CSN-254

Requirement and specification document

Dhruv Shandilya, 20114034

Deepak Tailor, 20114029

Arya Anand, 20114017

Nikhil, 20114063

Nishchay Narayan B, 20114065

Anmol Kumar Varshney, 20114012

1. **Project abstract**

Our application assigns a unique medical ID number to all our users, which can be used by medical professionals to get immediate access to medical information in case of any mishap. Further building upon this idea, the user can save medical information such as files and this information can be accessed by software which is exclusively present with hospitals. This will reduce the response time doctors need to read the medical reports and help determine what allergies the patient has, so that he/she can be given quick medical attention accordingly.

1. **Document revision history**

Rev. 1.0 date - initial version

1. **Customer**

Our application comprises of users, from particularly two domains: patients and doctors. The application will contain two login options, one for the patients, where they can enter their medical information and one for doctors where they can be verified via their credentials and given access to view these patients' information. The people whom this application can serve can also include people suffering from memory problems who cannot remember the things they are allergic to or what they are suffering from. This application can also serve people who do not speak English, as they can have trouble remembering English names.

1. **Competitive landscape**

Some competitors of our application along with their functionalities are:

* HealthVault
  + - * Web, Windows, iPhone
      * Store and share medical information
      * Helps maintain a record of user’s health such as weight and so on.
* MTBC PHR
  + - * Android
      * Tracks and manages the patient's personal information.
      * Stores reports of medical information.
      * Features secure messaging.
      * Has an “auto check-in" feature.
* MyPHR
  + - * Tracks, collects and shares medical information.
      * Information can be accessed by doctors.
* CapzulePHR
  + - * iPhone and iOS devices.
      * Tracks, collects and shares medical information.
      * Tracks the progress of health goals of the users.
      * Data on this can be transferred to other devices having this application over Wi-Fi.
      * Allows sharing of data and graphs with the user’s physician.
* GenexEHR
  + - * Android and iPhone.
      * Tracks, collects and shares medical information in the cloud.

Given above are the major players in the medical application industry.

Some strengths of the above applications are:

1) They are large scale projects and involved tech-based companies.

2) CapzulePHR stands out from the rest of these applications as it allows the sharing of graphs with the doctor, and it also tracks the personal goals of the user.

3) GenexEHR stores information on the cloud. This protects the user from data loss.

Some weaknesses of the above applications are:

1) They have the same fundamental idea and simply serve as storage applications and can be easily replaced by each other.

2) GenexEHR stores information on the cloud. This is a double-edged sword as the user might need to access information without an internet connection or suppose some mishap happened in a remote location.

3) They are generally not available across all platforms. Some are restricted to iOS devices exclusively, and some on the other hand are strictly web-based which is a tight spot to be in, in case of lack of internet services.

Points that differentiate our application from the competitors:

1) The competing applications only serve as mere storage devices. On the other hand, our applications has a closer semblance to an in case of emergency contact, in the sense that when a mishap occurs the medical professionals can access the patient’s medical information.

2) There is a possibility of expansion in our applications which might include an in-app scanner for pdfs, a messaging app, and a map which can show doctors/hospitals who use this app.

3) There will be obvious encryption for accessing the user medical information by third party who do not have the doctor’s registered software.

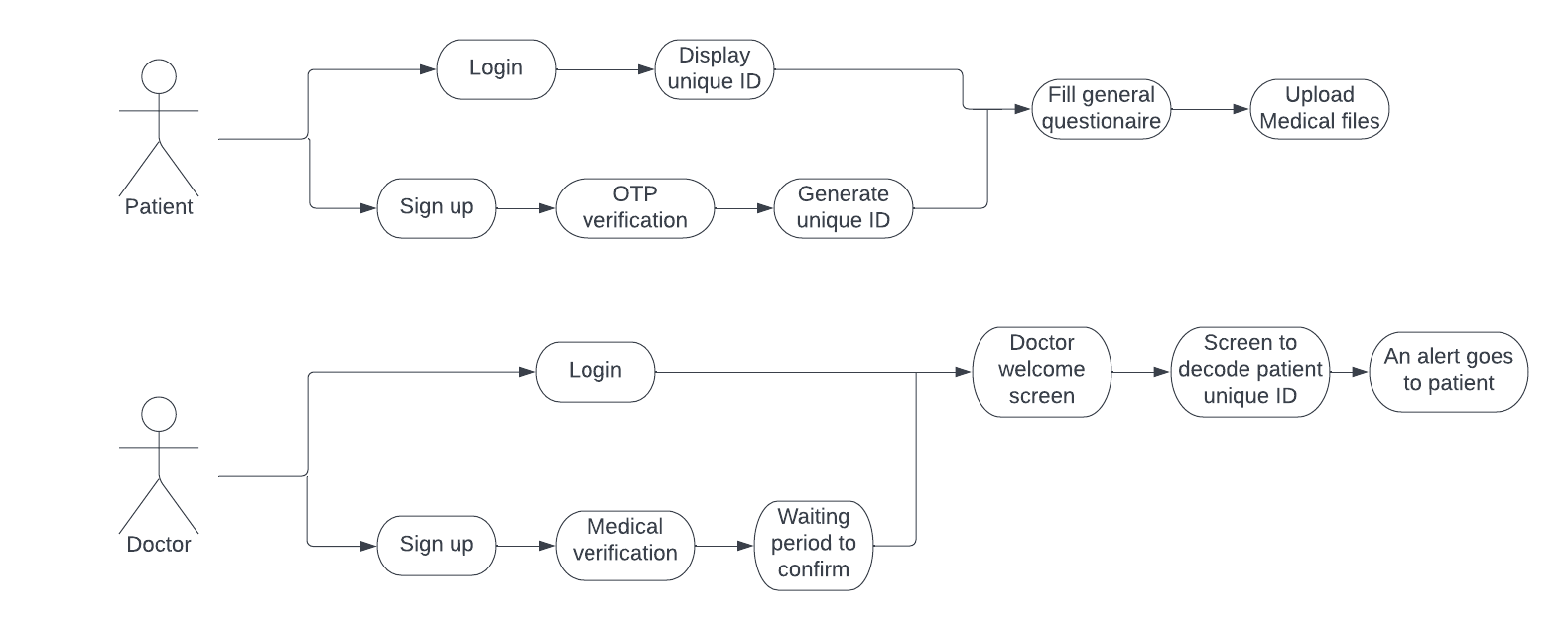
1. **System requirements**

* Functional requirements:

Login and sign-up page for the users. There must be separate login options for patients and doctors as the functionality provided varies. After this there should be a display screen of unique ID of the patient so that the patient can see it. After sign up option for patients there should be an OTP verification screen. This should be followed by generating a unique ID for the patient’s medical information which will be displayed on screen too. Then the patient will be asked to fill or update a general questionnaire about general information like their allergy to specific drug and so on and so forth. Finally, after this the patient can upload their medical files onto the application which can be accessed by the doctor in case of emergency.

For the doctor, login will be followed by a welcome page for the doctor. The signup page will be followed by medical verification of the doctor and checking their credentials as a medical practitioner. Finally, this will be followed by a waiting period to verify those credentials and to verify their practitioner license. After the completion of this process, a doctor welcome screen will be displayed, following which doctors can decode the patient's unique ID to access their medical information. Once this is done an alert will go to the patient to curb malpractice or abuse of user’s medical data.

* Use case Diagram:



* Nonfunctional requirements:

Timing constraint:

The complete project is expected to be completed in 4-5 weeks. We are using the iterative waterfall method, which is an iterative model, we will be delivering the model once for beta testing in 4 weeks. Then everything is integrated and deployed.

Security and Privacy issues:

We are making use of firebase to store the user data and we aim to use encryption to provide the user with security regarding their files, as this is very sensitive information.

Service availability:

Since our development process is done on flutter, we plan to deploy our application on Android, iOS and we aim to create a web application too to deploy the same on the web. This makes sure that we are available to our users on any platform of their choosing.

Software quality attributes:

Our software is designed to look sleek and user friendly and we’ll adopt new design practices, and it will have a graphical user interface, so our users will have ease of use.

hardware requirements:

Depending on how to generate user ID and whether it is a number or a QR code our user would need a device equipped with a camera to scan the QR code. On the other hand, if it is a number then it can easily be done via web application too.

Performance requirements:

We aim to minimize the response time of our application via effective and efficient code. The code will be quite responsive as it needs to extract data from cloud which can be done asynchronously.

1. **Checklist**

Have all the viewpoints of all the stakeholders been considered?

**-> Yes,** most viewpoints have been covered.

Do the requirements completely specify what the function of the program should be?

**-> Yes.**

Are all non-functional requirements (e.g. speed, memory, capacity) specified?

**-> No,** they have not been specified yet because it is predominantly cache and memory friendly.

If there are requirements imposed by the environment in which the program will be used, are those requirements specified?

**-> No,** there would not be any specific requirement as such.

Do the requirements avoid specifying a solution or a design?

-> **Yes**

Are separate requirements listed separately and not lumped together?

-> **Yes**

Are the requirements clear and precise, not ambiguous?

-> **Yes**

Are the requirements consistent, no contradictions?

-> **Yes**

Are the requirements accompanied by a rationale or justification?

-> **Yes**

Are the requirements given in a consistent format?

-> **Yes**

Are the requirements properly prioritized?

-> **No**

Are graphical and mathematical notations used appropriately?

-> **Yes**, graphical notations are provided in the form of UML.

Is the technical jargon kept to a minimum?

-> **Yes,** this represents the idea in an abstract mannerism.

Are the requirements realistic?

-> **Yes**, the requirements are realistic.