

Team 092 (SHRM) - Detailed Project Description

- **Describe what data is stored in the database. (Where is the data from, and what attributes and information would be stored?)**

The information stored in the database include basic information about users such as age, sex, country of origin, and a unique ID. It also stores information about any previous health conditions they might have and any current medication they use to treat it. It also includes data about any of the symptoms each of these users face as well as environmental factors such as food, weather, etc.

- **What are the basic functions of your web application? (What can users of this website do? Which simple and complex features are there?)**

The basic function of our web application would include users being able to learn more about various illnesses, and symptoms associated with them by entering in any symptoms they might already have. They will be able to learn more about which symptoms are similar to those of others with certain illnesses as well as other information about others such as the treatments they are undergoing, their environment, etc.

- **What would be a good creative component (function) that can improve the functionality of your application? (What is something cool that you want to include? How are you planning to achieve it?)**

- **Wishlist of specific symptoms/illnesses:** There may be some particular symptoms/diseases that users want to learn for themselves or their friends and family but may not be listed in the previously gathered dataset. We will implement a list by inserting a new table to store the users' interests. The collected data will be shown on the webpage for users to acknowledge what others may want to know. The website developer can update the requested information about those sicknesses in the future.
- **Here to warn you:** People may have a lot of bad habits that may affect their health without their awareness. We may implement a checklist for users to fill out with their daily habits. The checklist's options will be gathered from the dataset where patients previously stored the triggers of their illness. After users submit their results, we may list the possible illnesses they may face in the future if they keep on living like that.

- **What is the project title?**

Diagnosify

- **Provide a project summary.**

The project objective is to create a web-based symptom checker application that can suggest potential chronic illnesses based on user-entered symptoms. The application will be powered by a database of patient-reported data from an app that tracks symptoms, conditions, and treatments. The dataset is obtained from Kaggle and contains over 7.98 million records of daily check-ins from users. The app allows users to create their own set of "trackables," which are conditions, symptoms, and treatments, and track them over time.

Our team will perform data cleaning and preprocessing on the dataset to ensure that the data is accurate and usable. We will load the preprocessed data into a SQL database management system for efficient and reliable querying.

The symptom checker application will be designed to allow users to enter their symptoms, and the application will suggest potential chronic illnesses that match the symptoms. The suggested illnesses will be ranked by their likelihood based on the reported symptom severity, duration, and frequency. The application will also allow users to report new symptoms and illnesses, which will be added to the database, further improving the accuracy of the symptom checker.

Overall, this project will demonstrate our team's ability to build a database-backed web application that can handle large volumes of data and provide useful insights to users.

- **Provide the description of an application of your choice. State as clearly as possible what you want to do. What problem do you want to solve, etc.?**

Our application will aim to help users carry out a more enjoyable lifestyle by helping them identify their illnesses and what habits and medications they should partake in. It will store the users' personal information and any information pertaining to their health such as age, sex, weight, previous illnesses, etc.

The application will allow the user to input their information regarding their health as well as other factors that might affect their lifestyle such as their location, any current treatments etc, and symptoms. Using this information, We will be able to better educate users about how they should handle their present situation based on data such as what illnesses they may be facing, and how external factors such as their environment may be affecting it.

- **Explain as clearly as possible why your chosen application is useful. Make sure to answer the following questions: Are there any similar websites/applications out there? If so, what are they, and how is yours different?**

Our application is useful because diagnosing symptoms is something that is traditionally done by either going to a doctor (which can be expensive) or by using a search engine which may lead to inaccuracies. By using a database powered website, we can provide more accurate information to our users and this may help the health of the users (since many illnesses are very time sensitive, where an earlier diagnosis can lead to drastically different results from a late diagnosis).

While there are other symptom checker applications available, such as WebMD's <https://symptoms.webmd.com/>, our app offers a more comprehensive approach. Rather than simply cross-referencing your input with a database of ailments, our application is built on real user data. Additionally, we take into consideration the frequency of certain ailments and other user-specific factors to provide the most accurate possible results.

- **Describe what your data is and where you will get it.**

Our proposed dataset, suggested by TAs, is the Chronic illness dataset. The dataset contains information about the patient's symptoms, treatments and doses, and potential environmental triggers (foods, stress, allergens, etc.). We get our data from the Kaggle dataset. The composition of the given data table is pretty straightforward, including the following categories of information:

1. User basic information: *user_id*, *age*, *sex*, *country*, *check-in date*, *trackable_id* (Unique id of the trackable). This is where we fetch the background information of the patients.
2. Type of the trackable data: *trackable_type*, specifying the data as a symptom, treatment, condition, weather, etc.
3. Name of trackable from the preset database or user entry: *trackable_name*, more detailed descriptions following the previous type.
4. Value of trackable: *trackable_value*. Combining 2,3, and 4, we can get a complete view of the user input.

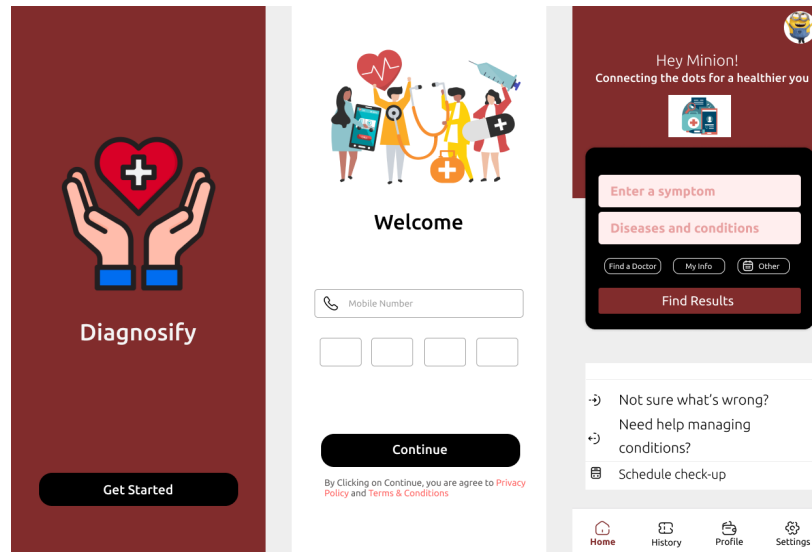
The dataset will require some pre-processing. We may first split up the *trackable_type* part to make the inputting type more organized, and it will be divided into *Symptom*, *Weather*, *Condition*, *Trigger*, and other selected types based on popularity.

As for the additional data set we're planning to create, it may consist of the following information:

1. Users' fundamental data.
2. Users' updates and follow-ups on their previous/ongoing illnesses.
3. Users' interest in specific symptoms or illnesses.

We may demonstrate this by filling in test data by ourselves or gathering data from friends and classmates.

- **Describe the functionality that your website offers. This is where you talk about what the website delivers. Talk about how a user would interact with the application (i.e., things that one could create, delete, update, or search for). Read the requirements for stages 4 and 5 to see what other functionalities you want to provide to the users. You should include the following:**
- **Symptom Checker** - The Symptom Checker is the main feature of the application. Users can enter their symptoms, and the application will suggest potential chronic illnesses that match the symptoms. The suggested illnesses will be ranked by their likelihood based on the reported symptom severity, duration, and frequency.
- **User Profiles** - Users can create profiles to track their symptoms, conditions, and treatments. This feature will allow users to view their check-ins, update their profile information, and delete their account if needed.
- **Data Management** - The data management feature will allow admins to manage user data and track the usage of the application. The feature will include creating, updating, and deleting user profiles, managing the symptom database, and tracking user activity.
- **External Data Integration** - The application can be integrated with external datasets to provide users with additional information about their symptoms and possible chronic illnesses.
 - **A low-fidelity UI mockup: What do you imagine your final application's interface might look like?**



- **The project work distribution: who would be responsible for each task or subtask?**

List of the person responsible for which exact functionalities in section 6. Explain how backend systems will be distributed across members.

- Symptom Checker - Meghna will be responsible for developing the symptom checker feature. This will include designing the algorithms to match symptoms with illnesses and implementing the user interface for the feature.
- User Profiles - Shrestha will be responsible for developing the user profile feature. This will include designing the user interface for profile creation, management, and deletion and implementing the backend logic for the feature.
- Data Management - Hsiang-Yin will be responsible for developing the data management feature. This will include designing the user interface for admin management of user data, developing the database schema, and implementing the backend logic for the feature.
- External Data Integration - Richard will be responsible for integrating external datasets with the application. This will include identifying relevant datasets, designing the user interface to display external data, and implementing the backend logic for the feature.

Backend Systems:

To distribute the backend systems across team members, we can use a monolithic architecture where all the backend systems are integrated into a single application. In this case, we will use a version control system, Git, to collaborate on the codebase and merge changes from different team members.