

NAME: EKUNDAYO OLUWATOBI FAVOUR

MATRIC NO: 18/ENG01/010

DEPARTMENT: CHEMICAL ENGINEERING

COURSE CODE: ENG 224

COURSE TITLE: STRUCTURED COMPUTER PROGRAMING

ASSIGNMENT

Covid-19 has caused 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 have 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

1. Requirement analyses
2. Planning
3. Software design such as architectural design
4. Software development
5. Testing
6. Deployment

1. First and foremost, let us see what the software development cycle is.

- Identify the Current Problems

This stage of the SDLC means getting input from all stakeholders, including customers, salespeople, industry experts, and programmers. Learn the strengths and weaknesses of the current system with improvement as the goal.

- Plan

In this stage of the SDLC, the team determines the cost and resources required for implementing the analyzed requirements. It also details the risks involved and provides sub-plans for softening those risks.

In other words, the team should determine the feasibility of the project and how they can implement the project successfully with the lowest risk in mind.

- Design

This phase of the SDLC starts by turning the software specifications into a design plan called the Design Specification. All stakeholders then review this plan and offer feedback and suggestions. It's crucial to have a plan for collecting and incorporating stakeholder input into this document. Failure at this stage will almost certainly result in cost overruns at best and the total collapse of the project at worst.

- Build

At this stage, the actual development starts. It's important that every developer sticks to the agreed blueprint. Also, make sure you have proper guidelines in place about the code style and practices.

For example, define a nomenclature for files or define a variable naming style such as `camelCase`. This will help your team to produce organized and consistent code that is easier to understand but also to test during the next phase.

- Code Test

In this stage, we test for defects and deficiencies. We fix those issues until the product meets the original specifications.

In short, we want to verify if the code meets the defined requirements.

- Software Deployment

At this stage, the goal is to deploy the software to the production environment so users can start using the product. However, many organizations choose to move the product through different deployment environments such as a testing or staging environment.

This allows any stakeholders to safely play with the product before releasing it to the market. Besides, this allows any final mistakes to be caught before releasing the product.

Now, the steps involved in designing the application are:

Step 1: Pick an Idea

Once we have a direction, write down in one sentence the purpose, and, if applicable, the most important target users for this app.

Step 2: Define the Core Functionalities

Here, we think about what the app should be able to do and list them out. If we end up listing a bunch of things, then, we take a better look and ask if this app really needs, for example, Facebook login to work? Does it REALLY need to upload data to some cloud to work?

It's great to have a dream project with awesome specs, but right now the point is not building something that's complete with a lot of cool features. Keep in mind that no app is ever complete, and everything starts out simple.

We take a look at the list of functions we've made, and if it's too long, we start crossing out functionalities the app can work without.

Step 3: Sketch The App

Nothing is quicker than a pen & paper. By now, we have a pretty solid idea of what our app should do, so, we sketch out the wireframe of the app's UI (user interface). Where buttons should be located, what the purpose of that button is, and so on.

We write down notes and flesh out how the app should work. We're still in the brainstorming stage, so we change stuff around until we're somewhat satisfied with it.

Step 4: Plan The App's UI Flow

We have a good idea of what your app will look like, what every element should do, and how users can interact with the app. Now it's time to figure out the app's UI flow. Meaning, how a user should use the app from start to end, every step

they have to take and every scenario they might encounter. Try to consider every use-case.

We include all the actions our user can take in the flowchart. For example, if our app requires users to log in, how do they create an account? What if users forgot their password? What if they inputted the wrong password? What should a user be able to do on each interface (add new list item > save; edit > save/delete)? So on, so forth. This is done the quickest with a pen and paper.

Step 5: Designing the Database

After we planned out every possible scenario, take a look at it to determine what sort of data you would need to keep. For example, if the app requires users to create an account, we would need to keep track of things such as the username, user ID, the user's email, the password, whether user's email is confirmed, when the account was created as well as every time the user has logged into your app.

If we're building a Twitter clone, we would need to know the tweet's ID, the tweet's content, when the tweet was published/retweeted, how many retweets it has, and how many stars it has.

We would also need to keep a record of a user's retweets and stars.

Step 6: UX Wireframes

Okay, you have the back-end planned. Now, it's time to plan our front-end.

Step 6.5 (Optional): Design the UI

This is an optional step, but if we intend to specialize in front-end development or if we care a lot about what our app is going to look like to the point where a prettier app would motivate us to code, definitely go ahead and design the app so all those wireframe UI elements can be replaced with nicer-looking ones.

Step 7: Researching Solutions

An important skill to learn as a programmer is to evaluate when you should use something another developer has already written, and when to build the function yourself.

Since everyone is building a unique app, not every use case is the same. Thus, we have to judge when to use an existing solution and when to build our own, and so, we get better at doing so with experience.

2. Software

- COVID-19 global hackathon: an online space with focus topics where developers can ideate, experiment and build software solutions to help address this crisis.
- Ovio: Curated a list of open source software and data science projects, repositories, databases, etc. tailored specifically to learn more about and track COVID-19.
- Kaggle: If you have ever used Kaggle to help prepare for your coding interviews, explore machine learning courses, or compete in challenges, you can give back by participating in their ongoing COVID-19 research challenge. You can devote yourself to any of 10 key task questions centered around forecasting, transmission and incubation, therapeutics, origins of the virus, and more depending on what appeals to you.
- Access controls
- Graphical User Interface(GUI)

Hardware

One of the major challenges we face has been a lack of personal protective equipment (PPE) to provide not only citizens, but more critically, health professionals and medical personnel on the front lines. Masks, face shields, and test swabs are in abysmal scarcity. I have been following the great work of various companies and organizations around the world scrambling to equip hospitals and clinics with this missing PPE day in and out. Other companies like Carbon and Formlabs, are pivoting business models and resources rapidly and creatively to begin producing this kind of PPE as well as parts required for ventilators.

- 3D Printing
- Design Chops

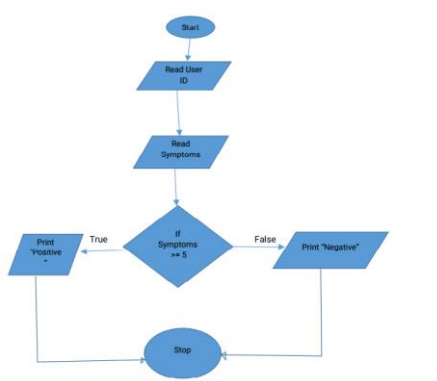
3. (a) Algorithm

1. Start
2. Read user ID
3. Read symptoms

4. If symptoms ≥ 5
 “print positive”
5. Else print “negative”
6. End

(b) Flowchart

4.



TOP-DOWN DESIGN APPROACH

