Project Report Format

1 INTRODUCTION

1.1 Project Overview

In this project, we build an effective electrocardiogram (ECG) arrhythmia classification method using a convolutional neural network (CNN), in which we classify ECG into seven categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are creating a web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the cited class will be displayed on the webpage.

1.2 Purpose

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death today. Over 17.7 million people died from CVDs in the year 2017 all over the world which is about 31% of all deaths, and over 75% of these deaths occur in low and middle-income countries. Arrhythmia is a representative type of CVD that refers to any irregular change from the normal heart rhythms. There are several types of arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia. Although a single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. So our purpose is to create a Machine Learning application that attempts to accurately classify the major types of Arrhythmia.

2 Literature Survey

2.1 Existing Problem

2.1 References

S. No	Paper Title	Abstract	Disadvantages
1.	An improved cardiac arrhythmia classification using an RR interval-based approach	Accurate and early detection of cardiac arrhythmia present in an electrocardiogram (ECG) can prevent many premature deaths. Cardiac arrhythmia arises due to the improper conduction of electrical impulses throughout the heart. In this paper, we propose an improved RR interval-based cardiac arrhythmia classification approach. The Discrete Wavelet Transform (DWT) and median filters were used to remove high-frequency noise and baseline wander from the raw ECG. Next, the processed ECG was segmented after the determination of the QRS region. We extracted the primary feature RR interval and other statistical features from the beats to classify the Normal, Premature Ventricular Contraction (PVC), and Premature Atrial Contraction (PAC). The K-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB), and Random Forest (RF)	Only the intra-patient validation was considered for this research thus, the results cannot reflect the generalization performance of the model

		classifier were utilised for classification. Overall performance of SVM with Gaussian kernel achieved Se % = 99.28, Sp % = 99.63, +P % = 99.28, and Acc % = 99.51, which is better than the other classifiers used in this method. The obtained results of the proposed method are significantly better and more accurate.	
2.	Classification of ECG Arrhythmia using Recurrent Neural Networks	In this paper, Recurrent Neural Networks (RNN) have been applied for classifying the normal and abnormal beats in an ECG. The primary aim of this paper was to enable automatic separation of regular and irregular beats. The MIT-BIH Arrhythmia database is being used to classify the beat classification performance. The methodology used is carried out using huge volume of standard data i.e. ECG time-series data as inputs to Long Short Term Memory Network. We divided the dataset as training and testing sub-data. The effectiveness, accuracy and capabilities of our methodology ECG arrhythmia detection is demonstrated and quantitative comparisons with different RNN models have also been carried out.	Recurrent Neural Networks (RNN) being used for this research is more suitable to handle temporal or sequential data. Convolutional Neural Network (CNN) has multiple convolutional filters working and scanning the complete feature matrix of the image making it a better choice compared to RNN.

3.	ECG-based heartbeat classification for arrhythmia detection: A survey	An electrocardiogram (ECG) measures the electric activity of the heart and has been widely used for detecting heart diseases due to its simplicity and non-invasive nature. By analyzing the electrical signal of each heartbeat, i.e., the combination of action impulse waveforms produced by different specialized cardiac tissues found in the heart, it is possible to detect some of its abnormalities. In the last decades, several works were developed to produce automatic ECG-based heartbeat classification methods. In this work, we survey the current state-of-the-art methods of ECG-based automated abnormalities heartbeat classification by presenting the ECG signal preprocessing, the heartbeat segmentation techniques, the feature description methods and the learning algorithms used. In addition, we describe some of the databases used for evaluation of methods indicated by a well-known standard developed by the Association for the Advancement of Medical Instrumentation (AAMI) and described in ANSI/AAMI EC57:1998/(R)2008 (ANSI/AAMI, 2008). Finally, we discuss limitations and	Implementation and inverse of Continuous Wavelet Transform (CWT) used in this paper are not readily available in the widely used tool boxes. A better choice would be to use Discrete Wavelet Transform.

		drawbacks of the methods in the literature presenting concluding remarks and future challenges, and also we propose an evaluation process workflow to guide authors in future works.	
4.	A deep learning approach for ECG-based heartbeat classification for arrhythmia detection	Classification is one of the most popular topics in healthcare and bioinformatics, especially in relation to arrhythmia detection. Arrhythmias are irregularities in the rate or rhythm of the heartbeat which, in some cases, may occur sporadically in a subject’s daily life. To capture these infrequent events, a Holter device is usually employed to record long-term ECG data. Therefore, the automatic recognition of abnormal heartbeats from a large amount of ECG data is an important and essential task. In the last two decades, a huge number of methods have been proposed to address the problem of ECG beat classification. At the same time, deep learning has advanced rapidly since the early 2000s and now demonstrates a state-of-the-art performance in various fields. In this paper, we propose a novel deep learning approach for ECG beat classification. We have conducted the experiments on the	Although vectorcardiogram (VCG) based features can provide comprehensive information about the heart condition, the reconstruction of the VCG requires more leads and thus the applicability of these features is rather limited.

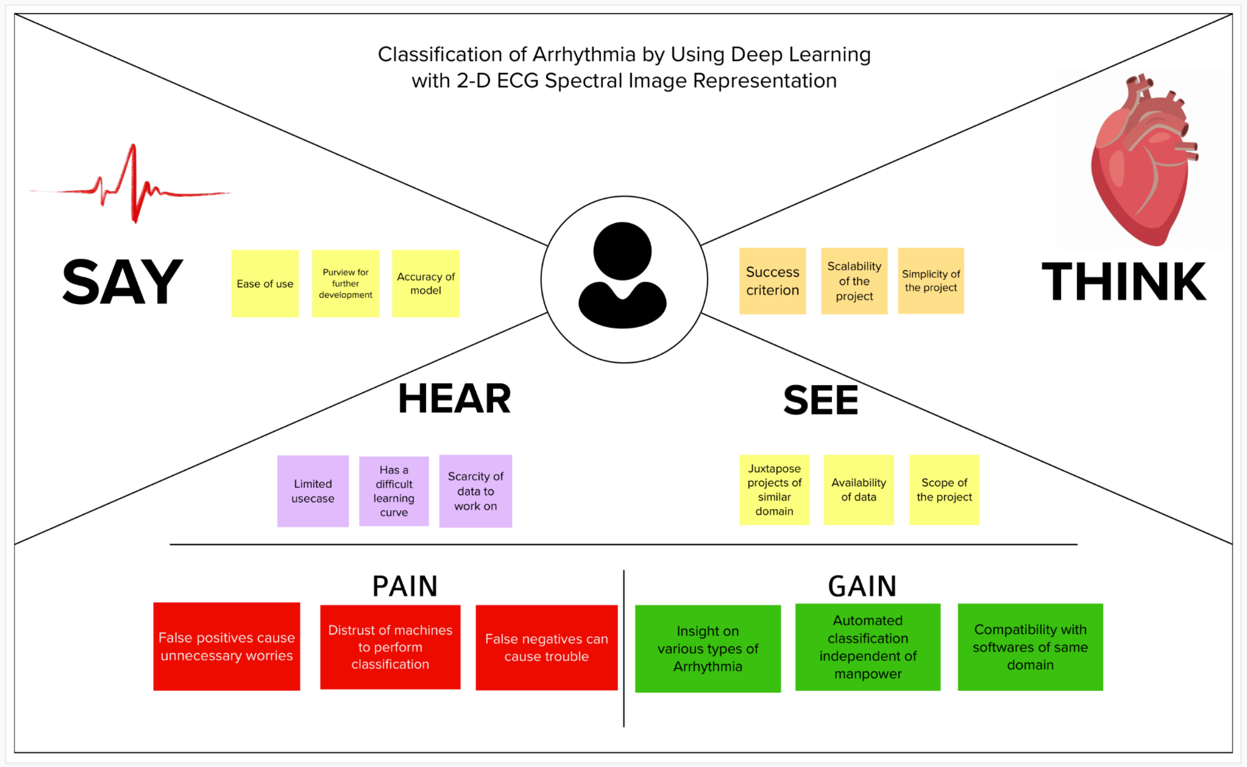
		well-known MIT–BIH Arrhythmia Database, and compared our results with the scientific literature. The final results show that our model is not only more efficient than the state of the art in terms of accuracy, but also competitive in terms of sensitivity and specificity	

2.3 Problem Statement Definition

The presented problem wants us to identify and classify the 6 major types of Arrhythmia provided in the dataset using Spectral Images of the Electrocardiogram that are employed for prediction of Cardiovascular Diseases.

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

SAY

Ease of use

Purview for further development

Accuracy of model

THINK

Success criterion

Scalability of the project

Simplicity of the project

HEAR

Limited usecase

Has a difficult learning curve

Scarcity of data to work on

SEE

Juxtapose projects of similar domain

Availability of data

Scope of the project

PAIN

False positives cause unnecessary worries

Distrust of machines to perform classification

False negatives can cause trouble

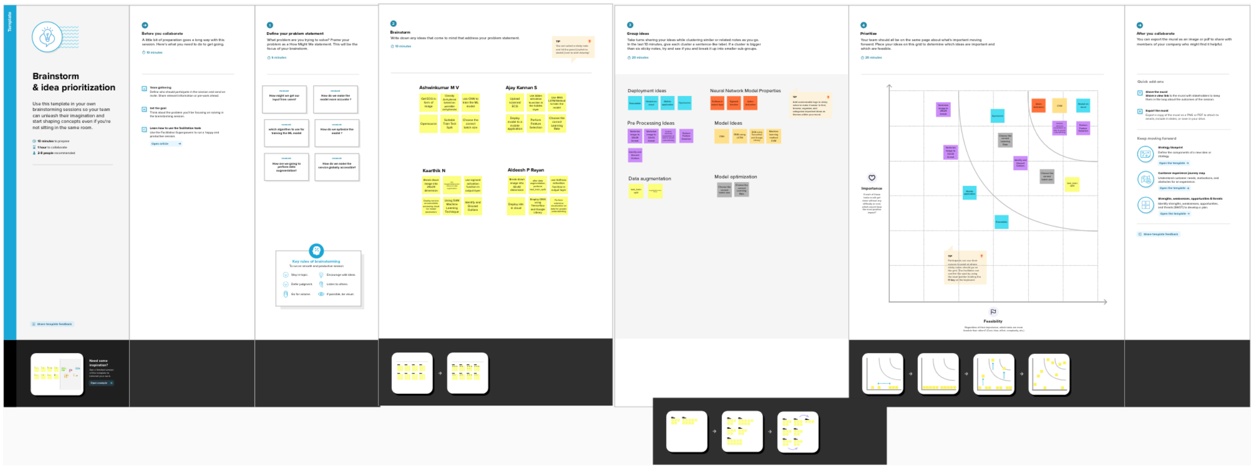
GAIN

Insight on various types of Arrhythmia

Automated classification independent of manpower

Compatibility with softwares of same domain

3.2 Ideation & Brainstorming



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Ashwinkumar M V

Get ECG in form of image

use CNN to train the ML model

Opensource

Suitable Train Test Split

Choose the correct batch size

Ajay Kannan S

Upload scanned ECG

Deploy model to a mobile application

Perform Feature Selection

Choose the correct Learning Rate

Kaarthik N

use sigmoid activation function in output layer

Using SVM Machine Learning Technique

Identify and Discard Outliers

Aldeesh P Rayan

Deploy site in cloud

Group ideas

20 minutes

Deployment Ideas

Neural Network Model Properties

Pre Processing Ideas

Model Ideas

Data augmentation

Model optimization

Importance

Feasibility

3.3 Proposed Solution

The given dataset for this problem initially undergoes various data pre-processing steps to identify various forms of noise in the dataset and denoise them to make the data suitable for training a deep learning model. We will employ 2-Dimensional Convolutional Neural Network Model to carry out this classification.

Novelty/Uniqueness :-

- ➢ Deploys the model to a mobile application by assigning all heavy pre-processing to the cloud through an API
- ➢ Usage of cloud based ML-training services
- ➢ Proposed solution will be deployed in the cloud making it easily accessible over the internet to people across the globe.
- ➢ Provide well detailed instructions or point the user to well equipped hospitals to get good treatment.

Social Impact/Customer Satisfaction :-

- ➢ Promotes Simplicity
- ➢ Promotes Self-Diagnosis
- ➢ Requires minimal effort and time
- ➢ Proposed solution abides by privacy laws and no private information of user is stored
- ➢ Delivers highly accurate results(classification of arrhythmia) in a short span of time.

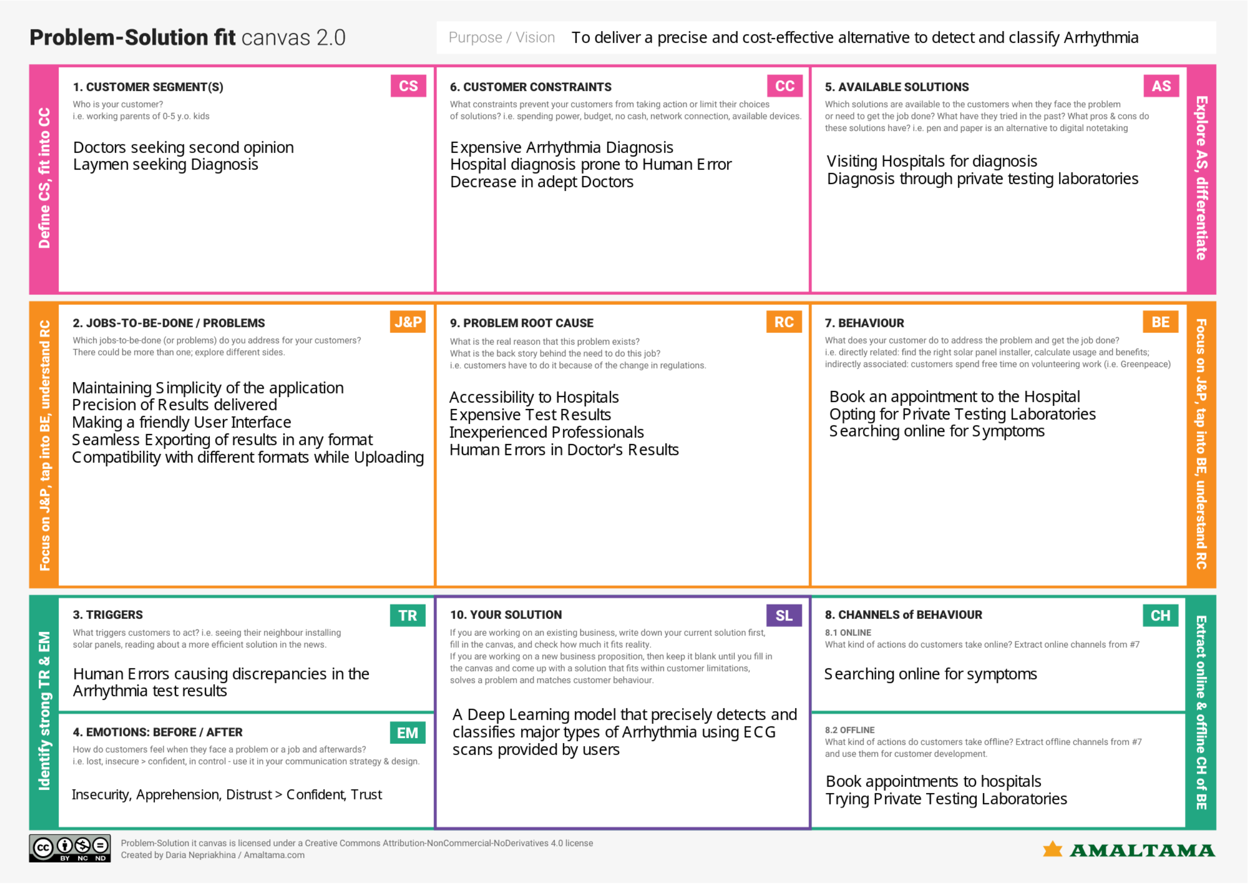
Business Model

- ➢ Our business model primarily covers the expense we incur by deploying the service in cloud platforms
- ➢ Primary consumers of our proposed service are hospitals who seek immediate consultation or use our service as a reference.
- ➢ Our service can be used by anybody who has access to internet services.
- ➢ Most of the competing products do not offer a
- ➢ Revenue is generated from Corporate Editions which has a monthly subscription, whereas the Community Edition is free for individuals.
- ➢ Users who would like to not travel to hospitals to get an ECG or get a self-diagnosis can rent/buy an ECG Machine through our service which will be the secondary source of income.

Scalability of the Solution

- Increasing the dataset used for model training will in turn increase the application's scalability
- Making the model more reliable will lead to hospitals using this application, this also eliminates human error
- More powerful Cloud Instances for Concurrent use of the application
- Periodically expanding the dataset and updating the model to increase scalability and reliability
- Cloud services guarantee high availability so there is very little probability that the service will face serious down times.

3.4 Problem Solution Fit



Problem-Solution fit canvas 2.0

Purpose / Vision: To deliver a precise and cost-effective alternative to detect and classify Arrhythmia

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Doctors seeking second opinion
Laymen seeking Diagnosis

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Expensive Arrhythmia Diagnosis
Hospital diagnosis prone to Human Error
Decrease in adept Doctors

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Visiting Hospitals for diagnosis
Diagnosis through private testing laboratories

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Maintaining Simplicity of the application
Precision of Results delivered
Making a friendly User Interface
Seamless Exporting of results in any format
Compatibility with different formats while Uploading

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Accessibility to Hospitals
Expensive Test Results
Inexperienced Professionals
Human Errors in Doctor's Results

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Book an appointment to the Hospital
Opting for Private Testing Laboratories
Searching online for Symptoms

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Human Errors causing discrepancies in the Arrhythmia test results

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Insecurity, Apprehension, Distrust > Confident, Trust

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

A Deep Learning model that precisely detects and classifies major types of Arrhythmia using ECG scans provided by users

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Searching online for symptoms

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Book appointments to hospitals
Trying Private Testing Laboratories

Extract online & offline CH of BE

AMALTAMA

4 REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	User Registration	Registration through website using email for verification
FR-2	User Confirmation	Verification of the OTP on the website
FR-3	User Data	Uploading of the ECG used for Arrhythmia Diagnosis
FR-4	User Diagnosis Results (Single Input)	Results of the Arrhythmia Prediction & Classification displayed in the website
FR-5	User Diagnosis Results (Multiple Input)	Results of the Arrhythmia Prediction & Classification sent to registered email

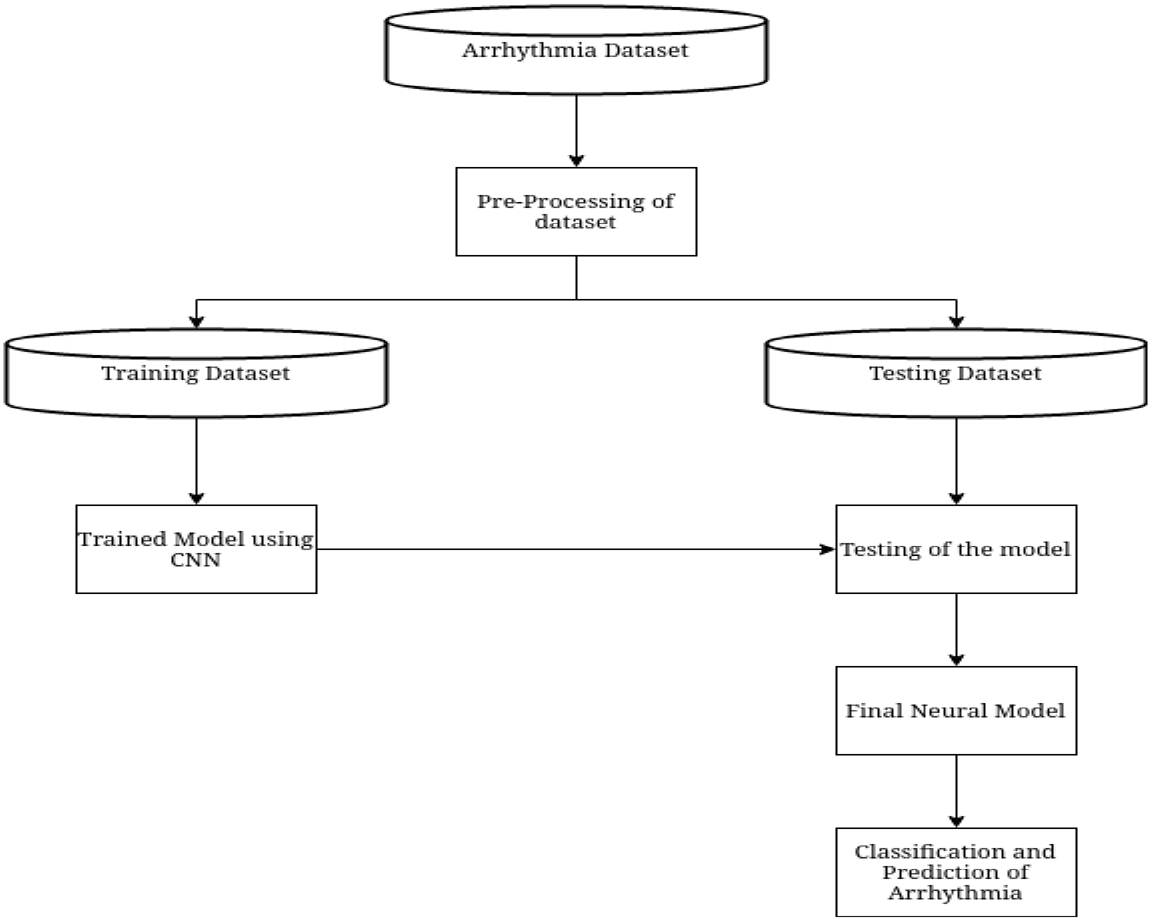
4.2 Non - Functional Requirements

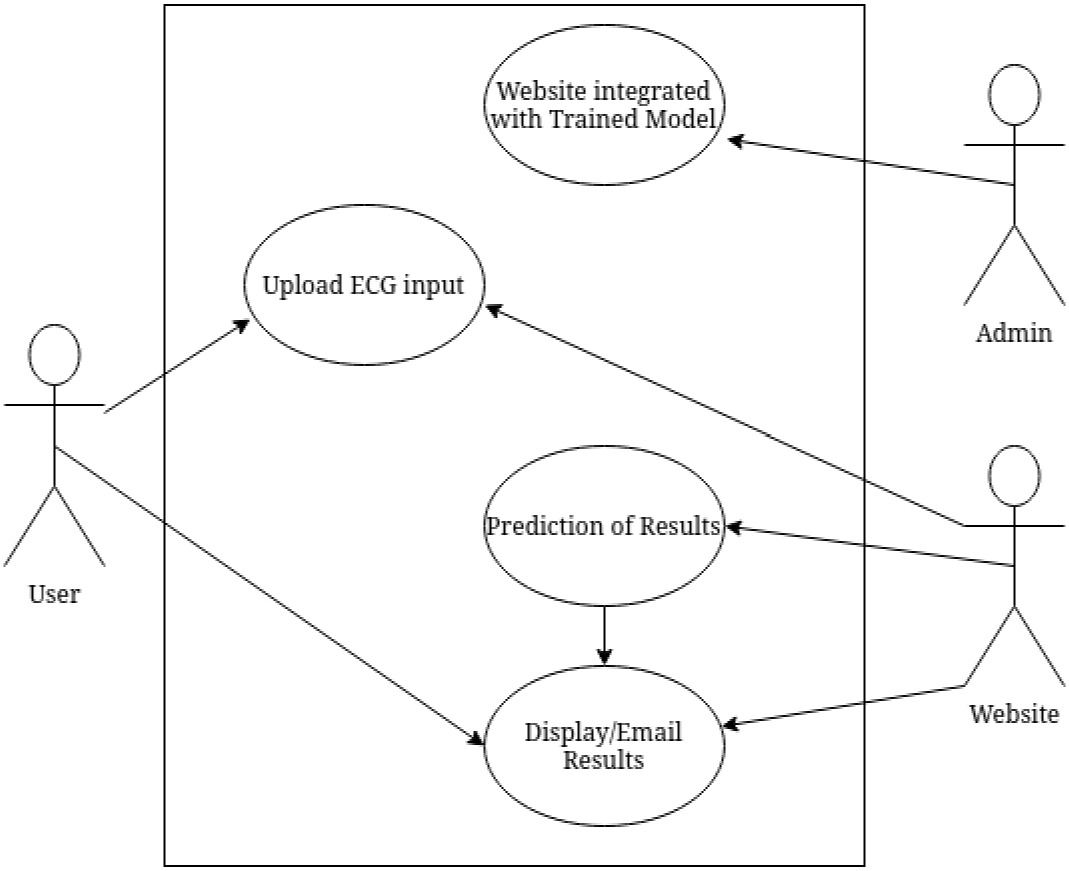
Following are the Non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
NFR-1	Usability	Efficiency and overall satisfaction of the user when using Diagnosis service through website
NFR-2	Security	Authentication and Authorization of the website
NFR-3	Reliability	Probability of successful operations for a specified time in an environment
NFR-4	Availability	Service should be readily available for 100% of the users
NFR-5	Scalability	Capacity of the website to handle growth

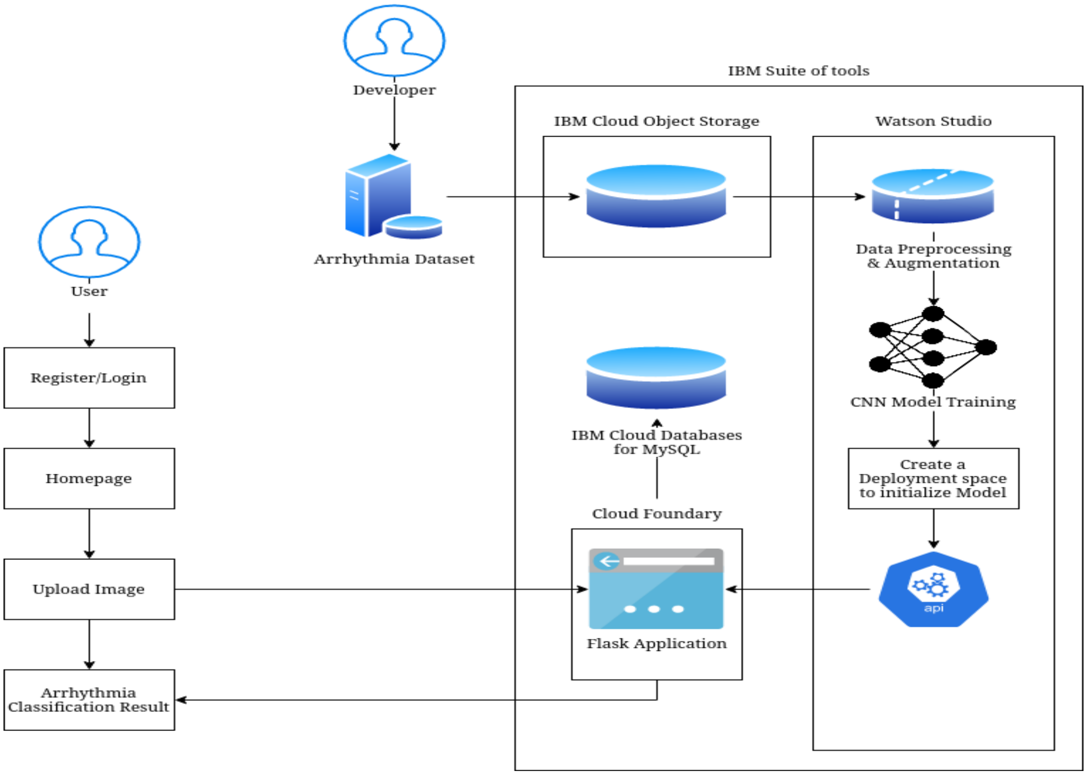
5 PROJECT DESIGN

5.1 Data Flow Diagrams

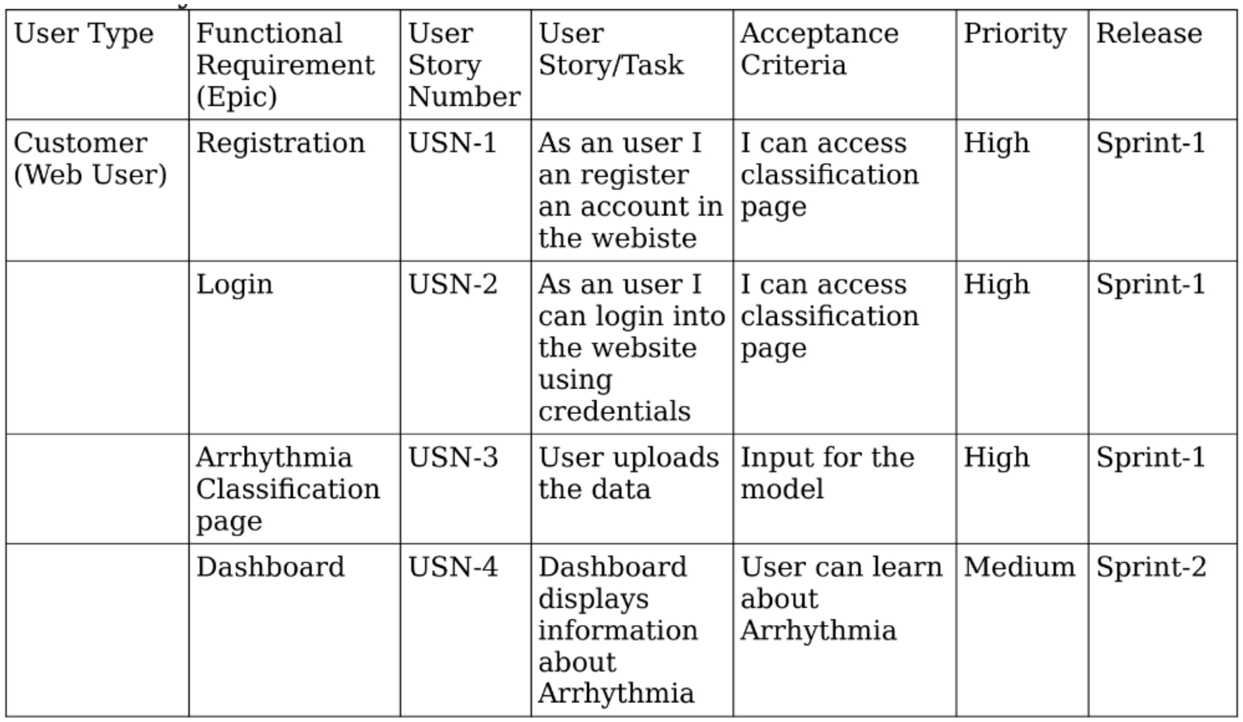




5.2 Solution Technical Architecture



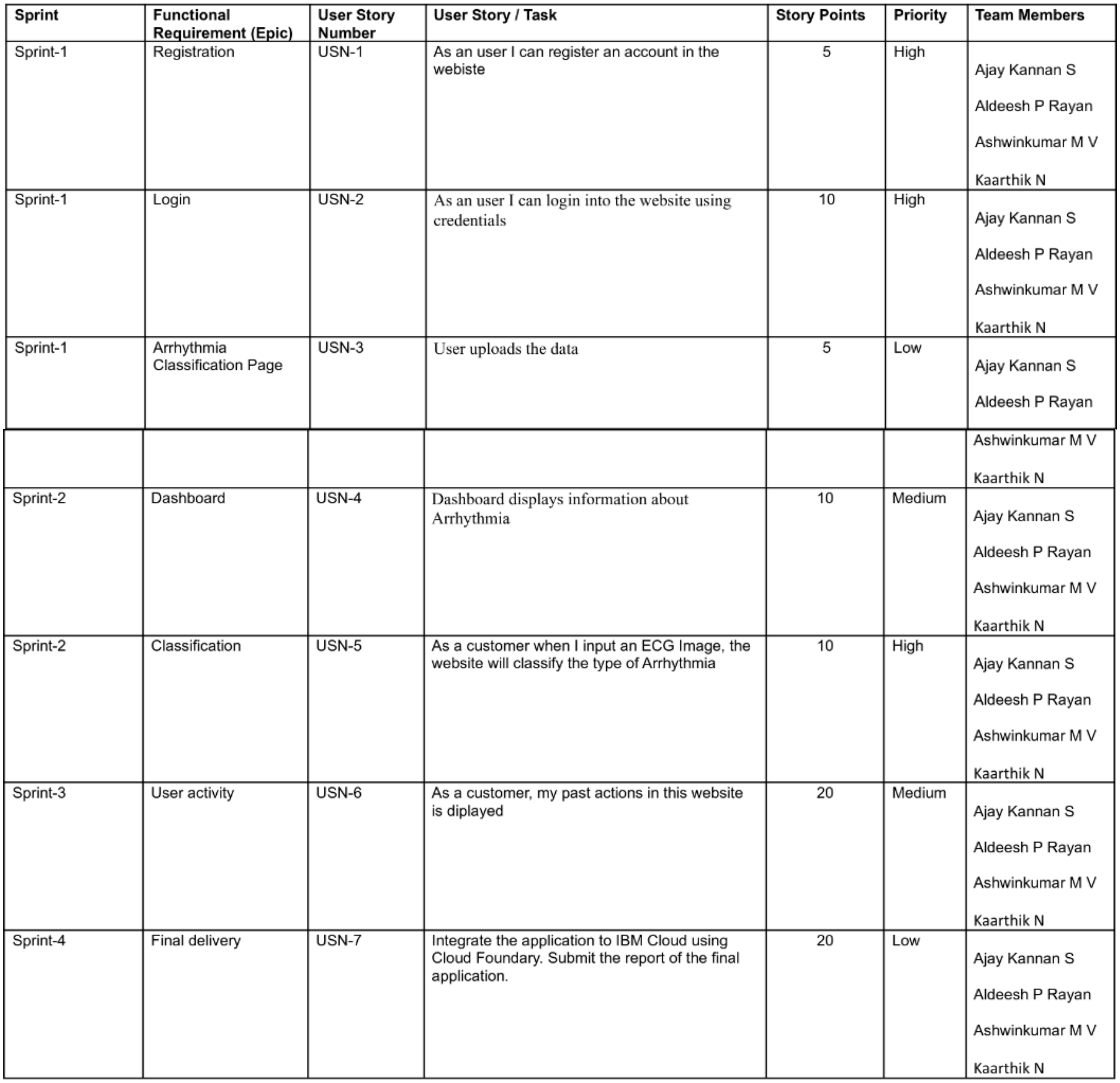
5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Customer (Web User)	Registration	USN-1	As an user I an register an account in the webiste	I can access classification page	High	Sprint-1
	Login	USN-2	As an user I can login into the website using credentials	I can access classification page	High	Sprint-1
	Arrhythmia Classification page	USN-3	User uploads the data	Input for the model	High	Sprint-1
	Dashboard	USN-4	Dashboard displays information about Arrhythmia	User can learn about Arrhythmia	Medium	Sprint-2

6 PROJECT PLANNING SCHEDULING

6.1 Sprint Planning & Estimation

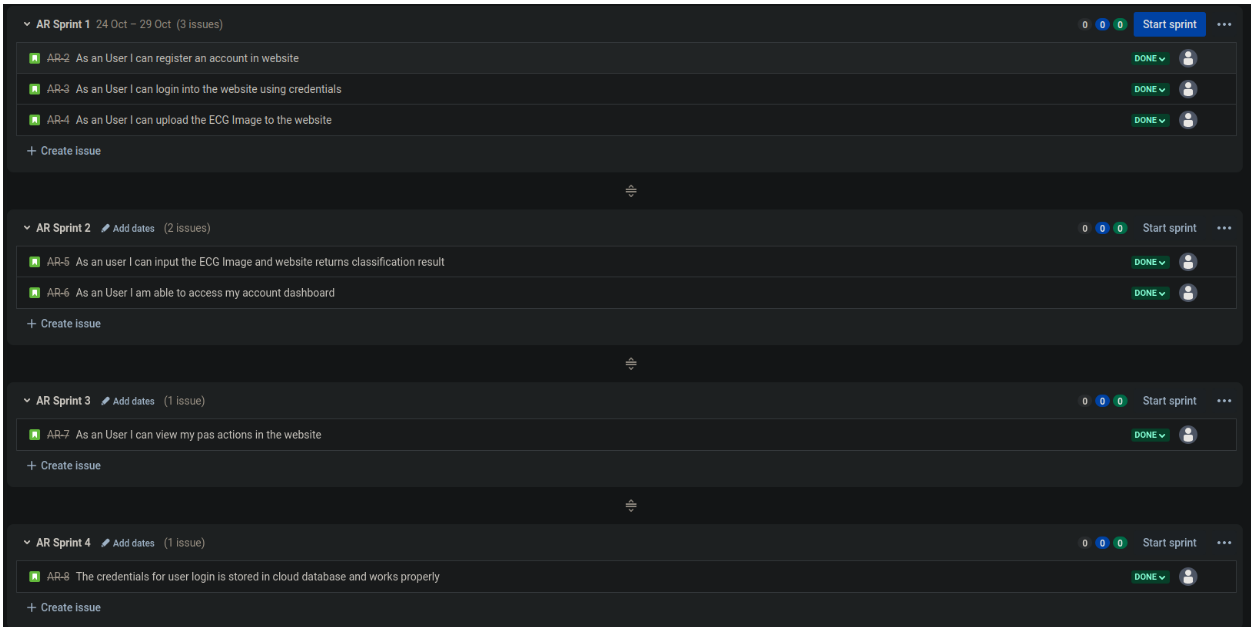


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As an user I can register an account in the webiste	5	High	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N
Sprint-1	Login	USN-2	As an user I can login into the website using credentials	10	High	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N
Sprint-1	Arrhythmia Classification Page	USN-3	User uploads the data	5	Low	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N
Sprint-2	Dashboard	USN-4	Dashboard displays information about Arrhythmia	10	Medium	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N
Sprint-2	Classification	USN-5	As a customer when I input an ECG Image, the website will classify the type of Arrhythmia	10	High	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N
Sprint-3	User activity	USN-6	As a customer, my past actions in this website is diplayed	20	Medium	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N
Sprint-4	Final delivery	USN-7	Integrate the application to IBM Cloud using Cloud Foundary. Submit the report of the final application.	20	Low	Ajay Kannan S Aldeesh P Rayan Ashwinkumar M V Kaarthik N

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA



AR Sprint 1 24 Oct – 29 Oct (3 issues)

- ~~AR-2~~ As an User I can register an account in website — DONE
- ~~AR-3~~ As an User I can login into the website using credentials — DONE
- ~~AR-4~~ As an User I can upload the ECG Image to the website — DONE

AR Sprint 2 Add dates (2 issues)

- ~~AR-5~~ As an user I can input the ECG Image and website returns classification result — DONE
- ~~AR-6~~ As an User I am able to access my account dashboard — DONE

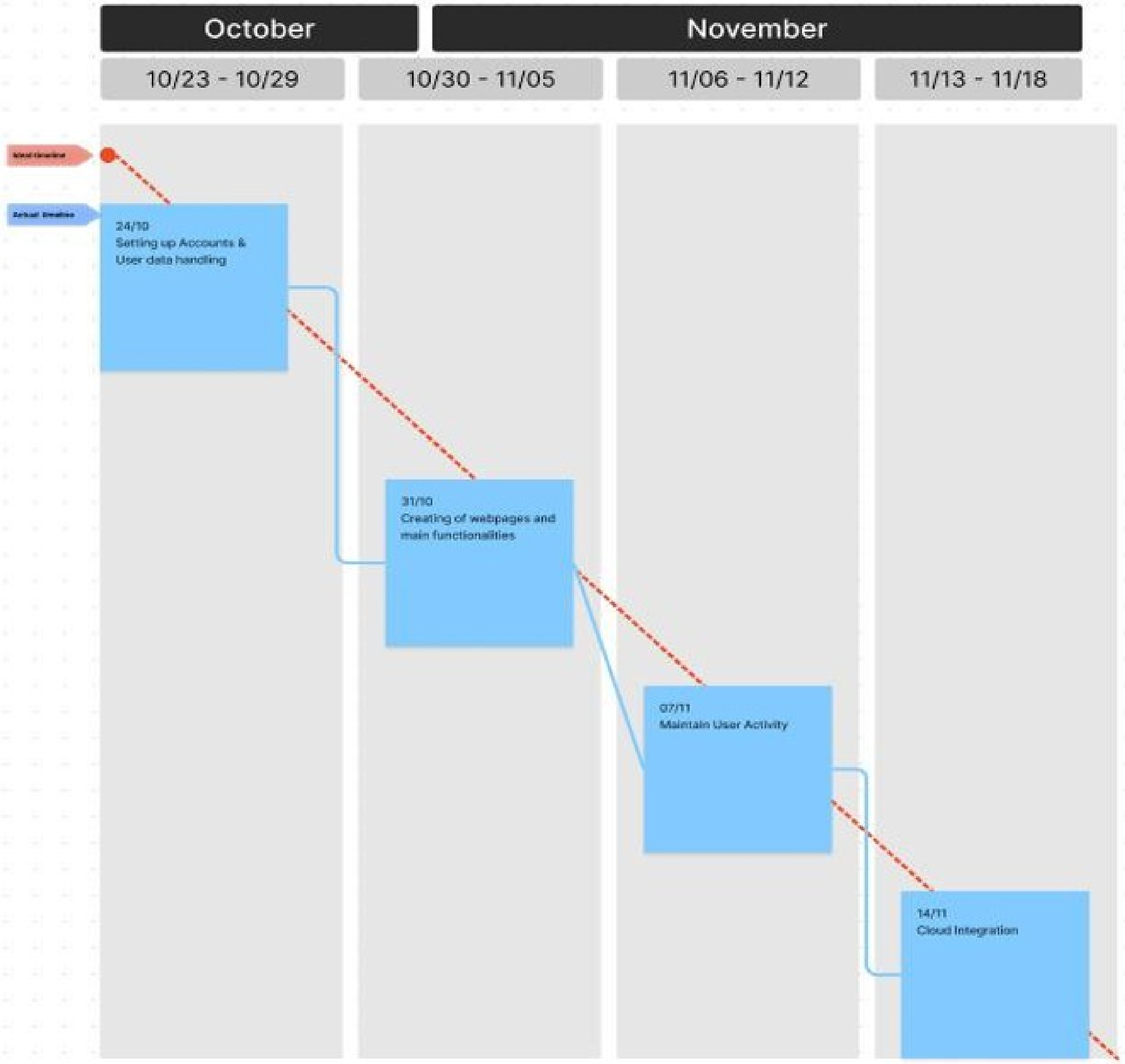
AR Sprint 3 Add dates (1 issue)

- ~~AR-7~~ As an User I can view my pas actions in the website — DONE

AR Sprint 4 Add dates (1 issue)

- ~~AR-8~~ The credentials for user login is stored in cloud database and works properly — DONE

June / July 2022



24/10
Setting up Accounts &
User data handling

31/10
Creating of webpages and
main functionalities

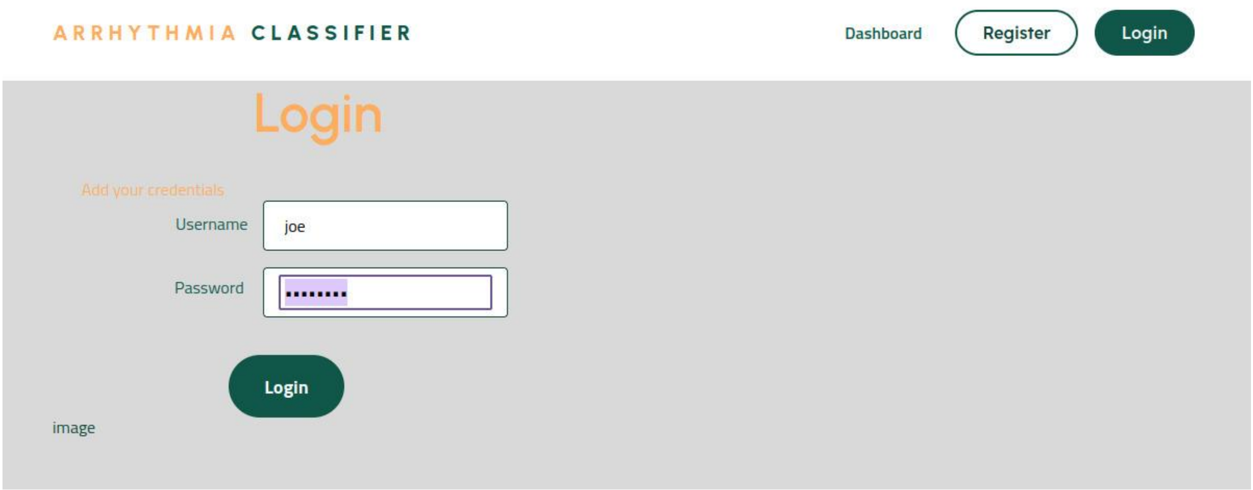
07/11
Maintain User Activity

14/11
Cloud Integration

Finished	2
To-do	3

7 CODING & SOLUTIONING

Login Feature



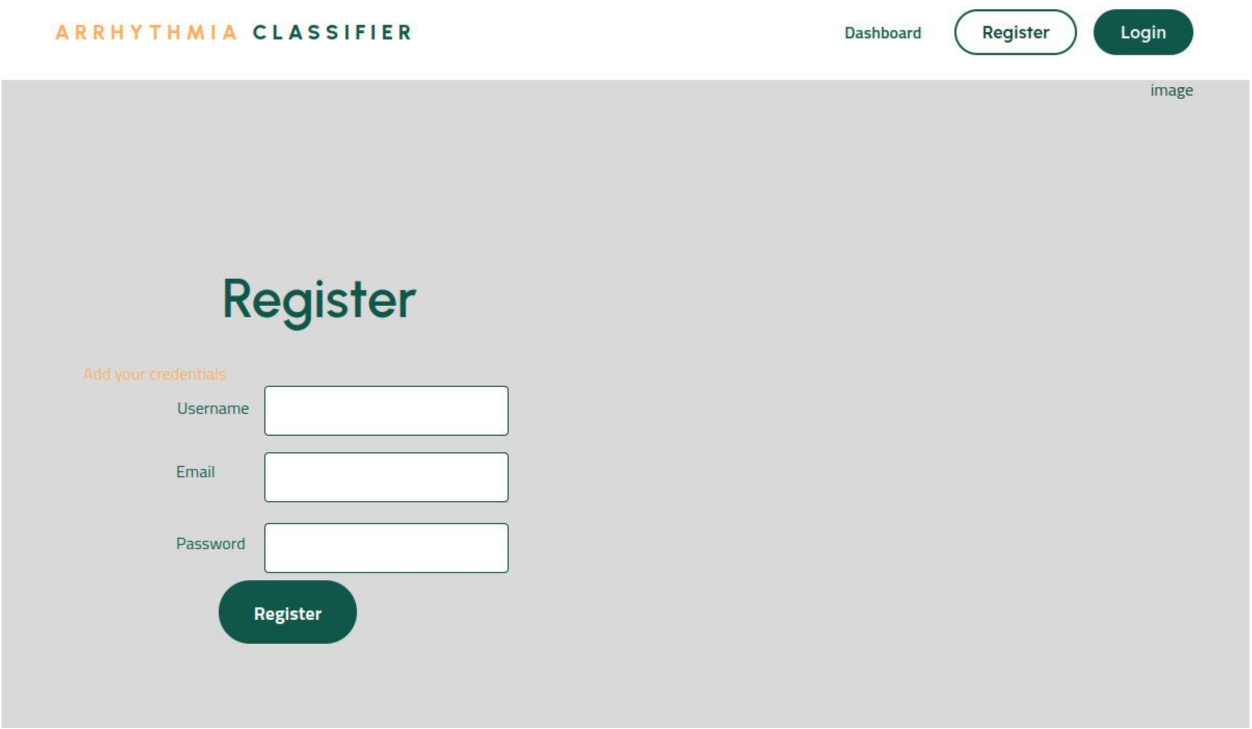
Login

Add your credentials

Username: joe

Password: ....

Register Feature



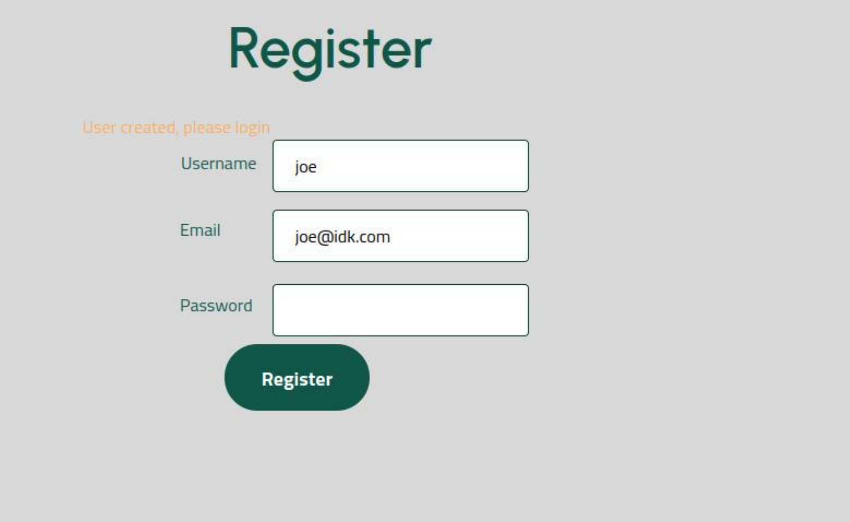
Register

Add your credentials

Username:

Email:

Password:



Register

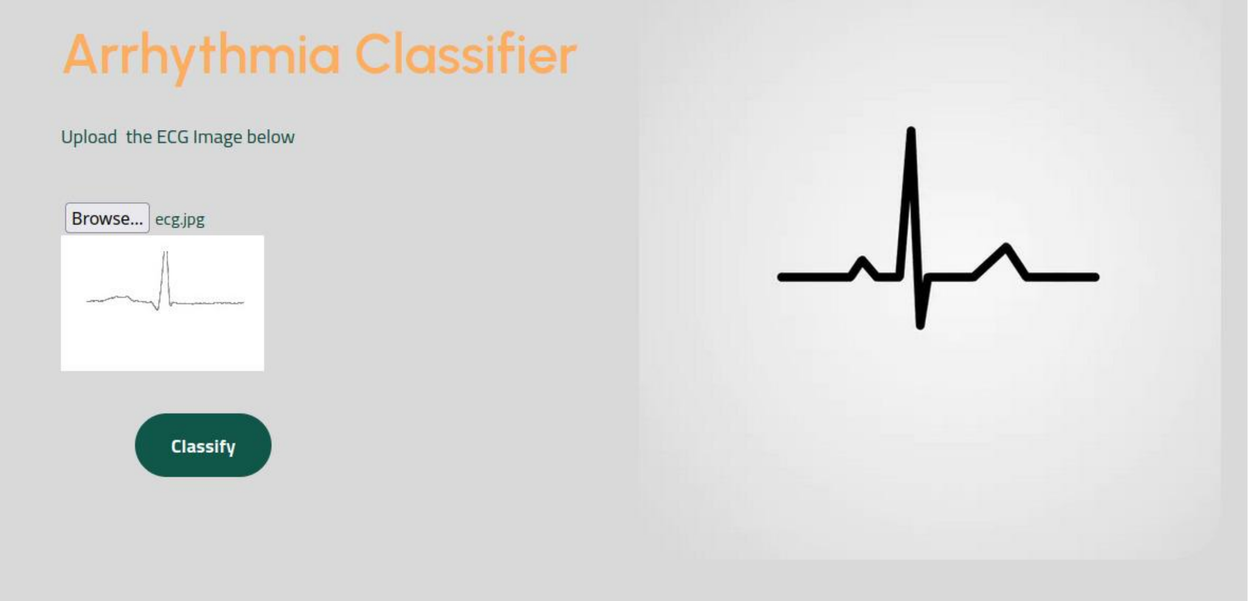
User created, please login

Username: joe

Email: joe@idk.com

Password:

Classification Feature

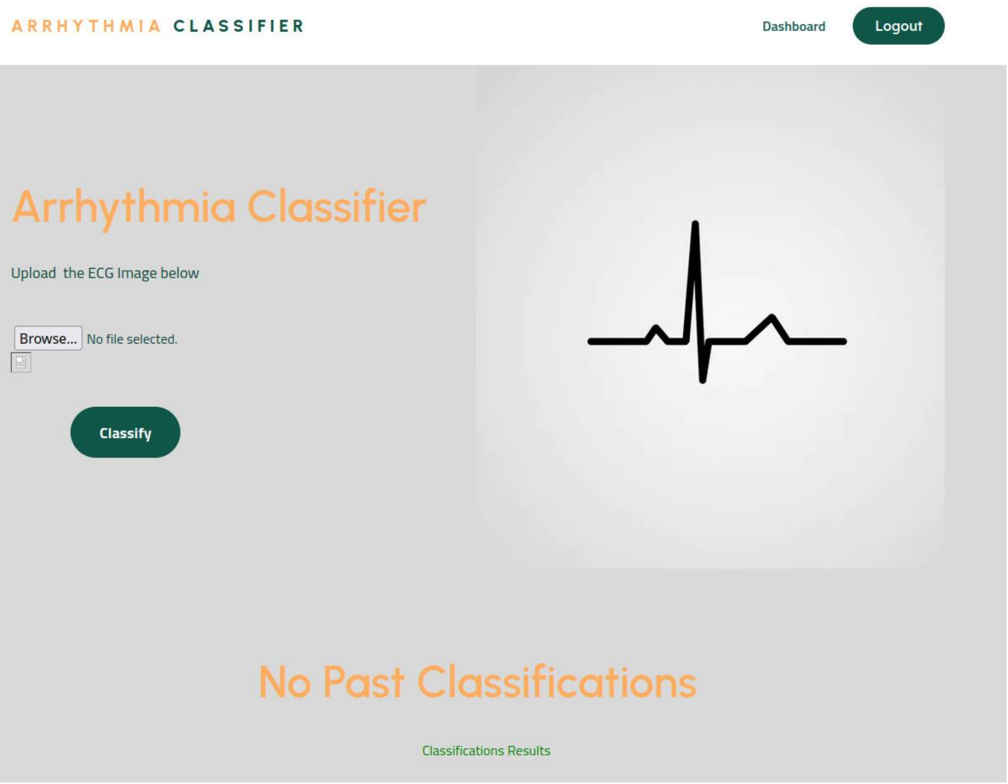


Arrhythmia Classifier

Upload the ECG Image below

ecg.jpg

Classification History Feature



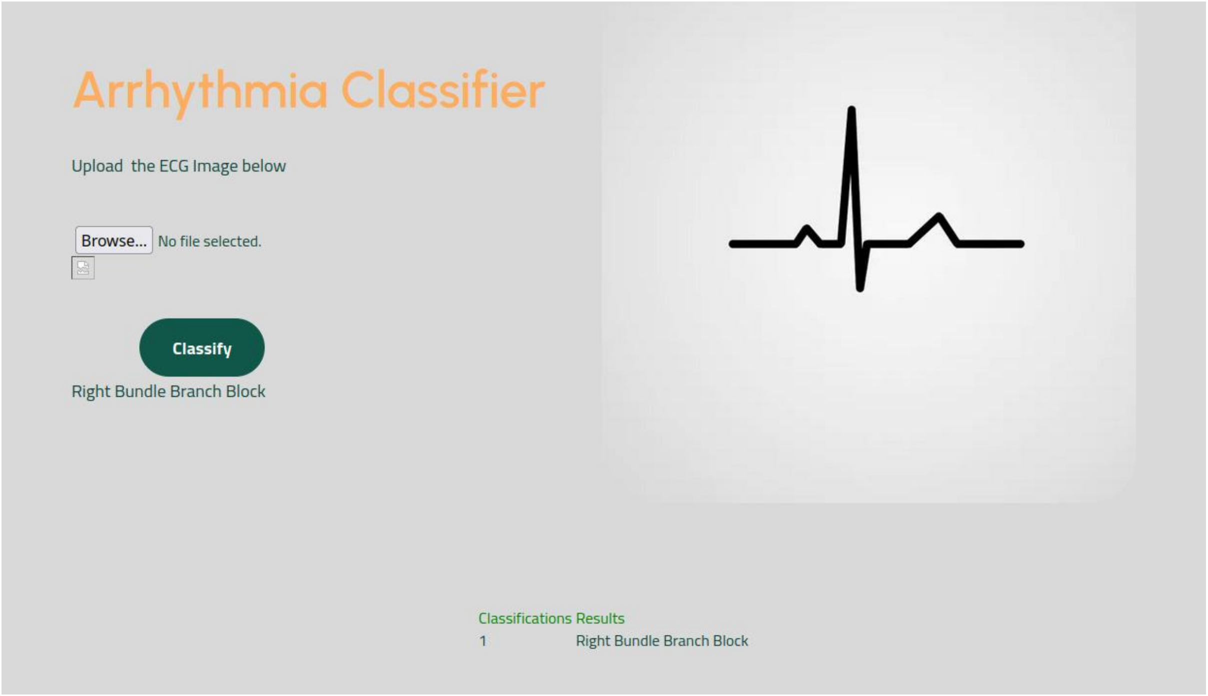
Arrhythmia Classifier

Upload the ECG Image below

No file selected.

No Past Classifications

Classifications Results



Arrhythmia Classifier

Upload the ECG Image below

No file selected.

Right Bundle Branch Block

Classifications Results

1   Right Bundle Branch Block

8 TESTING

8.1 Test Cases

8.1.1 Functional Testing

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

8.1.2 White Box Testing

Testing based on an analysis of internal workings and structure of a piece of software. This testing can be done sing the percentage value of load and energy. The tester should know what exactly is done in the internal program. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing.

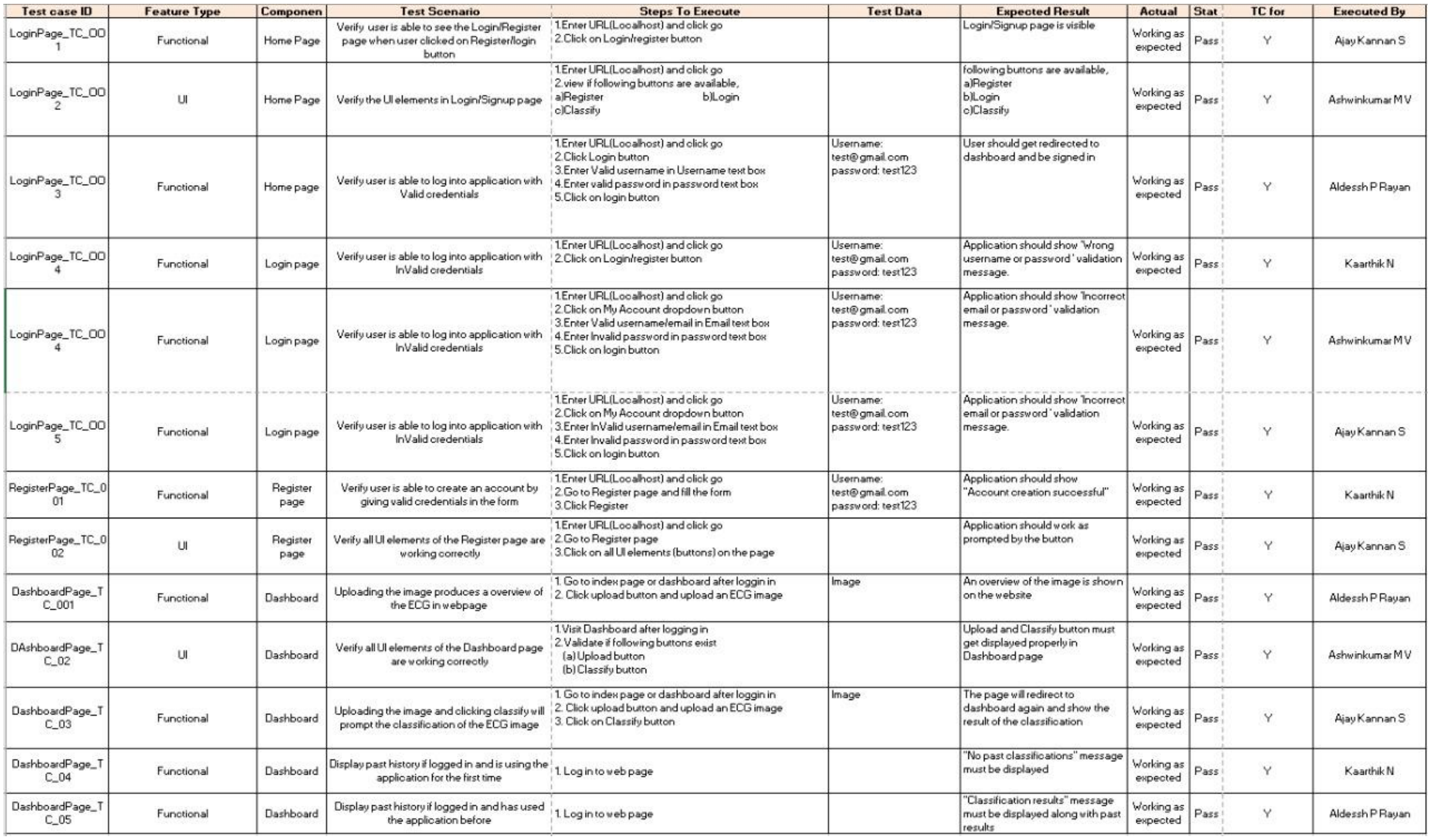
8.1.3 Black Box Testing

Testing without knowledge of the internal workings of the item being tested. Tests are usually functional. This testing can be done by the user who has no knowledge of how the shortest path is found.

8.2 User Acceptance Testing

Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages. The function of performance characteristics to specification and is accepted. A deviation from specification is uncovered and a deficiency list is created. User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

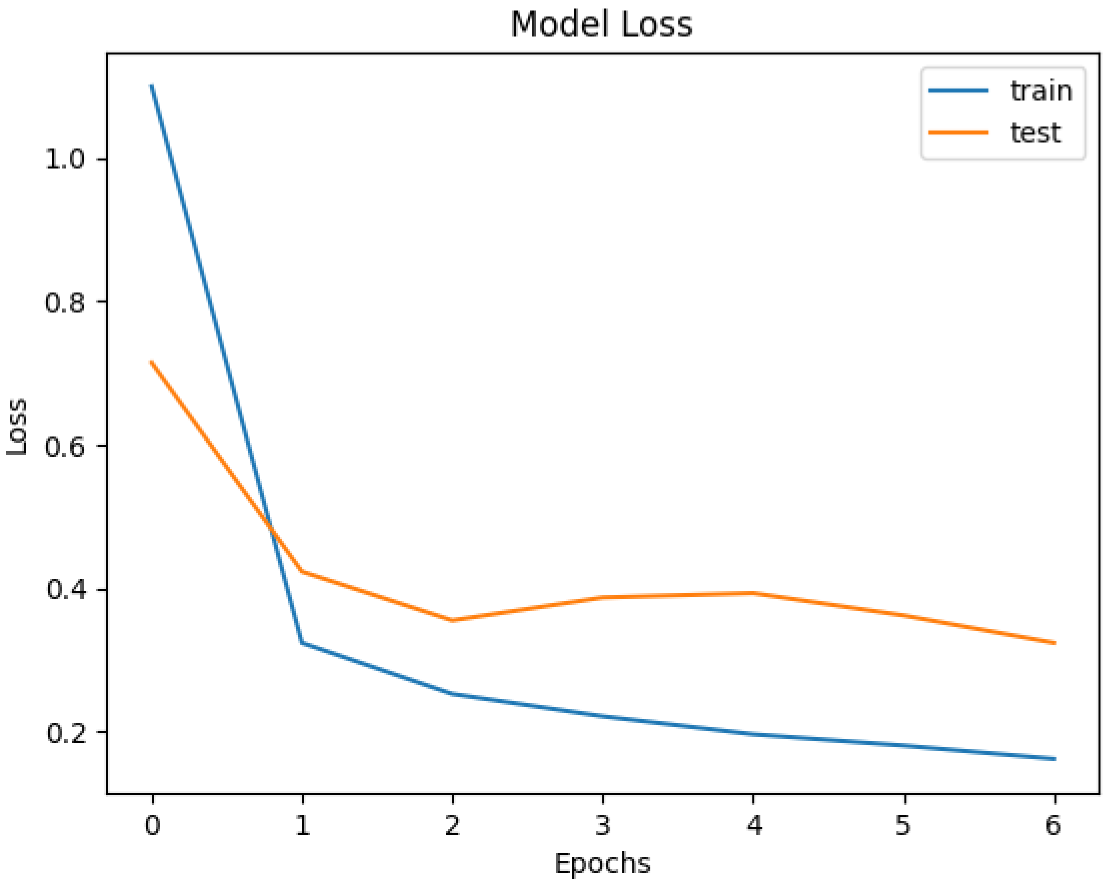
8.3 Test Results



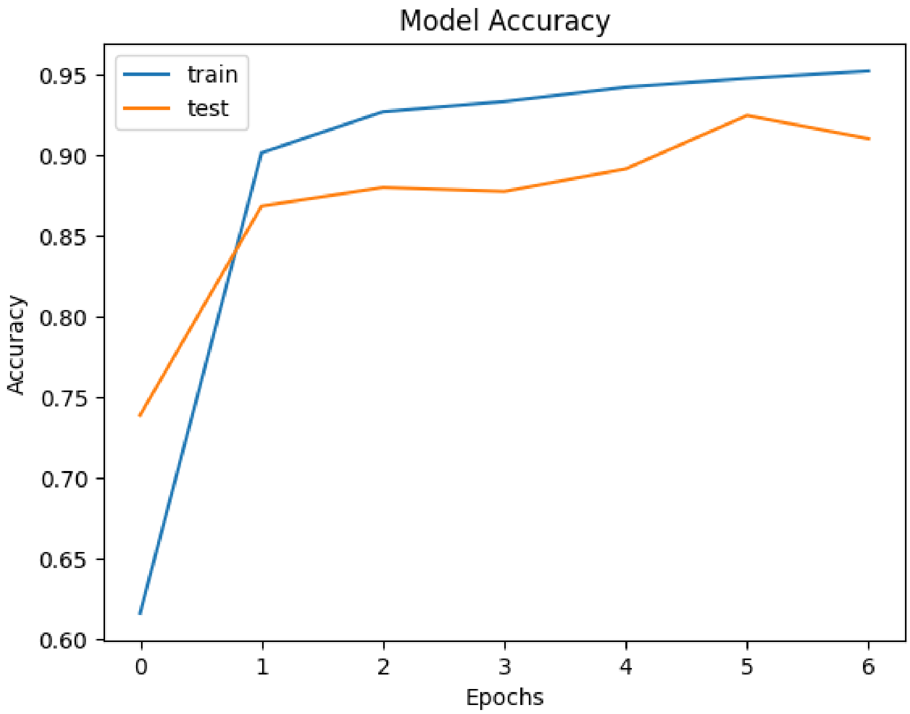
Test case ID	Feature Type	Componen	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual	Stat	TC for	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Register page when user clicked on Register/login button	1.Enter URL(Localhost) and click go 2.Click on Login/register button		Login/Signup page is visible	Working as expected	Pass	Y	Ajay Kannan S
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup page	1.Enter URL(Localhost) and click go 2.view if following buttons are available, a)Register b)Login c)Classify		following buttons are available, a)Register b)Login c)Classify	Working as expected	Pass	Y	Ashwinkumar M V
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL(Localhost) and click go 2.Click Login button 3.Enter Valid username in Username text box 4.Enter valid password in password text box 5.Click on login button	Username: test@gmail.com password: test123	User should get redirected to dashboard and be signed in	Working as expected	Pass	Y	Aldessh P Rayan
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on Login/register button	Username: test@gmail.com password: test123	Application should show 'Wrong username or password ' validation message.	Working as expected	Pass	Y	Kaarthik N
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: test@gmail.com password: test123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Y	Ashwinkumar M V
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: test@gmail.com password: test123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Y	Ajay Kannan S
RegisterPage_TC_001	Functional	Register page	Verify user is able to create an account by giving valid credentials in the form	1.Enter URL(Localhost) and click go 2.Go to Register page and fill the form 3.Click Register	Username: test@gmail.com password: test123	Application should show "Account creation successful"	Working as expected	Pass	Y	Kaarthik N
RegisterPage_TC_002	UI	Register page	Verify all UI elements of the Register page are working correctly	1.Enter URL(Localhost) and click go 2.Go to Register page 3.Click on all UI elements (buttons) on the page		Application should work as prompted by the button	Working as expected	Pass	Y	Ajay Kannan S
DashboardPage_TC_001	Functional	Dashboard	Uploading the image produces a overview of the ECG in webpage	1. Go to index page or dashboard after loggin in 2. Click upload button and upload an ECG image	Image	An overview of the image is shown on the website	Working as expected	Pass	Y	Aldessh P Rayan
DAshboardPage_TC_02	UI	Dashboard	Verify all UI elements of the Dashboard page are working correctly	1.Visit Dashboard after logging in 2.Validate if following buttons exist (a) Upload button (b) Classify button		Upload and Classify button must get displayed properly in Dashboard page	Working as expected	Pass	Y	Ashwinkumar M V
DashboardPage_TC_03	Functional	Dashboard	Uploading the image and clicking classify will prompt the classification of the ECG image	1. Go to index page or dashboard after loggin in 2. Click upload button and upload an ECG image 3. Click on Classify button	Image	The page will redirect to dashboard again and show the result of the classification	Working as expected	Pass	Y	Ajay Kannan S
DashboardPage_TC_04	Functional	Dashboard	Display past history if logged in and is using the application for the first time	1. Log in to web page		"No past classifications" message must be displayed	Working as expected	Pass	Y	Kaarthik N
DashboardPage_TC_05	Functional	Dashboard	Display past history if logged in and has used the application before	1. Log in to web page		"Classification results" message must be displayed along with past results	Working as expected	Pass	Y	Aldessh P Rayan

9 RESULTS

9.1 Performance Metrics



Finally, the model with training data yields an accuracy of 95% and evaluating it with test data yields an accuracy of 91%



10 ADVANTAGES & DISADVANTAGES

Advantages

- ➔ User friendly and simple user interface
- ➔ Get instantaneous results for classification
- ➔ High accuracy classification
- ➔ Simple account creation with option classification history
- ➔ Saves time

Disadvantages

- ➔ Traffic increase leads to service outage
- ➔ Size of entire web application too large for deployment in cloud foundry

11 CONCLUSION

In this project, we have built an effective ECG Classification method using Convolutional Neural Networks (CNN) in which we classify ECG into seven major categories, one being normal and the other being different types of arrhythmia using deep two dimensional CNN with grayscale ECG images. We created a web application where the user uploads an ECG image to the site. The image is then preprocessed and sent to the CNN model which was deployed into the cloud space created with the help of IBM and the result of the classification is sent as a response from which the needed information is passed and transformed to be displayed in the website. The user was also able to create account and login to keep track of the past classifications done by them

12 FUTURE SCOPE

In future work, we can design an integrated system to classify arrhythmia ECG signals, which will monitor and scan the patient’s ECG via the internal camera of the robot and will predict and diagnose the arrhythmia ECG signal to advise the medical expert. The current research relies on the use of a single ECG signal. The use of multi-channel data for categorizing ECG data will be useful in the future. We can also update the trained model with a more diverse dataset from different databanks to make the model more versatile with the inputs

13 APPENDIX

Source Code

```
import os, logging

from flask               import render_template, request, url_for, redirect, send_from_directory
from flask_login         import login_user, logout_user, current_user, login_required
from werkzeug.exceptions import HTTPException, NotFound, abort
from jinja2              import TemplateNotFound

from app        import app, lm, db, bc
from app.models import Users
from app.forms  import LoginForm, RegisterForm

import requests
from tensorflow.keras.models import load_model
from keras.preprocessing import image
import keras
import tensorflow as tf
import numpy as np
import json
from json import JSONEncoder
from ibm_watson_machine_learning import APIClient
import os


class NumpyArrayEncoder(JSONEncoder):
    def default(self, obj):
        if isinstance(obj, np.ndarray):
            return obj.tolist()
        return JSONEncoder.default(self, obj)

UPLOAD_FOLDER = "uploads"

@app.route('/classify', methods=['POST'])
def classify():
    if request.method == 'POST':
        file = request.files['fileupload']
        file_path = os.path.join(os.getcwd() + '/app/' + UPLOAD_FOLDER + '/' + file.filename)
        file.save(file_path)
        print(file_path)
        img = keras.utils.load_img(file_path, target_size=(64,64))
        x = tf.keras.utils.img_to_array(img)
        x = np.expand_dims(x, axis=0)

        numpyData = {"input_data": x}
        encodedNumpyData = json.dumps(numpyData, cls=NumpyArrayEncoder)

        API_KEY = os.environ["API_KEY"]
        token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
        mltoken = token_response.json()["access_token"]

        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

        payload_scoring = {
                    "input_data": [{
                            "fields": [],
                            "values": x.tolist()
                }]
        }

        response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/16df8175-8bb2-4ed2-9893-ac61d3349a89/predictions?version=2022-11-11', json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
        res = response_scoring.json()['predictions'][0]['values'][0][0]
        count = 0
        pos = res.index(1.0)
        types = ['Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricular Contractions', 'Right Bundle Branch Block', 'Ventricular Fibrillation']
        current_user.addHistory(types[pos])
        hist = current_user.getHistory()
        return render_template('index.html', result = types[pos], history=hist)
    return render_template('index.html')

# provide login manager with load_user callback
# return User(u.name,u.id,u.active)
@lm.user_loader
```

```
def load_user(user_id):
    email, uname, passwd = Users.getUser(user_id)
    return Users(uname, email, passwd)

# Logout user
@app.route('/logout')
def logout():
    logout_user()
    return redirect(url_for('index'))

# Register a new user
@app.route('/register', methods=['GET', 'POST'])
def register():

    # declare the Registration Form
    form = RegisterForm(request.form)

    msg     = None
    success = False

    if request.method == 'GET':

        return render_template( 'register.html', form=form, msg=msg )

    # check if both http method is POST and form is valid on submit
    if form.validate_on_submit():

        # assign form data to variables
        username = request.form.get('username', '', type=str)
        password = request.form.get('password', '', type=str)
        email    = request.form.get('email'   , '', type=str)

        # filter User out of database through username
        user = Users.userExists(username)

        # filter User out of database through username
        user_by_email = Users.emailExists(email)

        if user or user_by_email:
            msg = 'Error: User exists!'
        else:

            pw_hash = bc.generate_password_hash(password)

            user = Users(username, email, pw_hash)

            user.save()

            msg     = 'User created, please <a href="' + url_for('login') + '">login</a>'
            success = True

    else:
        msg = 'Input error'

    return render_template( 'register.html', form=form, msg=msg, success=success )

# Authenticate user
@app.route('/login', methods=['GET', 'POST'])
def login():

    # Declare the login form
    form = LoginForm(request.form)

    # Flask message injected into the page, in case of any errors
    msg = None

    # check if both http method is POST and form is valid on submit
    if form.validate_on_submit():

        # assign form data to variables
        username = request.form.get('username', '', type=str)
        password = request.form.get('password', '', type=str)

        # filter User out of database through username
        u = Users.getUserWithUname(username)
        user = Users(u[1], u[0], u[2])
        if user:
            if bc.check_password_hash(user.password, password):
                login_user(user)
                return redirect(url_for('index'))
            else:
                msg = "Wrong password. Please try again."
        else:
            msg = "Unknown user"

    return render_template( 'login.html', form=form, msg=msg )

# App main route + generic routing
@app.route('/', defaults={'path': 'index'})
@app.route('/<path>')
def index(path):
    try:
        if current_user.is_authenticated:
            hist = current_user.getHistory()
            return render_template('index.html', history=hist)

        return render_template( 'index.html' )

    except TemplateNotFound:
        return render_template('index.html'), 404

    except:
        return render_template('index.html'), 500

# Return sitemap
@app.route('/sitemap.xml')
def sitemap():
    return send_from_directory(os.path.join(app.root_path, 'static'), 'sitemap.xml')
```