BSCS-S21-004

03-134212-042 Laiba Tahir

03-134211-037 Rizwan Shahid

Predicting Arrhythmia

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: Dawood Akram

Department of Computer Sciences

Bahria University, Lahore Campus

December 2024

Certificate

We accept the work contained in the report titled,

"Predicting Arrhythmia"

written by

Laiba Tahir

Rizwan Shahid

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Dawood Akram

(Signature)

December 05, 2024

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134212-042	Laiba Tahir	
03-134211-037	Rizwan Shahid	

Date: December 05, 2024

Specially dedicated to

All our teachers, our parents, and especially our supervisor, **Dawood Akram**, whose invaluable guidance and unwavering support have been pivotal throughout the project.

Laiba Tahir

Rizwan Shahid

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our research supervisor, **Mr. Dawood Akram** for his invaluable advice, guidance, and his enormous patience throughout the development of the research.

Laiba Tahir

Rizwan Shahid

Predicting Arrhythmia

ABSTRACT

One of the many arrhythmia management advices is that diagnosis will always play an important role, but it is not possible because a person may have suffered from an arrhythmia without being aware of it; hence this is somewhat broad in terms of cardiovascular misunderstandings. Currently, this diagnosis is not going to be very accurate or far to find, thus opening much space for improvement from a medical point of view. This is the entire platform that we proposed for automatic arrhythmia classification and prediction through applied techniques of deep learning for several electrocardiogram image databases. What we would also want to take advantage here is the recent deep learning methods for extracting extremely interesting features from ECG images, thus allowing a differentiation between different types of arrhythmias.

The end product of this research study would be an easy-to-use mobile application, which will include real-time ECG analysis through trained deep learning models. The main users are people who will have access to an advanced application that would allow the taking of pictures through smartphone cameras and instantaneously analysis, thus enhancing accessibility to cardiac health care for all. This proposed system will achieve early detection, timely intervention, and reduced burden on the healthcare system. Incorporating advanced deep learning models and mobile technology into this project would prove to be a great achievement that could revolutionize cardiac diagnostics for patients and health providers alike.

TABLE OF CONTENTS

DECLARATION	**iii**
ACKNOWLEDGEMENTS	**v**
ABSTRACT	**vi**
TABLE OF CONTENTS	**vii**
LIST OF TABLES	**x**
LIST OF FIGURES	**xi**
LIST OF SYMBOLS / ABBREVIATIONS	**xii**

CHAPTERS

1	**INTRODUCTION**			**1**
	1.1	Background		1
	1.2	Problem Statement		2
	1.3	Aims and Objectives		3
	1.4	Scope of Project		3
2	**LITERATURE REVIEW (and/or SRS)**			**4**
	2.1	Overview		4
		2.1.1	Current Technologies for Arrhythmia Prediction	5
		2.1.2	Limitation of Traditional Systems	5
		2.1.3	Advancements with Deep Learning	6
	2.2	Deep Learning Models for Arrhythmia Prediction		6
		2.2.1	Convolutional Neural Networks (CNNs)	6
		2.2.2	Recurrent Neural Network (RNN)	7
	2.3	Data Challenges		8
	2.4	Gaps in Research		8

3	**DESIGN AND METHODOLOGY**		**9**
	3.1	System Design and Architecture	9
		3.1.2 Interaction between System Components	12
	3.2	Data Collection	13
		3.2.1 Dataset Overview	13
		3.2.2 Preprocessing of Images	14
	3.3	Framework Selection	15
	3.4	System Design Artefacts	15
		3.4.1 ERD Diagram	16
		3.4.2 DFD Diagram	17
		3.4.3 Sequence Diagram	18
		3.4.4 Class Diagram	19
4	**DATA AND EXPERIMENTS (and/or IMPLMENTATION)**		**20**
	4.1	Model Development	20
		4.1.1 Dataset Preparation	20
		4.1.2 Dataset Splitting	20
		4.1.3 Model Development	21
		4.1.4 Training Model	22
		4.1.5 Evaluation Model	22
		4.1.6 Deployment and Usage	23
	4.2	Model Comparison	23
5	**RESULTS AND DISCUSSIONS (or USER MANUAL)**		**24**
	5.1	Getting Started	24
	5.2	Users Access, Roles and Privileges	24
	5.3	User Interface	26
		5.3.1 Splash Screen 1	26
		5.3.2 Splash Screen 2	27
		5.3.3 Signup Screen	28
		5.3.4 Login Screen	29
		5.3.5 Upload Image Screen	30
		5.3.6 Result Screen	31

6 CONCLUSION AND RECOMMENDATIONS 32

REFERENCES 33

APPENDICES 34

LIST OF TABLES

TABLE	TITLE	PAGE
Table 1 Augmentation		15
Table 2: Evaluation Metrices		23

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 3.1: ERD Diagram		16
Figure 3.2 DFD Diagram		17
Figure 3.3 Sequence Diagram		18
Figure 3.4 Class Diagram		19
Figure 5.1 Splash Screen 1		26
Figure 5.2 Splash Screen 2		27
Figure 5.3 Signup Screen		28
Figure 5.4 Login Screen		29
Figure 5.6 Upload Image Screen		30
Figure 5.7 Result Screen		31

LIST OF SYMBOLS / ABBREVIATIONS

FYP	Final Year Project
AI	Artificial Intelligence
CNN	Convolutional Neural Network
ECG	Electrocardiogram
KNN	K-Nearest Neighbors
MI	Myocardial Infarction
RNN	Recurrent Neural Netwo

CHAPTER 1

INTRODUCTION

1.1 Background

As with heart diseases, arrhythmias are some of the most predisposing causes of mortality across the globe. Their early and accurate detection is highly crucial in relieving and managing the patient and in improving prognosis. However, the current diagnostics for arrhythmias are entirely manual expert dependent and lack computer-program integration. Such methods are time-consuming, can be have errors because of human flaws, and are inaccessible at times in the more peripheral areas; thus, creating a disparity in health care.

This innovation in artificial intelligence, primarily through deep learning, has disrupted the medical landscape through effective means of interpretation of complicated medical data. Usually, deep learning models outperform others when applied to vast and complicated datasets such as electrocardiograms: much subtle patterns and anomalies otherwise undetectable by the human eye become visible. These facilitate the development of ultra-reliable, accurate arrangement, thus paving the way for improved and efficient diagnostic pathways. It promises the automation of all the systems in health care and final reach to advanced diagnostics for destitute areas.

This study will focus on the utilization of latest advanced deep learning techniques in the development of a more robust arrhythmia detection system which can apply algorithms using a mobile application that is user-friendly. Hence, it bridges the difference between very expensive diagnostic tools to practically everyday access for patients. Such an application will allow real-time monitoring of the heart health of users and empowers them by creating awareness regarding their cardiac health concerning early medical intervention.

1.2 Problem Statement

Currently, systems for diagnosis ask for early elective detection of arrhythmias with the accuracy, access, and efficiency it requires. Classical methods of diagnosis relied much on human interpretation. It is time- consuming and may involve professional bias which ruins its reliability especially in difficult cases. Such methods do not scale up and do not really favour time-to-evaluation and time-to-diagnosis, especially in remote or resource-poor and limited sites. As such, it underscores a very urgent need for automation in the provision of accurate and real-time arrhythmia detection. These waiting periods, usually very protracted, for many diagnoses create demand on the health care system, produce very serious health complications, and put further pressure on already overstretched health services.

The long waiting times that patients experience in getting a diagnosis worsen health complications and also put enormous pressure on the already overburdened health care systems. So critical is the phrase in establishing such time limits, often regarding major illnesses, that it could even mean the balance between life and death if done at the right time. There is increasing injustice within health care delivery that without making diagnostics cheaper and more widely available will continue to exacerbate health disparities and keep a sizeable share of the underprivileged without any care for their hearts.

Such approaches will deny real-time trustworthy arrhythmia detection. They would empower people to have their cardiac evaluation on site instead of depending on overburdened health care facilities and would provide rapid diagnosis and initiation of treatment. They indeed have to be scaled up and made accessible toward turning the development of cardiac care inside out: bringing effective and timely diagnostics into the reach of patients, regardless of their geographical location. Such transformation would not only release a considerable burden off the healthcare systems but also enhance their quality.

1.3 Aims and Objectives

The objectives of the thesis are shown as following:

i) To develop an extensive system for the automatic recognition of arrhythmia from an ECG image using deep learning techniques.
ii) To create a mobile application that incorporates trained deep learning models that allow ECG analysis on real time.
iii) To enable access to arrhythmia detection by allowing users to capture and upload ECG images using their smartphone camera for instant diagnosis.
iv) To extend early detection and intervention to patients and healthcare practitioners by equipping them with a reliable diagnostic tool.

1.4 Scope of Project

This project is aimed at developing a platform for algorithm application through deep learning technology and models for arrhythmia detection and classification based on ECG images. It provides real-time processing of ECGs and thus valuable diagnostic information for patients and health professionals' lifestyles. It assists in closing the gaps in cardiac healthcare through an earlier and easier diagnosis and makes it more accessible to patients of different geographical areas, especially the underserved ones.

CHAPTER 2

LITERATURE REVIEW (and/or SRS)

2.1 Overview

It is the most important problem of arrhythmias, which cause disturbances of rhythms in the beat of heart, where arrhythmias can become one day again one of the most important basic causes of considerable health. At times, these conditions present can later result in fatal complications, for example, stroke, sudden cardiac arrest, or heart failure. This ailment afflicts millions every year round throughout the world. As such, arrhythmias account significantly for global morbidity and mortality.

Critical influence upon patient outcomes includes detecting as well as intervening with exactitude and in a timely manner so as to avert that arrhythmias pose to a patient's life as well as direct patient care in positive outcomes.

Diagnosis of arrhythmias is conventionally made by qualified professionals through interpretation of ECGs. While it has been effective in many cases, they are not absolutely fool proof. Time constraints, fatigue of the profession, and subtle nuances of the data all allow for misdiagnosis or non-diagnosis that limits the reliability of the manual interpretation. Add to that, reliance on human aptitude comes to accessibility and resource-poor or remote areas where that such trained professionals may not always be available.

The automated classifying and predicting system for arrhythmias is solution to these problems. High level algorithms with the continuous online processing and analysis improve the efficiency of work and increase the accuracy at the time of diagnosis. Continuous monitoring hence detects at a period when there is no medical supervision. This opened access to cardiac care for even the most remote population and brought quality diagnostics to underserviced populations.

Such innovations do not only serve to increase efficiency, as one may expect within the healthcare system, but also serve to have many benefits, such as improved accuracy.

2.1.1 Current Technologies for Arrhythmia Prediction

The conventional method to diagnose arrhythmia is through algorithms extensively used on ECG-derived features. These methods identified various types of abnormality as absence of continuous RR intervals, prolonged QRS durations, or lack of P-wave. Most of these are useful in precisely diagnosing what happens in arrhythmia; they don't have predictive capacity. Some are really hot under the collar when it comes to the noisier data from manual recordings or heavy with artefacts.

Deep learning techniques overcome any limitations in the previous approaches of recording raw ECG data instead of acquiring signals prior to analysis. While RNNs provide temporal dependence among successive cardiac cycles, CNNs can capture what the changes in morphology usually are in the form of ECG waveforms. Latest developments are leveraging large-scale datasets, such as the MIT-BIH and PTB-XL, to train models that claim outperforming traditional methods in terms of sensitivity, specificity, and predictive accuracy. Changes in the paradigm also denote the transition from early detection to predictive systems.

2.1.2 Limitation of Traditional Systems

Traditional systems suffer from the fact that arrhythmia detection is hobbled by rule-based frameworks and manual detection of features. Such thresholds on abnormalities, owing to their definition in terms of domain knowledge, do not serve well to detect early arrhythmias since those effect has used abnormal patterns that may be either very subtle or far too complex. Therefore, it makes the individual susceptible to greater false positives/negatives since variations among patients are not considered in the static threshold.

Again, it does not concern the major predicaments, such as the computationally inefficient manual methods, with having data sets that are of extensive size. This is because normal systems fail miserably when they are asked to scale over high-frequency data of ECG individual signals collected during long periods. Not only are they poor on prediction, they also only survive detection of arrhythmias. These loopholes in current systems have attracted serious considerations toward newer advancements such as deep learning, which, in addition to offering greater flexibility, might also increase automations in prediction functionality.

2.1.3 Advancements with Deep Learning

Deep learning has an excellent chance to replace prior methods by taking automatic feature extraction and learning directly from electrocardiogram (ECG) signals. For example, morphological abnormalities caused by irregular P waves or fragmented QRS complexes would most easily be detected using convoluted neural networks (CNN). The sequential dependencies predictive of arrhythmias can be detected by using recurrent neural networks (RNNs) and long short-term memory (LSTM) networks. These models are trained widely over considerable datasets and seem to generalize very nicely to unseen data; therefore, assessments of their predictions become more accurate and reliable.

In fact, one of the latest developments is the hybrid model, which uses CNNs for extracting spatial features and LSTMs for analysing temporal patterns. Attention-based transformer architectures have just begun to gain importance because of their ability to focus their attention on only the most relevant segments of an ECG signal. With such advances, today's deep-learning systems can easily overtake classical methods in prediction accuracy and robustness by providing state-of-the-art performances in arrhythmia prediction.

2.2 Deep Learning Models for Arrhythmia Prediction

2.2.1 Convolutional Neural Networks (CNNs)

The Convolutional Neural Network (CNN) is a promising approach for predicting arrhythmia by effectively capturing and extracting the spatial features of raw ECG signals. It applies convolutional filters on the ECG waveforms to recognize different patterns such as fragmented QRS complexes and ST-segment elevation associated with arrhythmias. A typical CNN is made up of several convolutional and pooling layers to come up with a high to low level cumulative feature for improved arrhythmia classification.

One-dimensional CNNs have been narrowed down to a single dimension and designed only for time-series data such as ECG signals, thus creating a less computational but very capable model. Morphological change detection is the most vital application for such a model; therefore, it can also be used for arrhythmia detection. Recently published works have taken this trend further by merging the CNN architecture with ensemble techniques where the model performance is enhanced by the summation of several other models to lower erroneous predictions.

2.2.2 Recurrent Neural Network (RNN)

RNNs especially those based on the LSTM architecture, have become rather inseparable in terms of temporal dependencies, especially when one looks into analysing signals from ECG. The virtues associated by this is that compared to CNNs, which are good for spatial features, LSTMs are masters at discerning sequential patterns such as those of heart rate variabilities or even changing patterns of cardiac cycles. Therefore, it is rightly used in arrhythmia predictions that usually depend much on together beating heartbeats.

They have incorporated into memory cells that long maintain the very important information of very long sequences for end results. They have hybrid architecture based uses combining LSTMS with CNN; they can be said to be supporting temporal-scale and spatial-dimension behaviour analyses of ECG signals. Such architectures have been demonstrated to perform best in such context as predicting arrhythmias, like for example the cases of atrial fibrillation and ventricular tachycardia.

2.3 Data Challenges

Deep learning models efficiency largely depended on large quality dataset but many open-access datasets such as MIT-BIH and PTB-XL fell short on varied criteria like diversity in patients and types of arrhythmias, thus limiting the scope of generalizability of any model trained. Other than most ECG signals being highly corrupted by noise and artefacts, pre-processing is an extremely labour-intensive activity for data quality.

The other large challenge is data absence, especially concerning some rare types of arrhythmias. Synthesis generation methodologies for data have been experimented on regarding their ability to generalize with the use of GANs but still fail to convince everyone that they indeed represent a real-world condition. Federated learning solves this problem by allowing decentralized training over many institutions and thus alleviating access limitations without compromising patient privacy.

2.4 Gaps in Research

There are many constraints in the present deep learning systems for the prediction of arrhythmias. Most models are either based on very strong theory in the area of classification of diseases, hence not applicable in detecting early warnings, or they do not have adequate data for training and integration of multi-modal for prediction-based learning. Another major gap is interpretability since physicians would require explanations of the models' predictions for acceptance and trust.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 System Design and Architecture

This prediction system, with its peripheral modular architecture of three interconnected components such as the Flutter Mobile Application, the Flask API for Model Inference, and Firebase Authentication, brings complete flexibility in handling most of the functionalities. All the components in a modular system execute a specialized specific function while integrating towards the accurate and real-time arrhythmia prediction system. Simplicity, performance, and usability are the core ideas building up the whole architecture, especially in resource-constrained environments.

The Flutter Mobile Application mentions the user interface system in which the users can upload their ECG records as well as visualize their predictions of anomalies. The Flask API, locally hosted, acts as the backend for taking care of the ECG data processing and interaction with the already-pre-trained Convolutional Neural Network (CNN) model. Firebase is useful only for authentication purposes because there is a need to ensure a reliable and safe login to the system without saving any kind of data in the database. All of these components together build an integrated system of arrhythmia prediction.

3.1.1.1 Mobile Application

The application is developed using flutter technology, which will augment the user mobility. With flutter, the application works as cross-platform mobile application in Android. The users can do the following tasks:

- **Uploading ECG data:** Users can upload ECG images in a specific formats.

- **Predictions:** Show features of arrhythmia investigation or found arrhythmias with their respective chances.
- **Secure Login**: Firebase authentication is used such that a user can only access the application if he or she is a registered user.

3.1.1.2 Flask API

The backend component that can interface the trained CNN model with a mobile application will be the Flask API. When the user uploads ECG images, the API will facilitate the prediction process by preprocessing the images, utilizing the trained model, and returning results in a format suitable for users. The main goal of this API is that it can ensure real-time communication effective, efficient, and accurate between the mobile application and the prediction model.

Key Features of the Flask API:

1. **CORS Support**:
   Flask-CORS is used by the API to enable cross-origin requests so that communication is not hindered between the flutter app and locally hosted Flask backend. This becomes really important when it comes to web/mobile clients that want to access API resources safely.

2. **File Handling and Uploads**:
   This is an API endpoint for uploading ECG images using /upload-image for analysis. It saves the uploaded files to a predefined uploads directory, which is created when it does not yet exist.

3. **Preprocessing**:
   Uploaded images are preprocessed before being passed to the model. Preprocessing steps include:
   - Images are converted to grayscale.
   - Images are resized to target dimensions (224x224 pixels) for CNN model.
   - Normalizing pixel values to the range [0, 1].

- Expanding dimensions to match the input shape expected by the model (batch size and channel).

4. **Prediction Logic**:
   The CNN model (sequentia_model.h5) predicts the class of the ECG image. The API uses a confidence threshold of 0.6 to determine if the prediction is reliable. If the confidence falls below this threshold, the system returns a result indicating uncertainty or an incorrect input image. Predictions are mapped to human-readable labels such as "Normal," "History of MI," and "Myocardial Infarction."

5. **Error Handling**:
   The API has also been outfitted to encounter situations such as the absence of files, unsupported formats, or preprocessing failures. Then again, it also produces quite meaningful error messages to the users to assist them in troubleshooting the issues.

Workflow of the Flask API:

1. **Endpoint Definition**:
   The main prediction endpoint is /upload-image, which handles HTTP POST requests. The user uploads an ECG image through this endpoint.

2. **File Storage**:
   The uploaded file is securely saved in the upload’s directory using its original filename.

3. **Image Preprocessing**:
   - The image is read and converted to grayscale using OpenCV.
   - It is resized to match the input size of the CNN model.
   - Normalization ensures that pixel intensity values are scaled to [0, 1].
   - Batch and channel dimensions are added to match the model input.

4. **Model Inference**:
   The preprocessed image is passed to the trained CNN model. The model outputs a prediction vector representing probabilities for each class. The class with the highest probability is selected as the predicted label, provided its confidence exceeds the threshold.

5. **Result Formatting**:
   The prediction result, including the label and confidence score, is returned to the Flutter app in JSON format. If the confidence is low, the response indicates uncertainty or incorrect input.

3.1.1.3 Firebase

Firebase Authentication is the most secure form of user login implementation. Unlike other systems, this doesn't require backend provision as Firebase boasts world-class authentication for its dedicated system, and focuses solely on the detection of arrhythmias. Log in with email and password while Firebase takes care of session persistence and verification. This makes sure that only authorized users have access to the system.

3.1.2 Interaction between System Components

The interaction among components is flattened for efficiency:

- The **Flutter Mobile Application** authenticates the user via the Firebase secure interface, allowing for ECG data upload.
- The uploaded ECG data is then sent to the **Flask API**, where it is processed and input into the model for inference.
- The model, residing in the Flask environment, receives and analyses the ECG data to predict arrhythmia likelihood.
- The **Flask API** returns the prediction results through the mobile app to the user in an easy-to-understand way.

- This interaction allows for seamless communication with real-time performance as the locally hosted Flask API cuts down latency and processes the ECG data quickly.

3.2 Data Collection

Firstly, the data set used to train the model was obtained from Kaggle [9]. The data set consists of recordings of arrhythmias pertaining to a number of different types and thus forms a good basis for model training.

The dataset was subjected to augmentations to increase its diversity and generalization; for instance, flipping, addition of noise and scaling. These generated larger datasets on which a model's robustness can be determined in handling unseen data.

3.2.1 Dataset Overview

The ECG Images Dataset of Cardiac Patients is an eclectic and vast collection aimed at improving cardiovascular research and diagnosis. It is divided into four major headings: MI Patients, Abnormal Heartbeat Patients, and Patients with a history of MI, and Normal Individuals. The documents under the MI category reflect ECG patterns typical to an MI case. Abnormal Heartbeat consists of ECG data containing irregular rhythms that can be advanced towards arrhythmias or some other cardiac abnormality. History of MI can depict recovery patterns and long-term effects after a heart attack in patients who have been documented in history. Then there is the Normal category, whose ECG patterns are those of individuals having no known pathologic cardiac conditions, which can serve as comparison for the pathologic populations. The dataset-bathes into a very important resource towards cardiac health understanding, modelling diagnostic models, and improving medical research around the cardio logical sphere.

3.2.2 Preprocessing of Images

In order to accomplish proper efficiency training along with the anticipated accuracy, all the ECG images undergo categorization through a preprocessing pipeline. Thus, the model standardizes all the input data into forms amenable to deep learning models and broadens the base data to develop generalization capacity. The section talks about the complete pipeline of preprocessing which would also include portions of image augmentation.

- **Rescaling:**

  The pixel intensity values in the images of ECGs are normalized by re-scaling them using the ratio 1/255. That means the pixel intensities will be available in the range [0, 1] instead of [0, 255] for a truly clearer yet consistent data presentation on the model. Hence, this normalization shall also enable the fastest convergence of the CNN model during training by noiselessness of digit values.

- **Resizing:**

  Resizing all images to a common size of 224x224 pixels, which is specifically designed to suit the required input dimensions of a CNN model, allows straightforward processing of the data. The dimensions of all images are hence equal, with computational overhead reduced, and the models utilize memory and time efficiently when training.

- **Augmentation:**

  The purpose of augmentation is to address bias in the dataset and generalized well over the model. Random rotation between 0° to 20° was added along with a command for randomly flipping the image horizontally to incorporate the asymmetries in patterns as a modification to throw variability in the ECG directions Fluctuations in brightness have been established at 80%-120% simulating different imaging situations. It is not including vertical flipping of the image since it affects the directionality of the ECG signal.

Table 1 Augmentation

Augmentation Techniques	**Values**
rotation_range	20
horizontal_flip	True
vertical_flip	False
brightness_range	[0.8, 1.2]
preprocessing_function	None

3.3 Framework Selection

A CNN Model has been acquired for prediction of arrhythmias using image-based datasets, and the development of the model followed step ladders. Grayscale ECG images were resized and further normalized because all images feed in must be of same input size. It consisted of convolutional layers for the extraction of image features, pooling layers for spatial dimension downscaling, and fully-connected dense layers used for the classification. Dropout was applied for prevention against overfitting. Activation functions like ReLU and softmax were involved for effective learning and multi-class prediction. So, the process of systematic design leads to prediction of arrhythmia well from such images input.

3.4 System Design Artefacts

Following artefacts included in this Chapter:

1. ERD diagram
2. Sequence Diagram
3. DFD Diagram
4. Class Diagram

3.4.1 ERD Diagram

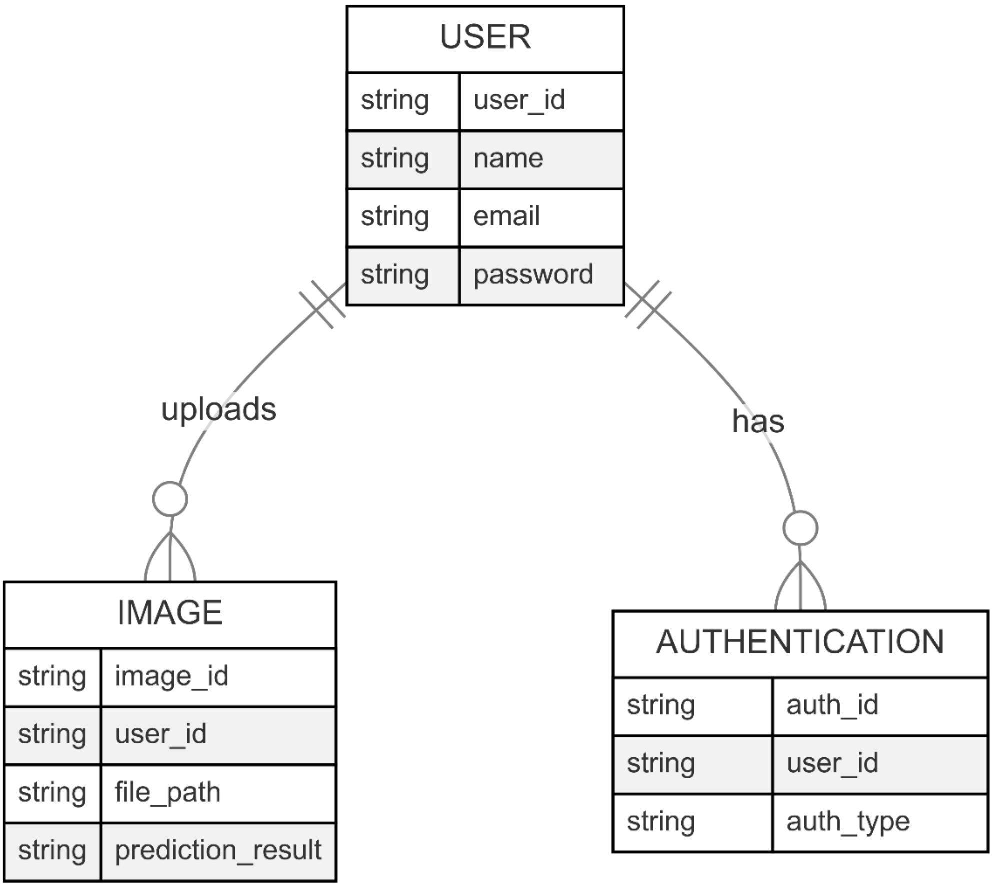


Figure 3.1: ERD Diagram

3.4.2 DFD Diagram

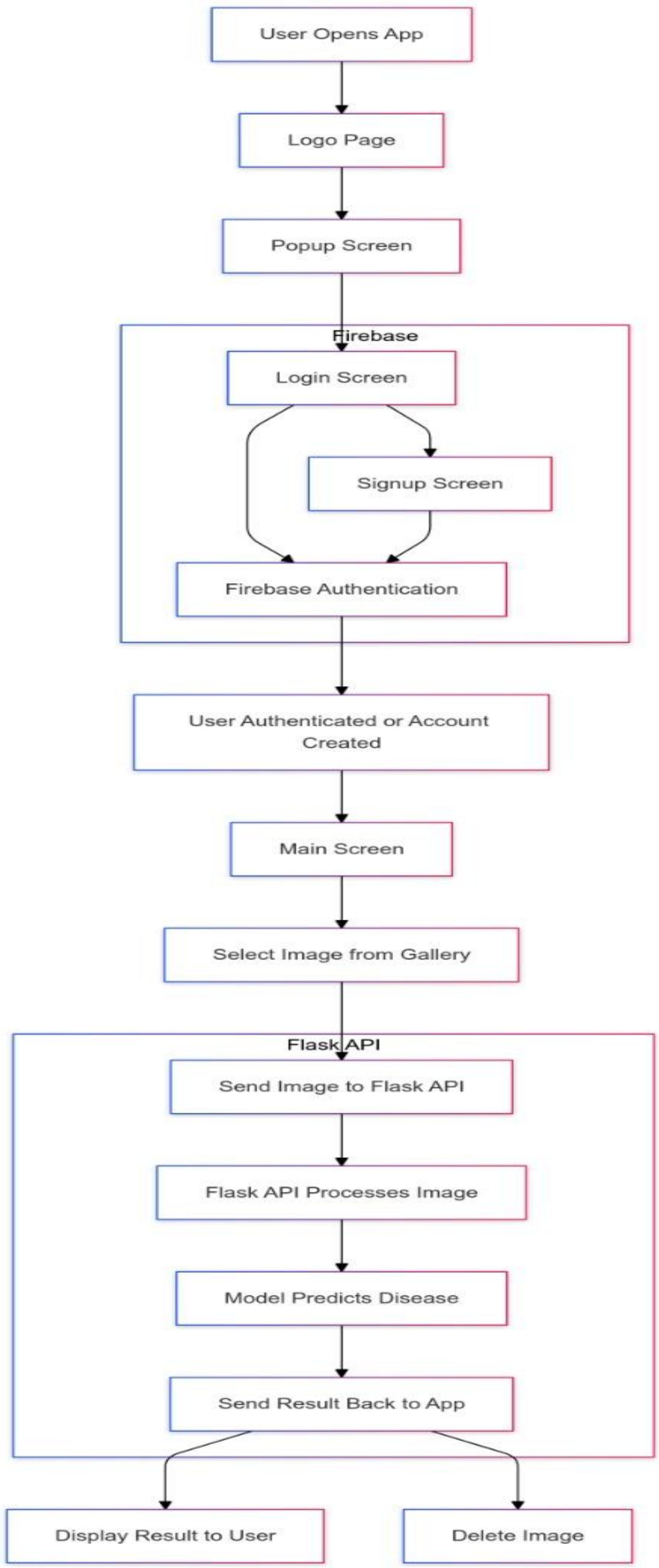


Figure 3.2 DFD Diagram

3.4.3 Sequence Diagram

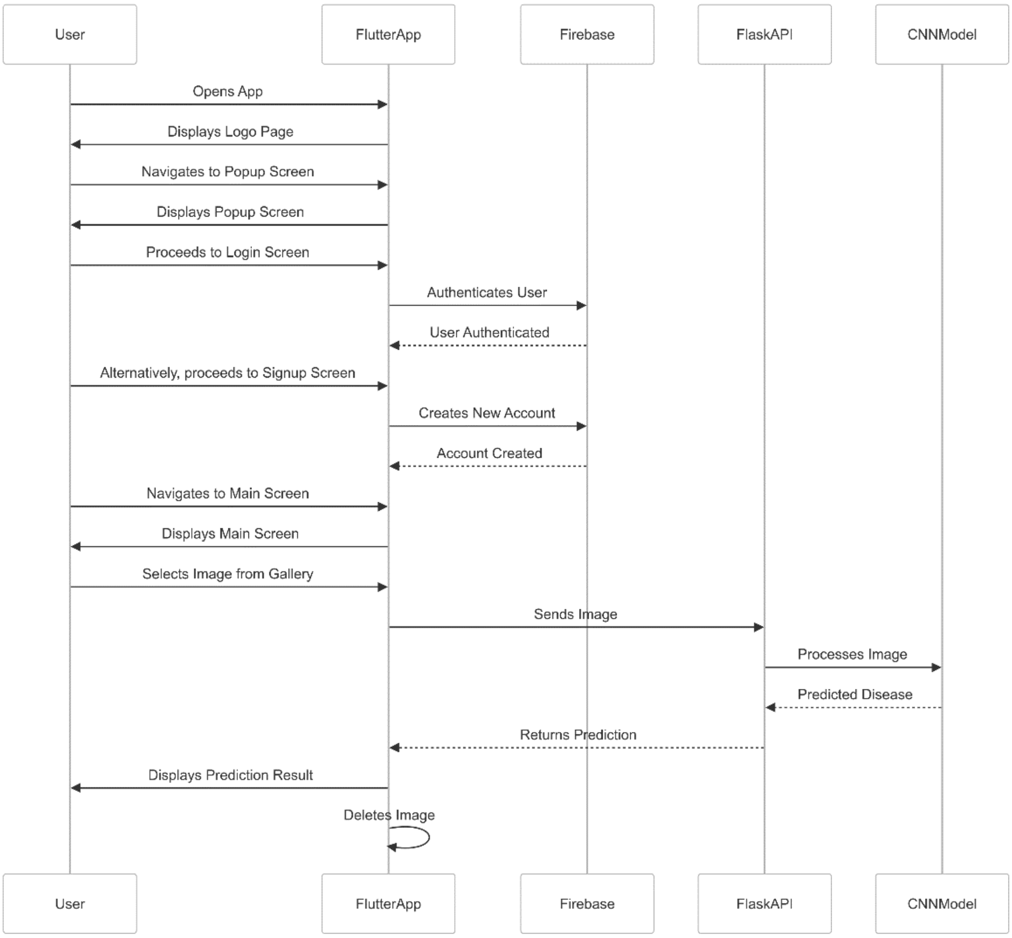


Figure 3.3 Sequence Diagram

3.4.4 Class Diagram

Upload Image Screen

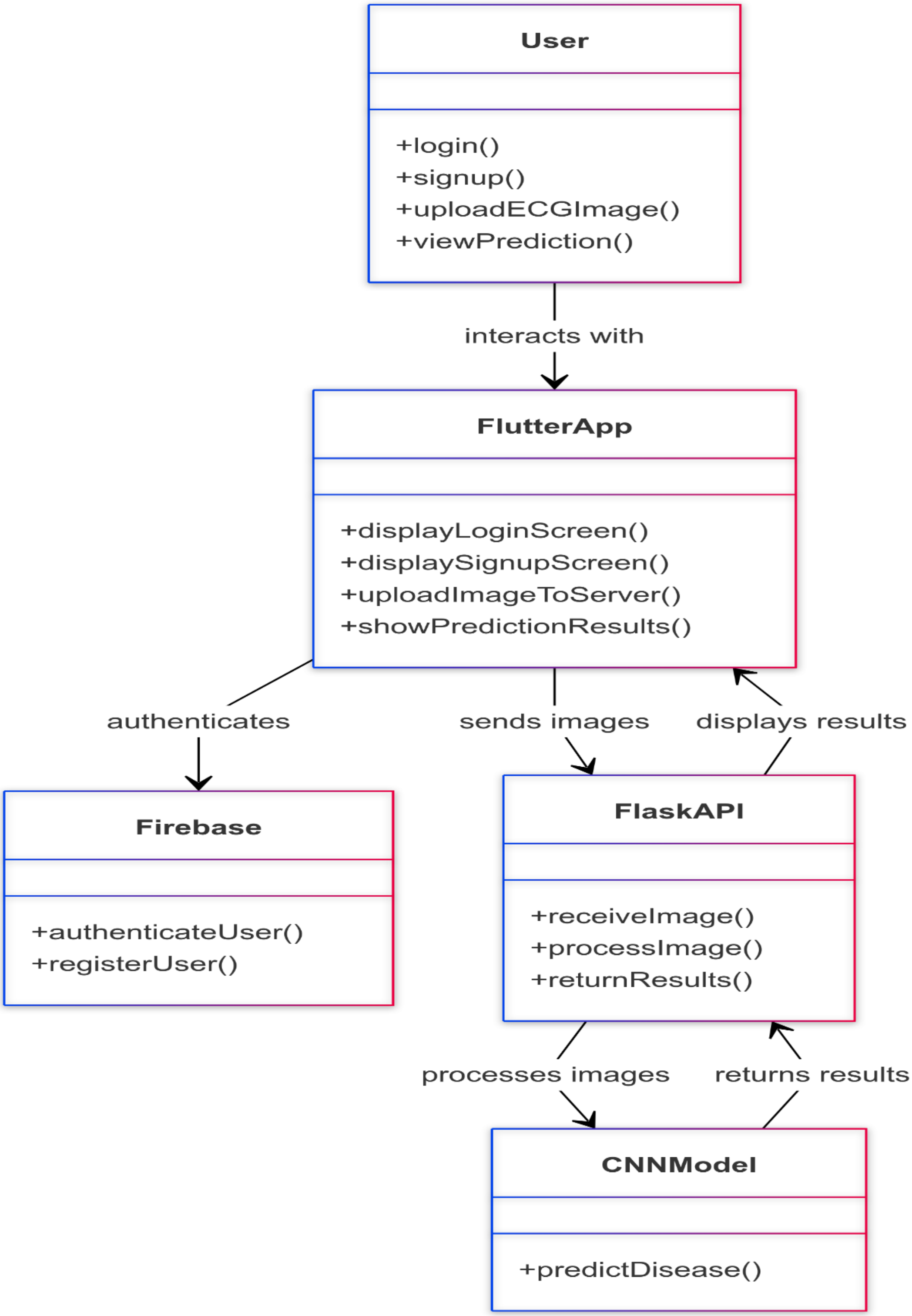


Figure 3.4 Class Diagram

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLMENTATION)

4.1 Model Development

Following is how the model is developed using different techniques.

4.1.1 Dataset Preparation

This data preparation is just a small part of the various stages one has to carry out for making the dataset optimal for training in CNN model for arrhythmia classification. First and foremost download the ECG Dataset and organize the files into subdirectories corresponding to the category e.g. Normal, Abnormal, Post Myocardial Infarction, and Myocardial Infarction. Every class can contain its independent folder; thus making them very easily manageable. Moving to handling the images, they are made uniform in dimension of 224 x 224 pixel, Grayscale to minimize complexity, and Normalization into a value of pixel between 0 and 1 to stabilize training. Furthermore, a channel dimension was appended to each picture to make its shape (224, 224, 1) so it can match the expected input format of the convolutional layer. Thorough cleaning, standardization, and data training-ready conditions have been ensured for the deep learning model.

4.1.2 Dataset Splitting

Training and evaluation of models are very effective when the dataset is partitioned into disjoint sets. Here, the main body for training is 80% and that for testing is 20% which allows for the evaluation of the model on unseen data for a more rounded assessment of performance. The training data is further divided into a validation data set, which typically occupies around 10% of the training data, for testing the trained models. This validation is checking the training of the model and is very useful for hyper parameter fine-tuning and early stopping of overfitting. Such a targeted split results in effective training, thorough evaluation with a model generalizing well under unseen conditions.

4.1.3 Model Development

We investigated a range of deep-learning architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Faster RNNs, DenseNet, and EfficientNet. The most efficient of them was CNN, which proved to be the best compromise between the computational efficiency and prediction accuracy for the arrhythmia prediction problem. Moreover, this affirms that CNNs can effectively learn spatial features from the ECG data, the reason for which we consider as the preferable model.

.

4.1.3.1 Model Architecture:

The CNN model is built with three convolutional layers specifically for the purpose of progressively extracting feature information from images.

- The first convolutional layer is made up of 32 filters, each of which has a 3x3 kernel and followed by the second layer which contains 64 filters with the same kernel size. Finally, the last convolutional layer employs 128 filters to take more complex features.
- Every convolution layer undergoes ReLU activation to impose non-linearity and enhance the level of classification accuracy.
- It is followed by a max-pooling with a kernel of 2x2 after every convolution layer to minimize the spatial size but which retains some important features.

4.1.3.2 Fully Connected Layers:

Flattening the output value from the last-convolution layer and passing through fully connected (dense) layers. The first dense layer, accordingly, has 128 neurons with activation using ReLU function. The final layer would consist of 4 neurons, where each class is represented for the arrhythmia classes: Normal, History of Myocardial Infraction (MI), Myocardial Infraction itself, and Abnormal Heartbeat. To find the understanding of each class for multiclass classification, two softmax activations are used in the output layer.

4.1.3.3 Model Compilation:

For an Adam optimizer to the model compiledizes an adaptive learning rate for the training. A learning rate of 0.001 is considered as initial and categorical cross-entropy loss type is used for multi-class classification tasks. The evaluation performance metrics such as accuracies, precisions, and recalls are monitored during training.

4.1.4 Training Model

The model has been trained with the training dataset while its performance was measured on the validation set. The model was trained for twenty epochs with the batch size of thirty-two, and then the weight at its best performance on training data were saved with the intent of preventing overfitting and increasing the likelihood that the model would generalize well on new data.

4.1.5 Evaluation Model

The model was evaluated with the help of previously unseen test datasets for competency as measured by accuracy, precision, recall, and F1-score. This has led to generalization in prediction for unseen data points. The test results reveal the much anticipated evaluation of the model in classifying ECGs into their respective defined arrhythmia categories. This will allow one to see what the company is capable of as far as true positives is concerned (recall) but with false positives kept to a minimum (precision) via high-level classification reliability. It would go on to eventually guarantee robustness and thus applicability of the model in real-life scenarios.

Table 2: Evaluation Metrices

Evaluation Metrics	Scores
Accuracy	90%
Precision	90%
Recall	90%
F1 Score	90%

4.1.6 Deployment and Usage

It classifies the new ECG images inside one of the four types arrhythmias in this instance. In a real-world application, the saved and loaded model is used.

4.2 Model Comparison

Analysing the potentials of the base models ResNet, Inception, EfficientNet, DenseNet, and VGG, along with a custom CNN for classifying myocardial infarction, associated cardiac conditions, and so on, has exposed the internal potentials of the models. While ResNet and DenseNet are very much fine for a huge set of datasets and in showing feature extraction and flowing of gradient, they can be costlier in computational expenses. Inception also catches up with the multi-scale features from different data patterns but is also costly in terms of its architectural complexity. It does well on performance and efficiency trade-offs, achieving the balanced trade between the two through optimized scaling; hence, it is suitable for resource-constrained systems. VGG, on the other hand, is already quite old concerning its parameter needs, even though it is simple and interpretable. A custom CNN that has been tailored on the basis of this small grayscale dataset should prove to be the best regarding speed, least cases of overfitting, and ease of implementation. Custom CNN perfectly suits the properties of being simple, practical, and having task-specific performance in this case, while the larger, complex models like ResNet, DenseNet, and EfficientNet are better used on larger datasets.

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 Getting Started

The Arrhythmia Predict App is a very simple application for real-time arrhythmia disease prediction based on advanced deep learning models. The user can log in securely to the application to upload any medical images and receive prediction results right away. .

To get started:

1. **Sign In**:
   - User use credentials to get logged in.
   - New users create an account using the app's simple registration process.
2. **Upload Image**:
   - Users navigate to the prediction screen and upload a medical image.
3. **Receive Results**:
   - Instant predictions are displayed in real-time.
4. **Data Confidentiality**:
   - The app processes all data during runtime and does not store user inputs or medical records.

5.2 Users Access, Roles and Privileges

The app supports role-based access with secure login for all users.

1. **User Roles**:

   - Includes healthcare professionals, researchers, and students.

- Have access to runtime prediction features after logging in.

2. **Privileges**:

    - All users can upload medical images and view results.
    - No additional privileges (e.g., data storage or history) are provided, ensuring a focused and lightweight user experience.

3. **Authentication**:

    - **Login Required**:
      Users must log in using their credentials to access the app’s features.

    - **SecureAccess**:
      Credentials are stored securely, ensuring that user accounts are protected.

5.3 User Interface

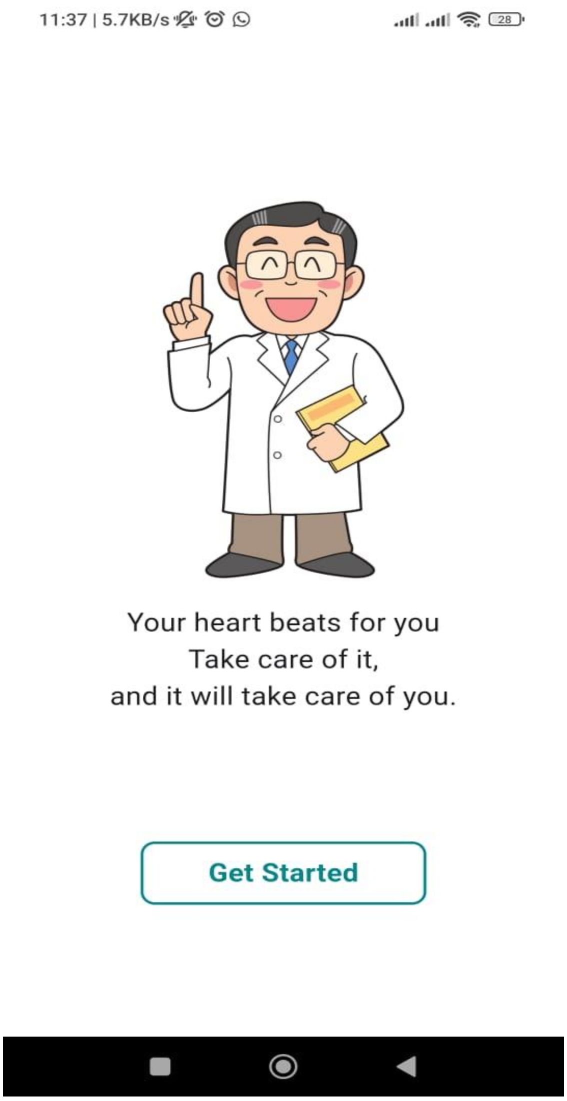
5.3.1 Splash Screen 1

After launching the app, the user will see a splash screen with app’s logo.

Figure 5.1 Splash Screen 1

5.3.2 Splash Screen 2

This screen provides an overview of why to take care of your heart.



5.3.3 Signup Screen

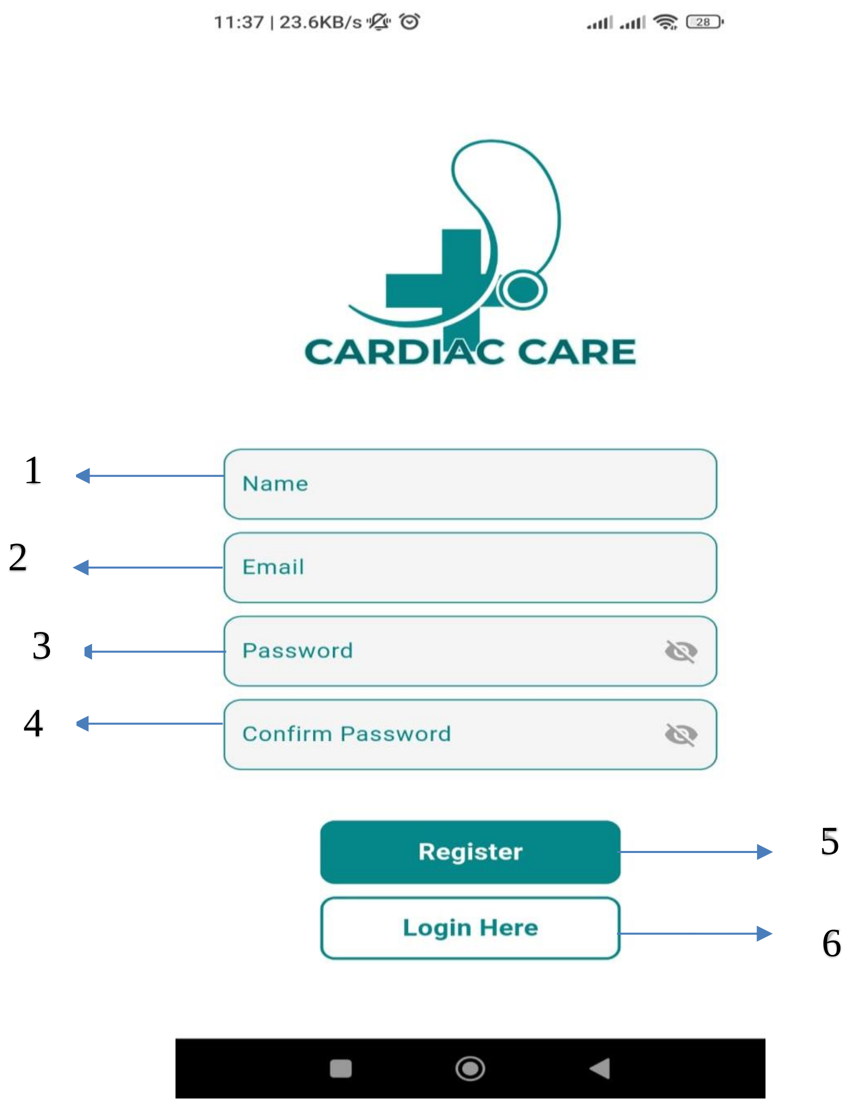


Figure 5.3 Signup Screen

1. Enter Name
2. Enter Email
3. Enter Password
4. Enter the old password uh chose to confirm password
5. Click to register your account
6. After getting registered click Login Here to get logged in

5.3.4 Login Screen

Enter your credentials to get logged in.

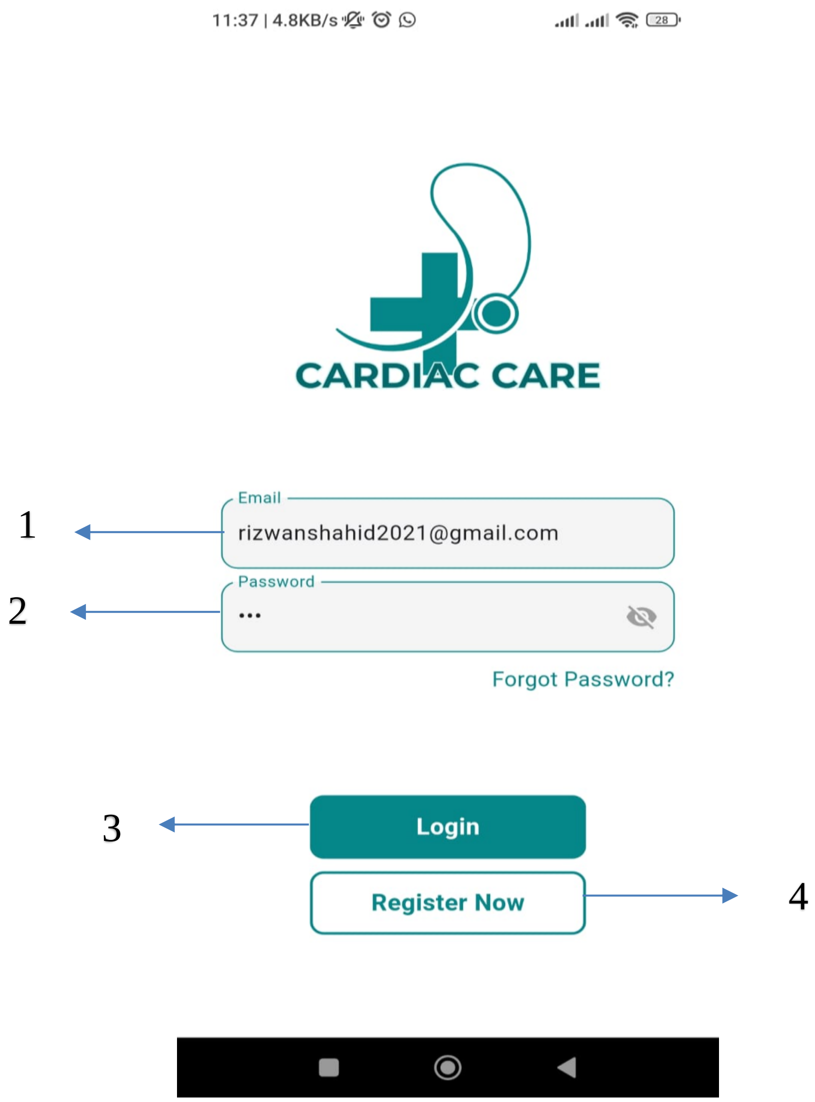


Figure 5.4 Login Screen

1. Enter Email
2. Enter Password
3. Click to get logged in to your account
4. If you don't have an account click to signup

5.3.5 Upload Image Screen

After getting logged in user will be able to upload images and get results

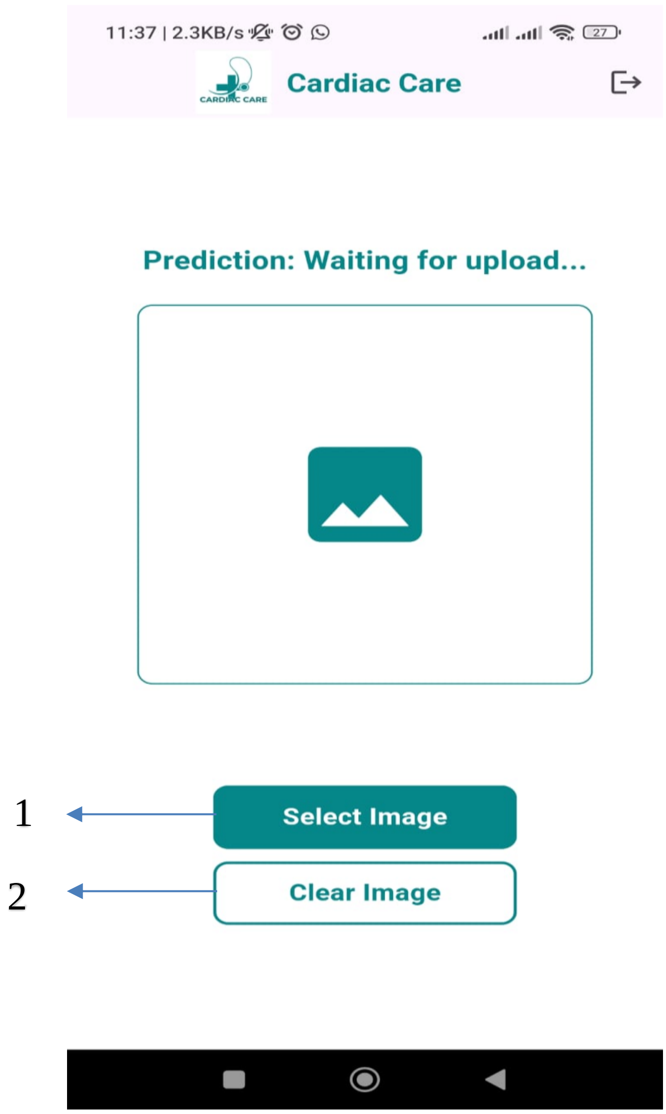


Figure 5.5 Upload Image Screen

1. Click on select to upload on image from gallery
2. Click on Clear Image to select the new image

5.3.6 Result Screen

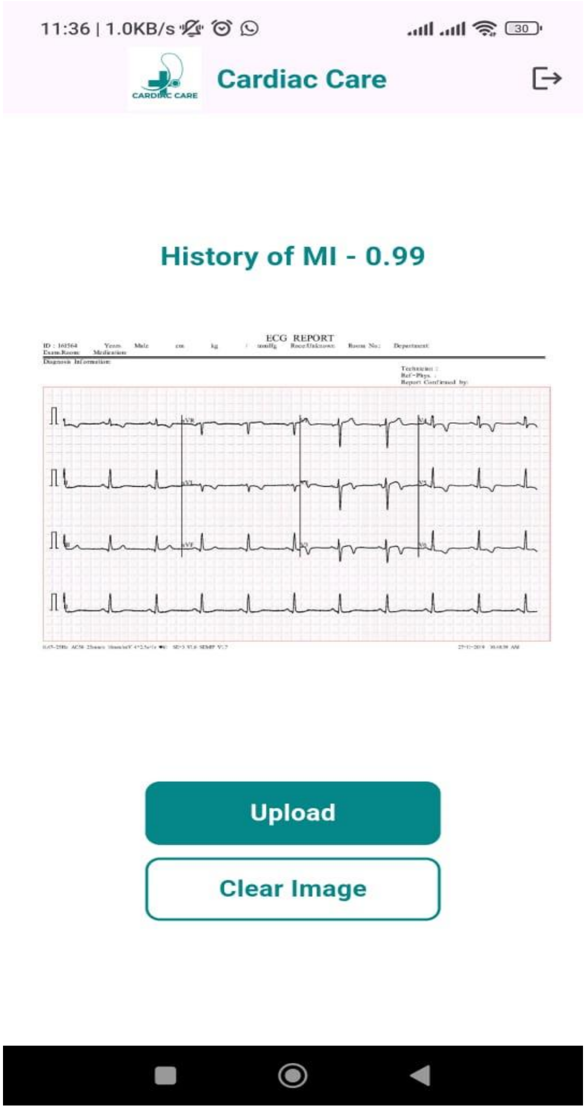


Figure 5.6 Result Screen

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

An advanced deep learning modeling system was developed for real-time arrhythmia prediction as the project objective. The entire development process emphasized usability and was functional to help the end-user have a smooth experience in making predictions.

The great systems proved very early on that they would really need to have a very efficient architecture to support predictions in real time. The application, however, was compiled with careful application of usage constraints to ensure a very simple and secure user interface.

The system has been shown to meet all of the functional and usability requirements defined through the implementation and testing stages; thus, the milestone for the project will be marked by the deployment of the application for runtime arrangement predictions. The system was reviewed and tested together with users for a better coverage of what is expected for the next version. Everything but the first set of requirements specified at the launch was, however, almost met; the rest could always be added later on for democratizing possible future applications. Improvements could involve more sophisticated prediction models and more user-friendly interfaces, or even further along a path toward being portable by associating the prediction with wearable devices.

REFERENCES

[1]. Brown, R., & Wilson, C. (2020). Machine Learning-Based Classification Algorithms for the Prediction of Coronary Heart Diseases. International Conference on Artificial Intelligence in Medicine (pp. 123-135). doi:10.5678/icaim.2020.003

[2]. Platt, J. C. (1999). Introduction to artificial neural networks. MIT Press. https://www.researchgate.net/publication/5847739_Introduction_to_artificial_neural_ networks

[3]. Zhang, L., & Wang, Q. (2019). Predicting Cardiovascular Diseases using Deep Learning Algorithms. Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 145-152. doi:10.789/bibm.2019.007

[4]. WebMD. (n.d.). Heart Disease and Cardiovascular Diseases. WebMD.https://www.webmd.com/heart-disease/guide/diseases-cardiovascular

[5].NHS.(n.d.).CardiovascularDisease.NHS.https://www.nhs.uk/conditions/cardiovascular –disease

[**6**].**World Health Organization (WHO)**. (2021). *Cardiovascular diseases (CVDs)*. Retrieved from [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)]

[7]. Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., et al. (2019). *Cardiologist-level arrhythmia detection with convolutional neural networks*. *Nature*

[8]. Murthy, S., & Singh, R. (2020). *Machine learning and deep learning models for arrhythmia detection using ECG data*. *Journal of Medical Systems*

[9]. Dataset from https://www.kaggle.com/datasets/evilspirit05/ecg-analysis/data

APPENDICES

Requirement List

This is a mobile application platform to run and execute finely at devices with:

- A camera capable of taking clear ECG images
- Running Android
- Flutter for cross-platform compatibility
- Flask or FastAPI for API integration
- TensorFlow or PyTorch framework for model execution
- Components of the project code will be tested in the implementation phase to ensure they work.
- The final integrated project code will be testing to ensure the full project is integrated and functioning correctly.
- Displays all the information after loading the app.