MINI PROJECT REPORT

On

WellBe – A Health Prediction App

Submitted By

Mr. Mohammad Awais **Mr. Mahesh Talmale**

Mr. Lokesh Patel **Ms. Mitali Manekar**

Submitted in partial fulfillment of the requirements
Of
Mini Project in third year of Bachelor of Engineering

Guided by-
Mr. Sameer Mendhe

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

S.B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

2021-2022

Institute Vision:

Emerge as a leading Institute for developing competent and creative Professionals.

Institute Mission:

- Providing Quality Infrastructure and experienced faculty for academic excellence.
- Inculcating skills, knowledge and opportunities for competency and creativity.
- Aligning with Industries for knowledge sharing, research and development.

Department Vision:

To become a center for quality education in the field of computer science & engineering and to create competent professionals.

Department Mission:

To provide academic ambience and latest software tools to prepare competent Software Engineers with strong theoretical and practical knowledge.

To foster professionalism and strong work ethics in students for the betterment of Society

To provide adequate infrastructure as well as experienced & skilled faculty members.

To encourage the spirit of entrepreneurship and adaptability in our students in view of the ever-changing scenario of the Software Industry.

Program Educational Objectives (PEO's)

Have analytical, design and implementation skills, to innovate, design and develop software products and systems.

Have strong work ethics and professionalism, reflected through communication skills, leadership, teamwork and sense of responsibility towards the society.

Be successful professionals through lifelong learning with allied objectives of higher education or research.

S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT AND RESEARCH, NAGPUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SESSION 2020-2021

CERTIFICATE

This is to certify that the Mini project titled **"WellBe - A Health Prediction App"** is a bonafide work of **Mr. Mohammad Awais, Mr. Mahesh Talmale, Mr. Lokesh Patel, Ms. Mitali Manekar,** carried out for the partial fulfillment of the requirement of Mini Project in third year of Bachelor of Engineering in **Computer Science & Engineering, Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur.**

Mr. Sameer Mendhe
Assistant Professor
(Project Guide)

Mr. Animesh Tayal
Head of Department

Dr. S. L. Badjate
Principal

External Examiner

Mr/Mrs/Ms/Dr. ____________________

Designation: ____________________

Institution: ____________________

Date: ____________________

DECLARATION

We, hereby declare that the Mini Project titled **“*WellBe – A Health Prediction App*”** submitted herein has been carried out by us in the Department of Computer Science & Engineering of S. B. Jain Institute of Technology Management and Research, Nagpur under the guidance of **Mr. Sameer Mendhe**. The work is original and has not been submitted earlier as a whole or in part for the award of any degree / diploma at this or any other Institution / University.

.

Mr. Mohammad Awais ________________

Mr. Mahesh Talmale ________________

Mr. Lokesh Patel ________________

Ms. Mitali Manekar ________________

Date: -   /   /

ACKNOWLEDGEMENT

We would like to express deep sense of gratitude to our Project Guide, **Mr. Sameer Mendhe**, **Department of Computer Science & Engineering**, for being the cornerstone of our project. It was her incessant motivation and guidance during periods of doubts and uncertainties that has helped us to carry on with this project.

We would like to thank **Mr. Animesh Tayal**, **Head of Department**, **Computer Science & Engineering** for providing necessary guidance, support, motivation and inspiration without which this project would not have been possible.

We would like to extend our special thanks to **Dr. S. L. Badjate**, **Principal** of **S.B. Jain Institute of Technology, Management & Research** for his encouragement and best wishes.

We would like to extend our sincere thanks to the **Management** of **S.B. Jain Institute of Technology, Management & Research** for providing all the necessary infrastructure and laboratory facilities.

We also like to acknowledge the help extended by the **faculty members and non-teaching staff** of Computer Science & Engineering Department for successful completion of our project.

Last but not the least, special thanks to our family members, friends and colleagues for their continuous support.

ABSTRACT

The usage of mobiles in today's world is more than ever. Mobile phones are everywhere and the mobile technology is growing at an exponential rate. The capabilities of a mobile phone have made it provide us services that make our lives better. One of the service that mobile phones can offer us is digital healthcare. Also, it is recognized that mobile applications that provide healthcare solutions are trending. Such applications provide a convenient and portable healthcare solution to all individuals. Such applications provide a better experience to a user and in this way, the users will come to know more about their health and body. Digital healthcare applications are capable of diagnosing a disease that a patient is suffering from using their symptoms. This information can be used further by a medical expert for later on consultation.

In this project we created Machine Learning Models comprises various Supervised ML classifiers like, Gradient Boosting, Decision Tree, Random Forest and Logistic Regression that have been used to deploy a model for disease prediction. It uses the existing datasets from the Kaggle database. The results generated impersonate that the Gradient Boosting classifier is achieving the highest accuracy score in such a way that prediction used by our model is of binary form in where 1 means a chance of having disease and 0 means no chance.

Key Words— Mobile Application, Machine Learning, Decision Trees, Logistic Regression, Random Forest Algorithm.

INDEX

LIST OF FIGURES

Chapter No 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. PROJECT BACKGROUND

In today's era, each and every human-being on earth depends on medical treatment and medicines. Every day we can hear some new diseases or new symptoms of the existing disease being discovered. But with the growing number of diseases and their symptoms, everyone cannot manage to be updated with it.

Cardio vascular disease is a widespread disease in all over a region. This type of disease may cause due to smoking, high blood pressure, diabetes, overweight, hyper tension, cholesterol etc. that has to be accumulated because of the fatty foods or unlimited intake of foods or non-moving to anywhere. Similarly, Diabetes is a widespread disease all over the world. Various factors like eating habits glucose levels, BMI, and other factors contribute towards the high risk of diabetes in people.

The availability of data and data mining techniques, especially machine learning and early detection of any disease, can help patients to anticipate a potential disease response. In the healthcare field, it is becoming more and more common nowadays to source large amounts of data (big data), streaming machines, advanced healthcare services, high throughput instruments, sensor networks, Internet of Things, mobile application applications, data archiving and processing, from many areas.

Python is most powerful programming language having numerous libraries which is used in this project with machine learning model. Machine learning is a subset model of artificial intelligence network in which uses complex algorithms and deep learning neural networks.

We are developing an android application "WellBe – A Health Prediction App" which has a list of number of diseases that predict the risk of heart attack or diabetes. All we need to do in this app is to add our symptoms the values of the required parameter and it predicts the high risk or low risk of the disease using a machine learning model deployed on our servers.

PROBLEM STATEMENT

Given the current conditions of the modern day world we encounter various diseases whose treatment can be started earlier if their detection is made easy. Thus we create a machine learning model which can detect the high or low risk of any disease. To provide a better availability to these models we create an android application to implement these models.

1.2. PROPOSED WORK

ML became the enhancing approach for the evolution of predictive models in health care industries and was decided to test various algorithms to check what extent their prediction scores estimate or ameliorate upon the results acquired. Researchers deploy various machine learning and data mining techniques over a set of enormous data of cardiovascular patients to attain the prediction for heart attacks before their occurrence for helping healthcare industries and professionals.

The availability of data and data mining techniques, especially machine learning and early detection of any disease, can help patients to anticipate a potential disease response. In the healthcare field, it is becoming more and more common nowadays to source large amounts of data (big data), streaming machines, advanced healthcare services, high throughput instruments, sensor networks, Internet of Things, mobile application applications, data archiving and processing, from many areas.

WellBe is a Prediction Application using Machine Learning Technique. Heart attack and Diabetes prediction will be found primarily on real-time processing, distributed and real-time classification and distribution, storage so; databases can be easily modified by the doctors. If you know all the attributes related to our health we can check easily how much chance to the Heart attack risk and Diabetes risk, using the system applications. It was recently used to train classification models. After that using extract the features that is condition to be find to be classified by Decision Tree (DT).Compared to existing; algorithms provides better performance. After classification, performance criteria including accuracy, precision is to be calculated. If you are concern about the heart attack risks, you might be referred to a specialist.

Some attributes are Heart Attack risk factors including which is the High blood pressure, high cholesterol and diabetes, increases your risk even more. Hence we are also checking your symptoms of heart attack and take about prevention. Various factors like eating habits glucose levels, BMI, and other factors contribute towards the high risk of diabetes in people.

Similar applications has been already made but it is not freely available in market. It is quite on the expensive side and requires more expensive hardware and high computation. Previously similar application is made in US, but they take money for that application and it is costlier.

Objectives

- To give early disease prediction mechanism.
- To get instant results of the risk factor of health.
- To provide convenience of an android application to check results any time.
- To provide accurate results using the ML models.

1.3. TECHNOLOGICAL BASE

This Project can be implemented by using various technologies like-

Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum from 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a batteries included" language due to its comprehensive standard library.

Features:

- Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read − Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain − Python's source code is fairly easy-to-maintain.
- Scalable − Python provides a better structure and support for large programs than shell scripting.

Flutter

Flutter is Google's SDK for crafting beautiful, fast user experiences for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source. We think Flutter will help you create beautiful, fast apps, with a productive, extensible and open development model, whether you're targeting iOS or Android, web, Windows, macOS, Linux or embedding it as the UI toolkit for a platform of your choice.

Features:

- Beautiful user experiences − Flutter's layered architecture gives you control over every pixel on the screen and its powerful compositing capabilities let you overlay and animate graphics, video, text, and controls without limitation.
- Fast results − Flutter is fast. It's powered by the same hardware-accelerated 2D graphics library that underpins Chrome and Android: Skia.
- Productive development − Flutter offers stateful hot reload, allowing you to make changes to your code and see the results instantly without restarting your app or losing its state.
- Extensible and open model − Flutter works with any development tool (or none at all), and also includes editor plug-ins for both Visual Studio Code and IntelliJ / Android Studio. Flutter provides tens of thousands of packages to speed your development, regardless of your target platform. And accessing other native code is easy, with support for both FFI and platform-specific APIs.

Chapter No 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1. RELATED WORKS

- **S. Suraj Kumar Gupta et al.** presents in his research paper intends to ideate the prediction for probabilities of occurrence of a heart attack in the patients. These classifiers have been deployed in pipeline approach of machine learning to attain the prediction using both ways i.e., without optimizations and feature transformations as well as vice-versa. The results impersonate that the Gradient Boosting classifier is achieving the highest accuracy score in such a way that prediction used by our model is of binary form in where 1 means a chance of heart attack and 0 means no chance. Some of the most influential attributes are chest pain type among which the typical angina is the most influential and asymptotic chest pain is least, cholesterol level in which the level greater than 200mg/dl are more prone, increased heart rate, thal, and age. The medical checkup of Blood-pressure level, cholesterol level and heart rate on daily basis along with meditation can help you prevent the major heart attacks. [1]

- **Naveenkumar S. et al.** provides the system based on predictive modelling, predicts the disease of patients/users on the basis of the symptoms that the user provides symptoms as an input to the system. The device analyses the symptoms given by the user/patient as input and provides the likelihood of the disease as output based on the prediction using the algorithm. Smart health predictions are made by the implementation of the Naïve Bayes Classifier. The Naïve Bayes Classifier measures the disease percentage probability by considering all its features that is trained during the training phase. Exact interpretation of disease data benefits early patient/user disease prediction and provides clear vision about the disease to the user. After a prediction, the user/patient can consult a specialist doctor. It uses machine learning algorithms and database management techniques to extract new patterns from historical data. The Forecast Accuracy can improve with the use of a machine learning algorithm and the user/patient will get fast and easy access to the application. [2]

- **Ramandeep Singh Sethi et al.** proposed the usage of mobile phones in today's world is more than ever. Mobile phones are everywhere and the mobile technology is growing at an exponential rate. One such service that mobile phones can offer us is digital healthcare. Also, it is recognized that mobile phone applications that provide healthcare solutions are trending. Such applications provide a convenient and portable healthcare solutions to all the individuals. Such applications provide a rich experience to a user and in this way, the users will come to know more about their health and body. Digital healthcare mobile applications are capable of diagnosing a disease that a patient is suffering from using his/her symptoms. This information can be used further by a medical practitioner for later on consultation. [3]

- **Mayuri Asabe et al.** proposed prediction using Machine Learning Technique in Big data analytics has started to play an important role in the healthcare practices and research. Disease prediction will be found primarily on real-time processing, distributed and real-time classification and distribution, storage so; databases can be easily modified by the doctors. If you know all the attributes related to our health we can check easily how much chance to the Heart attack risk, using the system applications. It was recently used to train classification models. After that using extract the features that is condition to be find to be classified by Decision Tree (DT).Compared to existing; algorithms provides better performance. After classification, performance criteria including accuracy, precision, F-measure is to be calculated. If you are concern about the heart attack risks, you might be referred to a heart specialist. [4]

- **Prashant Tiwari et al.** is proposing an android app that allows user to get instant guidance on their health issues through an intelligent health care system. The system is fed with different symptoms and the disease associated with those symptoms. The system allows user to share their symptoms and issues. Here we use some intelligent data mining techniques to guess the most accurate illness that can be associated with patient’s symptoms. The patients can easily identify the disease by just imputing their issues and the software generates what disease he/she may be infected from. The system will prove helpful in urgent cases where the patient is unable to reach hospital or in cases when there are no doctors available in the area. [5]

Chapter No 3

METHODOLOGY / PROPOSED SOLUTION

CHAPTER 3

METHODOLOGY / PROPOSED SOLUTION

3.1. PROPOSED SOLUTION

The intent of our project is that the users should be able to take health detection test easily and quickly. Our application will be a simple android application that will be user friendly. The principle of the application is to fetch the results from the ML model deployed on servers and give accurate results as quickly as possible. After the user gives the input our application shows a screen with high or low risk to inform the user about his health issues.

Modules:

- **User :-**

1. **Dataset Gathering and Pre-processing :**
    - We have used Kaggle dataset for training of our model. The dataset consists of approx. 10000 different test cases for each type of disease.
2. **Model training :**
    - The dataset is fed to the model for training and the test cases are run to create the output. The test cases are then verified and the accuracy is checked and improved.
3. **App design and Deployment :**
    - We have developed the application using flutter framework. Here user can select the type of model in the main menu and then goes to the input form where user can enter the data required and press button to get results.
    - The models are deployed using flask framework on Heroku servers and the APIs are created to fetch the results to the app from the servers.

3.2. SYSTEM ARCHITECTURE

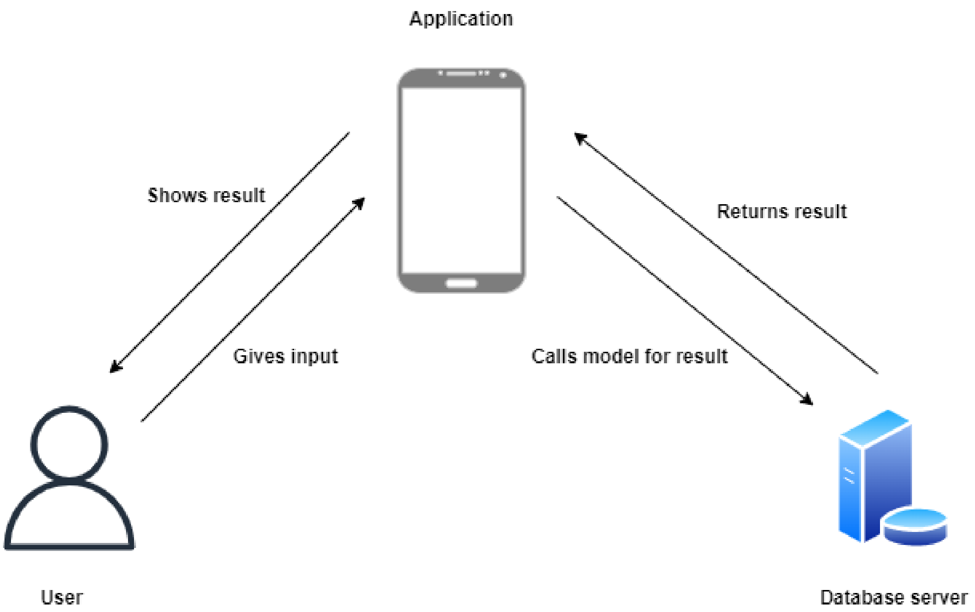


Fig 3.1 System Architecture

The primary focus of the android application is to provide user an interface to get an instant result of the test of disease. We have achieved this by creating model for the type of diseases. The user just have to provide their vitals or information and the model will generate the results.

The above architecture describes the complete view of the app and its usage in the perfect manner. It also shows the working of all the actors.

3.3. FLOW CHART

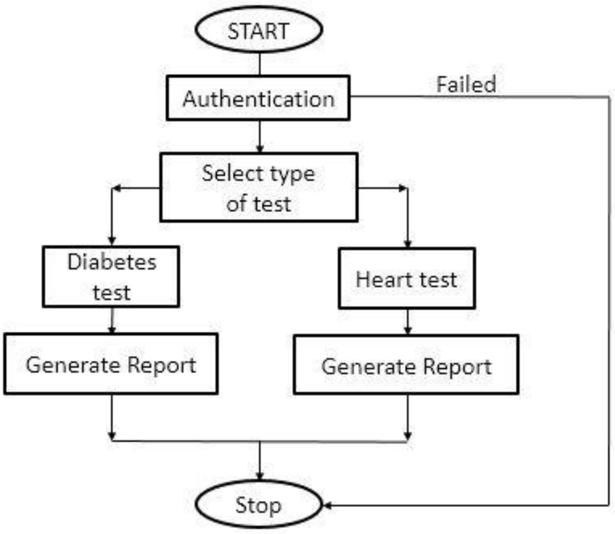


Fig 3.2 Flow chart

Above we have shown the flow of our Android application in the project. In this, the user just have to provide their data and the model will generate the results of user.

Chapter No 4

TOOLS/PLATFORM

CHAPTER 4

TOOLS/PLATFORM

4.1 SOFTWARE REQUIREMENT

a) **CLIENT SIDE TECHNOLOGY:** Flutter & Dart

b) **SERVER SIDE TECHNOLOGY:** Python and Flask

c) **IDE:** Visual Studio, Jupyter Notebook.

d) **LIBRARIES:** NumPy, Matplotlib, Pandas, SciKit-Learn.

e) **WEB SERVER:** Heroku, Firebase

f) **DATABASE:** Firebase,

g) **OPERATING SYSTEM:** Windows 10

h) **DESIGNING TOOLS:** Draw.io

i) **TESTING TOOL:** MS-EXCEL

a) CLIENT SIDE TECHNOLOGY :

- **Flutter and Dart:**

  Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. First described in 2015, Flutter was released in May 2017.

b) SERVER SIDE TECHNOLOGY:

- **Flask**:

  Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper and has become one of the most popular Python web application frameworks.

  Features:

  - Built-in development server, fast debugger.
  - Integrated support for unit testing.
  - RESTful request dispatching.

- **Python:**

  Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a battery included language due to its comprehensive standard library.

- **Features:**
  - Easy to code
  - Free and Open Source
  - Object-Oriented Language
  - High-Level Language
  - Dynamically Typed Language
  - Interpreted Language

c) IDE:

- **VSCode:**

  Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

d) LIBRARIES:

- **NumPy :**

  NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

- **Matplotlib :**

  Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

- **Scikit-Learn :**

  Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. Please note that sklearn is used to build machine learning models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.)

e) WEB SERVER:

- **Heroku:**

  Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages.

- **Firebase:**

  Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development.

f) DATABASE:

The database is of NoSql that is supported by the firebase database services.

g) OPERATING SYSTEM:

Any Operating System (preferably windows 7 or higher) which is having architecture of 64-bit is supported. We have used Windows 10 64-bit.

h) DESIGNING TOOLS:

- **Draw.io**

  Draw.io is completely free online diagram editor built around Google Drive(TM) that enables you to create flowcharts, UML, entity relation, network diagrams, mockups and more.

- **Features**:
  - Collaboration Tools
  - Data Import/Export
  - Drag & Drop Interface
  - Third Party Integration

i) TESTING TOOLS:

For testing purpose we used MS Excel to observe the results of application.

4.2 HARDWARE REQUIREMENT

PROCESSOR: Min. Intel i3 and above.

HARD DISK: Min. 128 GB.

RAM: Min 512 Mb and above

Chapter No 5

DESIGN & IMPLEMENTATION

CHAPTER 5

DESIGN & IMPLEMENTATION

5.1. SYSTEM DESIGN

5.1.1. USE CASE DIAGRAM

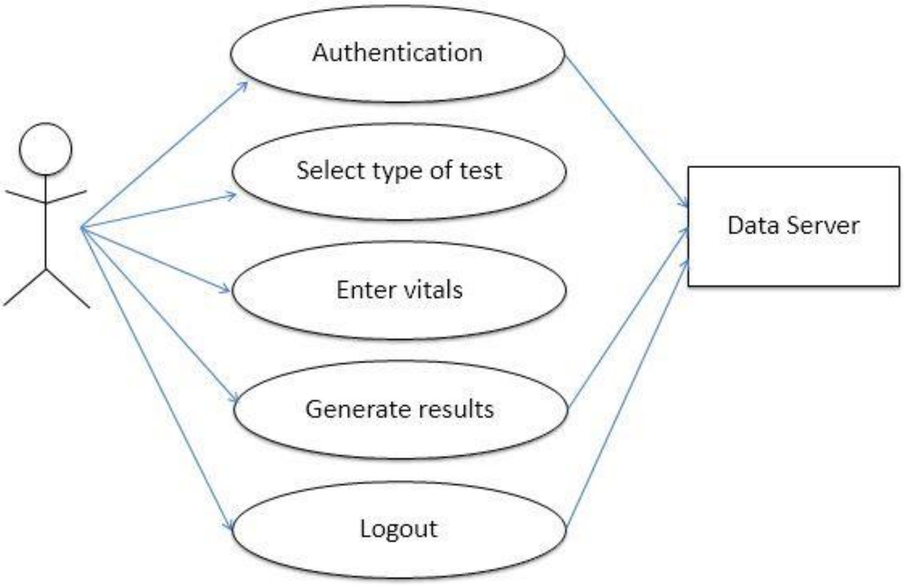


Fig 5.1 Use case diagram

The above use case diagram simply depicts the task which a user can perform through our application.

These tasks are as follows:

- User can register and login to the application.
- User can select any model and enter their vitals.
- User can view the final result.

5.1.2. CLASS DIAGRAM

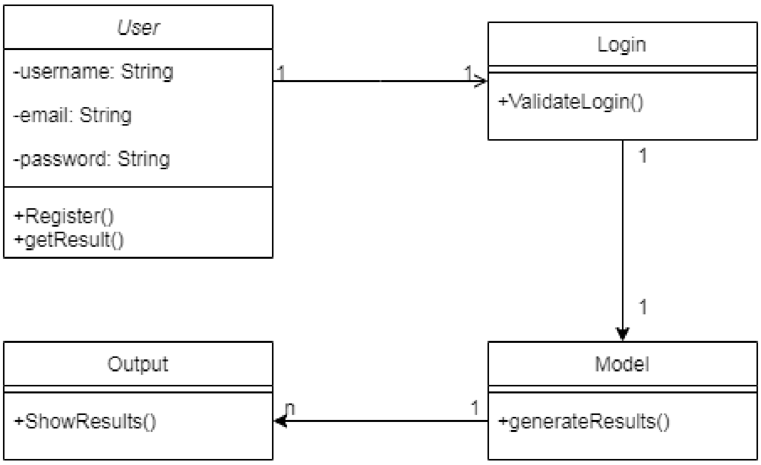


Fig 5.2 Class Diagram

In the above class diagram, the whole working of our application is represented in a certain manner. The user class is responsible for registration and validation of login info as well as sending data provided by the user to server and retrieve the final results. The login class is responsible for the systematic logging of the user in the android application. The Model class will generate the results based on the input.

5.1.2. SEQUENCE DIAGRAM

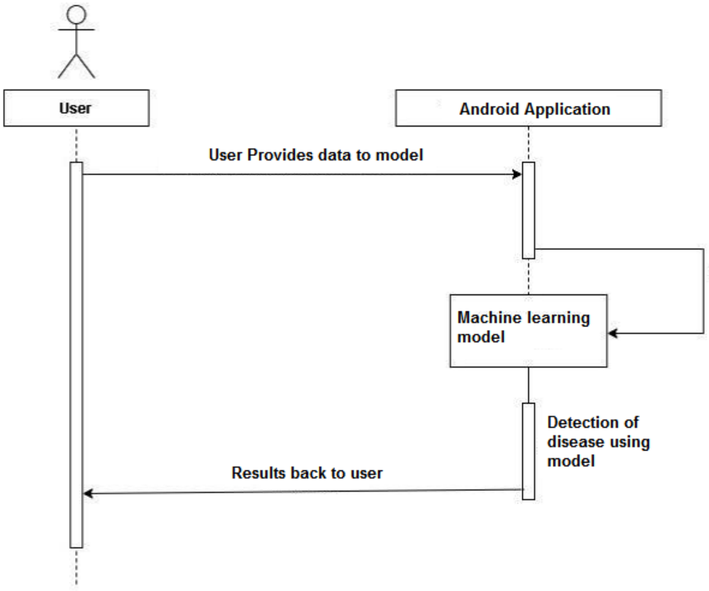


Fig 5.3 Sequence Diagram

1) Providing input: After logging in into the android application, the user will be required to select the type of model and provide the necessary inputs to the form.
2) Detection of disease: In this, the data given by the user will be processed and the high or low risk is found out and result of these will be sent to user.
3) Results: After obtaining the results from the model the app shows it using a dialog box.

5.2. IMPLEMENTED MODULES

We have completed the Dataset Gathering and Processing, Model training. We also completed the GUI Designing implementation using Flutter and the deployment using Flask. In this module, we have completed designing the user interface of android application.

Dataset Gathering and Pre-processing: We have used Kaggle dataset for training of our model. The dataset consists of approx. 10000 different test cases for each type of disease. A dataset is collected and preprocessed accordingly for training and testing purpose of the model.

Model Training: The dataset is fed to the model for training and the test cases are run to create the output. The test cases are then verified and the accuracy is checked and improved.

GUI Design and Deployment: We have developed the application using flutter framework. Here user can select the type of model in the main menu and then goes to the input form where user can enter the data required and press button to get results.
The models are deployed using flask framework on Heroku servers and the APIs are created to fetch the results to the app from the servers.

5.3. SAMPLE CODE

Python script to write API calls to the diabetes model:

```
from flask import Flask, jsonify
from requests import request
from flask import request
import pickle
import numpy as np

model = pickle.load(open('model1.pkl','rb'))

app = Flask(__name__)

@app.route('/')
def home():
   return "Welcome to Diabetic predictions"

@app.route('/pre',methods=["POST"])
def pre():
   Pregnancies=request.form.get('Pregnancies')
   Glucose=request.form.get('Glucose')
   BloodPressure=request.form.get('BloodPressure')
   SkinThickness=request.form.get('SkinThickness')
   Insulin=request.form.get('Insulin')
   BMI=request.form.get('BMI')
   DiabetesPedigreeFunction=request.form.get('DiabetesPedigreeFunction')
   Age=request.form.get('Age')



input_query=np.array([[Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age]])

   result =model.predict(input_query)[0]
```

```
    return jsonify({'person is diabatic':str(result)})

if __name__=='__main__':
    app.run(debug=True)
```

Python script to write API calls to the heart disease model:

```
from flask import Flask, jsonify
from requests import request
from flask import request
import pickle
import numpy as np

model = pickle.load(open('model2.pkl','rb'))

app = Flask(__name__)

@app.route('/')
def best():
    return "Welcome to Cardic Arrest predictions"

@app.route('/heart',methods=["POST"])
def heart():
    age =request.form.get(' age ')
    sex=request.form.get('sex')
    cp=request.form.get('cp')
    trestbps=request.form.get('trestbps')
    chol=request.form.get('chol')
    fbs=request.form.get('fbs')
    restecg=request.form.get('restecg')
    thalach =request.form.get('thalach')
    exang =request.form.get('exang')
    oldpeak =request.form.get('oldpeak')
    slope =request.form.get('slope')
    ca =request.form.get('ca')
    thal =request.form.get('thal')


input_query=np.array([[age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,th
al]])

    result =model.predict(input_query)[0]

    return jsonify({'person is ':str(result)})

if __name__=='__main__':
    app.run(debug=True)
```

Chapter No 6

TESTING, RESULTS & DISCUSSION

CHAPTER 6

TESTING, RESULTS & DISCUSSION

6.1. TESTING

6.1.1 TYPES OF TESTING

- **Manual Testing**

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

Following are the testing techniques that are performed manually during the test life cycle:

  - Acceptance Testing
  - White Box Testing
  - Black Box Testing
  - Unit Testing
  - System Testing
  - Integration Testing

- **Automation Testing**

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.

Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view.

Test Automation should be used by considering the following aspects of a software:

- Large and critical projects
- Projects that require testing the same areas frequently
- Requirements not changing frequently
- Accessing the application for load and performance with many virtual users
- Stable software with respect to manual testing
- Availability of time

6.1.2 LEVELS OF TESTING

There are four levels of testing: Unit, Integration, System and Acceptance.

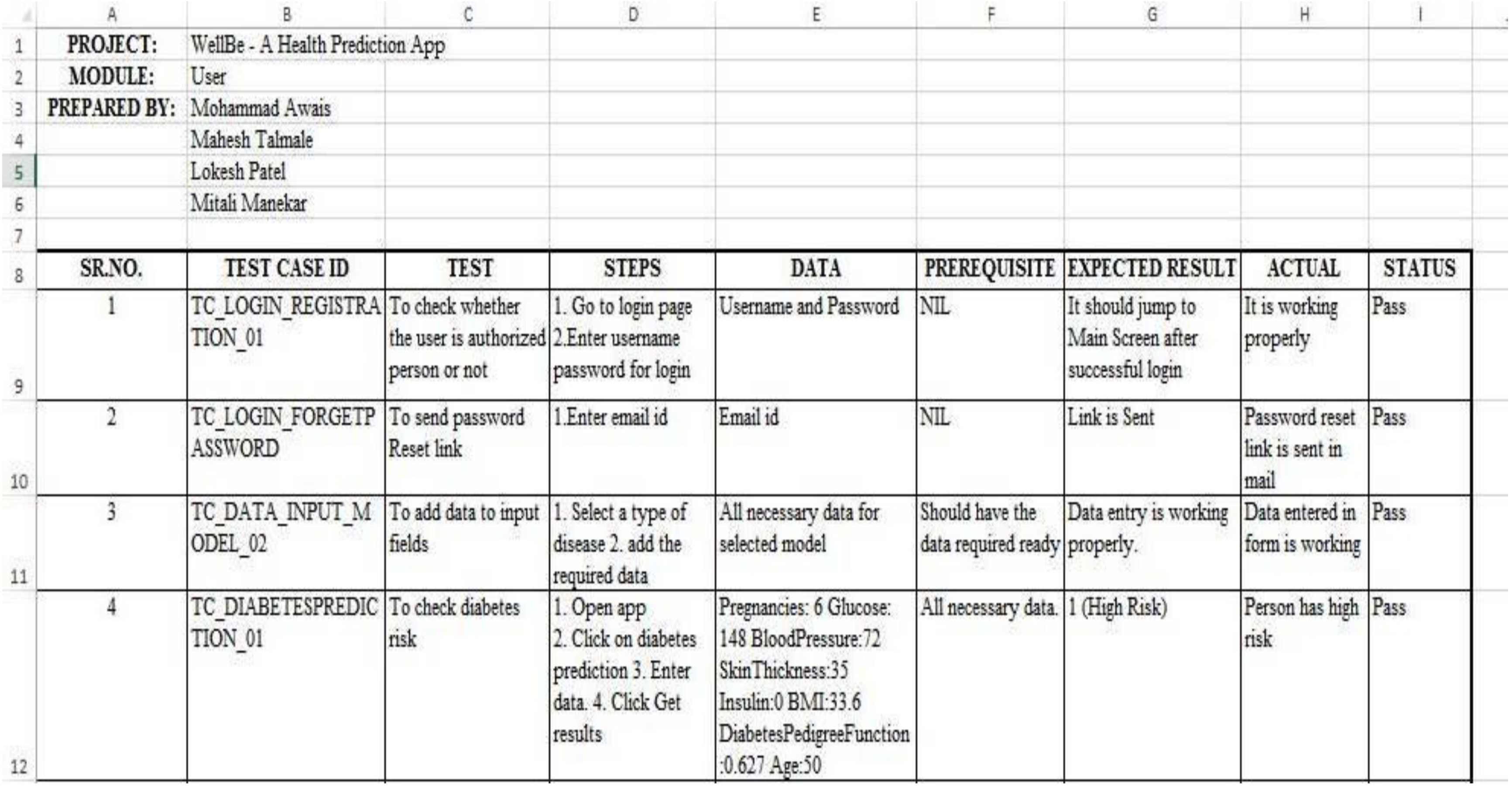
1. **Unit Testing:** A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

2. **Integration Testing:** A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units
.

3. **System Testing:** A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.

4. **Acceptance Testing:** A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery**.**

6.1.3 TESTING REPORT

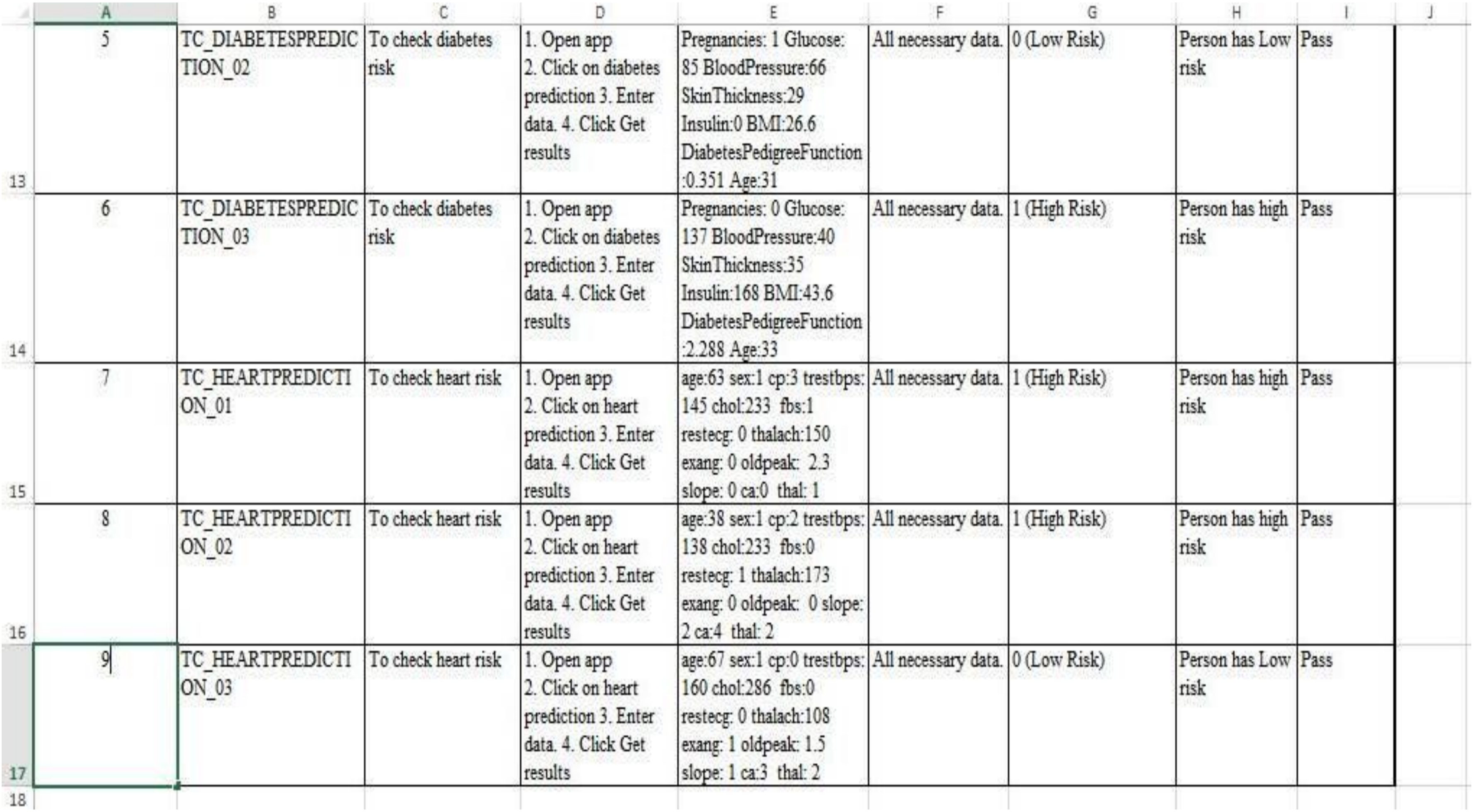


PROJECT: WellBe - A Health Prediction App

MODULE: User

PREPARED BY: Mohammad Awais
Mahesh Talmale
Lokesh Patel
Mitali Manekar

SR.NO.	TEST CASE ID	TEST	STEPS	DATA	PREREQUISITE	EXPECTED RESULT	ACTUAL	STATUS
1	TC_LOGIN_REGISTRATION_01	To check whether the user is authorized person or not	1. Go to login page 2.Enter username password for login	Username and Password	NIL	It should jump to Main Screen after successful login	It is working properly	Pass
2	TC_LOGIN_FORGETPASSWORD	To send password Reset link	1.Enter email id	Email id	NIL	Link is Sent	Password reset link is sent in mail	Pass
3	TC_DATA_INPUT_MODEL_02	To add data to input fields	1. Select a type of disease 2. add the required data	All necessary data for selected model	Should have the data required ready	Data entry is working properly.	Data entered in form is working	Pass
4	TC_DIABETESPREDICTION_01	To check diabetes risk	1. Open app 2. Click on diabetes prediction 3. Enter data. 4. Click Get results	Pregnancies: 6 Glucose: 148 BloodPressure:72 SkinThickness:35 Insulin:0 BMI:33.6 DiabetesPedigreeFunction :0.627 Age:50	All necessary data.	1 (High Risk)	Person has high risk	Pass



	A	B	C	D	E	F	G	H	I	J
13	5	TC_DIABETESPREDICTION_02	To check diabetes risk	1. Open app 2. Click on diabetes prediction 3. Enter data. 4. Click Get results	Pregnancies: 1 Glucose: 85 BloodPressure:66 SkinThickness:29 Insulin:0 BMI:26.6 DiabetesPedigreeFunction :0.351 Age:31	All necessary data.	0 (Low Risk)	Person has Low risk	Pass	
14	6	TC_DIABETESPREDICTION_03	To check diabetes risk	1. Open app 2. Click on diabetes prediction 3. Enter data. 4. Click Get results	Pregnancies: 0 Glucose: 137 BloodPressure:40 SkinThickness:35 Insulin:168 BMI:43.6 DiabetesPedigreeFunction :2.288 Age:33	All necessary data.	1 (High Risk)	Person has high risk	Pass	
15	7	TC_HEARTPREDICTION_01	To check heart risk	1. Open app 2. Click on heart prediction 3. Enter data. 4. Click Get results	age:63 sex:1 cp:3 trestbps: 145 chol:233 fbs:1 restecg: 0 thalach:150 exang: 0 oldpeak: 2.3 slope: 0 ca:0 thal: 1	All necessary data.	1 (High Risk)	Person has high risk	Pass	
16	8	TC_HEARTPREDICTION_02	To check heart risk	1. Open app 2. Click on heart prediction 3. Enter data. 4. Click Get results	age:38 sex:1 cp:2 trestbps: 138 chol:233 fbs:0 restecg: 1 thalach:173 exang: 0 oldpeak: 0 slope: 2 ca:4 thal: 2	All necessary data.	1 (High Risk)	Person has high risk	Pass	
17	9	TC_HEARTPREDICTION_03	To check heart risk	1. Open app 2. Click on heart prediction 3. Enter data. 4. Click Get results	age:67 sex:1 cp:0 trestbps: 160 chol:286 fbs:0 restecg: 0 thalach:108 exang: 1 oldpeak: 1.5 slope: 1 ca:3 thal: 2	All necessary data.	0 (Low Risk)	Person has Low risk	Pass	
18										

6.2. RESULTS AND DISCUSSION

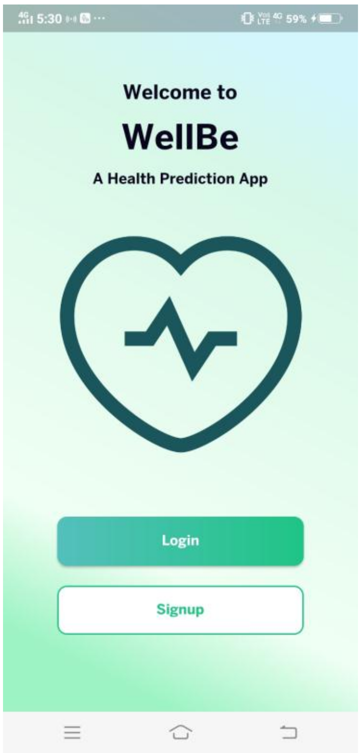
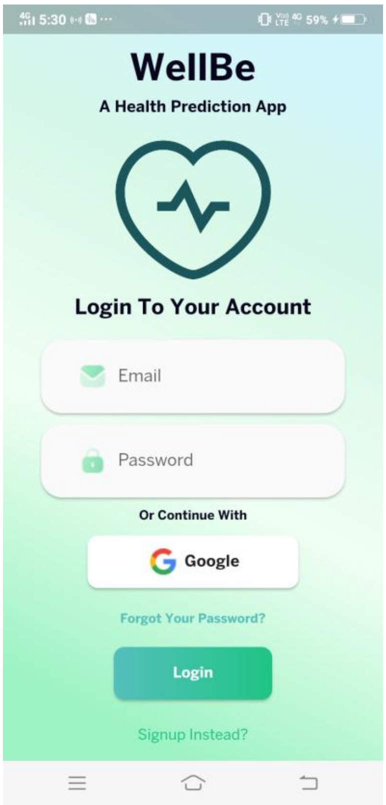


Fig 6.1 App Screen

This is the first screen that appears after opening the app.



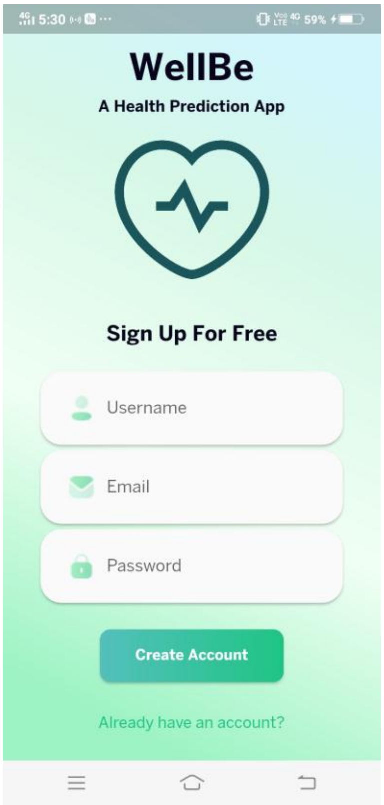
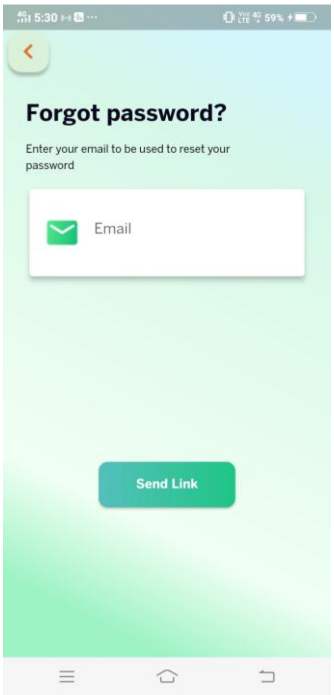


Fig 6.2 Login Screen **Fig 6.3 Signup Screen**

These are the login and signup pages of the app.

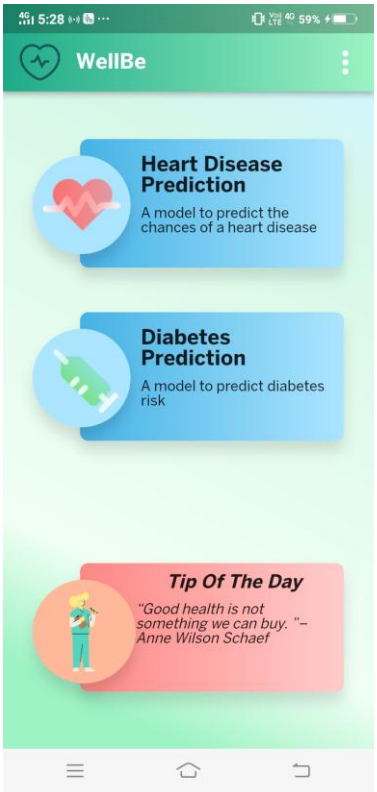


Forgot password?

Enter your email to be used to reset your password

Fig 6.4 Forget Password Screen

The forget password screen of the app. User enters the email and the link for resetting password is sent.



Heart Disease Prediction

A model to predict the chances of a heart disease

Diabetes Prediction

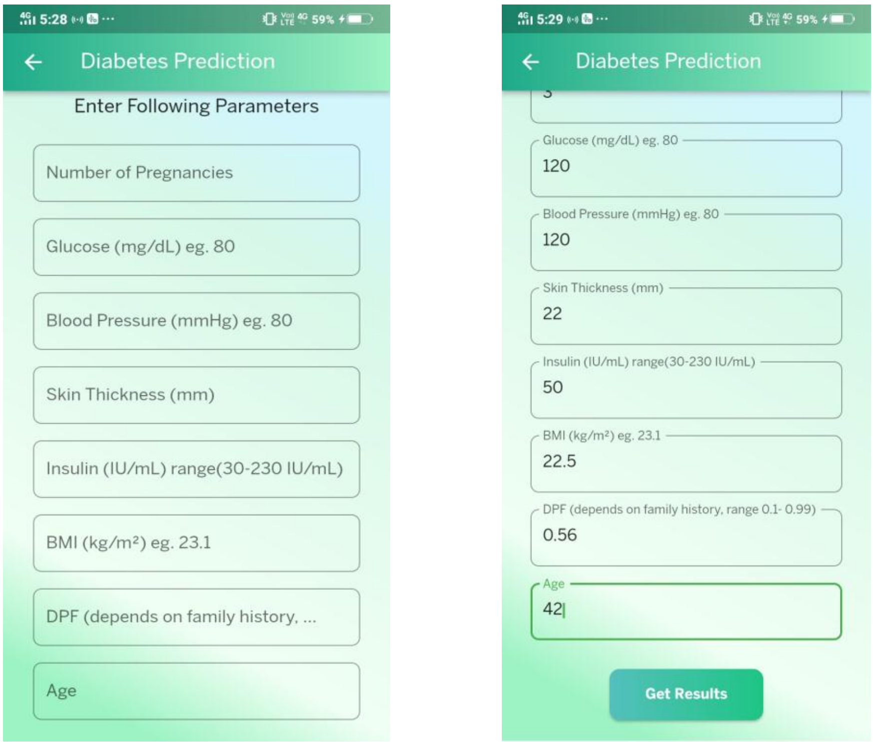
A model to predict diabetes risk

Tip Of The Day

"Good health is not something we can buy. "– Anne Wilson Schaef

Fig 6.5 Main Screen

This is the main screen of the app. User can select a type of disease and it takes us to the input form of that disease.



Diabetes Prediction

Enter Following Parameters

Number of Pregnancies

Glucose (mg/dL) eg. 80

Blood Pressure (mmHg) eg. 80

Skin Thickness (mm)

Insulin (IU/mL) range(30-230 IU/mL)

BMI (kg/m²) eg. 23.1

DPF (depends on family history, ...

Age

Diabetes Prediction

Glucose (mg/dL) eg. 80: 120

Blood Pressure (mmHg) eg. 80: 120

Skin Thickness (mm): 22

Insulin (IU/mL) range(30-230 IU/mL): 50

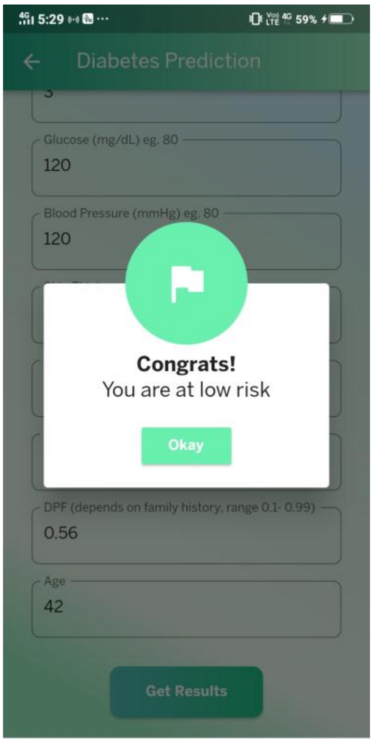
BMI (kg/m²) eg. 23.1: 22.5

DPF (depends on family history, range 0.1- 0.99): 0.56

Age: 42

Fig 6.6 Form Input Screens

This is the input form for diabetes prediction model. User has to enter their details and press get results for their test.



Diabetes Prediction

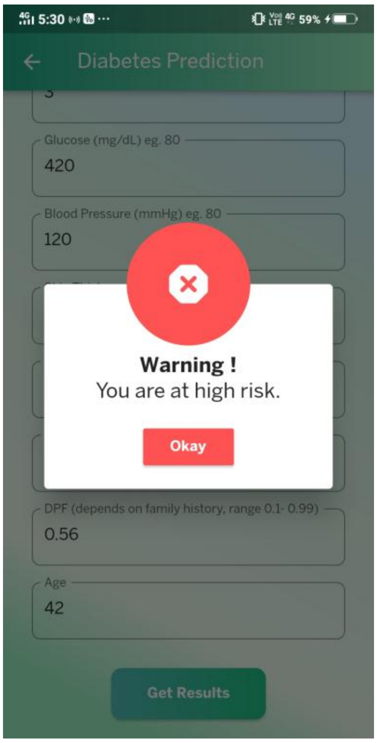
Glucose (mg/dL) eg. 80: 120

Blood Pressure (mmHg) eg. 80: 120

Congrats!
You are at low risk

DPF (depends on family history, range 0.1- 0.99): 0.56

Age: 42



Diabetes Prediction

Glucose (mg/dL) eg. 80: 420

Blood Pressure (mmHg) eg. 80: 120

Warning !
You are at high risk.

DPF (depends on family history, range 0.1- 0.99): 0.56

Age: 42

Fig 6.7 Low risk Screen **Fig 6.8 High risk Screen**

These are the output screens. It depends upon the result of the data entered.

Chapter No 7

ADVANTAGES AND APPLICATIONS

CHAPTER 7

ADVANTAGES AND APPLICATIONS

7.1. ADVANTAGES

1. **Time saving and efficient :**
   - This will save the time of users and also the money that they put in various tests which sometimes don’t give reports on time.

2. **Highly Accurate :**
   - Use of Machine learning enables significant knowledge, e.g., relationships between medical factors related to heart disease and patterns, to be established.
   - With more users using the app, the results generated can be more accurate over time

3. **Easy to use :**
   - The app created is user friendly and easy to use. Thus anybody can check their results easily.

7.2. APPLICATIONS

1. **Early detection of disease:**
   - Users can check any symptoms for their health risks before consulting a doctor.
2. **Easy for doctors to understand the test report generated:**
   - A detailed description of their health issues can be discussed with a doctor with a report generated by the vitals collected during the test in the app.
   - Hospitals can promote the use of this app for their diagnosis and getting the symptoms before the patients visits them to save time and cross check their diagnosis.
3. **Multiple Models supported in single App:**
   - Multiple disease detection in a single app can help the users and can become as a regular health checkup solution.

     .

Chapter No 8

CONCLUSION & FURTHER SCOPE

CHAPTER 8

CONCLUSION & FURTHER SCOPE

8.1. CONCLUSION

In this project, we have designed an android application and completed its development by applying **engineering knowledge** which provides an approach in building a platform where users can check results for their diseases. It solved the **societal problem** of users with a quick way to check health. We have identified and analyze problem in current world where diseases are increasing day by day, hence we found the solution by developing an android application which take input of the various parameters of the health factors, and then finally gives Output as high risk or low risk. We have used **modern tools** like Flutter and libraries like Scikitlearn, NumPy, Matplotlib, etc to implement this project. During the development of the project, we understood the importance of individual and **teamwork** while **project development and management**. While showcasing our project through various seminars and conference we enhanced our **communication skills** and displayed **professional ethics** which results in **lifelong learning**.

8.2. FURTHER SCOPE

1. More models such as kidney disease detection and breast cancer detection can be added to the app and app can be made as a one stop destination for the various types of disease detection.

2. A doctor consultation system can be added to the app where the user can consult any doctor for the reports generated in the app.

3. Various hospitals can use this app to generate reports at a faster level.

4. More analytics and a detailed report version can be created to give visual aids by graphs and charts.

5. Previous test results can be stored for a user and the improving or declining health can be tracked using graphs and suggestions can be given to the user for a better lifestyle.

REFERENCES

PAPERS

1. S. Suraj Kumar Gupta, Aditya Shrivastava, S. P. Upadhyay, Pawan Kumar Chaurasia, “A Machine Learning Approach for Heart Attack Prediction”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-10 Issue-6, August 2021

2. Naveenkumar S , Kirubhakaran R , Jeeva G , Shobana M , Sangeetha K, “Smart Health Prediction Using Machine Learning”, Special Issue of Second International Conference on Advancements in Research and Development (ICARD 2021), Volume 03 Issue 03S March 2021

3. Ramandeep Singh Sethi, Aniket Thumar, Vaibhav Jain, Sachin Chavan, “Disease Prediction Application based on Symptoms”, International Journal of Scientific Research in Computer Science Engineering and Information Technology, DOI:10.32628/CSEIT1952189 , March 2019.

4. Mayuri Asabe, Shweta Shilwant, Nagnath Dolare, Sulakshana Chorghade, K. R. Pathak, “Heart Attack Prediction and Analysis System Using Decision Tree Algorithm”, HBRP Publications, Advancement in Image Processing and Pattern Recognition Volume 3 Issue 2.

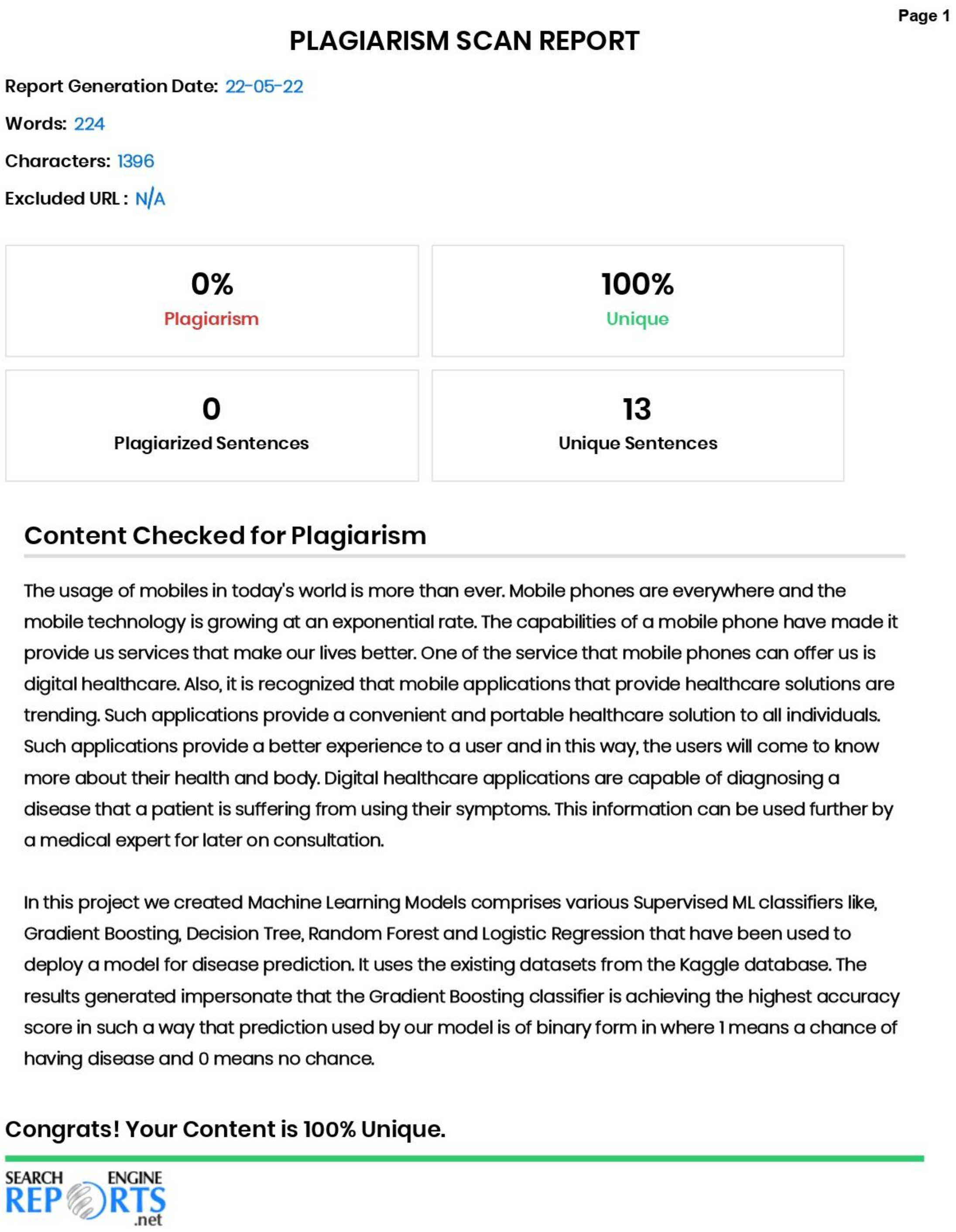
5. Prashant Tiwari, Aman Jaiswal, Narendra Vishwakarma, Pushpanjali Patel, “SMART HEALTH CARE (AN ANDROID APP TO PREDICT DISEASE ON THE BASIS OF SYMPTOMS)”, International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 04 | Apr -2017.

WEBSITES

1. **https://www.researchgate.net/**
2. **https://www.ijsdr.org/**
3. **https://www.irjet.net/**

APPENDIX I

PLAGIARISM REPORT



Page 1

PLAGIARISM SCAN REPORT

Report Generation Date: 22-05-22

Words: 224

Characters: 1396

Excluded URL : N/A

0% Plagiarism	100% Unique
0 Plagiarized Sentences	13 Unique Sentences

Content Checked for Plagiarism

The usage of mobiles in today's world is more than ever. Mobile phones are everywhere and the mobile technology is growing at an exponential rate. The capabilities of a mobile phone have made it provide us services that make our lives better. One of the service that mobile phones can offer us is digital healthcare. Also, it is recognized that mobile applications that provide healthcare solutions are trending. Such applications provide a convenient and portable healthcare solution to all individuals. Such applications provide a better experience to a user and in this way, the users will come to know more about their health and body. Digital healthcare applications are capable of diagnosing a disease that a patient is suffering from using their symptoms. This information can be used further by a medical expert for later on consultation.

In this project we created Machine Learning Models comprises various Supervised ML classifiers like, Gradient Boosting, Decision Tree, Random Forest and Logistic Regression that have been used to deploy a model for disease prediction. It uses the existing datasets from the Kaggle database. The results generated impersonate that the Gradient Boosting classifier is achieving the highest accuracy score in such a way that prediction used by our model is of binary form in where 1 means a chance of having disease and 0 means no chance.

Congrats! Your Content is 100% Unique.

SEARCH ENGINE REPORTS .net

APPENDIX II

Instructional Manual

On

“ WellBe – A Health Prediction App”

Submitted By

Mr. Mohammad Awais

Mr. Mahesh Talmale

Mr. Lokesh Patel

Mr. Mitali Manekar

Under the Guidance of

Mr. Sameer Mendhe

Department of Computer Science & Engineering

S. B. Jain Institute of Technology Management and Research
Nagpur-441501

2020-2021

1. Abstract

The usage of mobiles in today's world is more than ever. Mobile phones are everywhere and the mobile technology is growing at an exponential rate. The capabilities of a mobile phone have made it provide us services that make our lives better. One of the service that mobile phones can offer us is digital healthcare. Also, it is recognized that mobile applications that provide healthcare solutions are trending. Such applications provide a convenient and portable healthcare solution to all individuals. Such applications provide a better experience to a user and in this way, the users will come to know more about their health and body. Digital healthcare applications are capable of diagnosing a disease that a patient is suffering from using their symptoms. This information can be used further by a medical expert for later on consultation.

In this project we created Machine Learning Models comprises various Supervised ML classifiers like, Gradient Boosting, Decision Tree, Random Forest and Logistic Regression that have been used to deploy a model for disease prediction. It uses the existing datasets from the Kaggle database. The results generated impersonate that the Gradient Boosting classifier is achieving the highest accuracy score in such a way that prediction used by our model is of binary form in where 1 means a chance of having disease and 0 means no chance.

Key Words— Mobile Application, Machine Learning, Decision Trees, Logistic Regression, Random Forest Algorithm.

2. Modules Implemented

1. **Dataset Gathering and Pre-processing :**
   - We have used Kaggle dataset for training of our model. The dataset consists of approx. 10000 different test cases for each type of disease.
2. **Model training :**
   - The dataset is fed to the model for training and the test cases are run to create the output. The test cases are then verified and the accuracy is

checked and improved.

3. **App design and Deployment :**
   - We have developed the application using flutter framework. Here user can select the type of model in the main menu and then goes to the input form where user can enter the data required and press button to get results.
   - The models are deployed using flask framework on Heroku servers and the APIs are created to fetch the results to the app from the servers.

3. Software and Hardware Requirement

3.1 Software Requirement:

a) **CLIENT SIDE TECHNOLOGY:** Flutter & Dart

b) **SERVER SIDE TECHNOLOGY:** Python and Flask

c) **IDE:** Visual Studio, Jupyter Notebook.

d) **LIBRARIES:** NumPy, Matplotlib, Pandas, SciKit-Learn.

e) **WEB SERVER:** Heroku, Firebase

f) **DATABASE:** Firebase,

g) **OPERATING SYSTEM:** Windows 10

h) **DESIGNING TOOLS:** Draw.io

i) **TESTING TOOL:** MS-EXCEL

a) CLIENT SIDE TECHNOLOGY :

- **Flutter and Dart:**

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. First described in 2015, Flutter was released in May 2017.

b) SERVER SIDE TECHNOLOGY:

- **Flask**:

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Features:

- Built-in development server, fast debugger.
- Integrated support for unit testing.
- RESTful request dispatching.

- **Python:**

  Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a battery included language due to its comprehensive standard library.

- **Features:**
  - Easy to code
  - Free and Open Source
  - Object-Oriented Language
  - High-Level Language
  - Dynamically Typed Language
  - Interpreted Language

c) IDE:

- **VSCode:**

  Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

d) LIBRARIES:

- **NumPy :**

  NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

- **Matplotlib :**

  Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

- **Scikit-Learn :**

  Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. Please note that sklearn is used to build machine learning models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.)

e) WEB SERVER:

- **Heroku:**

  Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages.

- **Firebase:**

  Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development.

f) DATABASE:

The database is of NoSql that is supported by the firebase database services.

g) OPERATING SYSTEM:

Any Operating System (preferably windows 7 or higher) which is having architecture of 64-bit is supported. We have used Windows 10 64-bit.

h) DESIGNING TOOLS:

- **Draw.io**

  Draw.io is completely free online diagram editor built around Google Drive(TM) that enables you to create flowcharts, UML, entity relation, network diagrams, mockups and more.

- **Features**:
  - Collaboration Tools
  - Data Import/Export
  - Drag & Drop Interface
  - Third Party Integration

i) TESTING TOOLS:

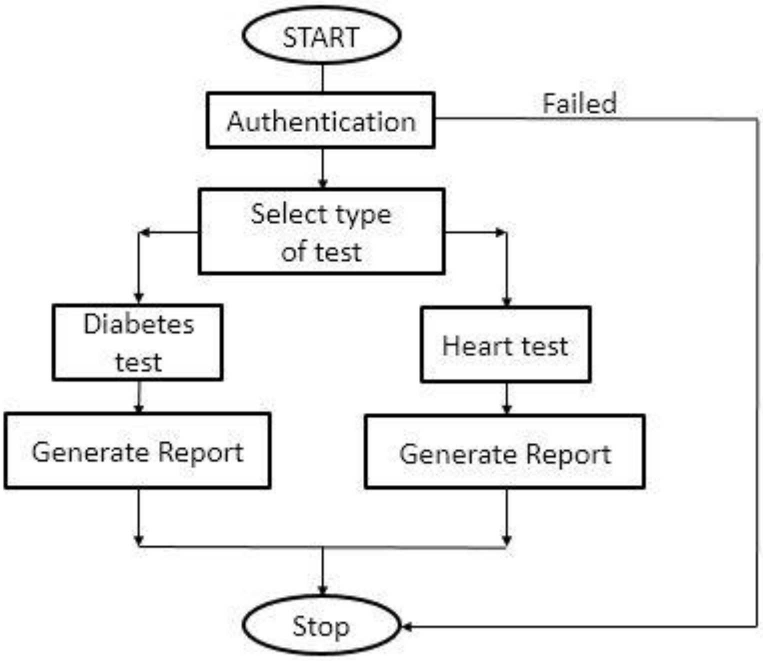
For testing purpose we used MS Excel to observe the results of application.

3.2 Hardware Requirement:

PROCESSOR: Min. Intel i3 and above.
HARD DISK: Min. 128 GB.

RAM: Min 512 Mb and above

3.3 Flowchart:



4. Steps to Run the Project

1. Install the app.
2. Firstly, the user needs to register then login using id and password to our WellBe application. (E.g. I'd : test@gmail.com and password : 12345678)
3. After successful login, the user will be able to access the application.
4. Now, the user needs to select the type of model which predicts the disease.
5. After selecting user will be taken to an input form where user can enter vitals.
6. After entering the data user should press on "Get Results" button.
7. After successfully fetching the results, the WellBe app will display the result.

5. Future Scope

1. More models such as kidney disease detection and breast cancer detection can be added to the app and app can be made as a one stop destination for the various types of disease detection.

2. A doctor consultation system can be added to the app where the user can consult any doctor for the reports generated in the app.

3. Various hospitals can use this app to generate reports at a faster level.

4. More analytics and a detailed report version can be created to give visual aids by graphs and charts.

5. Previous test results can be stored for a user and the improving or declining health can be tracked using graphs and suggestions can be given to the user for a better lifestyle.

6. Limitations

1. The overall speed of fetching data from server is a bit slow.
2. Some input fields are not understood by the user as they are medical terms.
3. The accuracy is nearly 80% so the test results can sometimes be wrong.