CSN 254

Design document

Dhruv Shandilya, 20114034

Arya Anand, 20114017

Anmol Kumar Varshney, 20114012

Deepak Tailor, 20114029

Nikhil, 20114063

Nishchay Narayan B., 20114065

1. Summary

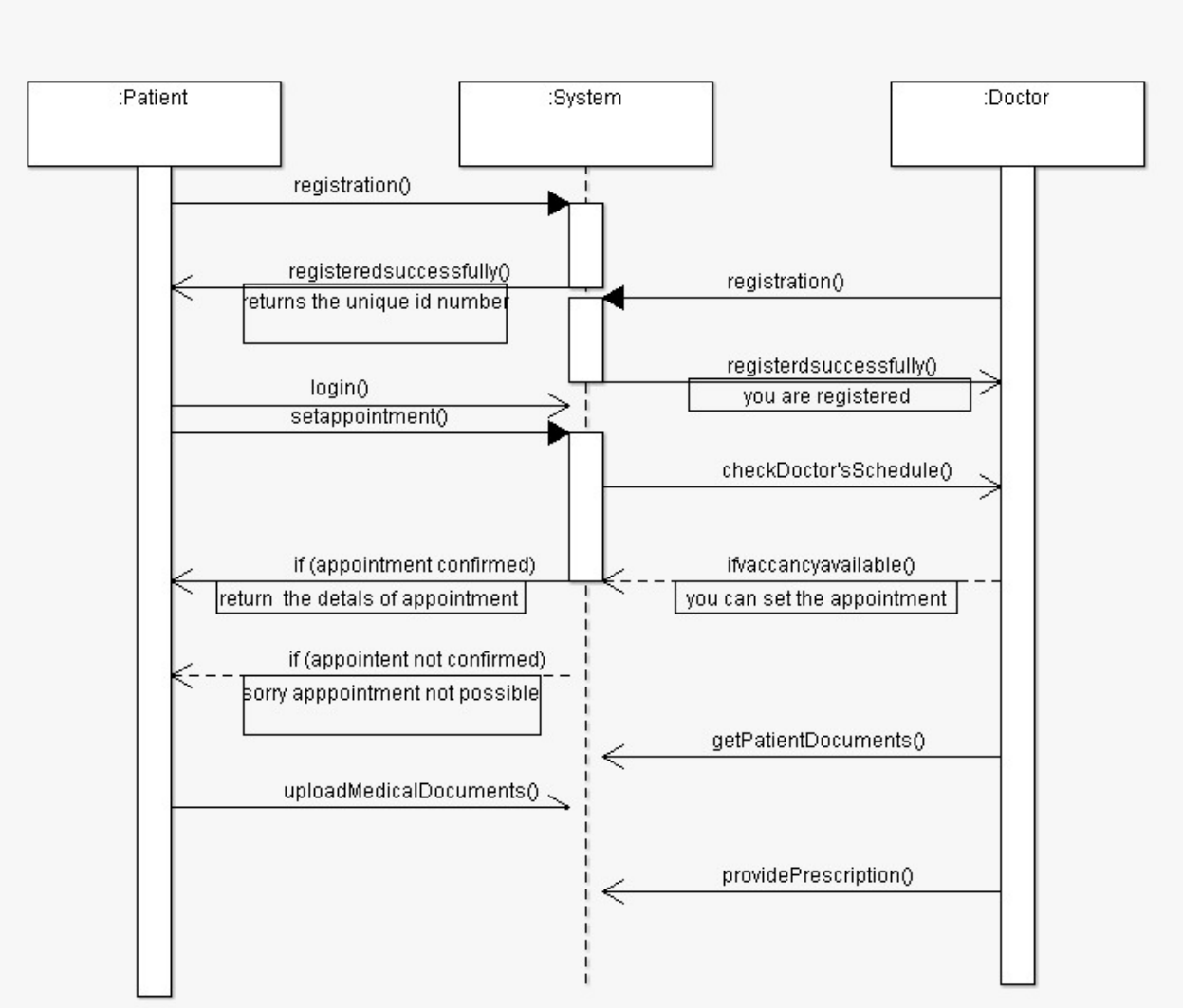
This design document has been developed to explain the design and workflow implementation of key features of Medical Emergency App. The primary objective that this design document aims to accomplish is to increase operational efficiency and to provide a sustainable base and framework around which the coding phase can be structured. It details both the logical and physical design considerations related to all components of our application.

This document identifies the points of contact of the project, lists implementation requirements, provides a brief description of all the deliverables and provides an overview of the implementation process for the project.

This guide is written for a web developer or programmer who is familiar with basic web development terminologies and applications. The scope of this document is specific to the implementation of the Medical Emergency App and its application for user files as defined later in the document.

The workflow of the user interaction uses cases and the deliverables involved have been explained in details for the programmer by making use of well-developed use case diagrams. The user workflow and interaction with the website frontend and its response by the application backend has been detailed further by employing a flow chart diagram, which will help in tackling any logical issues that arise with the working of the website in the coding phase.

The developer can also find a well-documented class diagram in this design document, which will prove to be very handy while coding up the entire project and if referenced properly at the right intervals, the documentation of the class diagram can provide useful insights.



Sequence diagram explanation

• Patient will register in the system

• The system will generate a unique ID for this patient

• If the patient is already registered then he needs to login

• Same way any doctor or nurse needs to register in the system

• Patient has to upload his or her previous medical history on the system

• Patient can ask for an appointment to get registered doctor on the system

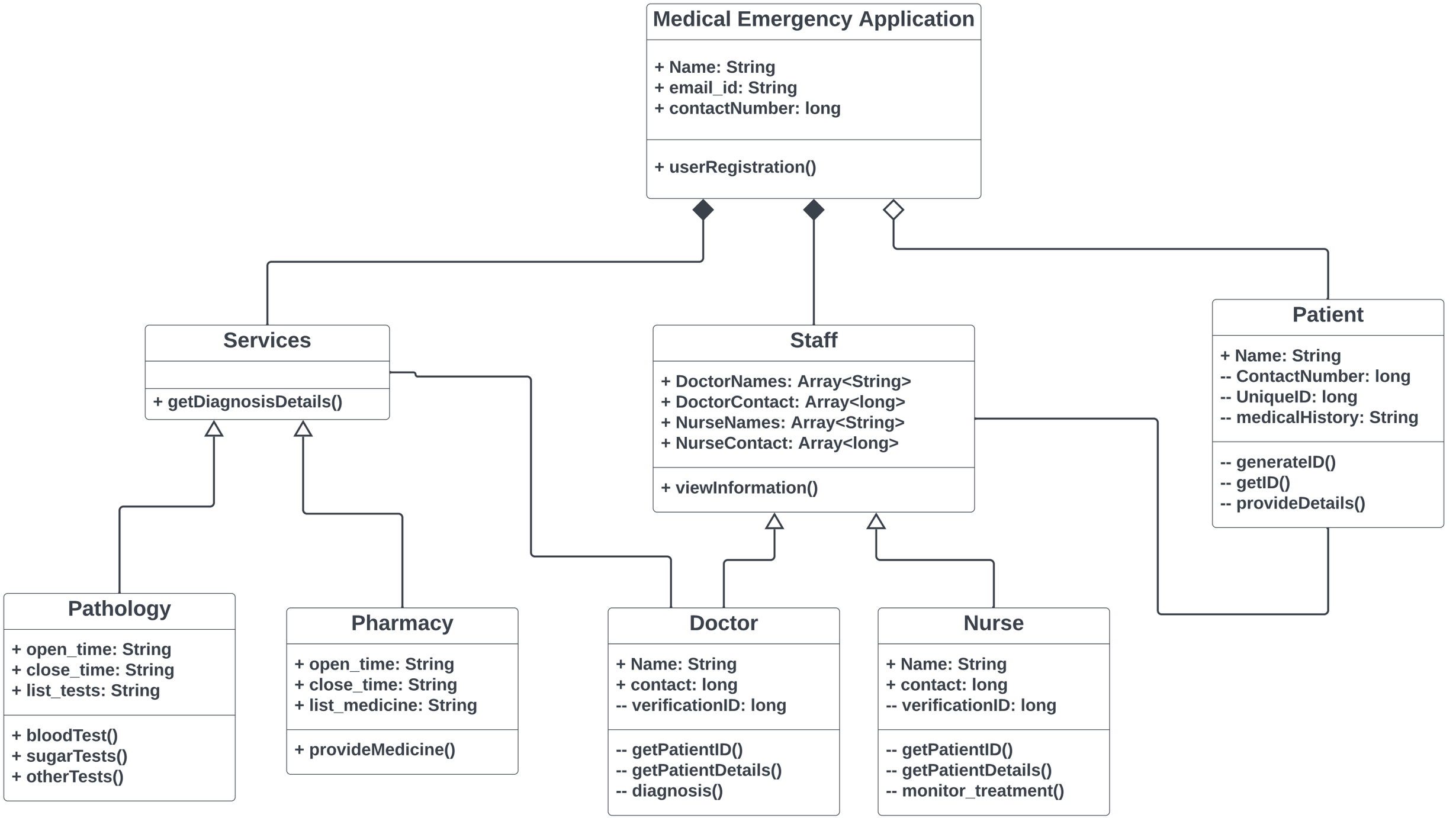
• The system will check the doctor concerned doctor’s appointment vacancies and then return a message to the patient

• If appointment is available then that appointment details are returned to the patient

• If no vacancy is available in the schedule then the system returns that there is no was available Slot

• Dr can get access to the previous medical history of the patient on the system

• Dr can upload the prescription on the system so that the patient can see it at any time


Medical Emergency Application
+ Name: String
+ email_id: String
+ contactNumber: long
+ userRegistration()
Services
+ getDiagnosisDetails()
Staff
+ DoctorNames: Array<String>
+ DoctorContact: Array<long>
+ NurseNames: Array<String>
+ NurseContact: Array<long>
+ viewInformation()
Patient
+ Name: String
-- ContactNumber: long
-- UniqueID: long
-- medicalHistory: String
-- generateID()
-- getID()
-- provideDetails()
Pathology
+ open_time: String
+ close_time: String
+ list_tests: String
+ bloodTest()
+ sugarTests()
+ otherTests()
Pharmacy
+ open_time: String
+ close_time: String
+ list_medicine: String
+ provideMedicine()
Doctor
+ Name: String
+ contact: long
-- verificationID: long
-- getPatientID()
-- getPatientDetails()
-- diagnosis()
Nurse
+ Name: String
+ contact: long
-- verificationID: long
-- getPatientID()
-- getPatientDetails()
-- monitor_treatment()

Class diagram

Our application will be used in medical emergency situation

Below are the class diagram explanation

Medical emergency class:

Name, email id , phone number are the attributes of the class.

Name is of string data type.

Email id is of string data type

Phone number is of long data type

Function of this class is registration.

Services class:

Function of this class it to get the diagnosis details of the patient. Also this class has a relationship of composition with the class, Medical Emergency Application (parent class).

Staff class:

This class consists of the attributes : doctors name, doctor contact no. ,nurse names, nurse contact. Also this class has a relationship of composition with the class, Medical Emergency Application (parent class).

Doctors name and nurse name are of Array<string> data type

And their contact details are of array<long> data type.

Function of this class is to view the information of the patient.

Patient class:

This class consists of attributes: contact no of contact number of patient, his/her unique id, and medical history.

Contact no and unique id are of long data type.

And medical history is of string data type.

Function of this class is to generate the unique id and send id. And to provide details of the patient to staff. Also this class has a relationship of aggregation with the class, Medical Emergency Application (parent class) and this class has a relationship of association with the class, Staff.

Pathology class:

This class consists of the attributes: open and closing time and list of the tests.

All the three attributes are of string data types.

Function of this class is to perform blood test, sugar test, and other medical tests. Also this class has a relationship of generalization with the class, Services.

Pharmacy class:

This class consists of attributes: opening and closing time and list of the medicines available.

All the three attributes are of string type.

Function of this class is to provide the required medicine. Also this class has a relationship of generalization with the class, Services.

Doctor class:

This class consists of attributes: name, contact no, and verification id.

Name is of string data type.

And contact no and verification id are of long data type.

Function of this class to get the id of patient, get his/her medical history and give diagnosis. Also this class has a relationship of generalization with the class, Staff and association relationship with Services class.

Nurse class:

This class consists of attributes: name, contact no, and verification id.

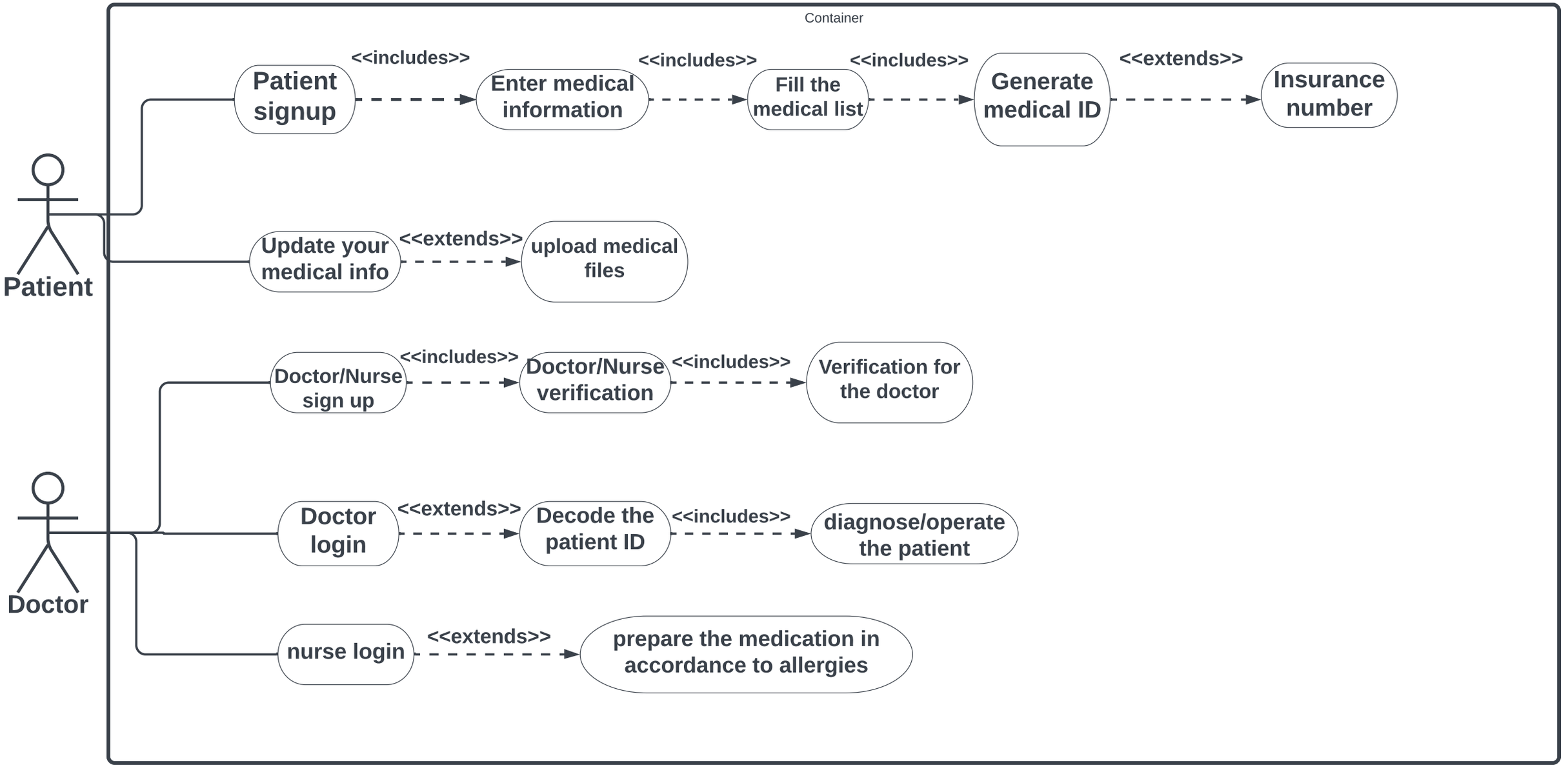
Name is of string data type.

And contact no and verification id are of long data type.

Function of this class to get the id of patient, get his/her medical history and to monitor the treatment of patient. Also this class has a relationship of generalization with the class, Staff.

Account ownership flow

ANMOL KUMAR VARSHNEY | March 29, 2022



USE CASE DIAGRAM

ACTORS:

- Patient
- Doctor

Use Cases:

1.Patient->

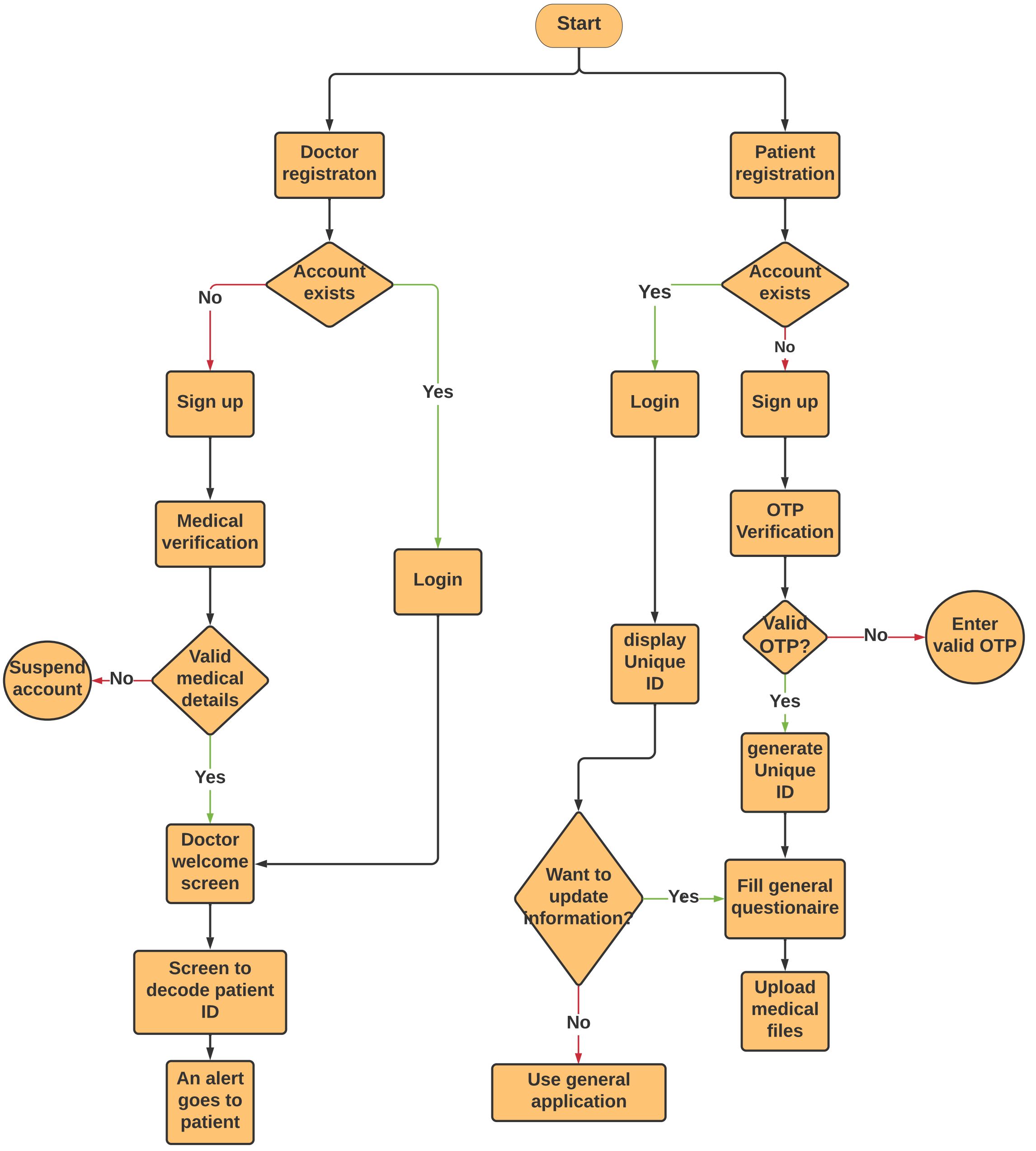
- Patient SignUp-Patient will use this function to SignUp so that he can use our App.
- Enter Medical Information-He will provide all his Biodata information.
- Fill the Medical List-Patient will fill Medical information so that it could be used by Doctors.
- Generate medical ID-The System will provide the patient with a medical ID which can be used to operate in our app.
- Update your medical Information- This function helps Patient to update all his medical information.
- Upload Medical Files-Here patient will provide its All Test Reports.

2.Doctor->

- Doctor/Nurse Signup- Doctor/Nurse will be able to Signup through this function.
- Doctor /Nurse Verification- Here they will provide their Practitioner ID.
- Doctor Login- Separate Login Portal for Doctors.
- Decode the Patient ID-Here Doctors will be able to check patient status from their ID.
- Operate on Patient-From here Doctors will get permission to Operate on Patient.
- Nurse Login- Separate Login Portal for Nurse.
- Prepare medicines/Reports-Nurse will prepare medical report for patient and will provide this to patient so that he can up
  ad it on the system.

Flowchart

Dhruv Shandilya | March 29, 2022



Main design part:

a) First the application begins the registration procedure for the users. In this the application accepts User attributes like name and so on.
b) Further depending on whether the user is a doctor or a patient, there are separate sign-up pages accordingly.
c) A verification procedure will be conducted for the doctor by our team. In this procedure we will look at the history of the doctor and his practicing license. This is done to curb malpractices.
d) If the user is a patient, then a registration is done involving OTP to verify the user. For this we will use firebase to store the user information and we will generate a unique ID using the random class in dart.
e) This random number is stored in the database and whenever a new number is generated, we will comb through the already existing unique IDs to make sure there is no overlap.
f) We will allow the users to upload pdf files and we will store the same over firebase using dart. We will use a map to map the unique ID to the patient details such as the name, medical information and so on.
g) Once this procedure is completed, the patient needs to fill a medical form stating the patient's allergies and chronic diseases the patient suffers from.
h) To store this information, we will create Boolean variables in dart which become true when a given option is selected. We will also allow the user to enter any other chronic disorders if they are not covered in the MCQs and this will be stored in a String datatype.
i) We will define a method which gives access to the medical information of the patient. The staff can enter the Unique ID on their screen and a request will be sent to the server to get information mapped to this specific key.
j) Doing this generates a message sent to the patient's contact using which they can verify who accessed their medical information. This is a security measure to prevent any mishaps.

2. Project Description

- Medical Emergency The app collects information about medical history of patient and ranks the medical reports on the basis of their seriousness.
- It builds a connection between a doctor and patient by which all the medical reports can be accessed by the registered doctor or any medical institution.
- For serious diseases like heart , diabetes, Blood Pressure related problems , lung diseases, liver diseases, kidney diseases, cancer, fatal viral and bacterial diseases the app reminds the patient for regular checkups within a specific time period
- The app automatically highlights the allergies that a patient has for example allergy specific chemical compounds used in making medicines.
- A lot of people suffer because it takes a very long time to collect various details about medical history.
- This app contains all the reports which can be easily accessed by medical officers in case of emergency cases like accidents, silent cardiac attacks, asthma etc.
- In case of accidents due to blood loss we require blood of particular blood group and it requires up to 15 – 20 blood group test.
- The app by facial recognition/fingerprint recognition gives the details about the patient like his name, age, gender, next of kin and their contact details.
- A unique id number will be provided to the user through which he/she can access the medical information at any time.
- Software highlights all the allergies the patient is suffering what medication is to be avoided in case of a particular disease.
- The registered doctors can access all the medical details of the patient. They don’t need to conduct test for everything.