# Digitised ECG Mobile App


Presented by:

Dimpho Sefora

Prepared for:

Dr. Yaaseen Martin

September 16, 2025

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for the degree of Electrical and Computer Engineering.

# Abstract

In this work we describe a novel Thesis Template, to be used by students in Electrical & Engineering at the University of Cape Town. This section entails the abstract of the document.

# Acknowledgments

# Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.

3. This final year project report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as there own work or part thereof.

.....

Dimpho Sefora

September 16, 2025

# Contents

# List of Figures

# List of Tables

# Nomenclature

**depolarization wave** the wave of electrical activity that spreads through the heart muscle, causing it to contract. 7

**ECG** Electrocardiogram. 1

# Chapter 1

# Introduction

## 1.1 Background

An electrocardiogram ECG is a technology that both measures and records the electrical signal patterns describing the rhythmic activity of the heart. These electrical pulses are what signal the heart's skeletal muscles to undergo ventricular contraction [1]. Irregularities from the expected patterns provide a wealth of information on the patient's condition.

Clinically, two ECG systems are in use: paper-recorded and computer-based. Paper ECGs remain the foundational format and are routinely requested as an initial diagnostic test [1]. Computer-based ECGs, first introduced in the 1960s [2], aimed to reduce the dependency on specialist interpretation, but continue to face issues of diagnostic reliability and digital record management [3]. Furthermore, due to significant technological and logistic barriers associated with accessibility and cost, the widespread adoption of these systems has been limited [4] [5]. Therefore, paper ECGs

remain widely produced. However, their physical form presents a persistent limitation: secure and accessible storage for later interpretation. Addressing this challenge motivates the digitisation of paper ECGs into formats that can be efficiently stored and accessed, especially for the long-term treatment of cardiac patients [6].

Manual digitisation, which involves manually scanning or transcribing paper ECG tracings into digital systems, is a popular solution to this issue. Although this technique eventually makes it possible to save ECGs in electronically, it is expensive, time-consuming, and prone to errors, especially in environments with lots of patients [7].

Recent developments in mobile health technologies suggest a potential pathway to address this limitation. Smartphones, with their widespread availability and camera facilities, enable paper ECGs to be captured, digitised, and stored within mobile applications [8] [9]. This strategy creates a way to bridge the gap between affordable availability to digital ECG formats and possibly addressing the storage problem [10]. Mobile platforms can enhance patient record management and continuity of treatment by enabling the direct archiving and retrieval of ECG information on portable devices. However, while promising, these developments also highlight the need to explore how paper-recorded ECGs can be reliably digitised and integrated into mobile-accessible formats.

## 1.2 Problem Statement

The digitisation of paper-based electrocardiograms continues to present a challenge, as reliance on physical records causes inefficiencies in storage, retrieval, and long-term accessibility [8,11]. Despite being widespread, manual digitisation is expensive,

inefficient, and prone to errors [7] and existing mobile scanning tools are not designed for ECG waveforms, leading to distortion, misalignment, and loss of signal fidelity [12]. Current methods are inadequate for efficiently converting paper ECGs into accurate, analysis-ready digital formats, highlighting the need for reliable, mobile-accessible digitisation solutions.

## 1.3 Objectives

These objectives guide the progression of the research, ensuring that the work done is centred on the problem statement. The following will be objectives will be explored throughout the report:

1. Review literature on ECG signal structure, interpretation, and digitisation techniques.
2. Investigate existing image processing and signal processing methods for waveform extraction from photographs.
3. Design and develop a mobile application (Android or iOS) for accurate ECG strip digitisation.
4. Implement algorithms for noise reduction, de-warping, filtering, and artifact removal.
5. Store processed ECG signals in a standardised digital format.
6. Test and evaluate the accuracy and usability of the developed application.

## 1.4 Scope and Limitations

**Scope:**

- Mobile application development for Android/iOS.
- Image capture, processing, and signal extraction for standard ECG paper strips.
- Local storage of digitised ECG in standard digital formats.

**Limitations:**

- No integration with electronic health record (EHR) systems in this phase.
- Limited dataset for testing.
- Optimised for standard-format single-lead or multi-lead strips.
- No integration of deep-learning analysis and diagnosis of digitised data

## 1.5 Thesis Outline

- **Chapter 2**: Literature Review — ECG basics, digitisation methods, and relevant image/signal processing techniques.
- **Chapter 3**: Methodology and Design — approach, algorithms, and system architecture.
- **Chapter 4**: Implementation — development process and app features.

- **Chapter 5**: Results and Discussion — evaluation and performance.
- **Chapter 6**: Conclusion — summary of findings and future work.

# Chapter 2

# Literature Review

This chapter provides a comprehensive evaluation of literature relevant to the research topic. It begins with an introductory section outlining the motivation for the study, focusing on the need to further investigate mobile app development for digitising and storing ECG paper records. A brief overview of the theory behind ECG signal formation follows, providing the foundation for understanding its digitisation.

The review then adopts a funnel approach: first examining existing methods of ECG record digitisation, then evaluating the potential of mobile applications for this purpose through image capture, and finally considering techniques for extracting ECG waveforms from images. The chapter concludes with a summary of key findings, identifying gaps in current research and setting the stage for the methodology that follows.

## 2.1 Introduction

The electrocardiogram (ECG) has been a fundamental tool in medicine since 1901 [8]. Over the years, a vast number of ECG paper records have been produced. Consequently, researchers have devised methods to digitise these records in order to ensure long-term preservation, improve accessibility, and facilitate future evaluation and diagnosis. It is therefore important to investigate and assess these methods to understand their respective strengths and limitations.

With the growing use of mobile applications in healthcare [11], new opportunities for ECG digitisation have emerged. This enables investigation into whether ECG images captured with mobile devices can be effectively processed for digitisation.

## 2.2 ECG Fundamentals

Electrocardiogram signals show how the path of the depolarization wave - the flow of a group of positive electric charges - moves during each heartbeat [13]. Hearts have natural pacemakers that initiate electric signals that cause atrial and ventricular contraction [14]. Electrocardiograms are built to sense these natural signals and convert them into useful information that portray the heart's condition.

### 2.2.1 ECG Signal Generation

The electric activity of the heart is governed by the sinoatrial (SA) and atrioventricular (AV) nodes. The SA node initialises the impulses and the AV node introduces delays to their delivery to co-ordinate between arterial and ventricular contractions

for optimum heart functionality [14].

As the positive charges move through the heart's muscles, their negative resting state changes to a positive active state. This change in charge of the internal muscles affects a change to the external charge surrounding it [13]. This creates the potential difference measured by the electrocardiogram electrodes. If detected it is noted as a deflection on the ECG waveform. The bigger the potential difference, the larger the deflection [13].

(image explaining or showing ECG deflections - so they know what to look for in the actual trace)

### 2.2.2 ECG Waveform Formation

An ECG signal consists of five deflections that together form the PQRST complex, representing a single heartbeat and the complete cycle of myocardial contraction and relaxation [1].

The P wave reflects atrial depolarisation, initiating atrial contraction. It is typically a low-amplitude, gently sloped deflection due to the slower propagation of the depolarisation wave through atrial myocardial cells [15]. The QRS complex corresponds to ventricular depolarisation and the onset of ventricular contraction, with the R wave being the largest positive deflection [16]. The T wave represents ventricular repolarisation, preparing the ventricles for the next cardiac cycle [16].

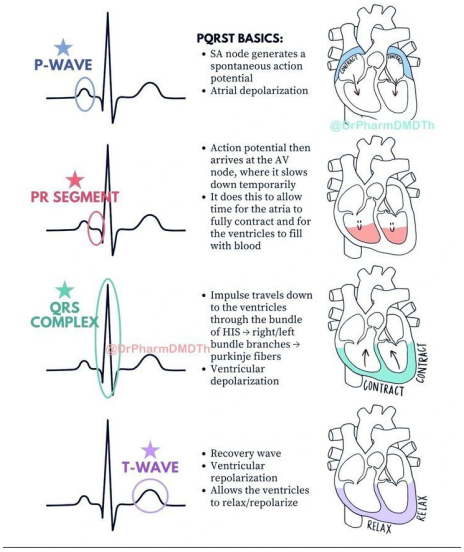


Figure 2.1: Placeholder for ECG diagram

The timing and amplitude of the PQRST wave are essential for patient health analysis [17]. Significant signal loss in the digisation process would thus render the ECG useless for diagnostic purposes.

### 2.2.3 ECG leads

The direction of this wave is dependent on the leads. The leads are defined by the electrodes used. In a 12-lead ECG, there are 10 electrodes used to measure the heart's activity from different angle and planes. Each lead provides different and vital information about arrhythmias and myocardial infarction (heart attacks).

There are the limb electrodes places by the left arm and leg and the right arm. Measurements between two of them, with the third being a reference, creates the I, II, and III dipoles leads. When a fourth electrode known as the reference or zero

electrode is introduced, the augmented leads are created. This is a measurement from one of the limb electrodes to the reference electrodes. This creates one of the unilateral leads.

In 1934, it was noted that there were areas in the heart that needed to be measured to better detect myocardial infarctions. This gave birth to the precordial leads through the use of six electrodes placed around the precordial (chest area about and around the heart).

These make up the 12 leads that illustrate the different waveform patterns seen in the electrocardiogram trace. They are used to section the ECG into 12 important regions which can be segmented and analysed further. [1]

## 2.3 Traditional ECG Digitisation Methods

Digitisation of ECG paper records has been an area of interest for decades, with various methods developed to convert analog ECG signals into digital formats. This section reviews traditional digitisation techniques, highlighting their methodologies, advantages, and limitations.

Flatbed scanning and manual transcription have been attempted. Advantage: Simple and low-cost. Disadvantage: Manual methods are time-consuming and scanning introduces distortions and noise requiring post-processing [18].

Scanning with morphological filtering was widely used. **Advantage**: Controlled scan environment ensures consistent image quality. Produces clean signals with high-resolution scans. **Disadvantage**: Not portable compared to smartphones, At lower resolution (300 dpi), characters cannot be removed effectively and waveform loss

occur. [**?**].

Standard paper ECGs use a grid with scaling of 25 mm/s (time) and 10 mm/mV (voltage), arranged in a 3x4 lead display

Factors like paper degradation and print resolution can introduce noise, affecting digitization accuracy [19].

## 2.4 Smartphone-based ECG Capturing

Smartphones enable easy ECG capture with modern cameras. Advantage: High accessibility and does not require hospital scanners. Disadvantage: Image quality highly dependent on lighting, perspective, and shadows [18].

Smartphones offer high-resolution cameras and processing power, making them viable for ECG capture. However, general document scanning apps lack ECG-specific features, such as grid removal and signal enhancement. Existing ECG apps often fail to handle highly noisy or binary scans effectively [19].

## 2.5 ECG Digitisation from Images

### 2.5.1 Image Pre-processing

For camera images, the following is performed: RGB-channel splitting, dilation to suppress dark shadows, median blurring, and normalization to reduce salt-and-pepper noise; enabling field capture and materially improves thresholding and downstream extraction, but it still struggles with non-uniform illumination and requires careful text removal to avoid signal damage where labels overlap the trace. [18]

#### De-warping and Perspective Correction

**Hough Transform–Based Rectification:** The Hough Transform exploits the regular grid structure typically present in ECG reports. It detects straight lines—whether slanted, vertical, or horizontal—and uses their intersections to classify geometric shapes. Once identified, geometric rectification is applied by mapping these intersections (corners) to an ideal square lattice. This process expands or contracts distorted lines so that rhombi and parallelograms are reformed into rectangles. Such an approach effectively aligns warped grids, although its performance is sensitive to scan quality and noise in the image [19].

**Geometric Rectification:** In addition to Hough-based detection, geometric rectification is widely used to correct perspective distortions introduced during image capture. For ECGs, perspective skew can significantly affect the readability of waveforms. By applying geometric transformations, images are realigned to restore proportionality between the waveform and grid axes. While this method is often combined with Hough detection for corner localization, its accuracy is dependent on

reliable feature extraction and the absence of severe scanning artifacts.

### Noise Reduction

Noise reduction is a crucial preprocessing step in ECG image analysis, as various methods offer trade-offs between suppressing artifacts and preserving clinically important waveform features. These approaches range from traditional filtering techniques to deep learning models, each with distinct advantages and limitations.

**Median Filtering:** Median filtering replaces each pixel with the median of its local neighborhood. It is particularly effective at removing salt-and-pepper noise and is computationally efficient. However, it has limited effectiveness against structured noise such as grid lines. Moreover, median filtering can blur sharp ECG features, most notably the QRS complexes, thereby reducing diagnostic accuracy [19].

**Adaptive Filtering:** Adaptive filtering modifies smoothing strength based on local noise levels, enabling better edge preservation compared to uniform filters. While this method reduces over-smoothing and retains more waveform detail, it continues to struggle with grid-line interference. Studies indicate that adaptive filters alone are insufficient for ECG applications and typically require additional post-processing to achieve acceptable results.

**Wavelet Denoising:** Wavelet-based approaches employ frequency-domain thresholding to suppress noise, and are effective for reducing Gaussian noise. However, they risk distorting high-frequency ECG components, including the QRS complex. The success of wavelet denoising is highly dependent on threshold selection, where overly aggressive thresholds may inadvertently remove clinically significant features [19].

**Grayscale Scanning Pathway:** As an alternative approach grayscale contrast adjustment (imadjust) can be used to enhance the visibility of waveforms before thresholding. This is followed by morphological background estimation and subtraction. The technique is advantageous because it requires only a few parameters and executes quickly. However, it often leaves residual artifacts that must be corrected later through region-based segmentation. Furthermore, high-resolution scanning (around 600 dpi) is necessary to avoid jagged or discontinuous waveform outputs [20].

**Deep Learning Approaches:** Recent advances include the use of U-Net architectures with ResNet blocks and attention mechanisms. In Li et al.'s evaluation, this method achieved a DICE similarity coefficient of 0.85 and a Pearson correlation of less than 0.9. The DICE coefficient measures the overlap between predicted and ground-truth segmentations, making it a critical index for assessing segmentation accuracy in biomedical imaging. This approach automatically handles diverse noise types without manual parameter tuning and outperforms traditional methods. However, it requires large annotated datasets for training, demands high computational resources, and still struggles with preserving sharp QRS transitions—an essential factor for accurate interval measurements.

Overall, all methods face the fundamental trade-off between noise suppression and feature preservation. Traditional techniques such as median, adaptive, and wavelet filtering are computationally efficient and straightforward to implement but struggle with structured noise and tend to degrade sharp ECG features. Deep learning methods demonstrate superior performance across diverse noise conditions but introduce challenges related to computational expense and training data requirements.

### Artifact Removal and Enhancement

Artifacts in ECG scans often arise from pen markings, smudges, paper folds, or printed annotations, and can significantly interfere with waveform extraction. Several approaches have been developed to suppress such artifacts while minimizing signal distortion.

**Morphological and Threshold-Based Methods:** Li et al. applied morphological operations combined with connected component analysis to remove small, disconnected artifacts while preserving the ECG waveform. Their approach used size and shape characteristics to differentiate true signal from noise. For pen markings, threshold-based masking was applied to eliminate high-contrast elements. Although effective, this approach sometimes removed genuine ECG segments when artifacts overlapped with waveforms, resulting in a reported 12

**Deep Learning Approaches:** Deep learning methods, particularly U-Net architectures, have been explored for artifact suppression. These models learn to identify and remove artifacts while preserving critical ECG structures such as P-QRS-T complexes. In Li et al.'s study, U-Net achieved 92% accuracy in artifact removal without degrading the ECG signal. However, performance dropped to 84% in cases with severe paper folds or tears, and artifacts that directly overlapped with key waveform features remained problematic. Manual correction was still required in 8% of cases. Despite these challenges, the U-Net approach demonstrated the best balance between artifact suppression and signal preservation overall [19].

- **Background Subtraction and Morphological Filtering:** Tun et al. proposed a pipeline where background subtraction is followed by local entropy filtering, morphological area opening, and region-based segmentation with size

thresholds. A final morphological "thicken" operation smooths residual edge artifacts. This method reliably suppresses printed text in high-resolution (600 dpi) scans and produces clean binary traces. However, at lower resolutions (e.g., 300 dpi), residual characters persist and waveform fragments may be lost due to over-aggressive filtering or inadequate sampling [20].

- **Post-Binarization Repair:** Mishra et al. addressed artifacts through post-binarisation techniques. Their method first dilates the ECG trace to bridge gaps caused by noise and then skeletonizes the signal back to a single-pixel width. Printed lead names were suppressed by a column-wise scanning technique that preserved the lowest white pixel (the waveform) and removed higher ones (text). This process generated continuous, single-pixel signals suitable for digital extraction. However, the column logic occasionally removed portions of steep R-waves or damaged traces overlapped by text, necessitating a repair step to restore continuity [18].

Overall, traditional morphological and threshold-based methods are computationally efficient but risk partial signal loss, especially when artifacts overlap with the waveform. Background subtraction and post-binarisation techniques offer stronger text suppression but are sensitive to scan resolution and may distort sharp features. Deep learning approaches outperform traditional techniques by adapting to diverse artifact types, though they require large training datasets and still face limitations when artifacts directly obscure critical ECG segments.

### 2.5.2 Grid Detection and Extraction or Grid Removal and Scaling

The grid is eliminated by applying the DL-predicted global threshold derived from the LOB curve; for scaling they either measure the pixel size of the standard boxes or isolate colored red squares to derive 0.2 s and 0.5 mV per square before converting pixel coordinates to time/voltage. Advantages include strong binarization accuracy on mixed datasets and explicit, data-driven scaling; disadvantages appear under non-uniform illumination or faint grids without prior luminance correction. [18]

Grid removal is performed by global grayscale-to-binary conversion (im2bw/graythresh) followed by morphological background estimation and subtraction; time-voltage scaling is then computed from scan DPI and standard paper settings (25 mm/s, 1 mV = 10 mm) to map pixels to physical units. This is deterministic and accurate at 600 dpi but degrades at 300 dpi where missing strokes and unresolved small characters undermine both cleanup and scaling fidelity. [20]

Grid lines were detected using horizontal/vertical pixel summation to identify periodic patterns. The method calculated autocorrelation peaks to determine grid spacing, achieving 94% accuracy for solid lines but only 72% for dotted lines. For extraction, Li et al. used dynamic programming with a cost function balancing distance and angle continuity. This preserved signal morphology but failed at sharp QRS complexes 18% of the time. The paper reported mean error of 0.4mm in grid alignment. Deep learning approaches treated grid removal as segmentation, with U-Net achieving 89% precision. The model learned grid patterns directly from data, handling irregular spacings better than algorithmic. Hybrid methods combined traditional grid detection with neural network refinement. These reduced errors by 32% compared to pure algorithmic approaches, but added computational complexity. All

methods performed worse on low-contrast scans (less than 600 DPI) [19].

Sobel edge detection was tested as a preprocessing step for grid removal, but proved inadequate for ECG-specific challenges. The operator achieved 84% edge detection accuracy for major waveforms but failed to distinguish between grid lines and signal contours in 31% of cases (cite ECG digitisation). Angular sensitivity caused partial detection of diagonal grid artifacts. [19]

### 2.5.3 Signal Detection and Extraction

#### Lead Segmentation

Take the image and use a OpenCV algorithm that semi-manually creates a grid where each block contains a lead. Each block is than converted into an image (producing 12 images from the one). The boxes were created by detecting the individual shapes of each lead [18].

#### Signal Extraction

**Using binary thresholding and deep-learning:** Manually find the threshold for each image that would produce the signal alone without the grid. Use the threshold value and the characteristic curve (defines the number of distinct values in a particular grayscale i.e. intensity value of the grid and the signal would create two slopes in the curve) as groundtruths to the training data for the deep learning-based model. The goal would be to have the model automatically produce the threshold and signal for any input image [18].

The Otsu algorithm was employed for initial binarization, automatically determining optimal thresholds to separate ECG signals from background. While effective for clean scans (92% accuracy), performance degraded to 68% for low-contrast or noisy images where histogram bimodality was less distinct [19]. False positives occurred particularly in regions with smudges or shaded artifacts.

Approaches combined Otsu with morphological operations, improving baseline performance to 79% accuracy for degraded scans. However, these hybrid methods were ultimately superseded by deep learning techniques which demonstrated superior robustness (DICE 0.85 vs 0.72 for traditional pipelines) [19].

## 2.6 Conclusion

- Summary of reviewed methods and gaps.
- Justification for proposed app.
- Key algorithmic challenges for this project.

- Broad description of subject
- Some relevant history
- Current implementations in industry
- New & Related Research on the subject

# Chapter 3

# Methodology and Design

This chapter details the design and engineering decisions taken to develop and evantually implement digitising ECG signals through mobile applications. These decisions are guided by the observations highlighted in the .

It will begin with a highlevel overview of the proposed approach for tackling the problem. It will then provide specific details on the methods and techniques used in the research, including any algorithms, tools, and technologies employed. The chapter will also discuss the rationale behind these choices, highlighting how they align with the research objectives and address the identified challenges.

## 3.1 Overview of Approach

The core principle of the proposed approach is iteration. With each new stage of development, earlier steps are revisited to ensure ongoing risk management and

continuous improvement. This cyclical process mirrors the spiral model commonly applied in system and software design.

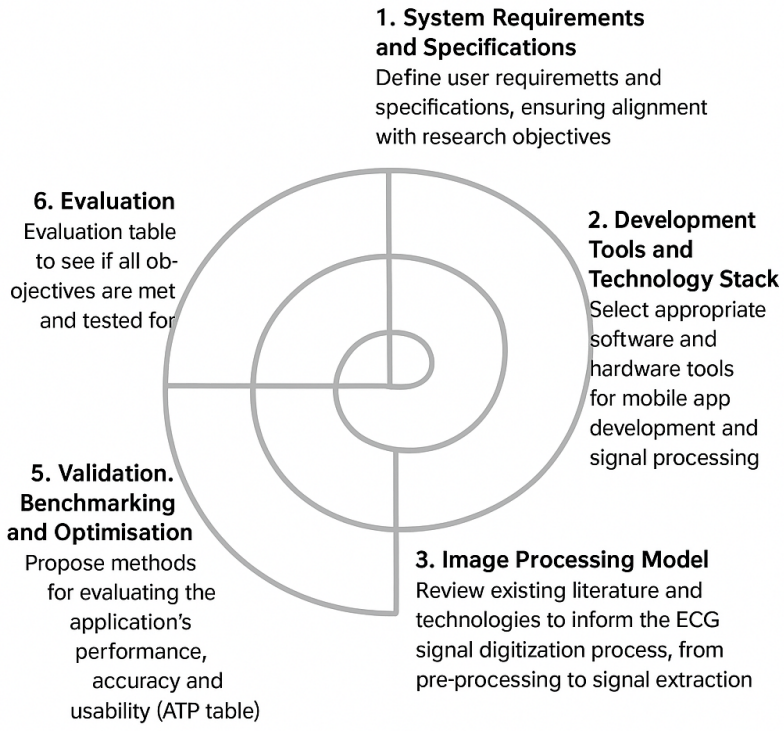


Figure 3.1: Spiral Model of Development

1. **System Requirements and Specifications:** Define user requirements and specifications, ensuring aignment with research objectives
2. **Development Tools and Technology Stack:** Select appropriate software and hardware tools for mobile app development and signal processing
3. **Image Processing Model:** Review existing literature and technologies to inform the ECG signal digitisation process, from pre-processing to signal extraction

4. **Design and Prototyping:** Create iterative versions of the mobile application to test core functionalities, using wireframes for illustration
5. **Validation, Benchmarking, and Optimisation:** Propose methods for evaluating the application's performance, accuracy, and usability (ATP table)
6. **Evaluation:** Evaluation table to see if all objectives are met and tested for

## 3.2 System Requirements and Specifications

The objectives of the research are clear: design a mobile application (Android or iOS) that can (1) facilitate the capturing of ECG paper records, (2) process the captured images to extract the ECG waveform, and (3) store the digitised ECG signals in a standardised digital format. The following user and functional requirements and specifications are derived from these objectives.

### 3.2.1 Functional User Requirements

Table 3.1: Functional User Requirements

| #no. | Requirement | Description |
|---|---|---|
| FR01 | Camera Access | Application must be able to access the mobile's camera |
| FR02 | Image Import | Access photo library for image upload |
| FR03 | Pre-processing | Must facilitate cleaning and aligning of image for signal extraction |
| FR04 | Signal extraction | Detect signal and extract it in pixel coordinates |
| FR05 | Reconstruction | Convert pixel coordinates to amplitude-time series |
| FR06 | Data storage | Facilitate storing of digitised data in external device or cloud in CSV format |
| FR07 | Visualisation | Display interactive ECG waveform |

### 3.2.2 Non-Functional User Requirements

Table 3.2: Non-Functional User Requirements

| #no. | Requirement | Description |
|---|---|---|
| NFR01 | Performance | Image capture and signal extraction should complete within acceptable time ($< 5$ sec) |
| NFR02 | Accuracy | Maintain reliable signal extraction across various ECG paper qualities |
| NFR03 | Usability | Interface must be simple, clear, and intuitive for users with minimal training |

| | | |
|---|---|---|
| NFR04 | Reliability | Handle errors gracefully and allow retry; ensure data persists between sessions |
| NFR05 | Compatibility | Support Android or iOS devices with standard camera specifications |
| NFR06 | Security | Store data securely on device; optionally encrypt sensitive information |

## 3.3 Tools and Technology Stack

Defining the development environment for any software project is crucial, as it influences the efficiency of the development process, the quality of the final product, and the ease of maintenance. This section outlines the chosen tools and technologies for developing the mobile application for digitising ECG signals. Notice that the chosen operating system is iOS because of its comprehensive set of libraries and frameworks for image processing through Xcode.

### 3.3.1 Development Tools and Specifications

- **Xcode:** The integrated development environment (IDE) for macOS, used for developing software for iOS. It provides a suite of tools for designing user interfaces, writing code, and debugging applications.
- **Apple Developer:** A platform that provides resources and tools for iOS app development, including access to beta software, advanced app capabilities, and app analytics. This is essential for deploying to a physical iOS device for testing through Testflight.

- **Simulator:** A tool within Xcode that allows developers to test and debug iOS applications on a Mac without needing a physical device. It simulates various iPhone and iPad models and iOS versions.
- **iPhone 13:** A physical device used for testing the application in real-world conditions, ensuring that it performs as expected on actual hardware and observing real user interactions.
- **VMware Workstation Pro:** A virtualisation software used to run macOS on a Windows machine, enabling the use of Xcode and other macOS-specific tools without needing a dedicated Mac. It provides a controlled environment for iOS development and testing. The specfic macOS installed is Sequioa, running on a Mac Mini 4. For compatibility with the Windows PC, the virtual machine is allocated 4GB RAM, 4 CPU cores and 128GB of disk space.

### 3.3.2 Technology Stack

**iOS SDK** The iOS SDK is Apple's official development toolkit for building applications on iPhones and iPads. It provides access to the operating system's APIs, device hardware (camera, sensors, storage), and system-level services. This research project requires close integration with the iPhone camera for ECG capture, efficient on-device signal processing, and secure file handling. The iOS SDK ensures native access to all necessary features while maintaining compliance with Apple's ecosystem.

It provides a complete and unified set of frameworks for image processing, file management, and graphics rendering, while being ptimized for Apple hardware, ensuring smooth performance and efficient energy usage. It offers long-term reliability and stability since it is fully supported by Apple.

- **Security:** Applications built with the iOS SDK run in a sandboxed environment, preventing unauthorized access to other apps or system files. It supports data encryption at rest and in transit, aligning with healthcare data protection needs (e.g., HIPAA compliance). It requires that all apps be **digitally signed**, which ensures authenticity and protects users from tampered or malicious code.
- **Performance:** The SDK provides frameworks (e.g., Core Image, Accelerate) that are hardware-accelerated using the CPU, GPU, and Neural Engine. It ensures real-time responsiveness, critical for ECG pre-processing and visualization tasks. The native compilation reduces overhead compared to cross-platform solutions, leading to faster execution and lower battery drain.
- **Integration:** The SDK seamlessly integrates with device features such as the camera, storage, sensors, and security mechanisms. Furthermore, it provides compatibility with higher-level frameworks such as Core Image (image processing), Vision (computer vision), and CloudKit (cloud storage). It as well ensures consistent user experience by adhering to iOS UI/UX design standards.

#### Core iOS Frameworks and Libraries

- **Core Image:** Apple's image processing framework designed for filtering, noise reduction, normalization, and geometric corrections such as dewarping ECG scans. It uses GPU acceleration, allowing fast and energy-efficient processing on mobile hardware. Core Image can work in parallel with Vision to prepare images for further feature extraction.
- **Vision Framework:** A high-level computer vision framework for feature detection, object recognition, and image alignment. Vision is useful for detecting

ECG grid lines and aligning the paper image before signal extraction. It complements Core Image by handling structural analysis while Core Image focuses on pixel-level filtering.

- **Accelerate Framework:** A low-level framework that provides optimized mathematical, signal processing, and digital signal processing (DSP) functions. It uses vectorization and hardware acceleration (CPU and GPU) to speed up operations. This is crucial for transforming images into ECG signals, and it often works in sequence after preprocessing by Core Image and Vision.

- **Foundation Framework:** A system-level framework that provides essential services such as data structures, file management, and I/O operations. It is used for generating and managing CSV files that store the extracted ECG time-series data. Its advantage lies in efficiency and reliability when handling structured data within the iOS environment.

- **Core Graphics / Quartz 2D:** A two-dimensional rendering engine that allows vector-based and pixel-accurate drawing. It is used in DigECG to visualize ECG waveforms on the app's interface. This framework complements Accelerate, where Accelerate computes the signal data and Core Graphics ensures precise visual representation.

**External Tools for Benchmarking and Evaluation**

- **OpenCV:** An open-source computer vision library that provides a wide range of algorithms for image segmentation, edge detection, and noise removal. OpenCV is advantageous for prototyping and experimentation, as it offers robust tools not natively available in iOS. In DigECG, it was used in parallel with Core

Image and Vision during development to validate preprocessing accuracy and ensure robustness under varied conditions.

- **MATLAB:** A high-level scientific computing environment with specialized toolboxes for biomedical signal processing and analysis. MATLAB was used to benchmark ECG extraction accuracy against ground truth datasets, and to apply advanced filtering and spectral analysis. It complements OpenCV by focusing on quantitative evaluation of the extracted signals, ensuring that image-to-signal conversion methods in the app are scientifically validated.

**Cloud Integration**

- **CloudKit:** Apple's cloud service framework for storage and synchronization across devices. CloudKit provides end-to-end encryption and secure authentication, making it suitable for sensitive healthcare data. For DigECG, it is planned as an extension to store digitized ECG files securely, ensuring data is not only stored locally but also accessible across devices for patient monitoring and medical record keeping. This complements local CSV file storage by adding reliability and redundancy through cloud backup.

## 3.4 Image Processing Model

In this section, the observations from the liteature will be used to draw up and choose a model for the image processing required for digitising the ECG signal.

## 3.5 Design and Prototyping

Prototyping bridges the conceptualisation of the app and its realisation. Low-fidelity wireframes are used to outline structure, navigation, and features. Iterative refinements ensure usability and alignment with requirements.

### 3.5.1 Wireframes

#### Home / Dashboard Wireframe

**Purpose:** Navigation hub.

**Features:**

- Upload/scan ECG image.
- Access recent digitised signals.
- Settings (signal scaling, export formats).

#### Image Pre-processing Screen Wireframe

**Purpose:** Present uploaded ECG scan and allow preprocessing.

**Features:**

- Toggle grid removal.
- Noise reduction preview.

- Region-of-interest selection.
- Undo/redo controls.

**Signal Extraction Screen Wireframe**

**Purpose:** Display detected ECG trace.

**Features:**

- Overlay of raw scan and extracted waveform.
- Baseline correction adjustments.
- Manual correction tools (erase/add points).

**Signal Reconstruction & Visualisation Wireframe**

**Purpose:** Display reconstructed digital ECG waveform

**Features:**

- Interactive graph (zoom, pan, scale).
- Lead/time scaling indicators (mm/s, mV).
- Resampling and export options.

**Export / Results Screen Wireframe**

**Purpose:** Final storage and sharing stage

**Features:**

- Export options (CSV, JSON, MATLAB, PDF).
- Share via email/cloud.
- Comparison view (digitised vs raw scan).

### 3.5.2 Iterations

The initial prototype offered only ...

Iterations added:

- Preprocessing toggles (grid/noise removal).
- Manual correction tools for overlapping artifacts.
- Interactive waveform viewer (zoom, pan, scaling).
- Multiple export formats.

### 3.5.3 User Flow

**Home Screen**

- Case 1: Upload or capture ECG.

- Case 2: Access recent digitised signals.
- Case 3: Open settings for scale and export configuration.

**Pre-processing Screen**

- Case 1: Apply automatic grid removal. (maybe remove in other iterations - can be a base)
- Case 2: Apply artifact filtering.
- Case 3: Select ECG region of interest or entire ECG.

**Signal Extraction Screen**

- Case 1: View automatically extracted waveform.
- Case 2: Manually adjust trace (add/remove points).
- Case 3: Re-run extraction with different thresholds.

**Signal Reconstruction & Visualisation**

- Case 1: View digitised ECG in interactive graph.
- Case 2: Compare digitised vs raw waveform overlay.
- Case 3: Adjust scaling (25 mm/s, 10 mm/mV).

**Export Screen**

- Case 1: Save waveform as CSV.
- Case 2: Export as PDF overlay.
- Case 3: Share via cloud/EHR system.

## 3.6 Validation, Benchmarking, and Optimisation

## 3.7 Evaluation

# Chapter 4

# Application Development

The Design is, as the name suggests, about the prototype or system you designed in order to achieve investigation or development goals of your research objective. The Design chapter is something that is typically found in engineering theses, hence our inclusion of that chapter. The scope and complexity of this chapter (or associated design chapters) depends on the level of the project: obviously a BSc final year project is going to be smaller scale and less complicated than a MSc project.

Commonly, systems that are built nowadays, and this relates especially to computer engineering or mechatronics types projects (but is relevant to many electrical engineering project too), involve multiple aspects. Typically: (a) the System Level; (b) the Hardware Level and (c) the Software Level. In addition, there may be considerations for the environment and/or 'test rigs' (i.e., the infrastructure that may need to be set up around the system under test in order to perform the testing, and the test rig may in itself be a complicated system that needs thorough explanation.)

## 4.1 System Design

As mentioned, the system you are designing may have multiple parts, both of the prototype and its surrounding test rig (you could call this 'System Level Design' if you prefer, or something more accurate for your particular project). The system level section of the design aims to explain what these big pieces or subsystems are that you are going to develop. Often, for embedded systems particularly, the design is divided into a front-end and a back-end.

The front-end provides the point of interaction with other systems and/or the user. A graphic user interface (GUI) may be part of the front-end (depending on the design) ... or the front-end might be signal conditioning and sampling electronics that then feeds into a front-end processor (e.g., FPGA) and further on into the system (e.g., towards back-end processing stages and storage). The user interface and GUI might be more in the back-end in some designs, e.g., a website services which the user or other programs connect to.

*Note:* It is usually imperative to have a clear and easy-to-follow diagram (e.g., Fig. 4.1) to illustrate the system level and to refer to in your explanations for this section.

The sections that follow the System Level depends on what your system involves. We have provided an example here of a system that has some Hardware Level aspects, some Software Level aspects and considerations for Integration (in this case setting up a test rig).

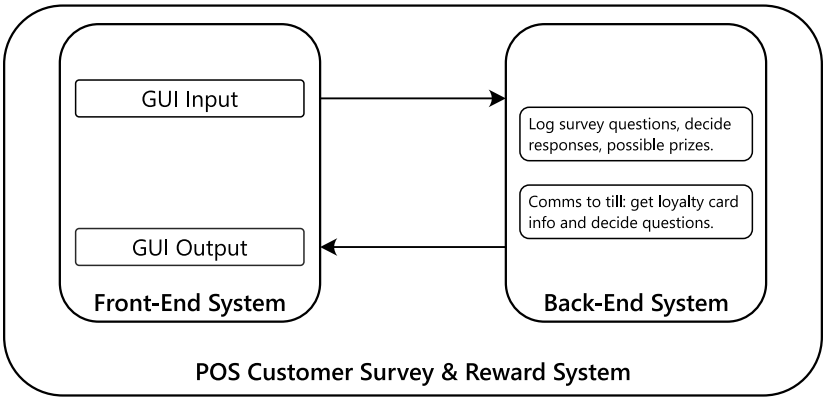


Figure 4.1: Example system level design illustration

## 4.2 Hardware Design

The Hardware Design sections include significant details on your hardware design, PCB considerations, hardware interfacing and connections and power considerations, among other aspects that are specific to the hardware concerns of your prototype or system being built. By 'significant details' it is suggested that you do not need to go into excessive minutae of the design – if you are building a significant piece of hardware largely from scratch, then you probably need a good amount of details to explain your choices etc. You can also use an appendix in which to park information that may be providing extraneous detail that you think is nevertheless needed but is causing the write-up to become too bulky.

*Note:* Even if your project is entirely software, you may still want to have a Hardware section to explain what platform and related components you were using; such information can help others to recreate your experiments, which are a desirable property of a good thesis. If you are doing software performance tests you would need to provide characteristics of the platform, thus another reason for having a hardware section (but if the hardware section is just there for platform specs, then you can

keep the section pretty short, likely not needing more than a page).

It's generally a good idea to include a block diagram and or schematics at this point. Do not simply have a text-heavy discussion of what parts were used with a detailed schematic and photo of the hardware device that was built (doing so would offload the explanations and logical progression from design to PCB to the examiner to figure out, which is certainly not an advisable approach even if it saves much space).

A representative block diagram, which provides a clear explanation of a specific piece of a design, is presented in Fig. 4.2. This figure was drawn in InkScape [21]. When you want to import an InkScape figure (SVG format) into LaTeX, simply save it to PDF (use the drawing extents as the media box area) and include the figure. This template includes a `'make figures'` target meant for this purpose.

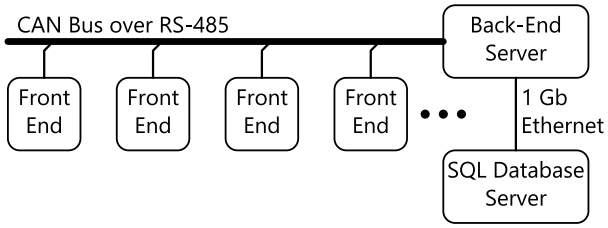


Figure 4.2: Example hardware level design illustration

## 4.3 Software Design

The software design section should go from the high-level design aspects, using for example block diagrams or UML class diagrams to explain the main parts of the system, and then going into details of specific operations or algorithms using some or a combination of, for example, pseudocode, UML state charts, UML activity diagrams, flow charts, or other appropriate figures to help the explanations.

You might decide to have some actual code (usually not more than code snippets, i.e., not whole programs) in the Software section, or you might decide to put such details into an implementation section (since the code is something that carries the design into an instantiated implementation).

Things you may have in the Software section include the following:

- Software designs drawings (e.g., block diagrams, UML diagrams, etc.)
- Algorithms (maybe in pseudocode, or actual code such as MATLAB)
- Code snippets (where relevant, used to illustrate how you went from algorithm, or an element of the software design, to executable implemented code)
- Implementation and development methods (for example specific software tools that had to be installed, scripts to run, parameters to use; but remember that some of this particular item may be better placed in the methodology – particularly if it relates to choices that were made earlier on or even before development started.)

## 4.4 Implementation

For a project that is largely hardware based, the implementation section is sometimes rather short, providing photos of the system and explaining some tips and methods on how it was put together (e.g., solutions that were learned for how to solder on parts effectively, or through the implementation experience realising parts that need to be handled with special care etc.) For a project that involves hardware and software, this section could include both tips on getting the hardware together, as well as details about implementing the software.

## 4.5 Integration or Test Rig

Some research projects require the development of surrounding infrastructure or a suitably conditioned or prepared environment in order to carry out the testing. This may involve developing some sort of test rig into which the prototype is placed or coupled so that testing can be performed on it. As a simple example, consider a vibration measuring device. If you want to test it in the lab, which has concrete floors but you want to test it for a range of flooring types, it may be necessary to build one or more test rigs that will provide the needed characteristics in order to test the product in a sufficiently authentic situation.

The integration section may alternatively, or in addition to the above point, explain how different subsystems of the system constructed are connected up. For example, this section might be used to explain the different ways to connect up a system that combines some software on a PC, a complicated DSP platform, and perhaps separate front-end conditioning circuitry, in order to complete experiments to test the achievement of different sub-objectives of the project.

# Chapter 5

# Results

## 5.1 Results

- Describe the experimental setup (ie. Hardware)
- Describe your experiments
- Describe your results

This chapter should generally present your finding in the form of figures, tables, equations, code listings, etc. It is for presenting and discussing your findings, which can split into sections if the experiment has multiple parts, or stages. You might also want to have a 'Discussion' or 'Discussion of Results' chapter, which may focus on either more detailed discussion of particular results, or more comprehensive discussion of the results and system performance as a whole. If you have a Discussion of Results section, then you do not want to discuss too much about the specific results

in this chapter and rather move the main discussion or reflective considerations of these to the Discussion of Results chapter.

However, and this is an important reminder, ensure that you do have some text of discussion for your results, to take the reader through your results and the figures you may provide. Make sure not to just put one figure after another without any attempt to explain the sequencing and what is being shown and perhaps some key issues in the figures presented (a monolithic sequence of figure after figure without any attempt at explanation will get undesirable responses from examiners).

If you have an experiments section, it is often useful to have a clear connection of experiment section corresponding to a result section showing results for that test. Examiners often appreciate that sort of clear and consistent structure that is easy to follow. For example, if you have section 5.1 as Modular Testing, then you could correspond to 6.1 Modular Testing Results (the two sections should be cross-linked with `\label` and `\ref`, in both directions)

## 5.2 Tips for Results Figures

Include good quality graphs (see Fig. 5.1 for an example).

An easy way to obtain more space for an article is to use wide, flat figures, such as Fig. 5.2.

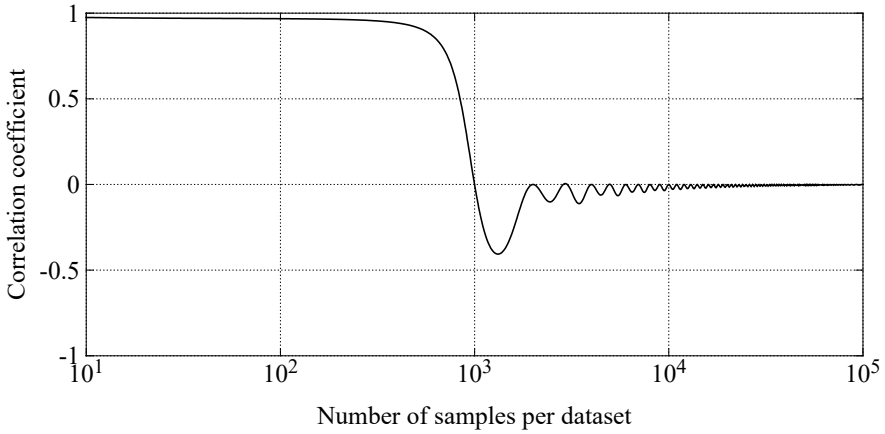


Figure 5.1: The correlation coefficient as a function of sample count

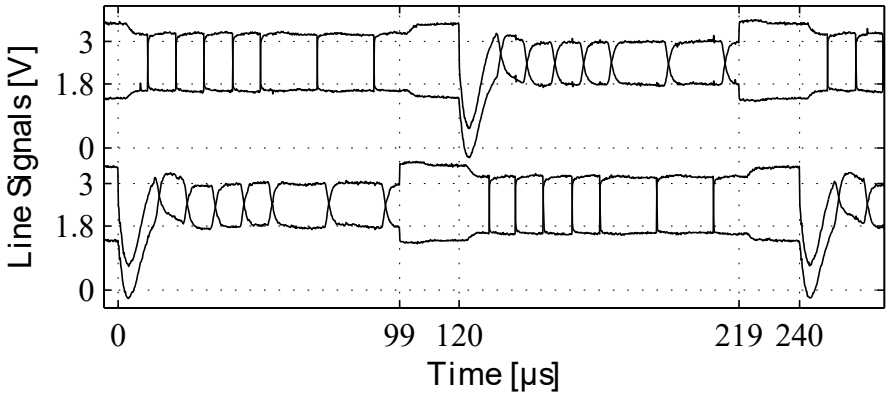


Figure 5.2: Oscilloscope measurement showing physical line signals on both ends of a transmission line during master switch-over [22]

## 5.3 Pictures and screenshots

When you include screenshots, pdfLaTeX supports JPG and PNG file formats. PNG is preferred for screenshots, as it is a loss-less format. JPG is preferred for photos, as it results in a smaller file size. It's generally a good idea to resize photos (not screenshots) to be no more that 300 dpi, in order to reduce file size. **Never change**

**the aspect ratio of screenshots and pictures!**

Fig. 5.3 shows an example of a PDF image with custom scaling.


Figure 5.3: An example image with custom scaling

Make sure to always use the best quality image possible. Use JPEG for photos, PNG for screenshots and PDF (scalable vector graphics) for everything else. JPEG is lossy, but good for photos, whereas PNG is lossless and good for images with large areas of solid colour.


(a) JPEG


(b) PNG


(c) SVG

# Chapter 6

# Conclusions

## 6.1 Conclusions

- Restate the problem. State the novel solution.
- Summarize what has been accomplished
- Summarize any limitations
- What worked really well and has a big impact?

## 6.2 Future Work

- How do you hope to continue work on this topic?
- Are there possible extensions?
- What are some improvements that could be made?

# Bibliography

[1] M. AlGhatrif and J. Lindsay, "A brief review: history to understand fundamentals of electrocardiography," *Journal of Community Hospital Internal Medicine Perspectives*, vol. 2, no. 1, Apr. 2012.

[2] V. Medved, S. Medved, and I. Kovac, "Critical appraisal of surface electromyography (semg) as a taught subject and clinical tool in medicine and kinesiology," *Frontiers in Neurology*, vol. 11, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:225062621

[3] J. Heaney, J. M. Buick, M. U. Hadi, and N. Soin, "Internet of things-based ecg and vitals healthcare monitoring system," *Micromachines*, vol. 13, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254440129

[4] B. Rubbo, N. K. Fitzpatrick, S. C. Denaxas, M. Daskalopoulou, N. Yu, R. S. Patel, and H. Hemingway, "Use of electronic health records to ascertain, validate and phenotype acute myocardial infarction: A systematic review and recommendations." *International journal of cardiology*, vol. 187, pp. 705–11, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:13961345

[5] J. Qiu, J. Zhu, S. Liu, W. J. Han, J. Zhang, C. Duan, M. Rosenberg, E. Liu, D. Weber, and D. Zhao, "Automated cardiovascular record retrieval

by multimodal learning between electrocardiogram and clinical report," in *ML4H@NeurIPS*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:265302359

[6] L. Ravichandran, C. Harless, A. J. Shah, C. A. Wick, J. H. McClellan, and S. Tridandapani, "Novel tool for complete digitization of paper electrocardiography data," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 1, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:1664058

[7] Z. J. Eapen *et al.*, "Defining a mobile health roadmap for cardiovascular care integration," *Journal of the American Heart Association*, vol. 5, no. 11, 2016.

[8] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado, and J. Herreros-González, "Mobile apps in cardiology: Review," *JMIR mHealth and uHealth*, vol. 1, no. 2, p. e15, 2013.

[9] H. Wu, K. H. K. Patel, X. Li, B. Zhang, C. Galazis, N. Bajaj, A. Sau, X. Shi, L. Sun, Y. Tao, H. Al-Qaysi, L. Tarusan, N. Yasmin, N. Grewal, G. Kapoor, J. W. Waks, D. B. Kramer, N. S. Peters, and F. S. Ng, "A fully-automated paper ecg digitisation algorithm using deep learning," *Scientific Reports*, vol. 12, 2022.

[10] Y. Liu, A. A. Abbasi, A. Aghaei, A. Abbasi, A. Mosavi, S. Shamshirband, and M. A. A. Al-qaness, "A mobile cloud-based ehealth scheme," *arXiv preprint arXiv:2004.11842*, 2020, preprint. [Online]. Available: https://arxiv.org/abs/2004.11842

[11] S. R. Steinhubl, E. D. Muse, and E. J. Topol, "Can mobile health technologies transform health care?" *Journal of the American Medical Association (JAMA)*, vol. 310, no. 22, pp. 2395–2396, 2013.

[12] M. V. McConnell *et al.*, "Mobile health advances in physical activity, fitness, and cardiovascular health," *Journal of the American College of Cardiology*, vol. 71, no. 23, pp. 2693–2704, 2018.

[13] Osmosis from Elsevier, "Electrocardiography (ecg/ekg) – basics," https://www.youtube.com/watch?v=xIZQRjkwV9Q, 2025, youTube video.

[14] Ignite Healthwise, LLC Staff, "Sa node and av node," https://myhealth.alberta.ca/Health/pages/conditions.aspx?hwid=sts14215, MyHealth.Alberta.ca, 2024, current as of October 24, 2024.

[15] C. O'Reilly, S. D. R. Oruganti, D. Tilwani, and J. Bradshaw, "Model-driven analysis of ecg using reinforcement learning," *Bioengineering*, vol. 10, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259268489

[16] J. O. Prima, B. S. Pamungkas, Nugraha, and Suprijanto, "Polyaniline as novel polymer materials for dry electrode- based electrocardiography (ecg)," *Jurnal Elektronika dan Telekomunikasi*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:139344133

[17] A. Sinha, T. Aljrees, S. K. Pandey, A. Kumar, P. Banerjee, B. Kumar, K. U. Singh, T. Singh, and P. Jha, "Semi-supervised clustering-based dana algorithm for data gathering and disease detection in healthcare wireless sensor networks (wsn)," *Sensors (Basel, Switzerland)*, vol. 24, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266381207

[18] S. Mishra, G. Khatwani, R. Patil, D. Sapariya, V. Shah, D. Parmar, S. Dinesh, P. Daphal, and N. Mehendale, "Ecg paper record digitization and diagnosis using deep learning," *Journal of Medical and Biological Engineering*, vol. 41, no. 4, pp. 422–432, Aug. 2021. [Online]. Available: https://doi.org/10.1007/s40846-021-00632-0

[19] Y. Li, Q. Qu, M. Wang, L. Yu, J. Wang, L. Shen, and K. He, "Deep learning for digitizing highly noisy paper-based ecg records," *Computers in Biology and Medicine*, vol. 127, p. 104077, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001048252030408X

[20] H. Tun, W. K. Moe, and Z. M. Naing, "Analysis on conversion process from paper record ecg to computer based ecg," *Applied Bionics and Biomechanics*, vol. 1, pp. 69–81, sep 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:196033483

[21] "InkScape Website," http://www.inkscape.org/.

[22] J. Taylor and J. G. Hoole, "Robust Protocol for Sending Synchronisation Pulse and RS-232 Communication over Single Low Quality Twisted Pair Cable," in *Proceeding of ICIT*. Taiwan: IEEE, Mar. 2016.

[23] "xkcd: Thesis defense," https://xkcd.com/1403/, (Accessed on 10/20/2017).

[24] "Big bang - warm kitty, soft kitty (sheldon's lullaby sick song) instrumental version lyrics — metrolyrics," http://www.metrolyrics.com/warm-kitty-soft-kitty-sheldons-lullaby-sick-song-instrumental-version-lyrics-big-bang.html?ModPagespeed=noscript, (Accessed on 10/20/2017).

[25] M. Shaw, "Writing good software engineering research papers," in *Software Engineering, 2003. Proceedings. 25th International Conference on*. IEEE, 2003, pp. 726–736.

[26] B. Paltridge, "Thesis and dissertation writing: an examination of published advice and actual practice," *English for Specific Purposes*, vol. 21, no. 2, pp. 125–143, 2002.

[27] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado, and B. Sainz-De-Abajo, "Mobile apps in cardiology: review," *JMIR mHealth and uHealth*, vol. 2, no. 1, p. e1, 2014.

[28] U. Utsha *et al.*, "A smartphone application for beat-by-beat ecg signal (visualization)," *Unspecified (science direct abstract)*, 2024, article focused on a smart-health app for ECG visualization and monitoring. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2352648324000011

[29] Interactive Biology, "The ultimate cardiac cycle video – most comprehensive on youtube!" https://www.youtube.com/watch?v=toZI5997hz8, 2025, youTube video.

[30] U. Eco, *How to write a thesis*. MIT Press, 2015.

[31] I. Graessler, J. Hentze, and T. Bruckmann, "V-models for interdisciplinary systems engineering," in *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, 2018, pp. 747–756.

[32] W. Royce, "The Software Lifecycle Model (Waterfall Model)," in *Proceedings of WESTCON*, Aug. 1070.

[33] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, May 1988.

[34] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl, "The Not So Short Introduction to LaTeX $2_\varepsilon$," https://tobi.oetiker.ch/lshort/lshort.pdf, Jul. 2015, version 5.05.

[35] "IEEE Conference Paper Templates," http://www.ieee.org/conferences_events/conferences/publishing/templates.html.

[36] A. Baboon, B. Charles, D. Ester, and F. Generalson, "An Amazing Title," Their Not-so-awesome University, Technical Report, Apr. 1492.

[37] B. van der Zander, J. Sundermeyer, and T. Hoffmann, "TeXstudio – A LaTeX Editor," https://www.texstudio.org/.

[38] "Voluntary Page and Overlength Article Charges," http://www.ieee.org/advertisement/2012vpcopc.pdf, 2014.

# Appendix A

# Supporting Data

Appendix sections are where you can place large figures, data tables, and spinets of code. Use appendices to your benefit to keep the body of your thesis concise!

The lyrics found below are for your enjoyment, but also serve an important role in demonstrating latex syntax for formatting text and in-text citations.

## A.1 Lyrics to Soft Kitty

Soft kitty, warm kitty
Little ball of fur
Happy kitty, sleepy kitty
Purr, purr, purr

This has been brought to you by Sheldon Cooper [24]

# Appendix B

# Satirical Support

This section provides some comic relief. In addition, it serves as an example of how to insert an image into your thesis with proper caption, label and citation.


Figure B.1: You must prepare to defend your thesis [23]