Machine Learning

The D's Application

By Aastha Parmar

Aastha Parmar
DHL
TYIT 2020

I would like to specially thank Dr. CH. Asrani for giving me an opportunity to build an application of this kind and providing me the useful data for building the application.


GROUP

Certificate of Appreciation

This certificate is awarded to

Aastha Parmar

for successful completion of D's Application

Dr. C.H. Asrani

Prof. Palash Ingle

InCHES Group - 5 Rajkamal, Opp Vidyanagari Kalina, Mumbai-400098 India | www.inchesgroup.in | info@inchesgroup.in

Table of Contents

Chapter 1:

Introduction

1.1 Background

My project is an estimation of medicine for Diabetes patients. In short it will be giving prescription for the diabetic patients. The idea of this project starts after dropping another idea; firstly I thought that Information Technology is a very good field that can be used for enhancing human health or helping the sick, hence I thought there has to be something that could be done to use technology with biological science and I came through an idea as we know Tuberculosis is a very Contagious and due to which many deaths occur in countries like India, Africa etc. Hence to help this out I came up with an idea to build a Software which provides prescription for this disease I came to know that when doctors give Antibiotics to the patients they never know if that medicine is going to work on that specific type of immunity and hence the patients conditions becomes worse. To overcome this situation, I planned to write an algorithm and build software which is going to show the doctor which medicine is going to work the best on the patients. So the main thing that I needed was data a proper number of accurate data. But in India doctors do not keep a record of the prescriptions and hence the doctor that I had a talk with suggested me to make something that gives suggestions on Diabetes or Hypertension, because he did not have any datasets he suggested me to go to a Tuberculosis firm where I could not go due to some reasons and hence I had to switch from Tuberculosis to Diabetes. So one of my doctors gave me a sufficient amount of dataset to work on and create a working model which is going to give estimation of the medication that we need to give to the patients.

1.2 Objectives

Objectives include a clear estimate of:

- Duration of treatment
- Diet needed
- Exercise
- Medicine to be given(Prescription)

To help doctors with good decision making.

1.3 Purpose, Scope, Applicability

Currently this project is only for diabetes but if we get proper data then we can make use of the current algorithm that we are using and then apply the same for the various diseases for better decisions making currently the main objective is to achieve prediction of diabetes medicine with different combinations for example: different age groups, some patients with thyroid, hypertension, pregnancy and for alcoholic people too.

Doctors get a lot of benefits they just have to enter the patient's information and then they get the results apart from that patients also get benefits as they do not have to go through the medicines that do not suit them or not heal them. It is of greater use to the local doctors who usually have a lot of Diabetic patients coming to them and also family doctors and many other hospitals and medical centers.

This product is applicable for Doctors who consult their patients everyday currently to train the model it is necessary that it is under the supervision of Professional doctors After the model becomes accurate normal patient can also use this application.

Chapter 2:

Survey of Technologies

Survey of Technologies

After surveying various applications it was found that many applications are just based on facts and figures , None of the applications provide prescription of Medicines of any type of disease hence it is a newly defined application and and it will be helpful in future cases.

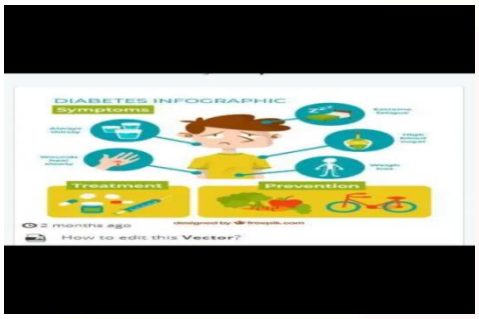
Here is a list of examples that were seen:

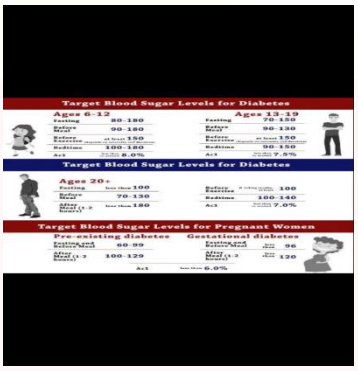
- **DIABETES TREATMENT App**

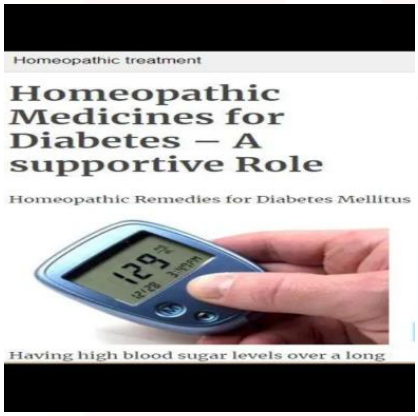
:

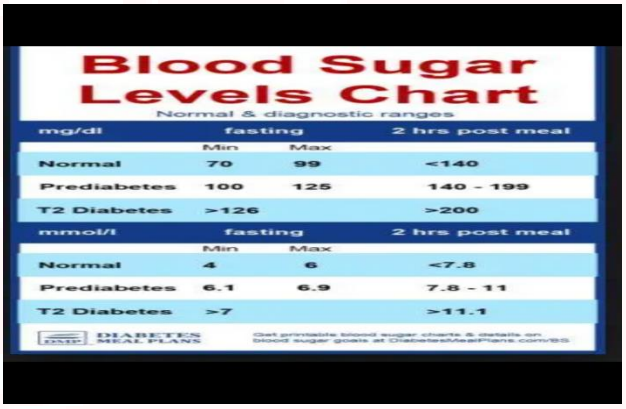
This application provides tips and the home made solutions for the treatment of the patient. It is a simple guide book app which updates periodically.

- **Diabetes treatment-latest research 100% effective**





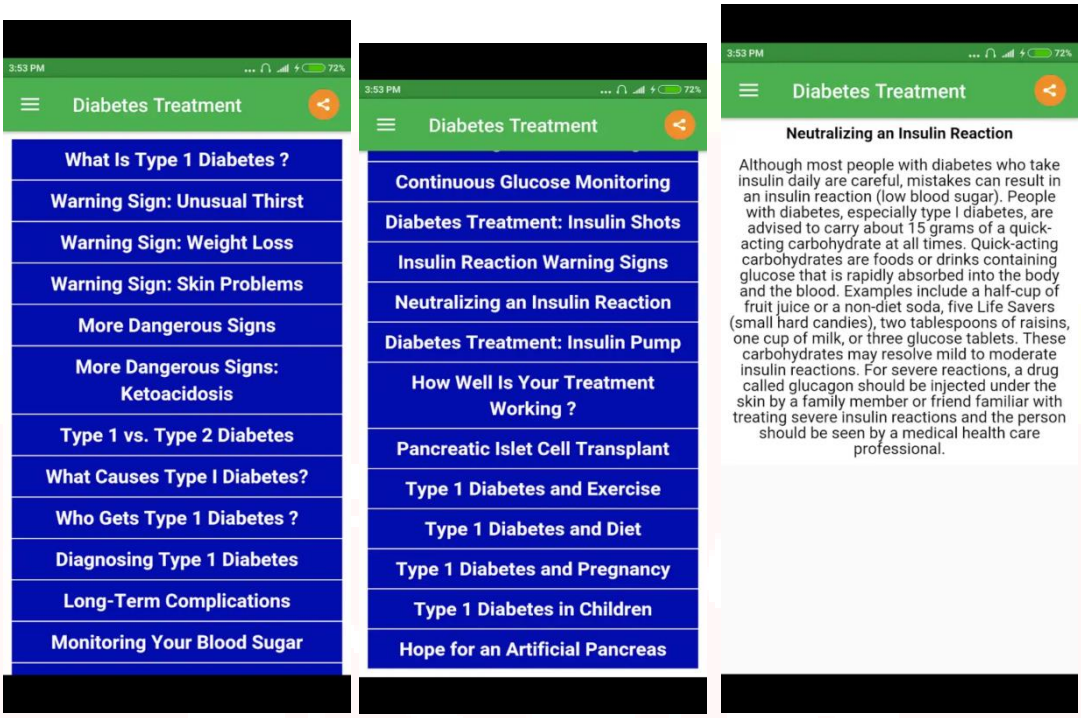




This application gives a complete guidance of what has to be provided to the predefined figures. It gives different strategies for optimal care, diabetes treatment options, and diabetes treatment in Hindi Language, diabetes treatment guidelines, things to treat diabetes naturally, diabetes management guideline.

<u>It is made using the latest guidelines for treatment of diabetes</u>

- **Diabetes Treatment**



This app is a collection of products and home remedies to cure and/or help prevent diabetes. All information **collected by people who have tried these methods for curing or diabetes treatment.**

As well as related social media content and other sources.

The drawbacks of these apps are:

- False treatments:
  Here the last application says that the remedies given are tried by people or taken through social media content which abruptly limits accuracy when it comes to medical guidance.
- The way of representation:
  The other application gives the remedies in the form of a book application users don't find it convenient to read because not all users are readers it is better if the users get the information with which they are convenient.

Chapter 3:

Requirements and Analysis

3.1 Problem Definition

The proposed projects consists of a machine learning algorithm which will process the input data; that is the medical data of the patients, that is the medical data of the person like the age, HbA1c level, physical examination, history etc., and then give out the right prescription according to the algorithm. The accuracy of the software targets to make the estimation more and more accurate.

3.2 Requirements Specification

Requirement specification for dataset:

- Age
- Physical examination
- HbA1c values
- Blood glucose:
  - Fasting Plasma Glucose
  - Post Prandial plasma glucose
- Duration
- Treatment
- Exercise
- Hypertension
- BMI
- Exercise
- Diet

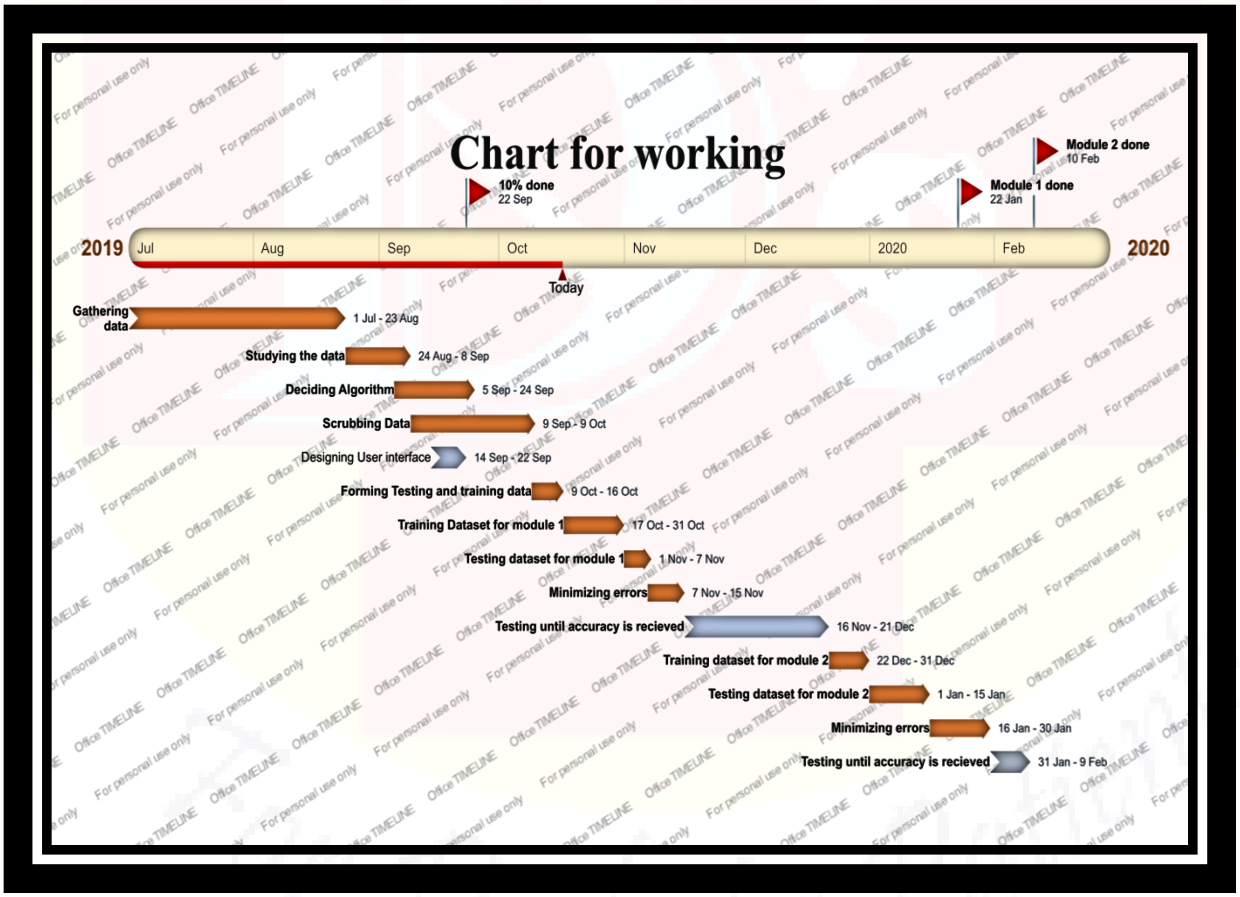
Information from Patient

- Age
- Physical examination
- HbA1c values

- ❖ Whether they are suffering from any other disease
  - Hypertension
  - Chronic Kidney disease
  - Dyslipidemia
  - Or other conditions like Pregnancy (in a diabetic)
- ❖ Blood glucose:
  - Fasting Plasma Glucose
  - Post Prandial plasma glucose

3.3 Planning and scheduling

This is the Gantt chart for the whole schedule for the project:



This is the chart with milestones and the whole schedule with dates.

3.4 Software and Hardware Requirements

- A machine
- Windows Operating system
- Dataset
- Python Interpreter
- Internet connection

Other requirements are specified in the requirements specification section.

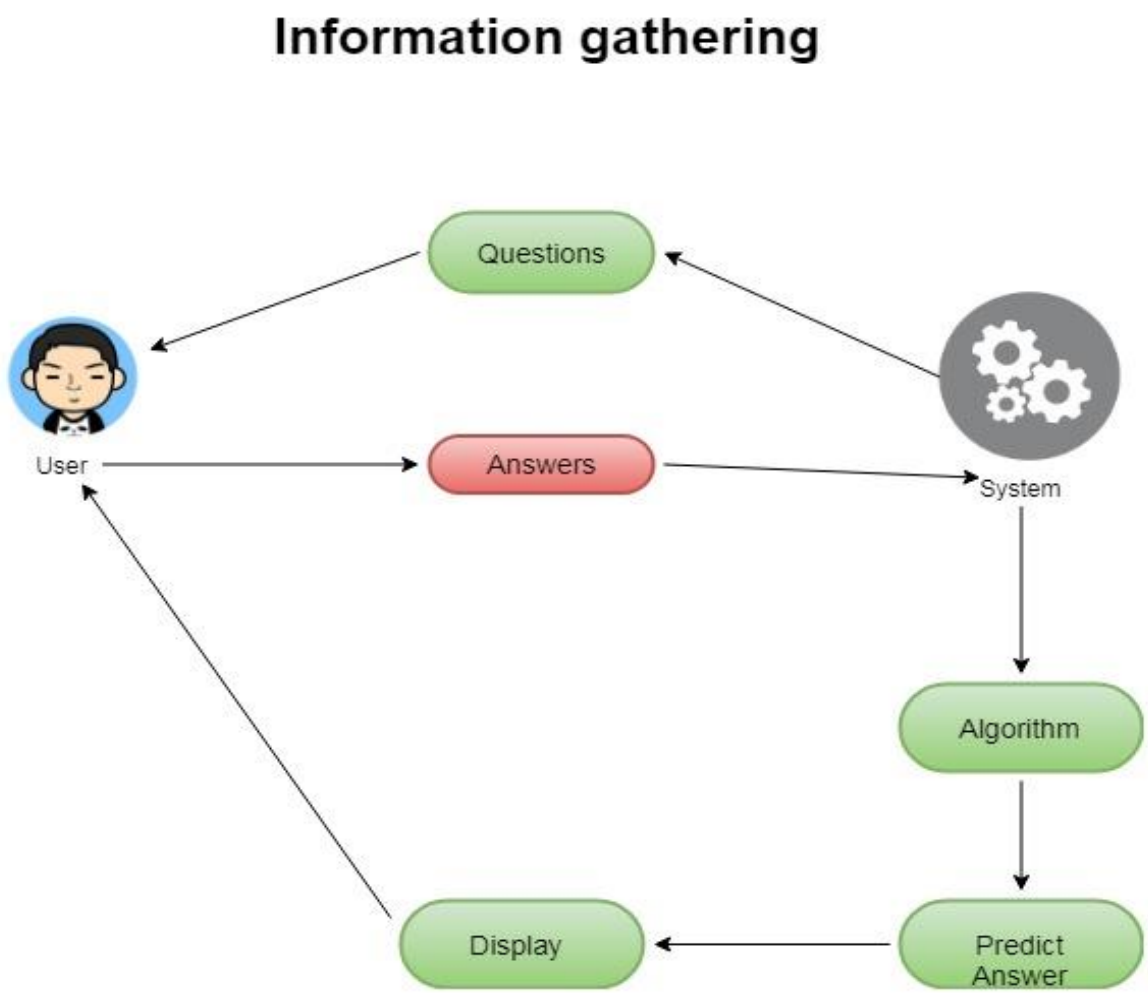
3.5 Preliminary Product Description

The objectives of the project is making an accurate application that provides medical advice as many people won't believe this feature to be true but computers do have the power to do impossible things with good speed. Hence by keeping this in mind I want to develop software of that type.

The requirements of the project are too many that is the data which is not easily available especially medical data which is very rare in a digital form. I had a talk with a couple of doctors who agreed to give me data in a digital format and hence it has become possible for me to do this project. Keeping every exception into consideration if my algorithm fails to teach itself well then it may lead to failure of the whole project but if the data is accurate and the algorithm is right then the project would be a great success for information technology in biology.

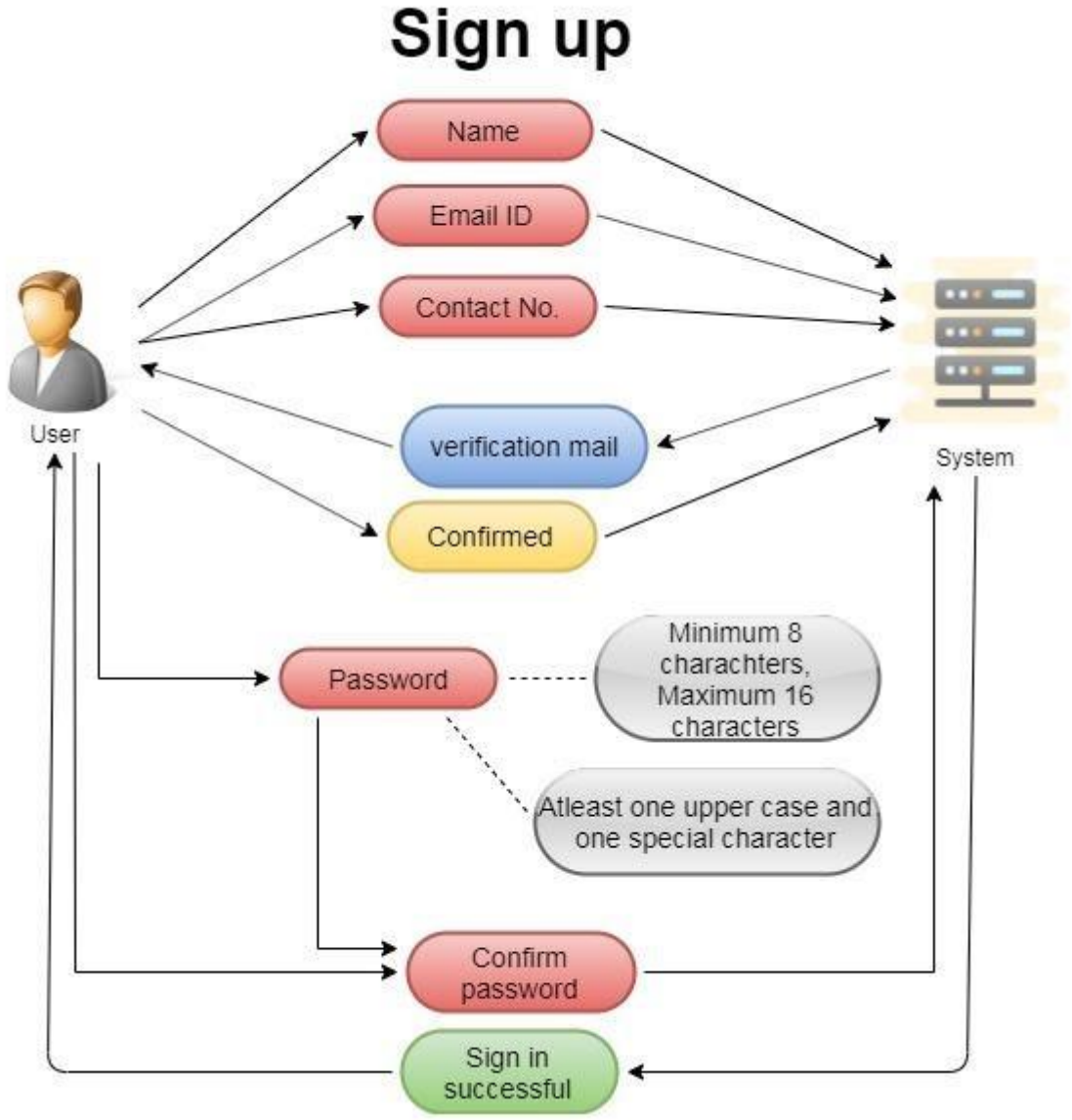
3.6 Conceptual Models

3.6.1 Use Case Diagrams



Here the steps are:

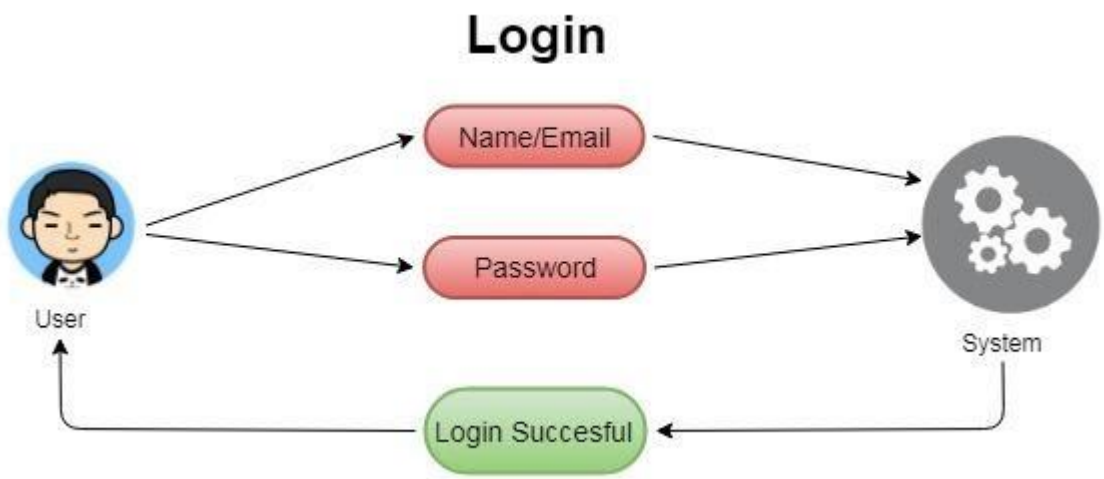
- Asking Questions
- Accepting Answers
- Then processing using the algorithm and predicting the answer
- Then displaying the Prediction



Here he procedure includes:

- Accepting Name
- Accepting Email
- Accepting Contact number(optional)
- Sending a verification mail
- Accepting a password with some validations
- Confirming the Password
- Successful Sign up

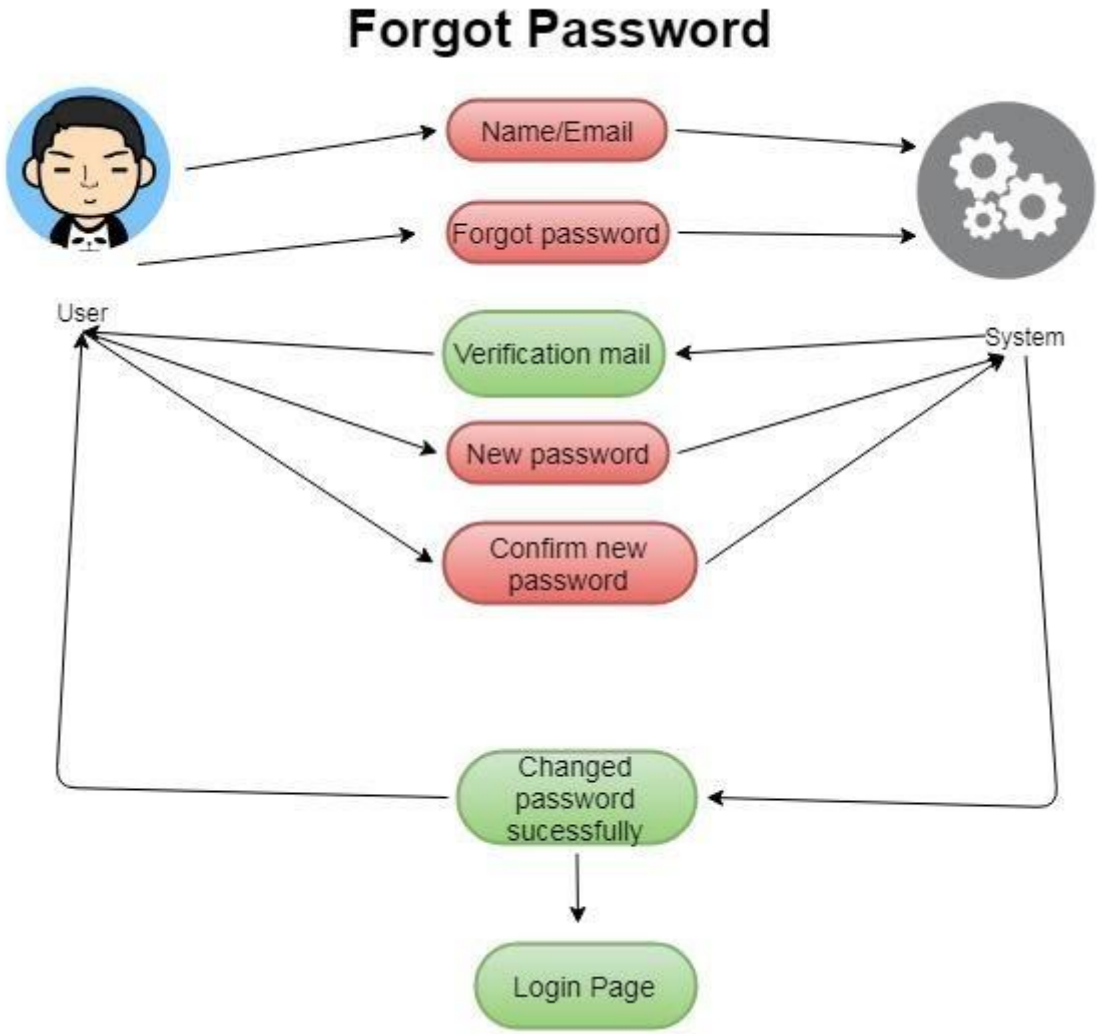
The next one is the Login Page:



Here the steps are as follows:

- Accept Name and Email
- Enter the Password
- If the Password is Correct
- Then Login Successful

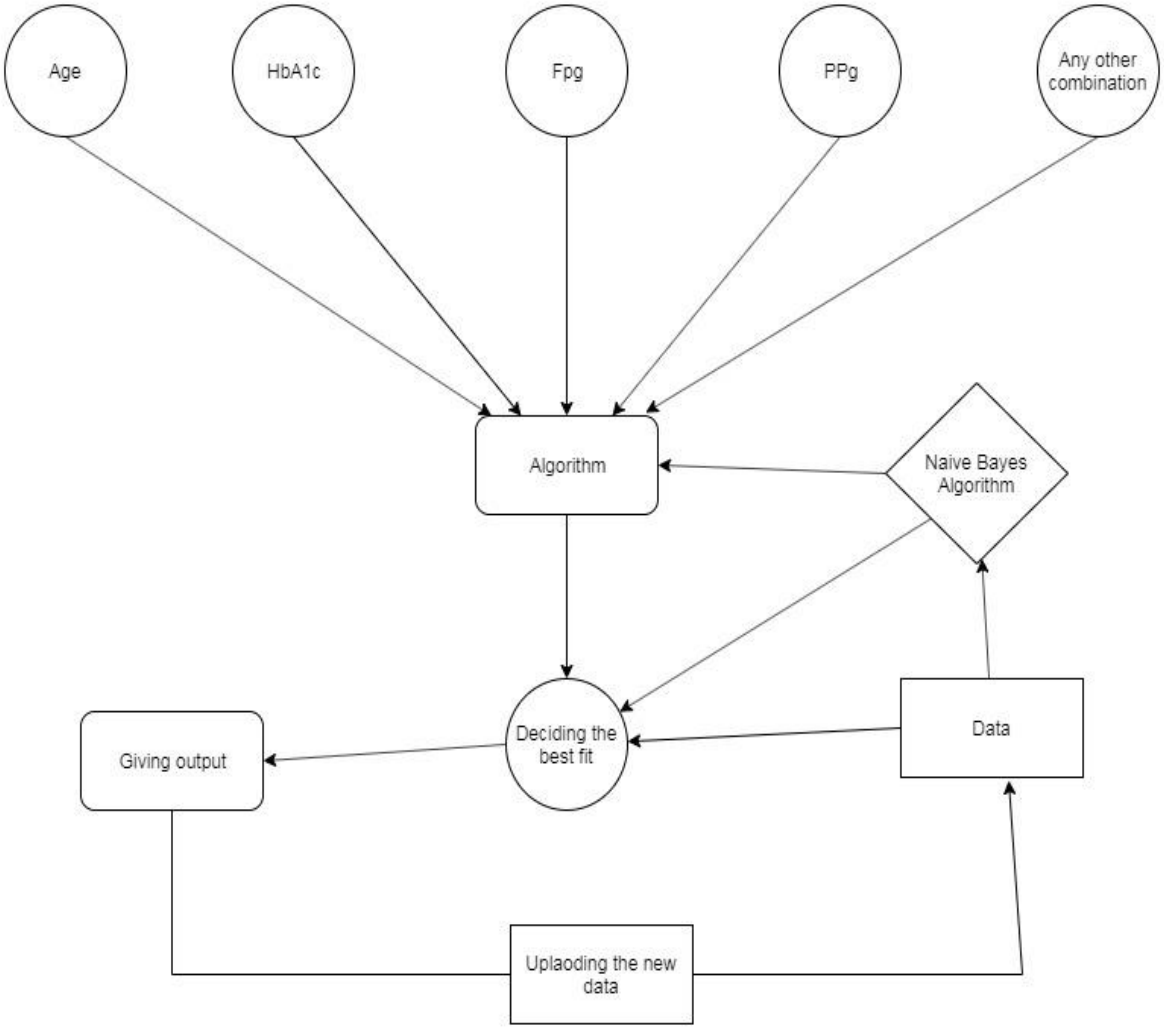
The next Page is the Forgot Password Page



Here forgot Password Page includes the following steps:

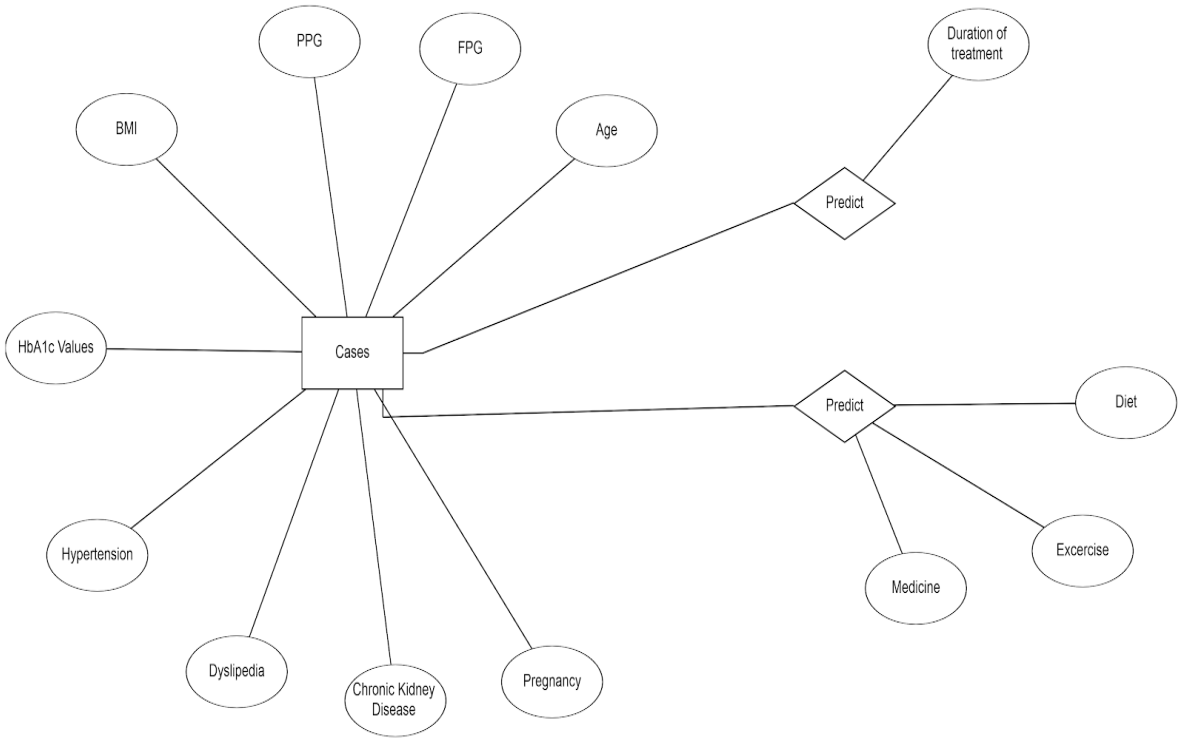
- If the user Clicks Forgot Password then send a verification email
- Accept the new Password
- Confirm the new Password
- Acknowledge that the Password was Changed
- Redirect to the Login Page

3.6.2 Data flow Diagram



The user data is taken and then processed using the algorithm and then the algorithm decides the best fit and then the output is given and then that data will get stored into the database for more predictions.

3.6.3 ER Diagram



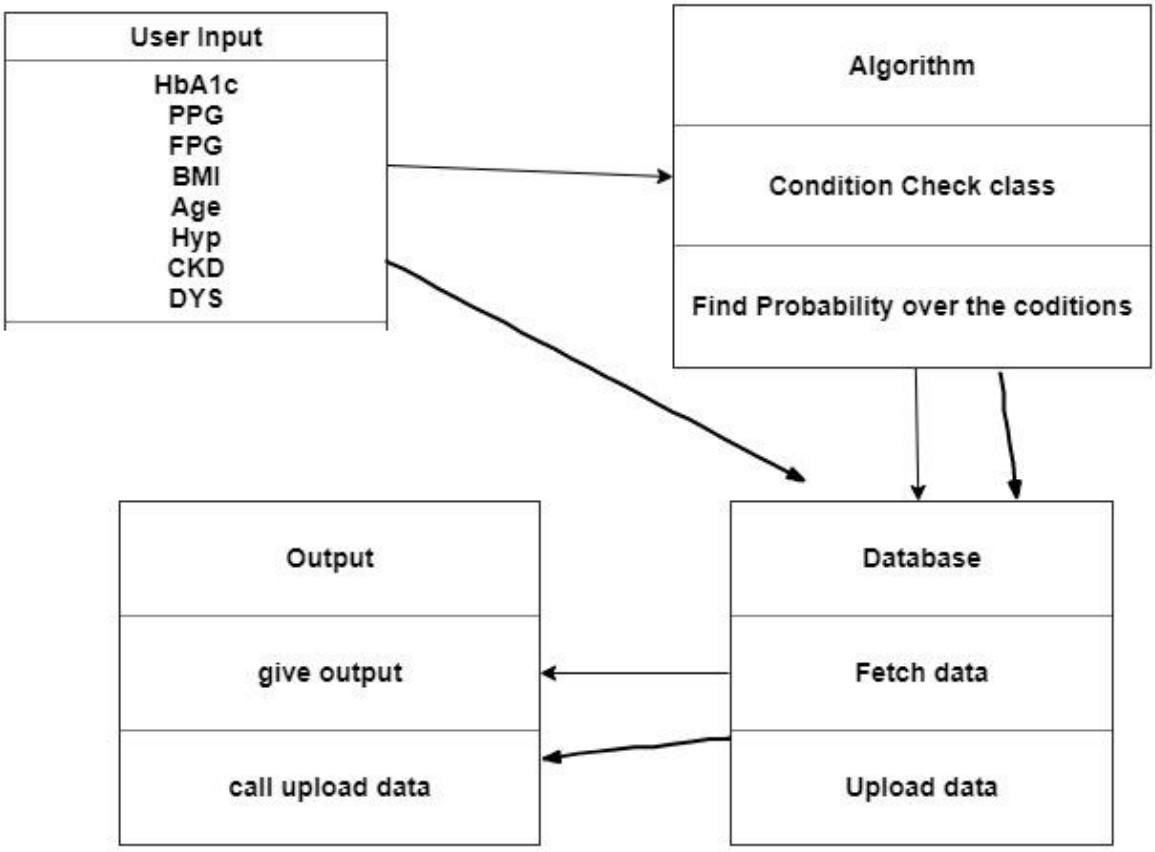
There are x labels that is the user inputs and the y labels that is the prediction the x labels are attributes here on the left side they are:

- Age
- FPG
- PPG
- BMI
- HbA1c values
- And the other combinations

The 'y' labels that are the output labels are the modules they are:

- Diet
- Exercise
- Medicine
- Duration of the treatment

3.6.4 Class Diagram



The user input class accepts the input the algorithm class processes it. It inherits from the data class fetches the data computes the probability and then give the output the output is then added to the database by calling the upload data function.

Chapter 4:

System Design

4.1 Significant Modules

Module 1 (Duration):

This module predicts the duration of the treatment so here it will accept the values of the data compare and then use the algorithm that is the Naïve Bayes and then predict the best probability and then give out the output.

Basically the Naïve Bayes Algorithm produces the probability of one attribute over the probability of the other it is usually helpful if we want to predict the probability of one attribute which is dependent on the other and we have the data for the other attribute.

Module 2(Diet), (Exercise), (Medicine):

Here the whole thing remains the same but as we are using words instead of numbers text classification will be used. Here we will need to train the data for text and predict the estimation.

Module 3(User interface and the authentication:

Designing the user interface such that it is compatible for all desktop devices and building Firebase authentication for user login and other procedures.

4.2 Data design

4.2.1 Schema design

The data is usually divided in user login data and the data which was used for training the model.

User data:

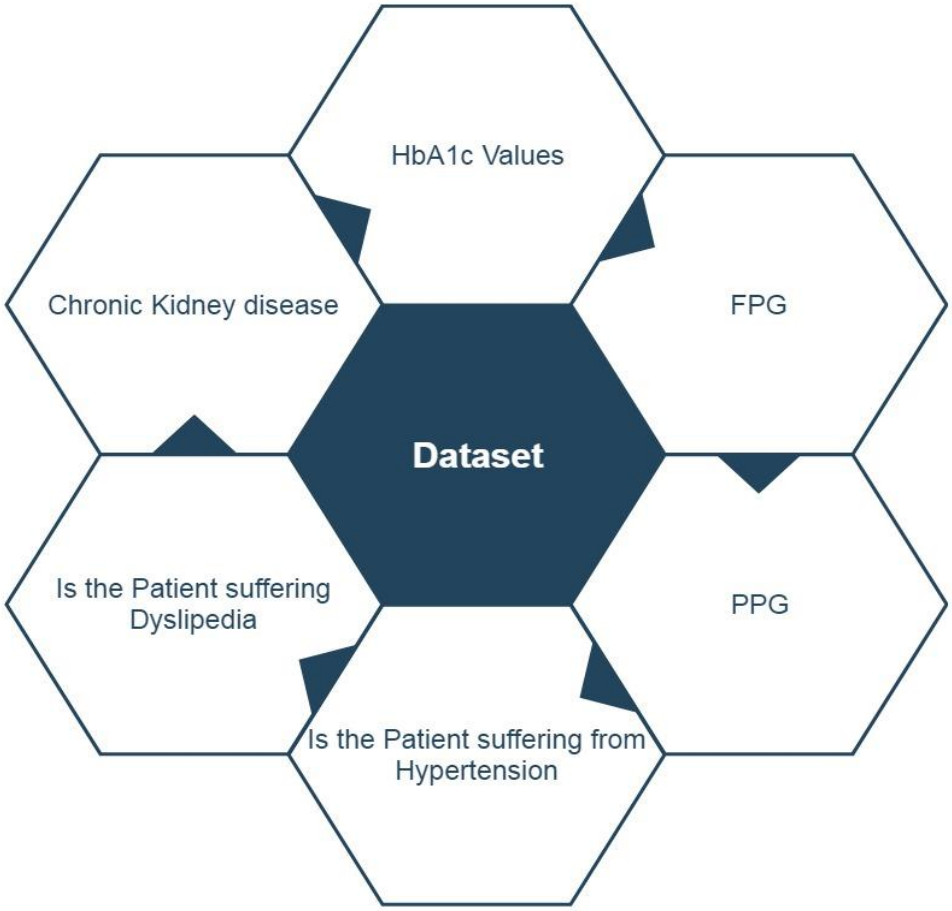
The database which I have used is Google Firebase which is quiet secure.

The database consists of two variables one which takes the email and the other password.

So the schema would look like:

Email	Password
• abc@gmail.com • xyz@gmail.com	• ..... • .....

Training data:



These are the different variables in my dataset they are stored in the following format.

Patient Information	Prediction
BMI	Duration of treatment
FPG	Treatment
PPG	Diet
HbA1c	Exercise
Any other cormobidities?	

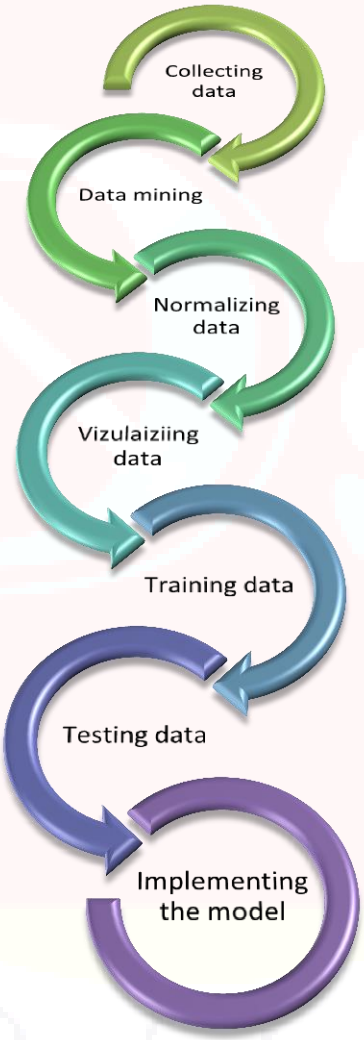
4.2.2 Data Integrity constraints

This chart will show you what constraints I have applied on my data

Name	•Only text allowed
Email	•regular expression used for the standard format of emails
Password	•combination of characters and words •Text not less than 4 words •Password hashed unseen when typed
Age	•Stictly number •Number not greater than 200
Weight	•Stictly number
Hba1c	•stictly number •size constaints 1 to 50
FPG	•strictly number •size constraints between 1 and 500
PPG	•stictly number •size constaints 1 to 500

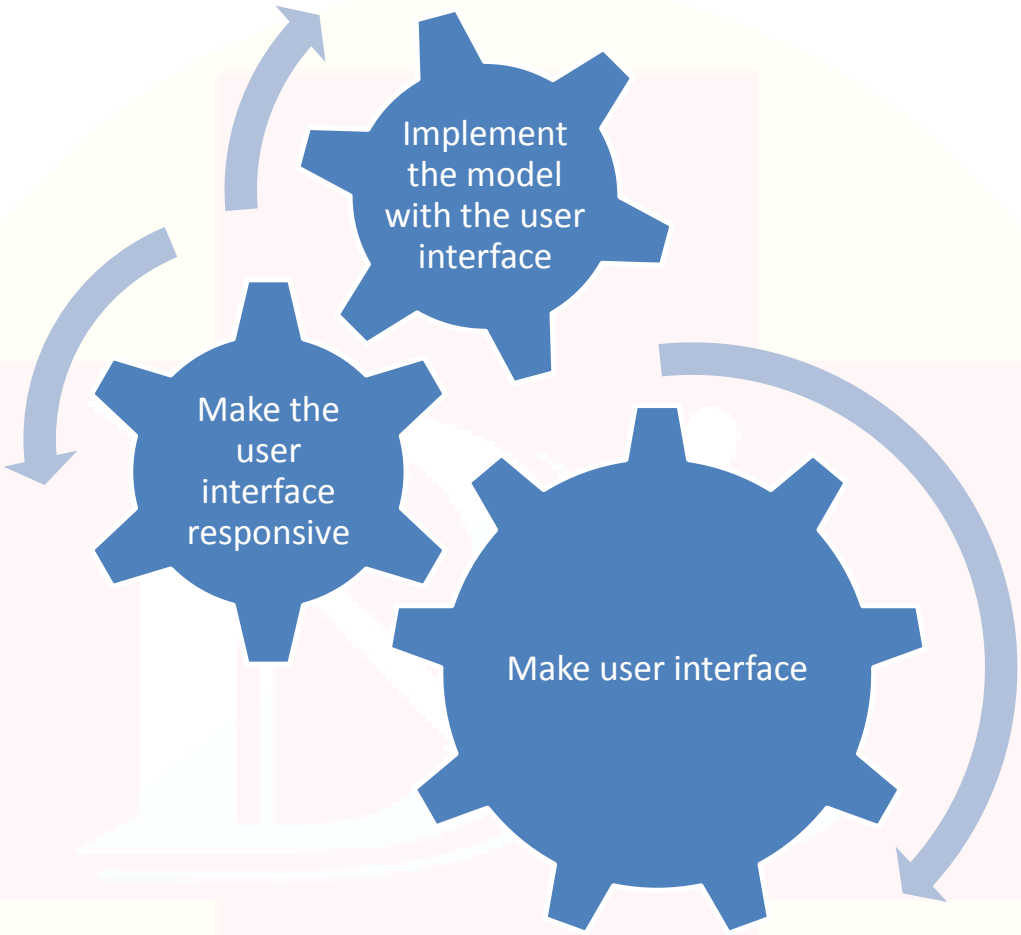
4.3 Procedural Design

4.3.1 Logic Diagram for building model



Training the model is followed by the steps above.

4.3.2 Algorithm Design



As the diagram here says I have implemented the project in the same way and further I have made the User interface.

4.3.2 Data Structures

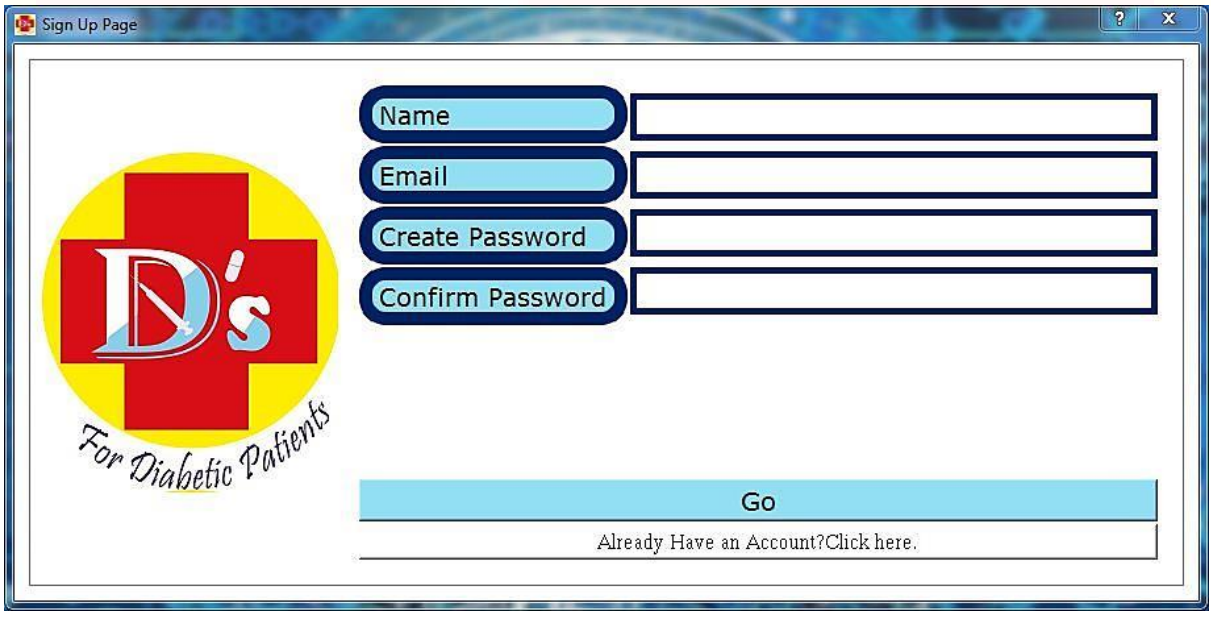
The data here given is raw data and hence data scrubbing is needed the data after scrubbing will look like:

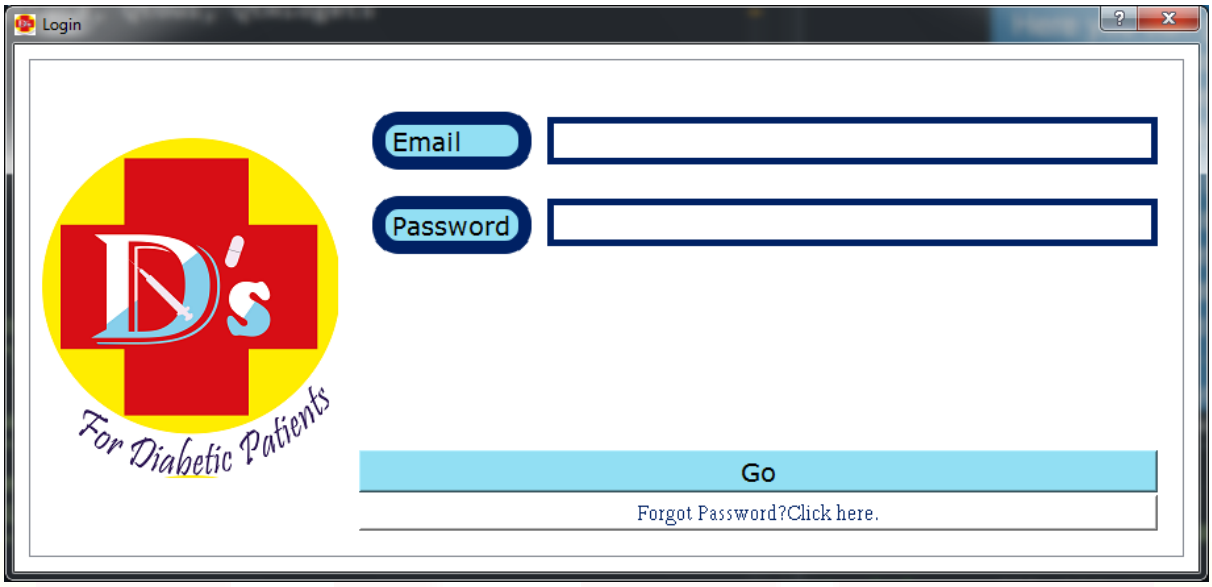
Name	Type
HbA1c	Float64
FPG	Float64
PPG	Float64
BMI	Float64
Age	Float64
HTN	Float64
DYS	Float64
CKD	Float64
Preg	Float64
Treatment	Object

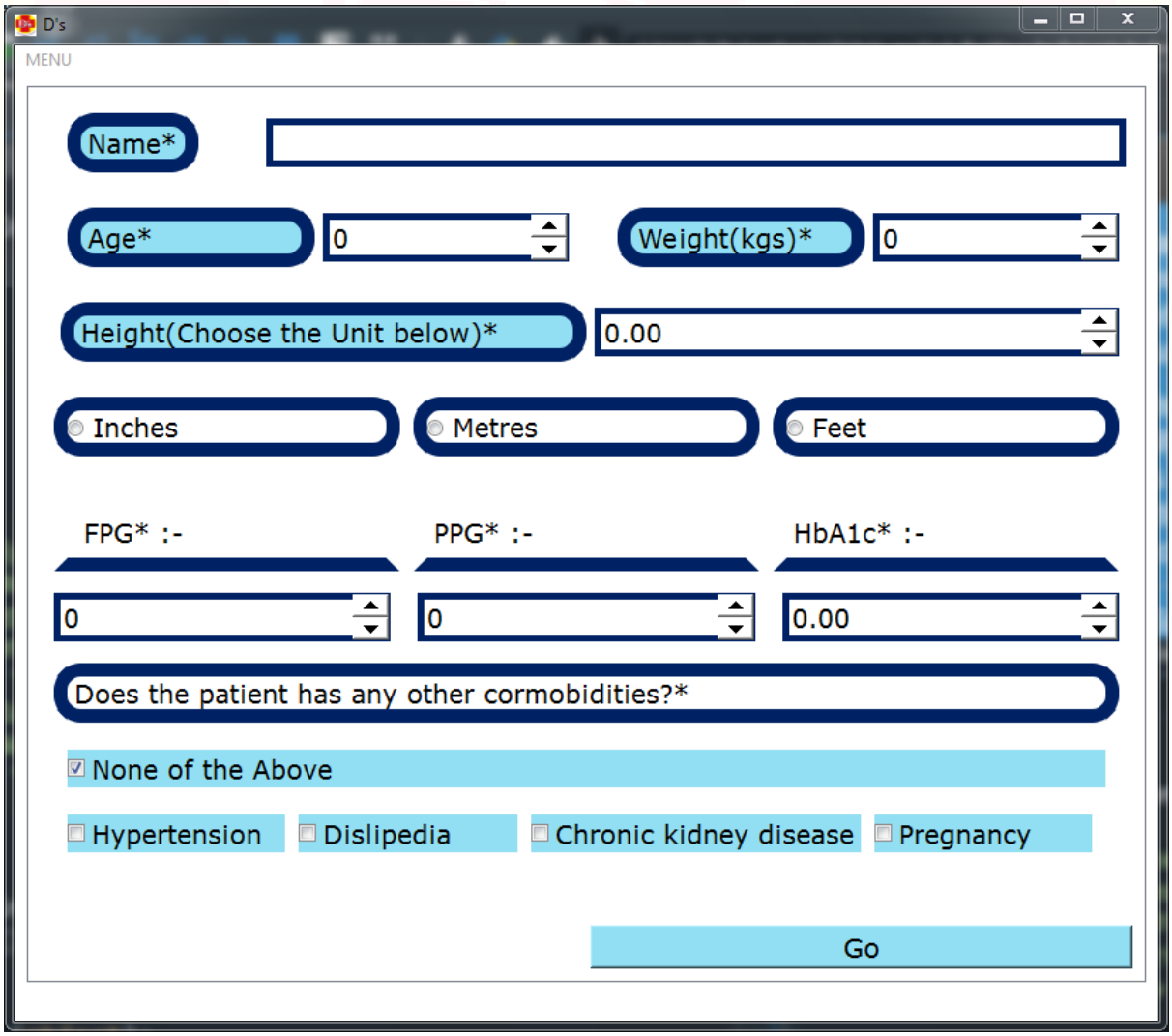
The data types here are not defined as primary key because they are not unique everywhere but I have made sure that everything in not null.

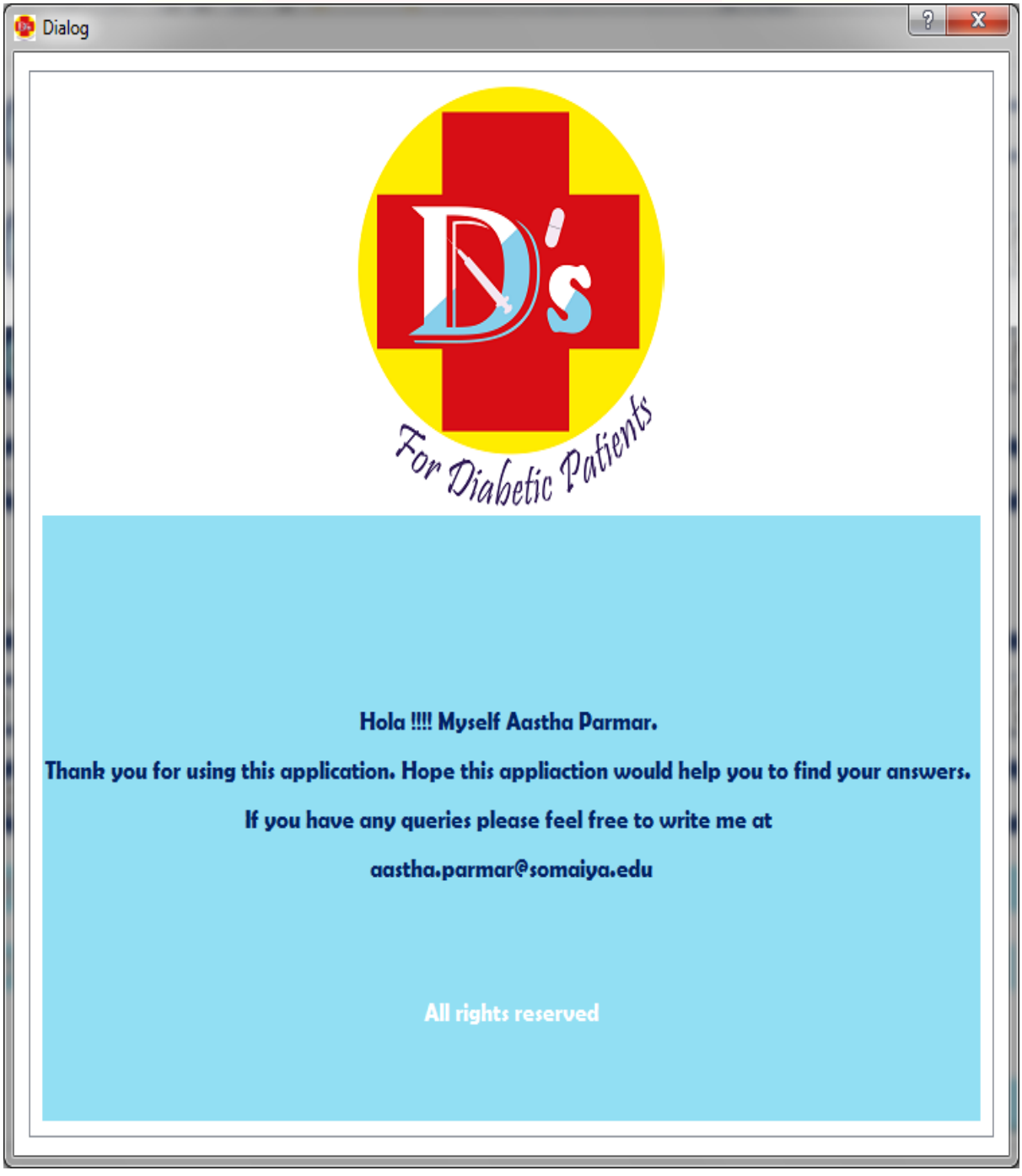
4.4 User interface design

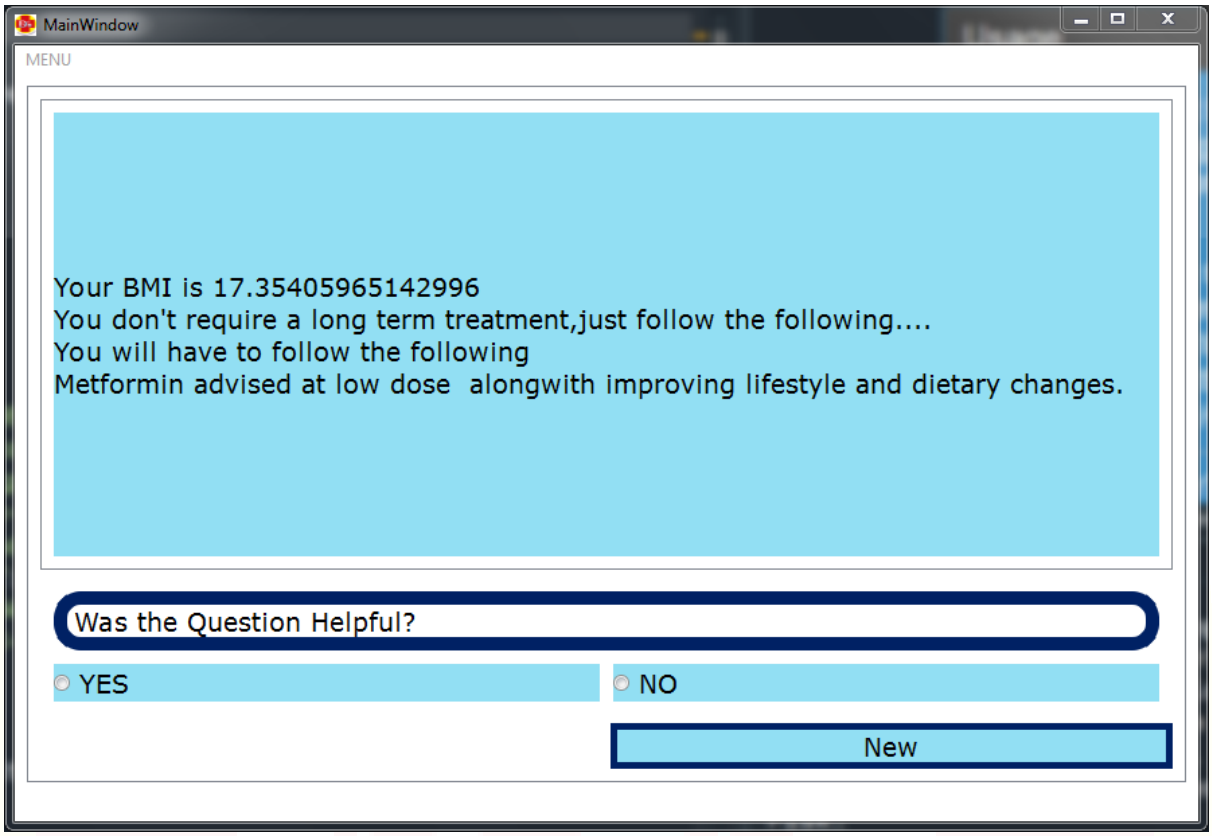
Designing the logo was the most creative part while making the UI, making the logo compatible with the type of application and the other things were considered, the logo was designed using Adobe Illustrator.




Login
Email
Password
D's
For Diabetic Patients
Go
Forgot Password?Click here.


D's
MENU
Name*
Age*
0
Weight(kgs)*
0
Height(Choose the Unit below)*
0.00
Inches
Metres
Feet
FPG* :-
PPG* :-
HbA1c* :-
0
0
0.00
Does the patient has any other cormobidities?*
None of the Above
Hypertension
Dislipedia
Chronic kidney disease
Pregnancy
Go


Dialog
D's
For Diabetic Patients
Hola !!!! Myself Aastha Parmar.
Thank you for using this application. Hope this appliaction would help you to find your answers.
If you have any queries please feel free to write me at
aastha.parmar@somaiya.edu
All rights reserved



The entire user interface were made using python which is compatible for Linux, Mac and Windows, they can be used in any platform, just build the installers for the respective Operating system and use it.

4.5 Security Issues

Security becomes the main issue in today's days, so it is very necessary to protect the user data and user information so I used Google Firebase for storing the database and Email password authentication, if any issue occurs feedback can be given by calling or email which is mentioned in the **"About page"**, admin can delete of modify user data.

4.6 Test case design

Test case design for algorithm:

Applied different algorithms and then measured train and testing accuracy, when the accuracy becomes acceptable and when bias and variance is not found then the algorithm gets accepted.

Apart from that the application stays under medical supervision so that if anything goes wrong the supervisor will identify it.

Test cases for other features:

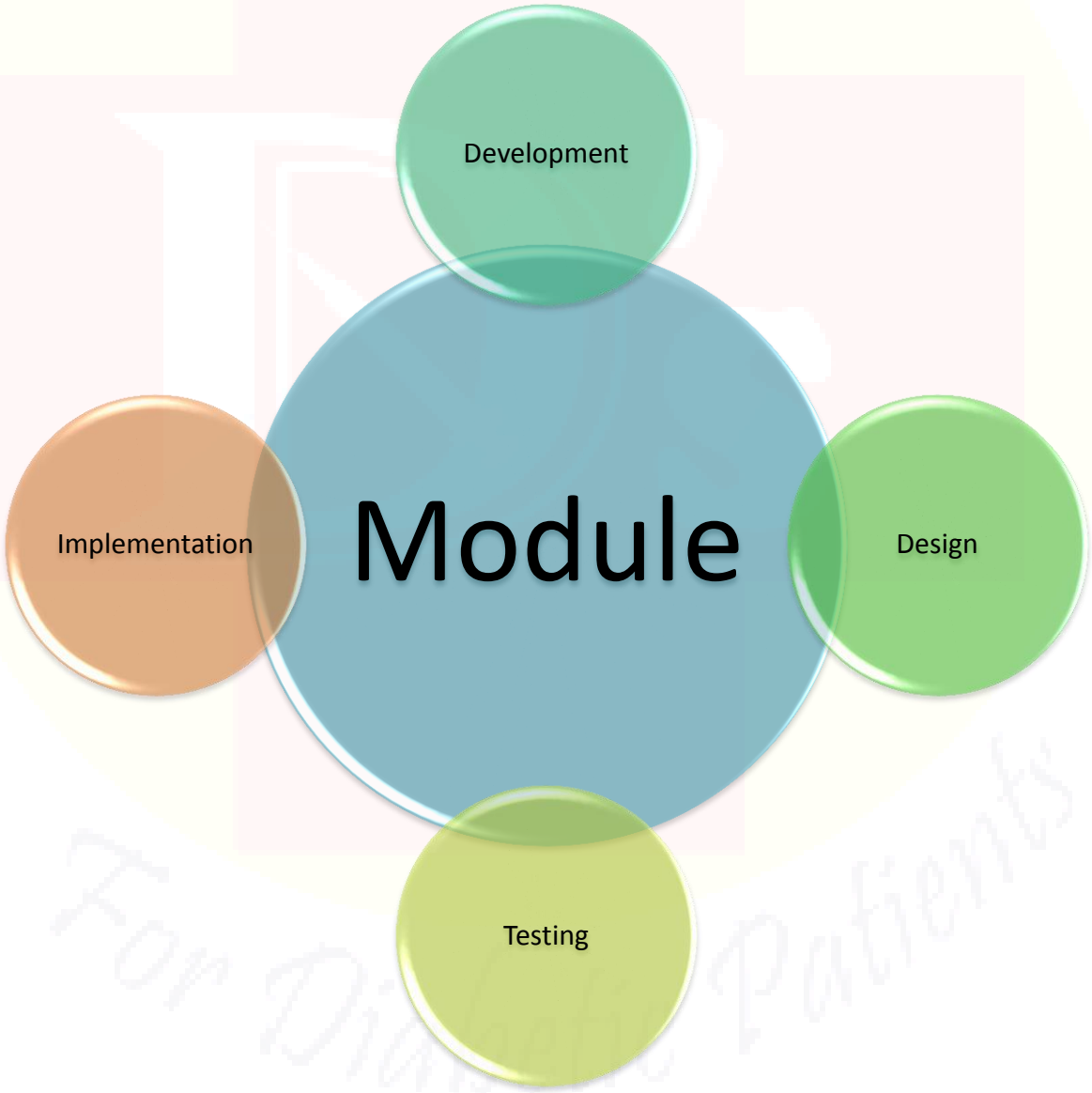
- User interface should be responsive
- Warning labels should be shown correctly
- Navigation through the pages should be correct
- Font should be visible
- Color should not be dull or too much attractive

Chapter 5:

Implementation and testing

5.1 Implementation approaches

The Software development life cycle model used for the project is incremental model which is the best suited for this type of project.



5.2 Code Efficiency:

The Algorithm:

The algorithm that we are using here is Decision Tree Algorithm in Machine Learning the main part to implement the algorithm was to save the algorithm. “Pickle” library was used to save the model. The following snippet implements the same.

```
import pickle    #importing the library
filename = 'Duration_model.sav'    #declaring the filename
pickle.dump(classifier, open(filename, 'wb'))  #saving the model in the file
```

The model was further loaded as:

```
import pickle   #importing the library
model = pickle.load(open('Duration_model.sav','rb'))  #loading the    saved  file
```

The prediction was then made
For the User interface PyQt5 was implemented using layouts,
They were implemented as:

```
self.gridLayout = QtWidgets.QGridLayout(self.centralwidget)  #declaring the
grid layout
self.gridLayout.setObjectName("gridLayout")   #set object name of the layout
```

whereas the widgets like button, lineEdit,label were created as:

```
self.label = QtWidgets.QLabel(self.scrollAreaWidgetContents) #declaring the
variable
self.label.setText("Hello")  #setting texxt of the variable
```

```
self.label.setObjectName("label")  #setting object name of the
variable

self.horizontalLayout_2.addWidget(self.label)  #Adding the label in the layout
self.New_button = QtWidgets.QPushButton(self.scrollAreaWidgetContents)
#declaring the push button
 self.New_button.setStyleSheet("#New_button\n"
"{background-color: rgb(146,223,243);\n"
"font: 14pt \"MS Reference Sans Serif\";\n"
"border-style:solid;\n"
"border-width:5px;\n"
"border-radius:20px;\n"
"border-color:rgb(0, 33,100);\n"
"}\n"
"#New_button:hover{\n"
"background-color: rgb(0, 170, 170);\n"
"}")   # stying the button using css
 self.New_button.setObjectName("New_button")  #setting the object name of the
button
 self.horizontalLayout_2.addWidget(self.New_button)   #Adding the button
using object name in the layout.
```

5.3 Testing Approach:

As Incremental model was used the testing was performed after every module was created. Further information is given in Unit testing.

5.3.1 Unit testing:

Development of the application is basically divided in two modules. Training the model and the user interface with authentication and then connecting them.

Module 1:

❖ Cleaning data testing:

Usually data needs a lot of cleaning and mining when we get it so usually testing whether data is in a good format we need to check it either visually or by some code, but checking a lot of data visually becomes a burden hence we do use the following code.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11999 entries, 0 to 11998
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Age                    11999 non-null  float64
 1   BMI                    11999 non-null  int64
 2   Hba1c                  11999 non-null  float64
 3   FPG                    11999 non-null  int64
 4   PPG                    11999 non-null  int64
 5   HTN                    11999 non-null  int64
 6   DYLP                   11999 non-null  int64
 7   CKD                    11999 non-null  int64
 8   Pregn                  11999 non-null  int64
 9   Duration lasting till  11999 non-null  int64
 10  Diabetes treatment     11999 non-null  object
dtypes: float64(2), int64(8), object(1)
memory usage: 984.4+ KB
```

And then we find the minimum and the maximum of each and every column by using the following code

```
data.describe()
```

	Age	BMI	Hba1c	FPG	PPG	HTN	DYLP	CKD	Pregn	Duration lasting till
count	11999.000000	11999.000000	11999.000000	11999.000000	11999.000000	11999.000000	11999.000000	11999.000000	11999.000000	11999.000000
mean	40.128011	2.459455	10.370431	223.891491	299.885990	0.444454	0.432453	0.128011	0.032003	1.841237
std	22.892397	1.151328	2.929657	94.430048	122.917073	0.496926	0.495437	0.334116	0.176014	0.951243
min	12.500000	0.000000	5.400000	95.000000	135.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	12.500000	1.000000	7.900000	135.000000	192.000000	0.000000	0.000000	0.000000	0.000000	1.000000
50%	30.500000	2.000000	10.500000	245.000000	300.000000	0.000000	0.000000	0.000000	0.000000	2.000000
75%	43.000000	3.000000	13.100000	300.000000	425.000000	1.000000	1.000000	0.000000	0.000000	3.000000
max	75.500000	4.000000	14.600000	385.000000	475.000000	1.000000	1.000000	1.000000	1.000000	3.000000

So now when we see the mean of all the columns there's a lot of difference between then and eventually they consume memory and also computation time when we use the algorithm. So basically data normalization is used. The formula for it is:

$$x' = \frac{x - average(x)}{max(x) - min(x)}$$

- Build the model and perform cross validation
- Build plots for visualization

Module2:

- After building the firebase Application check for the other modules like
  1. Sign up authentication
  2. Forgot Password Authentication
  3. Login Authentication

- For user interface check for data integrity and size constraints if there is any error or exception what message needs to be displayed, message should be visible to the user.
- Check if button click works.

5.3.2 Integrated testing:

❖ After implementing all the modules together testing should be done for

i. Login Page

a. Logo displayed correctly?
b. Authentication working?
c. Data constraints working?
d. Buttons Working?
e. Forgot Password working?
f. After Successful navigation is home Page displayed?

ii. Sign in Page

a. Logo Displayed Correctly?
b. Authentication working?
c. Data Constraints working?
d. Login Page navigation working?

iii. Home Page

a. Data constraints working?
b. About page working?
c. Button working?

iv. Model

a. Results displayed?
b. Any errors while computing results?

5.4 Modifications and Improvements:

The logo made before seemed to be dull or unattractive hence changes were made, in the logo and some errors related to validations were debugged.

Chapter 6:

Results and discussion

Test reports:

- After normalizing the data used lesser memory

```
dat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11999 entries, 0 to 11998
Data columns (total 9 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Age     11999 non-null  float64
 1   BMI     11999 non-null  float64
 2   Hba1c   11999 non-null  float64
 3   FPG     11999 non-null  float64
 4   PPG     11999 non-null  float64
 5   HTN     11999 non-null  int64  
 6   DYLP    11999 non-null  int64  
 7   CKD     11999 non-null  int64  
 8   Pregn   11999 non-null  int64  
dtypes: float64(5), int64(4)
memory usage: 843.7 KB
```

In [33]:
```
dat.describe()
```

Out[33]:

	Age	BMI	Hba1c	FPG	PPG	HTN	DYLP	CKD	Pregn
count	1.199900e+04	1.199900e+04	1.199900e+04	1.199900e+04	1.199900e+04	11999.000000	11999.000000	11999.000000	11999.000000
mean	2.746180e-17	-2.427890e-17	5.655207e-17	-2.783191e-17	-3.404968e-17	0.444454	0.432453	0.128011	0.032003
std	3.633714e-01	2.878320e-01	3.184409e-01	2.487620e-01	2.617485e-01	0.496926	0.495437	0.334116	0.176014
min	-4.385399e-01	-6.148637e-01	-5.402642e-01	-3.395456e-01	-3.511201e-01	0.000000	0.000000	0.000000	0.000000
25%	-4.385399e-01	-3.648637e-01	-2.685251e-01	-2.341715e-01	-2.297402e-01	0.000000	0.000000	0.000000	0.000000
50%	-1.528256e-01	-1.148637e-01	1.408360e-02	5.560724e-02	2.427800e-04	0.000000	0.000000	0.000000	0.000000
75%	4.558713e-02	1.351363e-01	2.966923e-01	2.004966e-01	2.664268e-01	1.000000	1.000000	0.000000	0.000000
max	5.614601e-01	3.851363e-01	4.597358e-01	4.244165e-01	3.729004e-01	1.000000	1.000000	1.000000	1.000000

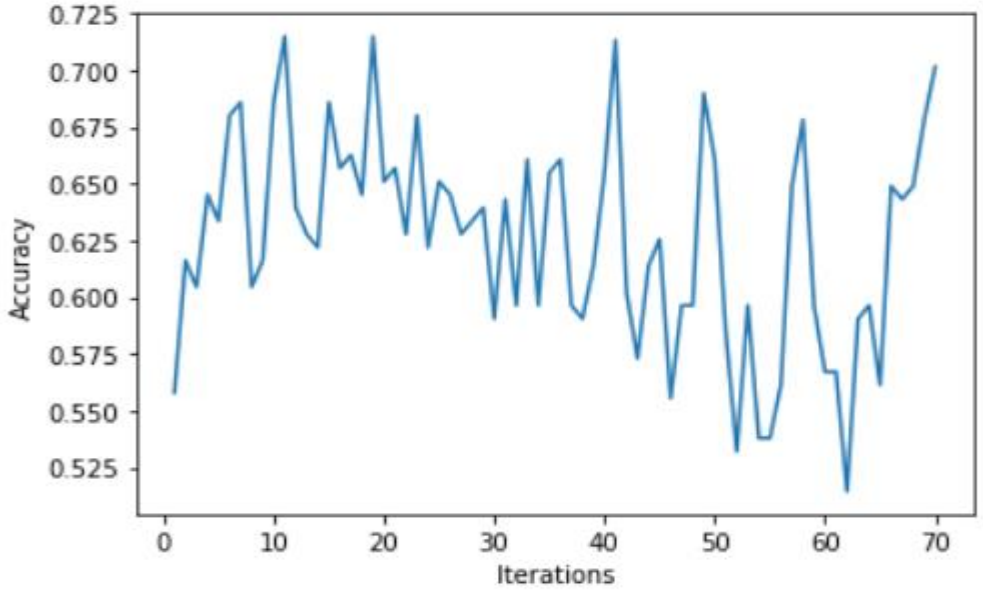
The model testing was used with various graphs and functions

In [38]:
```
cross_val_score(lr, X, y, cv=70)
```

Out[38]:
```
array([0.55813953, 0.61627907, 0.60465116, 0.64534884, 0.63372093,
       0.68023256, 0.68604651, 0.60465116, 0.61627907, 0.68604651,
       0.71511628, 0.63953488, 0.62790698, 0.62209302, 0.68604651,
       0.65697674, 0.6627907 , 0.64534884, 0.71511628, 0.65116279,
       0.65697674, 0.62790698, 0.68023256, 0.62209302, 0.65116279,
       0.64534884, 0.62790698, 0.63372093, 0.63953488, 0.59064327,
       0.64327485, 0.59649123, 0.66081871, 0.59649123, 0.65497076,
       0.66081871, 0.59649123, 0.59064327, 0.61403509, 0.65497076,
       0.71345029, 0.60233918, 0.57309942, 0.61403509, 0.62573099,
       0.55555556, 0.59649123, 0.59649123, 0.69005848, 0.66081871,
       0.58479532, 0.53216374, 0.59649123, 0.5380117 , 0.5380117 ,
       0.56140351, 0.64912281, 0.67836257, 0.59649123, 0.56725146,
       0.56725146, 0.51461988, 0.59064327, 0.59649123, 0.56140351,
       0.64912281, 0.64327485, 0.64912281, 0.67836257, 0.70175439])
```

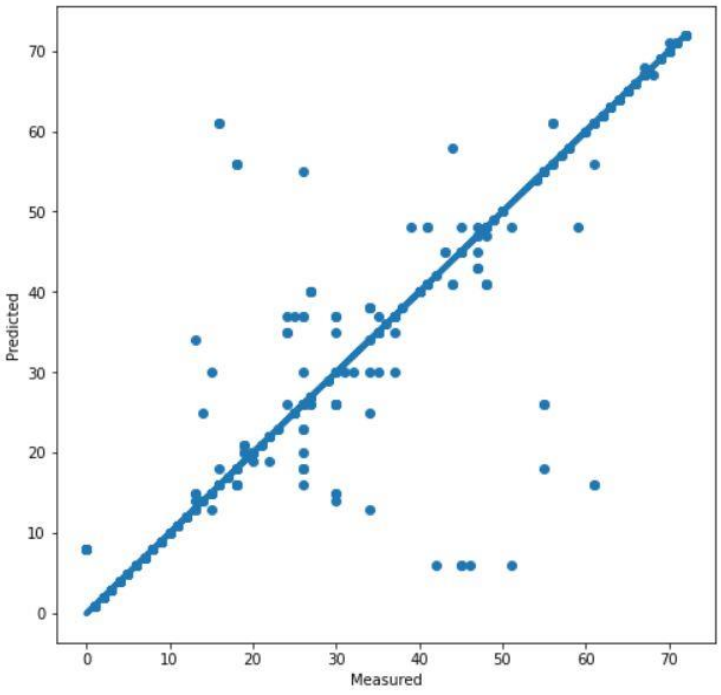
- It is very important that we use cross validation technique when we have less data. Cross validation technique with 70 iterations give the best training score.

❖ The Plot between Accuracy and iterations is very noisy but eventually it heads to the best result.

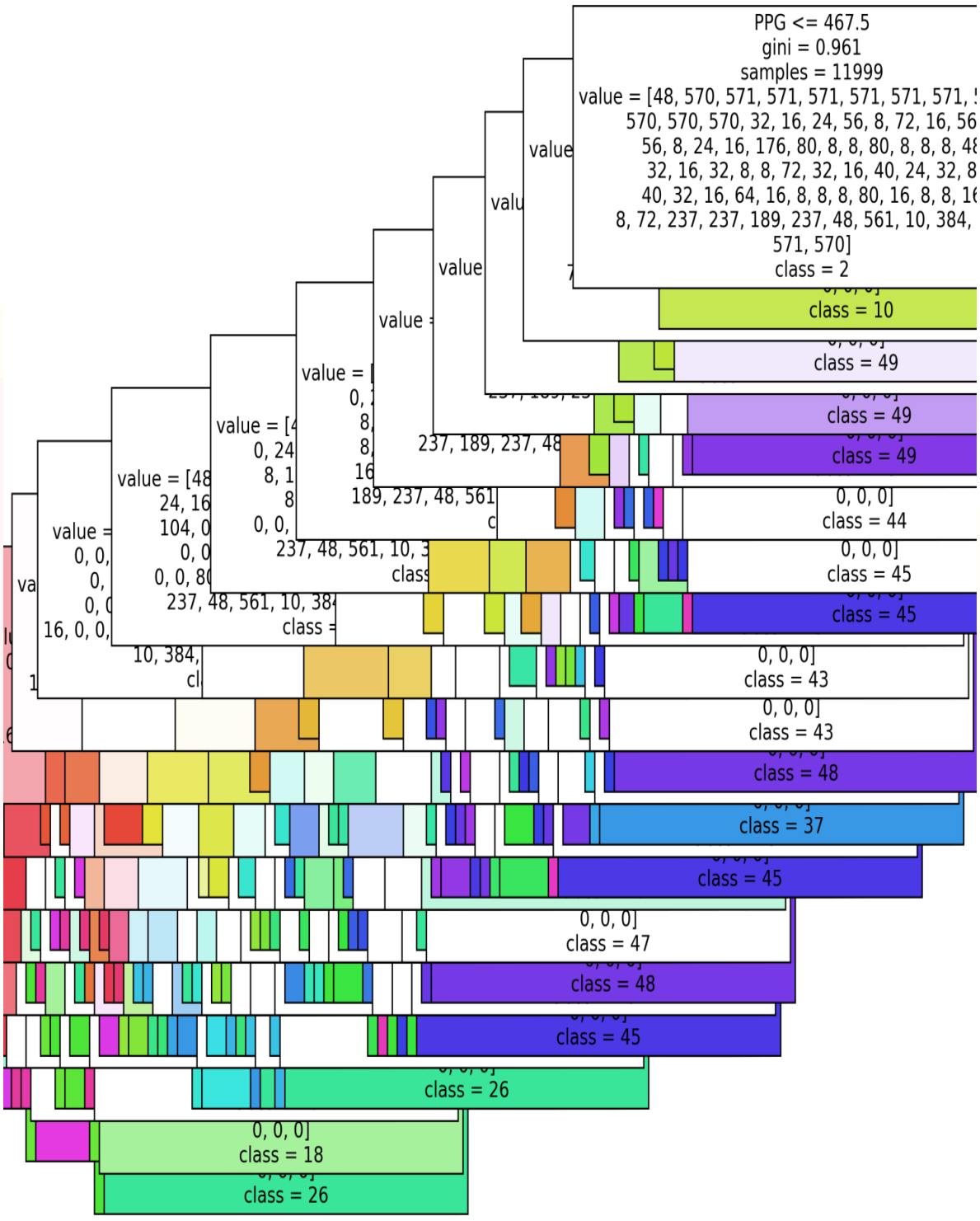


Similarly the treatment model gives good results the graph is plotted below

```
% (min_groups, self.n_splits)), UserWarning)
```



❖ As we can see this is looks like the best possible model for the data



❖ This is the plot of the tree that is how the tree is able to come to a specific decision. Here the model gives 99.5% accuracy.

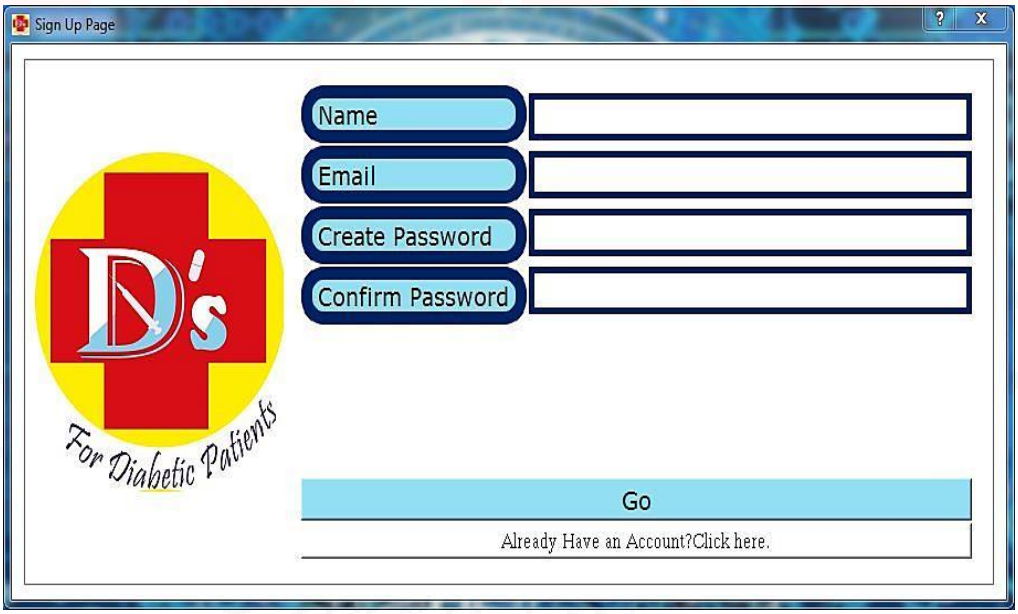
Module 2:

- All the functions and constraints work well and fine
- The button navigation works fine
- All the widgets are displayed correctly and messages are displayed correctly
- The output is also displayed in a promising manner

6.2 User Documentation

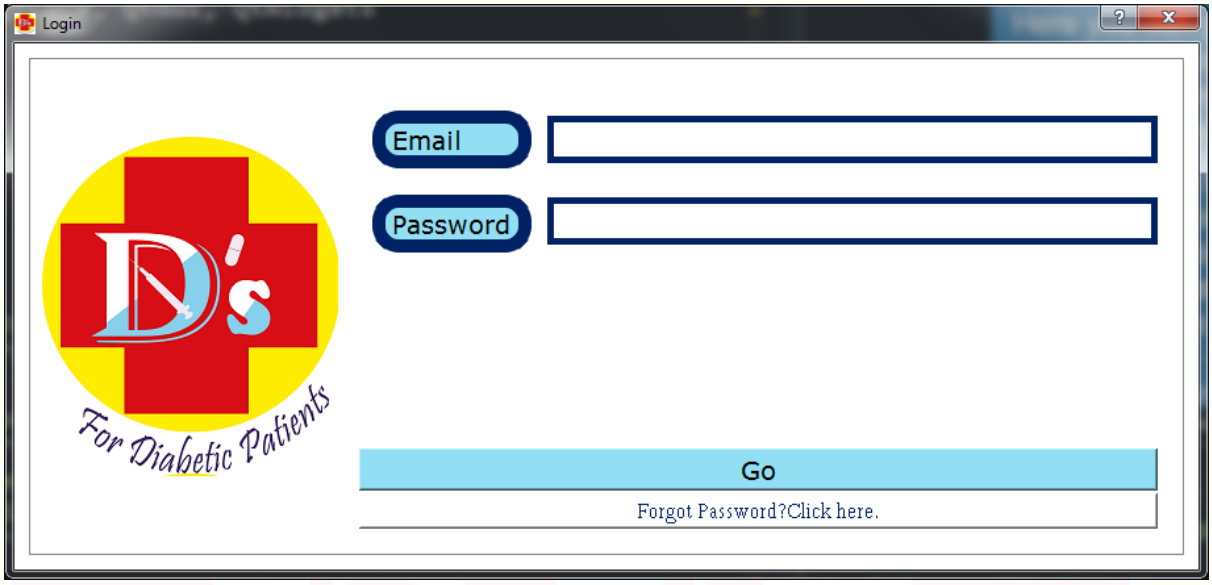
Steps for starting the application

- Install the windows installer file
- After the installation gets over you may check your desktop and then you will find a file there named **"D's"**
- Double click on the file and then you will see the sign up page is opened.
- Fill all the details and note the password.

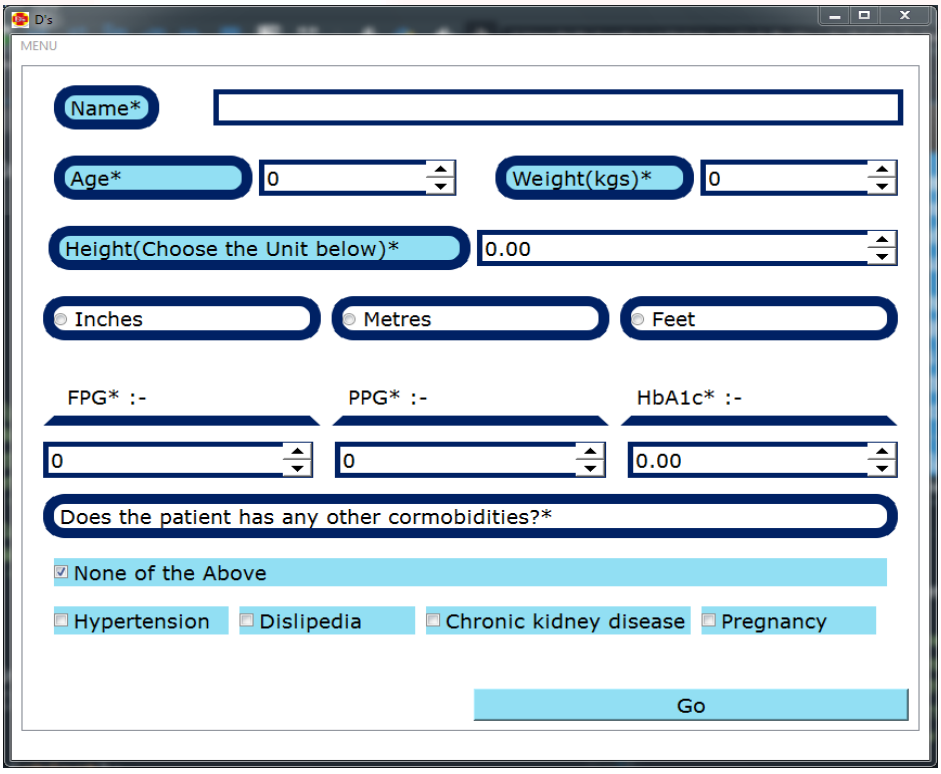


- After successful sign up you will be redirected to the login page enter your password and email here and then click go.

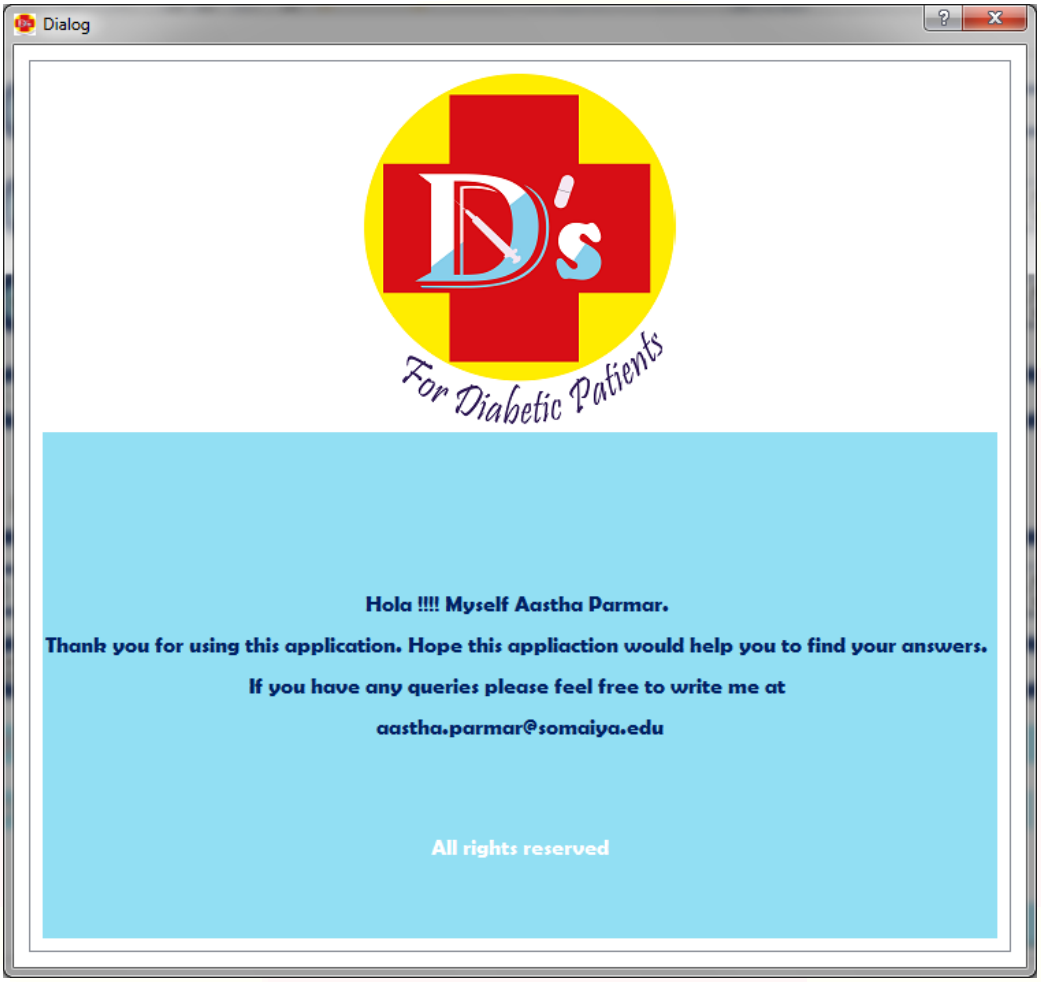
- Or else if you forget your password, click on the "**Forgot Password? Click here**" button and then write an email will be sent to you which contains a unique link don not share that link to anybody, click on the link now you may reset your password.



- After successful login you will be able to see the main home page
- Enter all the required details(*) and click GO button



- For complain and queries go to the about page in the menu and then contact using the email.



Thank you.

Chapter 7:

Conclusion

7.1 Significance of the system

With promising User Interface and correct prediction results the application is commercially made for the Inches Group. The user interface is responsive and can be used on platforms like Mac, Linux and Windows. Each and every module of the project is tested in every possible way the SDLC architecture proved to be very helpful as each and every module was tested as soon as it was built so the errors were removed before implementing it with the other module. So there were very less errors to debug after the whole thing was implemented and brought together. This application is created such that if it gets more data, the model will work more and more effectively and can be used for many other diseases and patients can directly use them, the idea towards making this application is that it can be used in regions where doctors are not available or the patients are unable to contact the doctor for many reasons, so here the application can be used for those people by just running the program anywhere possible they can get their desired prescription as the data is also given by a doctor who is a consultant in WHO and founder of INCHES Group and a professional medical practitioner. Very big thanks to him who gave me the data to start this beautiful research.

If further more data is provided then we can make different models and implement them for different diseases and help biology and humanity, and make human life more and more quality oriented and healthy!!

7.2/3 Limitations of the system and Future scope of the system

The processing capabilities have actually shown promising results in the predictions, Machine learning and Deep learning have tremendous capacity of automating everything using some mathematical calculations and some data mining and shaping we can build various automated applications, but it has some limitations not every physician will be tending to use this application or approve this application, although the application shows correct results everyone will need time to accept that a machine will be able to give prescription like a doctor to the patients. Many fake biological applications have made it impossible to make this kind of an application acceptable, still remembering when I was explaining my idea to my mentor she was just unable to accept what the application could do, because machine learning applications which are famous are used for detecting whether the patient has that type of disease or not but machine learning was never used to write a prescription to the patient. There are several reasons the main and important reason is just the "**data**", it is very difficult to collect data as not every doctor is willing to share the data or someone who is willing to share has inappropriate data (maybe his patients are getting horrible day by day or the data has a lot of noise.) sometimes the data which is shared is very less and hence it tends to over fit. Machine learning needs lots and lots of data and gathering that data takes a lot of time. And when we have lots of data then we get normal computers or laptops don't have that computation power and hence we need GPU high processing GPU's which will make the processing faster and faster but due to their high price they are not available for everyone. So these are the drawbacks that makes difficult for machine learning algorithms to compute.

References:

- *Thoughtful Machine Learning with Python- A Test Driven Approach by Matthew Kirk.*
- *Introducing Machine Learning – by MathWorks*
- *Introduction to Machine Learning with Python-A GUIDE FOR DATA SCIENTISTS by Andreas C. Müller & Sarah Guido*
- *https://scikit-learn.org/stable/*
- *https://pandas.pydata.org/*
- *https://matplotlib.org/*
- *https://console.firebase.google.com/*
- *https://play.google.com/store/apps/details?id=com.xtelltechnologies.diabetestreatment&hl=en_IN*
- *https://play.google.com/store/apps/details?id=com.xtelltechnologies.diabetestreatment&hl=en_GB*