Software development life cycle processes includes

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

Requirement analysis

The main concept for LMR system is a web based health care management system. For effective COVID19 healthcare, a network system monitoring one's vital signs and evaluating one's health conditions is highly desirable. In our laboratory, we have developed a vital sensing system for home healthcare. The purpose of this study is to design and implement a prototype web-based healthcare management system (WBHMS) to make effective use of the data that are measured by the vital sensing system.

Planning

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

Hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating 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.

Design

A well- defined algorithm for a web-based COVID19 Healthcare Management System

STEP1: Start

    2: Body status to the virus=0

    3: Add the COVID19 symptoms in the system

    4: Put the software involved

    5: Create a questionnaire

    6: Collect qualitative data

    7: Analyze Data

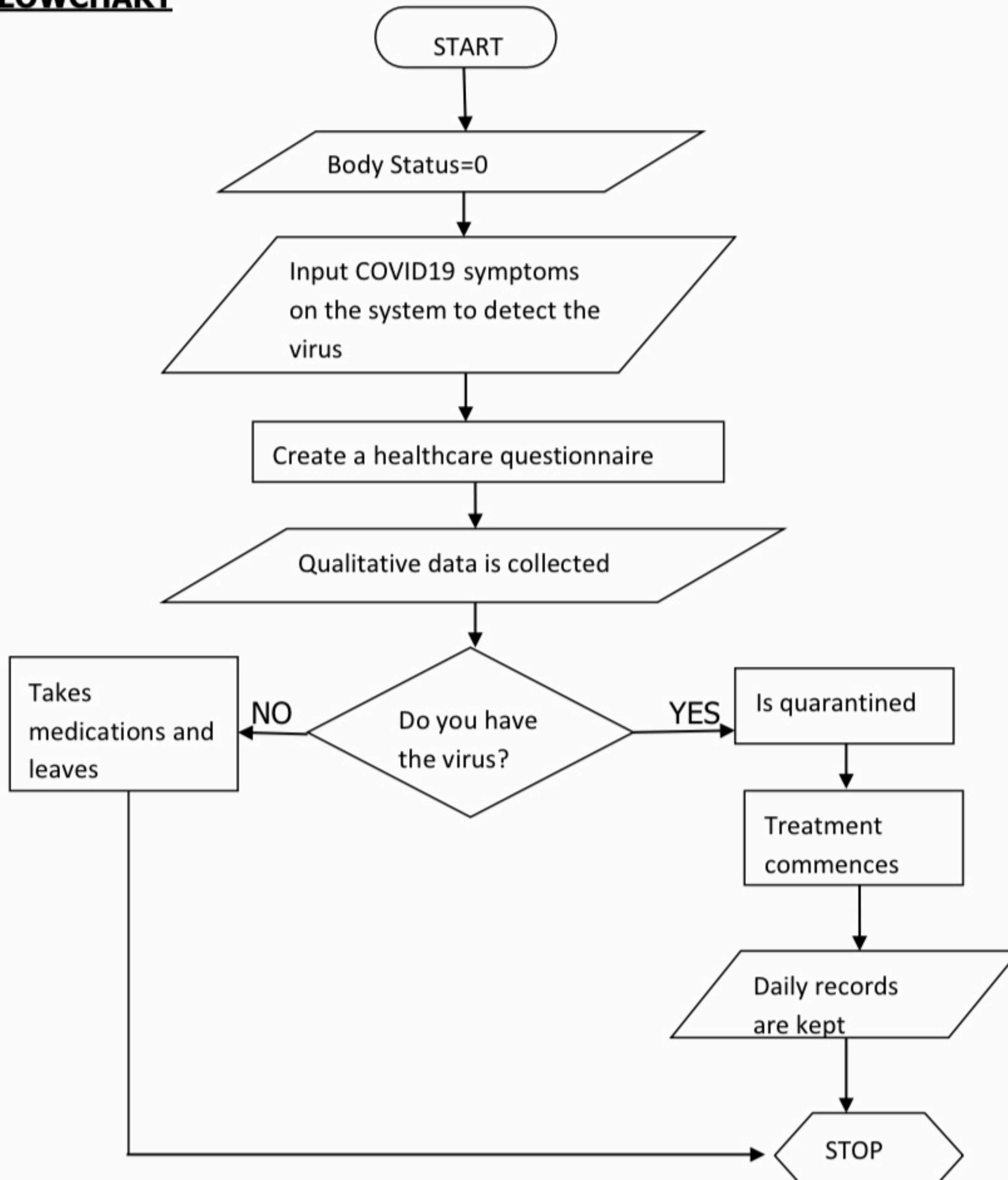   8: Body Status positive to the virus

    9: Else

    10: Body Status negative to the virus

    11: Display feedback

    12: Stop

FLOWCHART

```
                          ┌──────────────────┐
                          │      START       │
                          └──────────────────┘
                                   │
                                   ▼
                      ╱─────────────────────────╱
                     ╱      Body Status=0       ╱
                    ╱─────────────────────────╱
                                   │
                                   ▼
                  ╱─────────────────────────────╱
                 ╱   Input COVID19 symptoms    ╱
                ╱    on the system to detect the ╱
               ╱     virus                     ╱
              ╱─────────────────────────────╱
                                   │
                                   ▼
                   ┌─────────────────────────────┐
                   │ Create a healthcare questionnaire │
                   └─────────────────────────────┘
                                   │
                                   ▼
                 ╱─────────────────────────────╱
                ╱   Qualitative data is collected ╱
               ╱─────────────────────────────╱
                                   │
                                   ▼
```

Takes medications and leaves ◄── NO ── **Do you have the virus?** ── YES ──► **Is quarantined**

**Treatment commences**

Daily records are kept

STOP

Testing

The COVID19 Healthcare web apps need to be fool-proof, which is why testing the apps and their functionalities become so important. Right from testing the security and compliances to the workability and the integration of the app, testing can also be done with different operating systems, internet connections and hardware.

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

TOP- DOWN DESIGN APPROACH OF THE APPLICATION
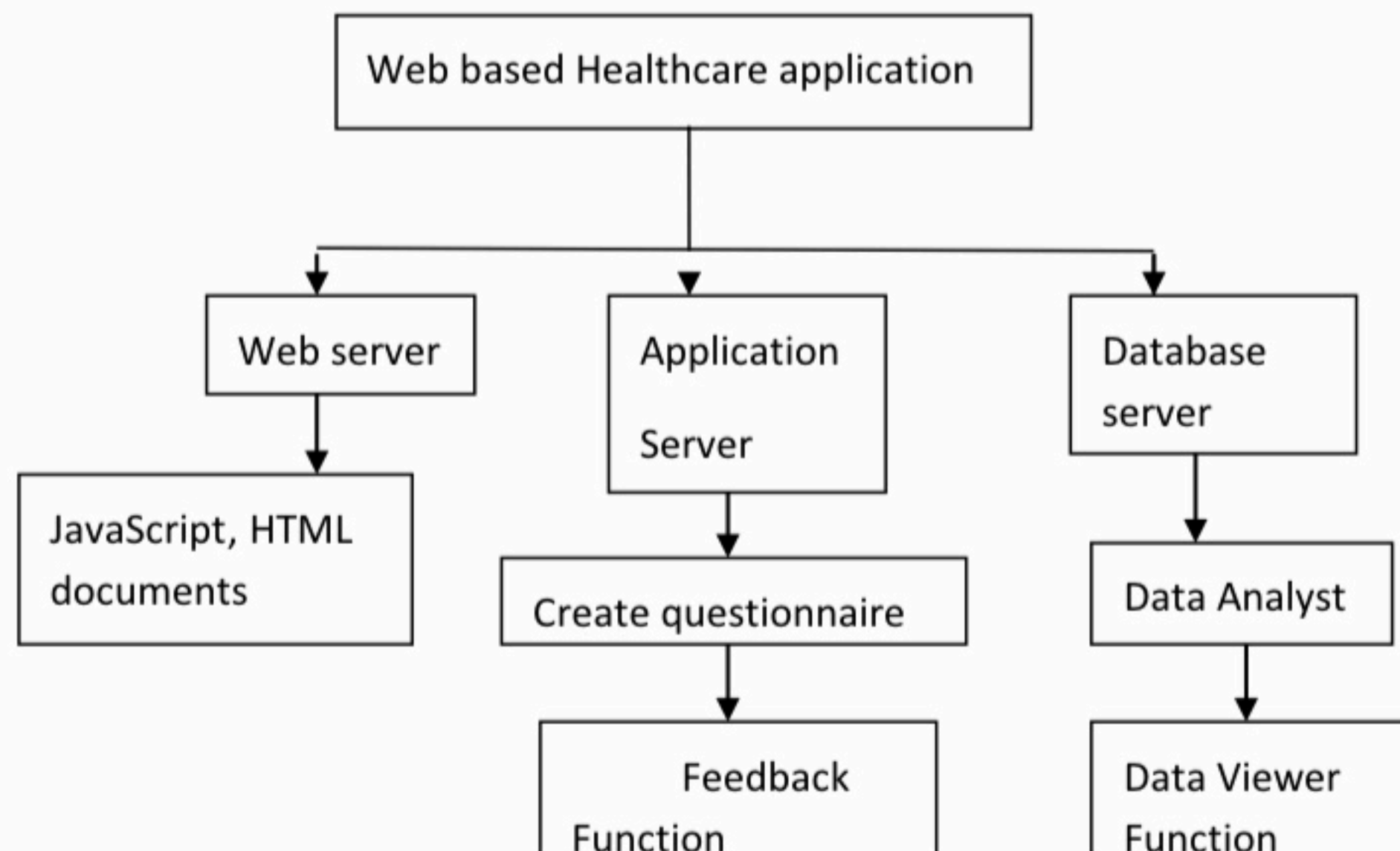
Software features should include;

- An administration interface which can only be accessed by specified users through an approved password, an admin can modify and make changes to various part of the application

- An interface that indicates number of recorded cases worldwide.

- A world map that shows countries affected by the disease with their numbers in a well defined legend

NAME:ADEDOKUN   PRECIOUS

ADEOLA

DEPARTMENT:COMPUTER

ENGINEERING

MATRIC NUMBER:18/ENG02/006

COURSE CODE:ENG224

COURSE   TITLE:STRUCTURED

COMPUTER PROGRAMMING