FIT3161: Advanced Computer Science Project 1
Project Initial Concept and Design
Project Supervisor: Dr. Soon Lay Ki

Group: MCS15
Ooi Yi Sen – 30720699
Chan Wai Han - 31555373
Nawwaf Ali - 31261949
Yeonsoo Kim - 29584612

Table of Contents

1.0 Overview

1.1 Project Objectives

Coronavirus disease is an infectious disease caused by the SARS-CoV-2 virus. Many people might not have the appropriate knowledge, acceptance, and perception of Covid. Our business goal is to provide a Q&A system for this virus. Our team believes that our Q&A system can provide useful information related to covid, and ensure the information is accessible to everyone. Our main business objective is to provide useful information relating to Covid. Our social objective of this project is to improve everyone's knowledge, acceptance, and perception of Covid. Our human objective in this project is to create more job opportunities and also the development of human resources. The final objective of our project is to provide an automatic COVID-19 answering website system that will be open to the public. It will offer both expert-level information and customer-level information.

1.2 Project Scope

In-scope:

- Access to the latest information about Covid. Users must be able to search for information about Covid, read the user guide on how to use our system, and see how to apply some of the information in real life
- Access to reliable databases. So that our system can provide accurate information to our users
- Access to technical tools, languages, and software for our project.
- Able to provide accurate information when our users key in their enquires

Out-of-scope:

- Other features suggested by the users if they add value to the business
- Access via the web, tablet, or mobile
- Detailed search option for finding COVID-19 information
- Automated update on latest COVID-19 information

2.0 Architecture Design

2.1 Creating a chatbot (sketch)

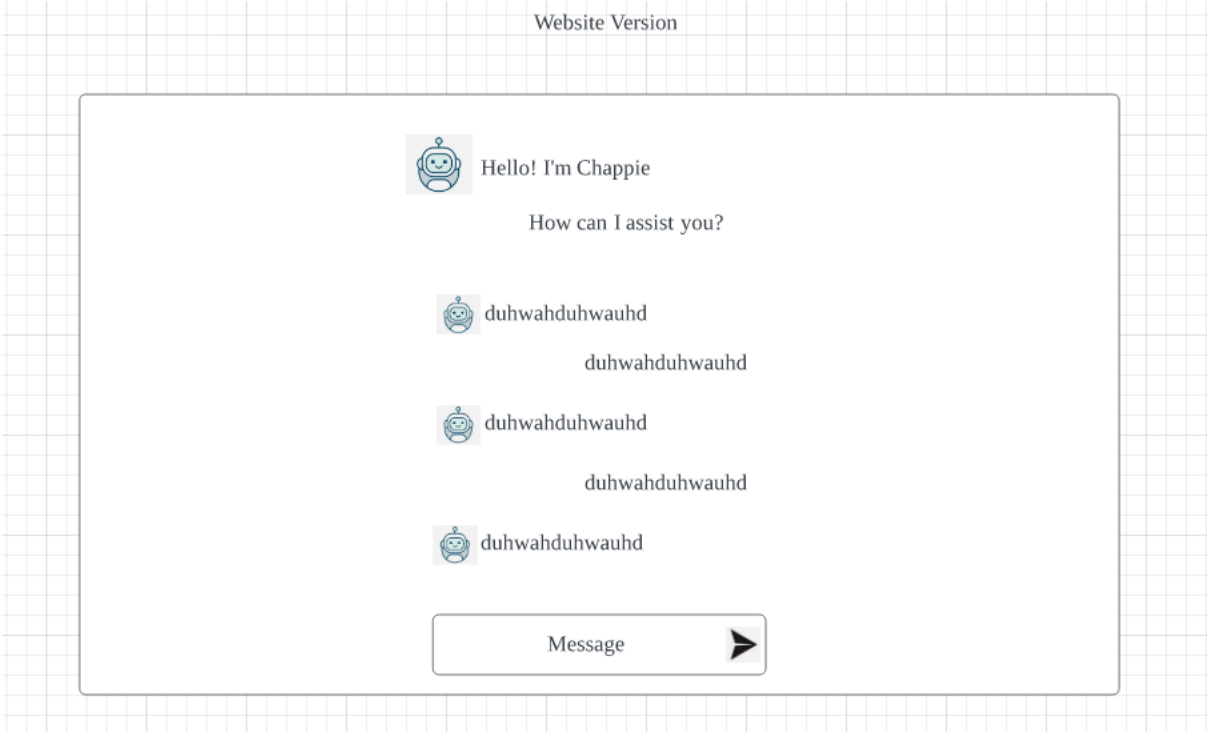


Figure 1: Sketch of the chatbot

In Figure 1, we came up with a sketch of the website version for our chatbot. The chatbot would generate an automated message to greed the user whenever they come into the website. Then, the user can input their question via the textbox provided. After the user finishes their query, there will be an upload button for the user to upload their question.

2.2 Flowchart Diagram for the chatbot system

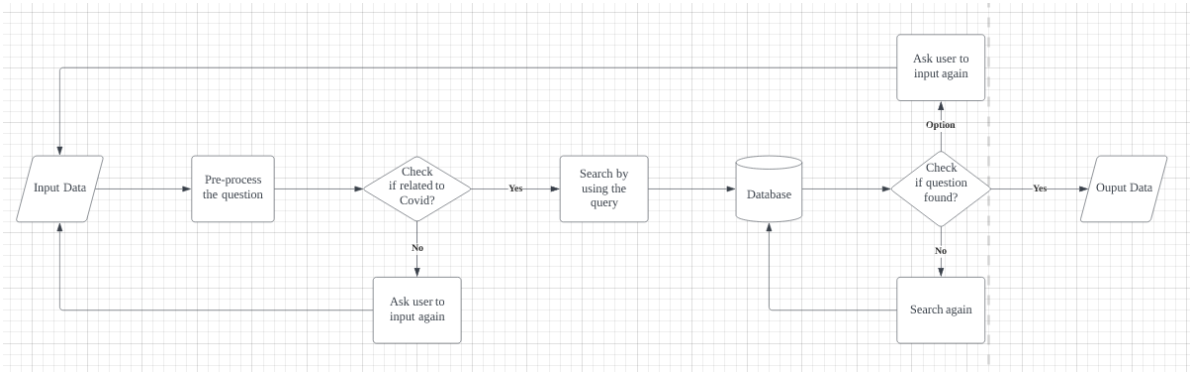


Figure 2: Flowchart Diagram for the chatbot system

In Figure 2, we use a flowchart diagram to show the process for our chatbot. For our project, we do not start the project from scratch. We use a pre-build algorithm and modify it to make further improvements. The steps shown in Figure 2 are just a rough idea of what the whole flow of the chatbot system would look like, as changes will be made as we progress more into the unit. Firstly, our chatbot will take an input question from the user, and the algorithm will then pre-process the question. After pre-processing the question, we can check whether the user input is covid related. Vice versa, if the questions are not covid related, our algorithm will then ask the user to input their query again. Next, if the question is covid related, the algorithm will search our database using the query. If the question is in the database, we will then output the response to the user. If the question is undiscovered after the first search, the algorithm will search the database again. Finally, if the algorithm cannot find the query, the system will prompt the user to input another question or rephrase their original question.

3.0 Software and Hardware Specification

3.1 Component specification

There will be multiple software components that would be used in developing our software product, from the frontend to the backend, to the data management. The components that will be used are as follows:

1. HTML/CSS source codes for website development
2. Python back-end development
3. SQL Developer for the database server of the system
4. GitHub project
5. Testing scripts that include sample consumer questions and sample expert questions
6. Software documentation on the workings of the program
7. Database schema and a simple UML diagram of the source codes

There is multiple software that we could use to design the user interface. Due to the multiple potential designs that we could use, we have identified a few software that may assist us in developing them:

- Adobe XD
- Figma
- Sketch
- UXPin
- Proto.io

In terms of the hardware specification, we would be using the best laptop from Michael's article (n.d.) where he recommends budget laptops for computer science students. The minimum hardware specifications for the laptop that would be our development platform is as follows:

- CPU: AMD Ryzen 5 Quad-Core up to 3.7GHz
- RAM: 8GB
- GPU: AMD Radeon Vega 8
- Storage: 512GB SSD

- Display: 15.6 inch FHD(1920 x 1080)

3.2 Justification of choices

3.2.1 Choosing Software Artefacts and Components

1. Our team decided to use HTML/CSS for the development of the website. We initially planned to use Django or Flask which are Python-based web development frameworks, however, we thought that those frameworks are too heavy for the system and decided to use simple HTML/CSS instead.

2. The back-end of the system will be based on Python language because there are many libraries and tools available to support the workings of the system.

3. SQL Developer is a database program that will be linked to the front-end website and the back-end Python system. SQL Developer is chosen as it is the basic and the familiar DB system.

4. We will have manual testing scripts which include the sample consumer-level questions and expert-level questions to test out both mobile and web version programs.

5. All the program source codes will be stored inside a GitHub project for easy cooperation between the team members.

6. Database schema and UML diagrams for the programs can be supporting documents for the software documentation on the workings of the program and the diagrams will be helpful in the GitHub project as well to give a better understanding of the structure of the programs which will lead to better co-operation.

3.2.2 User Interface

When identifying and designing a user interface there has to be a set way in which we approach this, Kelden Lin's article extracts a few common UI principles from multiple other sources to help us identify and design a great user interface. Firstly a great UI has to have consistency in the way you design the elements in it. These elements should look the same if their behaviors are also similar (Lin, 2016). The second principle is Familiarity/design disciplines, this essentially means that the UI design can and should follow already existing design disciplines within the platform they are in (Lin, 2016), basically saying that we do not need to always make something new and complex. This makes the user interface easier to use as the user already will be familiar with the design should they have any prior experience with an interface that follows the same design discipline. The third principle is to reduce the cognitive load, this principle fundamentally means the interface design should make up for human’s limited capacity for information processing(Lin, 2016). Grouping together elements that have certain relations as well as making sure that each screen is for a set purpose rather than multiple elements with different roles appearing on the same screen overloading the user with too much information.

Since we are going to be designing a web application, the design we are choosing falls under Graphical User Interface(GUI) which is a type of User Interface(UI) that uses a tactile UI input with a

visual UI output (Indeed Editorial Team, 2021). Each UI has its own advantages as well as disadvantages mentioned in the article written by Alan blog(n.d) but GUI is the most suitable one for our implementation.

Some of the Advantages of GUI are(taken from Alan Blog(n.d.):

- that it is suitable for non-technical users.
- The complexity of actions is hidden from users.
- Immediate visual feedback
- Enhanced by attractive visual
- Enables the use of multiple input devices such as keyboard, mouse, etc…

Some of its disadvantages are(taken from Alan Blog(n.d.):

- That it requires power and memory resources
- Can overwhelm users with the growing amount of control elements
- Might have low discoverability.
- Hidden commands need to be searched for.

As you can see, the advantages of us choosing this User interface, Graphical User Interface, clearly outweigh the disadvantages when we take into account the principles mentioned before.

3.2.3 Visual Studio Code

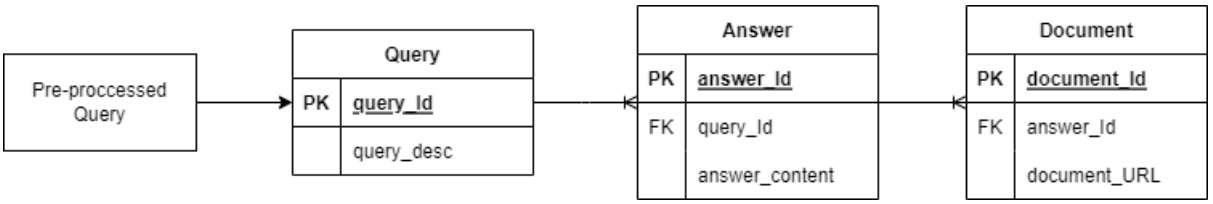
The development platform that our group would be using is Visual Studio Code. Visual Studio (VS) Code supports hundreds of languages, and for our project, we might need to use a few languages. VS Code provides syntax highlighting, auto-indentation, bracket matching, and more. It is best to use VS code because there is a built-in interactive debugger that can step through our code, inspect variables, and even execute commands in the console. VS Code also includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring. Users are allowed to install third-party extensions that can suit our unique needs. VS Code has support for Git, so we can work with source control without leaving the editor. It also includes superb web technologies such as HTML, CSS, and JSON. In our project, we would require to create our own website and a mobile app hence by using the features provided by VS Code, we would be able to create our own website.

4.0 Data

4.1 Data Processing

When the user asks a question using the system, the user must specify the level of answers he/she wants for later use in the database.

This is a simple diagram that shows what will be stored in the database and what processes the database will go through. So, the answers to the questions will be stored in the database and the system will use the stored answers to respond to the users. Every ‘Answer’ entity will be linked with ‘Document’ entities as the references of the answer.



When the user inputs a question, the program will pre-process the question using algorithms and will make the question into a specific query that is already stored in the database. For example, if the user asks 'What is the number of COVID-19 positive patients?', the program must pre-process the question into a short system-friendly query like 'covid patients number'. Then, the program will send the query (or query ID directly) to the database to find an 'Answer' that has the matching query.

4.2 Data Processing Components and Data Flow

Based on an article called "What is Data Processing?" (n.d.) and Duggal's article (2022), our team managed to identify the different data processing components of our software product. Consequently, we were able to identify the data flow across our software product. The first data processing component is data collection. This occurs during the document collection phase, where scientific and government articles about anything related to the coronavirus, and community response information is collected. The articles included are research articles and consumer articles. Besides that, human-generated answers and sentence-level annotations are collected for the judgement process. Furthermore, two sets of questions, in the form of expert-level and consumer-level questions will be collected. The next component is the processing of data. The questions collected will be run on an evaluation script and an answer will be generated after two evaluation rounds - the answer-key generation and the passage annotation round. The next data processing component would be the data output or interpretation stage. This is where the data is finally delivered to the target users.

5.0 Software Quality

In order to ensure that the software product that we will develop is up to standard, we will be adhering to the ISO/IEC 25010 software quality model (ISO 25000, n.d.). We believe that through software quality assurance, risk factors may be better mitigated and handled. Besides that, we believe that properly defining software quality will help us work towards the scope and requirements of the software product, ensuring that we meet the stated and implied customer needs.

Firstly, in terms of the backend, we identified functional suitability as the most important characteristic. We would have to ensure that upon deployment, our software product meets functional completeness, correctness, and appropriateness. Essentially, our software product would have to meet all functional and non-functional requirements. We can consider our software product as fully functionally suitable as long as it is able to perform its functions correctly and appropriately while staying within scope. Another factor that we highly considered is the performance efficiency of our software product. As our software product is essentially a chatbot, we have to ensure that it has a large capacity, while utilizing as few resources as possible, while also having a short response and processing time. There is a low likelihood that our users would want to use slow and laggy software, especially when wanting their queries answered quickly.

Besides that, our software product would have to be compatible, by exchanging information with other systems and components. Our software product will have a high emphasis on interoperability, where we would have to ensure two components would be able to exchange and use information. For example, our software product will probably be using multiple software artefacts, such as Django or Flask on the front end to communicate with the users. Then, the software would have to fetch data from a data source and use the information provided. If our software product is not interoperable, communication between the components would fail and crash the software. Another software quality characteristic that is important to our system product would be its reliability. As it is a chatbot, we will have to ensure its availability will not be affected by any condition. Our software product has to essentially be functional at all times. Besides that, the recoverability of our software product will significantly affect its users. If our software product was to be interrupted or fail, we have to ensure that it can be quickly restored and re-established to its functional state. If not, users who would like to make an inquiry would not be able to do so. Furthermore, our software product has to be portable, in terms of being transferred from one usage environment to another. Essentially, our software product has to have adaptability and installability, by being able to be successfully installed in different environments and on different hardware. For example, our software product has to be functional on different devices such as Android, Apple, or PC, while also being able to be installed from different platform markets such as the Google Play Store or the App Store.

On the other hand, we would ensure the quality of the front end of our software product through the usability characteristic. The user interface and user experience are pivotal to our software product, as it heavily relies on interaction with the users to achieve its basic functionality. Therefore, we have to ensure that our software product is recognizable and appropriate for the user's needs. This can be achieved by designing pleasing and satisfying user interface aesthetics, which are simple, learnable, and operable to anyone of any age. Furthermore, user error protection methods such as smart defaults and warnings will ensure that the user has a smooth experience when using our software product (Bowman, n.d.).

6.0 References

Bowman, J. (n.d.). *How Designers Can Prevent User Errors*. UX Tools. https://uxtools.co/blog/how-designers-can-prevent-user-errors/

Code.visualstudio.com. (2022). *Why Visual Studio Code?*. https://code.visualstudio.com/docs/editor/whyvscode#:~:text=For%20serious%20coding%2C%20you'll,tough%2C%20the%20tough%20get%20debugging.

Duggal, N (2022, March 3). *What Is Data Processing: Cycle, Types, Methods, Steps and Examples*. Simplilearn. https://www.simplilearn.com/what-is-data-processing-article

Indeed Editorial Team(2021,September 18) *What Is a User Interface(Definition,Types and Examples)* https://www.indeed.com/career-advice/career-development/user-interface

ISO/IEC 25010. (n.d.). ISO25000. https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?start=3

Lin, K.(2016, Oct 19) *Identifying great User Interfaces. https://medium.theuxblog.com/identifying-great-user-interfaces-1a34545ef70c*

Michael, P. (n.d.). *7 Best Budget Laptops for Computer Science Students | 2022*. Media Peanut. https://mediapeanut.com/best-budget-laptops-for-computer-science-students/

Types of User Interface. (n.d.) Alan Blog. https://alan.app/blog/types-of-user-interface/#graphicaluserinterface

What is Data Processing? (n.d.). Talend. https://www.talend.com/resources/what-is-data-processing/