

WEB-BASED APPLICATION DESIGN USING SOFTWARE DEVELOPMENT LIFE CYCLE:

This project was instigated upon the knowledge that our company has a task regarding software development outsourced to us. A multi-national health company has requested the services and assistance of our company's software development specialists in designing a web-based, disease-specific detection and analysis application. Discussing the project scope, we found out that the customer needed a Covid-19 specific application that can detect, display, rate (degree of infection), and store and transmit data obtained wirelessly while being able to access the data via the web, whilst being given creative freedom on what else the application could do in addition to the previous mentioned functions. The new application was to provide the most efficient ways of detecting and varying the scale of infection of the long-familiar Covid-19 disease.

With the above in mind, we got down to the application development. We first decided on the web applications functionality which could include permitting users to create an account, set, change, and retrieve passwords; this ensures security of the user's health results from the app. After deciding on the application's functionality, the team sketched the web app. This sketching involves considering the following: Navigation, branding, forms, buttons and any other interactive element. The team made several annotations and outlined how the application should work in this stage. Next the team set up the applications workflow, which included: How a user signs up, if published health results will be received by user via email, how users navigate the application, payment methods and others.

The application was initialized with reference values to determine the severity of user's condition. These reference values will be compared with the user's input values; an air sample and a CT scan photograph; to determine whether the user's condition is at stages A, B, C or D.

At the solution stabilization stage, a test automation specialist ensured that the system worked as intended. The custom web framework we created specifically for this project was able to facilitate continuous and efficient testing.

Then the application was finally delivered to the customer. They got an innovative Covid-19 testing, rating and display application to help curb the spread of the virus with quick and efficient real-time detection of the virus.

Our team has been supporting and maintaining the platform for several years already, by regularly updating and delivering new features.

HARDWARE AND SOFTWARE COMPONENTS:

INTERFACE: Using JavaScript as mentioned in the Software Development Life Cycle (SDLC) to enable application-specific methods such as access to keyboard and mouse. Client-side scripting was used to add functionality, most especially to create an interactive experience that does not require page reloading. ASP.NET was used to coordinate client-side scripting with server-side technologies.

STRUCTURE: The application was broken into three-tiers; a web browser being the first tier (presentation), an engine using some dynamic Web content technology (in this case JSP/Java) is the middle tier (application logic), and a database is the third tier (storage). The web browser sends requests to the middle tier, which services them by making queries and updates against the database and generates a user interface.

DEVELOPMENT: A web application framework was used for this application framework. The web application framework used is the ASP.NET. This use of ASP.NET helped reduce the number errors in a program, making the code simpler, and allowing several members of the team to concentrate on the framework while another focuses on a specified user case.

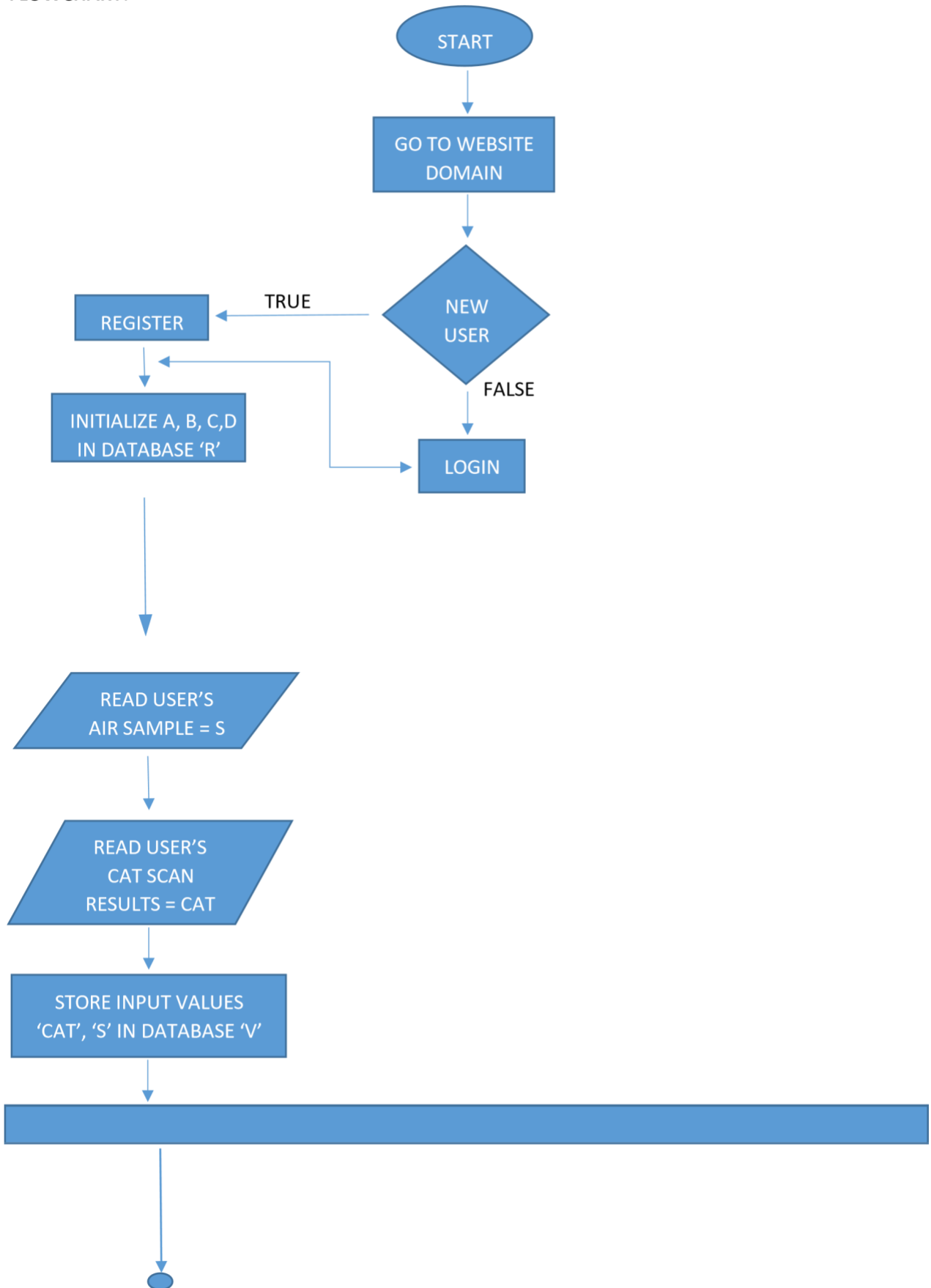
In-depth details of the rest of the software is explained below;

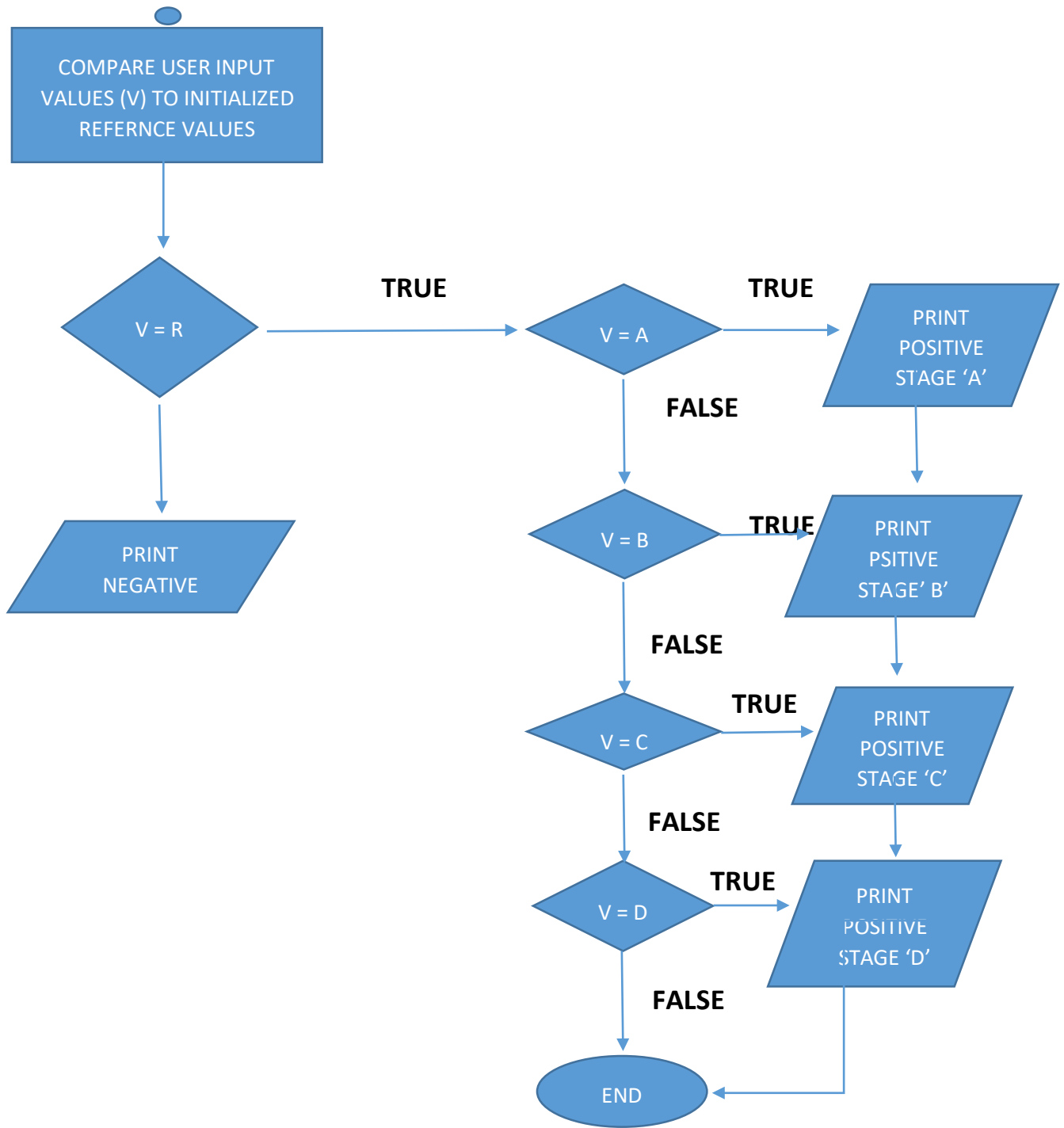
The application's sketches are turned into a wireframe/prototype in this process. Wire framing is the process of designing a blueprint of your web application. Prototyping is adding an interactive display to that blueprint. The team applied Adobe XD in the prototype/wireframe design. It is decided on information to be stored in a proposed database. And from that decision, a Database Management System (DBMS) was chosen. An SQL database was used, specifically a Microsoft SQLServer. Next a data separation method was deciding upon; Physical separation, taking into full consideration the security of our user's information. Physical separation, as it happens to be renowned for being very secure and durable. The next step involved building the frontend; the visual element of the application; and backend; what manages your data. The frontend was developed using JavaScript using server pages while the backend was developed using the tool Budibase. The job of the backend is to: provide HTTP endpoints for your front end, authenticate users, authorization and finally serve the frontend. Finalizing the application development process involved hosting (running the web app on a particular server) the web application. Since Budibase was used in generating the front end, the team simply used Budibase hosting and a domain was purchased.

The hardware features include a data transmission cable to transfer data from an external CAT scanner to be compared with the previously mentioned, initialized reference values and a router and servers for internet connectivity and to allow shared resources among the users.

FLOWCHART AND ALGORITHM:

FLOWCHART:





ALGORITHM:

1. START
2. READ USER ID
3. READ A, B, C, D // INITIALIZED REFERENCE VALUES = R
4. STORE A, B, C, D IN R
5. READ AIR SAMPLE = S
6. READ CAT SCAN RESULTS = CAT
7. STORE S,CAT IN V //USER'S INPUT VALUES = V
8. IF V=R
9. PRINT "POSITIVE"
10. ELSE
11. PRINT "NEGATIVE"
12. WHILE V=R
13. IF V=A
14. PRINT "POSITIVE, STAGE A" 15. IF V=B
16. PRINT "POSITIVE, STAGE B"
17. IF V=C
18. PRINT "POSITIVE, STAGE C"
19. IF V=D
20. PRINT "POSITIVE, STAGE D"
21. END

TOP-DOWN DESIGN APPROACH:

