NAME: OGBENI JOSHUA

MAT NO: 18/ENG06/053

DEPARTMENT: MECHANICAL ENGINEERING

COURSE: Structured Computer Programming (ENG234)

DATE: FRIDAY 10TH APRIL 2020

ASSIGNMENT: Covid-19 has causes a serious pandemic across the world, with serious impacts been felt in all areas of humanities. As a young engineer working with a multi-national health company, you are saddle with a huge responsibility of designing web-based application that can detect, display, rate (degree of infection), store, transmit data obtained wirelessly and access the data via the web together with other features which the board of directors allow you to come up with.

1. Design the application following the software development cycle.
2. Critically discuss the hardware and software features.
3. Support your answer with a flowchart and an algorithm.
4. Draw the Top-down or Bottom-up design approach of the application.

A WEB- BASED HEALTHCARE MANAGEMENT SYSTEM

Software development life cycle processes includes:

1. Requirement analysis
2. Planning
3. Design
4. System development
5. Testing
6. Deployment

Requirement Analysis:

- It's a web-based Application.
- Users will Input their Full Name, Age, Email, Password and other important details.
- The Users details will be stored in the cloud database
- The cloud database will be accessible only to health officials so they will be able to get the data an analyse it.
- It will be Simple web application with a simple UI (User Interface) design.
- It will have a contact / Location section which will help users search for the nearest healthcare units for the Covid-19.
- The Cloud Database Will Possess the survey Questions, Location data of both the health officials and the user.
- The Survey Questionnaire will ask for the current state and the symptoms the user possess.
- The web-based application will ask the user to turn on their location of their device.
- The result of the survey will be made available to doctors for analysis
- The Cloud Database will also store symptoms of the user.

Planning Phase:

In system design, we adopted a platform-independent web based system for its easy use. Then we considered security and privacy because the personal data were handled via the Internet. Moreover, we considered that the users were able to check not only the data from the vital sensing system but also the analyzed report as feedback. The main concept for LMR system is a web based health care management system for effective COVID19 healthcare, The Outbreak of the disease has been a serious issue over the past months globally. The (WHO) World Health Organisation has termed the virus a pandemic due to its ability to spread at a fast rate. A method for controlling this pandemic hasn't been discovered. However, developing a web application which will be able to gather Data concerning the symptoms and location around the world, in other reduce physical contact which will be able to reduce the spread of the Covid-19. The Data will be stored on a Cloud Database which will be access strictly by the health officials. The point of adding a location feature to alert the users using the web application of someone that has the symptoms and where the person is located.

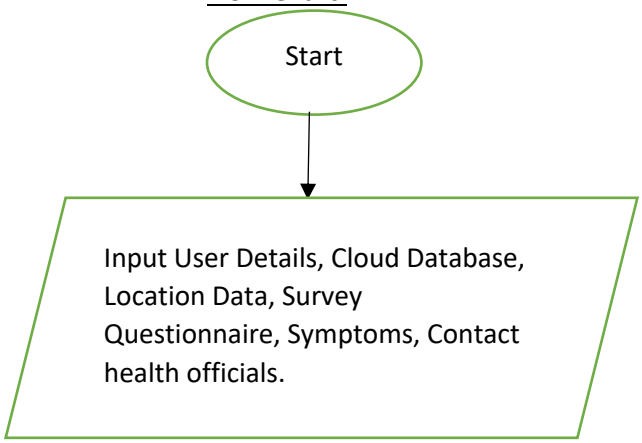
Hardware requirements: list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. A HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular system or application. The following sub-sections discuss the various aspects of hardware requirements. These include, Intel dual Core, i3 as the processor of the Os, Internet connection for the health center, clinical thermometer.

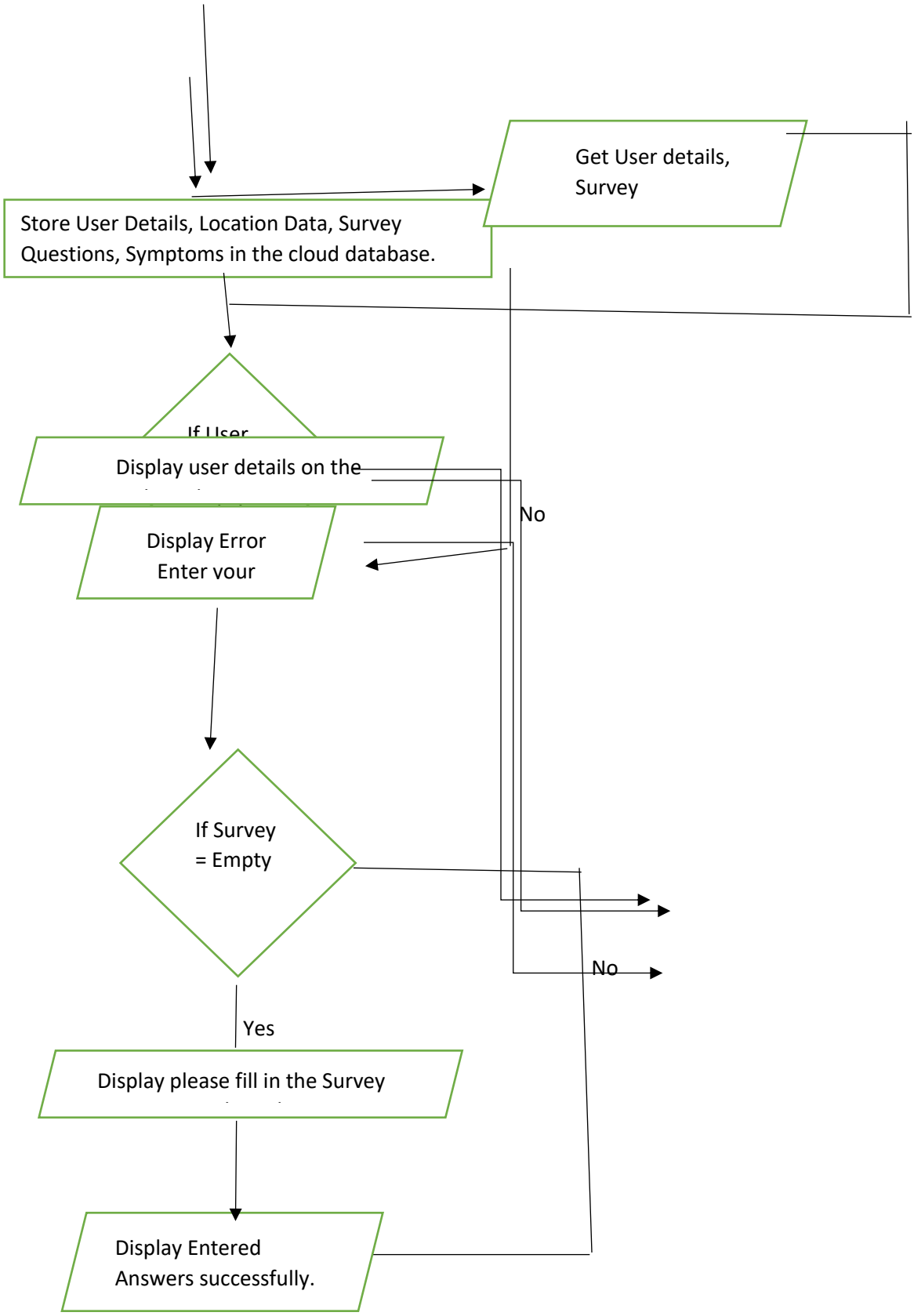
Software Requirements: deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

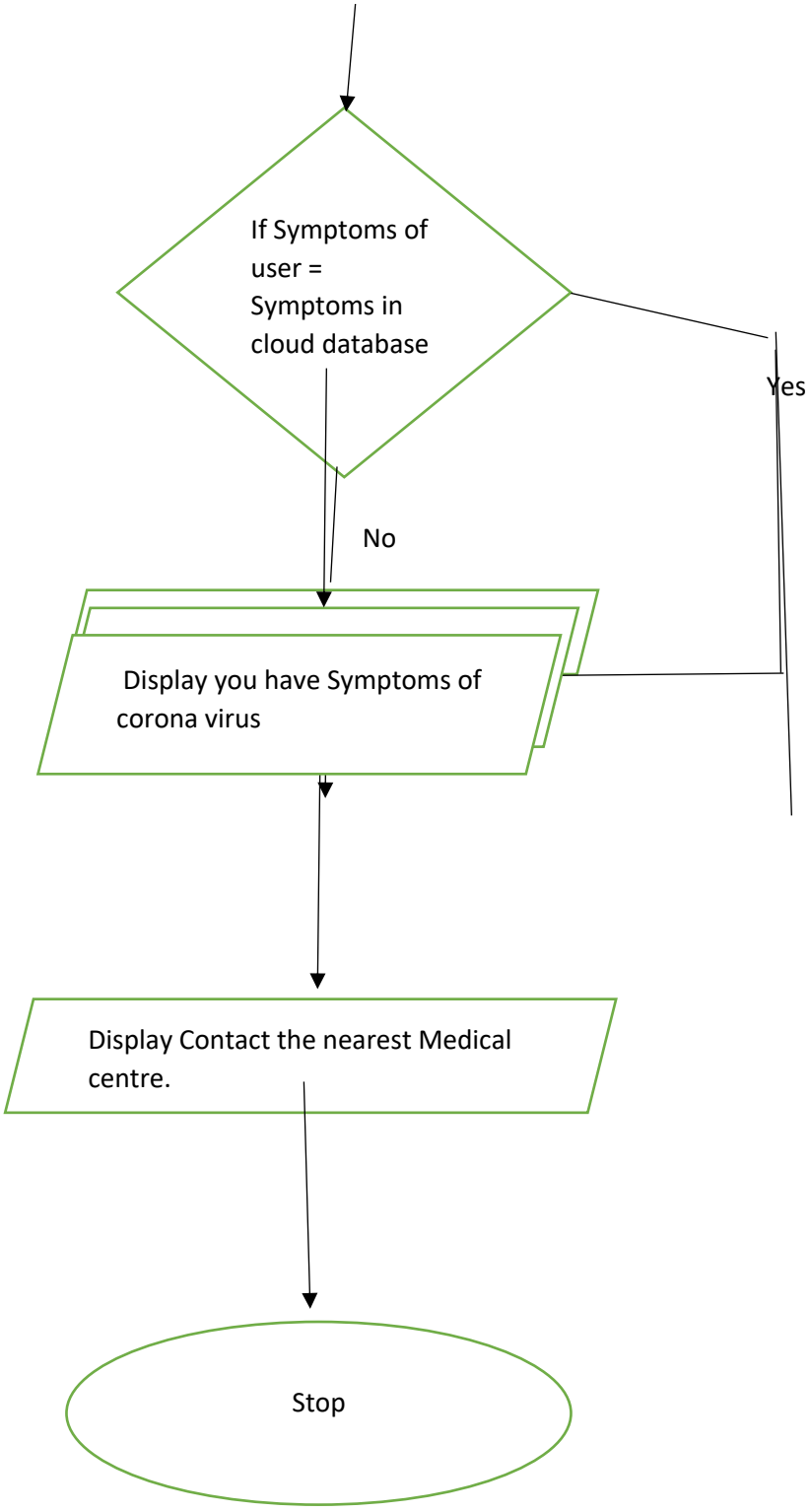
Algorithm:

1- Start
2- Input User Details, Cloud Database, Location Data, Survey Questionnaire, Symptoms, Contact health officials.
3- Store User Details, Location Data, Survey Questions, Symptoms in the cloud database.
4- Get User details, Survey.
5- If User details = Empty
   Display Error
   Else
   Display user details on the web application page.
6- If Survey = empty
   Display please fill in the Survey Questions
   Else
   Display Entered Answers successfully.
7- If Symptoms of user = Symptoms in cloud database
   Display you have Symptoms of corona virus
   Else
   End Survey
8- Display Contact the nearest Medical centre.
9- Stop

Flow Chart:




Get User details,
Survey
Store User Details, Location Data, Survey
Questions, Symptoms in the cloud database.
If User
Display user details on the
No
Display Error
Enter vour
If Survey
= Empty
No
Yes
Display please fill in the Survey
Display Entered
Answers successfully.


If Symptoms of user = Symptoms in cloud database
Yes
No
Display you have Symptoms of corona virus
Display Contact the nearest Medical centre.
Stop

Implementation:

The Web based Application will be written in Nodejs and python. Nodejs is the language that will be used for the user interface and front-end design and python will for the back-end implementation with the use of Django framework. Django framework is a python framework used for the development of web-based applications. The Cloud database which we are going to be using is called Firebase Database it is a google based cloud database which can be used for various implementations.

Testing:

The Web based application will be tested by various random people that will judge the web application strictly then give a feedback regarding the UI (User Interface) and UX (User Experience) of the application and if any error is found in the process of testing it shall be solved with the use of debugging.

Deployment:

This application is being released to detect, display the rate of virus, store, transmit and access data through the web together and its updated when necessary based on the health center feedback any new discovery will be implemented and sent as an update for the users.

Top-down design approach of the application:

A top-down design is the decomposition of a system into smaller parts in order to comprehend its compositional sub-systems. In top-down design, a system's overview is designed, specifying, yet not detailing any first-level subsystems.