

Lab Configuration

1. Timeline software – Example, Gantt Chart

Questions to answer for your project

Functional Requirements

System Functionality

1. What will the system do?

The system will allow a medical professional to input a patient's medical diagnosis and any relevant notes or files, allow an AI to interpret it into language that the common person can understand and allow patient symptom tracking and management. The patient or patient's caregiver can view what the doctor has added and allow the patient to ask questions to an AI and allow the patient to track their symptoms.

2. When will the system do it?

- a. The application will be available on demand as a user needs it. It should be available to a patient or caretaker at any time.

3. Are there several modes of operation?

Yes, we have a medical professional mode and a patient/ caregiver mode.

4. What kinds of computations or data transformations must be performed?

Uploading and saving files and text responses. Keeping track of symptoms and converting it into a visual (usually a graph). AI will be used to interpret a medical professionals text in order to make it easier to understand.

5. What are the appropriate reactions to possible stimuli?

The application will not take any physical external stimuli. It will respond to selections, typing, scrolling, etc on a computer.

Data

1. For both input and output, what should be the format of the data?

For input, there will be text input from all user types, as well as input of jpg or pdf files uploaded by the doctor/technician.

Output data would be in text format only.

2. Must any data be retained for any period of time?

Yes, individual users' relevant provided info would be used in features such as symptoms tracking, etc.

3. Design Constraints

Patient confidentiality, suitable for those with various medical conditions and limited experience with technology

Physical Environment

1. Where is the equipment to be located?

On a smartphone or a website.

2. Is there one or several locations?

Many locations since it can be on any device.

3. Are there constraints on size of the system (Handheld/Server/PC etc)?

The system will be operable on a web browser, so any device with the ability to open a web browser will have access to the system. A handheld device or laptop are sufficient.

4. Are there any constraints on programming language, OS because of existing software components?

This is a new project, so there are no existing software components.

Interfaces

1. Is input coming from one or more other systems ("upstream")?

Input will go from the user interface to the database, and input will go from the AI API to the database/user interface.

2. Is output going to one or more other systems ("downstream")?

Whatever module is holding the AI API tool will output into another module

3. What is the protocol for the upstream and downstream systems?

To avoid circular dependencies and document every instance of data handling for robust confidentiality and to prevent bottlenecking of dataflow.

End-Users

1. Who will use the system?

Doctors, patients, caregivers.

2. Will there be several types of users?

Yes, there will be two user types, patient/caregiver and doctor.

3. What is the skill level of each user?

Beginner to intermediate

Quality Requirements

Performance

1. Are there constraints on execution speed, response time or throughput?

The program will need to be usable and not have abnormally long wait times

2. How much data will flow through the system

The data in the system will be equivalent to the number of users multiplied by the average size of the user's associated data

3. How often will data be received or sent?

Data will be received and sent depending on user access of the system. The system will not be continuously sending any data.

Usability and Human Factors

1. What kind of training will be required for each type of user?

There will be some tutorials if someone needs it but the documentation should be sufficient

2. How easy should it be for a user to understand and use the system?

The user interface will be as straightforward as possible due to the wide range of user skill levels. Users with very limited computer skills will require assistance from a caregiver or similar.

Security

1. Must access to the system or information be controlled?

Yes due to HIPPA regulations

2. Should each user's data be isolated from the data of other users?

Yes due to HIPPA regulations

3. Should user programs be isolated from other programs and from the OS?

Yes, there will be a distinct and frontend and distinct backend

4. Address Cyber security issues

HIPPA needs to be kept in this case

Reliability and Availability

1. Must the system detect and isolate faults?

The system must have robust debugging practices in place in the event of perceived faults

2. What is the prescribed Mean Time between Failures?

Out of scope

3. Is there a maximum time allowed for restarting the system after a failure?

24 hours

4. How often will the system be backed up?

Once a day

5. Must backup copies be stored at a different location?

Ideally, yes

Maintainability

1. When and in what ways might the system be changed in the future?

Features targeting usability and efficiency may be added via a continuous integration pipeline.

2. How easy should it be to add features to the system?

Should be relatively easy if CI/TDD practices are employed.

3. How easy should it be to port (or migrate) the system from one platform to another?

Web applications should be versatile in their deployment possibilities.

Gitlab

1. At the end of the project, create an issue on the project and assign it to a member of the project. Named lab3.
2. Create a branch for the issue with the same name as the issue.
3. Add all the documents created so far in the labs.
4. Commit and submit a Pull Request.
5. Approve the PR.