ABUAD

COVID DETECTION APP

AMADI-DURU,C.MELVIN

18/ENG04/013

ELECT/ELECT

AN ENG224 PRESENTATION BY AMADI-DURU,C.MELVIN

Conceptualization: the idea being a revolutionary application that can help with the detection and analysis of the general health condition of a human being. Design conceptualization is the process of generating ideas for an optimum solution to the design problem. These ideas should stem originally from the product idea and stated definitions of the design problem

Specification: A specification language is a formal language in computer science used during systems analysis, requirements analysis, and systems design to describe a system at a much higher level than a programming language, which is used to produce the executable code for a system. This is the breakage of the application into modules.

Design

Prerequisites

To make a data-centric web app from the bottom-up, it is advantageous to understand:

1. Backend language (e.g. Python, Ruby) - control how your web app works
2. Web front end (HTML, CSS, JavaScript) - for the look and feel of your web app
3. DevOps (Github, Jenkins) - Deploying / hosting your web app

The design involves the use of a flowcharts and algorithms to represent in detail the design stages of the web based app development. The below are the design protocols followed

-SKETCHING: During the sketching stage; *navigation, branding, forms, buttons and any other interactive* elements that'll help the patient communicate with the health personnel are required. Taking the COVID app's functionality into consideration, a conscious effort on the overall design is put into play. Also, notes shall be taken if in the nearest future alterations are to be made.

- **PLANNING THE WORKFLOW:** In simple terms this would involve putting ourselves in the shoes of the user. It'll involve putting down various workflows for the application such as:

  -How does a user signup on the COVID app

  -Do they receive a verification email from the health officials

  -How does a user login on the app?

  -How does a user change their password?

  -How does a user navigate through the app e.g. from the sign in page to the test results

  -How does a user change their user settings e.g. the health official assigned to the user?

  After which all the different web pages the app will contain is written down and also the stages of each page.

- **WIRE FRAMING/PROTOTYPING THE UI**: Wire framing is the process of designing a blueprint of your web application while prototyping is taking wire framing a step further, adding an interactive display. Both of these can be done with the use of AdobeXD; a wire framing/prototyping tool.

There are a number of activities performed for designing user interface. The process of GUI design and implementation is alike SDLC. Any model can be used for GUI implementation among Waterfall, Iterative or Spiral Model.

A model used for GUI design and development should fulfill these GUI specific steps

- **GUI Requirement Gathering** - The designers may like to have list of all functional and non-functional requirements of GUI. This can be taken from user and their existing software solution.
- **User Analysis** - The designer studies who is going to use the software GUI. The target audience matters as the design details change according to the knowledge and competency level of the user. If user is technical savvy, advanced and complex GUI can be incorporated. For a novice user, more information is included on how-to of software.
- **Task Analysis** - Designers have to analyze what task is to be done by the software solution. Here in GUI, it does not matter how it will be done. Tasks can be represented in hierarchical manner taking one major task and dividing it further into smaller sub-tasks. Tasks provide goals for GUI presentation. Flow of information among sub-tasks determines the flow of GUI contents in the software.

- **GUI Design & implementation** - Designers after having information about requirements, tasks and user environment, design the GUI and implements into code and embed the GUI with working or dummy software in the background. It is then self-tested by the developers.
- **Testing** - GUI testing can be done in various ways. Organization can have in-house inspection, direct involvement of users and release of beta version are few of them. Testing may include usability, compatibility, user acceptance etc.

Implementation

It is during this phase that the project becomes visible to outsiders, to whom it may appear that the project has just begun. In summary, the web based application is put out into the public for usage and early feedback.

There are several tools available using which the designers can create entire GUI on a mouse click. Some tools can be embedded into the software environment (IDE).

GUI implementation tools provide powerful array of GUI controls. For software customization, designers can change the code accordingly.

There are different segments of GUI tools according to their different use and platform.

Examples

Mobile GUI, Computer GUI, Touch-Screen GUI etc. Here is a list of few tools which come handy to build GUI:

- FLUID
- AppInventor (Android)
- Lucid Chart
- Wavemaker
- Visual Studio

Testing & Debugging

Also, it's important to know the testing that is done during debugging has a different aim than final module testing. Final module testing aims to demonstrate correctness, whereas testing during debugging is primarily aimed at locating errors.

Release & Update

This is the publishing of the web application for usage to the public.

HARDWARE & SOFTWARE FEATURES

HARDWARE SYSTEM

-SENSORING SYSTEM: This includes the retina and temperature scanner hardware attached to the system. It takes the temperature and vitals of the patient

-DISPLAY SYSTEM: This displays the result of the sensoring system test.

-PRINTER: This prints out the final result and address of medical centers near the patient that he could get treatment if found positive to any ailment.

SOFTWARE FEATURES

The software becomes more popular if its user interface is:

- Attractive
- Simple to use
- Responsive in short time
- Clear to understand
- Consistent on all interfacing screens

UI is broadly divided into two categories:

- Command Line Interface
- Graphical User Interface

Command Line Interface (CLI)

CLI has been a great tool of interaction with computers until the video display monitors came into existence. CLI is first choice of many technical users and programmers. CLI is minimum interface software can provide to its users.

CLI provides a command prompt, the place where the user types the command and feeds to the system. The user needs to remember the syntax of command and its use. Earlier CLI were not programmed to handle the user errors effectively.

A command is a text-based reference to set of instructions, which are expected to be executed by the system. There are methods like macros, scripts that make it easy for the user to operate.

CLI uses less amount of computer resource as compared to GUI.

A text-based command line interface can have the following elements:

- **Command Prompt** - It is text-based notifier that is mostly shows the context in which the user is working. It is generated by the software system.
- **Cursor** - It is a small horizontal line or a vertical bar of the height of line, to represent position of character while typing. Cursor is mostly found in blinking state. It moves as the user writes or deletes something.
- **Command** - A command is an executable instruction. It may have one or more parameters. Output on command execution is shown inline on the screen. When output is produced, command prompt is displayed on the next line.

Graphical User Interface

Graphical User Interface provides the user graphical means to interact with the system. GUI can be combination of both hardware and software. Using GUI, user interprets the software.

Typically, GUI is more resource consuming than that of CLI. With advancing technology, the programmers and designers create complex GUI designs that work with more efficiency, accuracy and speed.

GUI Elements

GUI provides a set of components to interact with software or hardware.

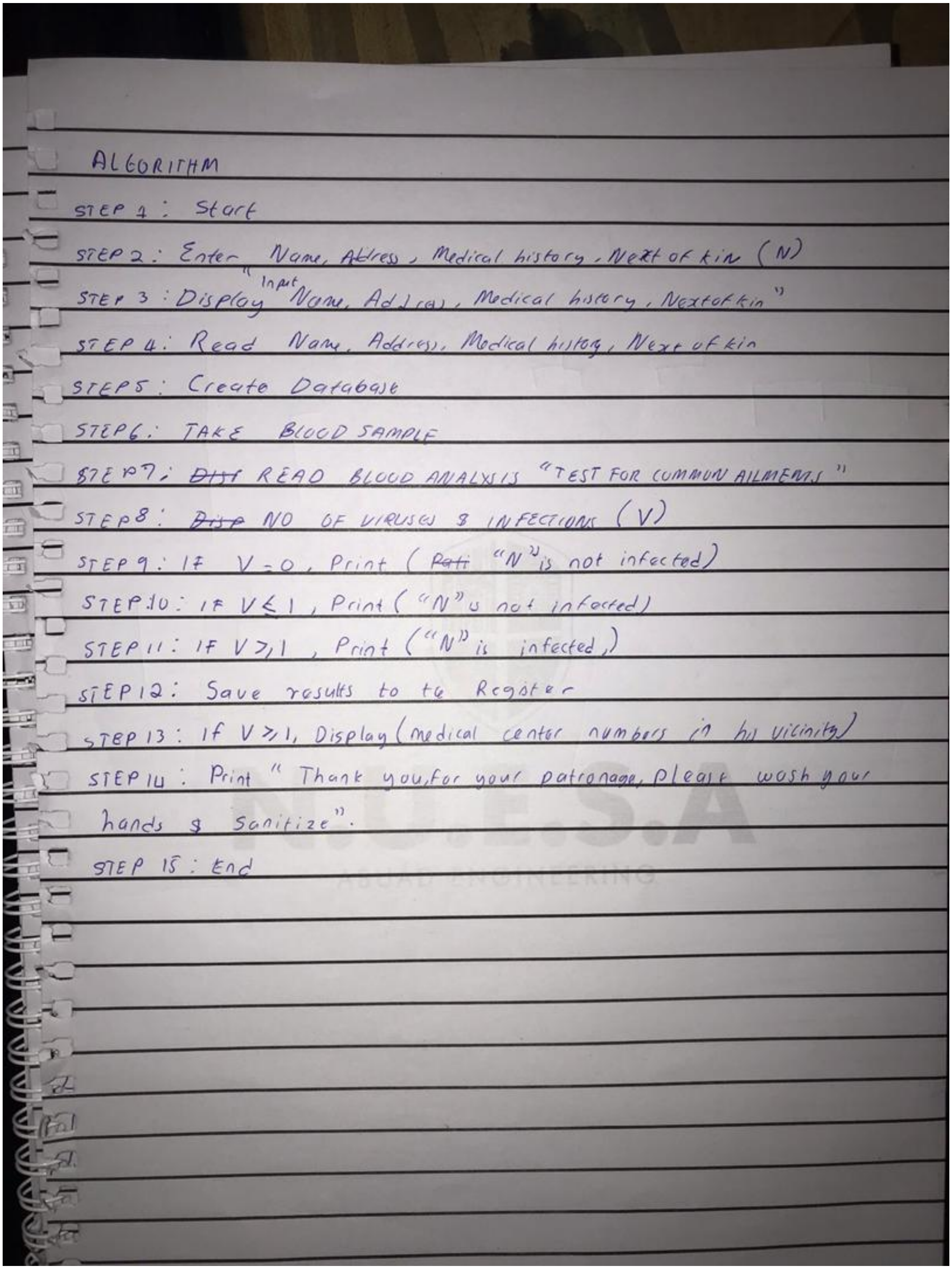
Every graphical component provides a way to work with the system. A GUI system has following elements such as:

- **Window** - An area where contents of application are displayed. Contents in a window can be displayed in the form of icons or lists, if the window represents file structure. It is easier for a user to navigate in the file system in an exploring window. Windows can be minimized, resized or maximized to the size of screen. They can be moved anywhere on the screen. A window may contain another window of the same application, called child window.
- **Tabs** - If an application allows executing multiple instances of it, they appear on the screen as separate windows. **Tabbed Document Interface** has come up to open

multiple documents in the same window. This interface also helps in viewing preference panel in application. All modern web-browsers use this feature.

- **Menu** - Menu is an array of standard commands, grouped together and placed at a visible place (usually top) inside the application window. The menu can be programmed to appear or hide on mouse clicks.
- **Icon** - An icon is small picture representing an associated application. When these icons are clicked or double clicked, the application window is opened. Icon displays application and programs installed on a system in the form of small pictures.
- **Cursor** - Interacting devices such as mouse, touch pad, digital pen are represented in GUI as cursors. On screen cursor follows the instructions from hardware in almost real-time. Cursors are also named pointers in GUI systems. They are used to select menus, windows and other application features

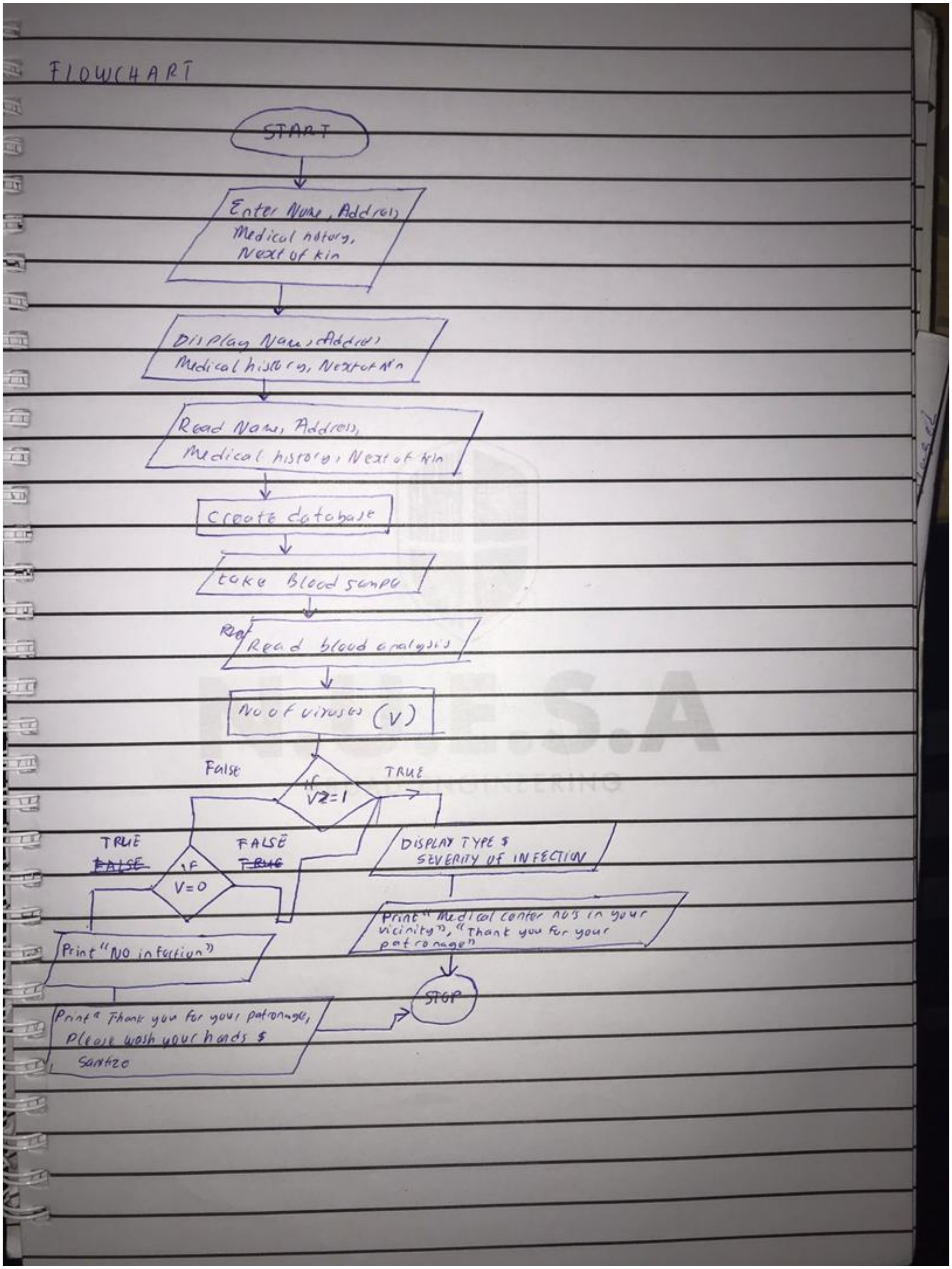
ALGORITHM



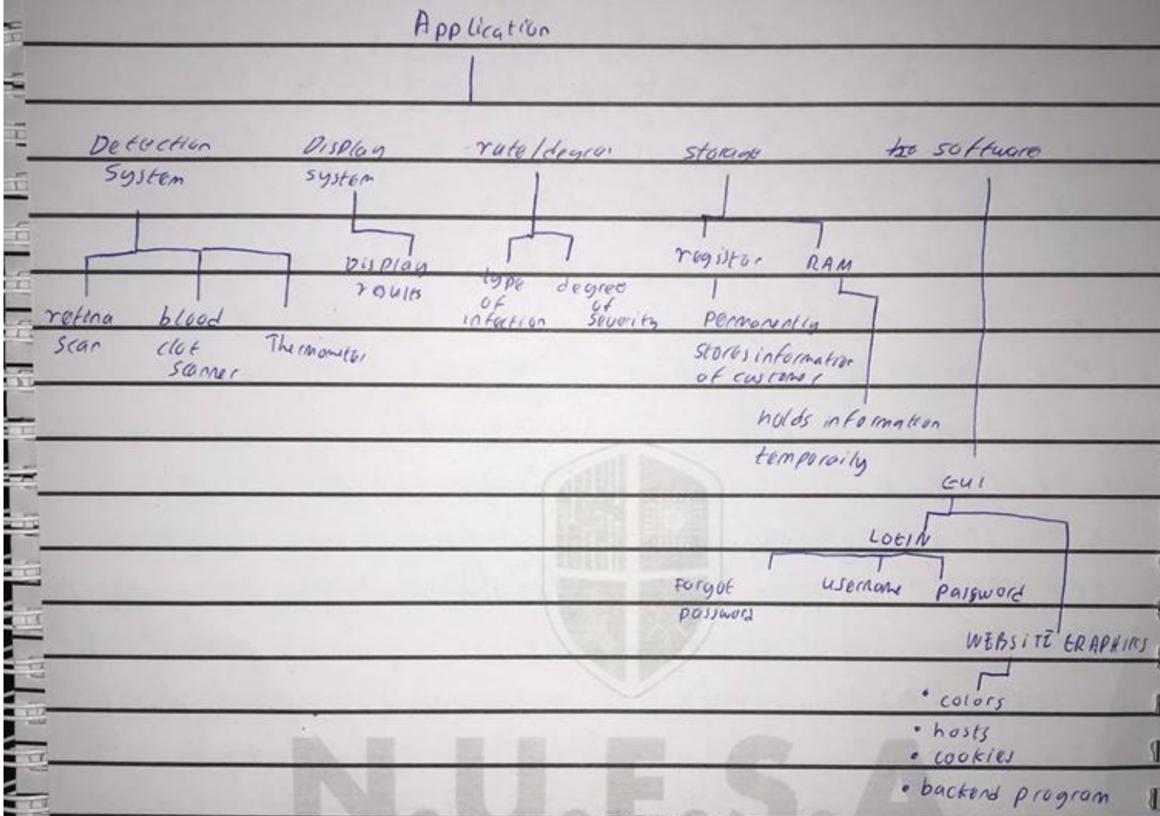
ALGORITHM

```
STEP 1 : Start
STEP 2 : Enter Name, Adress, Medical history, Next of kin (N)
STEP 3 : Display "Input Name, Address, Medical history, Nextofkin"
STEP 4 : Read Name, Address, Medical histoy, Next of kin
STEP 5 : Create Database
STEP 6 : TAKE BLOOD SAMPLE
STEP 7 ; ~~Dist~~ READ BLOOD ANALYSIS "TEST FOR COMMON AILMENTS"
STEP 8 : ~~Disp~~ NO OF VIRUSES & INFECTIONS (V)
STEP 9 : If $V = 0$, Print ( ~~Pati~~ "N" is not infected)
STEP 10 : IF $V \leq 1$, Print ("N" is not infected)
STEP 11 : IF $V \geq 1$, Print ("N" is infected,)
STEP 12 : Save results to te Register
STEP 13 : If $V \geq 1$, Display (medical center numbers in his vicinity)
STEP 14 : Print " Thank you, for your patronage, please wash your
          hands & sanitize".
STEP 15 : End
```

FLOW CHART


FLOWCHART
START
Enter Name, Address
Medical history,
Next of kin
Display Name, Address
Medical history, Next of kin
Read Name, Address,
Medical history, Next of kin
Create database
take Blood sample
~~Rea~~ Read blood analysis
No of viruses (V)
False
TRUE
If
V≥1
TRUE
~~FALSE~~
FALSE
~~TRUE~~
IF
V=0
DISPLAY TYPE $
SEVERITY OF INFECTION
Print "NO infection"
Print "Medical center no's in your vicinity", "Thank you for your patronage"
Print " Thank you for your patronage,
Please wash your hands $
Sanitize
STOP

TOP-DOWN


Application
Detection System
Display system
rate/degree
Storage
~~to~~ software
retina scan
blood clot scanner
Thermometer
Display results
type of infection
degree of severity
register
RAM
Permonently
Stores information of customer
holds information temporaily
GUI
LOGIN
Forgot password
username
Password
WEBSITE GRAPHICS
• colors
• hosts
• cookies
• backend program