LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

**CS6P05NI**

# Final Year Project

# Disease Prediction (E Check-up)

**Year and Semester**

**2019-2020**

**Autumn**


**Submitted By:**
Rishav Dahal

17031146(London Met ID)


**Submitted To:**

**Internal Supervisor**

Subekshya Shrestha

**External Supervisor**

Ishwor Shrestha

Group: C8

Submission Date: 7th Jan 2020

Word Count: 2,932

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*

## Abstract:

This project is developed as an informative system which maps the possible symptoms that a person is facing into the most probable disease. It is a Machine Learning application which makes use of multiple algorithms to draw a conclusions based on the user symptoms. The user can select multiple symptoms and they are collectively directed towards one of each disease based on algorithms. Multiple predictions are made based on the data set by which the model is trained. Classification method is used and three different algorithms with different accuracy and complexity are used and a final prediction is obtained with acceptable error rate. Data is one of the major players in this project and data are accessed through a study conducted at Columbia University. This project aims to provide an open web-based application to general people who can pre analyze their symptoms before visiting to a medical professional.

# Contents

Table of Figures:

Table of tables:

# 1) Introduction:

## **1.1** Overview and Motivation

With wide coverage of Internet technology today, people around the globe are familiar with search engines and internet surfing techniques. This upgrade in technology has many good aspects but some downsides also. People often tend to search every problem they have on the internet. Internet Technology have updated many fields and made the working easier.

Talking precisely about Medical field, technology is improving medical care with the use of Health wearable, Artificial organs, Telemedicines, etc. Talking about the other side of medical technology, we receive unnecessary search results in internet which makes us think once about that matter.

## **1.2** Problem Statement

When we type in the basic symptoms that we might be observing for few days in Google or Yahoo or any other search engines, we often get through unnecessary conclusions or predictions that mostly ends in death. Search engines are vague and provides both authorized and unauthorized results.

**eCheckUp** is intended to solve that particular problem by pre-analyzing the symptoms to conclude the possibility of one or more particular disease so that we can proceed into further treatment consulting medical professionals.

## **1.3** Objective Of The Project

### **1.3.1** General Objective

To implement different machine learning algorithms into a specific data set by training a model and compare the outputs of the algorithms with different accuracy rate and determine the best algorithm into the scenario.

### **1.3.2** Specific Objective

To develop a web application which maps your occurring symptoms into one or more possible disease this helps to minimize vague disease prediction. To know where your symptoms are leading to and proceed with the necessary checkups and meet the specific medical professionals.

## **1.4** Scope Of The Project

The scope of the project are the general people who have certain recognized symptoms and are searching for any possible disease prediction. Displaying them some particular disease based on their symptoms is the main working of the project.

# CHAPTER 2: BACKGROUND RESEARCH

### **2.1)** About the client/end-user

The targeted end-user for this system is general public. As the number of diseases are increasing in number and people are being panicked due to some minor disease. The main use of this application is to test those minor disease and do not let people to panic. This will be helpful for the people as the cost of seeing doctor is saved. They will also be safe from fraud.

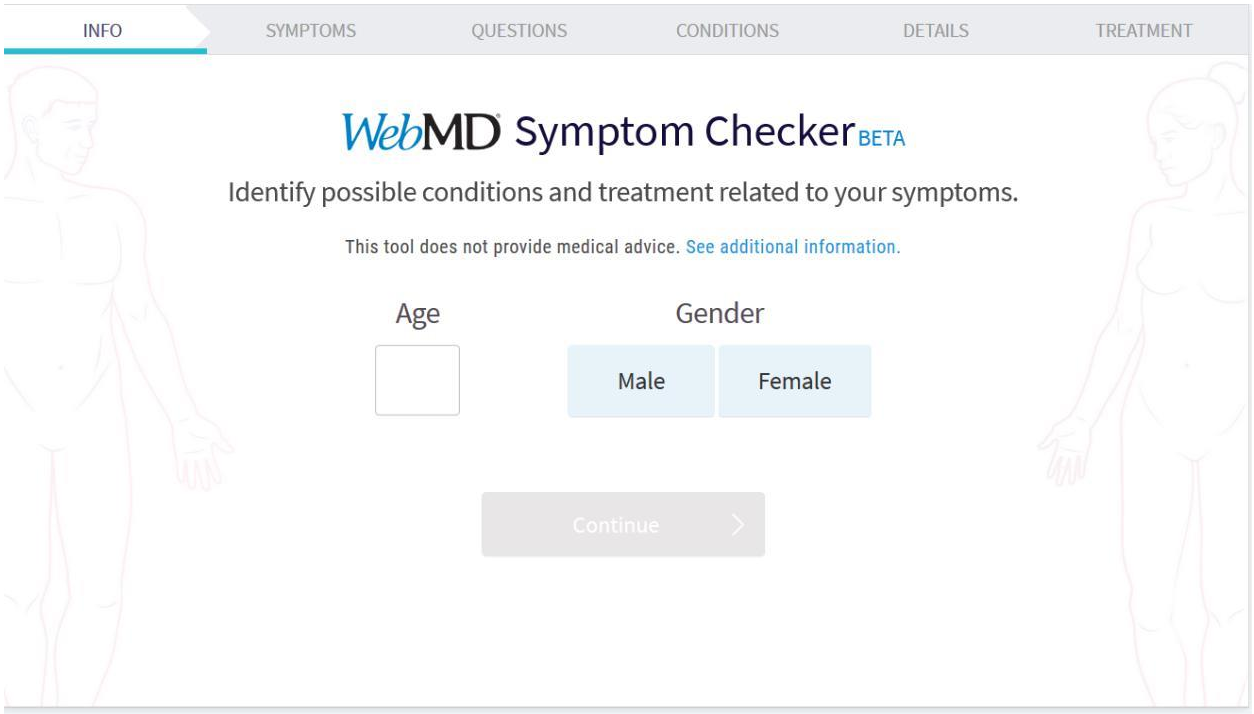
## 2.2. Literature Review

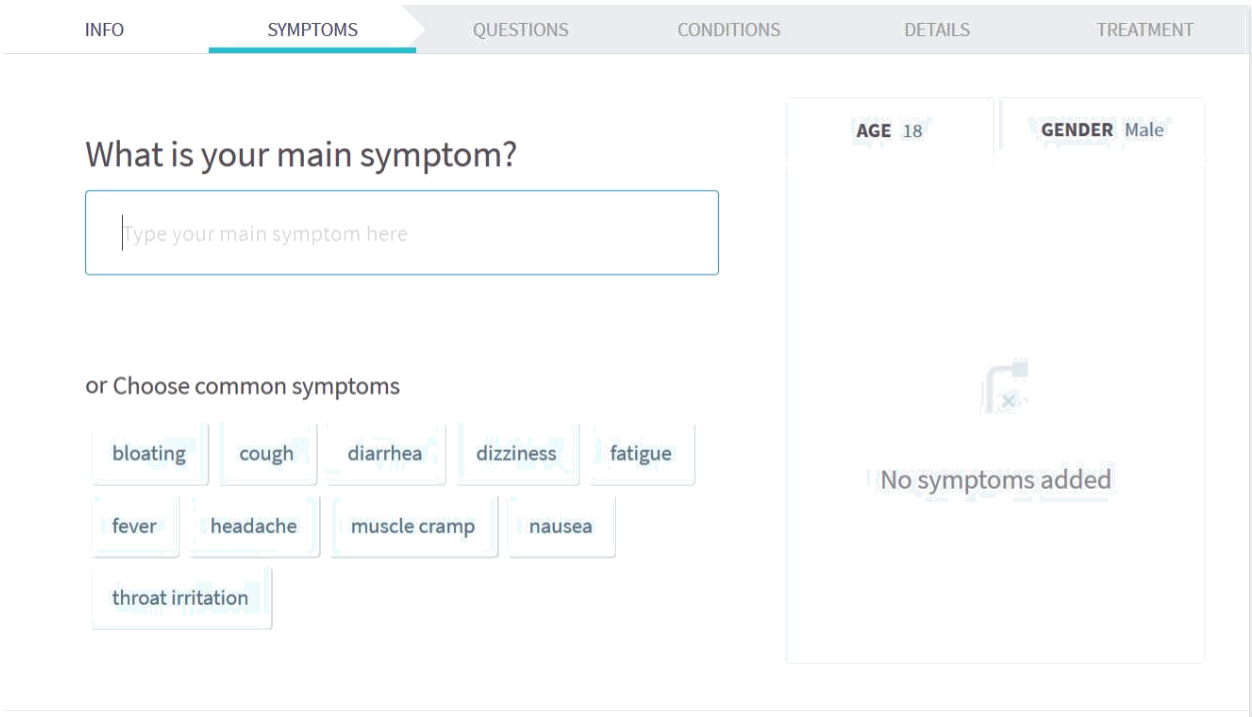
**Web MD Symptom Checker** (https://symptoms.webmd.com/default.htm#/info)

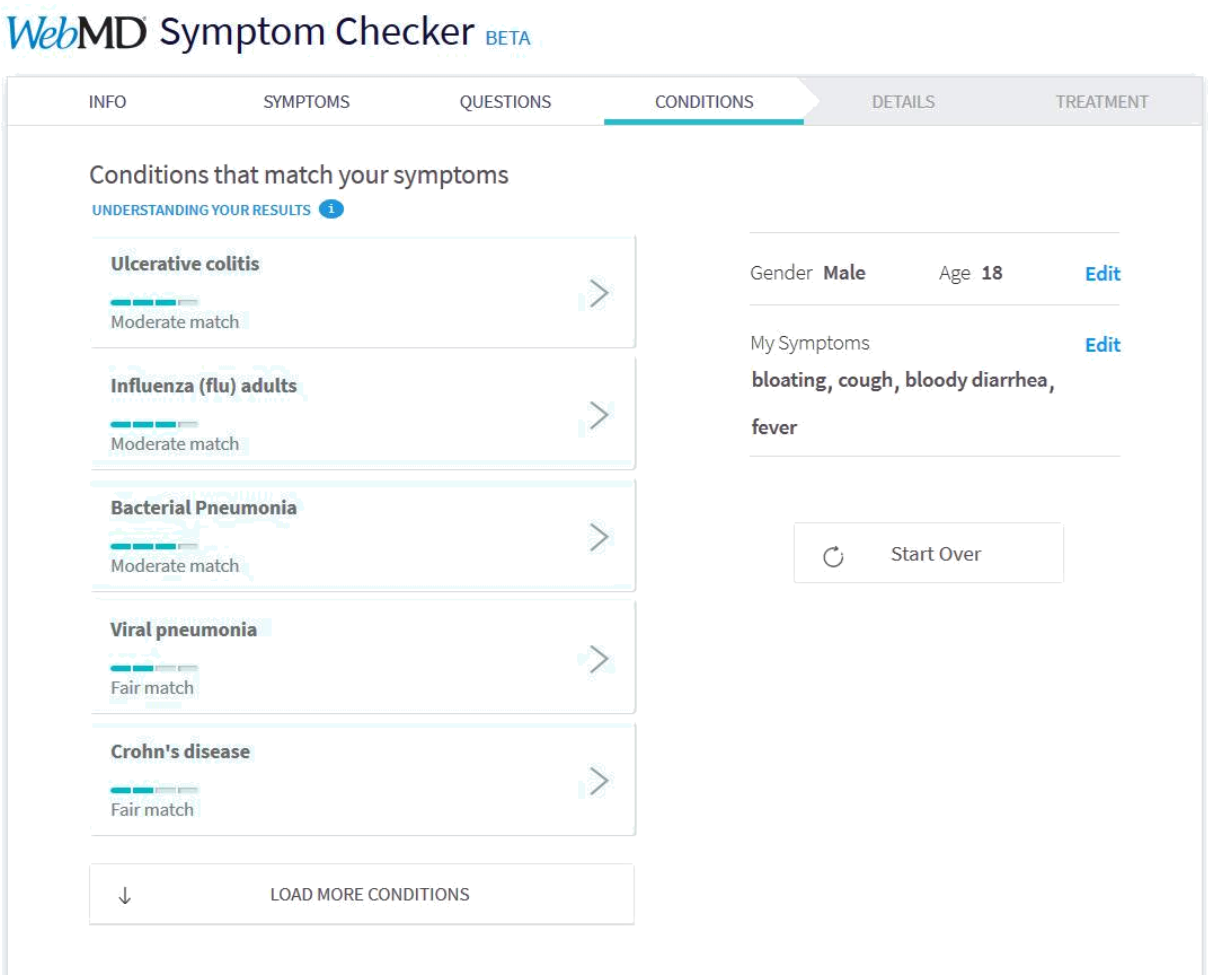
There is an online Symptom Checker application called WebMD Symptom Checker. It consists a full data set of symptoms and probable disease with conditions and treatments. eCheckUp is similar to WebMD Symptom Checker.

## 2.3. Current System

Talking about the current system i.e. Web MD Symptom Checker, it analyzes the symptoms of a user based on various factors as age, gender, symptoms, questions, conditions and details. It is a fully functioning application with big data set than eCheckUp. This project is a mini project inspired by the Web MD Symptom Checker.




INFO
SYMPTOMS
QUESTIONS
CONDITIONS
DETAILS
TREATMENT
What is your main symptom?
Type your main symptom here
or Choose common symptoms
bloating
cough
diarrhea
dizziness
fatigue
fever
headache
muscle cramp
nausea
throat irritation
AGE 18
GENDER Male
No symptoms added


WebMD Symptom Checker BETA
INFO
SYMPTOMS
QUESTIONS
CONDITIONS
DETAILS
TREATMENT
Conditions that match your symptoms
UNDERSTANDING YOUR RESULTS
Ulcerative colitis
Moderate match
Influenza (flu) adults
Moderate match
Bacterial Pneumonia
Moderate match
Viral pneumonia
Fair match
Crohn's disease
Fair match
LOAD MORE CONDITIONS
Gender Male
Age 18
Edit
My Symptoms
Edit
bloating, cough, bloody diarrhea, fever
Start Over

## **2.4.** The Problem With Current System

Since Web MD Symptom Checker is a complete working web application, there are no reported problem in it. It clearly states that it does not provide medical advice but shows some treatment methods. Web MD is in beta version and it does not show which data set it is making use of and accuracy. This is the problem with the current system but as a whole, it is a practical application with easy to use GUI and Treatment feature included.

# CHAPTER 3: SPECIFICATION AND DESIGN

## 3.1. Requirement Elicitation And Analysis

The basic requirement of this project is a computer system which allows python applications to run. This project is based up on Python3, so the computer must support Python program to be executed successfully. Data set and training of the model is also major requirement of this project. The data set is divided into two parts: Training data set and Testing data set. The training data set is used to train the model while the testing data set is used to test the model. The accuracy obtained is approximately 95%.

### 3.1.1. Functional Requirement

Functional requirements of eCheckUp can be:

- The User should open the web application
- The system should ask for the Name of the user and 5 symptoms as input
- The user should click on predictions to create 3 predictions based on 3 different algorithms
- The user should be able to understand the predictions and understand them for further processing and also save the data to save the user info in the system.
- Python3 and libraries as numpy, tkinter and pandas.

### **3.2.2.** Non-Functional Requirement

Similarly, some of the Non-functional requirements are:

- The data must be in CSV format to train the model
- The format of data must be in correct order as it is places
- The GUI must be designed correctly to provide correct output

## **3.3.** System Design

The design of this system is easy compared to other system. This project do not contain any databases so diagrams like ER, Relational can be omitted. The system is designed using web markup languages and the processing part was done using python while GUI is made on a framework of python called as Tkinter.

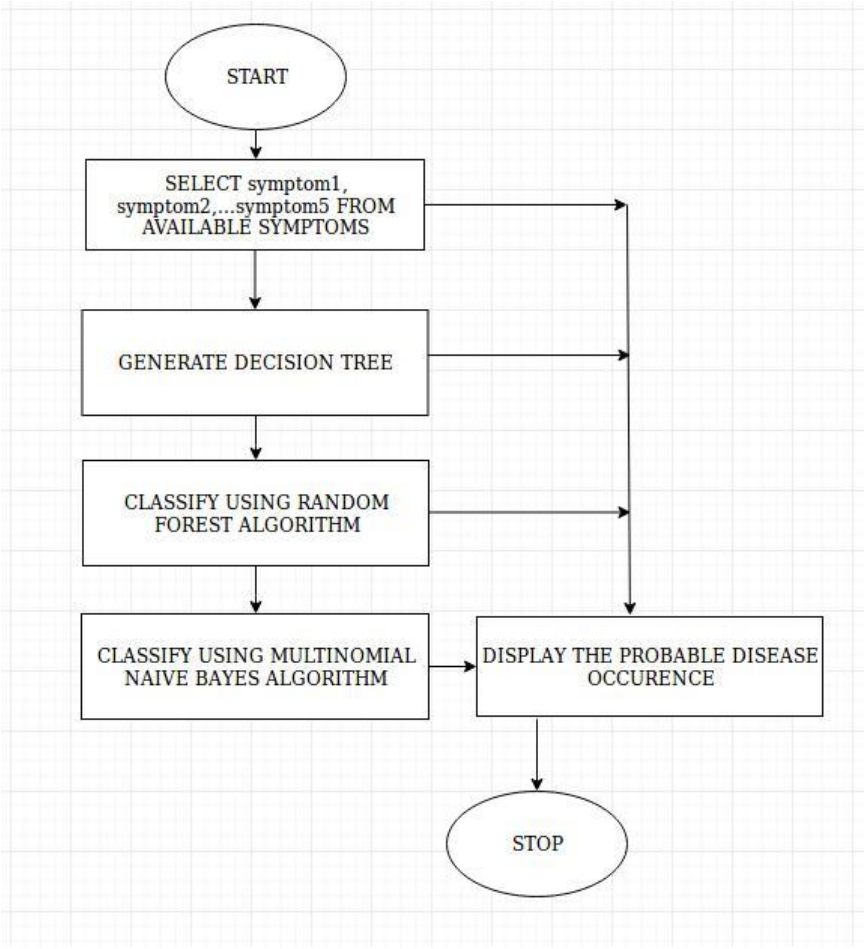



*Figure 1 WorkFlow Diagram*

This is the work flow chart of the system. The user can select upto 5 symptoms from a given list of symptoms and can proceed to obtain the predictions. The program calls 3 functions: Decision Tree, Random Forest and Naive Bayes algorithms generate the outputs and the probable disease based on all of them individually is displayed. There is a save option which enters the user info into a file.

### **3.3.1.** Use Case Diagram

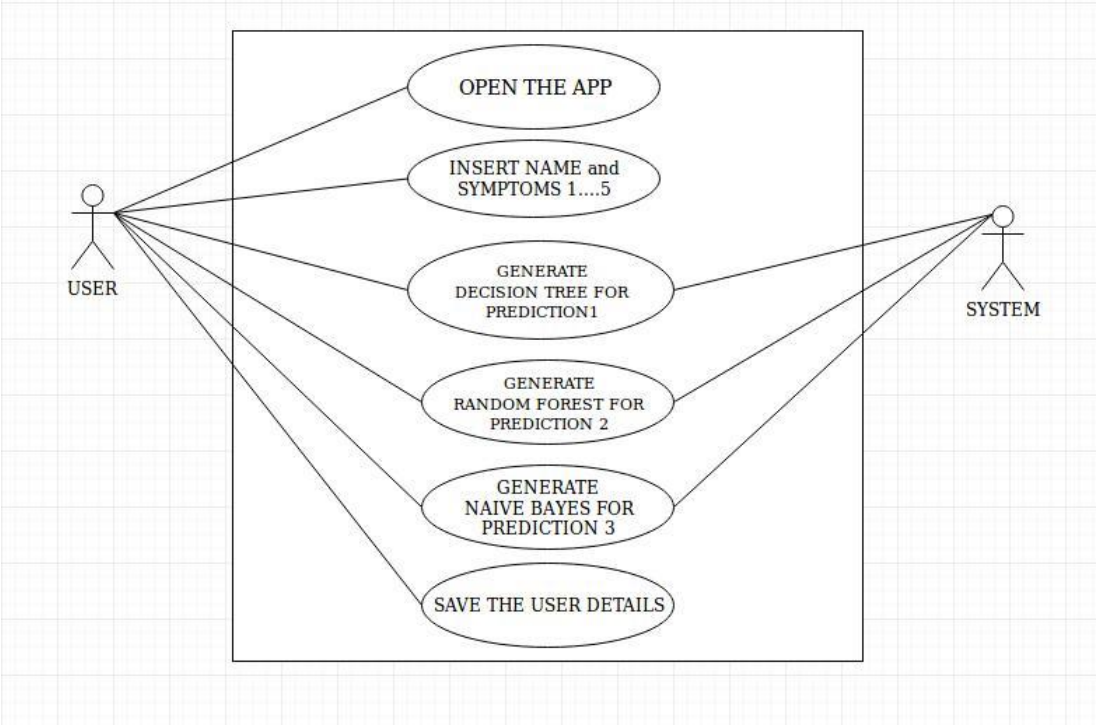



*Figure 2 Use Case Diagram*

There are two actors in this project: User and System. User has access to the GUI part of the project while System can generate the predictions that can be viewed by the User. The above is the use case diagram of the project.

# CHAPTER 4: IMPLEMENTATION AND EVALUATION

## **4.1.** Tool And Technology

The tools and technology used in this project along with their function are listed below:

- Python3 is used as the prime coding language
- Python Libraries as numpy, pandas and tkinter is used to ease the working of the project
- CSV format is used to place the training data, testing data and save records of patients
- Any IDE that supports python is used to develop the project.

## **4.2.** Implementation

The Implementation of this project is all to deal with mapping user inputs i.e. symptoms into probable disease with an acceptable error rate. The Classification method ML will used including three different algorithms: Decision Tree, Random Forest and Naive Bayes. These all algorithms have their different complexity measure and accuracy level. Working with three algorithms will help to compare the outputs on a given set of input. The main challenge is to obtain accurate result based on the trained model using the data set.

A **Decision Tree** is a map of the possible outcomes of a series of related choices. It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits. They can be used either to drive

informal discussion or to map out an algorithm that predicts the best choice mathematically. (lucidchart, 2019)

The **Random Forest** algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random

Forest".The following are the basic steps involved in performing the random forest algorithm:

1. Pick N random records from the dataset.

2. Build a decision tree based on these N records.

3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

4. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote. (Synced, 2019)

A **Naive Bayes** classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis.

A naive Bayes classifier is not a single algorithm, but a family of machine learning algorithms that make uses of statistical independence. These algorithms are relatively easy to write and run more efficiently than more complex Bayes algorithms. The formula used for naive Bayes is:

$$P(c|x) = (P(x|c) * P(c)) / P(x)$$ (Brownlee, 2019)

These three algorithms will be implemented in the project with their functions to predict the disease.

### **4.3.** Evaluation

This application is intended to show the disease from symptoms and it was made so. It contain a simple and easy-to-use design which has no difficulty in using. Users after inserting up to 5 symptoms by selecting from the list are able to predict the probable disease they might be suffering from. This application provides no medical evidences but however it could be as a pre requisite before visiting to any medical organization or professionals. Talking about one of the predictions that could be made was if you are suffering from mild fever, belly pain, loss of smell, congestion and diarrhoea can be possibly typhoid and user can proceed to test for it by visiting a doctor.

## 5) Development

For the development phase, I have considered three methodologies i.e. Waterfall Model, Agile Methodology and Iterative Model.

### **5.1)** Waterfall Model:

The Waterfall Model was the first Process Model to be introduced. It is very simple to understand and use. In a Waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. The waterfall model is the earliest Software development life cycle approach that was used for software development.

In the waterfall approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of

Conception, Initiation, Analysis, Design, Construction, Testing, production/Implementation, and Maintenance. (toolsqa, 2018)

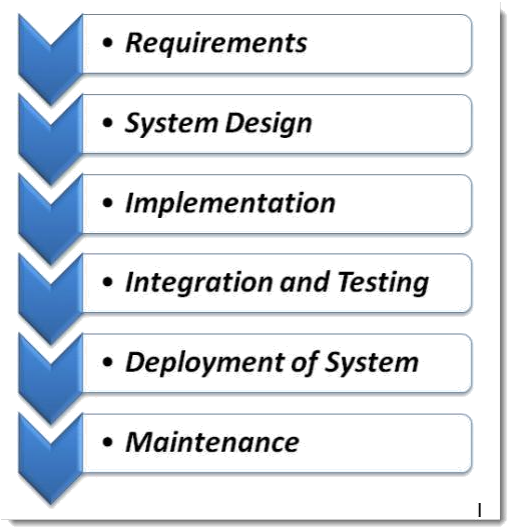



*Figure 3 Water Fall mode*

## **5.2)** Scrum Methodology:

Scrum Methodology is a simple framework that facilitates team collaboration on complex projects. It is that scrum is easy to understand but it is hard to master.

Certainly, such type of a methodology is named after the cooperative rugby strategy in which teams gain distance by passing the ball back and forth. Although Scrum is actually an "inspect and adapt" framework for software development and not a project management methodology, per se, it has applications for other types of projects, and thus is considered by many to be an agile project management tool. Another good thing about it is that it minimizes peril by creating software in short time boxes, known as iterations, which happen to last from one week to one month. (projectmanager, 2020)

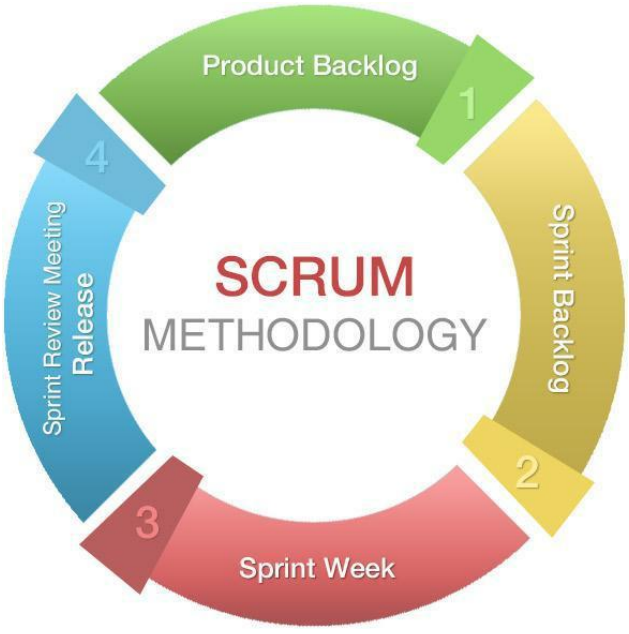



*Figure 4 Scrum Methodology*

**5.3)** Iterative Model:

I have chosen to utilize iterative model for my project. My tasks don't endeavour to begin with full particular of necessities. Rather improvement starts by determining and execute portions of the application. Every one of web customers will be assessed so as to improve and recognize further prerequisites in each stage. The procedure is then repeated, creating better server and the customers. My undertaking is evaluated to be done in 24 weeks.

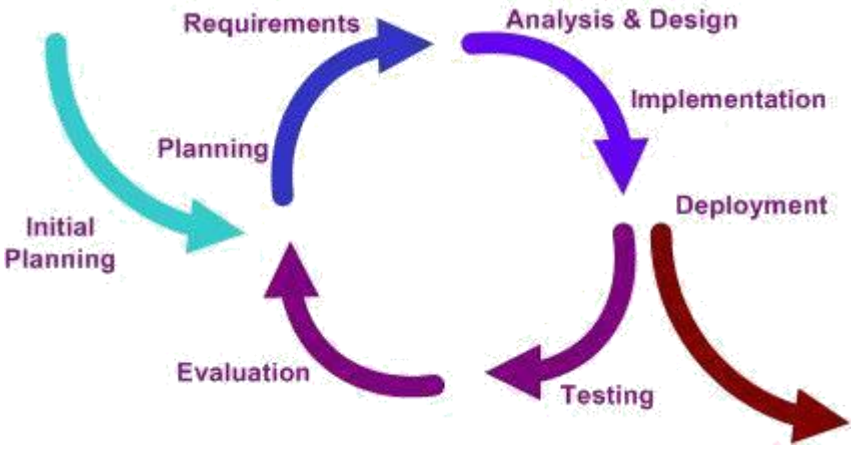



*Figure 5 Iterative Model*

Considering an iterative life cycle model, it consists of repeating the following four phases:

- Requirements phase
- Design phase
- Implementation Phase
- Review Phase

Initial Planning

In this stage, all the arranging of the project is finished. Initially, I will make arrangements and guidelines to finish my activities in due time. I will set time period for each piece of the tasks. I will recognize my weakness in topic and dispense more opportunity to improve it. I will likewise investigate about the subjects required for the application advancement. I will also research about the algorithm used for analysis of symptoms.

Requirement Phase

In this stage, requirements for the software are accumulated and broke down. I will begin with essential requirements for web customer. I will begin by coding straightforward working code which can recognize diseases. With every Iteration requirement will help to grow bigger that give more highlights.

Design Phase

This stage starts with basic program plan and database structure. UI and UX design of server and customers is executed when the necessity is made. In this stage, requirements are outlined in configuration structure. This might be another program structure, or an augmentation of a prior plan. Every cycle stage will give reliable and straightforwardplan.

## Implementation

In this stage, execution of design and requirement are finished by coding. Most importantly, server is composed utilizing Python. Web customers are composed when the requirement and design is made. Testing is additionally performed is this stage. In this stage customers are incorporated with the server and tested. With every cycle, server and customers are made bug-less and stable.

## Review Phase

A Review stage begins by assessing the server and customers written in execution stage. This phase also reviewed the current requirements that is weather the feature made in requirement phase is implemented or not. If the feature are advanced and is not possible to implement, then the requirements are changed. Nature of code is additionally assessed in this stage. Iteration makes the survey increasingly powerful and productive.

## Deployment Phase

This is the final steps where the final output of a project can be find. In this phase, users can find the stable and final server and client that can be run in their devices.

Cycle Diagram:

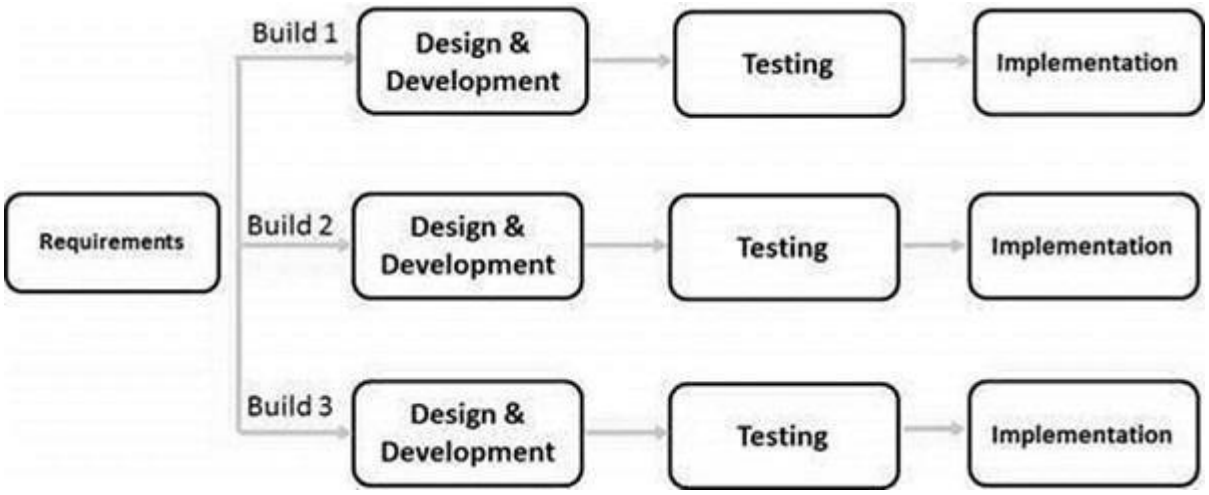



*Figure 6 Cycle Diagram*

## 6) Test/Analysis:

Test is an integral part of the software development. After the proper testing only, any software will be ready for launch. While developing this project I too had to test the system. As I am a learner of these algorithm used in this system, I had to test the system mostly. The main part of development of this system is testing and fixing the issues. So, after the completion of the system I made a test plan for testing the system.

### Test Plan:

| | |
|---|---|
| Execution of the GUI (Unit Testing) | Success |
| Selection of the symptoms (Unit testing) | Success |
| Results as per the Decision Tree in accordance of the selected symptoms (System Testing) | Success |
| Results as per the Random Forest in accordance of the selected symptoms (System Testing) | Success |
| Results as per the Naïve Bayes in accordance of the selected symptoms (System Testing) | Success |
| Predicted data is being saved locally in a csv file. (System Testing) | Success |

*Table 1 Testing Plan*

## 6.1) Test cases

### Test case 1: Execution of the GUI

| | |
|---|---|
| Test Type | Unit testing |
| Performed action | The command for e-checkup were executed. |
| Expected Outcome | The GUI of e-checkup should be launched. |
| Obtained Outcome | The GUI of e-checkup is launched. |
| Test Result | Success |
| Conclusion | The above performed test was necessary in order to confirm whether the methods set for launching the GUI of the E-checkup were functioning or not. |

*Table 2 Test case 1*

Proof for test case 1:

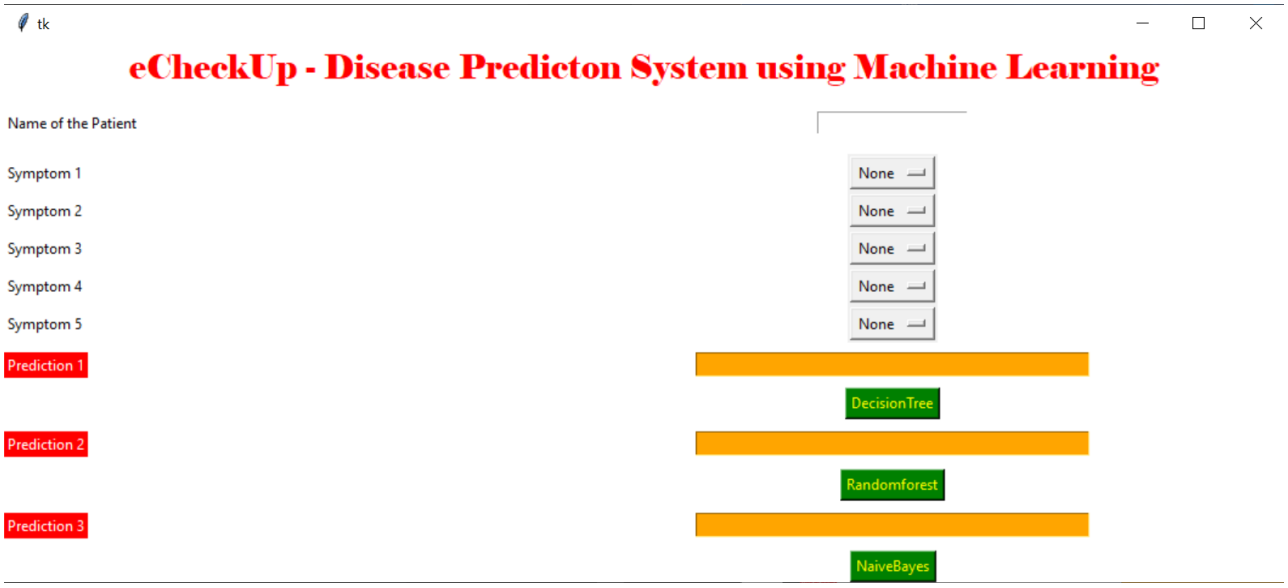



*Figure 7Proof for test case 1*

Test Case 2:  Selection of the symptoms

| | |
|---|---|
| Test Type | Unit testing |
| Performed action | The symptoms were selected. |
| Expected Outcome | The symptoms should be selected. |
| Obtained Outcome | The symptoms are selected. |
| Test Result | Success |
| Conclusion | The above performed test was necessary in order to confirm whether the methods set displaying the list of symptoms were executing properly. |

*Table 3 Selection of symptoms*

Proof for test 2:

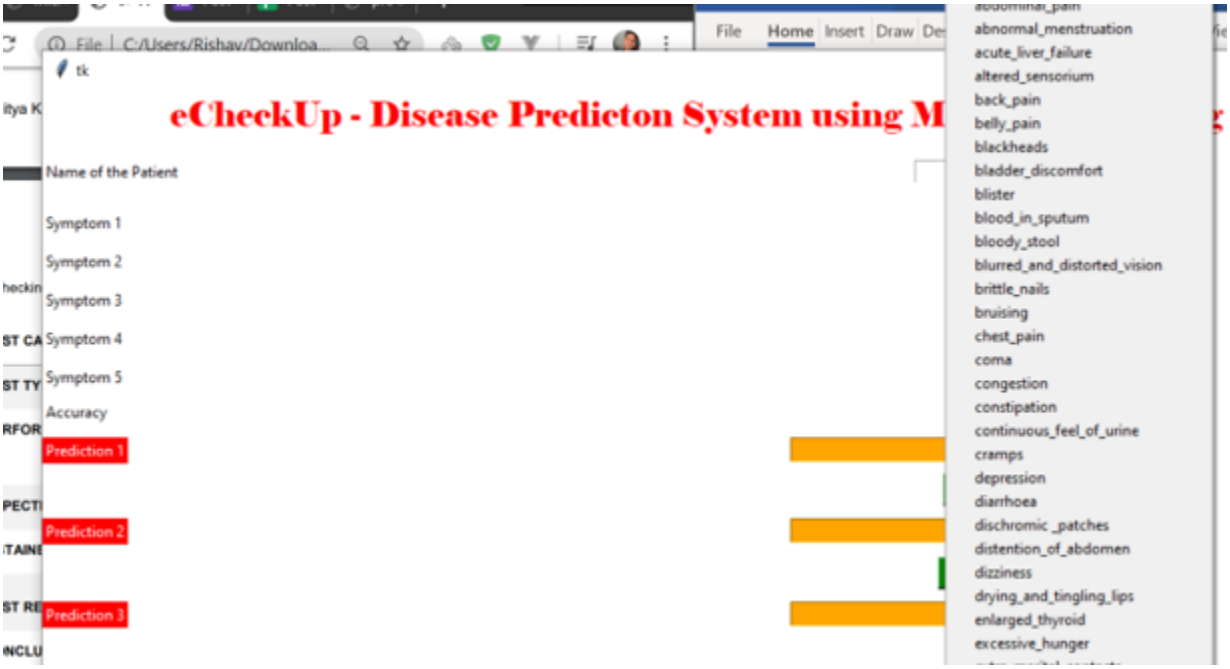



*Figure 8 Proof for test 2*

This is the process to select symptoms. When the user clicks in the area, these symptoms are appeared and can choose & select the symptoms.

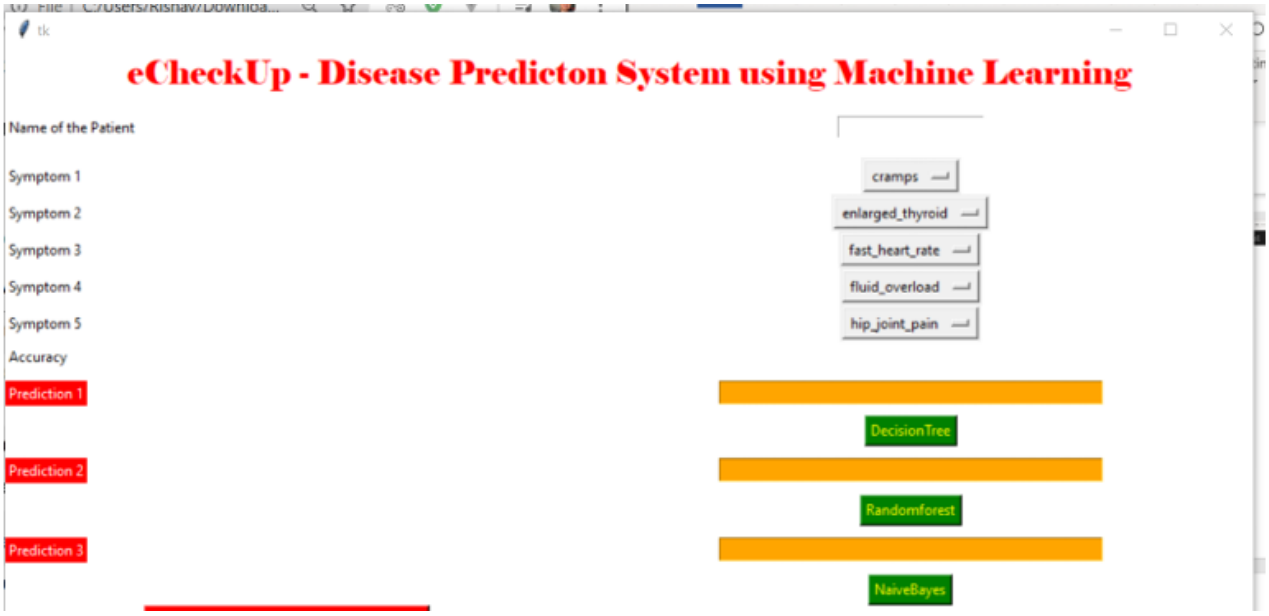



*Figure 9 Second proof for test 2*

This is the picture of the system after the selection of the symptoms.

Test case 3: Results as per the Decision Tree in accordance of the selected symptoms

| | |
|---|---|
| Test Type | System testing |
| Performed action | The button labeled Decision Tree was clicked. |
| Expected Outcome | A probable disease should be displayed in the text field. |
| Obtained Outcome | A probable disease is shown in the text field. |
| Test Result | Success |
| Conclusion | The above performed test was necessary in order to confirm whether the methods set for prediction using decision tree is processed and displayed. |

*Table 4 Results as per the Decision Tree in accordance of the selected symptoms*

Proof for test 3:

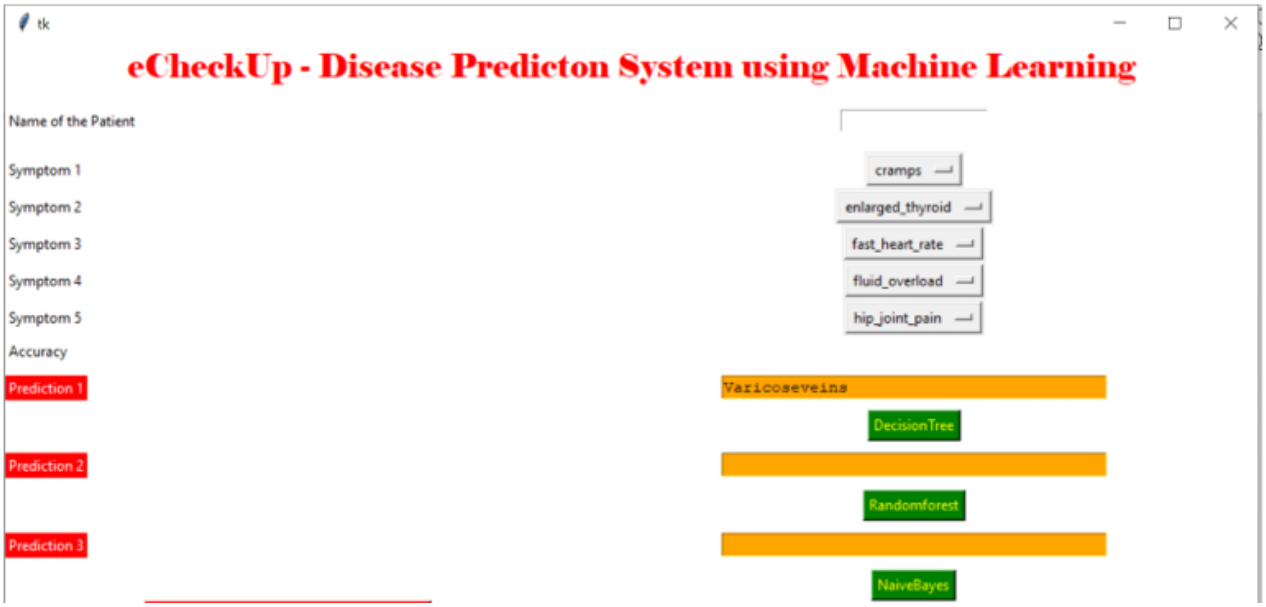



*Figure 10 Proof for test 3*

This image shows us the result or prediction done by the Decision tree.

Test case 4: Results as per the Random Forest in accordance of the selected symptoms

| | |
|---|---|
| Test Type | System testing |
| Performed action | The button labeled Random Forest was clicked. |
| Expected Outcome | A probable disease should be displayed in the text field. |
| Obtained Outcome | A probable disease is shown in the text field. |
| Test Result | Success |
| Conclusion | The above performed test was necessary in order to confirm whether the methods set for prediction using random forest is processed and displayed. |

*Table 5 Results as per the Random Forest in accordance of the selected symptoms*

Proof for test 4:

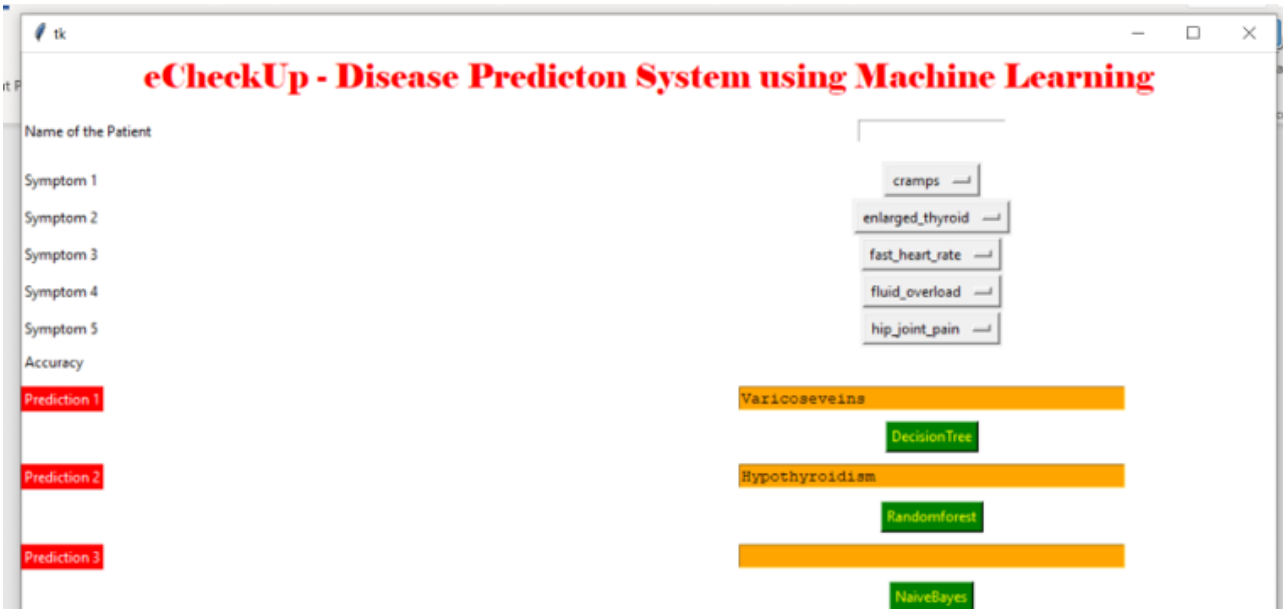



*Figure 11 Proof for test case 4*

This image shows us the result or prediction done by the Random Forest.

Test case 5: Results as per the Naïve Bayes in accordance of the selected symptoms

| | |
|---|---|
| Test Type | System testing |
| Performed action | The button labeled Naive Bayes was clicked. |
| Expected Outcome | A probable disease should be displayed in the text field. |
| Obtained Outcome | A probable disease is shown in the text field. |
| Test Result | Success |
| Conclusion | The above performed test was necessary in order to confirm whether the methods set for prediction using Naive Bayes is processed and displayed. |

*Table 6 Results as per the Naïve Bayes in accordance of the selected symptoms*

Proof for Test 5:

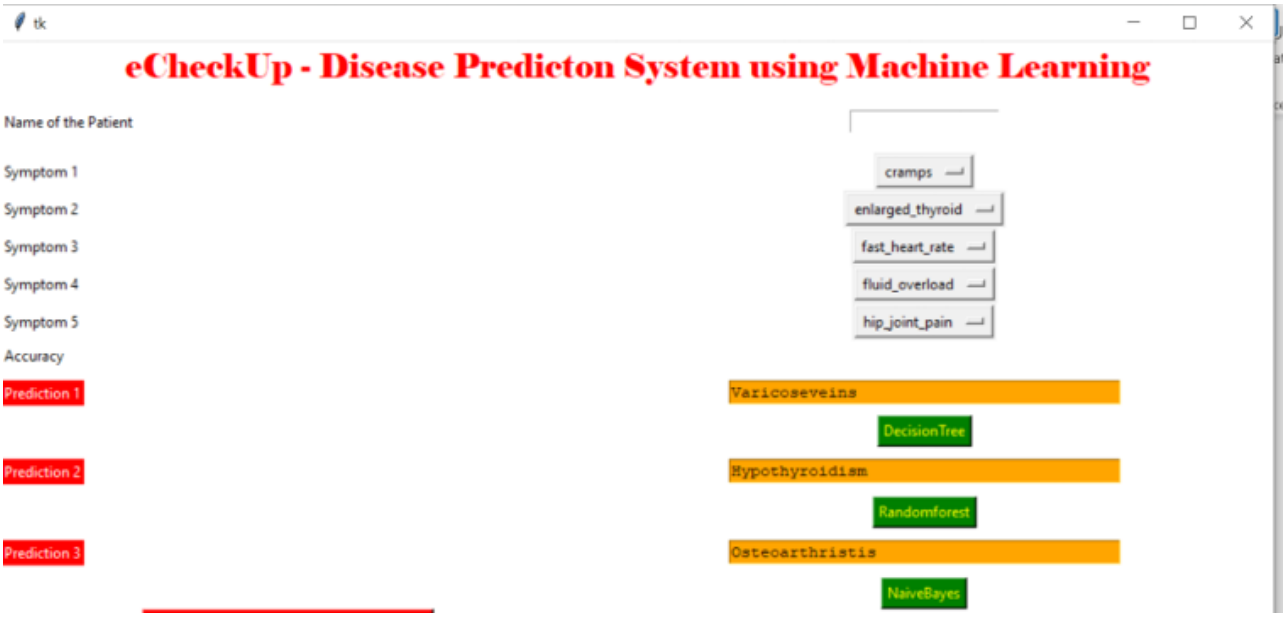



*Figure 12 Proof for test 5*

This image shows us the result or prediction done by the Naïve Bayes.

Test Case 6: Predicted data is being saved locally in a csv file.

| | |
|---|---|
| Test Type | System testing |
| Performed action | The process of prediction was completed. |
| Expected Outcome | A csv formatted file should save all the predicted data in user's local storage. |
| Obtained Outcome | A csv formatted file which was named as data has saved all the records. |
| Test Result | Success |
| Conclusion | The above performed test was necessary in order to confirm whether the methods set for exporting and saving the data in local storage is working or not. |

*Table 7 : Predicted data is being saved locally in a csv file*

Proof for test 6:

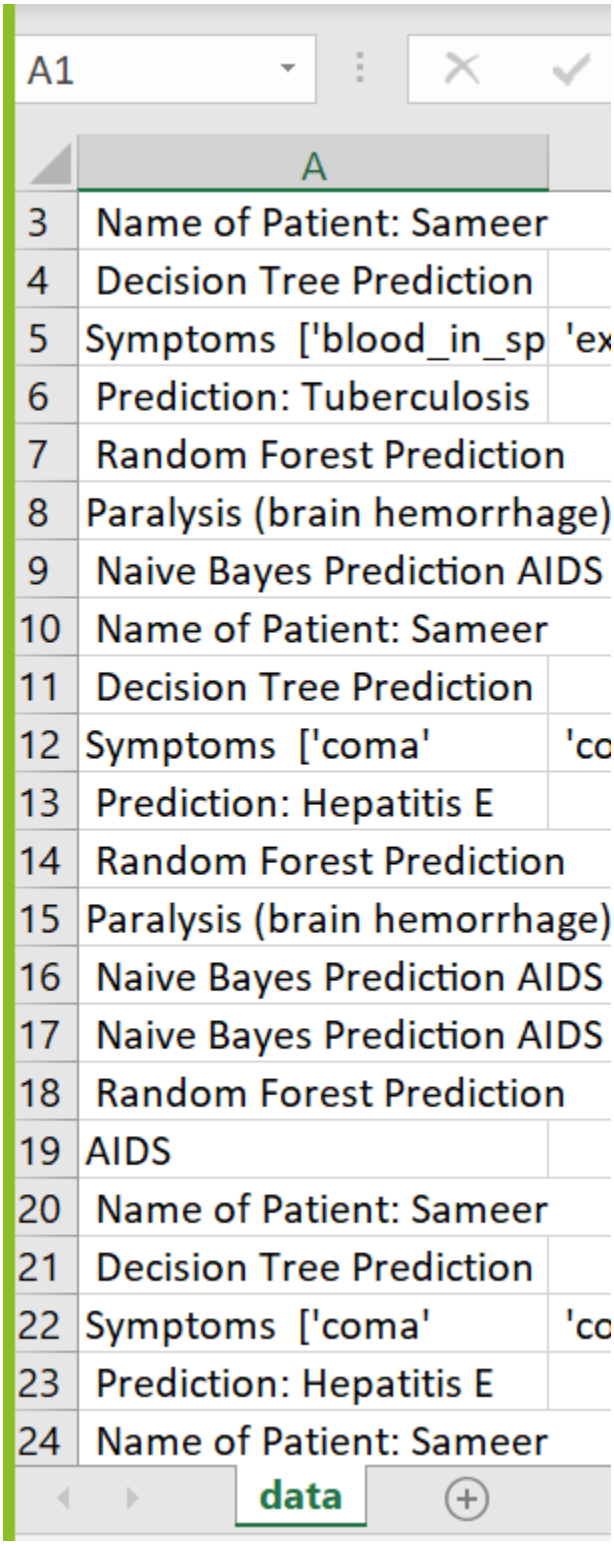

A1
A
3 Name of Patient: Sameer
4 Decision Tree Prediction
5 Symptoms ['blood_in_sp 'ex
6 Prediction: Tuberculosis
7 Random Forest Prediction
8 Paralysis (brain hemorrhage)
9 Naive Bayes Prediction AIDS
10 Name of Patient: Sameer
11 Decision Tree Prediction
12 Symptoms ['coma' 'co
13 Prediction: Hepatitis E
14 Random Forest Prediction
15 Paralysis (brain hemorrhage)
16 Naive Bayes Prediction AIDS
17 Naive Bayes Prediction AIDS
18 Random Forest Prediction
19 AIDS
20 Name of Patient: Sameer
21 Decision Tree Prediction
22 Symptoms ['coma' 'co
23 Prediction: Hepatitis E
24 Name of Patient: Sameer
data

*Figure 13 Proof for test 6*

## 6.2) Critical Analysis:

After all self-conducted unit and system tests were successful. The predction system is functioning properly with the features included. With the final ready system, test were conducted by the general public and their interaction with the system had some feedbacks.

The collected feedbacks pointed out some strong and weak points in system. The three decisions were making the public confused. I had to explain that there is a certain accuracy in every method of prediction. The public was curious about saving the data. So, I generated a function to export the data to a CSV format and save to their local storage. Most of them wanted it in their mobile phones and tablets. The user found this system abit uneasy to use as the disease predicted are new to them. This is all because of the datasets used in the system. The dataset used is scrapped from a study held in University ofColumbia(http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html). (J.Vijayashree, 2020)

# 7) Conclusion:

eCheckUp is a Disease prediction symptom based on ML classification technique. This project is developed for personal use and creates no medical evidences. It is not a substitute for professional medical advice, diagnosis or treatment. It is like a pre-requisite for medical professional. This project measures an accuracy of around 95% with very less error rate. Going through this project, one can know about where their symptoms are taking them towards. The three algorithms are compared and their different predictions which include two normal predictions and one final prediction.

Also, there is a data saving option which allows user to save data and store in a file including information as Name of the user, symptoms and predictions that are made. In this way, many other predictions can be extracted from the project and we can be aware of our probable diseases.

## 7.1) Legal, Social and Ethical Issues:

### a) Legal Issues:

This project aims to predict the disease based on the symptoms. The project is designed in such a way that the system takes symptoms from the user as input and produces output i.e. predict disease. Average prediction accuracy probability of 95%is obtained. This sort of system is not legally applicable in many countries. In Nepal, even the prescription should be hard copy. Soft copy prescription is filled as crime.

Clinicians, scientists, consumers, and patient groups have raised concerns about the data mining of the other patients and predicting the disease. They have been stating this is not possible to work on as every patient has different behavior. Eating behavior of every patient is difference. So, for this system there are many legal issues. Here doctors have filled a complaint against the draft protocol of in generic testing saying the privacy of the patient is violated. (Genetics and Public Policy Center, 2009)

### b) Social Issues:

According to the surveys conducted for the disease prediction system, it can be observed that the predication system has high scope of easily dissolving in the market. However, social harmony cannot be encouraged for the product due to its probable high risk. It has often been noticed that the emergence of new technologies has made people compromise on social interactions and healthy eating. The disease prediction system would likely increase those factors even further as they will be sure about the diseases and even may take the medicines without the permission of doctor. On the bright sight, they can be prevented from placebo (fraud from doctor, prescribing medicine to patient even s/he is not sick). They can track their health records.

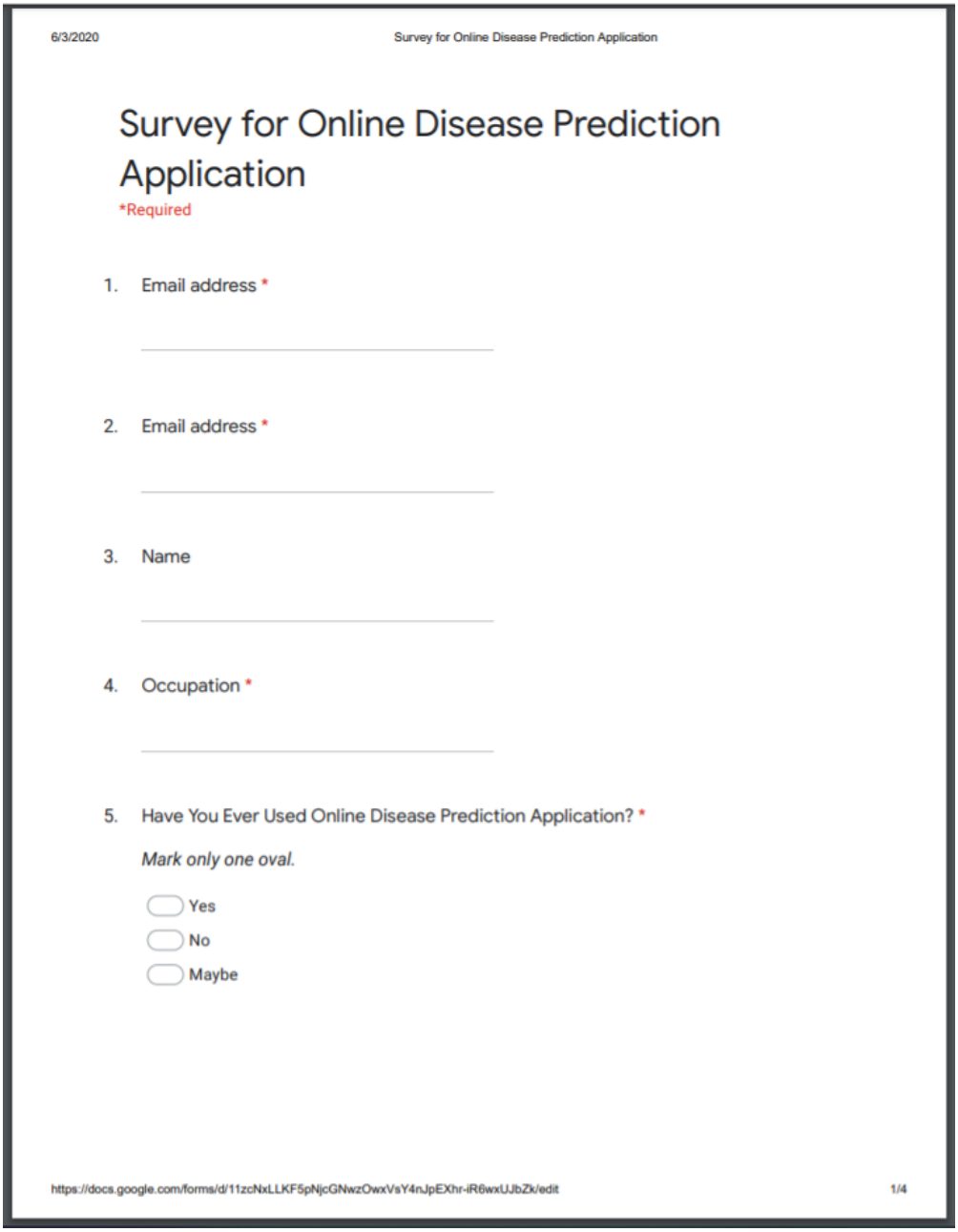
c) Ethical Issues:

As I have listed that the accuracy of prediction of my system is 95%, and this vary with the use of different dataset. While using the system, if a user gets a result of serious disease then the user might panic. Furthermore, this is a system generated for prediction not for accurate result. So, the blind faith can also be issue in this system.

# 8)APPENDIX

## 8.1) Appendix -A Pre-Survey

a) Pre-Survey Form:



6/3/2020 Survey for Online Disease Prediction Application

Survey for Online Disease Prediction Application

*Required

1. Email address *

2. Email address *

3. Name

4. Occupation *

5. Have You Ever Used Online Disease Prediction Application? *

*Mark only one oval.*

- Yes
- No
- Maybe

https://docs.google.com/forms/d/11zcNxLLKF5pNjcGNwzOwxVsY4nJpEXhr-iR6wxUJbZk/edit 1/4

*Figure 14 Screenshot of pre survey form 1*

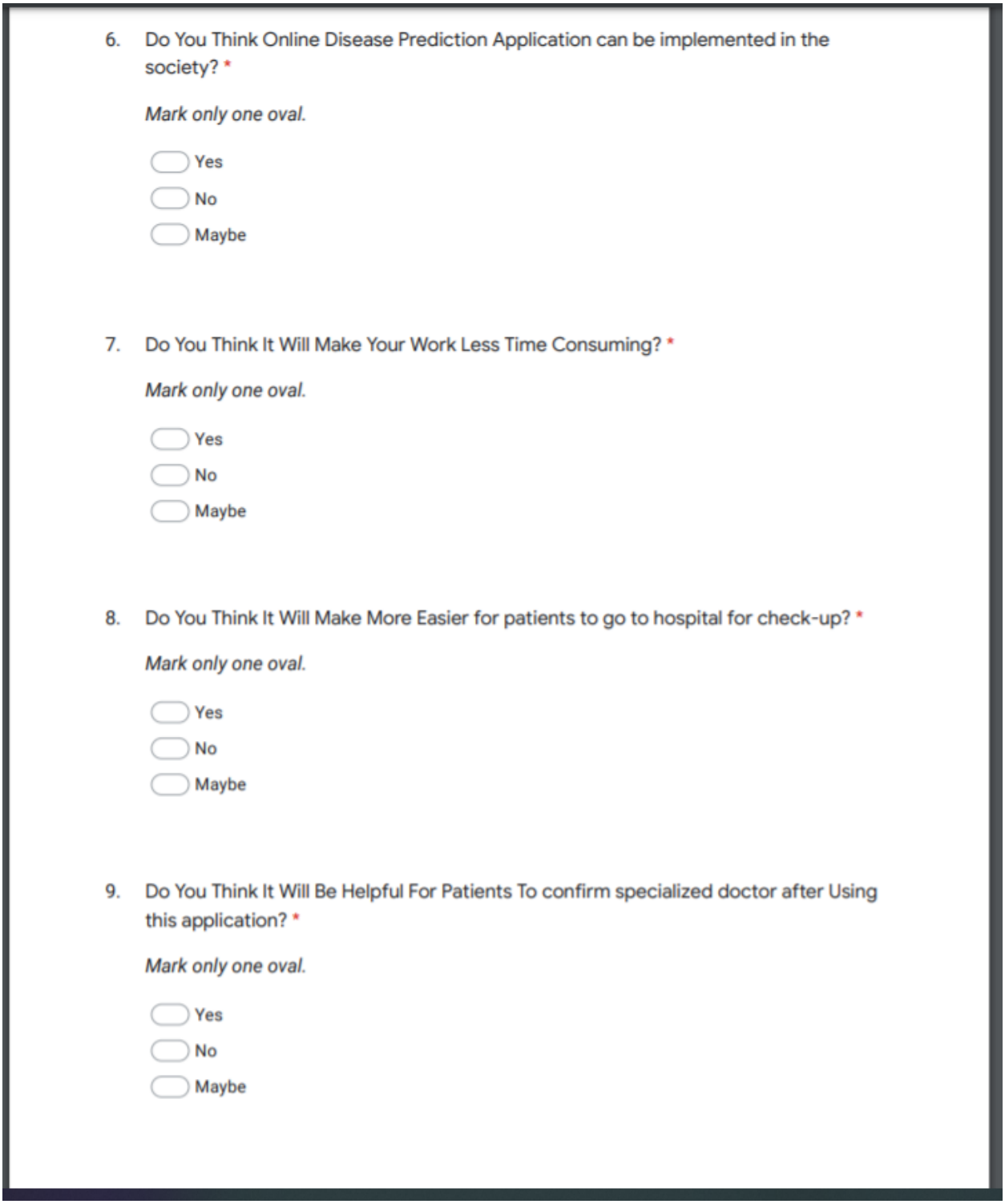



*Figure 15 Screenshot of pre survey form 2*

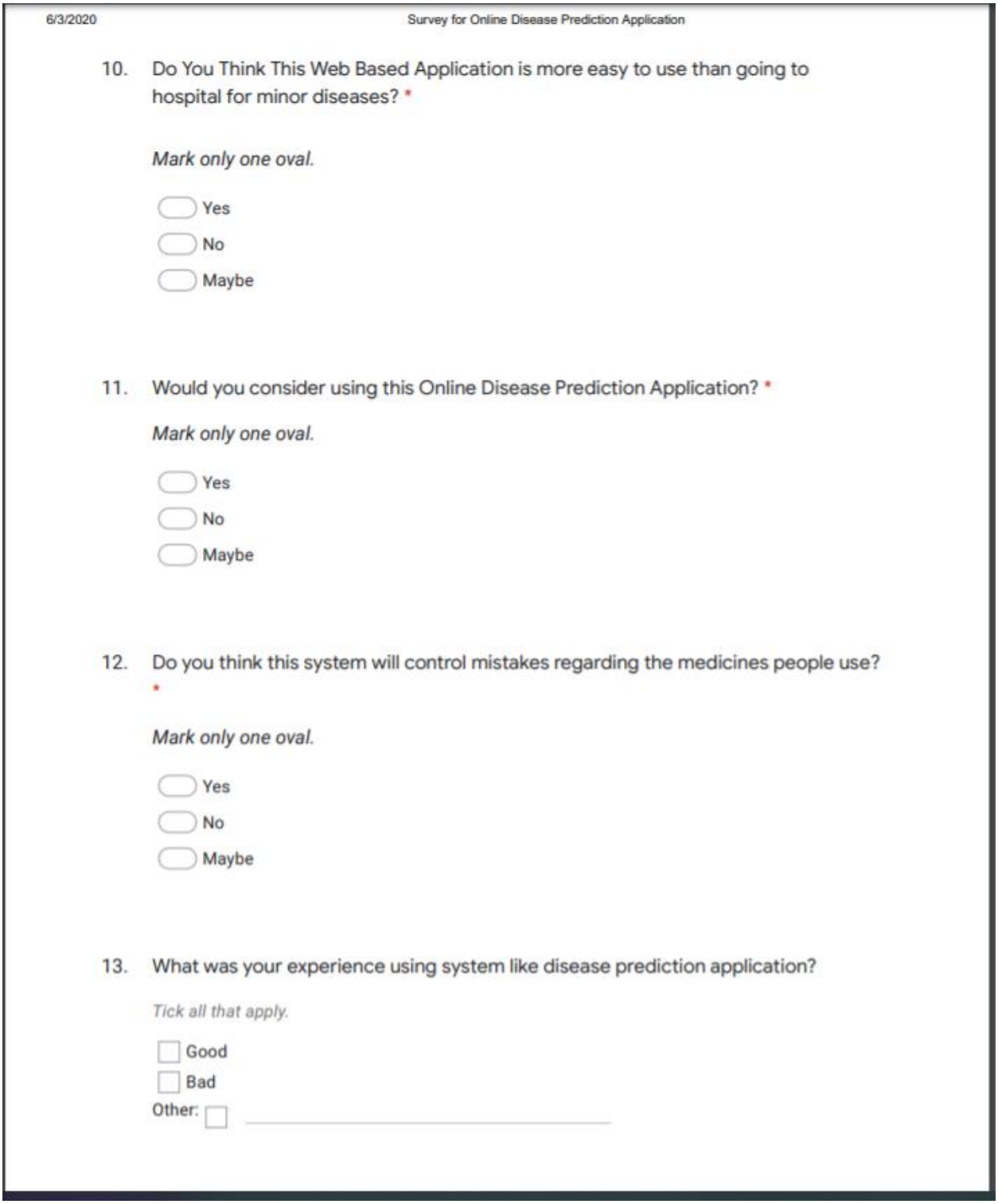



*Figure 16 Screenshot of pre survey form 3*

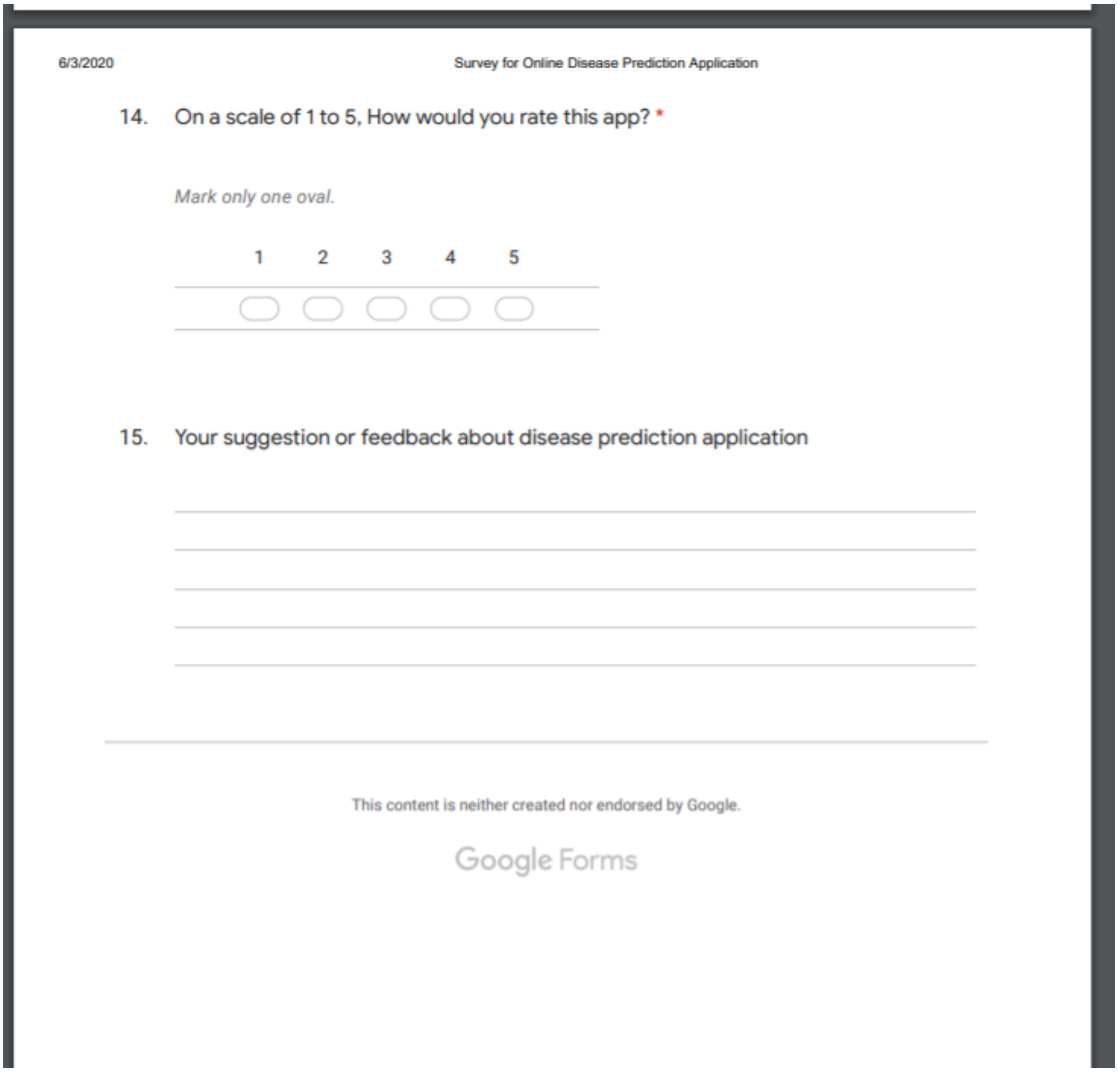



*Figure 17 Screenshot of pre survey form 4*

b) Pre-Survey Results:

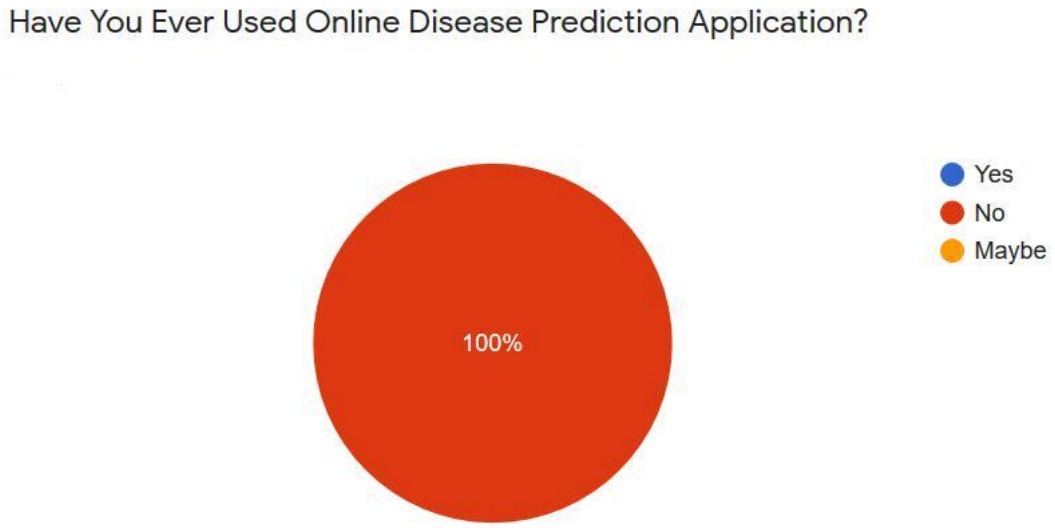



*Figure 18 Pre Survey Results(1)*

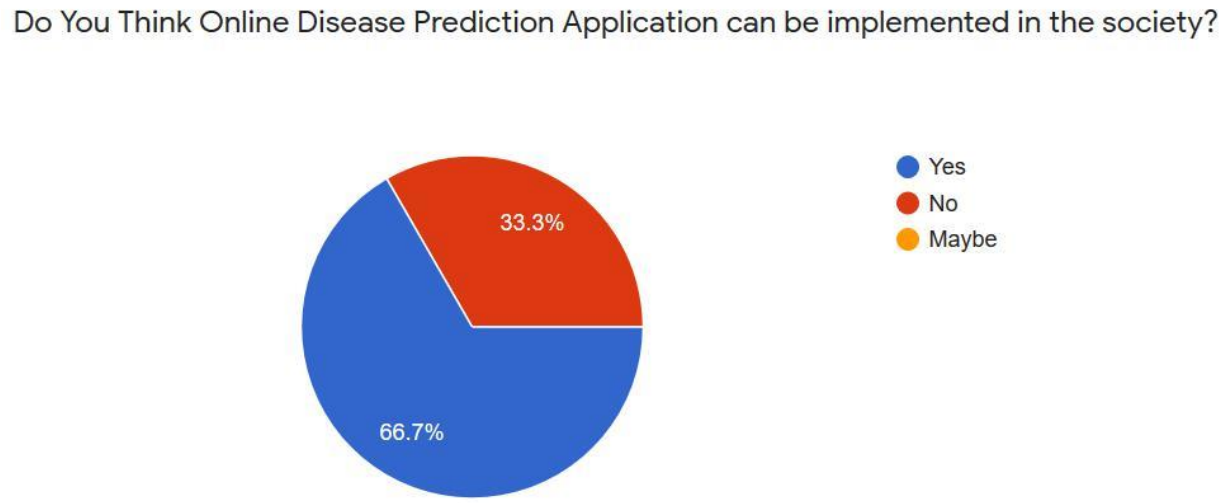



*Figure 19 Pre Survey Results(2)*

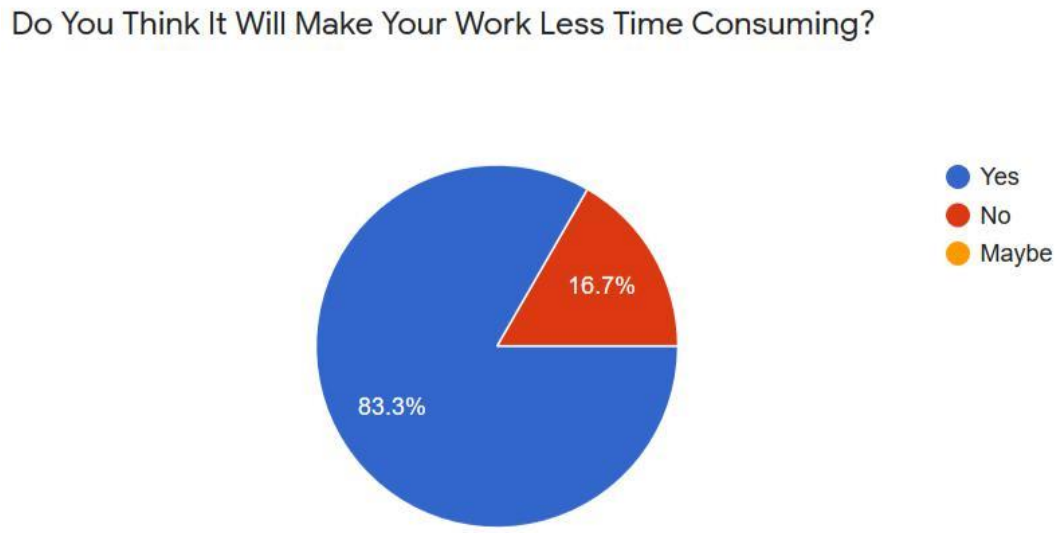



*Figure 20 Pre Survey Results(3)*

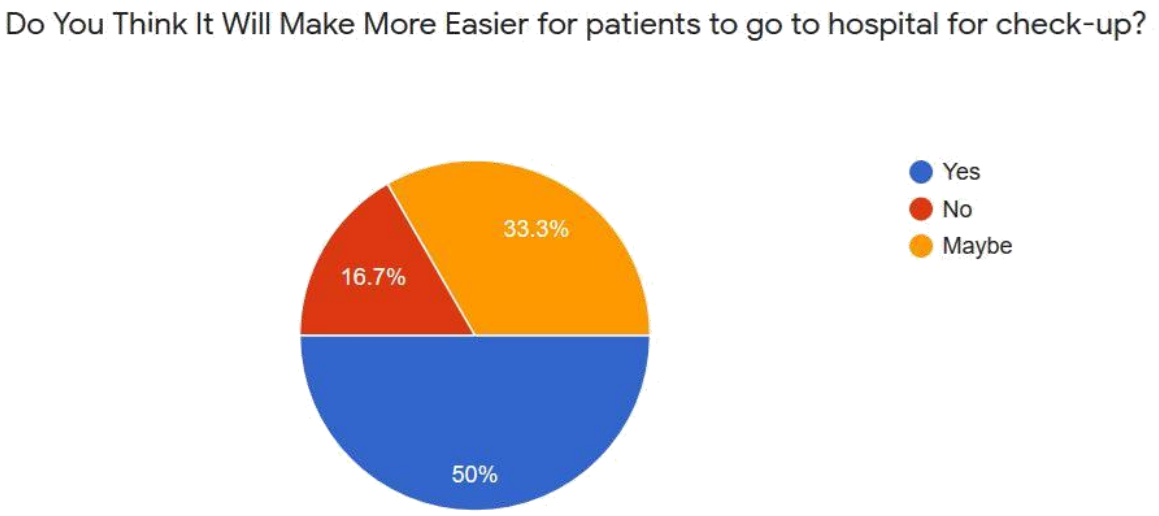



*Figure 21 Pre Survey Results(4)*

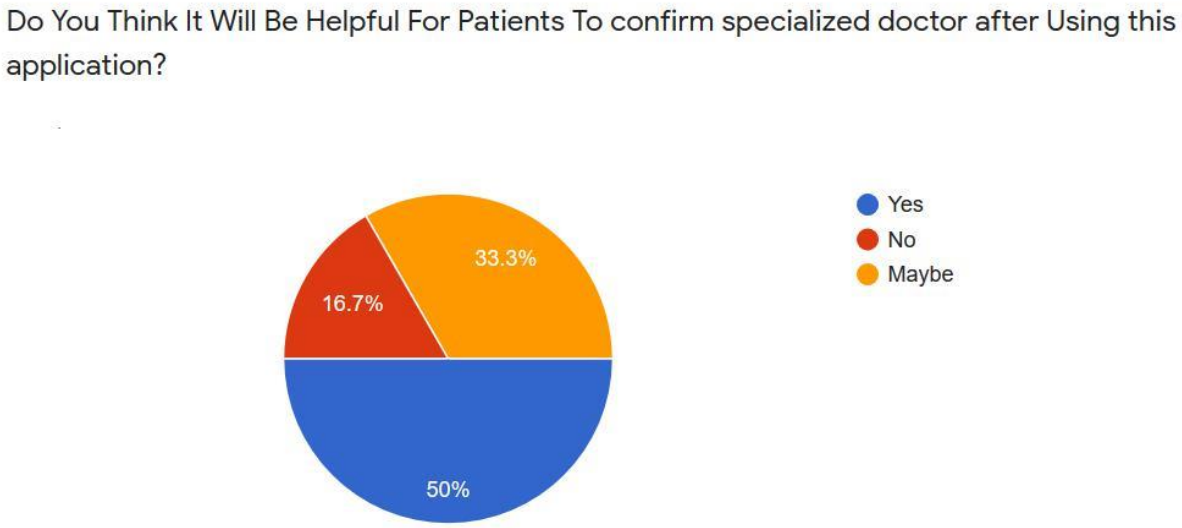



*Figure 22 Pre Survey Results(5)*

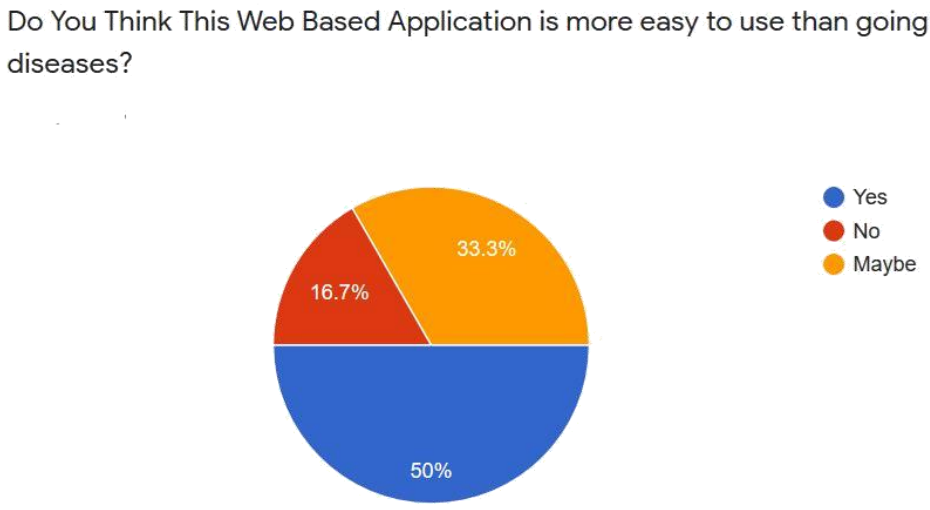



*Figure 23 Pre Survey Results(6)*

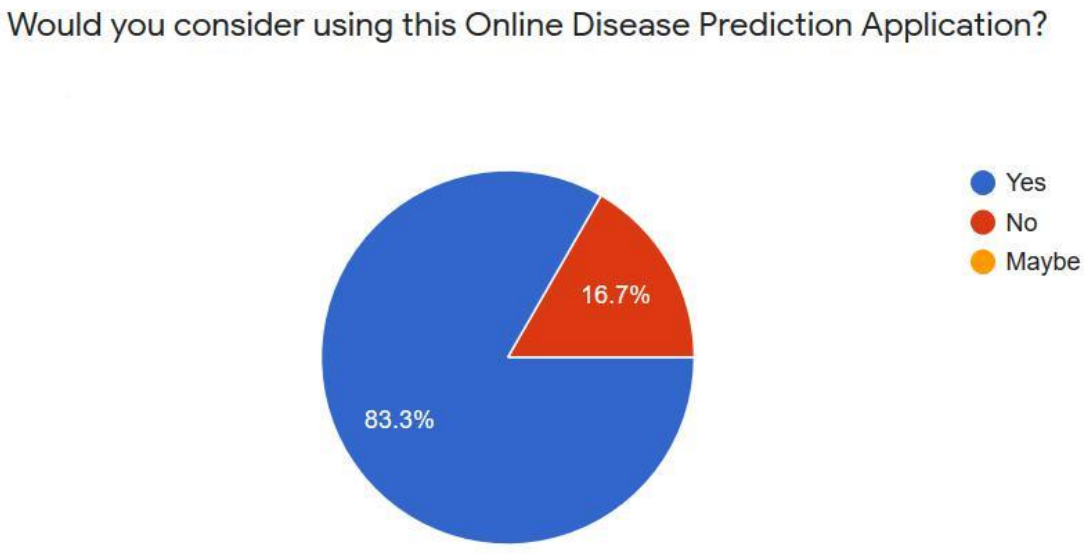



*Figure 24 Pre Survey Results(7)*

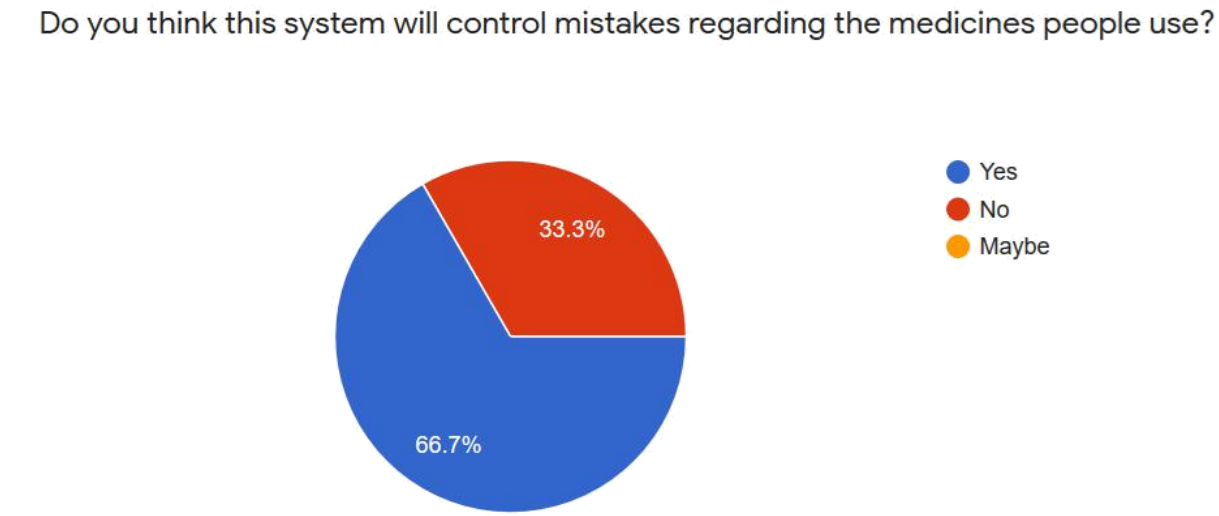



*Figure 25 Pre Survey Results(8)*

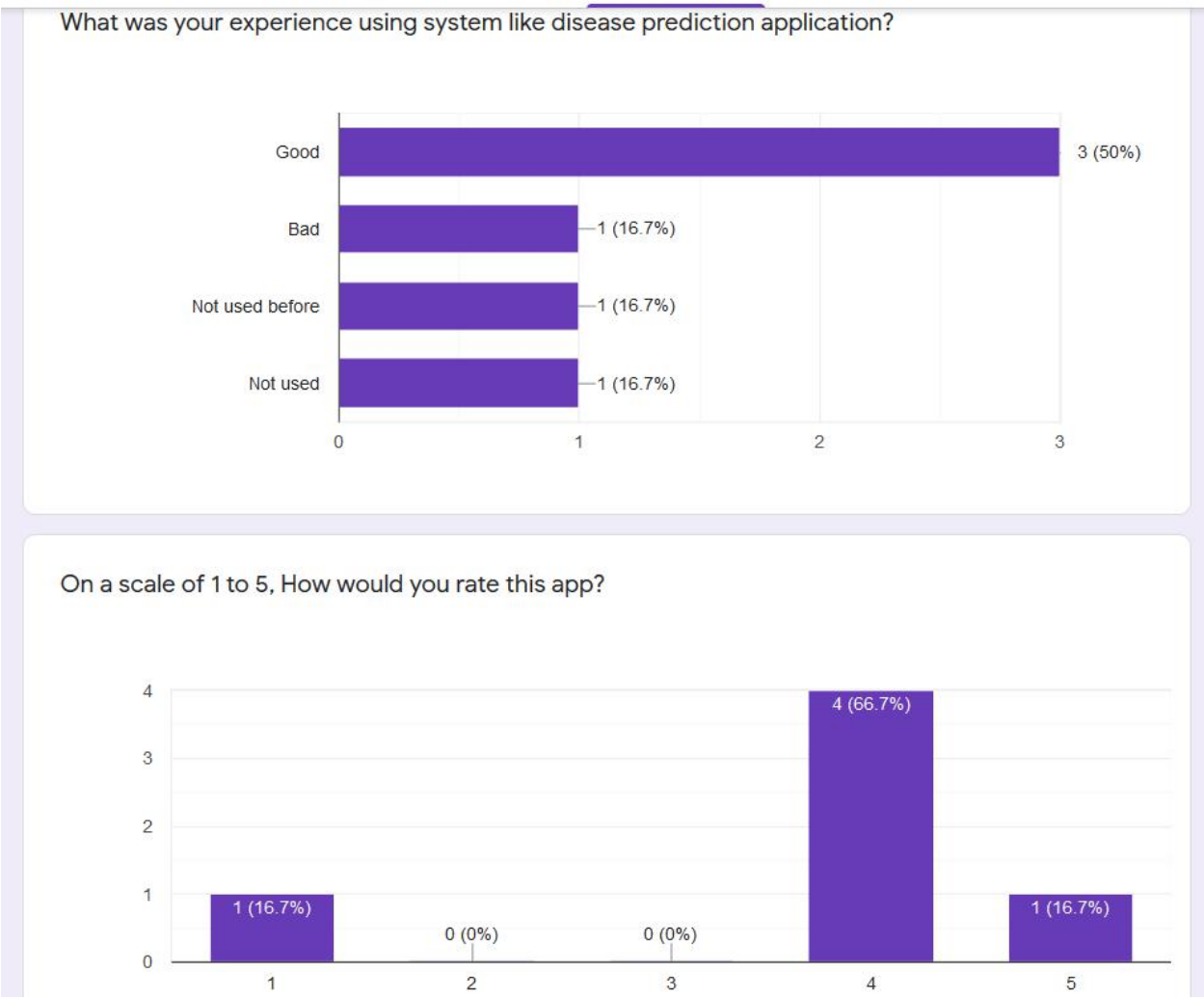



*Figure 26 Pre Survey Results(9)*

## 8.2) Appendix – B Post Survey

### a) Post Survey Form:



# Post Survey

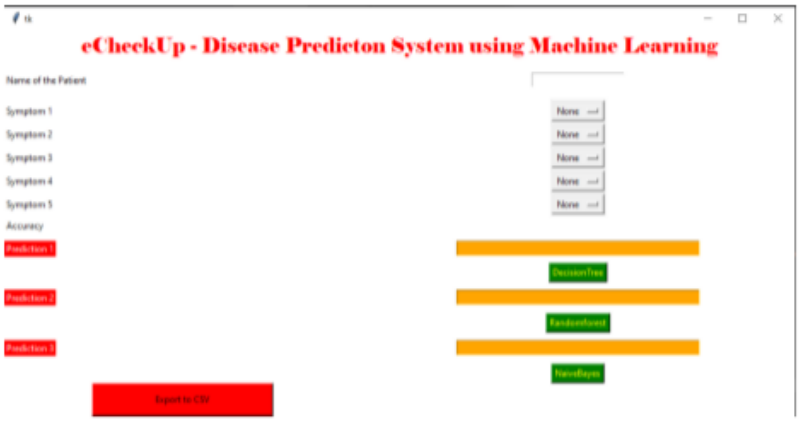
Note: This survey is for education purpose only and your answers and information will be kept confidential.

* Required

1. Email address *

2. Name *

Note: This survey is for education purpose only and your answers and information will be kept confidential.



3. How do you feel about the user interface? *

*Mark only one oval.*

- ◯ Excellent
- ◯ Good
- ◯ Fair
- ◯ Satisfying

4. Do you think the design is user-friendly? *

*Mark only one oval.*

- ◯ Yes
- ◯ No

5. What do you think is the best applicable area for this project? *

_____

6. How do you think you will use this echeckup in daily life? *

_____

_____

_____

_____

_____

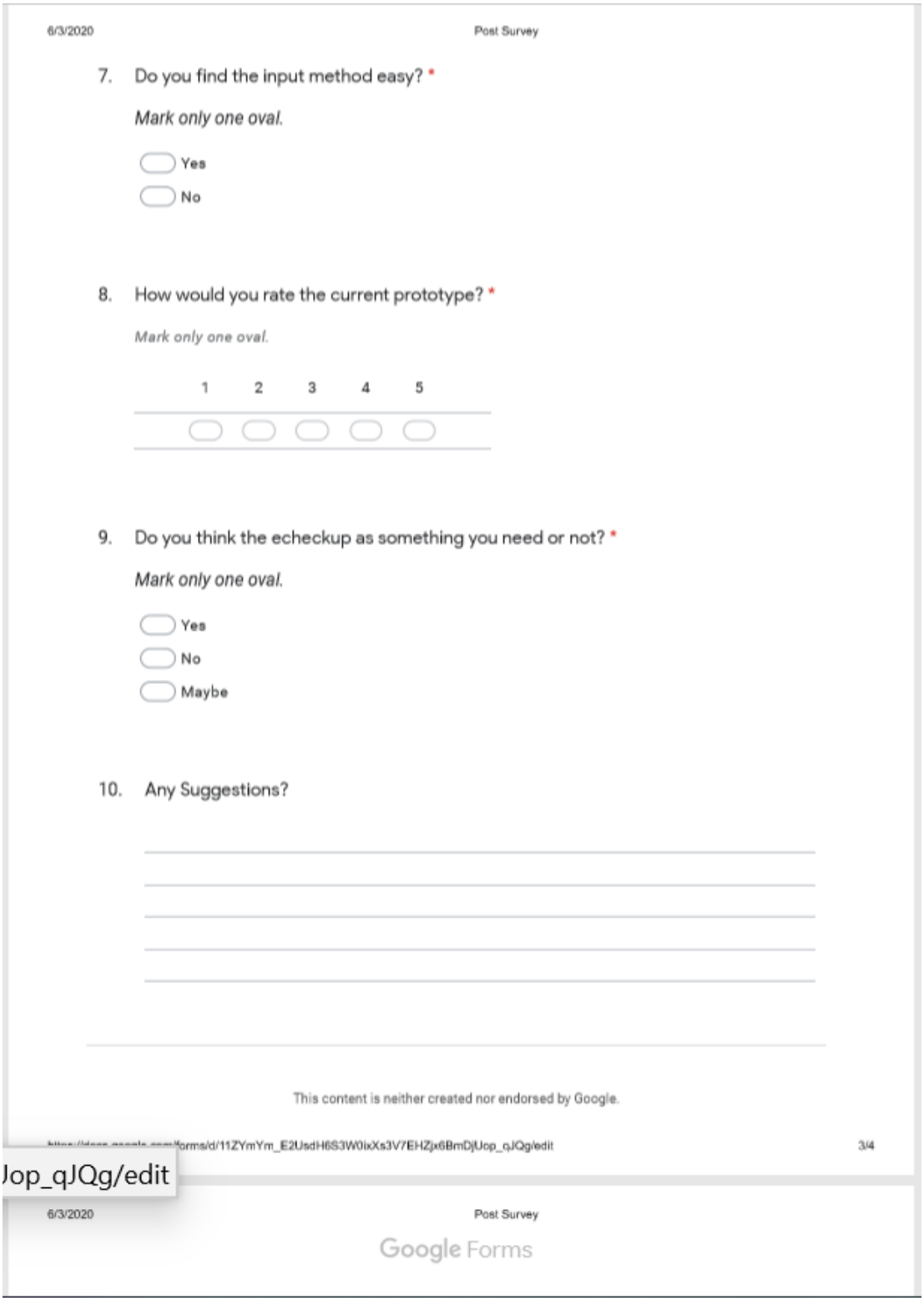
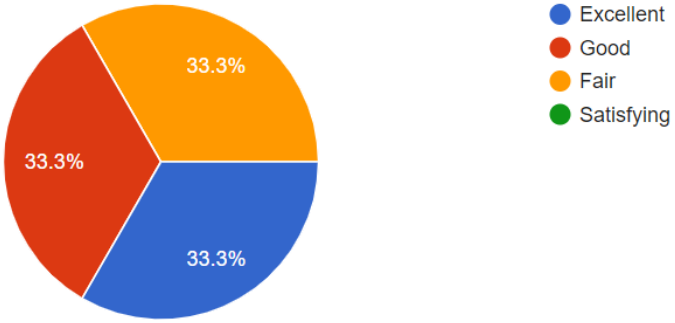



*Figure 27 Post Survey Form*

b) Post Survey Results

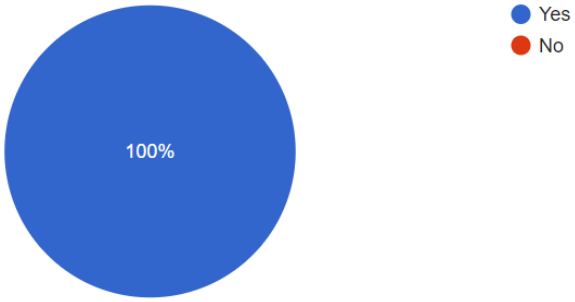
How do you feel about the user interface?

6 responses



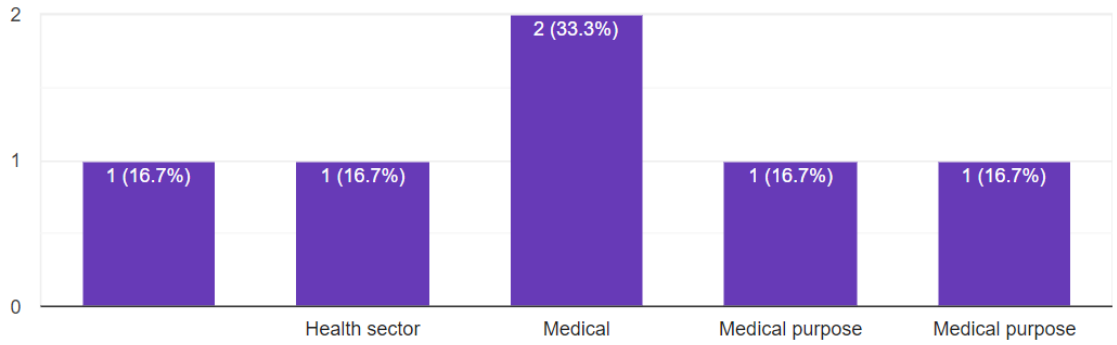

Do you think the design is user-friendly?

6 responses



What do you think is the best applicable area for this project?

6 responses




How do you think you will use this echeckup in daily life?

6 responses

Unpredictablw

Daily

Most likely.

.

For sugar testing and normal testing

Most likely

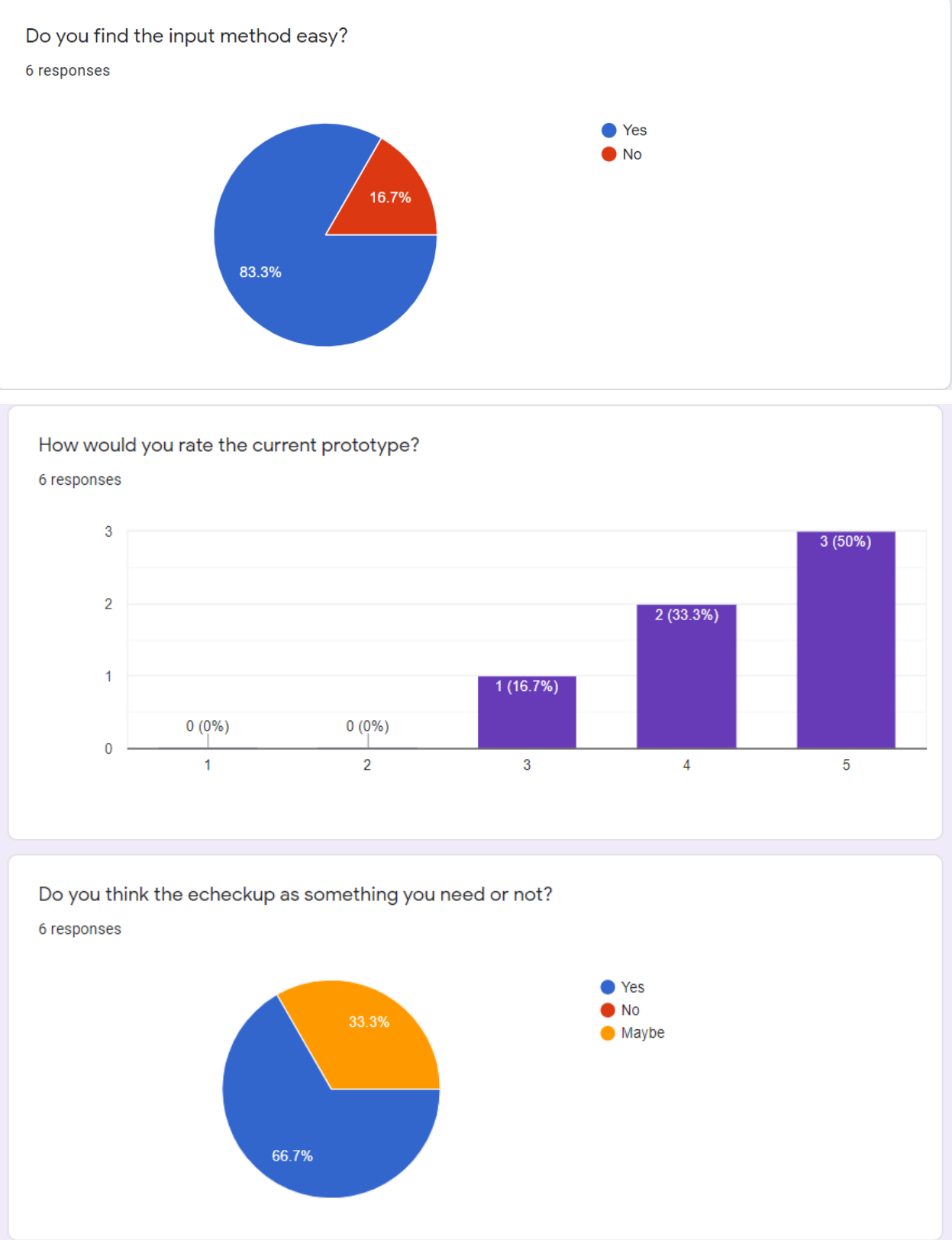



*Figure 28 Post Survey Results*

## 8.3) Appendix C: Sample Codes:

a) Sample code of the UI:

Code for the selection of the symptoms:

```
# from gui_stuff import *

l1 = ['back_pain', 'constipation', 'abdominal_pain', 'diarrhoea', 'mild_fever', '
yellow_urine',
      'yellowing_of_eyes', 'acute_liver_failure', 'fluid_overload', 'swelling_of_
stomach',
      'swelled_lymph_nodes', 'malaise', 'blurred_and_distorted_vision', 'phlegm',
 'throat_irritation',
      'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'congestion', 'chest_pai
n', 'weakness_in_limbs',
      'fast_heart_rate', 'pain_during_bowel_movements', 'pain_in_anal_region', 'b
loody_stool',
      'irritation_in_anus', 'neck_pain', 'dizziness', 'cramps', 'bruising', 'obes
ity', 'swollen_legs',
      'swollen_blood_vessels', 'puffy_face_and_eyes', 'enlarged_thyroid', 'brittl
e_nails',
      'swollen_extremeties', 'excessive_hunger', 'extra_marital_contacts', 'dryin
g_and_tingling_lips',
      'slurred_speech', 'knee_pain', 'hip_joint_pain', 'muscle_weakness', 'stiff_
neck', 'swelling_joints',
      'movement_stiffness', 'spinning_movements', 'loss_of_balance', 'unsteadines
s',
      'weakness_of_one_body_side', 'loss_of_smell', 'bladder_discomfort', 'foul_s
mell_of urine',
      'continuous_feel_of_urine', 'passage_of_gases', 'internal_itching', 'toxic_
look_(typhos)',
      'depression', 'irritability', 'muscle_pain', 'altered_sensorium', 'red_spot
s_over_body', 'belly_pain',
```

```
      'abnormal_menstruation', 'dischromic _patches', 'watering_from_eyes', 'incr
eased_appetite', 'polyuria', 'family_history', 'mucoid_sputum',
      'rusty_sputum', 'lack_of_concentration', 'visual_disturbances', 'receiving_
blood_transfusion',
      'receiving_unsterile_injections', 'coma', 'stomach_bleeding', 'distention_o
f_abdomen',
      'history_of_alcohol_consumption', 'fluid_overload', 'blood_in_sputum', 'pro
minent_veins_on_calf',
      'palpitations', 'painful_walking', 'pus_filled_pimples', 'blackheads', 'scu
rring', 'skin_peeling',
      'silver_like_dusting', 'small_dents_in_nails', 'inflammatory_nails', 'blist
er', 'red_sore_around_nose',
      'yellow_crust_ooze']

disease = ['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis', 'Drug Re
action',
           'Peptic ulcer diseae', 'AIDS', 'Diabetes', 'Gastroenteritis', 'Bronchi
al Asthma', 'Hypertension',
           ' Migraine', 'Cervical spondylosis',
           'Paralysis (brain hemorrhage)', 'Jaundice', 'Malaria', 'Chicken pox',
'Dengue', 'Typhoid', 'hepatitis A',
           'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E', 'Alcoholic
 hepatitis', 'Tuberculosis',
           'Common Cold', 'Pneumonia', 'Dimorphic hemmorhoids(piles)',
           'Heartattack', 'Varicoseveins', 'Hypothyroidism', 'Hyperthyroidism', '
Hypoglycemia', 'Osteoarthristis',
           'Arthritis', '(vertigo) Paroymsal  Positional Vertigo', 'Acne', 'Urina
ry tract infection', 'Psoriasis',
           'Impetigo']

l2 = []
```

Code for the buttons and labels used in the system:

```
# entry variables
Symptom1 = StringVar()
Symptom1.set(None)
Symptom2 = StringVar()
Symptom2.set(None)
Symptom3 = StringVar()
Symptom3.set(None)
Symptom4 = StringVar()
Symptom4.set(None)
Symptom5 = StringVar()
Symptom5.set(None)
Name = StringVar()

# Heading
w2 = Label(root, text="eCheckUp -
 Disease Predicton System using Machine Learning",
           fg="red", bg="white")
w2.config(font=("Elephant", 20))

w2.grid(row=1, column=0, columnspan=2, padx=100)

w2.grid(row=2, column=0, columnspan=2, padx=100)

# labels
NameLb = Label(root, text="Name of the Patient", fg="black", bg="white")
NameLb.grid(row=6, column=0, pady=15, sticky=W)


S1Lb = Label(root, text="Symptom 1", fg="black", bg="white")
S1Lb.grid(row=7, column=0, pady=2, sticky=W)

S2Lb = Label(root, text="Symptom 2", fg="black", bg="white")
S2Lb.grid(row=8, column=0, pady=2, sticky=W)

S3Lb = Label(root, text="Symptom 3", fg="black", bg="white")
```

```
S3Lb.grid(row=9, column=0, pady=2, sticky=W)

S4Lb = Label(root, text="Symptom 4", fg="black", bg="white")
S4Lb.grid(row=10, column=0, pady=2, sticky=W)

S5Lb = Label(root, text="Symptom 5", fg="black", bg="white")
S5Lb.grid(row=11, column=0, pady=2, sticky=W)

# accur = Label(root, text="Accuracy", fg="black",
#               bg="white", COMMAND=print(nacc))
# accur.grid(row=12, column=0, pady=2, sticky=W)


lrLb = Label(root, text="Prediction 1", fg="white", bg="red")
lrLb.grid(row=14, column=0, pady=8, sticky=W)


destreeLb = Label(root, text="Prediction 2", fg="white", bg="red")
destreeLb.grid(row=16, column=0, pady=10, sticky=W)

ranfLb = Label(root, text="Prediction 3", fg="white", bg="red")
ranfLb.grid(row=18, column=0, pady=10, sticky=W)

# entries
OPTIONS = sorted(l1)

NameEn = Entry(root, textvariable=Name)
NameEn.grid(row=6, column=1)

S1En = OptionMenu(root, Symptom1, *OPTIONS)
S1En.grid(row=7, column=1)

S2En = OptionMenu(root, Symptom2, *OPTIONS)
S2En.grid(row=8, column=1)

S3En = OptionMenu(root, Symptom3, *OPTIONS)
```

```
S3En.grid(row=9, column=1)

S4En = OptionMenu(root, Symptom4, *OPTIONS)
S4En.grid(row=10, column=1)

S5En = OptionMenu(root, Symptom5, *OPTIONS)
S5En.grid(row=11, column=1)


dst = Button(root, text="DecisionTree",
             command=DecisionTree, bg="green", fg="yellow")
dst.grid(row=15, column=1, padx=10)

rnf = Button(root, text="Randomforest",
             command=randomforest, bg="green", fg="yellow")
rnf.grid(row=17, column=1, padx=10)

lr = Button(root, text="NaiveBayes",
            command=NaiveBayes, bg="green", fg="yellow")
lr.grid(row=19, column=1, padx=10)

# textfileds
t1 = Text(root, height=1, width=40, bg="orange", fg="black")
t1.grid(row=14, column=1, padx=10)

t2 = Text(root, height=1, width=40, bg="orange", fg="black")
t2.grid(row=16, column=1, padx=10)

t3 = Text(root, height=1, width=40, bg="orange", fg="black")
t3.grid(row=18, column=1, padx=10)

t3 = Text(root, height=1, width=40, bg="orange",
          fg="black", command=print(dacc))
t3.grid(row=18, column=1, padx=10)
```

```
# calling export to CSV function
# sv = Button(root, text="Export to CSV", bg="red", fg="black", padx=80, pady=10)
# sv.grid(row=21, column=0)
```

b) Sample code For the prediction:

Code for Testing and training:

```
# TESTING DATA df -------------------------------------------------------------
----------------------
df = pd.read_csv("Training.csv")

df.replace({'prognosis': {'Fungal infection': 0, 'Allergy': 1, 'GERD': 2, 'Chroni
c cholestasis': 3, 'Drug Reaction': 4,
                          'Peptic ulcer diseae': 5, 'AIDS': 6, 'Diabetes ': 7, 'G
astroenteritis': 8, 'Bronchial Asthma': 9, 'Hypertension ': 10,
                          'Migraine': 11, 'Cervical spondylosis': 12,
                          'Paralysis (brain hemorrhage)': 13, 'Jaundice': 14, 'Ma
laria': 15, 'Chicken pox': 16, 'Dengue': 17, 'Typhoid': 18, 'hepatitis A': 19,
                          'Hepatitis B': 20, 'Hepatitis C': 21, 'Hepatitis D': 22
, 'Hepatitis E': 23, 'Alcoholic hepatitis': 24, 'Tuberculosis': 25,
                          'Common Cold': 26, 'Pneumonia': 27, 'Dimorphic hemmorho
ids(piles)': 28, 'Heart attack': 29, 'Varicose veins': 30, 'Hypothyroidism': 31,
                          'Hyperthyroidism': 32, 'Hypoglycemia': 33, 'Osteoarthri
stis': 34, 'Arthritis': 35,
                          '(vertigo) Paroymsal  Positional Vertigo': 36, 'Acne':
37, 'Urinary tract infection': 38, 'Psoriasis': 39,
                          'Impetigo': 40}}, inplace=True)

# print(df.head())
nacc = dacc = racc = 0
X = df[11]

y = df[["prognosis"]]
np.ravel(y)
# print(y)
```

```
# TRAINING DATA tr ----------------------------------------------------------
-----------------
tr = pd.read_csv("Testing.csv")
tr.replace({'prognosis': {'Fungal infection': 0, 'Allergy': 1, 'GERD': 2, 'Chroni
c cholestasis': 3, 'Drug Reaction': 4,
                          'Peptic ulcer diseae': 5, 'AIDS': 6, 'Diabetes ': 7, 'G
astroenteritis': 8, 'Bronchial Asthma': 9, 'Hypertension ': 10,
                          'Migraine': 11, 'Cervical spondylosis': 12,
                          'Paralysis (brain hemorrhage)': 13, 'Jaundice': 14, 'Ma
laria': 15, 'Chicken pox': 16, 'Dengue': 17, 'Typhoid': 18, 'hepatitis A': 19,
                          'Hepatitis B': 20, 'Hepatitis C': 21, 'Hepatitis D': 22
, 'Hepatitis E': 23, 'Alcoholic hepatitis': 24, 'Tuberculosis': 25,
                          'Common Cold': 26, 'Pneumonia': 27, 'Dimorphic hemmorho
ids(piles)': 28, 'Heart attack': 29, 'Varicose veins': 30, 'Hypothyroidism': 31,
                          'Hyperthyroidism': 32, 'Hypoglycemia': 33, 'Osteoarthri
stis': 34, 'Arthritis': 35,
                          '(vertigo) Paroymsal  Positional Vertigo': 36, 'Acne':
37, 'Urinary tract infection': 38, 'Psoriasis': 39,
                          'Impetigo': 40}}, inplace=True)

X_test = tr[l1]
y_test = tr[["prognosis"]]
np.ravel(y_test)
```

Code for Prediction:

```
def DecisionTree():
    dacc = 0
    from sklearn import tree

    clf3 = tree.DecisionTreeClassifier()   # empty model of the decision tree
    clf3 = clf3.fit(X, y)

    # calculating accuracy-------------------------------------------------------
------------
    from sklearn.metrics import accuracy_score
```

```
    y_pred = clf3.predict(X_test)
    dacc = accuracy_score(y_test, y_pred, normalize=False)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred, normalize=False))
    # ----------------------------------------------------
    name = Name.get()
    psymptoms = [Symptom1.get(), Symptom2.get(), Symptom3.get(),
                 Symptom4.get(), Symptom5.get()]

    for k in range(0, len(l1)):
        print(k)
        for z in psymptoms:
            if(z == l1[k]):
                l2[k] = 1

    inputtest = [l2]
    predict = clf3.predict(inputtest)
    predicted = predict[0]

    h = 'no'
    for a in range(0, len(disease)):
        if(predicted == a):
            h = 'yes'
            break
    if (h == 'yes'):
        t1.delete("1.0", END)
        t1.insert(END, disease[a])
        finalpredict = disease[a]
    else:
        t1.delete("1.0", END)
        t1.insert(END, "Not Found")
    # writing to csv
    data_file = open("data.csv", 'a')
    data_file.write("\n Name of Patient: "+name+"\n Decision Tree Prediction \n"+
"Symptoms \t "+str(psymptoms) +
```

```
                    "\n Prediction: "+finalpredict)


def randomforest():
    racc = 0
    from sklearn.ensemble import RandomForestClassifier
    clf4 = RandomForestClassifier()
    clf4 = clf4.fit(X, np.ravel(y))

    # calculating accuracy-------------------------------------------------------
------------
    from sklearn.metrics import accuracy_score
    y_pred = clf4.predict(X_test)
    racc = accuracy_score(y_test, y_pred, normalize=False)

    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred, normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(), Symptom2.get(), Symptom3.get(),
                 Symptom4.get(), Symptom5.get()]

    for k in range(0, len(l1)):
        for z in psymptoms:
            if(z == l1[k]):
                l2[k] = 1

    inputtest = [l2]
    predict = clf4.predict(inputtest)
    predicted = predict[0]

    h = 'no'
    for a in range(0, len(disease)):
        if(predicted == a):
            h = 'yes'
```

```
            break

    if (h == 'yes'):
        t2.delete("1.0", END)
        t2.insert(END, disease[a])
        finalpredict = disease[a]
    else:
        t2.delete("1.0", END)
        t2.insert(END, "Not Found")

    # writing to csv
    data_file = open("data.csv", 'a')
    data_file.write("\n Random Forest Prediction \n" +
                    finalpredict)


def NaiveBayes():
    from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
    gnb = gnb.fit(X, np.ravel(y))

    # calculating accuracy-------------------------------------------------------
------------
    from sklearn.metrics import accuracy_score
    y_pred = gnb.predict(X_test)
    nacc = accuracy_score(y_test, y_pred)

    print(nacc)
    print(accuracy_score(y_test, y_pred, normalize=False))
    # -----------------------------------------------------

    psymptoms = [Symptom1.get(), Symptom2.get(), Symptom3.get(),
                 Symptom4.get(), Symptom5.get()]
    for k in range(0, len(l1)):
        for z in psymptoms:
```

```
            if(z == l1[k]):
                l2[k] = 1

    inputtest = [l2]
    predict = gnb.predict(inputtest)
    predicted = predict[0]

    h = 'no'
    for a in range(0, len(disease)):
        if(predicted == a):
            h = 'yes'
            break

    if (h == 'yes'):
        t3.delete("1.0", END)
        t3.insert(END, disease[a])
        finalpredict = disease[a]
    else:
        t3.delete("1.0", END)
        t3.insert(END, "Not Found")
    # writing to csv
    data_file = open("data.csv", 'a')
    data_file.write("\n Naive Bayes Prediction \t"+finalpredict)


# def text2csv(screenname):
#     t = ""
#     screenname = str(screenname)
#     data_file = open("data.csv", 'a')
#     for s in screenname:
#         if(s != ',' and s != '\n'):
#             t = t+s
#     data_file.write(t)
```

Function for writing to csv:

```
data_file = open("data.csv", 'a')
    data_file.write("\n Name of Patient: "+name+"\n Decision Tree Prediction \n"+
"Symptoms \t "+str(psymptoms) +
                    "\n Prediction: "+finalpredict)
```

```
data_file = open("data.csv", 'a')
    data_file.write("\n Random Forest Prediction \n" +
                    finalpredict)
```

```
data_file = open("data.csv", 'a')
    data_file.write("\n Naive Bayes Prediction \t"+finalpredict)
```

## 8.4) Appendix D: Designs

Task Breakdown:

| | | | | |
|---|---|---|---|---|
| Backgroung Reading | T | 19/09/2019 | 01/10/2019 | 9 days |
| Conceptual Design | T | 02/10/2019 | 16/10/2019 | 11 days |
| Technical Design | T | 17/10/2019 | 11/11/2019 | 18 days |
| Prototype | T | 12/11/2019 | 18/12/2019 | 27 days |
| Complete Prototype | T | 19/12/2019 | 30/12/2019 | 8 days |
| Complete Interim Report | T | 31/12/2019 | 09/01/2020 | 8 days |
| Analysis & Design | T | 10/01/2020 | 10/02/2020 | 22 days |
| Start Coding | T | 11/02/2020 | 23/03/2020 | 30 days |
| Check Coding | T | 23/03/2020 | 01/04/2020 | 8 days |
| Start Report Writing | T | 01/04/2020 | 30/04/2020 | 22 days |
| Check, Test and Finalize | T | 01/05/2020 | 12/05/2020 | 8 days |
| Final Check | T | 13/05/2020 | 26/05/2020 | 10 days |

*Figure 29 Task Breakdown*

Gantt Chart

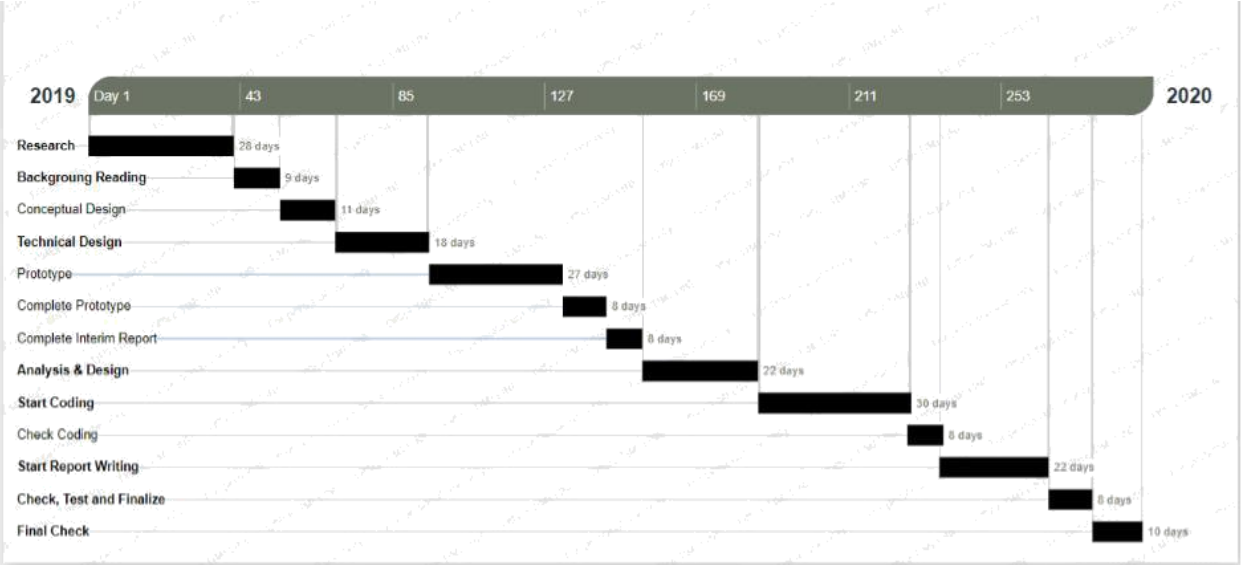



*Figure 30 Gantt Chart*

Work-flow Diagram:

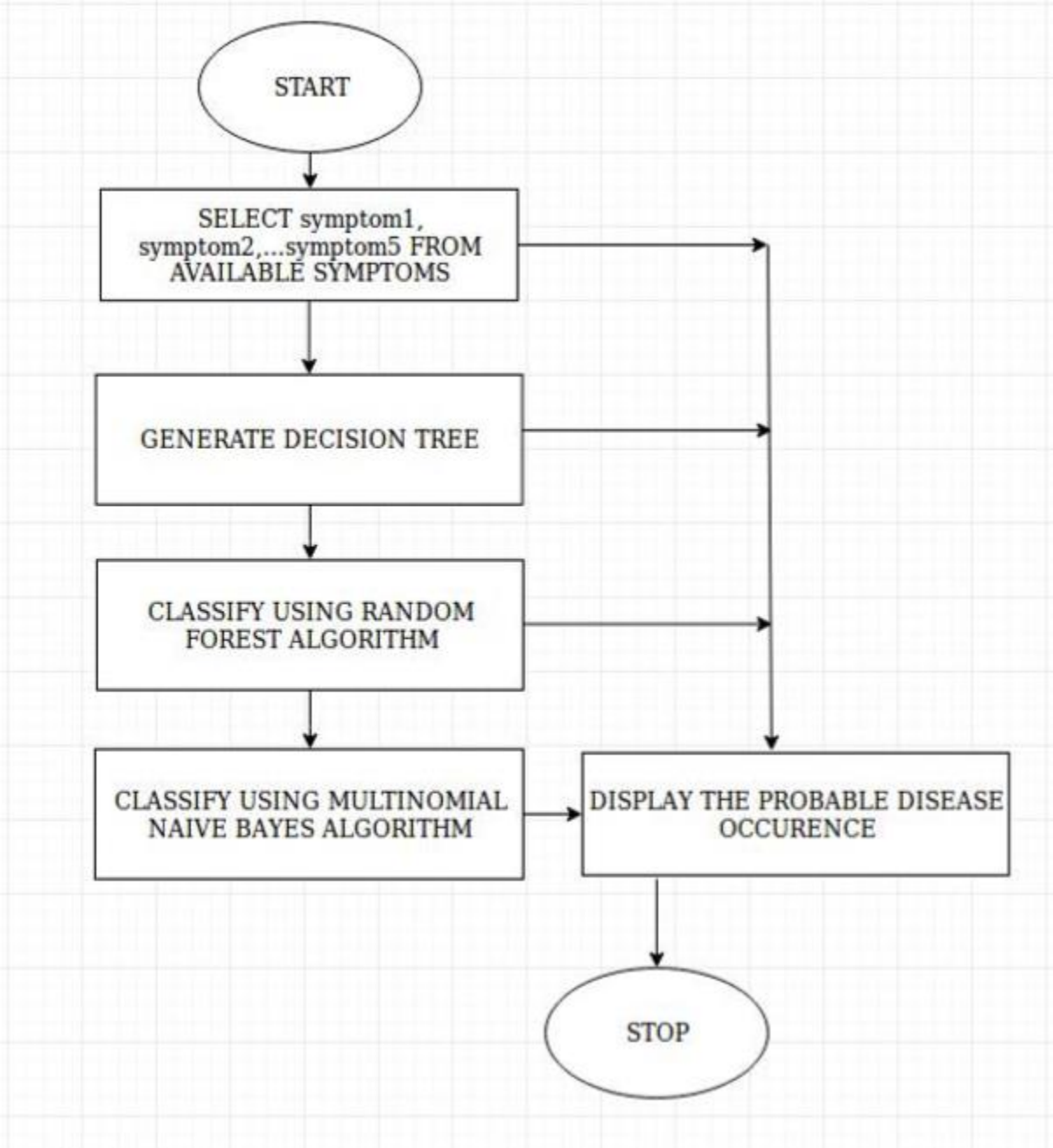



*Figure 31 Work Flow Diagram*

Use Case Diagram:

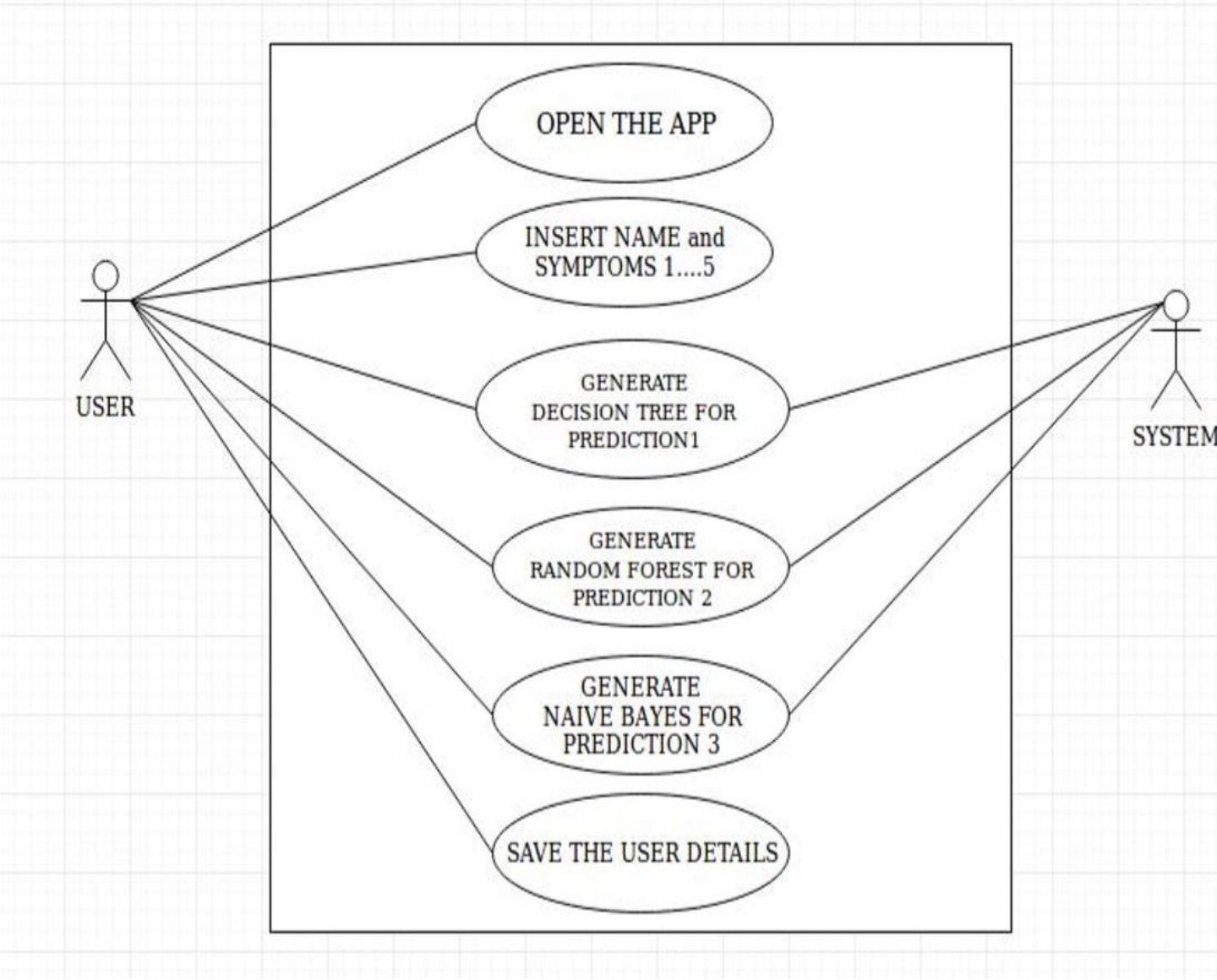



*Figure 32 Use Case Diagram*

## 8.5) Appendix E: Screenshots of the system:

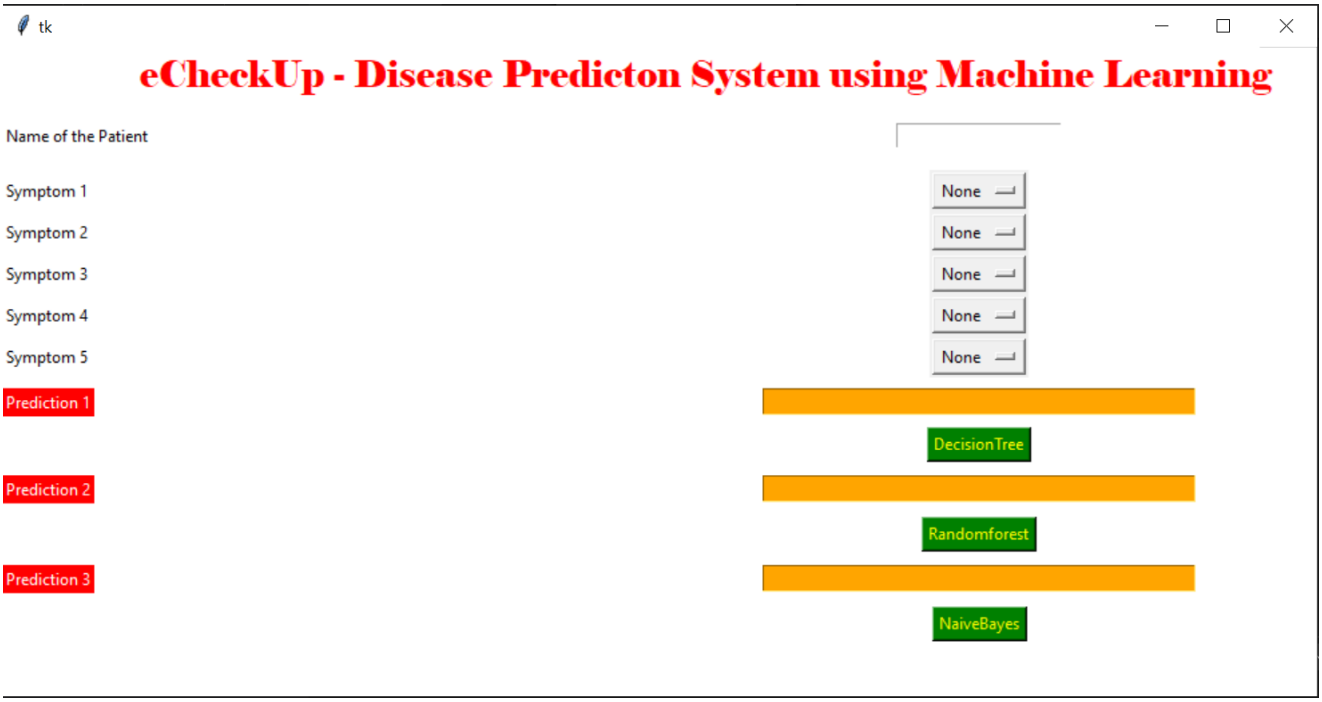



*Figure 33 Screenshot of system*

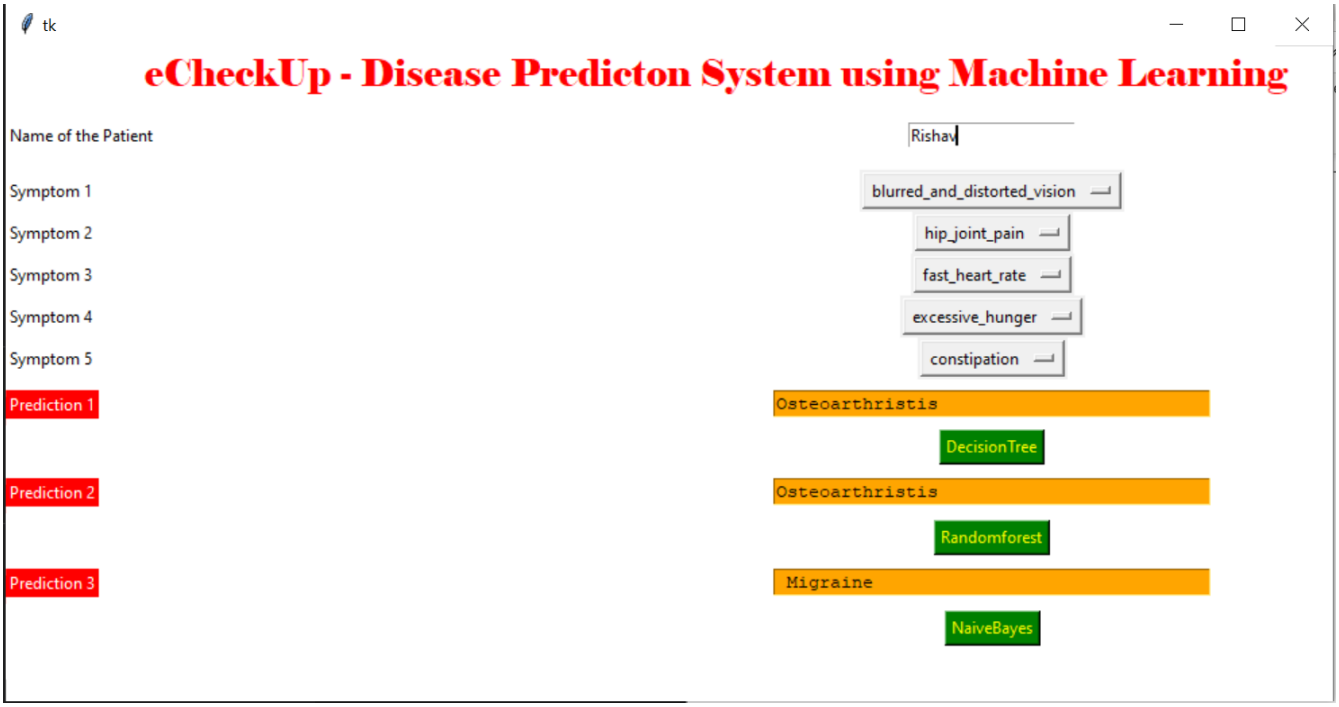


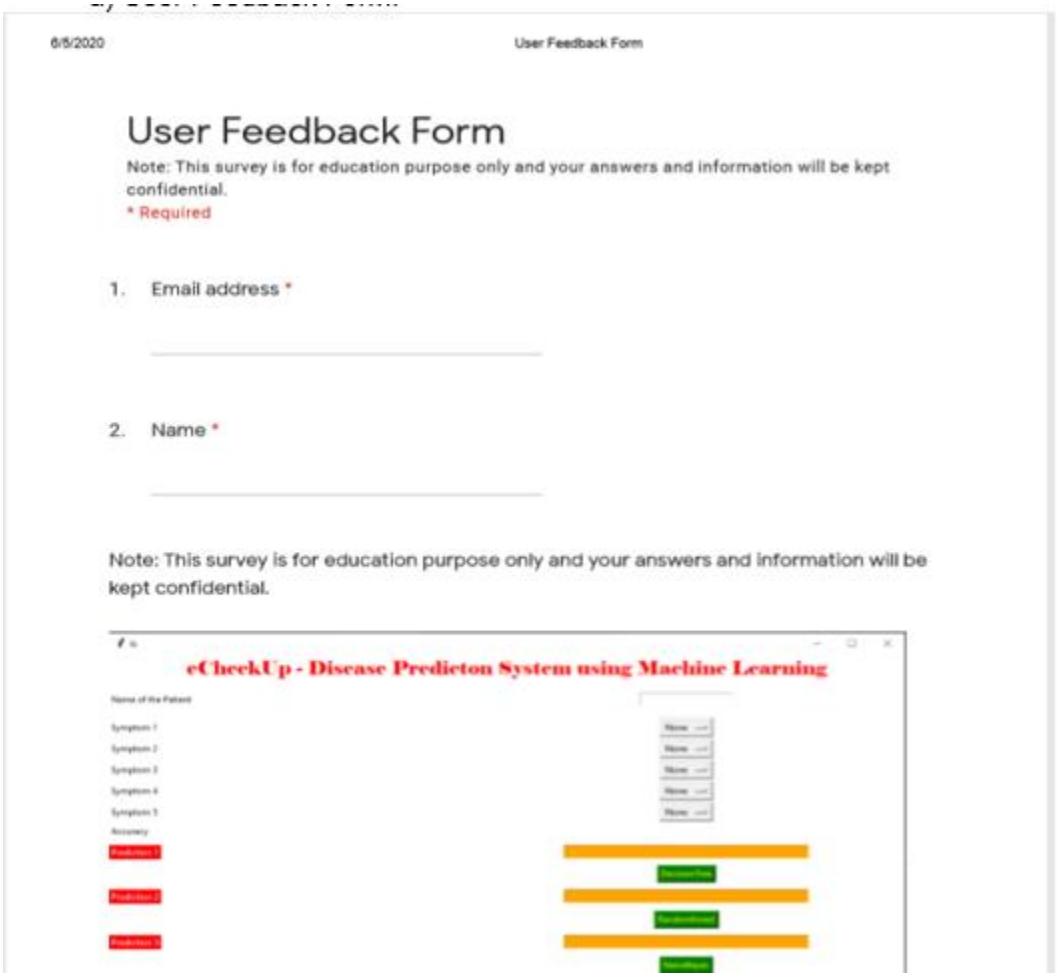

*Figure 34 Screenshot of system with symptoms*

```
Rishav@DESKTOP-LQA2BEP MINGW64 ~/Desktop/echeckup (master)
$  env C:\\Users\\Rishav\\AppData\\Local\\Programs\\Python\\Python38-32\\python.exe c:\\Users\\Rishav\\.vscode\\extensions\\ms-python.python-2020.
5.80290\\pythonFiles\\lib\\python\\debugpy\\no_wheels\\debugpy\\launcher 62959 -- c:\\Users\\Rishav\\Desktop\\echeckup\\clean_code.py
0
```

*Figure 35 Process to launch system*

## 8.6) Appendix F: User Feedback:

### a) User Feedback Form:



6/5/2020 User Feedback Form

# User Feedback Form

Note: This survey is for education purpose only and your answers and information will be kept confidential.

* Required

1. Email address *

2. Name *

Note: This survey is for education purpose only and your answers and information will be kept confidential.

3. How do you feel about the user interface? *

*Mark only one oval.*

- ◯ Excellent
- ◯ Good
- ◯ Fair
- ◯ Satisfying

4. Was the prediction right as per your view? *

*Mark only one oval.*

- ◯ Yes
- ◯ No
- ◯ Maybe

5. What do you think is this project can change the medical system in Nepal? *

*Mark only one oval.*

- ◯ yes
- ◯ No
- ◯ Maybe

6. How do you think you will use this echeckup in daily life? *

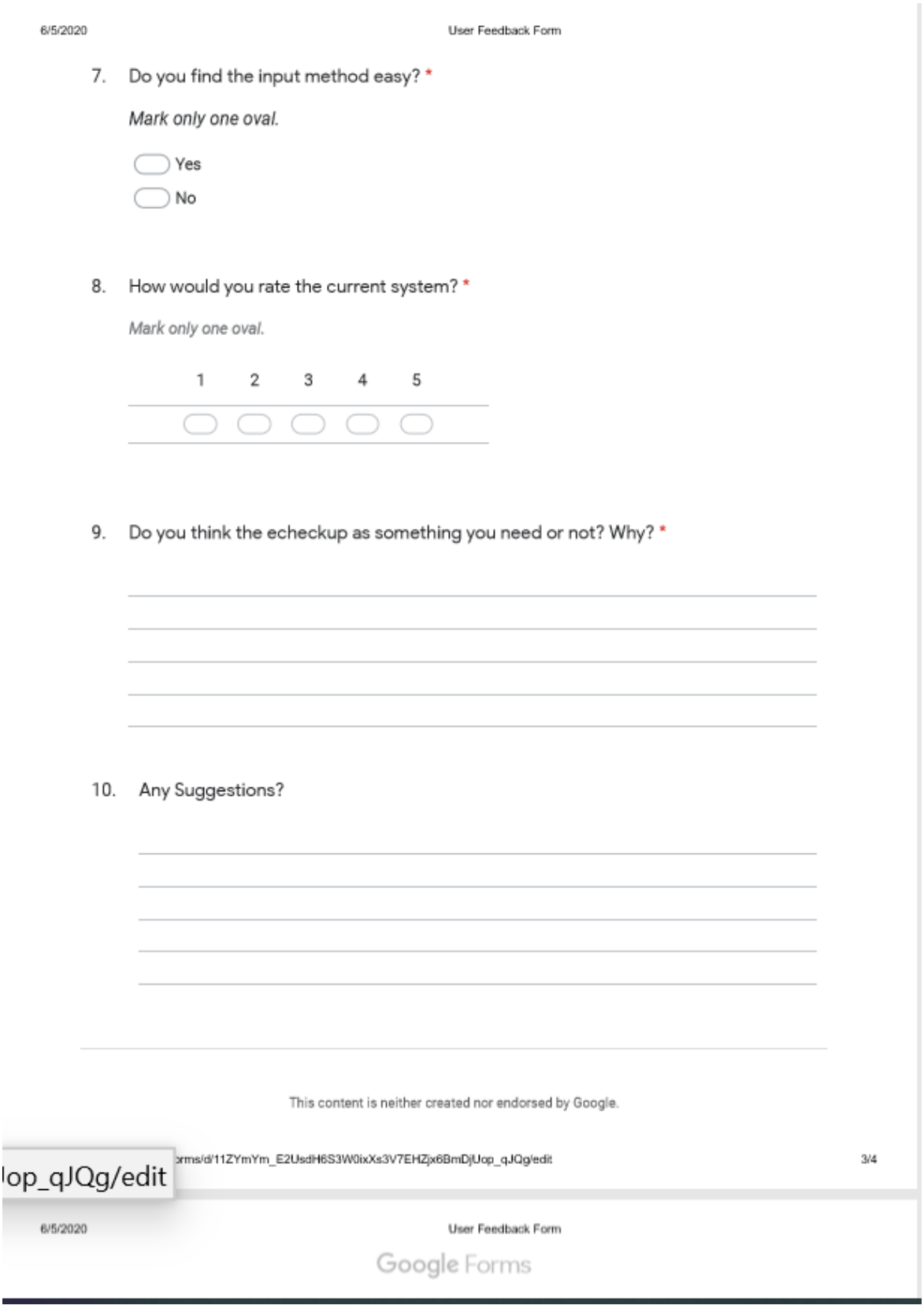



*Figure 36 User feedback form*

b) Sample of filled User feedback Form:

# User Feedback Form

Note: This survey is for education purpose only and your answers and information will be kept confidential.

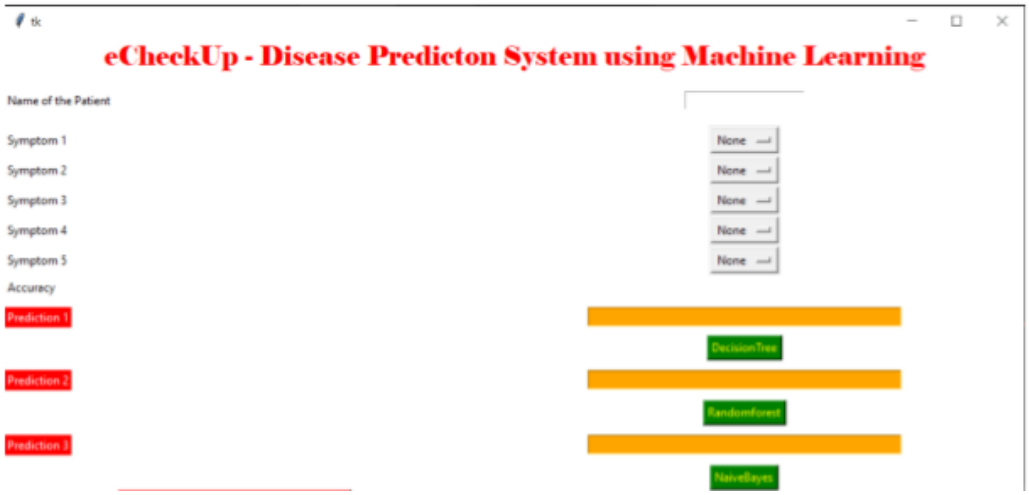
Email address *

shambhavisharma@gmail.com

Name *

Shambhavi Sharma

Note: This survey is for education purpose only and your answers and information will be kept confidential.



How do you feel about the user interface? *

- ○ Excellent
- ○ Good
- ◉ Fair
- ○ Satisfying

Was the prediction right as per your view? *

- ○ Yes
- ○ No
- ◉ Maybe

What do you think is this project can change the medical system in Nepal? *

- ○ yes
- ○ No
- ◉ Maybe

How do you think you will use this echeckup in daily life? *

If the prediction is accurate then, I may use it to check if the seen symptoms are leading to a serious issue or not.

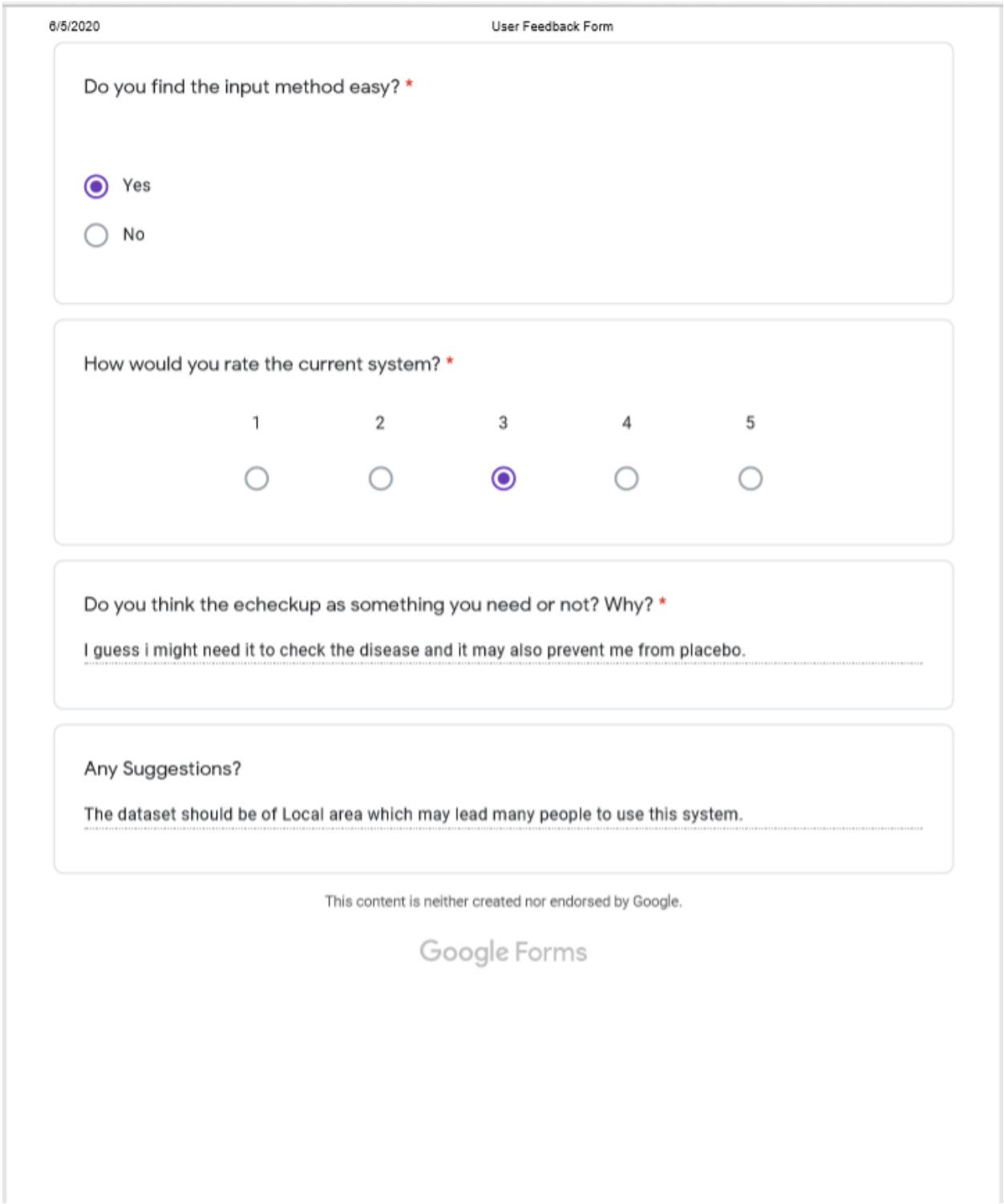



*Figure 37 User Feedback Result*

## 8.7) Appendix G: Future Work

a) Readings for future:

The system developed is ready to go but there may be many things that I could integrate. I am planning to make this project more efficient. I will deploy it with proper database and give users more features to interact. I am going to learn more about the ML and data mining. I will make this system more accurate by using many methods.

## References


https://www.annualreviews.org/doi/full/10.1146/annurev.genom.9.081307.164319#_i26
https://pdfs.semanticscholar.org/8610/4041dd610f78b6a5718e7ebe3cf6affb6f28.pdf
https://www.researchgate.net/publication/325973417_Improving_Disease_Prediction_by_Machine_Learning
https://theconversation.com/artificial-intelligence-in-medicine-raises-legal-and-ethical-concerns-122504
http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html
https://www.academia.edu/36086763/An_Algorithm_for_Predictive_Data_Mining_Approach_in_Medical_Diagnosis
https://www.academia.edu/36410403/Medicine_Prescription_System_by_using_Machine_Learning
https://www.academia.edu/16408680/Predicting_Disease_Onset_from_Mutation_Status_Using_Proband_and_Relative_Data_with_Applications_to_Huntington_s_Disease

# Bibliography

Genetics and Public Policy Center, J. H. U. W., 2009. The Current Landscape for Direct-to-Consumer Genetic Testing: Legal, Ethical, and Policy Issues. *Genetc Testing .*
J.Vijayashree, 2020. *semanticscholar.* [Online]
Available at:
https://pdfs.semanticscholar.org/8610/4041dd610f78b6a5718e7ebe3cf6affb6f28.pdf
[Accessed 2020].

Brownlee, J. (2019) *machinelearningmastery* [Online]. Available from: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/ [Accessed 2020].

Klinger, C. (n.d.) *Iterative Software Development Approact* [Online]. Available from: https://wiki.nci.nih.gov/display/CommonProjects/Iterative+Software+Development+Approach [Accessed 05 April 2017].

lucidchart. (2019) *lucidchart* [Online]. Available from: https://www.lucidchart.com/pages/decision-tree [Accessed 2020].

projectmanager. (2020) *projectmanager* [Online]. Available from: https://www.projectmanager.com/blog/scrum-methodology [Accessed 2020].

Solutions, B. (n.d.) *What is a software development process* [Online]. Available from: http://www.selectbs.com/analysis-and-design/what-is-a-software-development-process [Accessed 05 April 2017].

Synced. (2019) *medium* [Online]. Available from: https://medium.com/@Synced/how-random-forest-algorithm-works-in-machine-learning-3c0fe15b6674 [Accessed 2020].

toolsqa. (2018) *toolsqa* [Online]. Available from: https://www.toolsqa.com/software-testing/waterfall-model/ [Accessed 2020].