**Remote Patient Monitoring System (RPMS)**

**Project Report**  
**Course:** CS-212 Object-Oriented Programming  
**Semester:** Spring 2025  
**Class:** BSAI-1A  
**Instructor:** Nazia Perwaiz

**Group Members:**

Alap Gohar – 502082

Abdullah Farooq – 509528

Huma Ejaz -

**1. Project Overview**

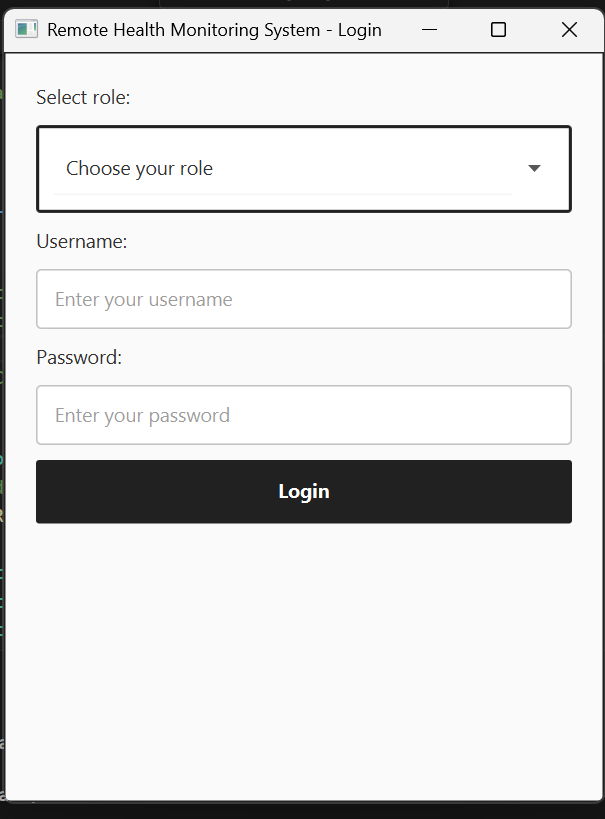
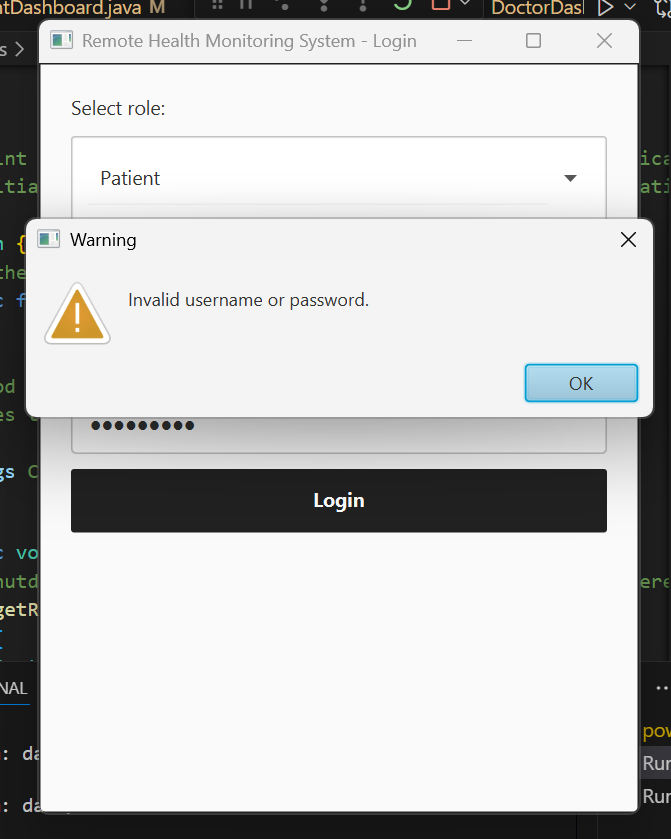
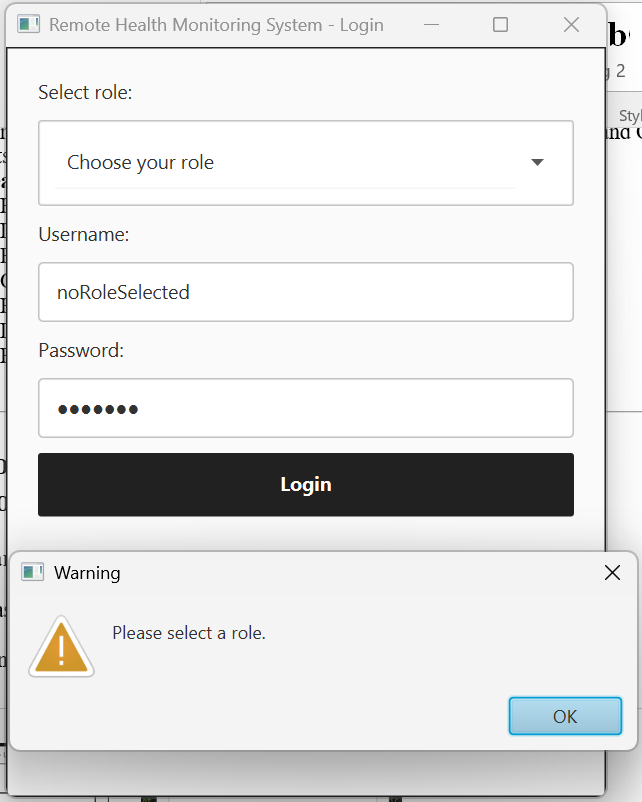
* **Objective:**  
  The goal of RPMS is to enable remote tracking of patient vitals and facilitate communication between patients and doctors using Java-based technologies and OOP concepts.
* **Key Features:**
  + Health Data Upload via CSV
  + Doctor Feedback & Medication
  + Emergency Alert System
  + Chat/Video Consultation
  + Health Report Generation
  + Data Visualization
  + Role-Based User Management
  + Object Serialization/Deserialization for Storing/Loading data

**2. Functional Requirements Implementation (GUI Screenshots)**

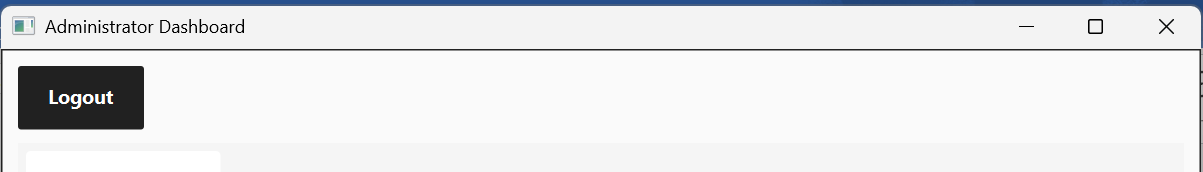
**2.1. User Management**

* Role-based login and dashboards for Patients, Doctors, and Admins.

1. Login Screen (Also showing secure login/logout):

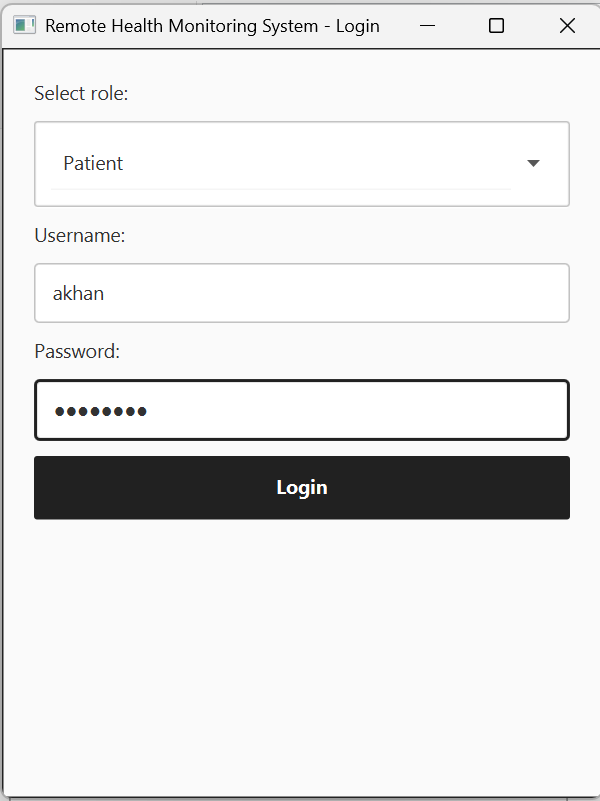
Main Screen User validation  
  

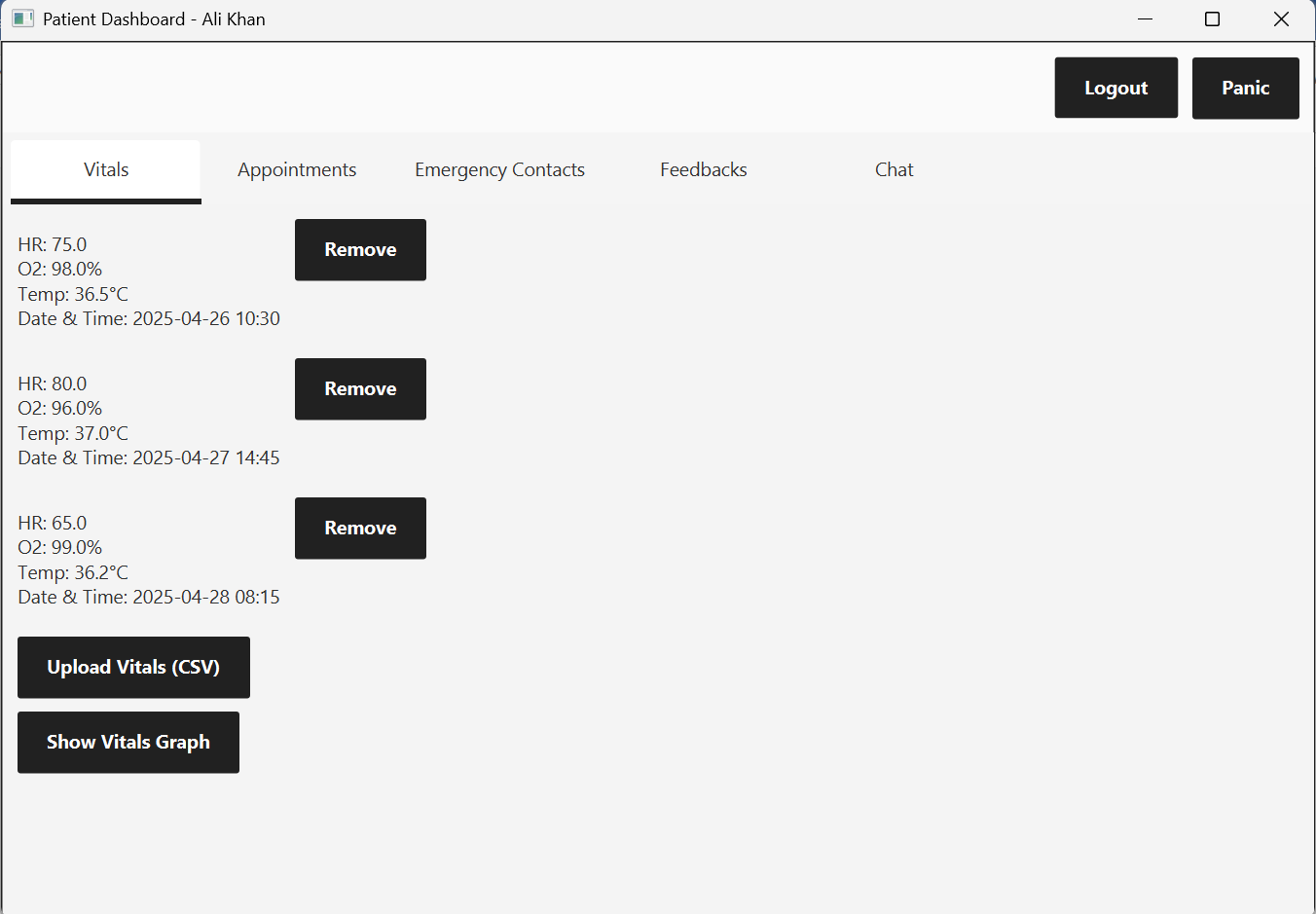
Logging out:

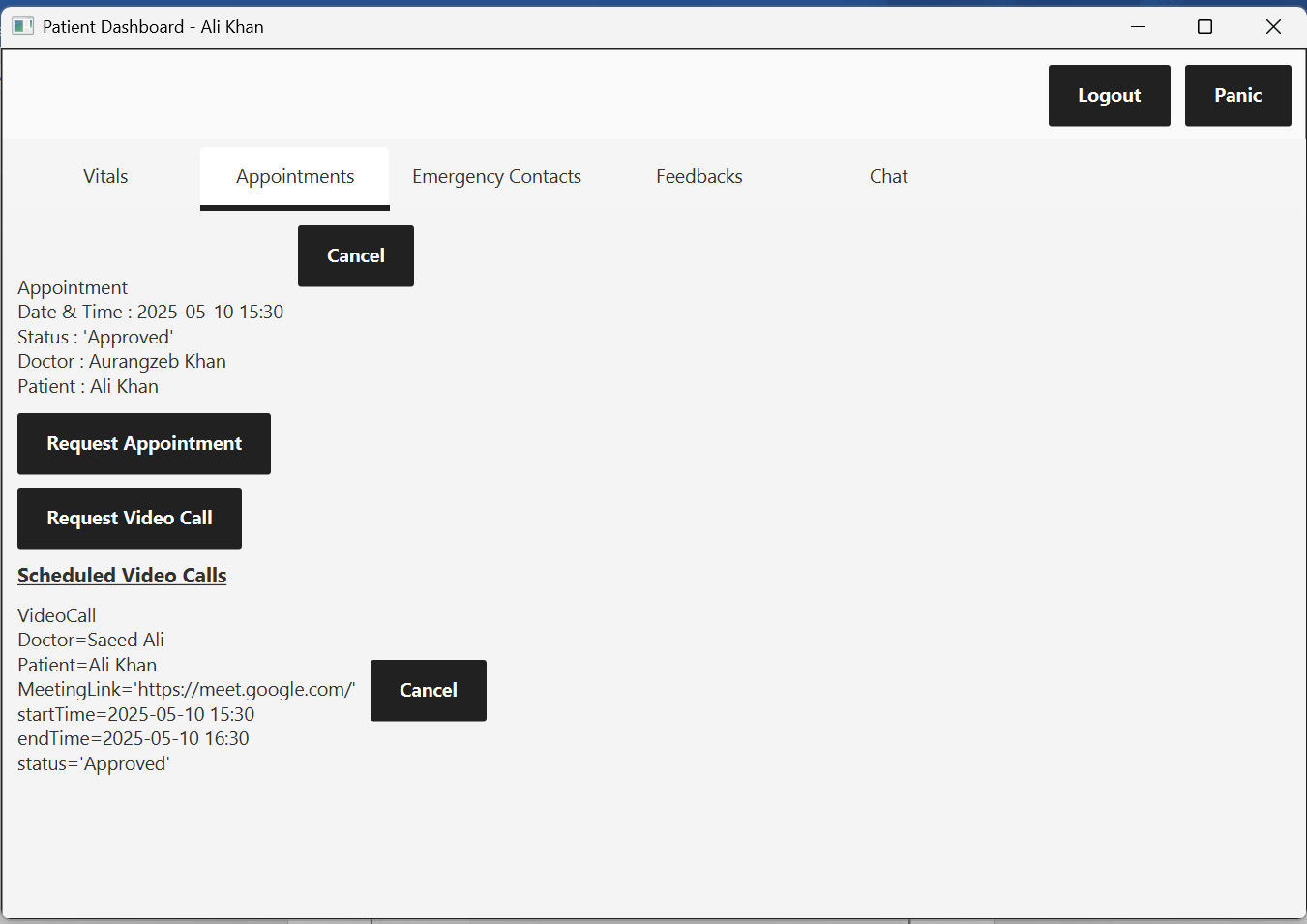


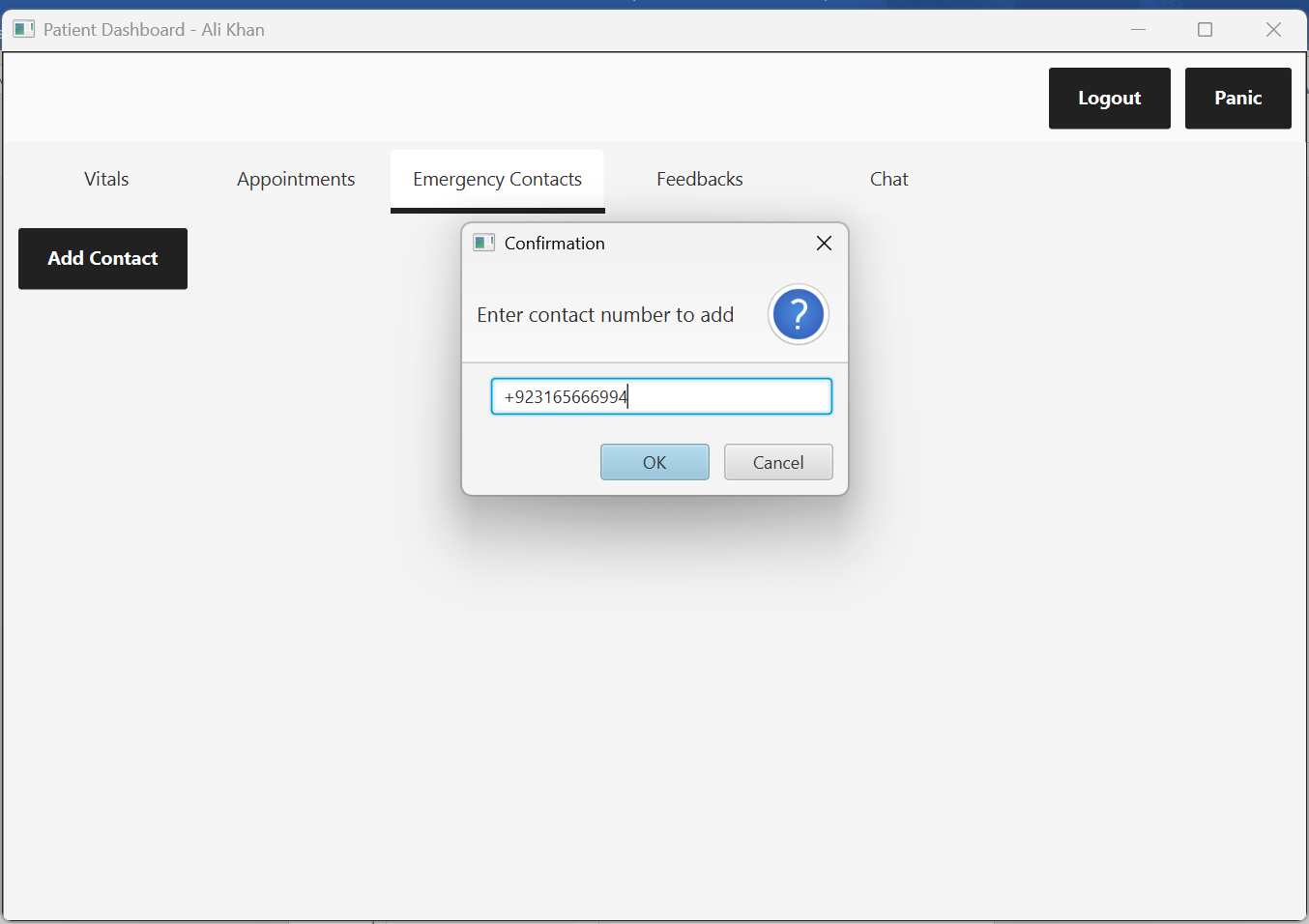
2. Role based dashboards with different functionalities

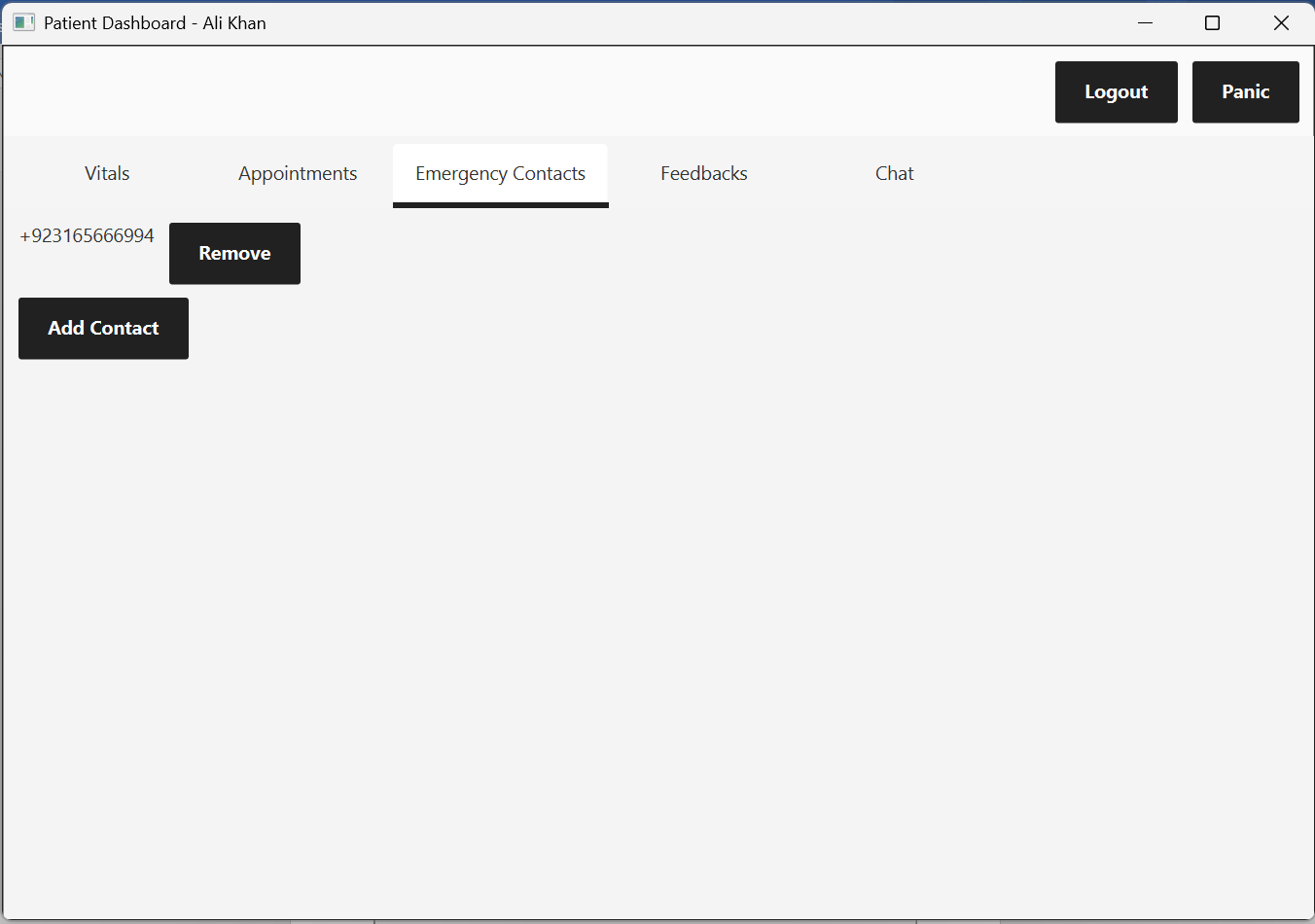
a. **Patient Dashboard** (with all different tabs and available functionalities)

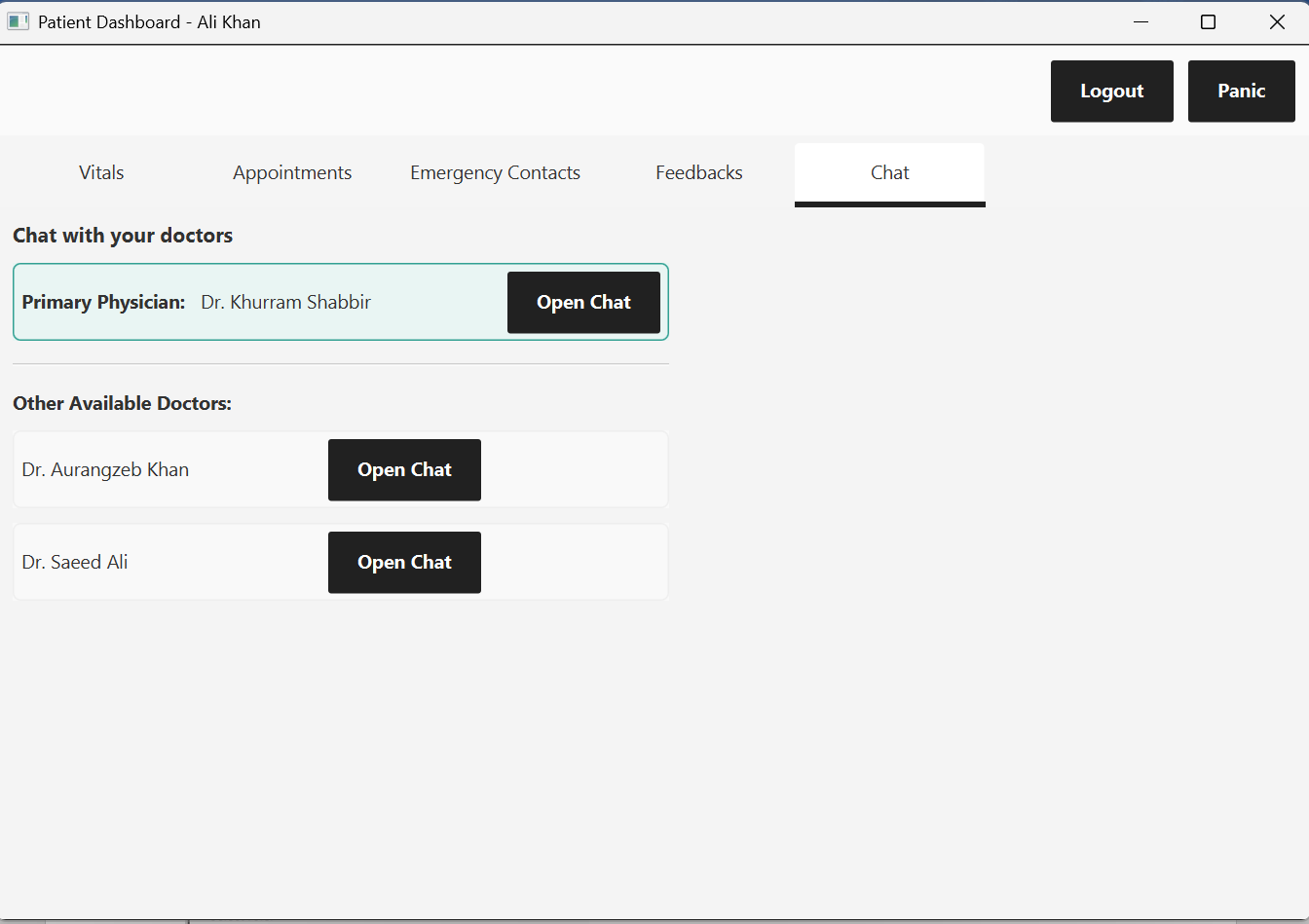




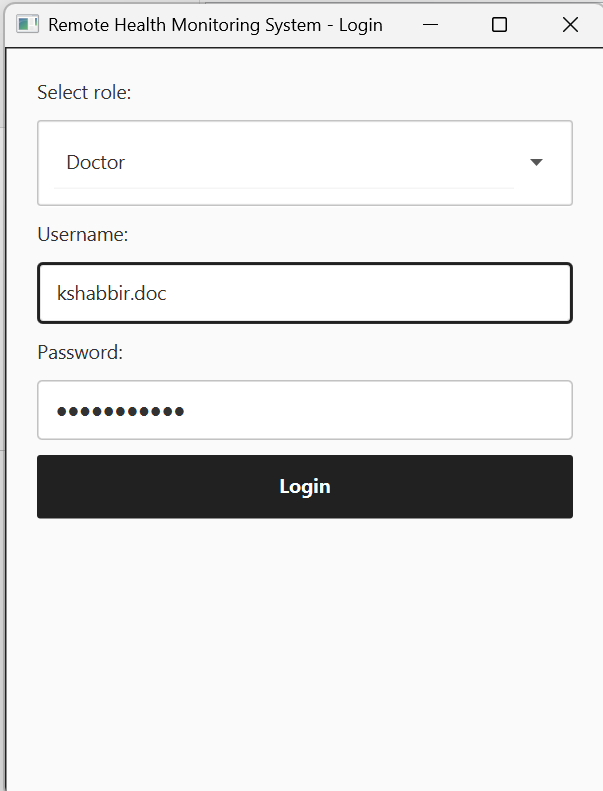


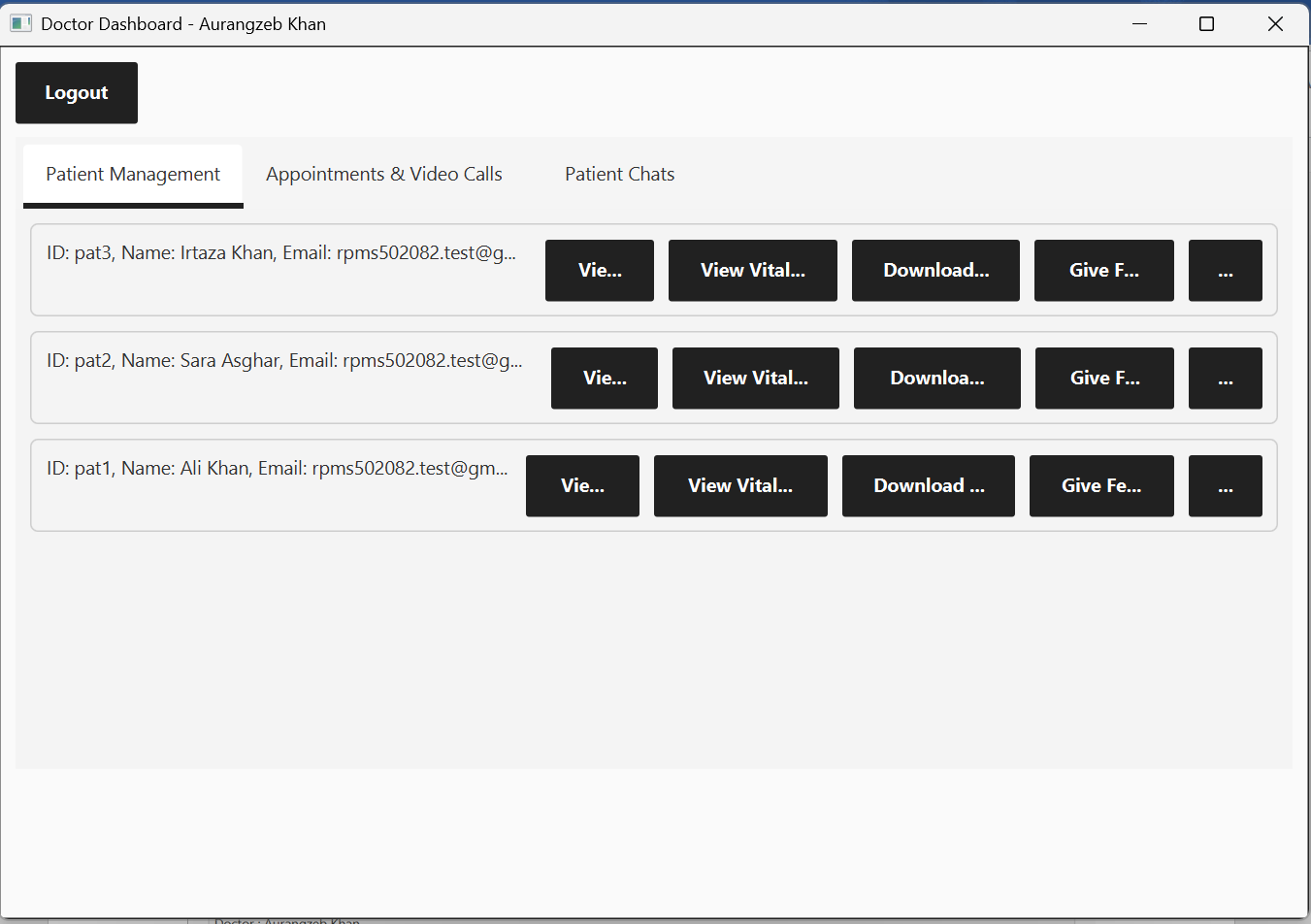


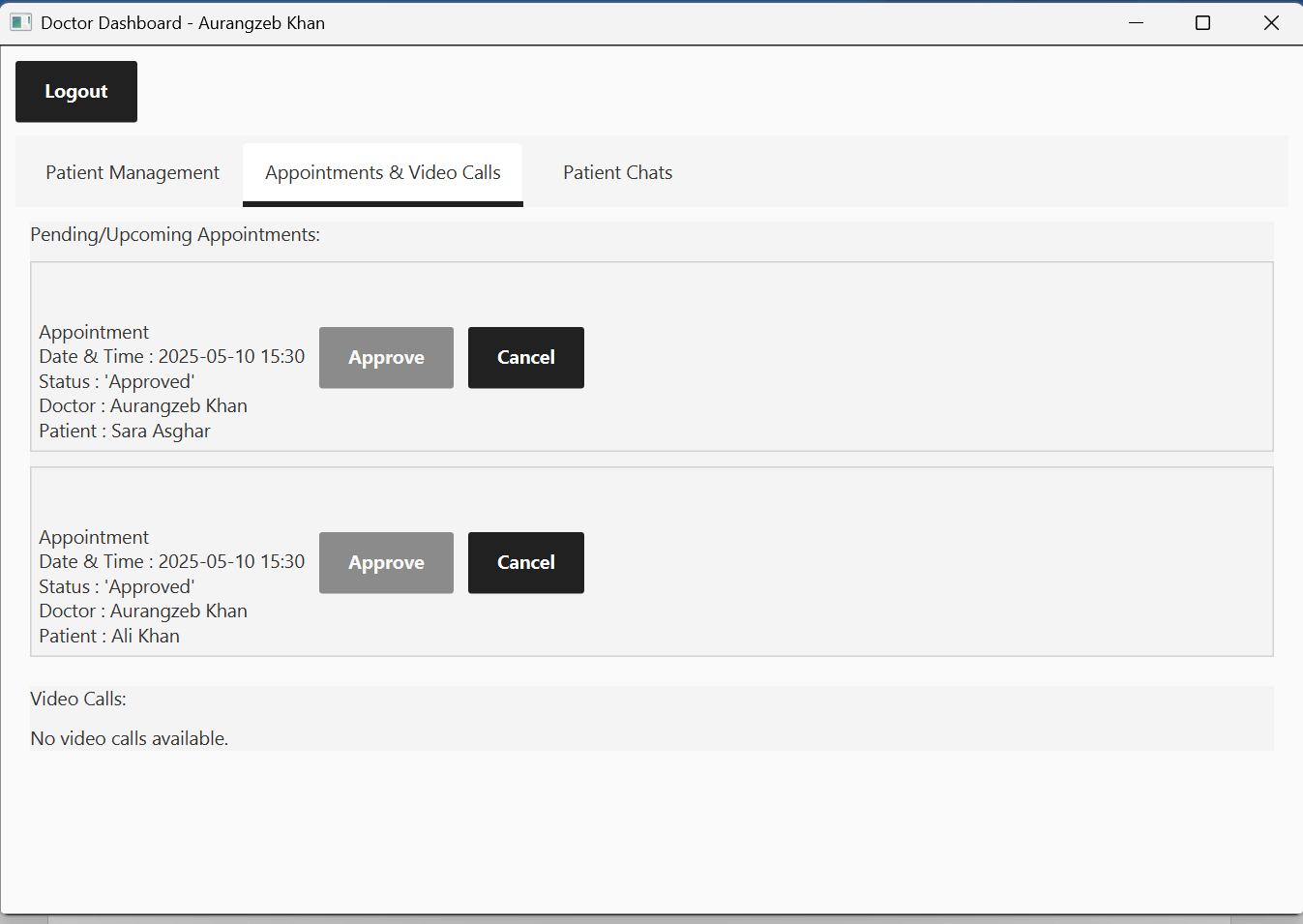


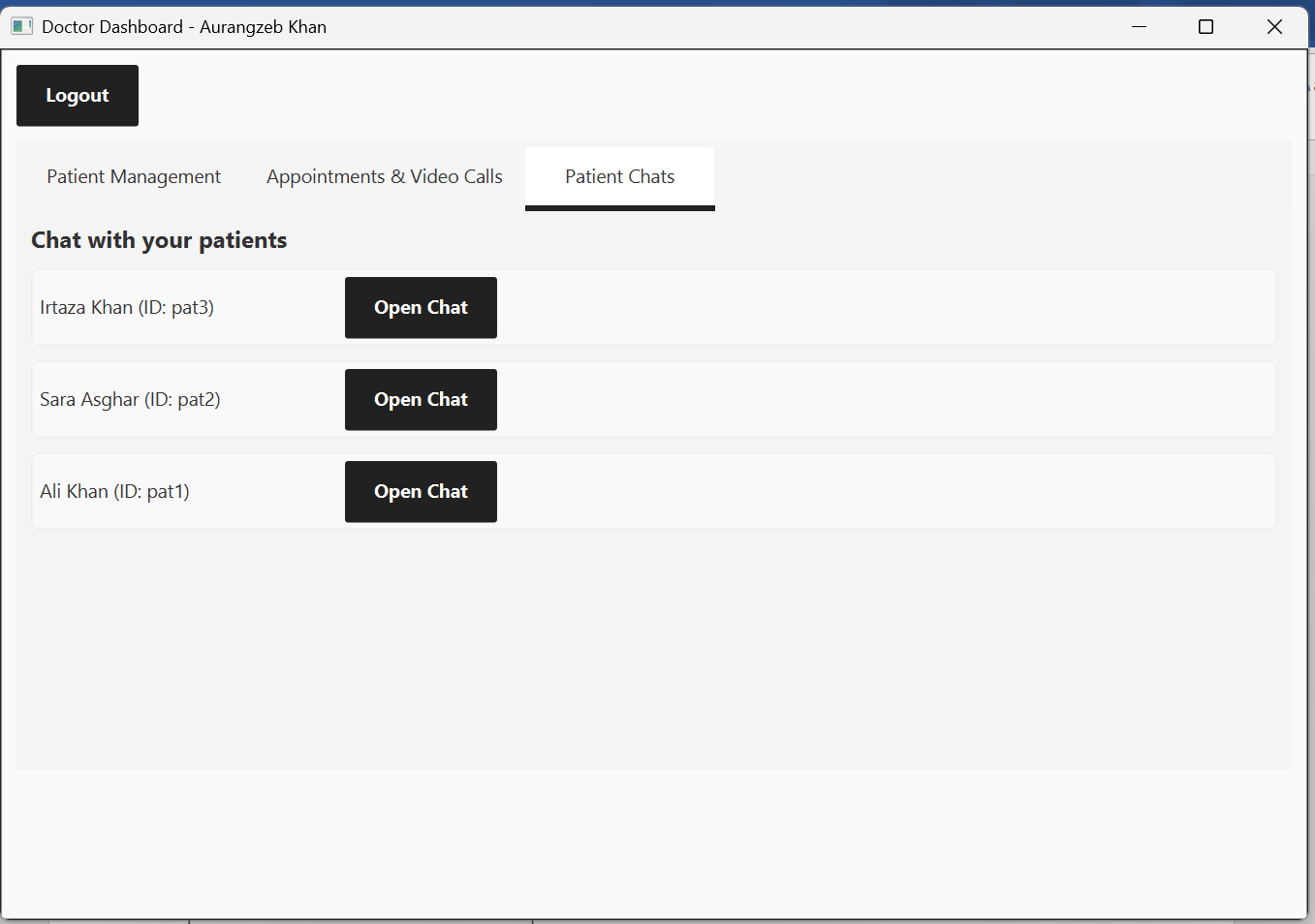


b. **Doctor Dashboard** (with all different tabs and available functionalities)

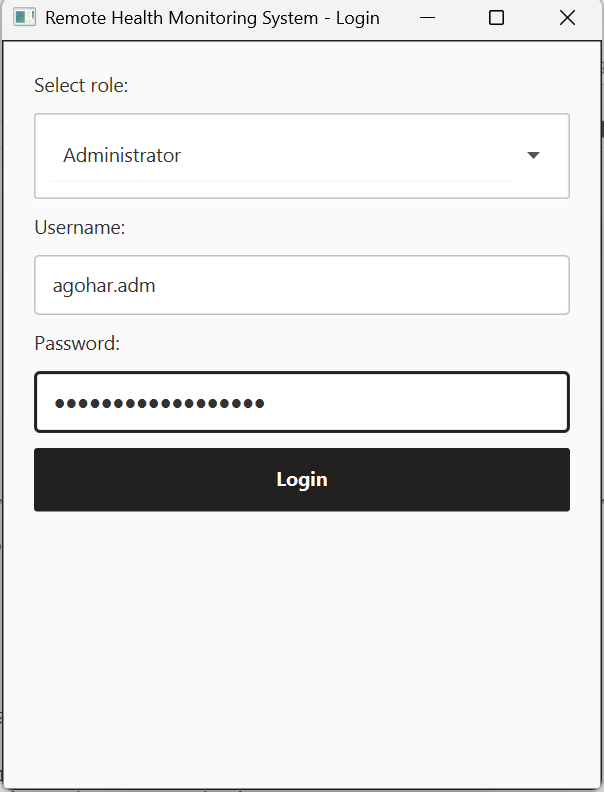


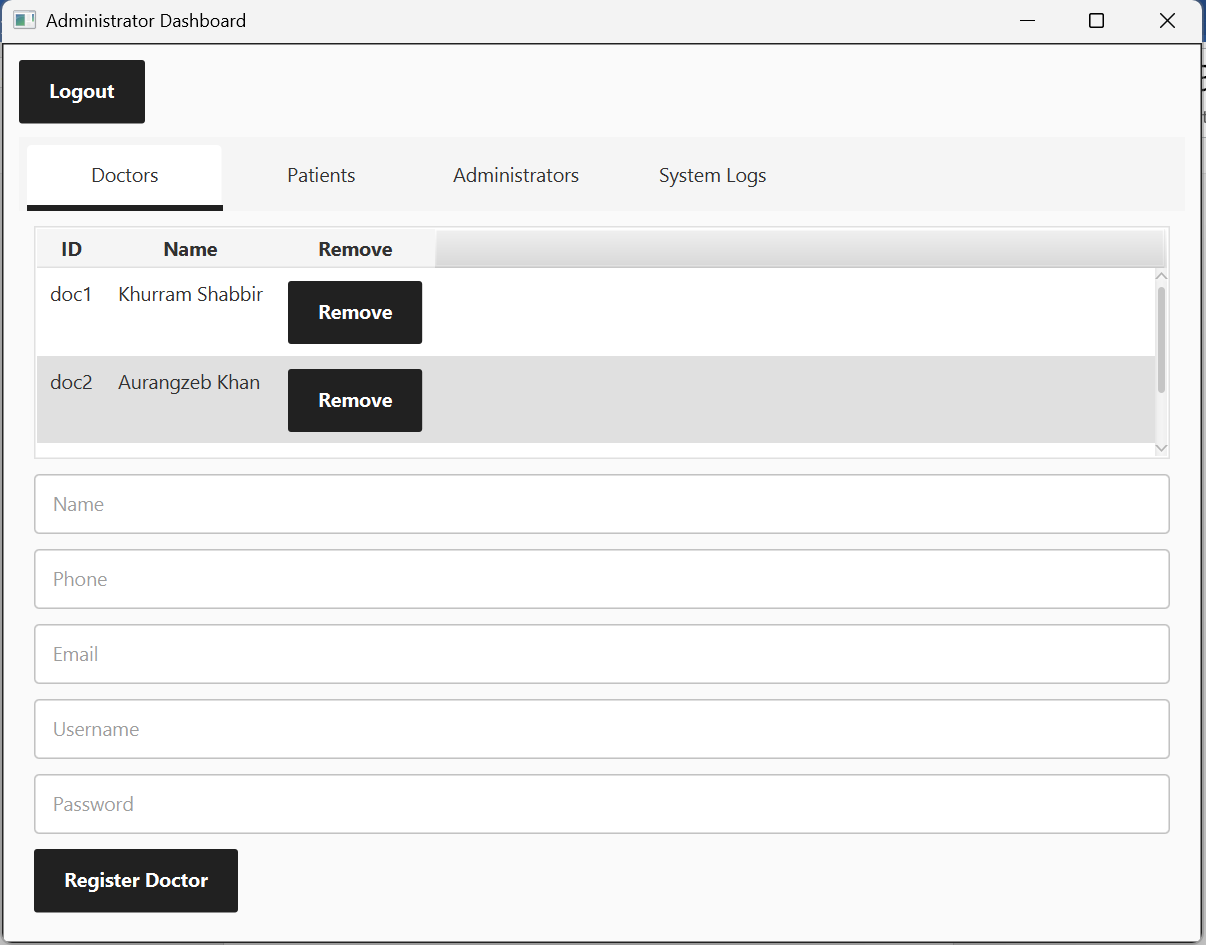


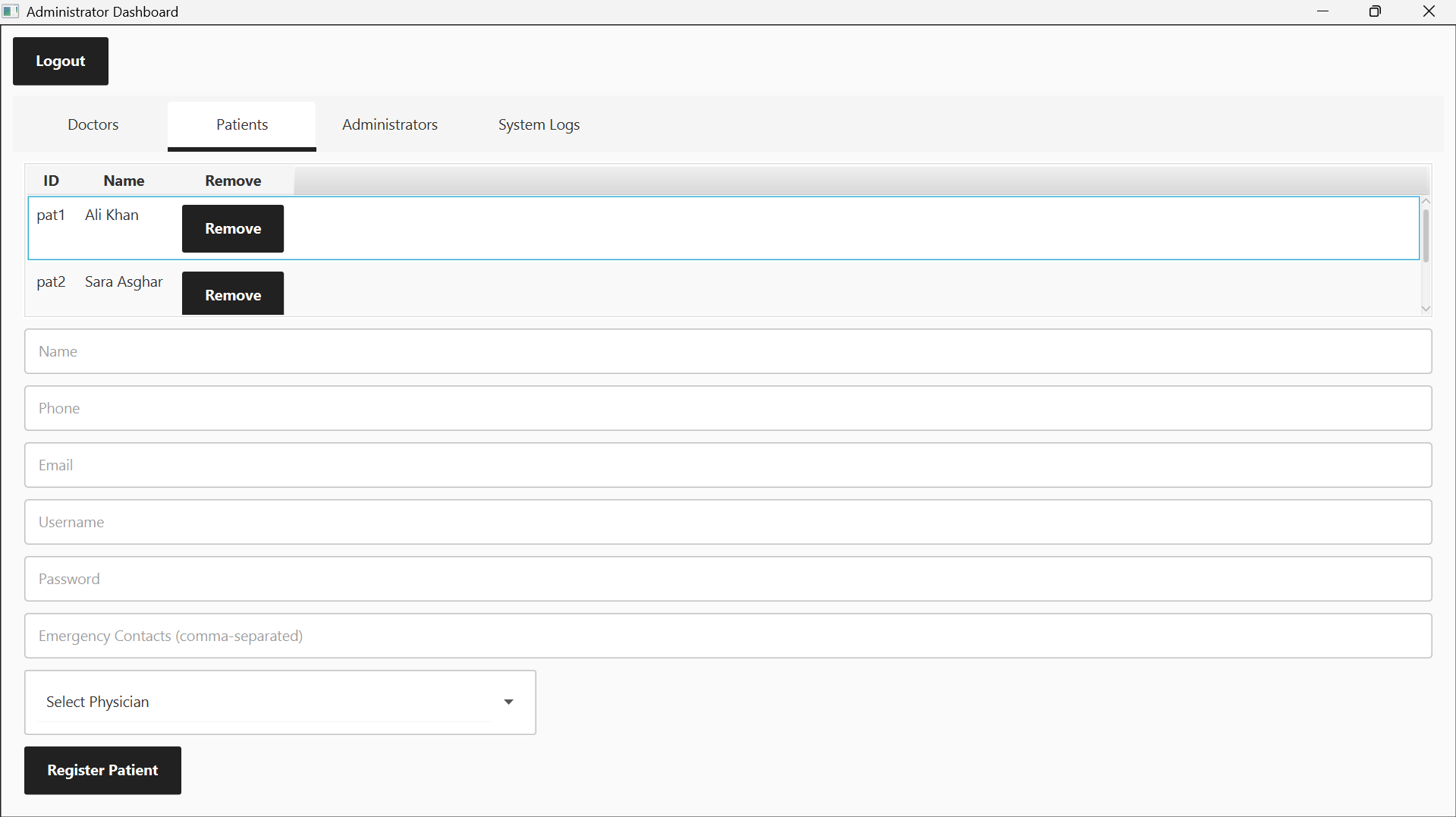


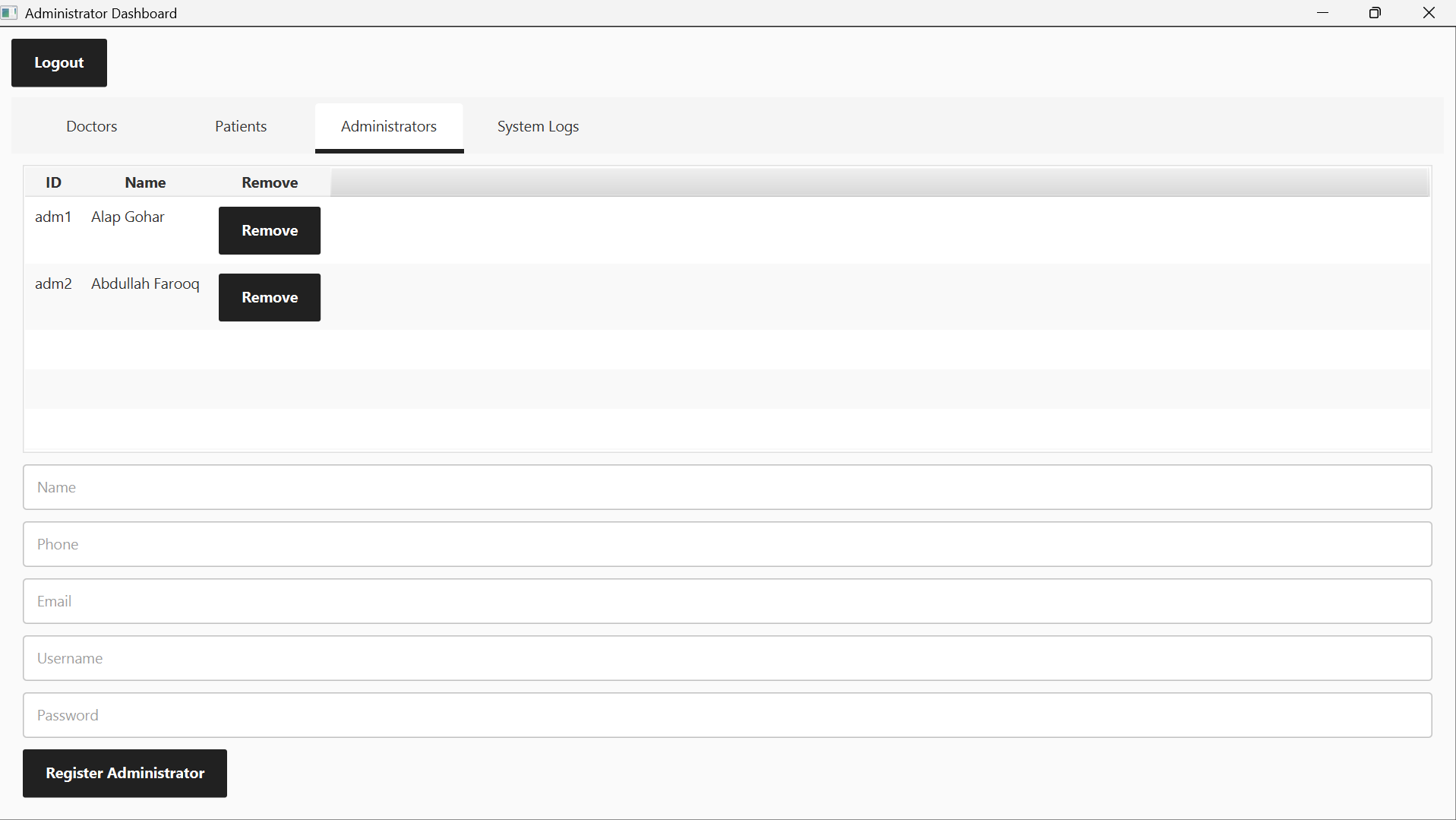


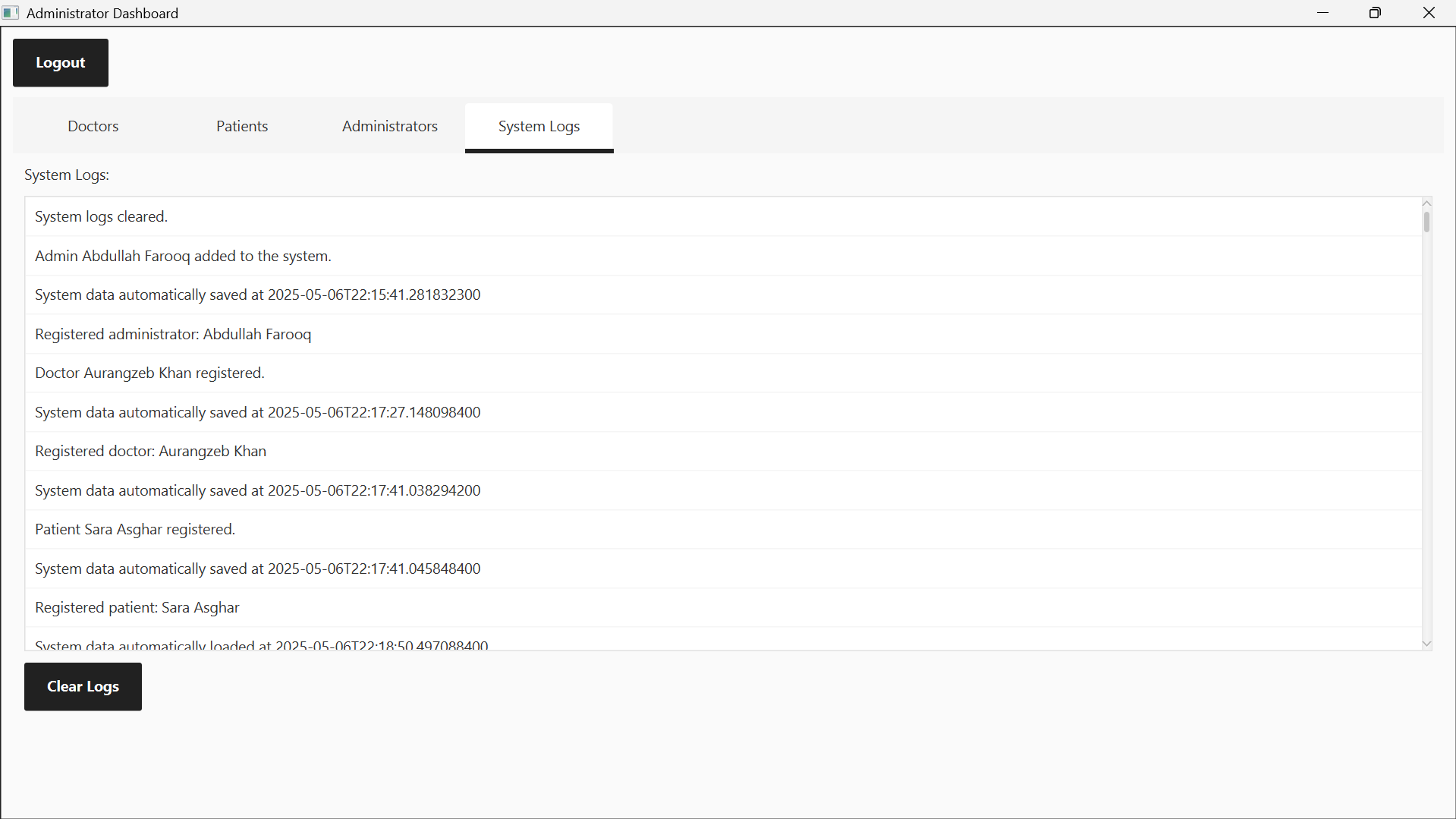
c. **Administrator Dashboard (with available buttons and functionalities)**

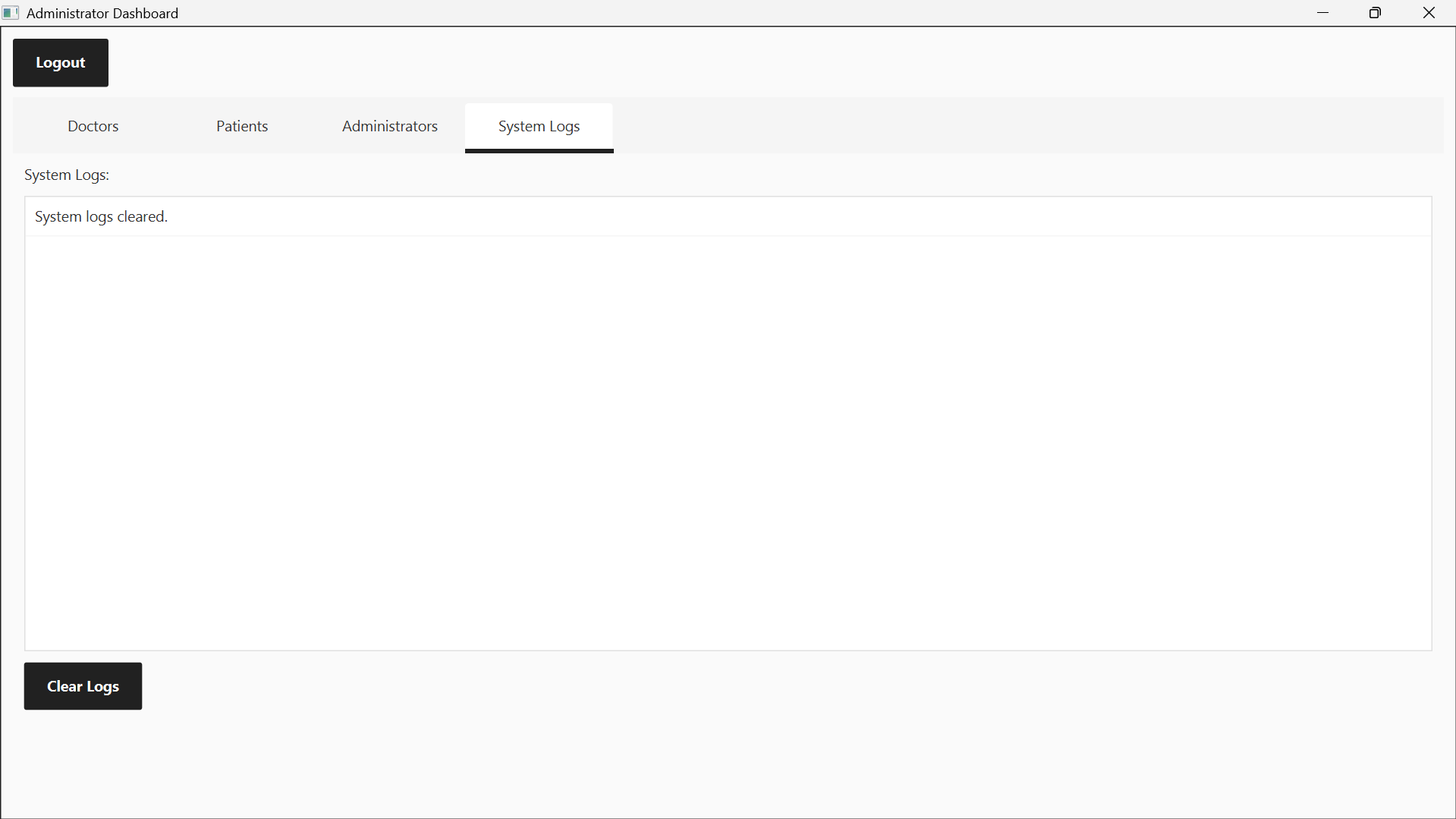








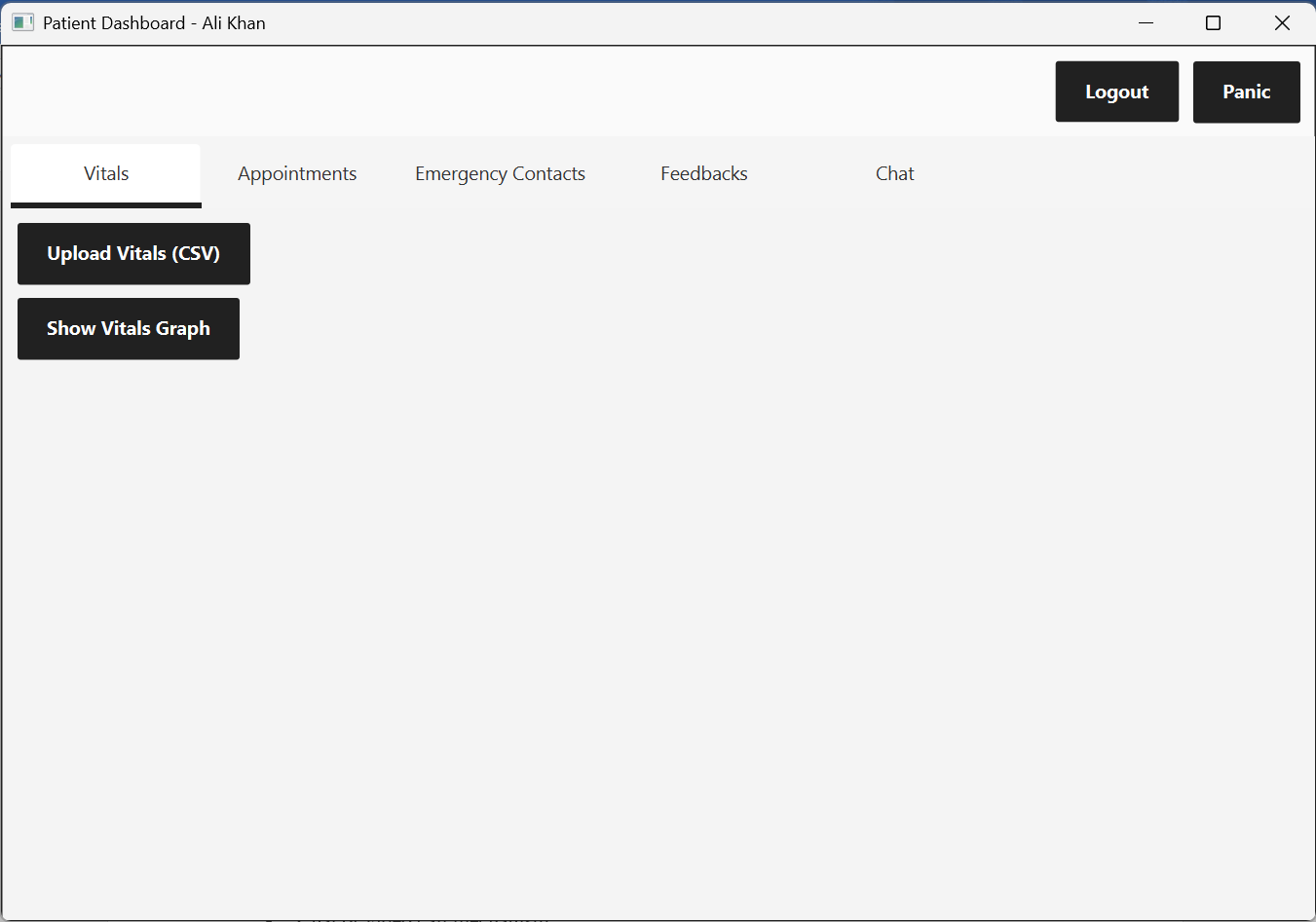




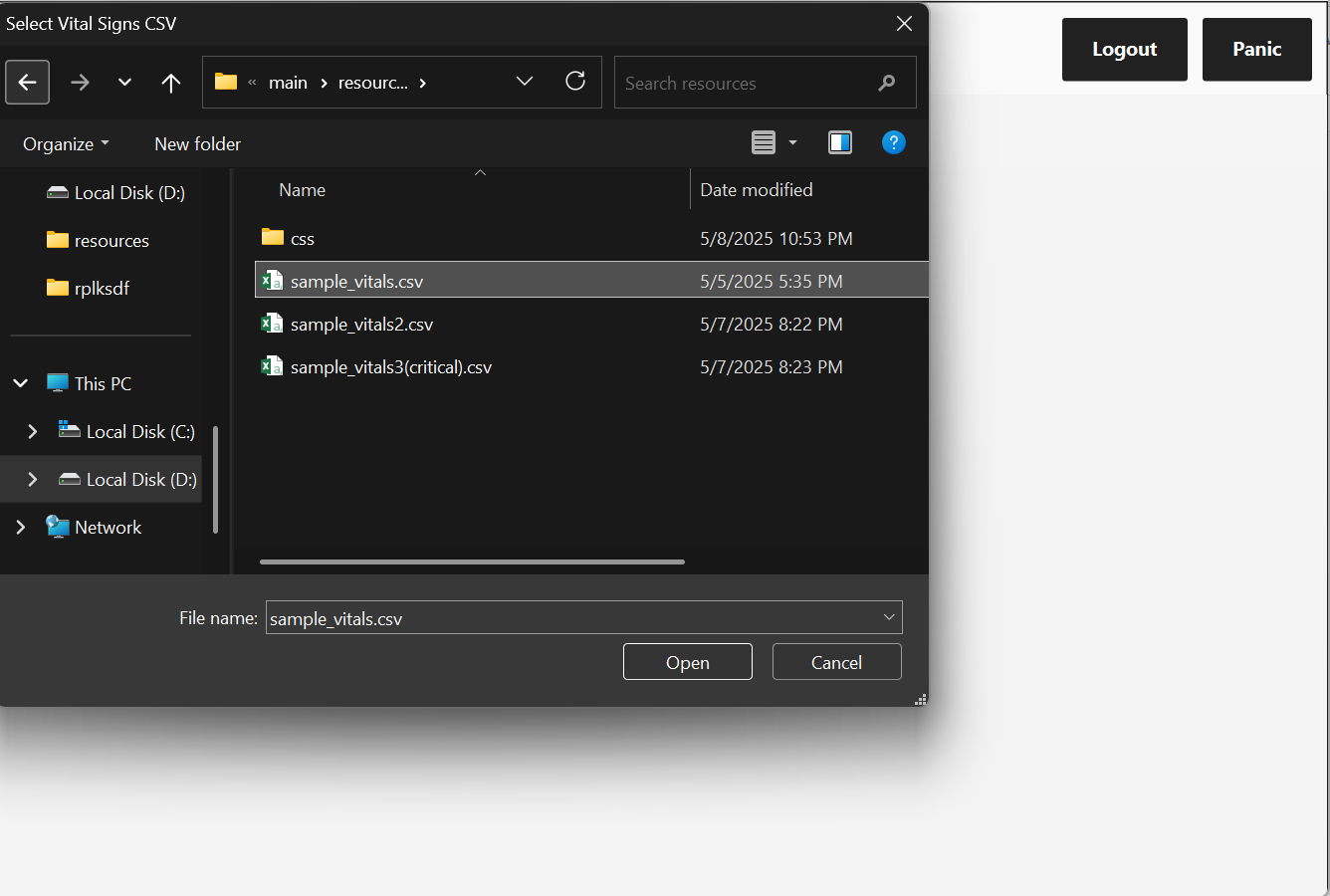
**2.2. Health Data Upload & Processing**

* Description of how patients upload vitals via CSV.

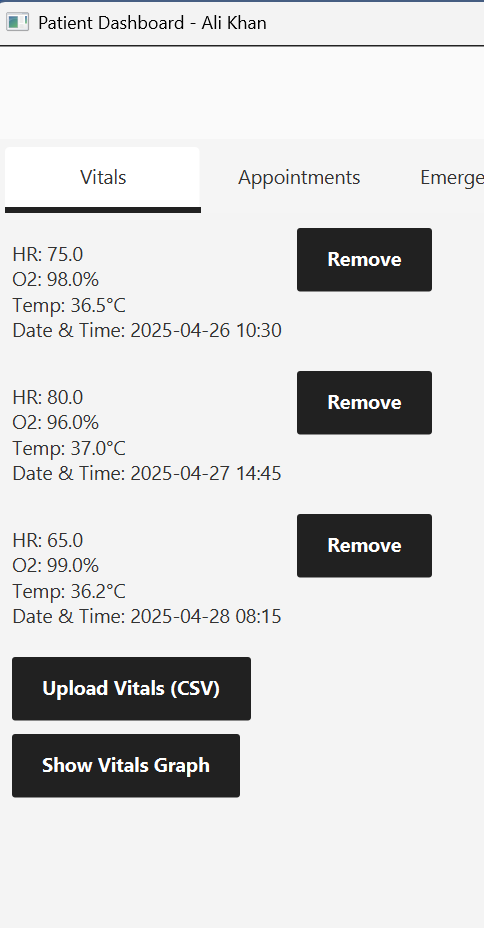
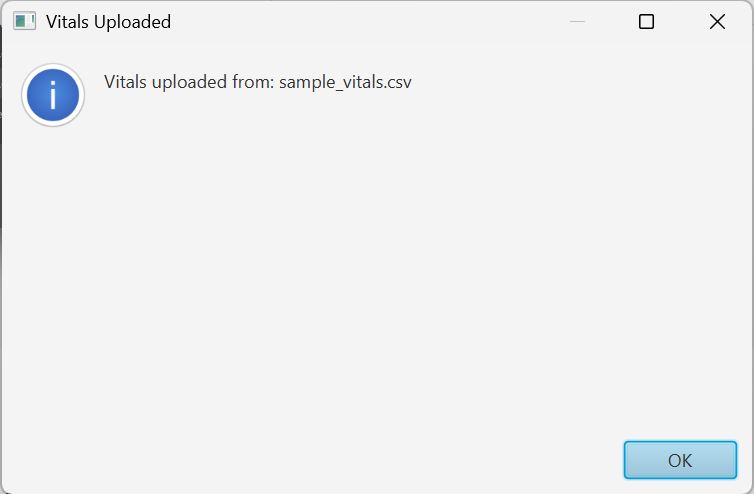
1. Login to the Patient Dashboard



1. Click on the Upload Vitals (CSV) button



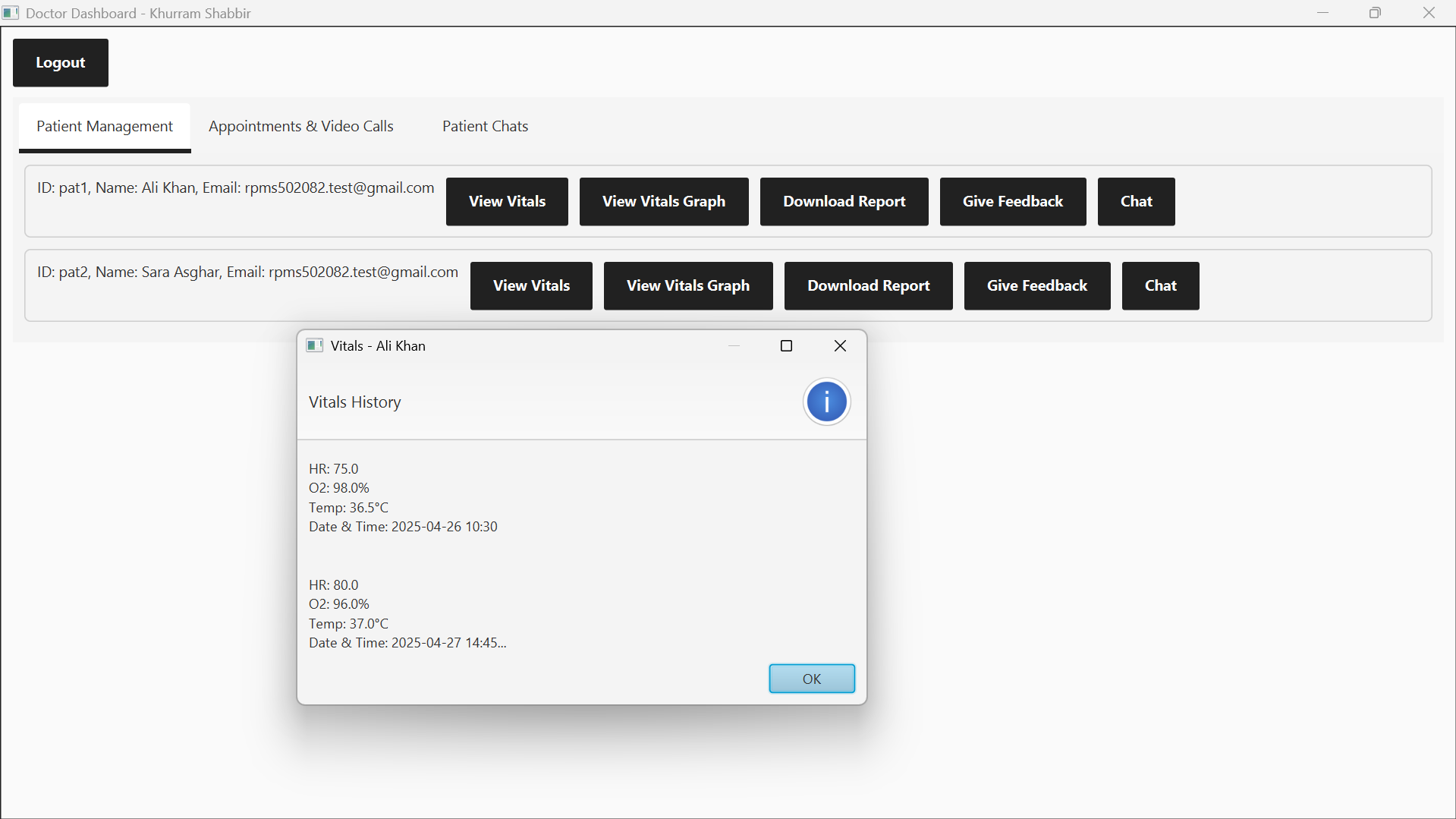
1. Select a valid csv file and upload it



**2.3. Doctor-Patient Interaction**

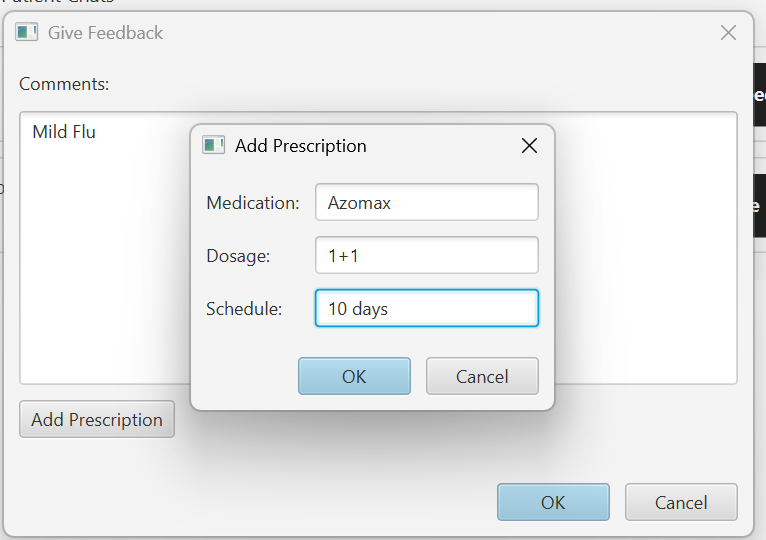
* Viewing vitals, giving feedback, and prescribing medications/tests.

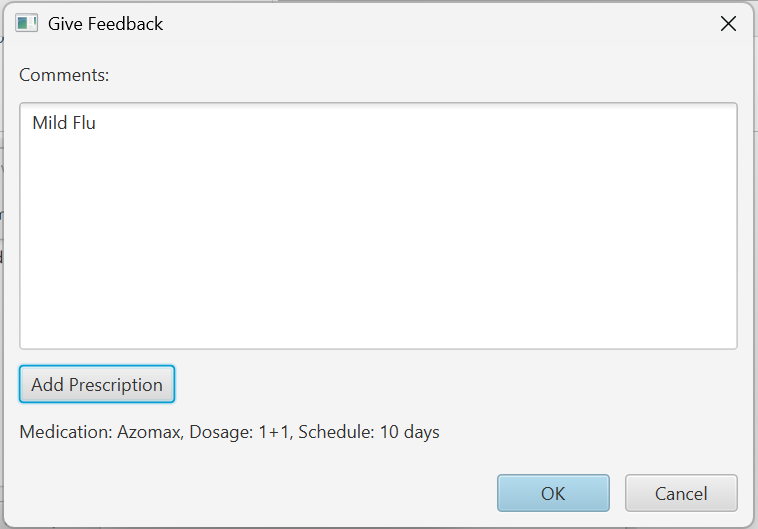
1. Viewing patient’s vitals:

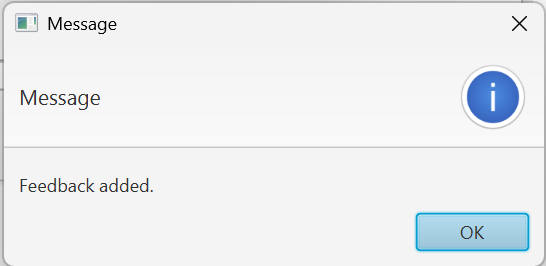


Click on the “view vitals” button to view the vitals.

1. Giving feedback and prescribing medicine



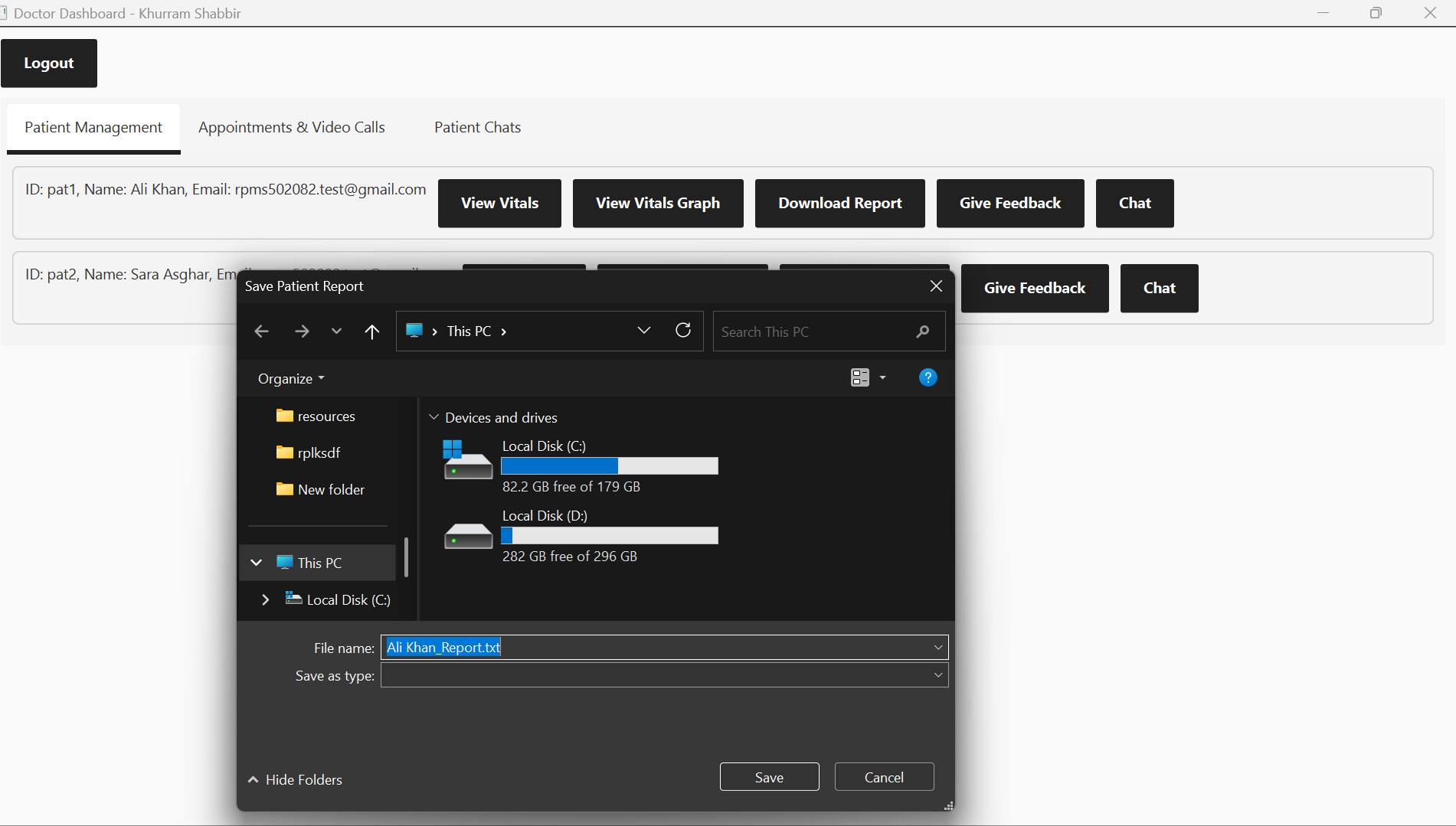


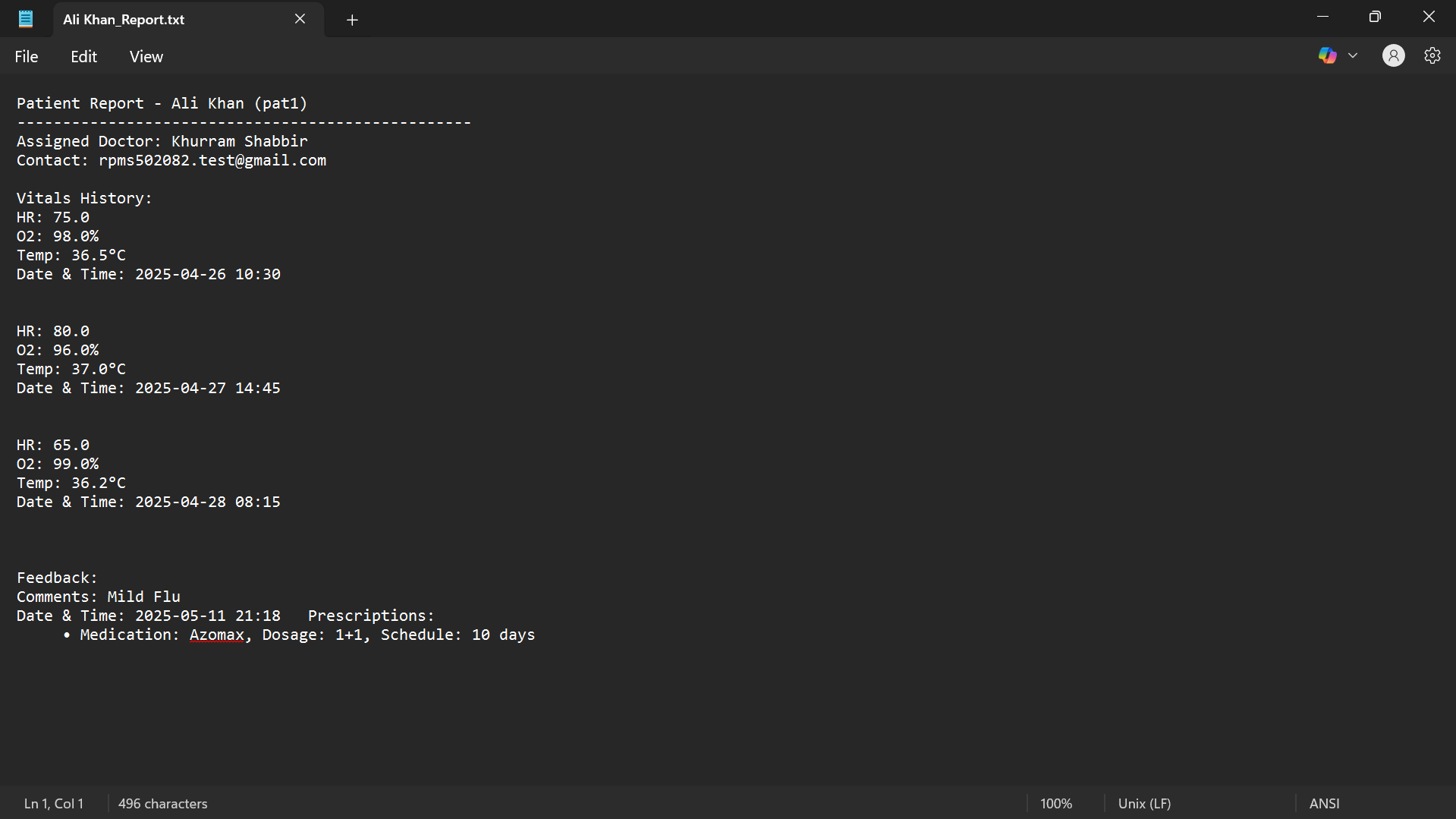


* Viewing history and feedback flow.

Past history will be displayed in a patient report

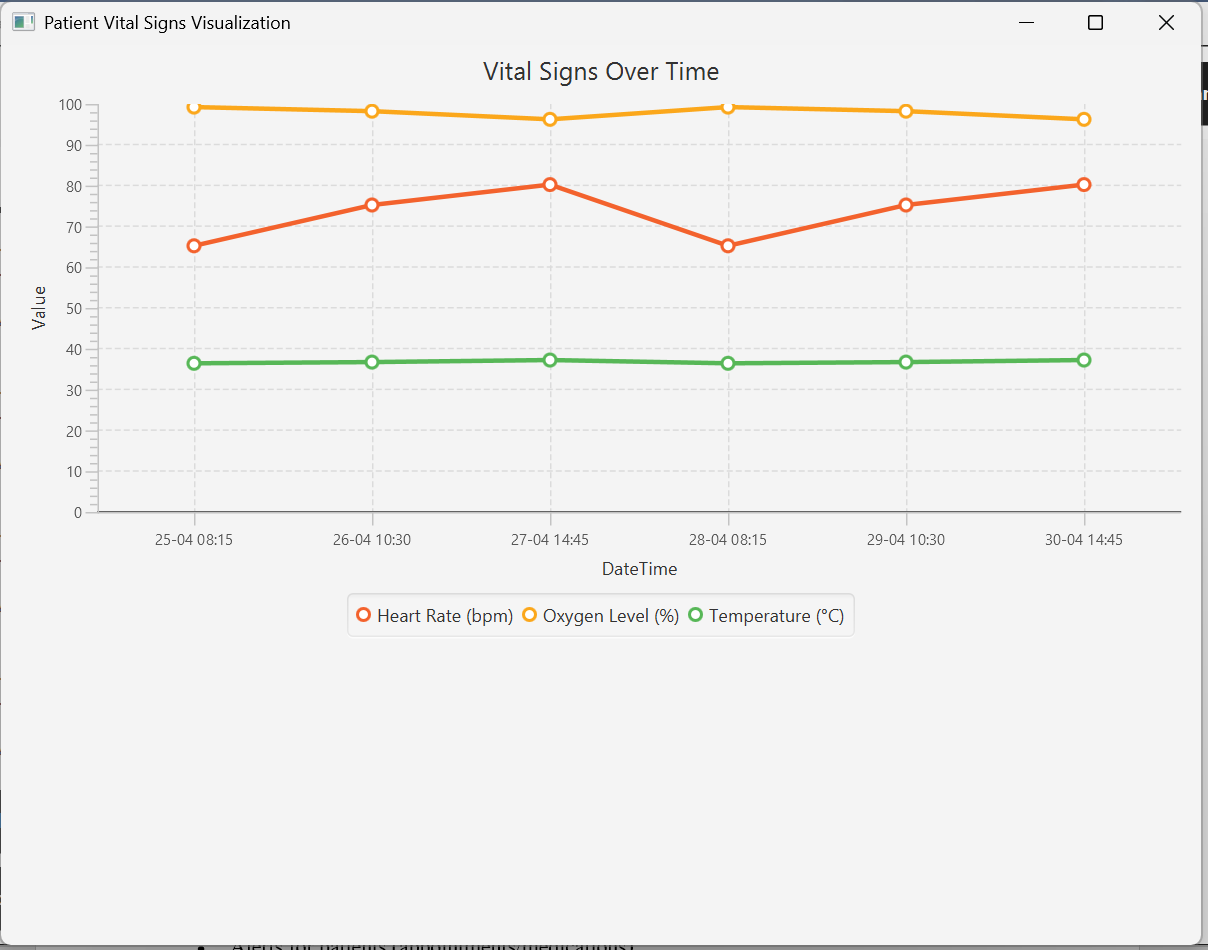
Click on the download report button and select a path to save the report as a text file.

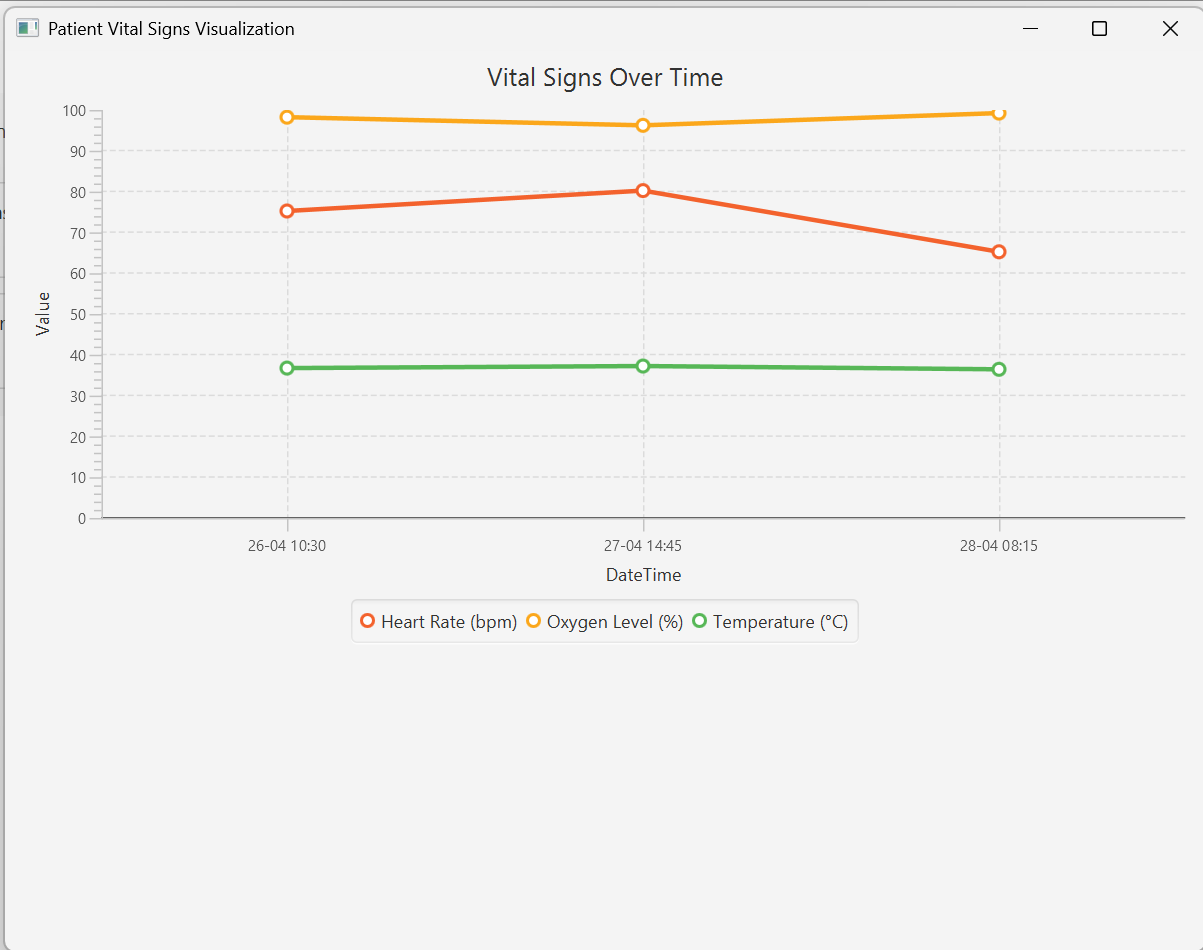




**2.4. Health Trends, Visualization & Report Generation**

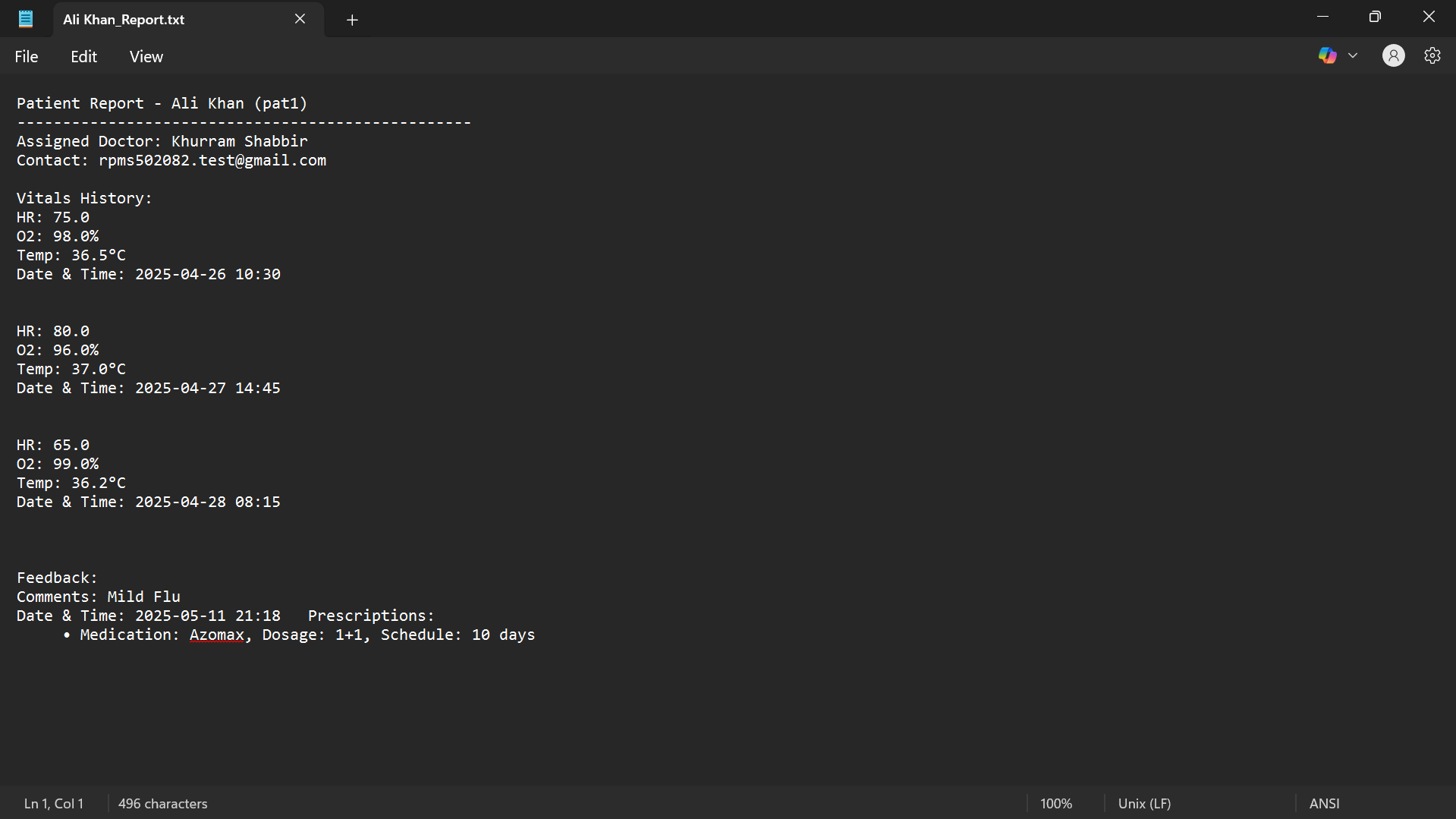
* Vitals graphs generated.





Both the patient and the doctor can view from their dashboards.

* Summary of downloadable health reports.

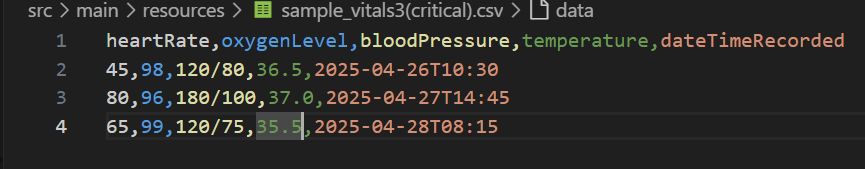


The Health report consists of Patient’s Information, Vitals History and Previous Feedbacks.

**2.5. Emergency Alert System**

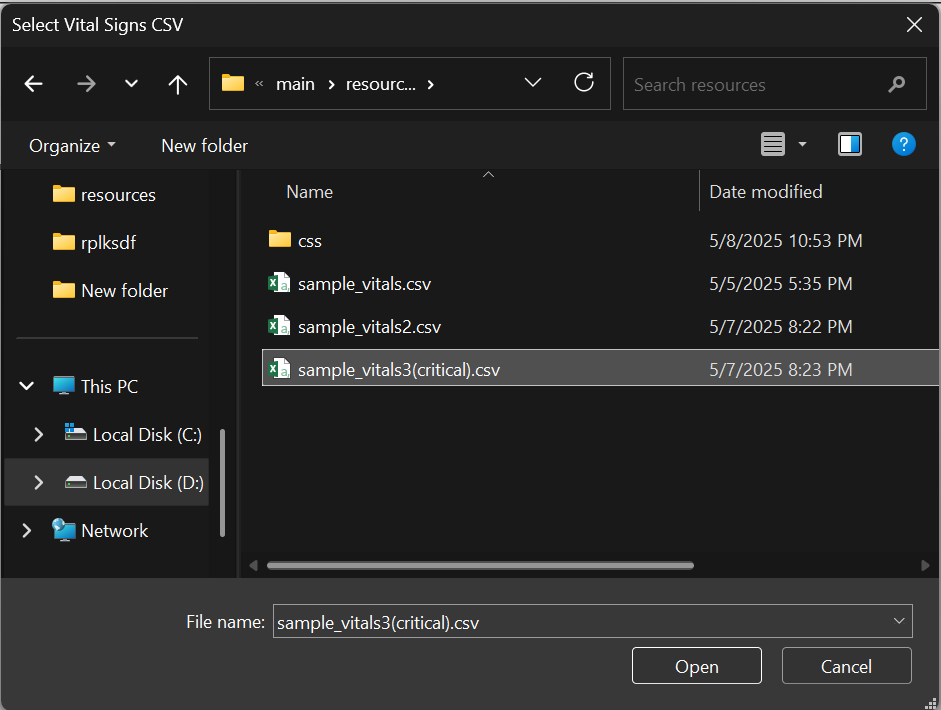
* Automatic alerts on abnormal vitals.

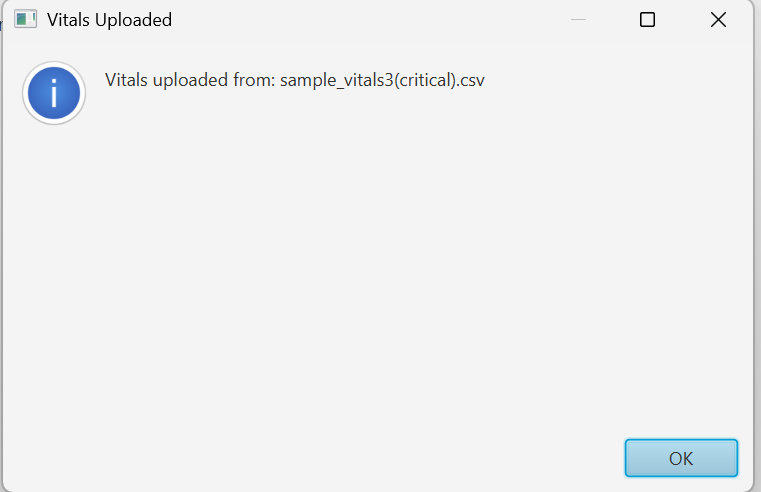
Now, we will see what happens when a patient uploads critical vitals.



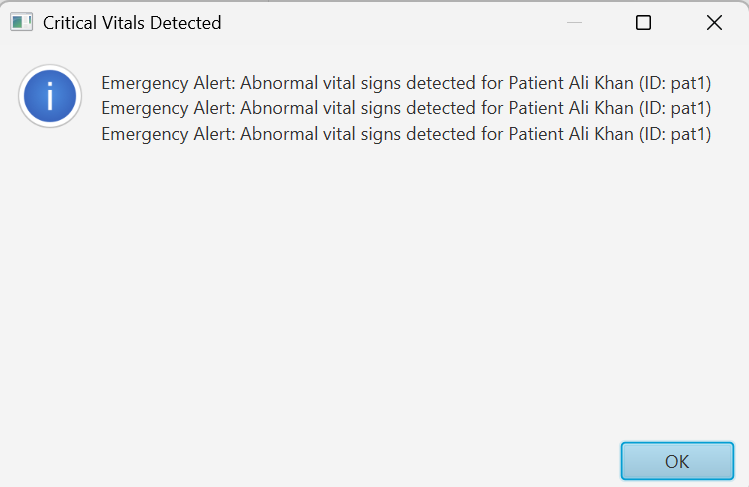
As we can see, this files consists of irregular/ critical vitals.

Let’s upload them now.



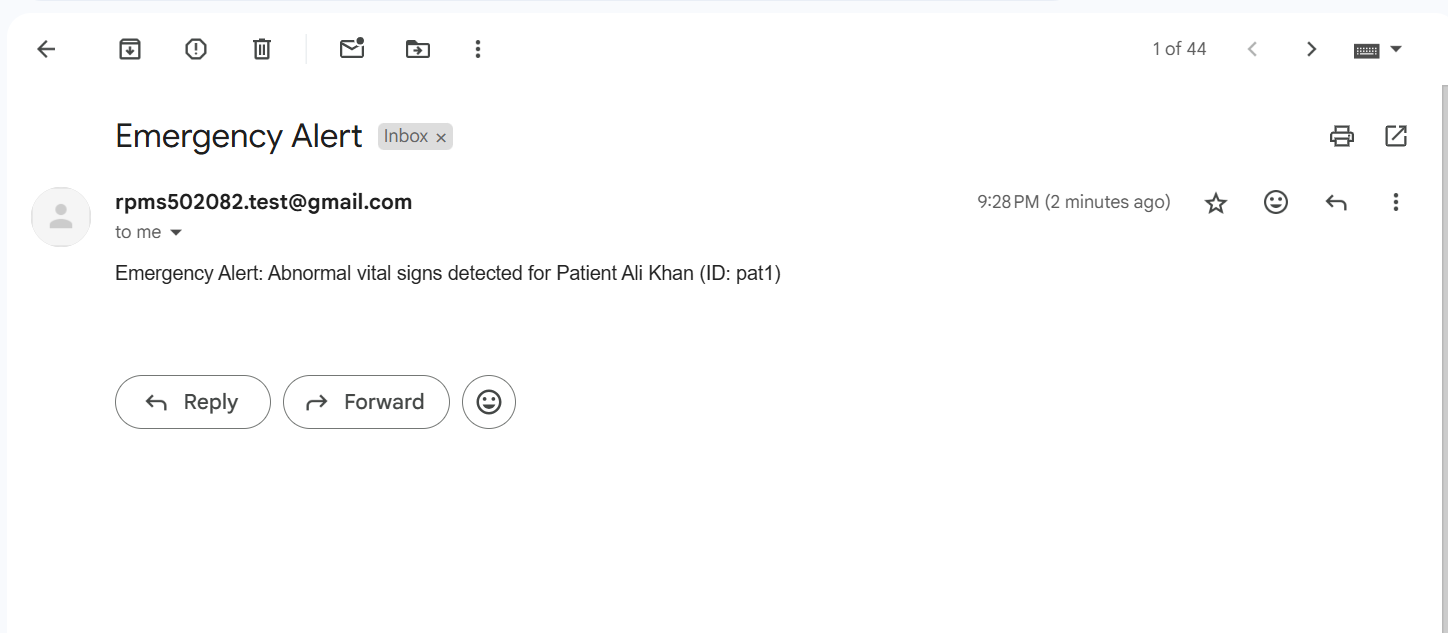


* 1. Patient receives alert:

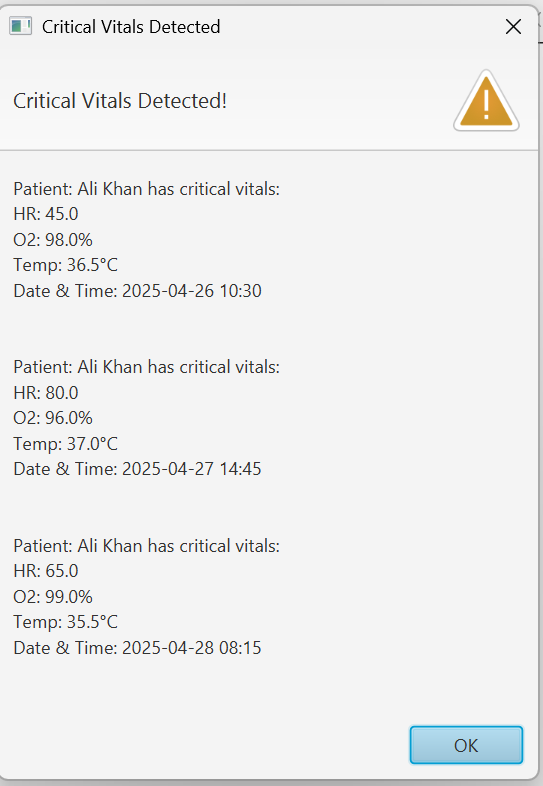


* 1. Email sent to emergency contacts of the patient and the assigned doctor (SMS not fully integrated)

NOTE: only 1 default email has been used in this project for testing purposes

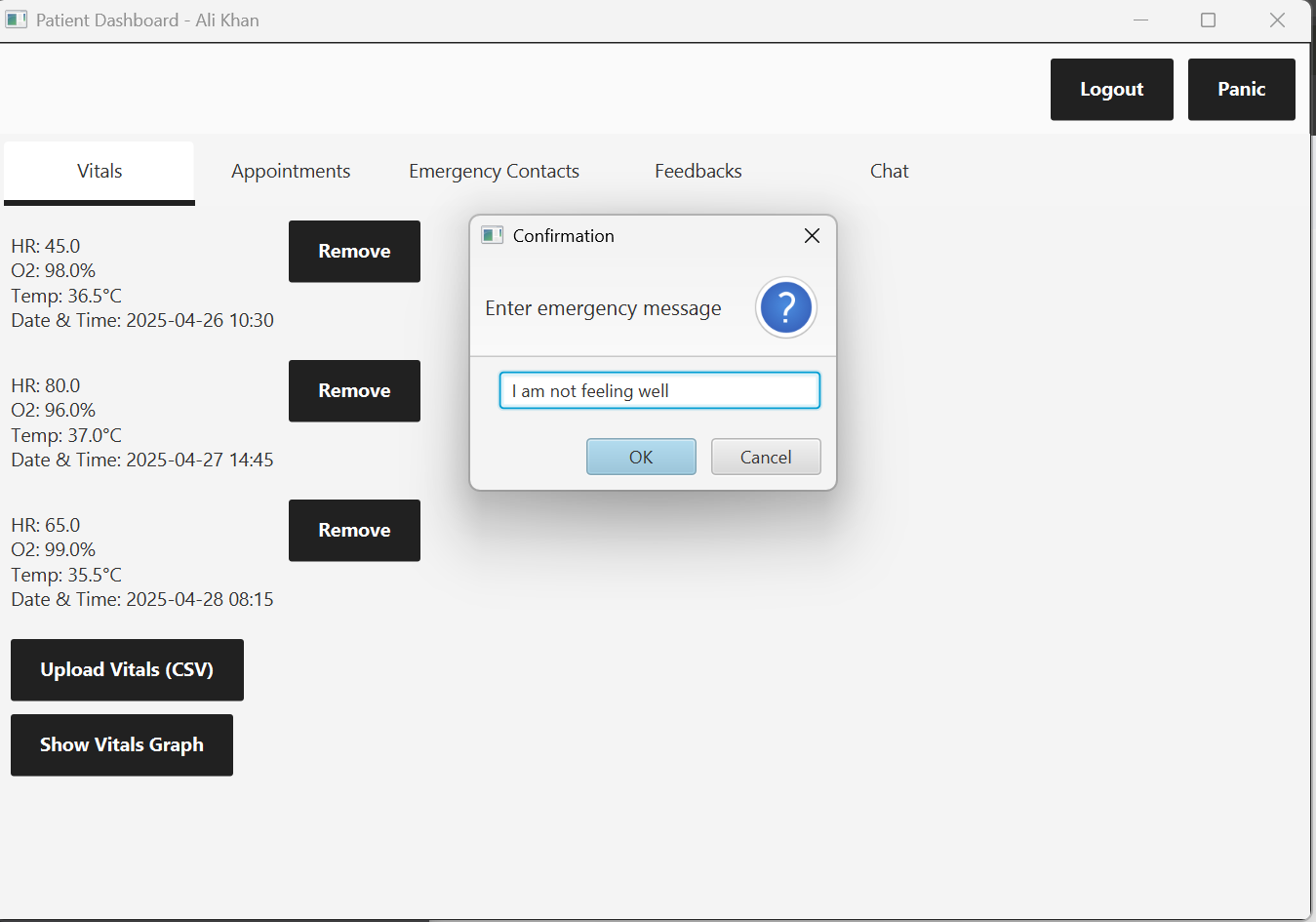


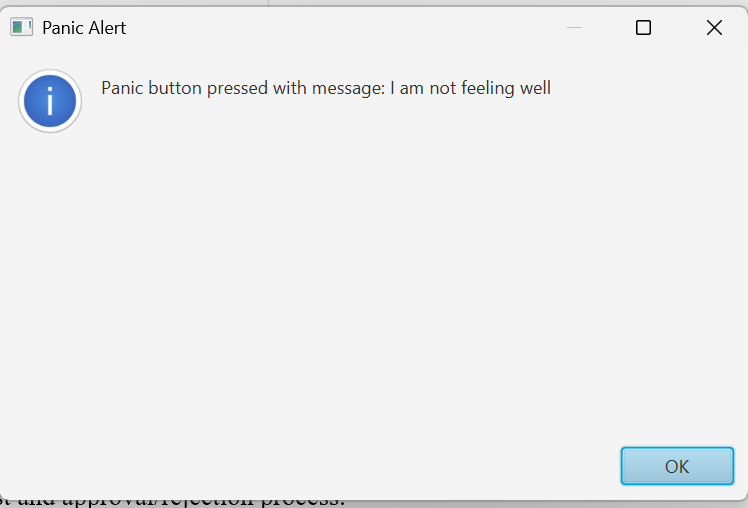
* 1. The assigned doctor (Physician) will receive an alert on their dashboards when they login.

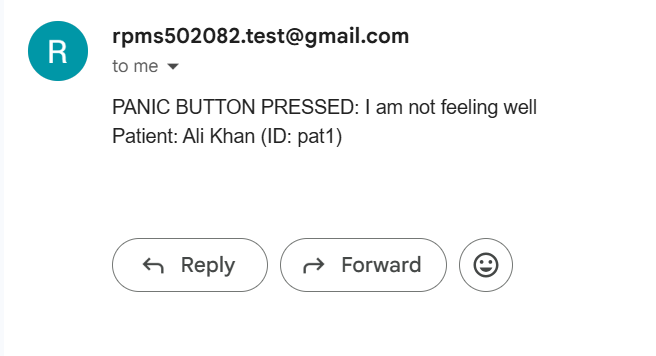
****

* Manual panic button and notification flow.

Patient has a panic button that they can press to enter a message to be sent to their emergency contacts and physician.





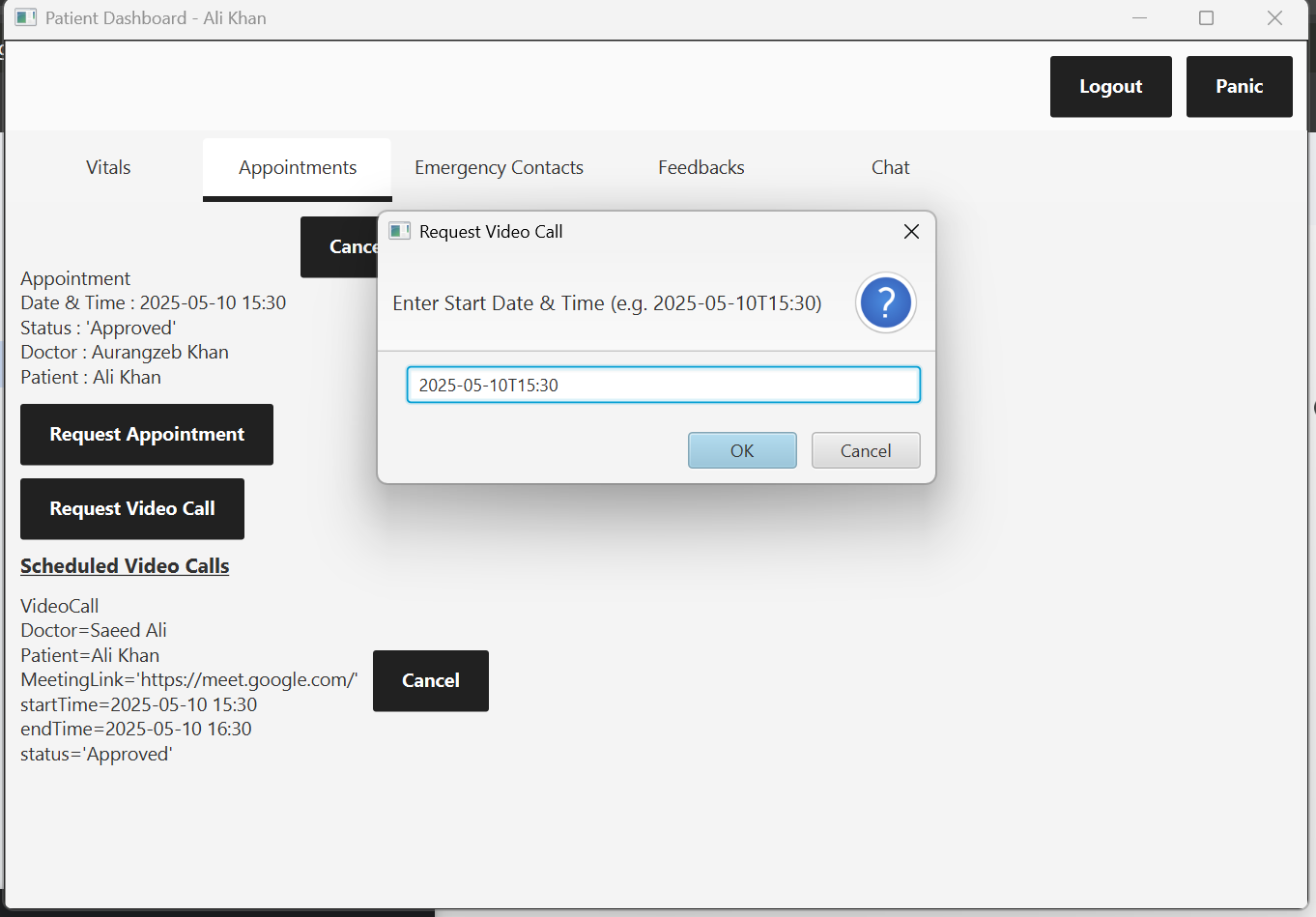


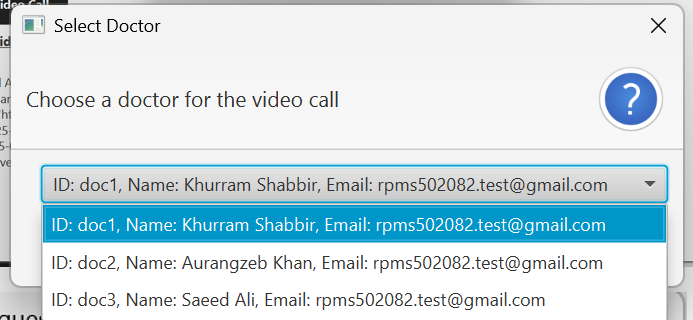
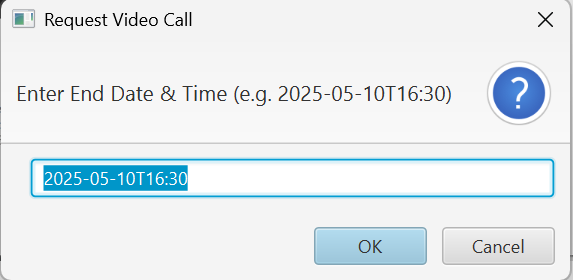
**2.6. Chat or Video Consultation**

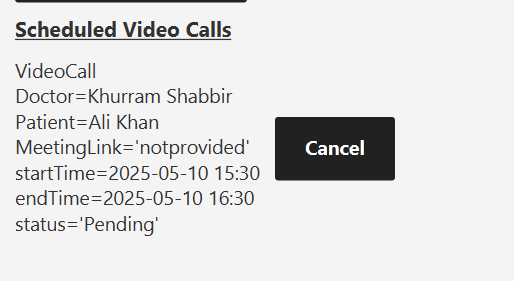
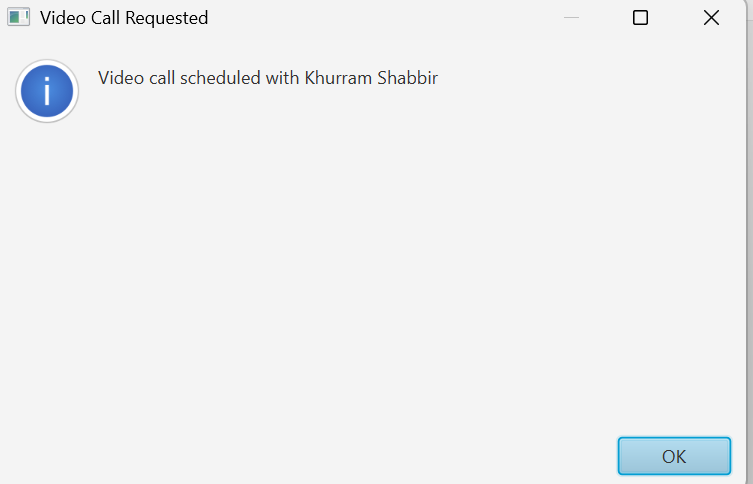
* Chat or video call mechanism.

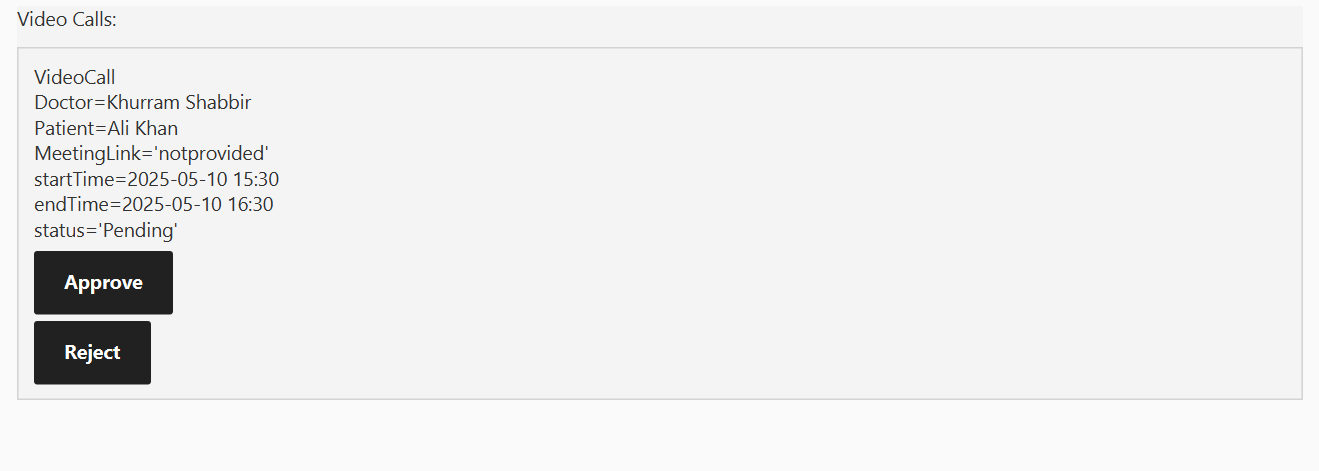
1. Video Call

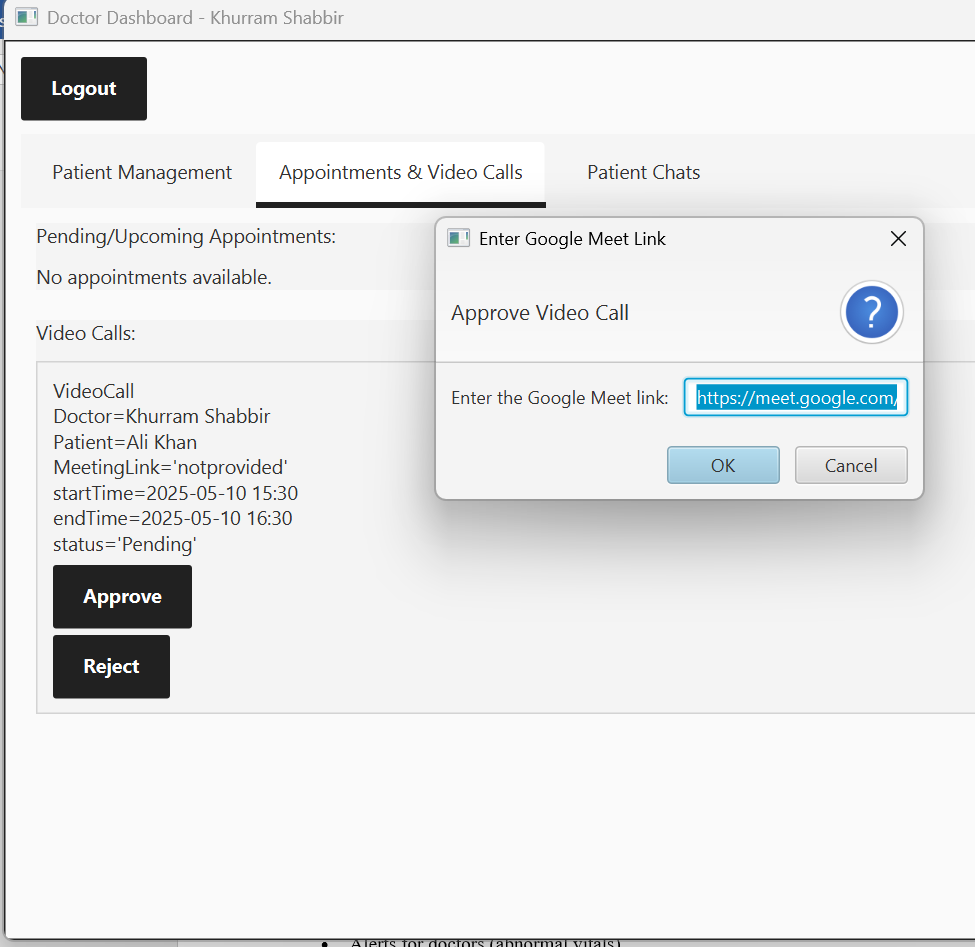
Patient can request a video call. On approval, doctor has to provide a google meet link. This link can then be used to join the meeting on the scheduled time.

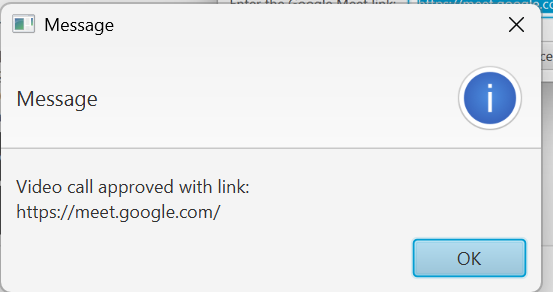




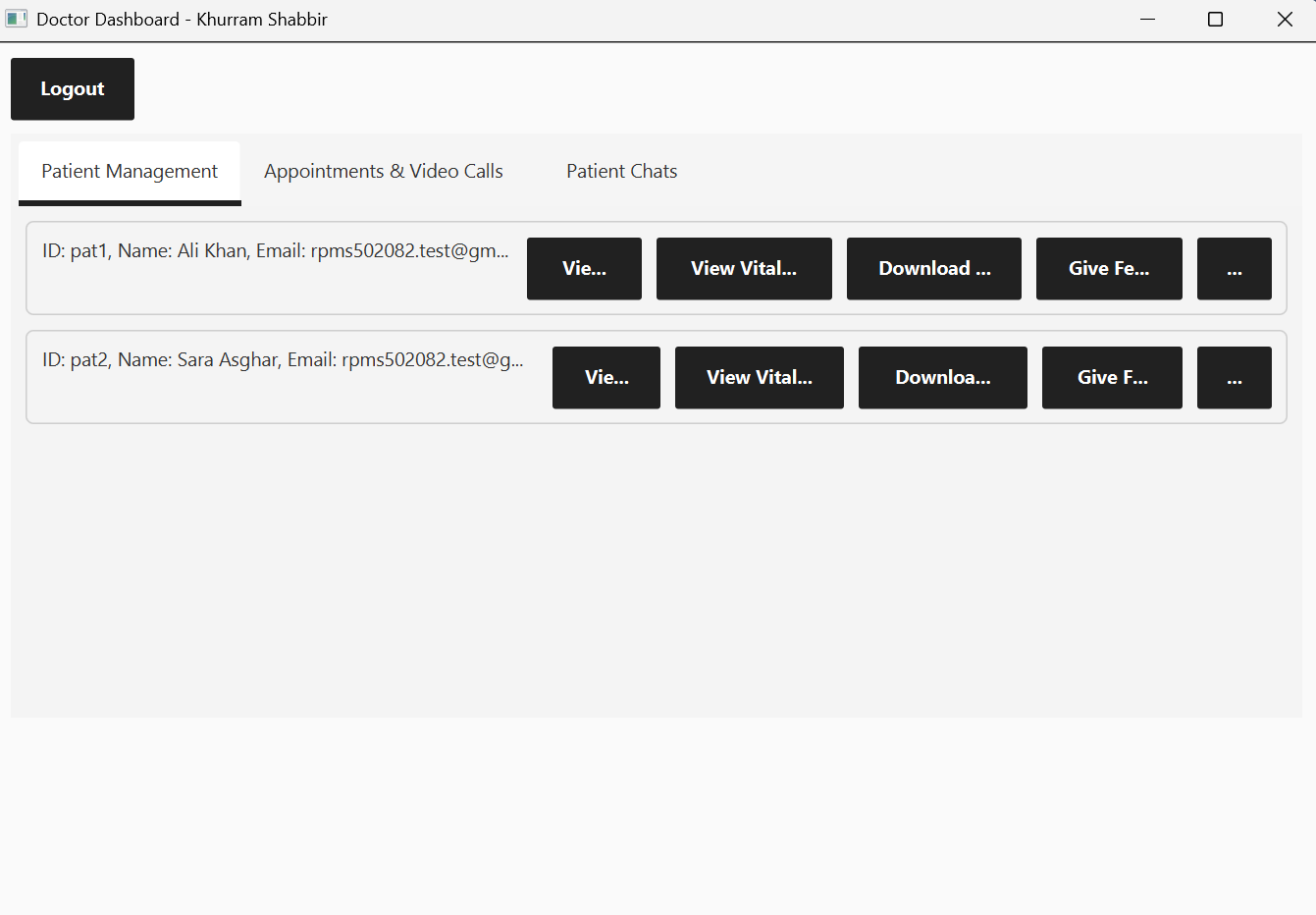


DOCTOR DASHBOARD NOW:  


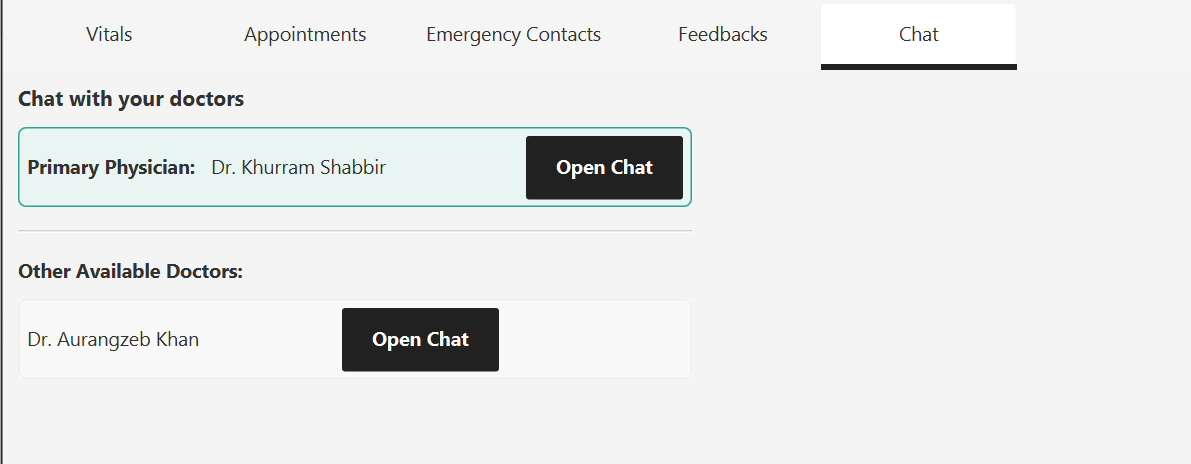


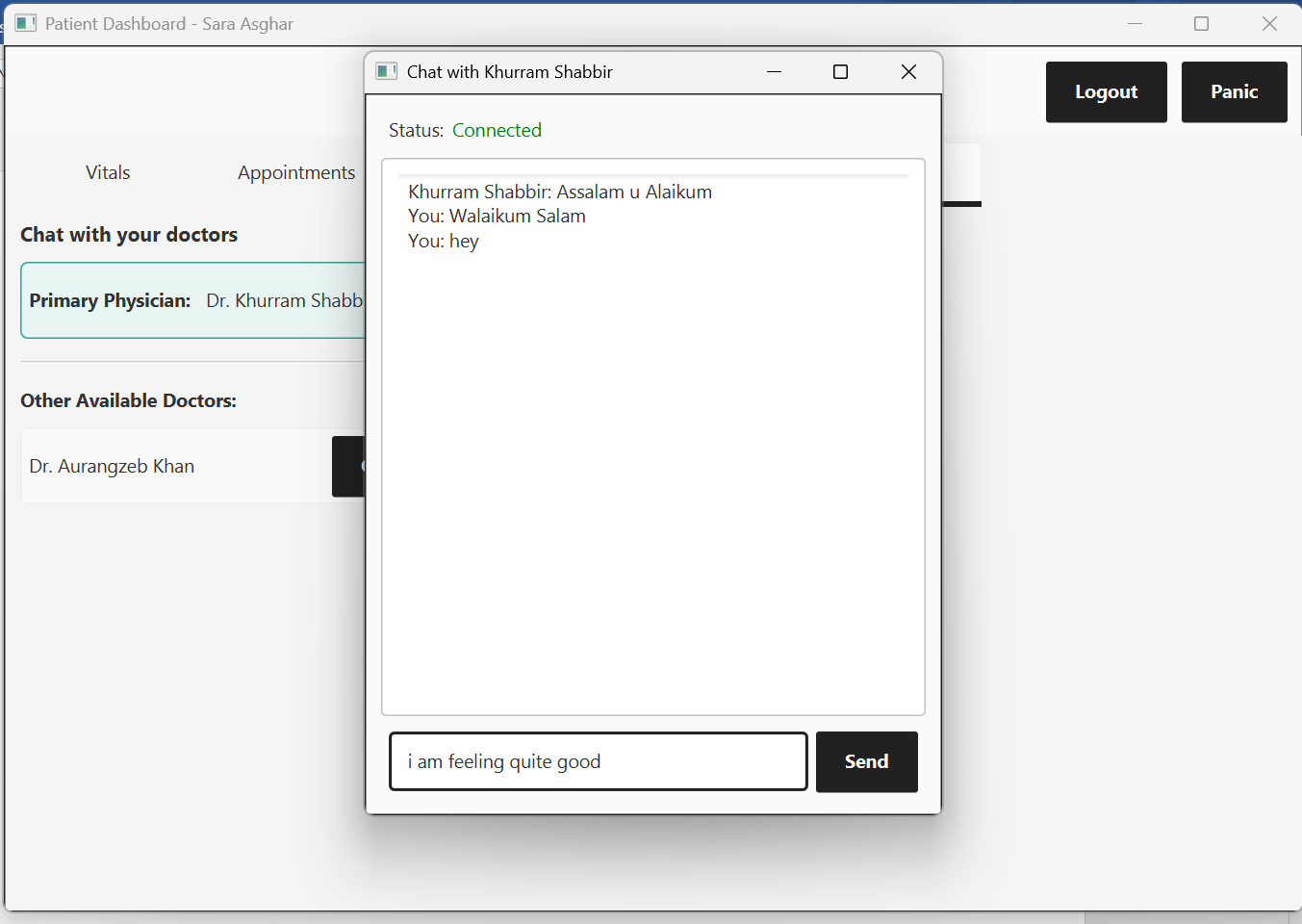
.

Whenever a doctor approves an appointment, that specific patient is added to the list of patients for that doctor:

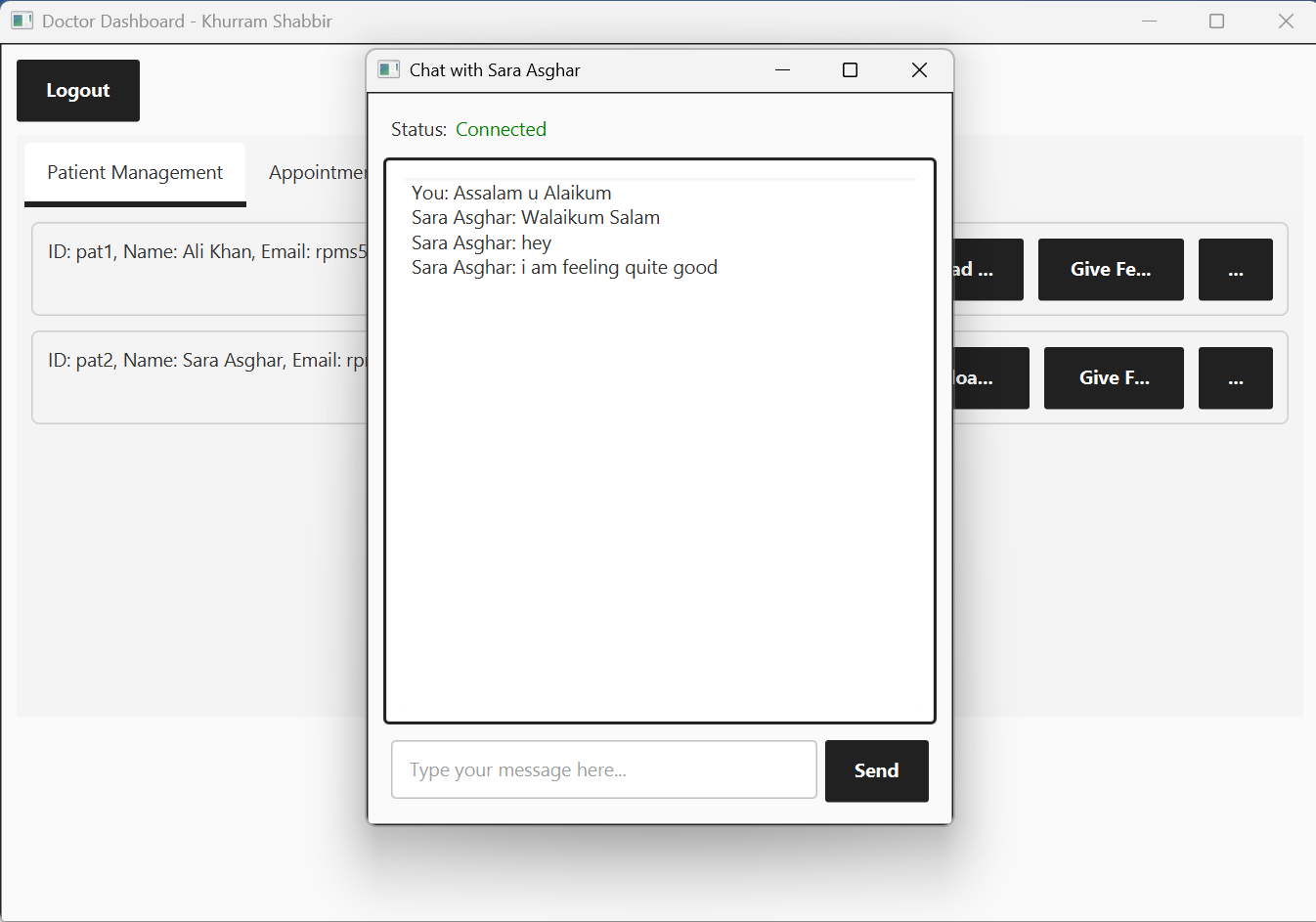


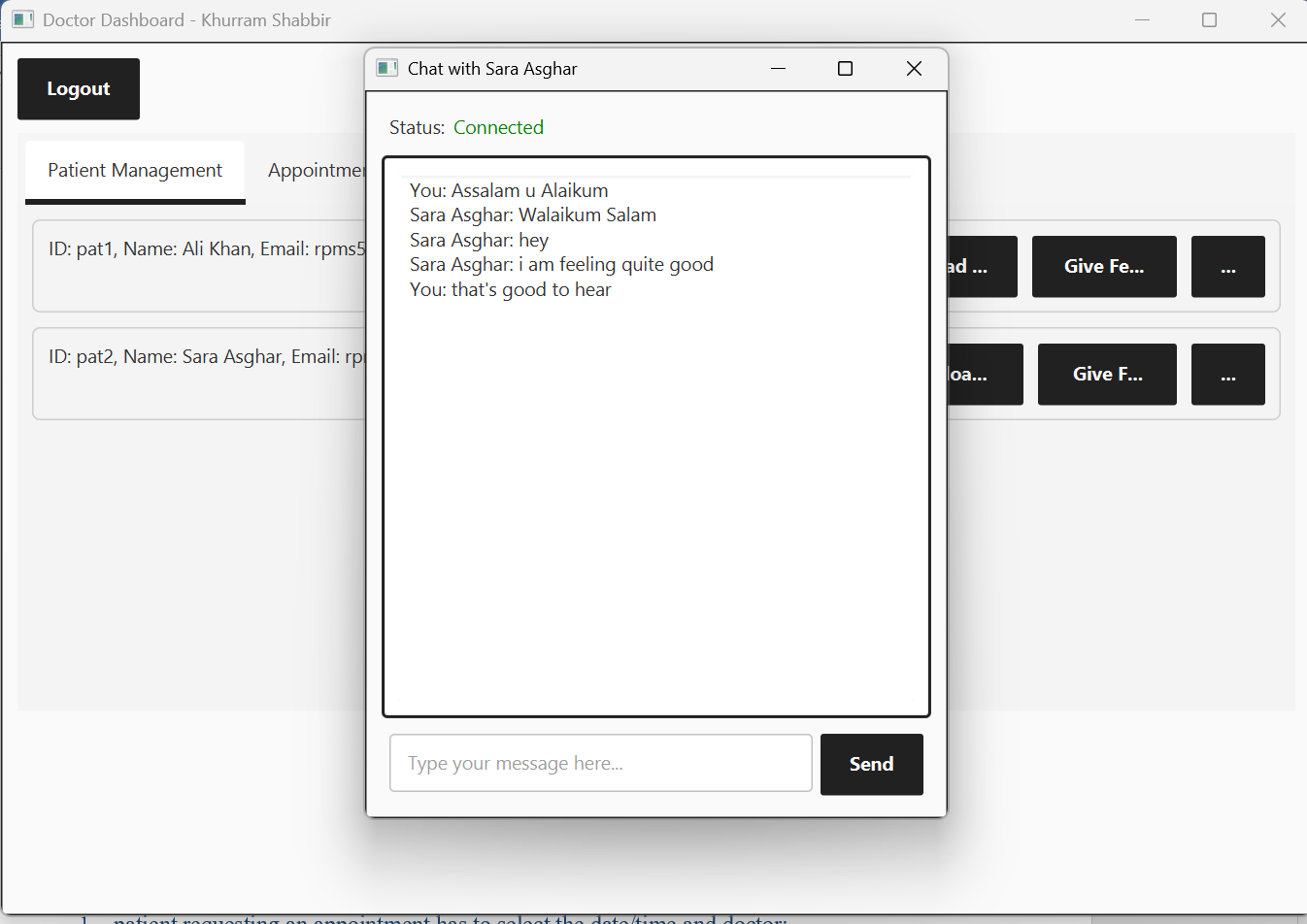
1. Chat Feature





Now, the doctor’s point of view:



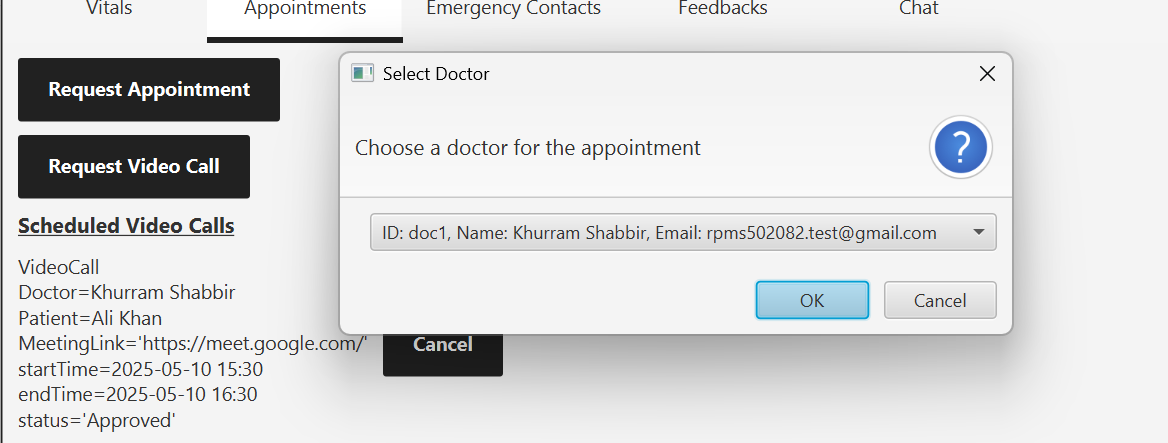


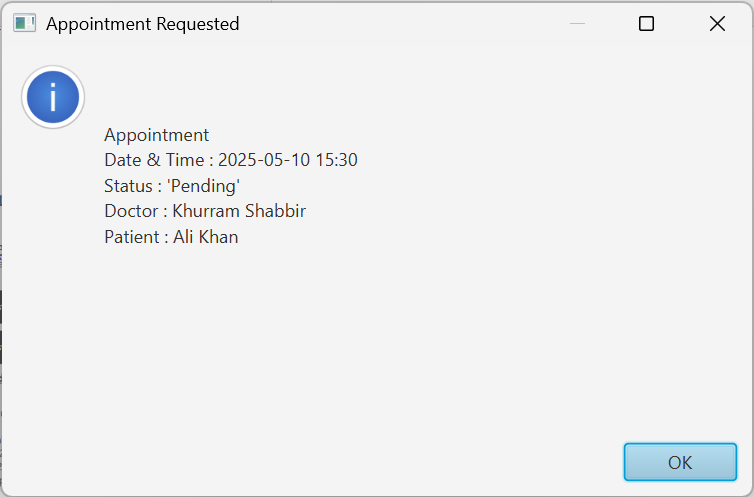
Chat Histories are also automatically loaded via object serialization.

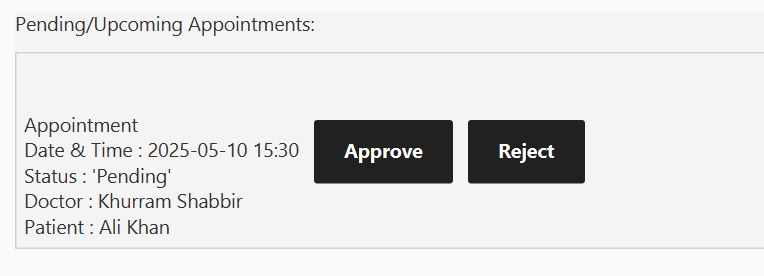
**2.7. Appointment Scheduling**

* Request and approval/rejection process.
* Display of confirmed appointments.

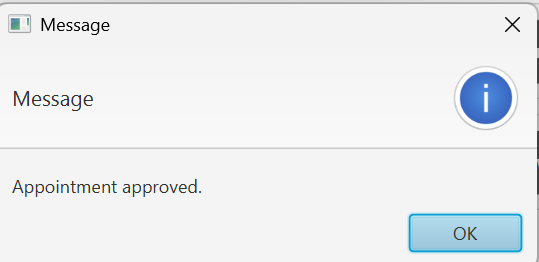
The same procedure as video calls.

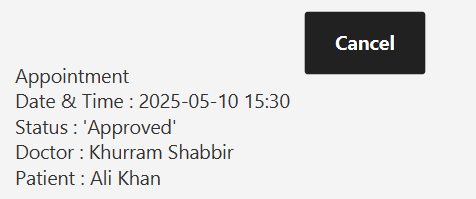
1. patient requesting an appointment has to select the date/time and doctor:  
   



Doctor’s Dashboard:  


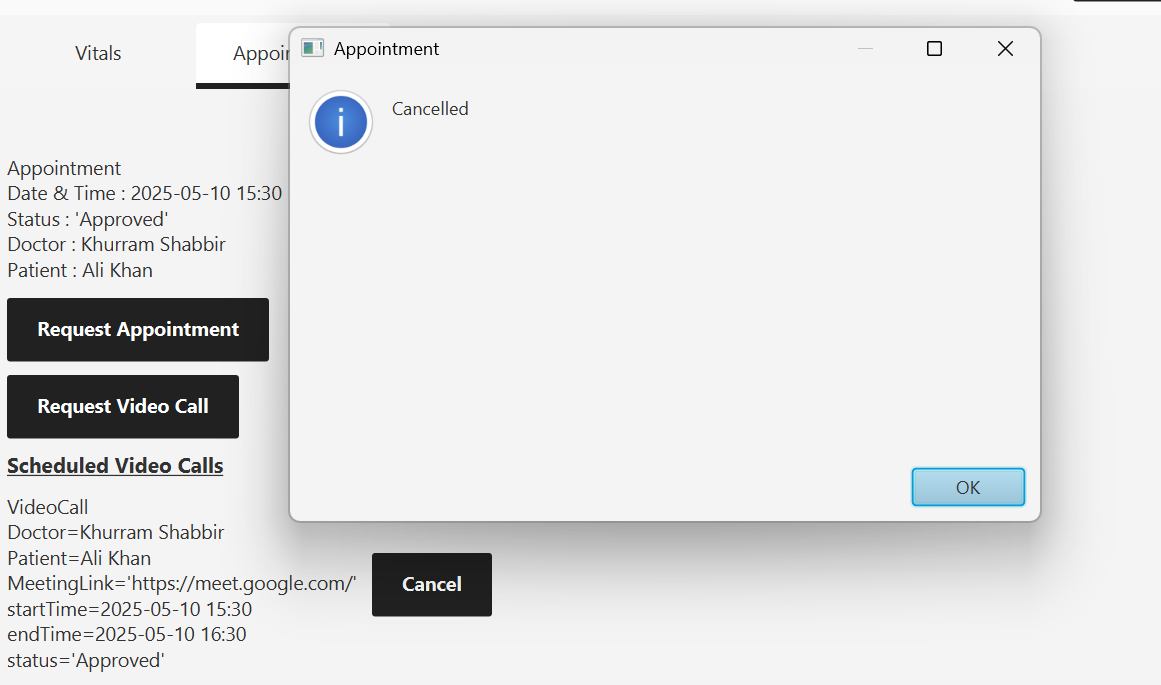
* 1. Approving

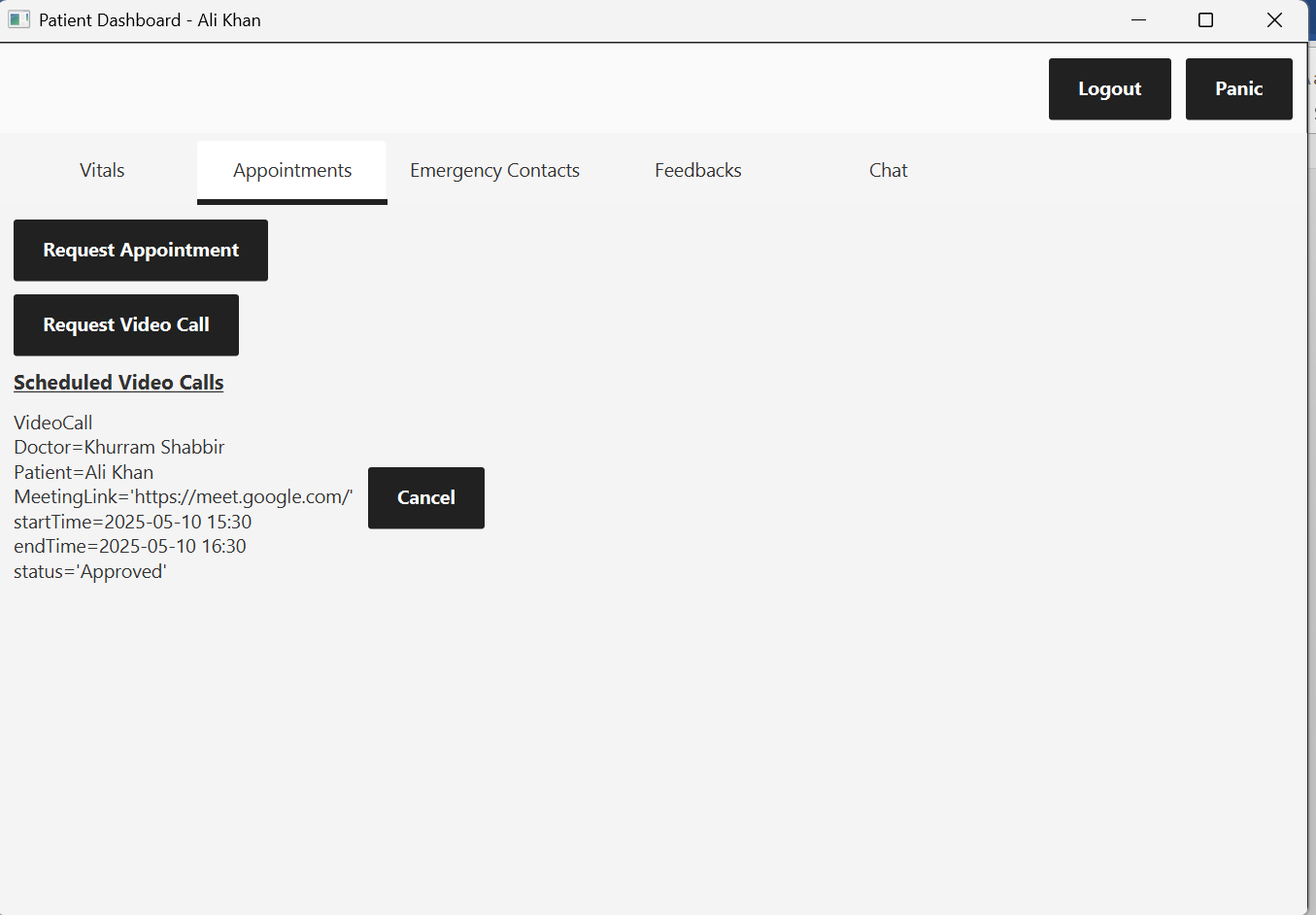




* 1. Rejecting

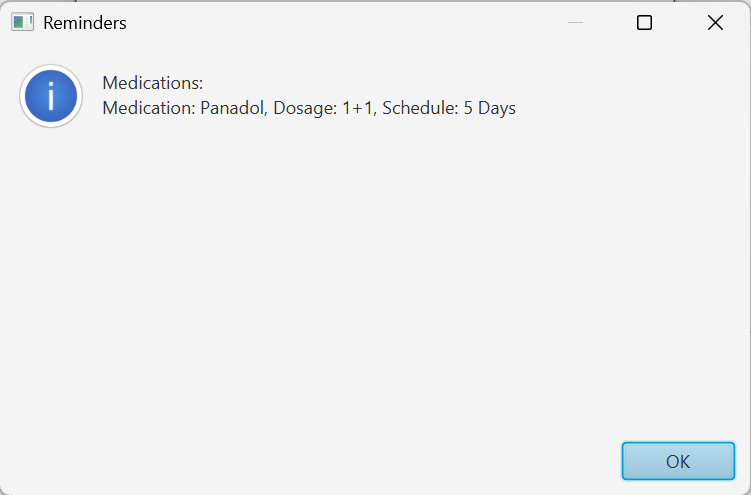
Either a patient can cancel an appointment or the doctor can reject/ cancel. In both cases, the appointment is removed from their tabs.



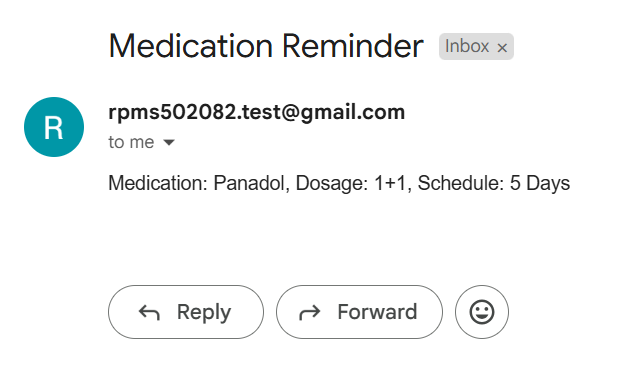


**2.8. Notifications & Reminders**

* Alerts for patients (medications).

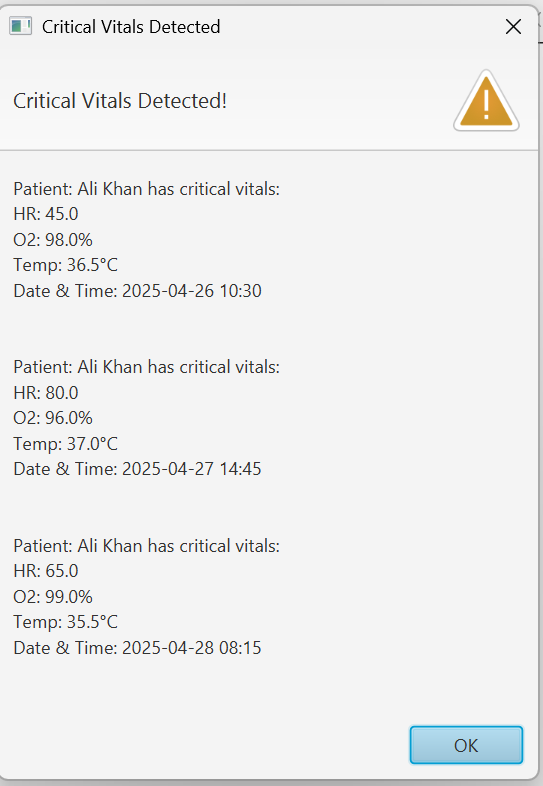


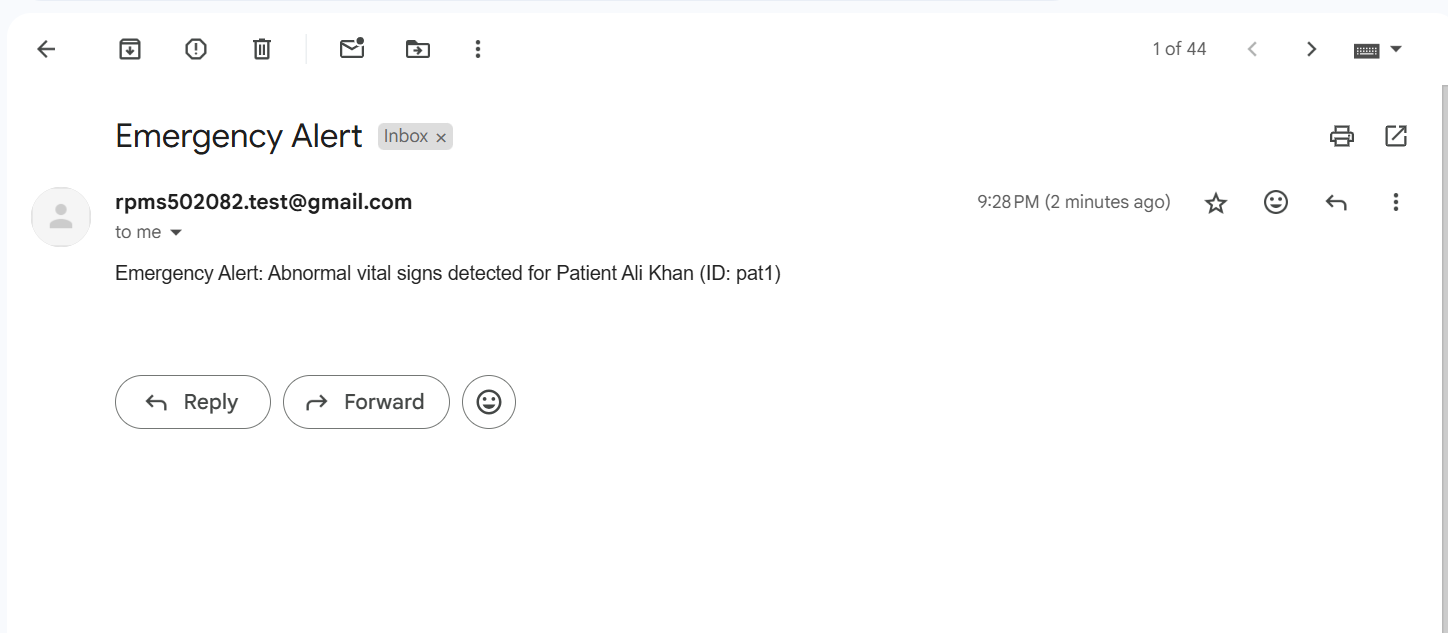
Patient receive medication and appointment reminders on dashboard and email whenever they login.



* Alerts for doctors (abnormal vitals).

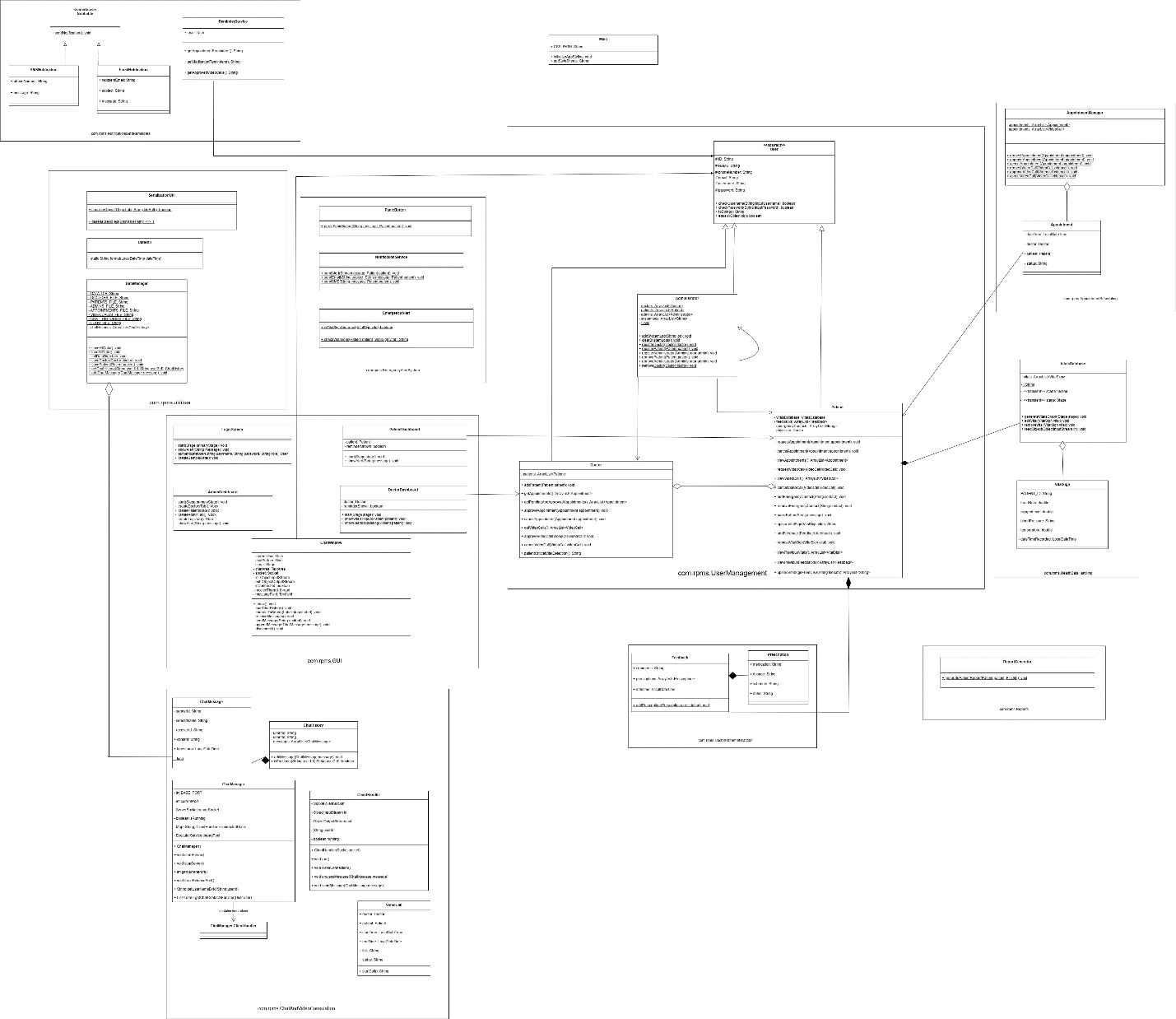
Doctors will also receive alerts on dashboard and email.

****



**3. System Diagrams**

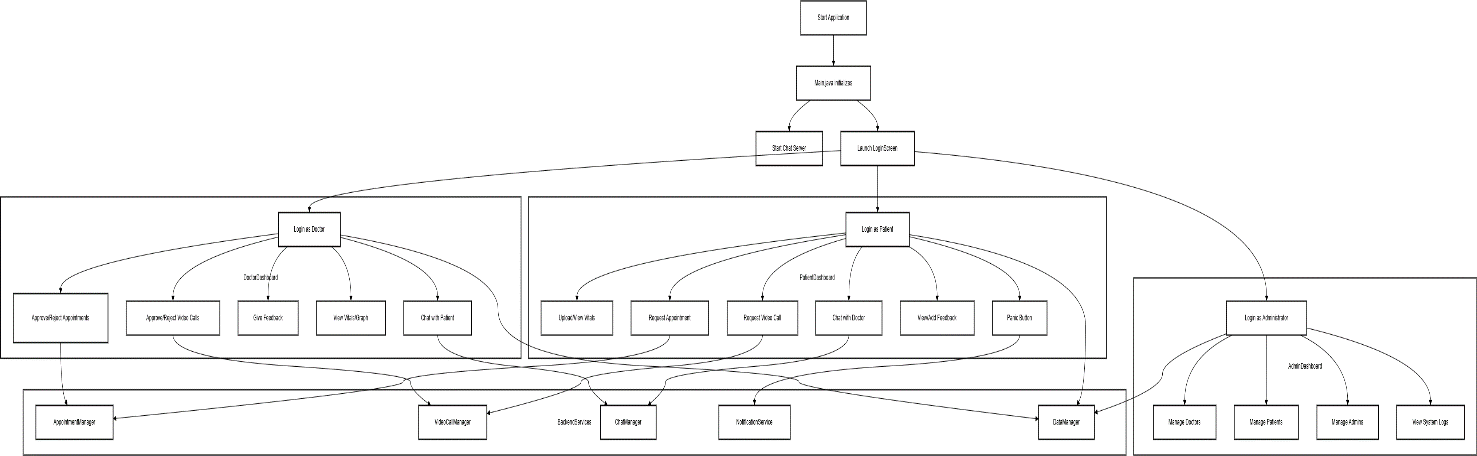
**3.1. UML Class Diagram**

****

**NOTE: THE DIAGRAMS MIGHT NOT BE PROPERLY VISIBLE HERE. SO, to get a detailed view.**

* 1. **Go to draw.io**
  2. **Open the diagram provided in the root directory of the project.**

**5.2. System Flowchart**



**4. Sample CSV files**

* Sample CSV file structure and sample content is provided in the following package:

“src/main/resources”