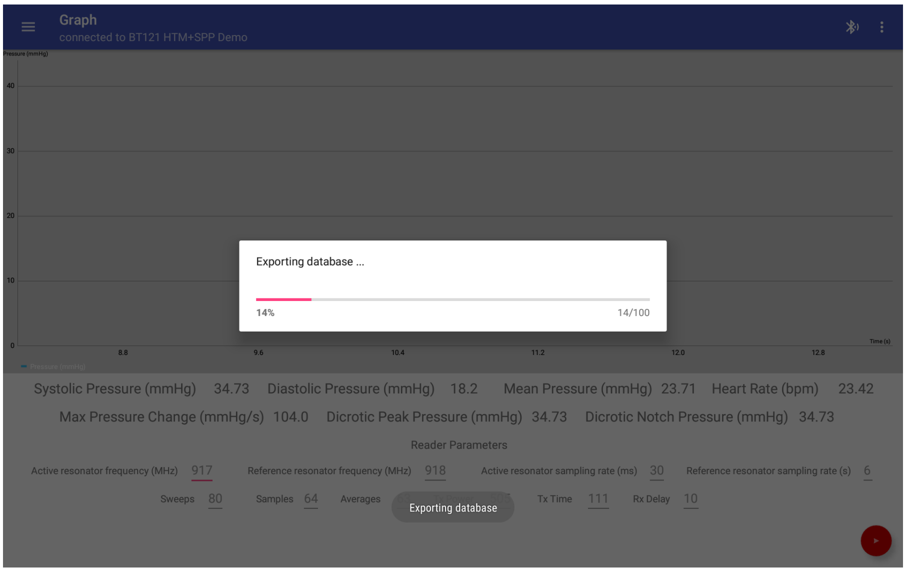

Figure 8.7: Export Progress Bar.

8.3 Menu

The menu is the part of the app that links different pages together so it is important to provide an intuitive menu. The main alternatives in Android are tabbed menus which are displayed directly on the action bar or underneath and navigation drawers which is a hidden menu accessible through a hamburger icon. Figure 8.8 shows an example containing both types of menus.

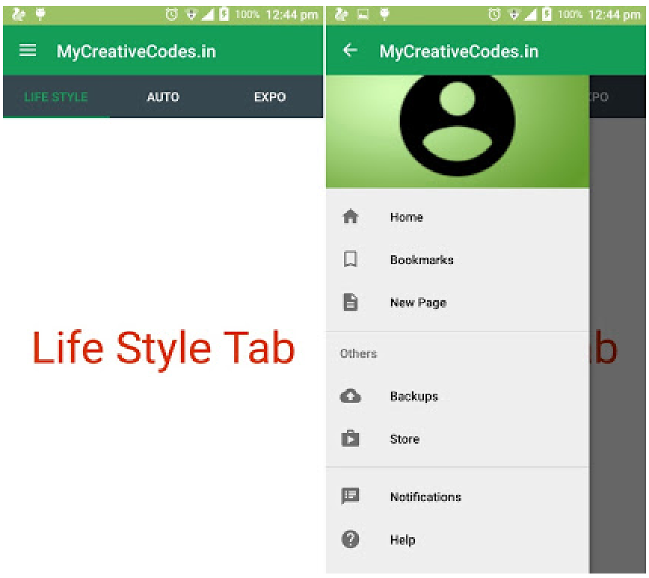

Figure 8.8: Navigation Drawer and Tabbed Menu Hybrid App.

For a scalable application, navigation drawers are preferred as the number of items can be increased whereas the number of tabbed items is limited by the space on the action bar. Thus, a navigation drawer was chosen as seen in figure 8.9. The navigation drawer has the added property of being able to display user information as seen in the space above the menu items. Furthermore, menu items can be logically separated as in figure 8.8.

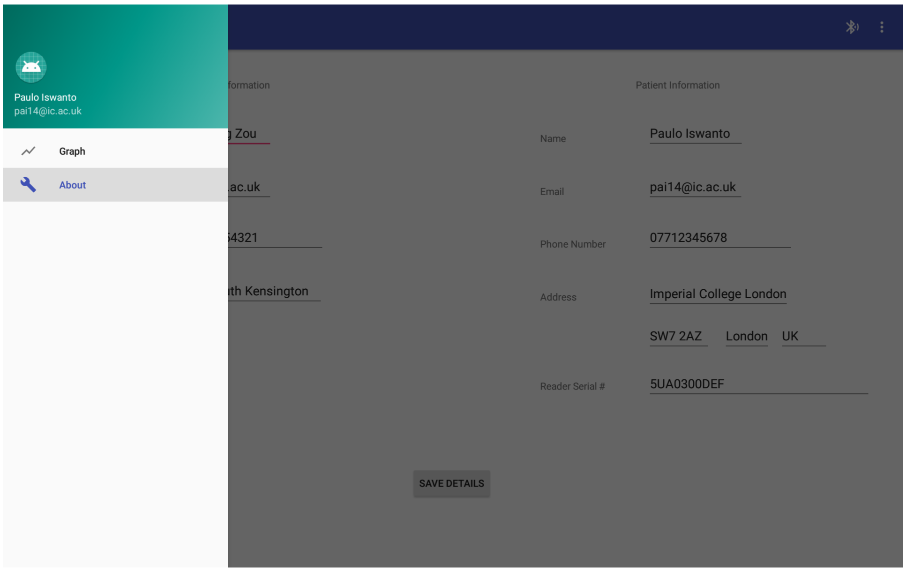

Figure 8.9: Navigation Drawer.

8.4 About Page

The about page in figure 8.10 displays all generic information related to patients and their clinicians. The information can be used to send information such as emailing a clinician when the heart rate is critically low. The page also includes a disclaimer due to user privacy concerns. Finally, patient details in the page are linked to the navigation drawer. Thus, an update of these details is reflected in the navigation drawer as seen in figure 8.11.

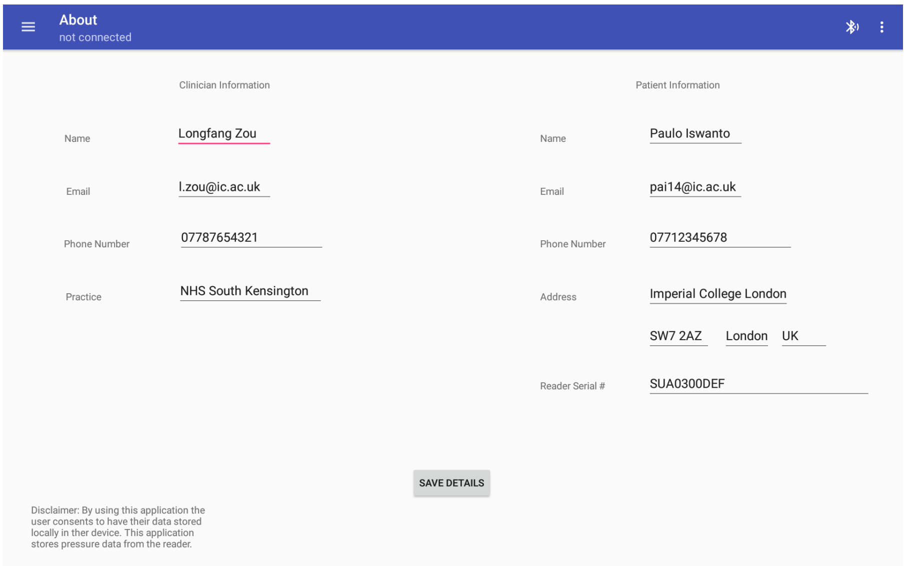

Figure 8.10: About Page.

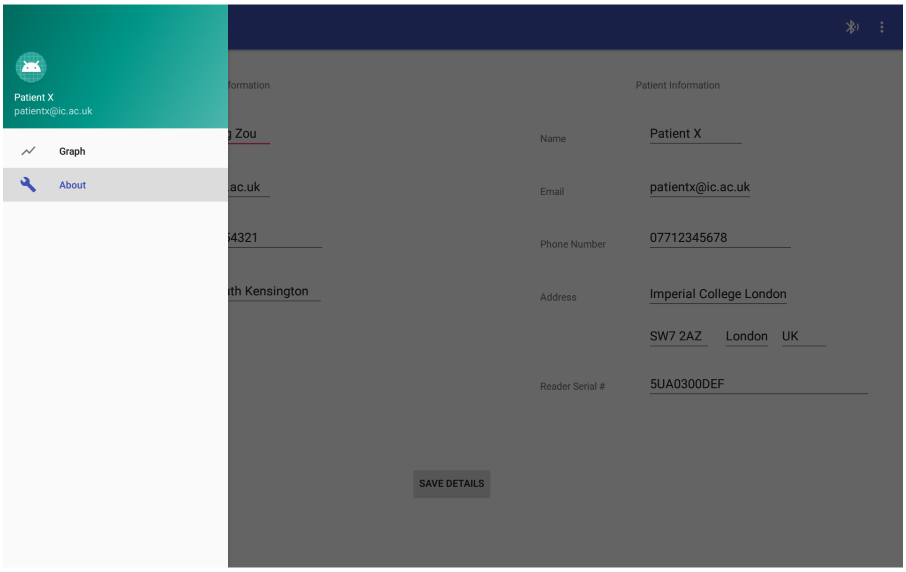

Figure 8.11: Navigation Drawer Update.

Chapter 9

Testing

9.1 Motivation and Objectives

Testing an Android app is part of the quality assurance (QA) process that ensures that users will get a pleasant user experience. Thus, it is important to have clearly defined tests against which the app will be checked to make sure that it conforms to specifications.

The testing pyramid below illustrates the three major types of tests:

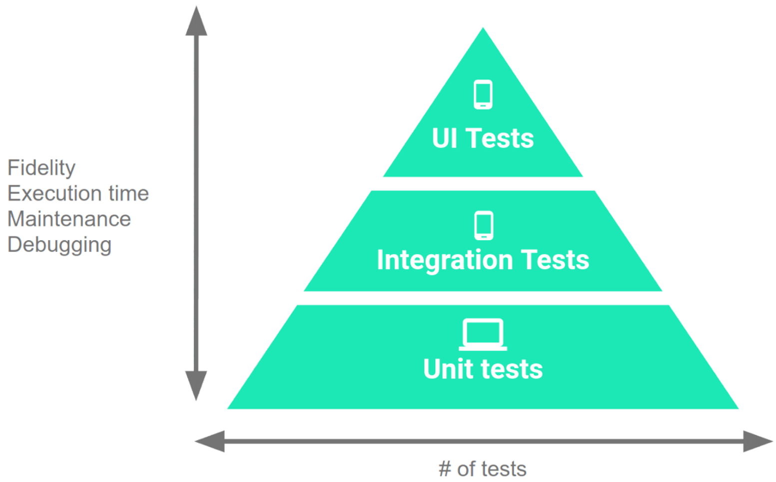

Figure 9.1: The testing pyramid [58].

Small unit tests should be used to test individual components whilst mocking out any interface.

Integration tests involve several components and check that their interfaces all well-defined and work.

UI tests are the most encompassing tests featuring end-to-end testing by performing a UI workflow that includes typical user tasks.

A well-tested app contains a combination of the aforementioned types of tests. More tests are needed going down the pyramid, but complexity decreases which allows for compact and succinct code.

9.2 Testing Overview

Before any other testing, the app was tested against core Android app qualities as defined by Google [59]. These tests set the minimum bar for any Android application.

Below are the different tests that the Android app will have to conform to in order to ensure a good user experience:

9.2.1 Unit tests

- **GUI**: UI components should work. This involves checking that clicking buttons, text views and menus does not break the application.
- **Logging**: The app should be able to store pressure data. This can be verified by manually logging data and checking if it was saved to the database.

9.2.2 Integration tests

- **Data Communication**: The device should be able to connect to the reader. Thus, the app should be able to continuously receive pressure data at a rate of a sample every 20ms for at least 5 minutes.

9.2.3 UI tests

- **Usability**: Users must find the app intuitive and useful.

- **Performance**: As the most telling metric of performance, frame rate will be used to determine whether if the app conforms to specification. The app should run at a minimum median frame rate of 20fps at full capacity when visualizing pressure data. Furthermore, its FPS stability as defined in chapter 7 should be above 50%.
- **Signal Processing**: The feature extraction algorithm should be validated with expertly annotated measurements of key pressure waveform features. Estimated values should not deviate from annotated values by more than a pre-specified threshold.

9.3 Unit Tests

9.3.1 GUI

All UI controls were manually tested by clicking them and seeing the corresponding output in Logcat through Android Studio by installing the app in debug mode.

9.3.2 Logging

The ability to log data was tested by connecting the app to the reader and displaying measurements until at least one set of feature values had been displayed. The database was then exported and inspected in Excel to check for new entries.

9.4 Integration Tests

9.4.1 Data Communication

The device was connected to the reader in both secure and insecure modes. Data was displayed continuously for 5 minutes during which common tasks were performed such as setting the reader parameters.

9.5 UI Tests

UI tests are the most encompassing tests which can give performance insight about the app. As such, results for this part are discussed in the following chapter.

9.5.1 Usability

Volunteers were shown a demo of the application which included all features as described in this chapter. The app was rated on the Mobile Application Rating Scale (MARS) [60], a rating scale directly designed to evaluate mobile health apps.

9.5.2 Performance

The app performance was tested using GameBench in strict mode. Over a 2 minute period, data was continuously visualized and reader parameters sent. Then, all settings items were clicked and the about page modified which constitutes a full user session.

9.5.3 Signal Processing

Signal processing algorithms were validated using the PAP dataset in chapter 6.

Chapter 10

Results

10.1 Usability

A group of $n = 19$ volunteers where shown a demo of the app as detailed in chapter 9. They were then instructed to rate the app using the Mobile Application Rating Scale in appendix B.

Dimension	**Score**
Engagement	3.2
Functionality	4.1
Aesthetics	2.3
Information	3.2
Overall	3.2
Subjective	2.6
App-Specific	4.3

Table 10.1: Rating Results.

Table 10.1 shows that users rated the app highly in functionality which is expected as it was developed with performance in mind for a very specific goal (real-time visualization and signal processing). The aesthetics score is the lowest which can be explained by the lack of high quality graphics in modern apps such as pictures or special fonts as seen in Facebook or Snapchat. Most answers were centered around 2-4 scores with a few outliers. The question that received the most 1s was the very first one: Is the app fun? This is expected as the app is designed to be used to treat illnesses so there is no gamification or such. More interestingly, users have a high score to the app-specific section which is concerned with the perceived impact of the app and actual changes in target behavior. Additionally, one of the volunteers who happened to have been diagnosed with pulmonary hypertension commented on the usefulness of the app and how it takes some worries off by delegating health monitoring.

The results from the survey have shown the relevance of the app and its potential application is post-hospitalization treatment. Also, it highlighted areas for improvement such as aesthetics.

10.2 Performance

The performance of the app was evaluated using Gamebench in strict mode as in chapter 5. During the 2 minute session, data was visualized continuously and all actions performed such as setting reader parameters or exporting the database. Figure 10.1 shows the performance during the session.

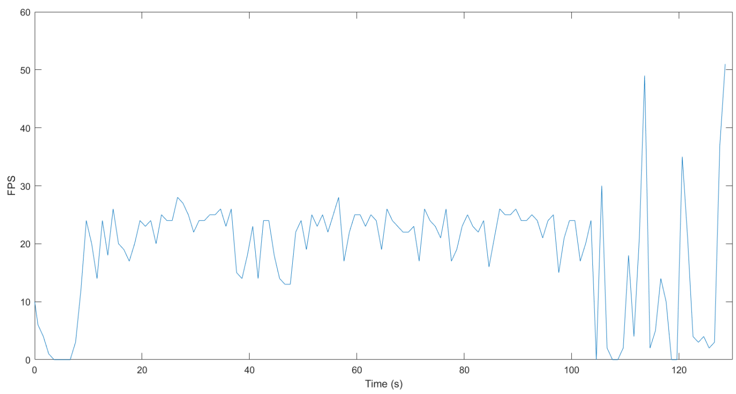

Figure 10.1: App performance during a typical session.

Median FPS (FPS)	FPS Stability (%)	Stability Index
22	63	5.5

Table 10.2: Performance Statistics.

Table 10.2 shows that the app conforms to targets set in chapter 3 as the median FPS is bigger than 20fps and the FPS Stability is larger than 50%. However, the stability index is quite high meaning that the app suffered from frame rate jumps. This happened around the time the database was exported and concurrently the graph was displaying data. One possible solution would be to disable exporting until no data is being visualized in real-time.

10.3 Signal Processing

As found in chapter 6, the average RMSE for systolic, diastolic and mean pressure estimates was smaller than 5 and less than 20 for heart rate estimates. The other feature errors cannot be analyzed directly as there were no true feature values for them in the PAP dataset.

Running the feature extraction algorithm through the app yields reasonable values for all parameters and abnormality detection was shown to be useful in detecting irregular beats for patients with heart related illnesses.

Chapter 11

Evaluation

In the end, all features were implemented for the Android app: two-way data communication via Bluetooth, real-time visualization, feature extraction and abnormality detection. Additionally, data can be exported for further processing as exporting was implemented as an extra feature. This Android app differs from most mHealth apps in the Android store as it is performance oriented and intended for medical use whereas mHealth apps are more about raising awareness and providing generic medical advice. The app is so far the first to implement detailed feature extraction and abnormality detection algorithms for use in Android devices. Apps exist to interface with hemodynamic sensors such as myCardioMEMS [61]. However, they all have restricted access and act more as an interface to send data to clinicians instead of visualizing and analyzing hemodynamic data.

Unfortunately, the reader was not ready during the timeline of the project so Bluetooth communication was scoped to interacting with the reader simulation.

The biggest shortcoming of the project is not having a clinical trial to evaluate relevance of the app. The app is intended for medical use, yet there is no concrete evidence of usefulness amongst its target population.

Concerning the app itself, SciChart would have been the preferred graphing library as it significantly outperformed every open-source graphing library for Android. As it is a proprietary product, it was not possible to integrate it with the app. Performance was evaluated on a single Android device, yet the Android ecosystem is composed of multiple devices with different amounts of processing resources. For a streamlined user experience, it would have been better to implement the app in iOS for predictable performance due to strict regulations of the Apple hardware.

Also, the signal processing algorithms were not fully validated due to missing feature data from the PAP dataset. However, no physiological signal database was found during the timeline of the project that included the missing features. It would be possible although time consuming to expertly annotate records.

Finally, abnormality detection was implemented only for short-term feedback on a beat-by-beat basis. This hampers the usefulness of flagging abnormalities as most heart related diseases manifest over a period of time and symptoms are not discernible in the instant. As explained in chapter 6, an online resource would be most appropriate to handle long-term abnormality detection.

Chapter 12

Conclusions and Further Work

12.1 Conclusions

This report outlined the development of an Android app for real-time visualization of blood pressure waveforms along with feature extraction and abnormality detection through Bluetooth connection with a reader for a novel sensor designed at Imperial College London. Multiple feature extraction and abnormality detection algorithms were discussed and implemented both in Matlab and Java. For their validation, a framework was designed to interact with the MIMIC-III database from Matlab and a PAP dataset created containing both PAP waveforms and features. Results show that the app is useful to patients and clinicians and may be used as a potential monitoring tool for hemodynamic data.

12.2 Future Work

Future work in the immediate future includes integrating the reader with the app as this link has not been tested due to the reader being in development.

Potential avenues of research include more complex feature extraction and abnormality detection algorithms along with a proper dataset to validate all estimated features. The app should be tested in a clinical trial before even being considered for medical use.

Chapter 13

User Guide

This chapter details the necessary steps to continue the development of the project. The use of the Android app is detailed in chapter 8 and the required hardware in chapter 3.

13.1 App Development

In order to continue app development, the following are required:

- Install Android Studio along with the simpleUMLCE plugin and the Instant Run feature disabled.
- Install Samsung Kies
- Enable USB debugging on the Android device.

13.2 Reader

In order to simulate the reader using the Bluetooth Development Kit, the following are required:

- Install TeratTerm

13.3 Signal Processing

In order to continue feature extraction and abnormality detection research, the following are required:

- Request access and install the MIMIC-III database.
- Install Matlab along with the WFDB toolbox and add PostgreSQL support.

Bibliography

[1] Paul A Heidenreich, Nancy Albert, Larry A Allen, David A Bluemke, Javed Butler, Gregg Fonarow, John S Ikonomidis, Olga Khavjou, Marvin A Konstam, Thomas Maddox, Graham Nichol, Michael Pham, Ileana Pina, and Justin Trogdon. Forecasting the impact of heart failure in the united states: A policy statement from the american heart association. In *Circulation: Heart Failure*, volume 6, pages 606–619. 05 2013.

[2] Christopher M. O'Connor. High heart failure readmission rates: Is it the health system's fault? In *Journal of the American College of Cardiology*, volume 5. 2017.

[3] Andrija Vidic, John Chibnall, and Paul Hauptman. Heart failure is a major contributor to hospital readmission penalties. In *Journal of cardiac failure*, volume 21. 12 2014.

[4] Sheila M Manemann, Alanna Chamberlain, Cynthia M Boyd, Susan A Weston, Jill Killian, Cynthia L Leibson, Andrea Cheville, Jennifer St Sauver, Shannon Dunlay, Ruoxiang Jiang, and Véronique L Roger. Skilled nursing facility use and hospitalizations in heart failure: A community linkage study. In *Mayo Clinic proceedings*, volume 92. 03 2017.

[5] William T. Abraham and Leor Perl. Implantable hemodynamic monitoring for heart failure patients. In *Journal of the American College of Cardiology*, volume 70, pages 389–398. 07 2017.

[6] Friedrich Koehler, Sebastian Winkler, Michael Schieber, Udo Sechtem, Karl Stangl, Michael Böhm, Herbert Boll, Gert Baumann, Marcus Honold, Kerstin Koehler, Götz Gelbrich, Bridget-Anne Kirwan, and Stefan Anker. Impact of remote telemedical management on mortality and hospitalizations in ambulatory patients with chronic heart failure the telemedical interventional monitoring in heart failure study. In *Circulation*, volume 123, pages 1873–80. 03 2011.

[7] Lee Nguyen and Pierre Squara. Non-invasive monitoring of cardiac output in critical care medicine. In *Frontiers in Medicine*, volume 4. 11 2017.

[8] Olive Murphy, Mohammad Bahmanyar, Alessandro Borghi, Chris Mcleod, Manoraj Navaratnarajah, Magdi H Yacoub, and Christofer Toumazou. Continuous in vivo blood pressure measurements using a fully implantable wireless saw sensor. In *Biomedical microdevices*, volume 15. 04 2013.

[9] Edith Applegate. *The anatomy and physiology learning system.* Saunders/Elsevier, St. Louis, Mo, 2011. ISBN 1437703933.

[10] Mariana Ruiz Villarreal. Human heart shown during systole and diastole, 2008. URL `https://commons.wikimedia.org/wiki/File:Human_healthy_pumping_heart_en.svg`.

[11] Eugene Braunwald. *Braunwald's heart disease : a textbook of cardiovascular medicine.* Saunders, Philadelphia, 2012. ISBN 1437703984.

[12] John Hall. *Guyton and Hall textbook of medical physiology.* Saunders/Elsevier, Philadelphia, Pa, 2011. ISBN 1416045740.

[13] Hanna Blackledge, J Tomlinson, and Iain Squire. Prognosis for patients newly admitted to hospital with heart failure: Survival trends in 12 220 index admissions in leicestershire 1993-2001. In *Heart (British Cardiac Society)*, volume 89, pages 615–20. 06 2003.

[14] Willem J. Remme, John J.V. McMurray, Bernhard Rauch, Faiez Zannad, Karen Keukelaar, Alain Cohen-Solal, Jose Lopez-Sendon, F.D. Richard Hobbs, Diederick E. Grobbee, Alessandro Boccanelli, Charles Cline, Cezar Macarie, Rainer Dietz, and Witold Ruzyllo. Public awareness of heart failure in europe: first results from shape. *European Heart Journal*, 26(22):2413–2421, 2005. doi: 10.1093/eurheartj/ehi447. URL `+http://dx.doi.org/10.1093/eurheartj/ehi447`.

[15] F D R Hobbs, J Doust, J Mant, and M R Cowie. Diagnosis of heart failure in primary care. *Heart*, 96(21):1773–1777, 2010. ISSN 1355-6037. doi: 10.1136/hrt.2007.139402. URL `http://heart.bmj.com/content/96/21/1773`.

[16] Martin Cowie, Stefan Anker, John G. F. Cleland, Gary Felker, Gerasimos Filippatos, Tiny Jaarsma, Patrick Jourdain, Eve Knight, Barry Massie, Piotr Ponikowski, and José López-Sendón. Improving care for patients with acute heart failure: Before, during and after hospitalization. In *ESC Heart Failure*, volume 1. 01 2015.

[17] Harlan M Krumholz, Peter M Currie, Barbara Riegel, Christopher Phillips, Eric D Peterson, Renee Smith, Clyde Yancy, and David Faxon. A taxonomy for disease management a scientific statement from the american heart association disease management taxonomy writing group. In *Circulation*, volume 114, pages 1432–45. 10 2006.

[18] Barbara Riegel, Debra Moser, Stefan Anker, Lawrence J Appel, Sandra Dunbar, Kathleen Grady, Michelle Z Gurvitz, Edward Havranek, Christopher Lee, Joann Lindenfeld, Pamela N Peterson, Susan J Pressler, Douglas Schocken, and David Whellan. State of the science promoting self-care in persons with heart failure a scientific statement from the american heart association. In *Circulation*, volume 120, pages 1141–63. 09 2009.

[19] What works in chronic care management: The case of heart failure. volume 28, pages 179–189. 2009. doi: 10.1377/hlthaff.28.1.179. URL `https://www.healthaffairs.org/doi/abs/10.1377/hlthaff.28.1.179`. PMID: 19124869.

[20] Trupp, Brian Hiestand, and Chadwick D. Miller Elsie Selby W. Frank Peacock Gregg C. Fonarow Gail Dodge, Christy O. Hayes. Patient education. volume 7, pages 116–121. Lippincott Williams and Wilkins, 6 2008.

[21] Andreas Michalsen, G König, and W Thimme. Preventable causative factors leading to hospital admission with decompensated heart failure. In *Heart (British Cardiac Society)*, volume 80, pages 437–41. 12 1998.

[22] Catherine Ecock Connelly. An empirical study of a model of self-care in chronic illness. In *Clinical nurse specialist CNS*, volume 7, pages 247–53. 10 1993.

[23] Jean Rockwell and Barbara Riegel. Predictors of self car in persons with heart failure. 30: 18–25, 02 2001.

[24] Judith Hibbard, Jean Stockard, Eldon R Mahoney, and Martin Tusler. Development of the patient activation measure (pam): Conceptualizing and measuring activation in patients and consumers. 39:1005–26, 09 2004.

[25] David Mosen, Julie Schmittdiel, Judith Hibbard, David Sobel, Carol Remmers, and Jim Bellows. Is patient activation associated with outcomes of care for adults with chronic conditions? 30:21–9, 03 2007.

[26] Adroitent. Impact of telemedicine, 2015. URL `http://adroitent.com/blog/wp-content/uploads/2016/11/Adroitent-Info-design-02.png`.

[27] Farhad Fatehi and Richard Wootton. Telemedicine, telehealth or e-health? a bibliometric analysis of the trends in the use of these terms. *Journal of Telemedicine and Telecare*, 18 (8):460–464, 2012. doi: 10.1258/jtt.2012.gth108. URL `https://doi.org/10.1258/jtt.2012.gth108`. PMID: 23209265.

[28] Rashid Bashshur, Gary Shannon, Brian R Smith, Dale Alverson, Nina Antoniotti, William Barsan, Noura Bashshur, Edward M Brown, Molly Coye, Charles Doarn, Stewart Ferguson, Jim Grigsby, Elizabeth Krupinski, Joseph Kvedar, Jonathan Linkous, Ronald C Merrell, Thomas Nesbitt, Ronald Poropatich, Karen S Rheuban, and Peter Yellowlees. The empirical foundations of telemedicine interventions for chronic disease management. 20, 06 2014.

[29] Sarwat I. Chaudhry, Jennifer A. Mattera, Jeptha P. Curtis, John A. Spertus, Jeph Herrin, Zhenqiu Lin, Christopher O. Phillips, Beth V. Hodshon, Lawton S. Cooper, and Harlan M. Krumholz. Telemonitoring in patients with heart failure. *New England Journal of Medicine*, 363(24):2301–2309, 2010. doi: 10.1056/NEJMoa1010029. URL `http://dx.doi.org/10.1056/NEJMoa1010029`. PMID: 21080835.

[30] M. K. Ong, P. S. Romano, S. Edgington, H. U. Aronow, A. D. Auerbach, J. T. Black, T. De Marco, J. J. Escarce, L. S. Evangelista, B. Hanna, T. G. Ganiats, B. H. Greenberg, S. Greenfield, S. H. Kaplan, A. Kimchi, H. Liu, D. Lombardo, C. M. Mangione, B. Sadeghi, B. Sadeghi, M. Sarrafzadeh, K. Tong, and G. C. Fonarow. Effectiveness of Remote Patient

Monitoring After Discharge of Hospitalized Patients With Heart Failure: The Better Effectiveness After Transition – Heart Failure (BEAT-HF) Randomized Clinical Trial. *JAMA Intern Med*, 176(3):310–318, Mar 2016.

[31] M. R. Zile, T. D. Bennett, M. St John Sutton, Y. K. Cho, P. B. Adamson, M. F. Aaron, J. M. Aranda, W. T. Abraham, F. W. Smart, L. W. Stevenson, F. J. Kueffer, and R. C. Bourge. Transition from chronic compensated to acute decompensated heart failure: pathophysiological insights obtained from continuous monitoring of intracardiac pressures. *Circulation*, 118(14):1433–1441, Sep 2008.

[32] "Robert C. Bourge, William T. Abraham, Philip B. Adamson, Mark F. Aaron, Juan M. Aranda, Anthony Magalski, Michael R. Zile, Andrew L. Smith, Frank W. Smart, Mark A. O'Shaughnessy, Mariell L. Jessup, Brandon Sparks, David L. Naftel, and Lynne Warner Stevenson". Randomized controlled trial of an implantable continuous hemodynamic monitor in patients with advanced heart failure: The compass-hf study. *Journal of the American College of Cardiology*, 51(11):1073 – 1079, 2008. ISSN 0735-1097. doi: https://doi.org/10.1016/j.jacc.2007.10.061. URL "http://www.sciencedirect.com/science/article/pii/S0735109708001356".

[33] SJM. Cardiomems hf system, 2016. URL https://www.sjm.com/~/media/galaxy/hcp/featured-products/hf/cardiomems-hf-system/cardiomems-transmission.jpg?la=en.

[34] Gartner. Global mobile os market share in sales to end users from 1st quarter 2009 to 2nd quarter 2017, 2017. URL https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/.

[35] Inc. Bluetooth SIG. Legacy core specifications, 2017. URL https://www.bluetooth.com/specifications/bluetooth-core-specification/legacy-specifications.

[36] Google LLC. Bluetooth low energy, 2017. URL https://developer.android.com/guide/topics/connectivity/bluetooth-le.html.

[37] Donald Fisher. *Handbook of driving simulation for engineering, medicine, and psychology.* Taylor & Francis Group, Boca Raton, 2011. ISBN 1420061003.

[38] Silicon Laboratories. Bluegiga dkbt bluetooth smart ready development kit, 2018. URL `https://www.silabs.com/products/development-tools/wireless/bluetooth/dkbt-bluetooth-smart-development-kit`.

[39] Android Arsenal. Android developer portal with tools, libraries, and apps, 2018. URL `https://android-arsenal.com/tag/40?sort=created`.

[40] SciChart. Android chart performance comparison, 2016. URL `https://www.scichart.com/android-chart-performance-comparison/`.

[41] Timo E Strandberg and Kaisu Pitkala. What is the most important component of blood pressure: Systolic, diastolic or pulse pressure? In *Current opinion in nephrology and hypertension*, volume 12, pages 293–7. 06 2003.

[42] Marilyn Prasun and Nancy Albert. The importance of heart rate in heart failure and reduced ejection fraction. In *The Journal of Cardiovascular Nursing*, page 1. 04 2018.

[43] Philippe Morimont, Bernard Lambermont, Thomas Desaive, Nathalie Janssen, James Chase, and Vi D'Orio. Arterial dp/dtmax accurately reflects left ventricular contractility during shock when adequate vascular filling is achieved. In *BMC cardiovascular disorders*, volume 12, page 13. 03 2012.

[44] P Shi, Sijung Hu, Y Zhu, J Zheng, Yihong Qiu, and Peck Cheang. Insight into the dicrotic notch in photoplethysmographic pulses from the finger tip of young adults. 33:628–33, 11 2009.

[45] PhysioNet. Physionet, 2018. URL `https://www.physionet.org/`.

[46] Alistair Edward William Johnson, Tom Joseph Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Edward Moody, Peter Szolovits, Leo Anthony G. Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. In *Scientific Data*, volume 3, page 160035. 05 2016.

[47] Jan A. Staessen, Robert Fagard, Lutgarde Thijs, and Antoon Amery. A consensus view on the technique of ambulatory blood pressure monitoring. *Hypertension*, 26(6):912–918, 1995. ISSN 0194-911X. doi: 10.1161/01.HYP.26.6.912. URL http://hyper.ahajournals.org/content/26/6/912.

[48] Wei Zong, Thomas Heldt, G.B. Moody, and R.G. Mark. An open-source algorithm to detect onset of arterial blood pressure pulses. In *Computers in Cardiology*, volume 30, pages 259 – 262. 10 2003. ISBN 0-7803-8170-X.

[49] G A Ewy, J C Rios, and Frank Marcus. The dicrotic arterial pulse. In *Circulation*, volume 39, pages 655–61. 06 1969.

[50] Felix Scholkmann, Jens Boss, and Martin Wolf. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. In *Algorithms*, volume 5, pages 588–603. 11 2012.

[51] Gerald Smetana. Rethinking "abnormal" blood pressure: What is the value? In *Journal of general internal medicine*, volume 26, pages 678–80. 07 2011.

[52] J.X. Sun, A.T. Reisner, and R.G. Mark. A signal abnormality index for arterial blood pressure waveforms. In *Comput. Cardiol*, volume 33, pages 13 – 16. 10 2006. ISBN 978-1-4244-2532-7.

[53] American College of Cardiology. Guidelines for diagnosis and treatment of pulmonary hypertension, 2015. URL http://www.acc.org/latest-in-cardiology/ten-points-to-remember/2015/09/15/15/19/2015-esc-ers-guidelines-for-the-diagnosis-and-treatment-of-ph.

[54] MS Consulting. Pm-agile, 2017. URL https://www.mandsconsulting.com/wp-content/uploads/PM-Agile.jpg.

[55] Inc. The MathWorks. Matlab to iphone and android made easy, 2018. URL https://uk.mathworks.com/videos/matlab-to-iphone-and-android-made-easy-107779.html.

[56] Oracle. Jni functions, 2018. URL https://docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/functions.html.

[57] Information Commissioner's Office. Privacy inmobile apps, 2018. URL https://ico.org.uk/media/for-organisations/documents/1596/privacy-in-mobile-apps-dp-guidance.pdf.

[58] Google LLC. Fundamentals of testing, 2018. URL https://developer.android.com/training/testing/fundamentals.html.

[59] Google LLC. Core app quality, 2017. URL https://developer.android.com/docs/quality-guidelines/core-app-quality.

[60] Stoyan Stoyanov, Leanne Hides, David Kavanagh, Oksana Zelenko, Dian Tjondronegoro, and Madhavan Mani. Mobile app rating scale: A new tool for assessing the quality of health mobile apps. 3:e27, 03 2015.

[61] St. Jude Medical. mycardiomems, 2018. URL https://play.google.com/store/apps/details?id=com.sjm.companion&hl=en.

Appendix A

Additional documents

All documents mentioned throughout the report including the code for the Android app, datasets or scripts can be found in Github. Below is a list of repositories, all of which can be found at `https://github.com/PauloIswanto`:

- **BPApp**: This repository contains all code related to the Android application.
- **MIMIC-PAP-Dataset**: This repository contains all PAP records derived from the MIMIC-III database as outlined in chapter 7.
- **Final-Year-Project**: This repository contains any other document mentioned in the report.

Appendix B

MARS Questionnaire

App Quality Ratings

The Rating scale assesses app quality on four dimensions. All items are rated on a 5-point scale from "1.Inadequate" to "5.Excellent". Circle the number that most accurately represents the quality of the app component you are rating. Please use the descriptors provided for each response category.

SECTION A

Engagement – fun, interesting, customisable, interactive (e.g. sends alerts, messages, reminders, feedback, enables sharing), well-targeted to audience

1. **Entertainment: Is the app fun/entertaining to use? Does it use any strategies to increase engagement through entertainment (e.g. through gamification)?**
   1 Dull, not fun or entertaining at all
   2 Mostly boring
   3 OK, fun enough to entertain user for a brief time (< 5 minutes)
   4 Moderately fun and entertaining, would entertain user for some time (5-10 minutes total)
   5 Highly entertaining and fun, would stimulate repeat use

2. **Interest: Is the app interesting to use? Does it use any strategies to increase engagement by presenting its content in an interesting way?**
   1 Not interesting at all
   2 Mostly uninteresting
   3 OK, neither interesting nor uninteresting; would engage user for a brief time (< 5 minutes)
   4 Moderately interesting; would engage user for some time (5-10 minutes total)
   5 Very interesting, would engage user in repeat use

3. **Customisation: Does it provide/retain all necessary settings/preferences for apps features (e.g. sound, content, notifications, etc.)?**
   1 Does not allow any customisation or requires setting to be input every time
   2 Allows insufficient customisation limiting functions
   3 Allows basic customisation to function adequately
   4 Allows numerous options for customisation
   5 Allows complete tailoring to the individual's characteristics/preferences, retains all settings

4. **Interactivity: Does it allow user input, provide feedback, contain prompts (reminders, sharing options, notifications, etc.)? Note: these functions need to be customisable and not overwhelming in order to be perfect.**
   1 No interactive features and/or no response to user interaction
   2 Insufficient interactivity, or feedback, or user input options, limiting functions
   3 Basic interactive features to function adequately
   4 Offers a variety of interactive features/feedback/user input options
   5 Very high level of responsiveness through interactive features/feedback/user input options

5. **Target group: Is the app content (visual information, language, design) appropriate for your target audience?**
   1 Completely inappropriate/unclear/confusing
   2 Mostly inappropriate/unclear/confusing
   3 Acceptable but not targeted. May be inappropriate/unclear/confusing
   4 Well-targeted, with negligible issues
   5 Perfectly targeted, no issues found

A. Engagement mean score = ____________

SECTION B

Functionality – app functioning, easy to learn, navigation, flow logic, and gestural design of app

6. **Performance: How accurately/fast do the app features (functions) and components (buttons/menus) work?**

   1 App is broken; no/insufficient/inaccurate response (e.g. crashes/bugs/broken features, etc.)
   2 Some functions work, but lagging or contains major technical problems
   3 App works overall. Some technical problems need fixing/Slow at times
   4 Mostly functional with minor/negligible problems
   5 Perfect/timely response; no technical bugs found/contains a 'loading time left' indicator

7. **Ease of use: How easy is it to learn how to use the app; how clear are the menu labels/icons and instructions?**

   1 No/limited instructions; menu labels/icons are confusing; complicated
   2 Useable after a lot of time/effort
   3 Useable after some time/effort
   4 Easy to learn how to use the app (or has clear instructions)
   5 Able to use app immediately; intuitive; simple

8. **Navigation: Is moving between screens logical/accurate/appropriate/ uninterrupted; are all necessary screen links present?**

   1 Different sections within the app seem logically disconnected and random/confusing/navigation is difficult
   2 Usable after a lot of time/effort
   3 Usable after some time/effort
   4 Easy to use or missing a negligible link
   5 Perfectly logical, easy, clear and intuitive screen flow throughout, or offers shortcuts

9. **Gestural design: Are interactions (taps/swipes/pinches/scrolls) consistent and intuitive across all components/screens?**

   1 Completely inconsistent/confusing
   2 Often inconsistent/confusing
   3 OK with some inconsistencies/confusing elements
   4 Mostly consistent/intuitive with negligible problems
   5 Perfectly consistent and intuitive

B. Functionality mean score = ___________

SECTION C

Aesthetics – graphic design, overall visual appeal, colour scheme, and stylistic consistency

10. **Layout: Is arrangement and size of buttons/icons/menus/content on the screen appropriate or zoomable if needed?**

    1 Very bad design, cluttered, some options impossible to select/locate/see/read device display not optimised
    2 Bad design, random, unclear, some options difficult to select/locate/see/read
    3 Satisfactory, few problems with selecting/locating/seeing/reading items or with minor screen-size problems
    4 Mostly clear, able to select/locate/see/read items
    5 Professional, simple, clear, orderly, logically organised, device display optimised. Every design component has a purpose

11. Graphics: How high is the quality/resolution of graphics used for buttons/icons/menus/content?

1 Graphics appear amateur, very poor visual design - disproportionate, completely stylistically inconsistent
2 Low quality/low resolution graphics; low quality visual design – disproportionate, stylistically inconsistent
3 Moderate quality graphics and visual design (generally consistent in style)
4 High quality/resolution graphics and visual design – mostly proportionate, stylistically consistent
5 Very high quality/resolution graphics and visual design - proportionate, stylistically consistent throughout

12. Visual appeal: How good does the app look?

1 No visual appeal, unpleasant to look at, poorly designed, clashing/mismatched colours
2 Little visual appeal – poorly designed, bad use of colour, visually boring
3 Some visual appeal – average, neither pleasant, nor unpleasant
4 High level of visual appeal – seamless graphics – consistent and professionally designed
5 As above + very attractive, memorable, stands out; use of colour enhances app features/menus

C. Aesthetics mean score = ______________

SECTION D

Information – Contains high quality information (e.g. text, feedback, measures, references) from a credible source. Select N/A if the app component is irrelevant.

13. Accuracy of app description (in app store): Does app contain what is described?

1 Misleading. App does not contain the described components/functions. Or has no description
2 Inaccurate. App contains very few of the described components/functions
3 OK. App contains some of the described components/functions
4 Accurate. App contains most of the described components/functions
5 Highly accurate description of the app components/functions

14. Goals: Does app have specific, measurable and achievable goals (specified in app store description or within the app itself)?

N/A Description does not list goals, or app goals are irrelevant to research goal (e.g. using a game for educational purposes)
1 App has no chance of achieving its stated goals
2 Description lists some goals, but app has very little chance of achieving them
3 OK. App has clear goals, which may be achievable.
4 App has clearly specified goals, which are measurable and achievable
5 App has specific and measurable goals, which are highly likely to be achieved

15. Quality of information: Is app content correct, well written, and relevant to the goal/topic of the app?

N/A There is no information within the app
1 Irrelevant/inappropriate/incoherent/incorrect
2 Poor. Barely relevant/appropriate/coherent/may be incorrect
3 Moderately relevant/appropriate/coherent/and appears correct
4 Relevant/appropriate/coherent/correct
5 Highly relevant, appropriate, coherent, and correct

16. Quantity of information: Is the extent coverage within the scope of the app; and comprehensive but concise?

N/A There is no information within the app
1 Minimal or overwhelming
2 Insufficient or possibly overwhelming
3 OK but not comprehensive or concise
4 Offers a broad range of information, has some gaps or unnecessary detail; or has no links to more information and resources
5 Comprehensive and concise; contains links to more information and resources

17. Visual information: Is visual explanation of concepts – through charts/graphs/images/videos, etc. – clear, logical, correct?

N/A There is no visual information within the app (e.g. it only contains audio, or text)
1 Completely unclear/confusing/wrong or necessary but missing
2 Mostly unclear/confusing/wrong
3 OK but often unclear/confusing/wrong
4 Mostly clear/logical/correct with negligible issues
5 Perfectly clear/logical/correct

18. Credibility: Does the app come from a legitimate source (specified in app store description or within the app itself)?

1 Source identified but legitimacy/trustworthiness of source is questionable (e.g. commercial business with vested interest)
2 Appears to come from a legitimate source, but it cannot be verified (e.g. has no webpage)
3 Developed by small NGO/institution (hospital/centre, etc.) /specialised commercial business, funding body
4 Developed by government, university or as above but larger in scale
5 Developed using nationally competitive government or research funding (e.g. Australian Research Council, NHMRC)

19. Evidence base: Has the app been trialled/tested; must be verified by evidence (in published scientific literature)?

N/A The app has not been trialled/tested
1 The evidence suggests the app does not work
2 App has been trialled (e.g., acceptability, usability, satisfaction ratings) and has partially positive outcomes in studies that are not randomised controlled trials (RCTs), or there is little or no contradictory evidence.
3 App has been trialled (e.g., acceptability, usability, satisfaction ratings) and has positive outcomes in studies that are not RCTs, and there is no contradictory evidence.
4 App has been trialled and outcome tested in 1-2 RCTs indicating positive results
5 App has been trialled and outcome tested in ≥ 3 high quality RCTs indicating positive results

D. Information mean score = ____________ *

* Exclude questions rated as "N/A" from the mean score calculation.

App subjective quality

SECTION E

20. Would you recommend this app to people who might benefit from it?

1	**Not at all**	I would not recommend this app to anyone
2		There are very few people I would recommend this app to
3	**Maybe**	There are several people whom I would recommend it to
4		There are many people I would recommend this app to
5	**Definitely**	I would recommend this app to everyone

21. How many times do you think you would use this app in the next 12 months if it was relevant to you?

1 **None**
2 1-2
3 3-10
4 10-50
5 >50

22. Would you pay for this app?

1 No
3 Maybe
5 Yes

23. What is your overall star rating of the app?

1	★	One of the worst apps I've used
2	★★	
3	★★★	Average
4	★★★★	
5	★★★★★	One of the best apps I've used

Scoring

App quality scores for

SECTION

A: Engagement Mean Score = ________________________

B: Functionality Mean Score = ________________________

C: Aesthetics Mean Score = ________________________

D: Information Mean Score = ________________________

App quality mean Score = ________________________

App subjective quality Score = ________________________

App-specific

These added items can be adjusted and used to assess the perceived impact of the app on the user's knowledge, attitudes, intentions to change as well as the likelihood of actual change in the target health behaviour.

SECTION F

1. **Awareness: This app is likely to increase awareness of the importance of addressing [insert target health behaviour]**

   Strongly disagree — Strongly Agree

   1 2 3 4 5

2. **Knowledge: This app is likely to increase knowledge/understanding of [insert target health behaviour]**

   Strongly disagree — Strongly Agree

   1 2 3 4 5

3. **Attitudes: This app is likely to change attitudes toward improving [insert target health behaviour]**

   Strongly disagree — Strongly Agree

   1 2 3 4 5

4. **Intention to change: This app is likely to increase intentions/motivation to address [insert target health behaviour]**

   Strongly disagree — Strongly Agree

   1 2 3 4 5

5. **Help seeking: Use of this app is likely to encourage further help seeking for [insert target health behaviour] (if it's required)**

   Strongly disagree — Strongly Agree

   1 2 3 4 5

6. **Behaviour change: Use of this app is likely increase/decrease [insert target health behaviour]**

   Strongly disagree — Strongly Agree

   1 2 3 4 5