

# Project: Medical Decision Assistant (QUICK+)

## Project Summary

Our project aim is to create a website that assists the user with making medical decisions by listing out possible diseases they might have and their likelihood based on the user's input symptoms, suggesting which type of doctor to consult, and estimating the consultant fee based on user's location and the type of doctor they will be consulting. The website will also display information about the commercial medicines associated with the disease, including their descriptions, substitutes, and side effects.

As for additional features, the user can also directly search for information about the diseases to display their symptoms, specialized doctors, consultant fees, and drug information. Specific drugs can also be directly searched up to display their information and the diseases they are associated with.

## Project Description

The purpose of this project is to assist the user in deciding which possible disease they might have, who to consult, and which drugs might help with the symptoms if they are severe enough to require immediate intervention. The web app will give more statistically supported suggestions than simply googling the symptoms, alleviating users' worries about the symptoms. These suggestions may also be useful when consulting doctors. Consulting doctors may be costly, so the app will give the user the estimated consultant fee of the doctor specialized in the disease they're suspected of having so that they can prepare for the fee or find cheaper alternatives, like consulting cheaper doctors.

## Usefulness

When people notice that they have a medical symptom, they can become very worried about what disease they might have. Googling symptoms up, or as they call consulting "Dr. Google", can result in outlandish diagnoses, such as diagnosing the user with a brain tumor for having a headache, which may cause even more worries, leading to other problems such as mental health issue and affecting work performance. By using actual data about diseases and their respective symptoms, the webapp can give the user a more statistically backed diagnosis of which diseases they are likely to have and how likely they are to have it, so that they can decide better whether they should be worried about the symptoms or not.

If they decide that they should look more into the symptoms and want to consult doctors, the consultant fees can sometimes be unexpectedly high. By having an estimation of the consultant fee for the type of doctor they are going to consult in their area, they will know the price range they are expected to pay and decide whether to proceed with the consultation or to find cheaper alternatives.

A product similar to this is WebMD's symptom checker. Compared to this webapp, WebMD will have a more sophisticated symptom questioning diagnosis algorithm, with actual medical professionals behind

the website and a regularly maintained and updated database, resulting in more accurate and digestible search results, including well-written treatment instructions. Furthermore, WebMD has many more features than our webapp, including the ability to get in contact with doctors directly as opposed to just estimating the price. The two features that this webapp has that WebMD doesn't is the ability to estimate a doctor's consultant fees before getting in contact with them and listing out commercial drugs instead of just a generic description of the drug type.

## Creative Component

We want to add a natural language processing feature so that instead of searching for raw predefined symptoms from our database, a user can write something like "I cant breathe in one side of my nose" and the product can parse that into the most likely symptoms and automatically add them to the list of symptoms. This can be achieved if we use chatGPT API keys so we have NLP and we can learn how to do this by following youtube tutorials online.

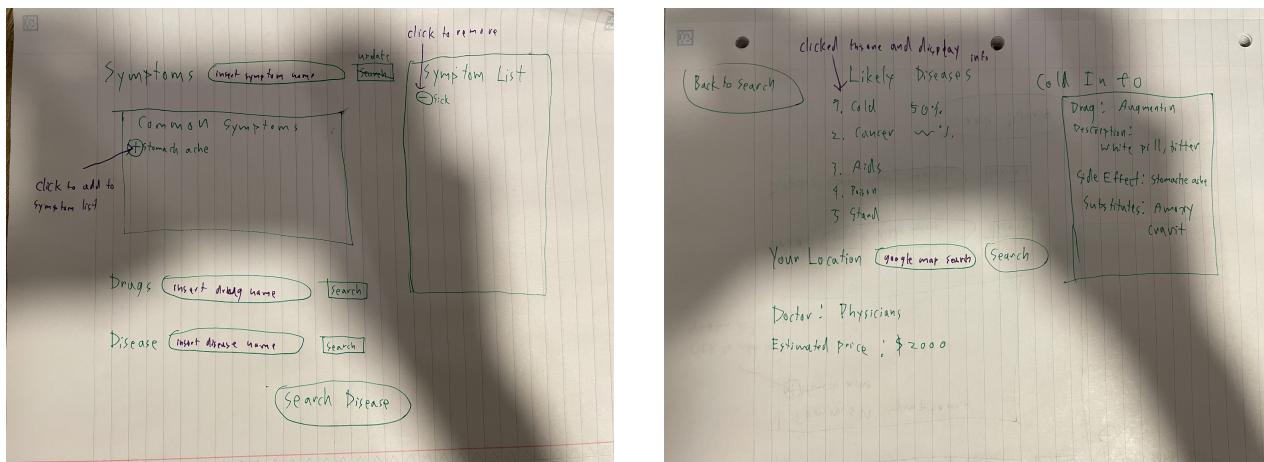
## Data Stored and 'Realness'

1. User data (self-created): This is the database that will keep the user data. This will include the primary key of patientID, Name, Age, sex, etc. The data will be collected by the manual inputs from the patient when they use the service.
2. Symptoms -> Disease (<https://www.kaggle.com/datasets/dhivyeshrk/diseases-and-symptoms-dataset>): This is the data source that links diseases with their symptoms. There are 773 diseases and 377 Symptoms that are one-hot encoded. Using this symptom -> disease database, we can get the user inputs and match the symptoms with the disease to diagnose the patient. Data type: csv.
3. Doctor consultant fee (2020), "Predict A Doctor's Consultation Fee"(<https://www.kaggle.com/datasets/sureshmeacad/predict-a-doctors-consultation-fee/data>): This is the data that keeps track of the consultant fee for doctors for features such as qualification, place, and specialty. This data will be used to give an estimate of the total fee. Data type: xls.
4. Disease -> Type of doctor (self-created): This is the data set to find out which doctor to go to for the diagnosed disease. This will be done manually since it is simple data labeling and we can find out which disease falls under which category. We can map all 723 diseases to their respective doctor type by searching the disease. Data type: csv.
5. Possible drugs for each disease "11000 Medicine details" (<https://www.kaggle.com/datasets/singhnavjot2062001/11000-medicine-details>): This is the dataset needed to find which medicine is prescribed to treat the following disease. This would be a one-to-many relationship and will be handy for the prescription of medicine for certain conditions. Data type: csv.
6. Medicine description "Medicine\_Descriptions"([https://www.kaggle.com/datasets/mohitkundu1/medicine-descriptions?select=TA\\_SK.xlsx](https://www.kaggle.com/datasets/mohitkundu1/medicine-descriptions?select=TA_SK.xlsx)) : This dataset is a text description of what each medicine does. Since medicine names are abstract and are unfamiliar to patients, this data can be displayed as a note explaining the effects of the medicine. Data type: csv.
7. Medicine Substitution and side effects "250k Medicines Usage, Side Effects, and Substitutes"(<https://www.kaggle.com/datasets/shudhanshusingh/250k-medicines-usage-side-effects-and-substitutes>): Data set includes possible side effects and possible substitution for each medicine. This will be displayed with the medicine description as the users get the result of which medicine they need for their symptoms. Data type: csv.

# Functionality

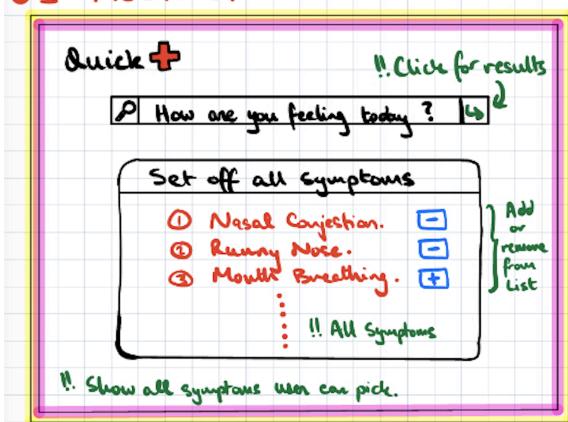
The application will have a search bar where the user will input their symptom. After the text is inputted, a list of matching symptoms will show up, where the user can click on one of them to turn their inputted text into the name of that symptom then click the “update button next to the search bar to add the symptom to the list of symptom displayed. Next to the added symptom in the symptom list will be a minus sign that the user can click to remove the symptom from the list. There will also be a box displaying common symptoms associated with symptoms the user has inputted where the user can click the plus sign next to each entry to add them to their symptom list. Once they have finished adding symptoms, they can click the “Search Disease” button at the bottom to move on to the search result information page. There will also be search bars for drugs and diseases in case the user wants to search for information about them from the database.

## Low Fidelity UI Mockup -

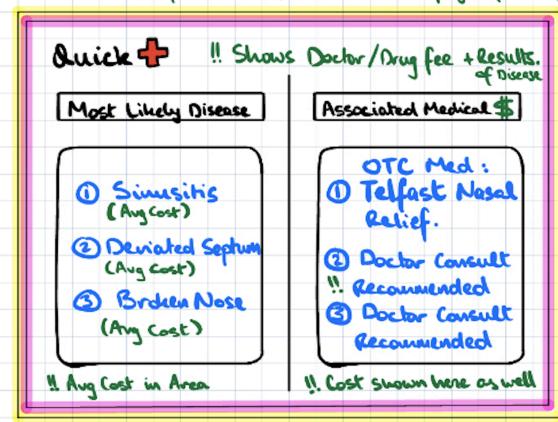


On the search result page, there will be a list of likely diseases the user may have and their likelihood displayed (may or may not be a number) in the middle. Once a disease on that list is clicked, the information associated with that disease will appear in the form of a drug information box on the right and a doctor and estimated consulting price on the bottom. The application will access the user's location for estimating the price in the area. There will be a search bar above the displayed doctor-type result for changing the user's location. Lastly, the “Back to Search” button on the top left can be clicked to go back to the search page.

## UI MOCK-UP



!! Once user picks symptoms / second page after algo.



# Work Distribution

The front end will be handled by Shaurya. This includes:

- Creating a product with similar design to UI mockup using Javascript framework React
- Conduct testing to identify bugs and ensure the frontend works seamlessly
- Help with other members on whatever they need help with and ensuring that their implementations work seamlessly with the frontend

The back end will be worked on by Max and Ian. This includes:

- Setting up the webapp
- Integrating the database into the webapp
- Develop and implement algorithm for estimating consultant fee
- Develop and implement algorithm for suggesting likely diseases the user is likely to have
- Develop and implement a method for taking in user's location for use with estimation of consultant fee, whether that is through Google Map api or through manual input
- Conduct testing to identify bugs and ensure the system works seamlessly

The development of the database will be handled by Nobt. This includes:

- Drawing the UML diagram of the database
- Set up database to store all the relevant data efficiently
- Cleaning up and preprocessing the datasets
- Integrate the processed datasets into the database
- Develop query processes to transfer relevant data to the front end
- Conduct testing to identify bugs and ensure the database works seamlessly