



**Name:** Sama Ayman Mokhtar Amin

# Healthcare Management System

## Table of Contents

1	Project overview .....	2
2	Project's UML .....	2
3	Screen Shots of the User Interface .....	5
3.1	Home page.....	5
3.2	After clicking on doctors.....	6
3.3	After clicking delete on a specific doctor object .....	7
3.4	After clicking edit on a specific doctor object .....	8
3.5	After clicking add new doctor .....	9
3.6	Error message because of missing information.....	10
3.7	After clicking on patients button .....	11
3.8	After clicking emergency patients .....	12
3.9	After clicking view a specific emergency patient .....	13
3.10	After clicking the appointments button .....	14
3.11	After clicking on neurologist .....	15
3.12	After selecting a specific doctor and clicking proceed.....	16

# 1 Project overview

This project stores a list of doctor's and patient's information and enables users to make appointments. Any changes that are made are stored in and retrieved whenever the program is reopened. Whenever the program runs, it reads information stored in a file and constructs objects filling them with this information. Then, so long as the program is on, the information stored in objects are modified by the user. Finally, as the user closes the program, all of information in the objects overwrites the information that was originally in the file. In the following three sections, we will explore the project's UML with a brief explanation, and the user interface. In the end of the document, there is an appendix with the project's code.

# 2 Project's UML

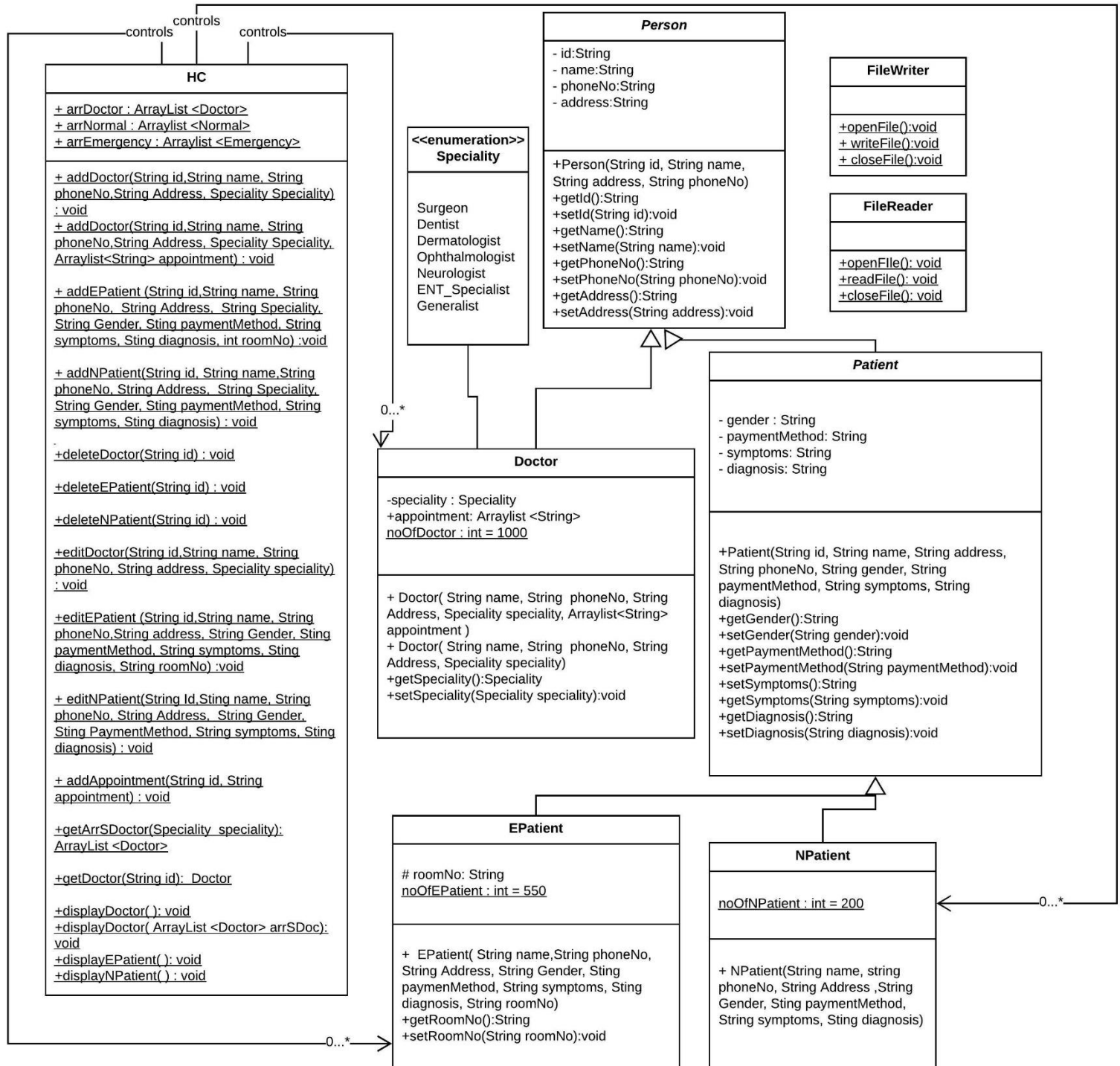
The UML diagram, shown below, has an abstract class called person. Two abstract classes inherit from it. The first is the doctor class. The second is another abstract class called patient. Two concrete classes inherit from patient. The first is n patient, which stands for non-emergency patient, while the second is e patient, which stands for emergency patient. This structure helps in avoiding code repetition. For instance, common data fields and their getters and setters are grouped in the parent classes. Moreover, this structure facilitates the use of polymorphism which is useful when drawing patient panes. These methods take patient objects and behave differently according to whether the object is instance of e patient or n patient.

There is an enum called specialty that enables us to categorize doctors and make it easier to deal with them. Besides, there is a class called HC which has all necessary function for viewing, adding, deleting, and editing doctor, e patient, or n patient. Also, it has the methods that deal with specialty and appointments. Finally, the HC class has, also, three array lists each of which store either the doctor, e patient, or n patient objects.

Moreover, there are two classes the file reader and the file writer which are mainly responsible for storing information.



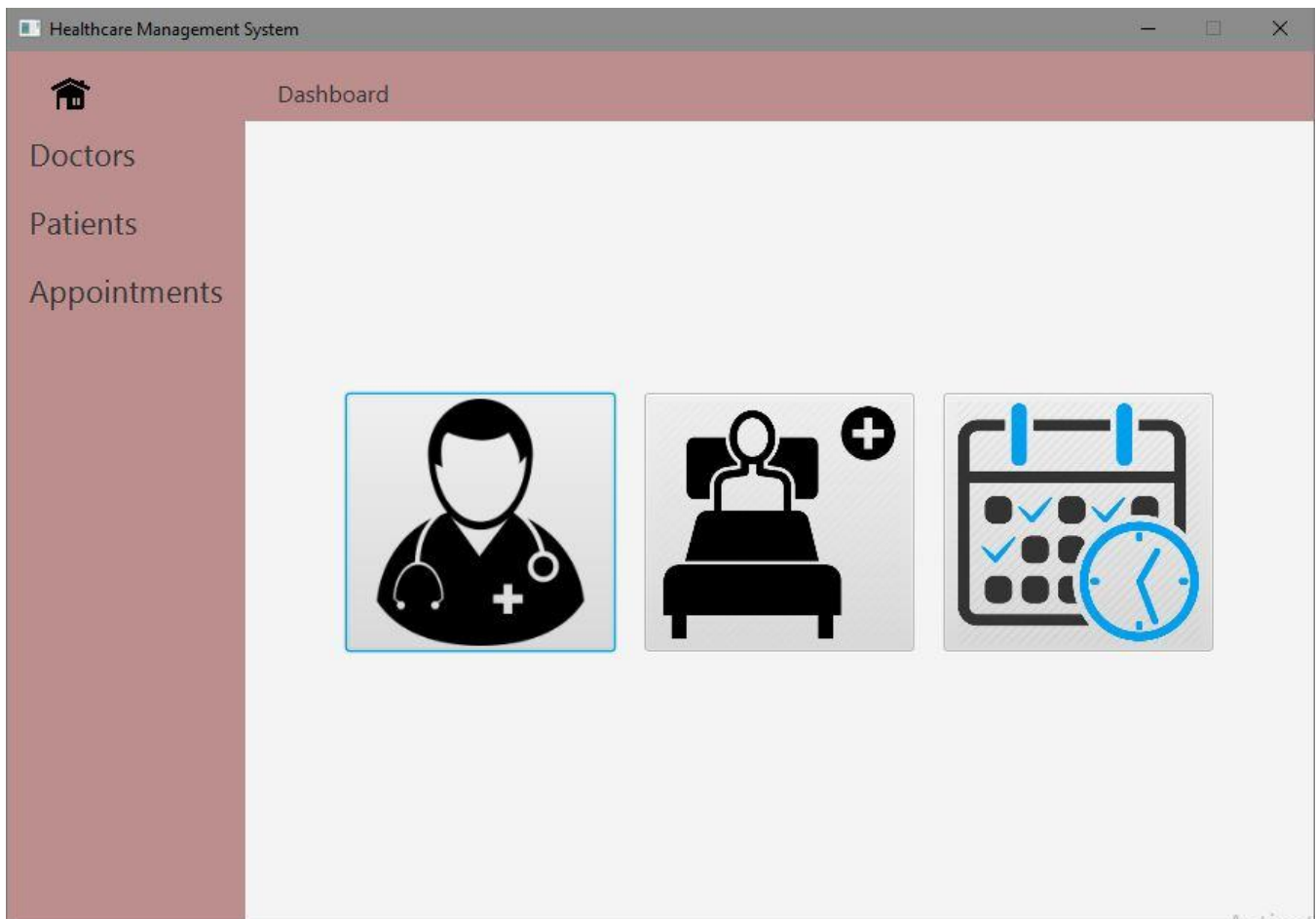
Lastly, the main class, which is not shown in the UML diagram, has the javafx code that is responsible for the GUI. The main class has three inner classes. The doctorButtonFX, patientButtonFX, and the appointmentButtonFX. Each of these inner classes group all the functionalities related to that specific button. Some panes- as the side pane, the top pane, and the border pane that holds all them- are not related to a specific button thus it is placed inside the main class but outside any of the three inner classes.



## 3 Screen Shots of the User Interface

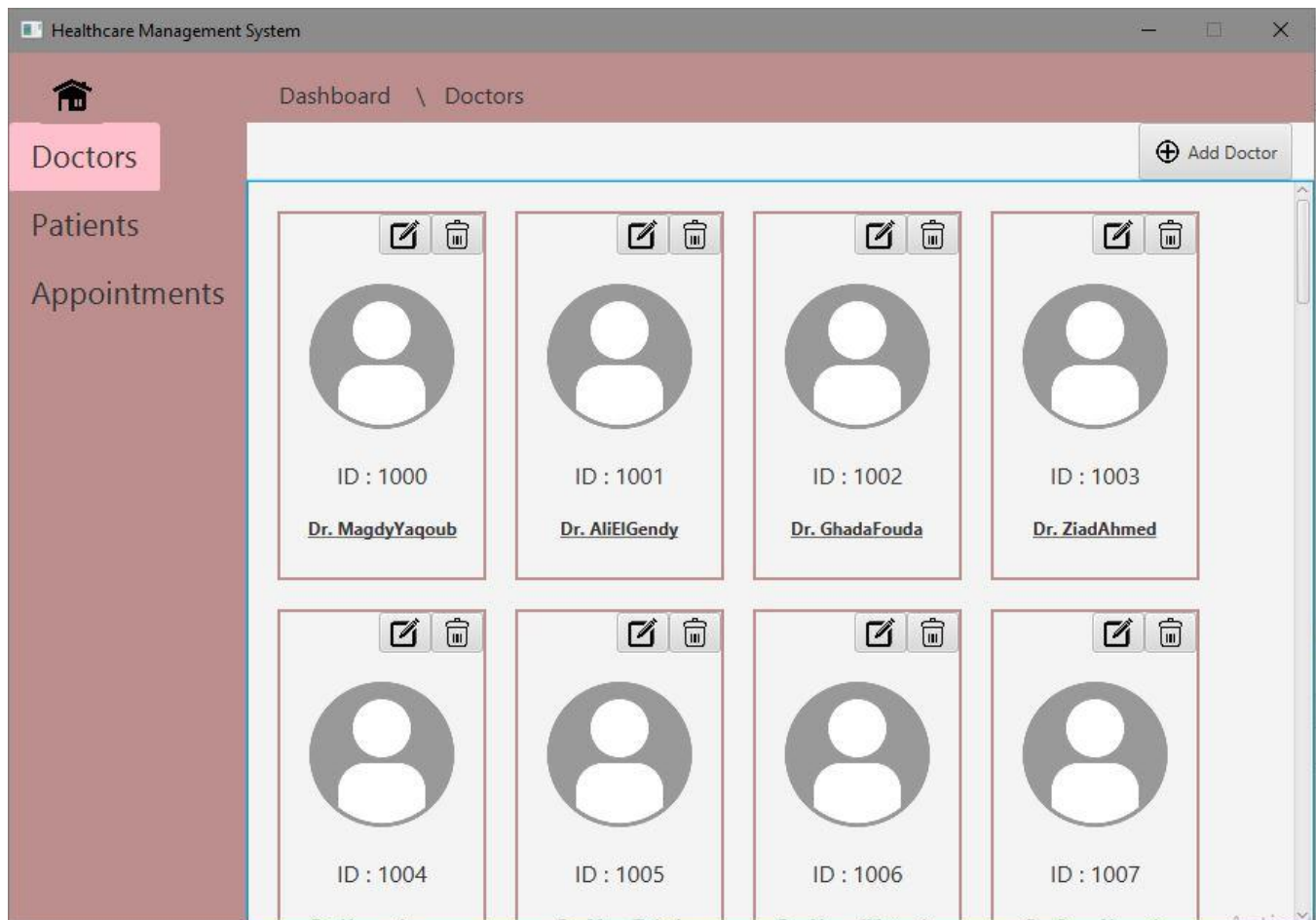
### 3.1 Home page

There are three buttons that can be accessed: doctors, patients, and appointments. The next screen shots explore each of them. The top menu helps in tracking your clicks.

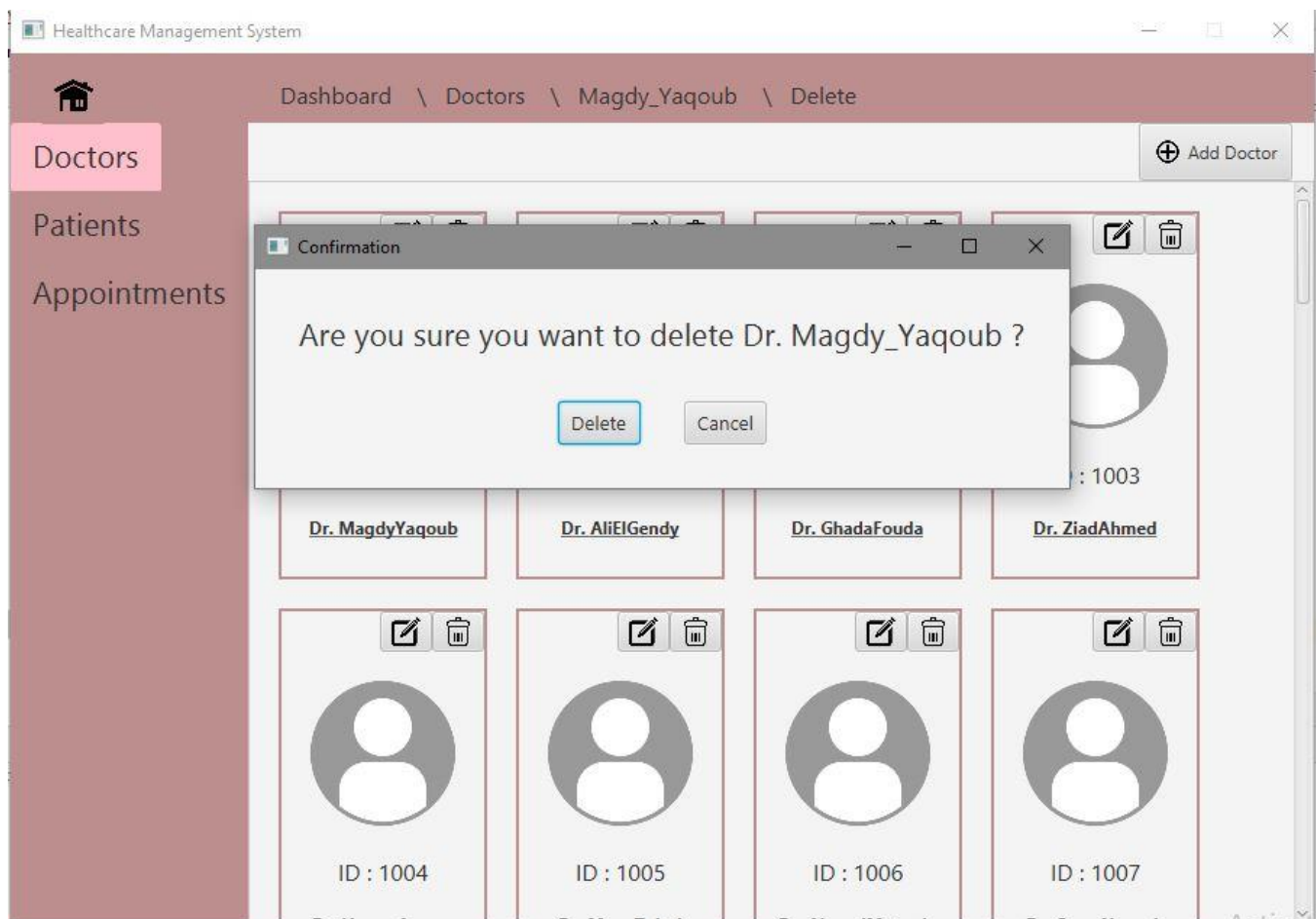


## 3.2 After clicking on doctors

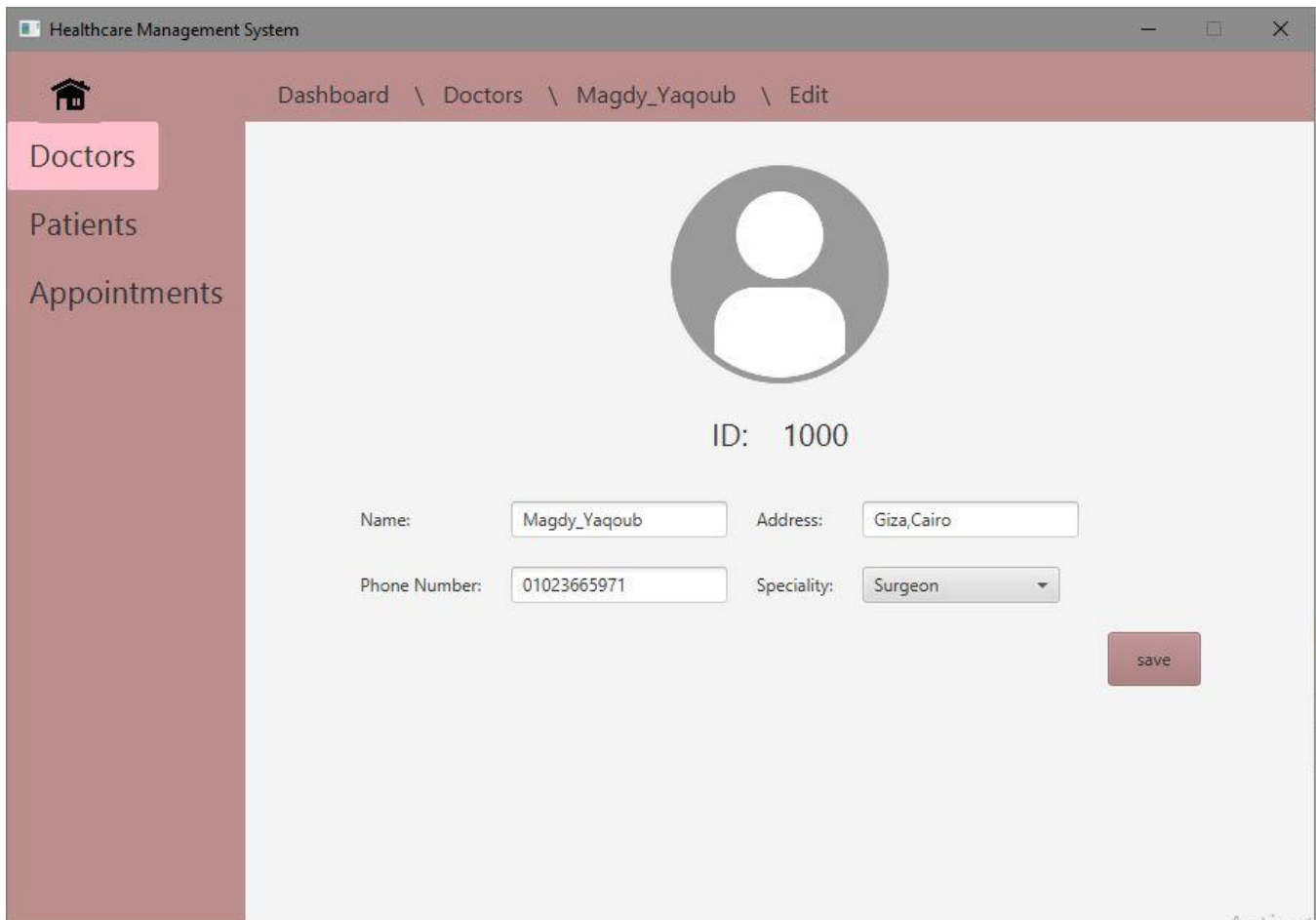
There are several buttons that can be clicked. **Add Doctor** that is at the top right. Besides the **edit and delete** at the top right of each doctor object pane. Lastly, the **underlined name** is clickable and views that specific doctor's information.



### 3.3 After clicking delete on a specific doctor object



### 3.4 After clicking edit on a specific doctor object



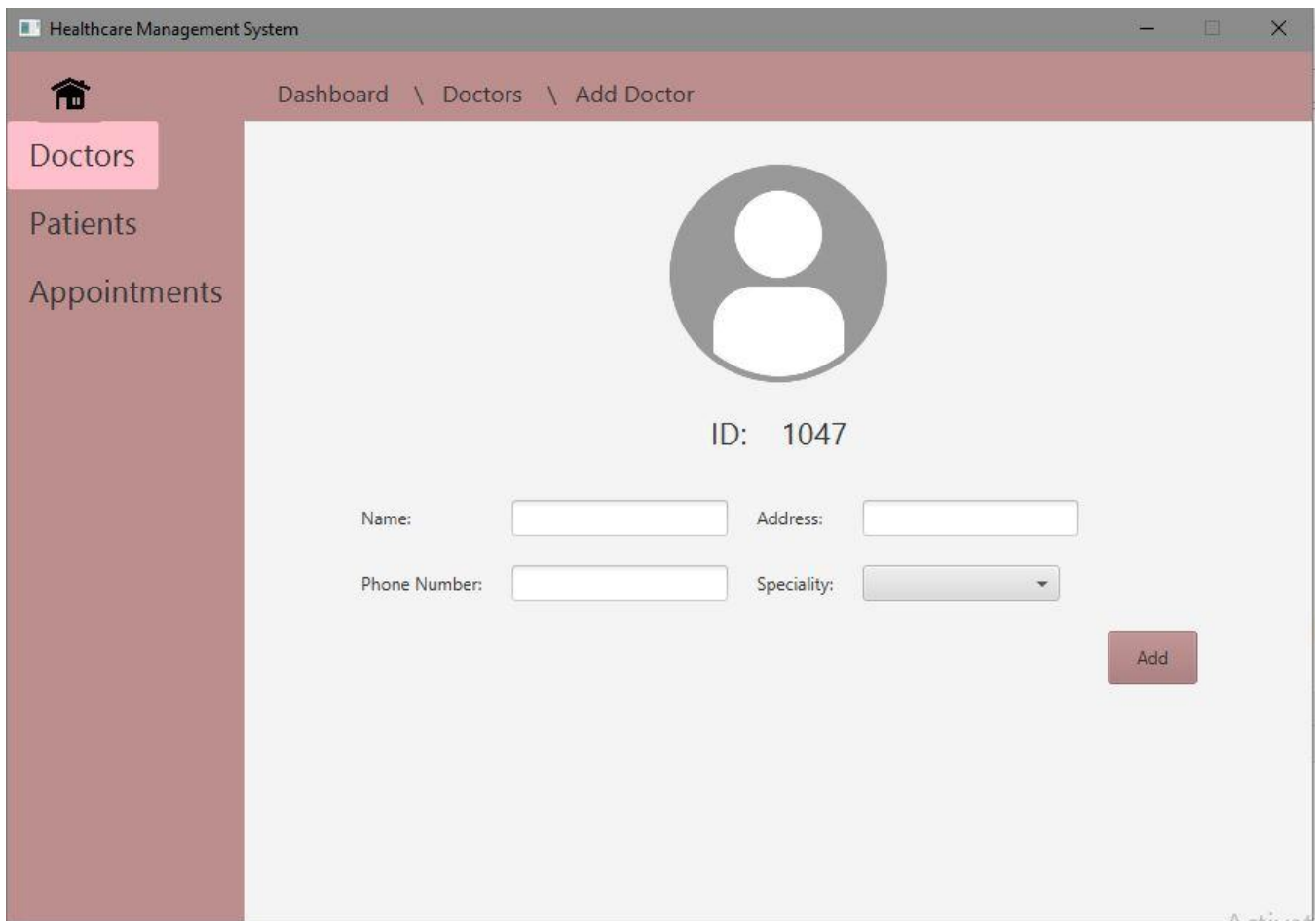
The screenshot shows a web application window titled "Healthcare Management System". The breadcrumb navigation at the top reads "Dashboard \ Doctors \ Magdy\_Yaqoub \ Edit". On the left, a sidebar contains a home icon and three menu items: "Doctors" (highlighted in pink), "Patients", and "Appointments". The main content area features a large circular placeholder for a profile picture, with the text "ID: 1000" below it. The form contains four input fields: "Name" with the value "Magdy\_Yaqoub", "Address" with the value "Giza,Cairo", "Phone Number" with the value "01023665971", and "Speciality" with a dropdown menu showing "Surgeon". A "save" button is located at the bottom right of the form.

Name:	Magdy_Yaqoub	Address:	Giza,Cairo
Phone Number:	01023665971	Speciality:	Surgeon

save



### 3.5 After clicking add new doctor



The screenshot shows a web application window titled "Healthcare Management System". The breadcrumb navigation at the top reads "Dashboard \ Doctors \ Add Doctor". On the left, a sidebar contains a home icon and three menu items: "Doctors" (highlighted in pink), "Patients", and "Appointments". The main content area features a large grey circular placeholder for a profile picture, with the text "ID: 1047" below it. The form includes four input fields: "Name:", "Address:", "Phone Number:", and "Speciality:". The "Speciality:" field is a dropdown menu. An "Add" button is located at the bottom right of the form area.

Healthcare Management System

Dashboard \ Doctors \ Add Doctor

Doctors

Patients

Appointments

ID: 1047

Name:

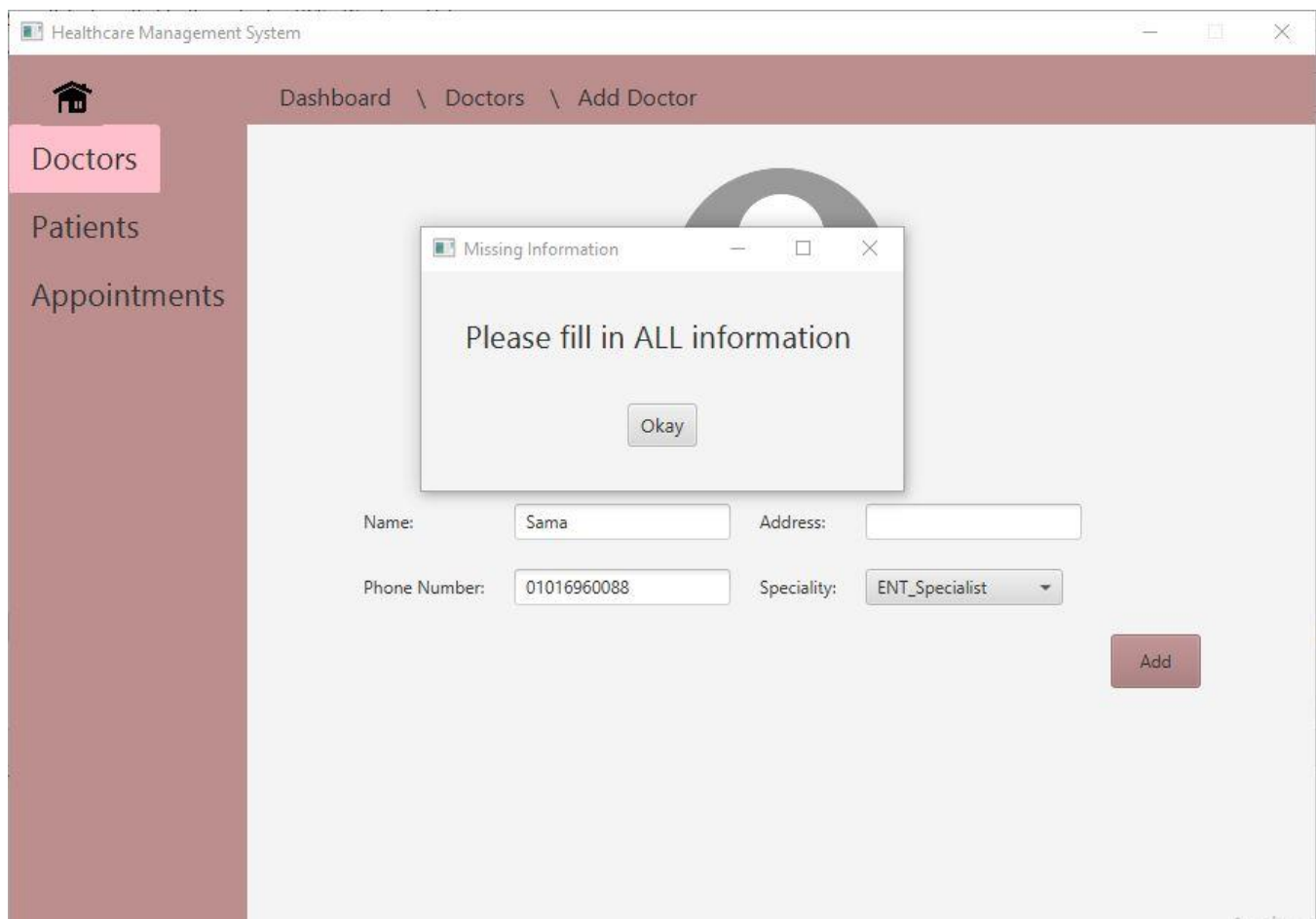
Address:

Phone Number:

Speciality:

Add

### 3.6 Error message because of missing information



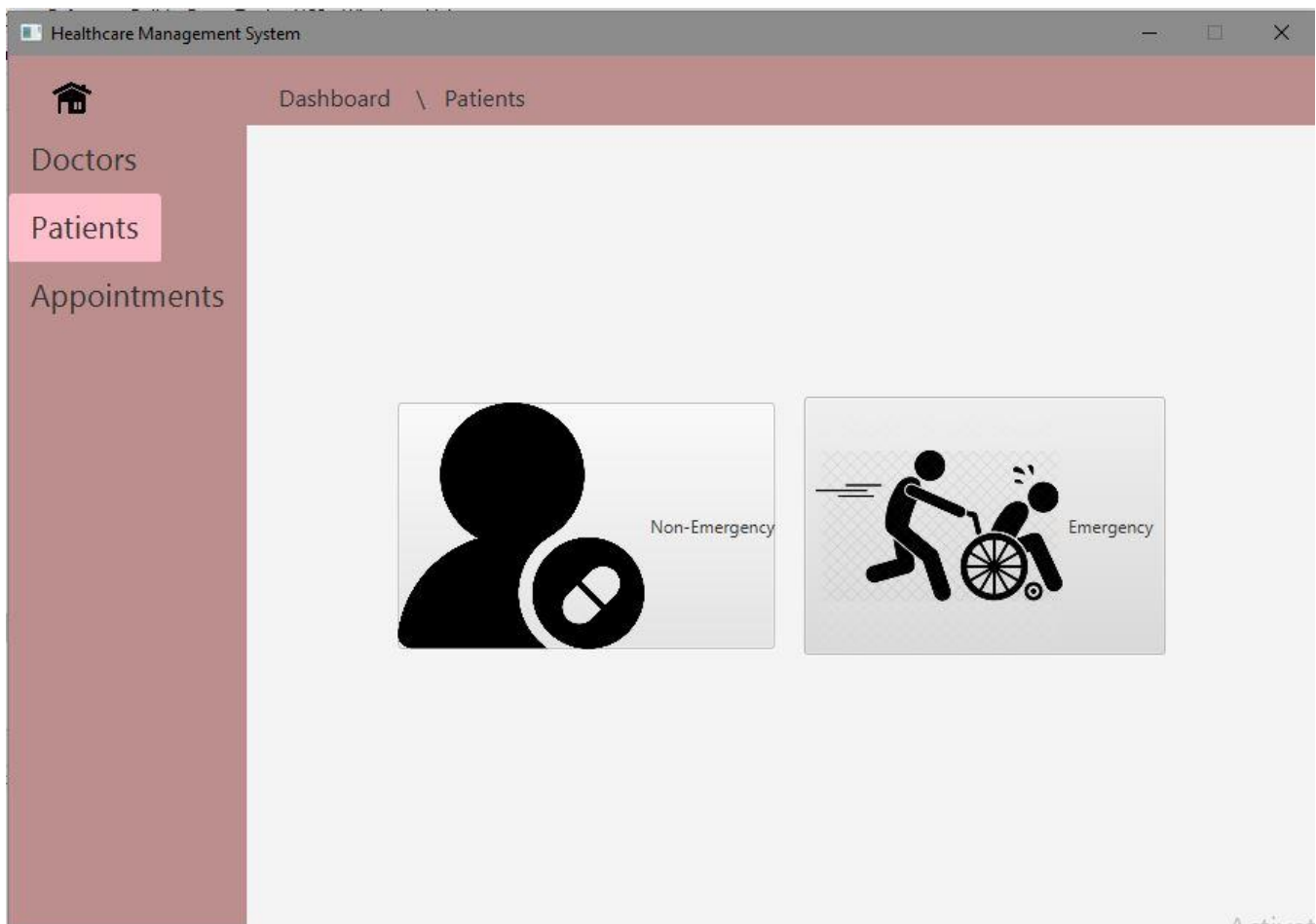
The screenshot displays a web application titled "Healthcare Management System". The navigation menu on the left includes "Doctors", "Patients", and "Appointments", with "Doctors" currently selected. The breadcrumb trail at the top indicates the path: "Dashboard \ Doctors \ Add Doctor". A modal dialog box titled "Missing Information" is centered on the screen, displaying the message "Please fill in ALL information" and an "Okay" button. Below the dialog, the "Add Doctor" form is visible, containing the following fields:

Field	Value
Name:	Sama
Address:	
Phone Number:	01016960088
Speciality:	ENT_Specialist

An "Add" button is located at the bottom right of the form.

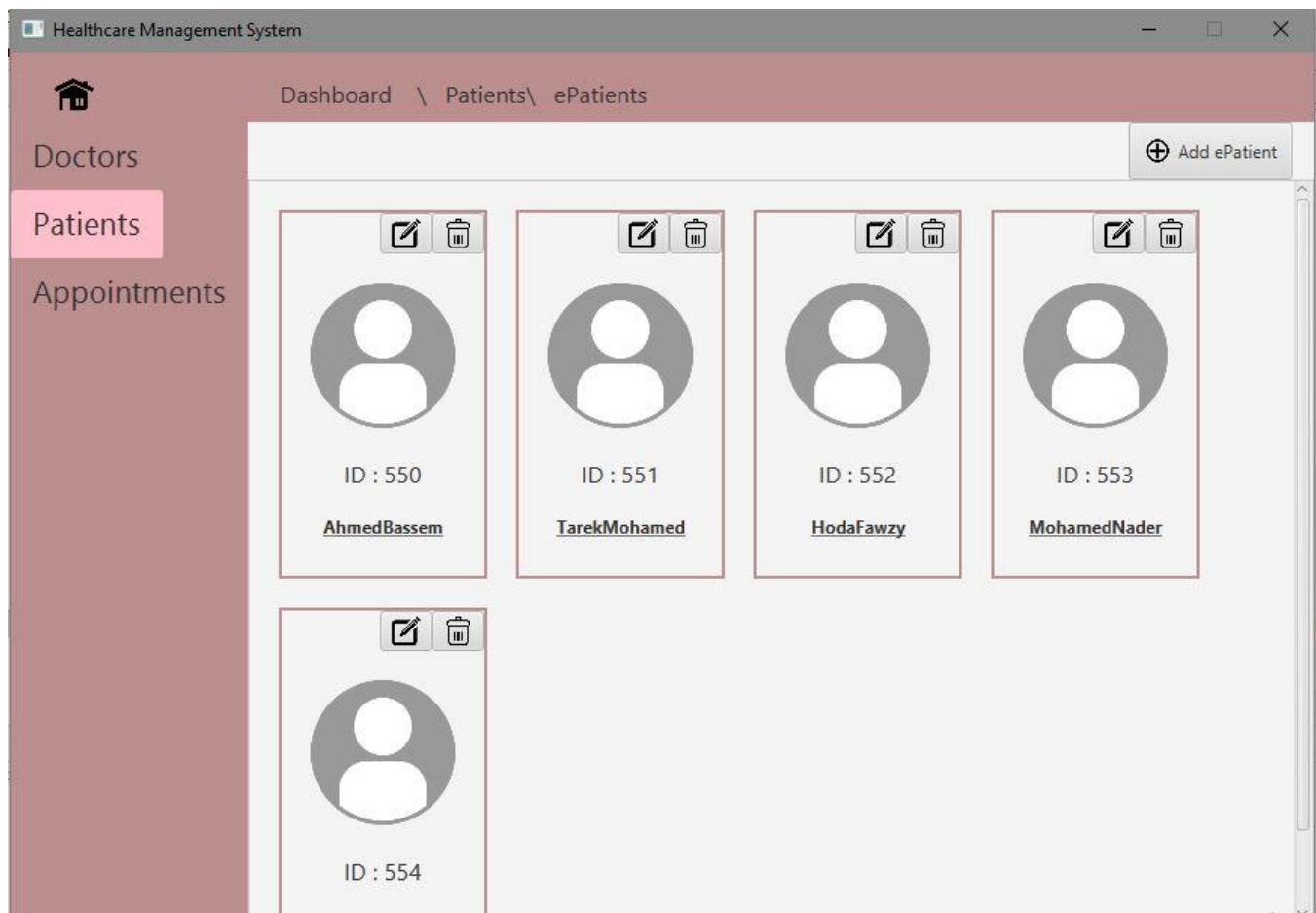
### 3.7 After clicking on patients button

There are two options, emergency patients and normal patients.



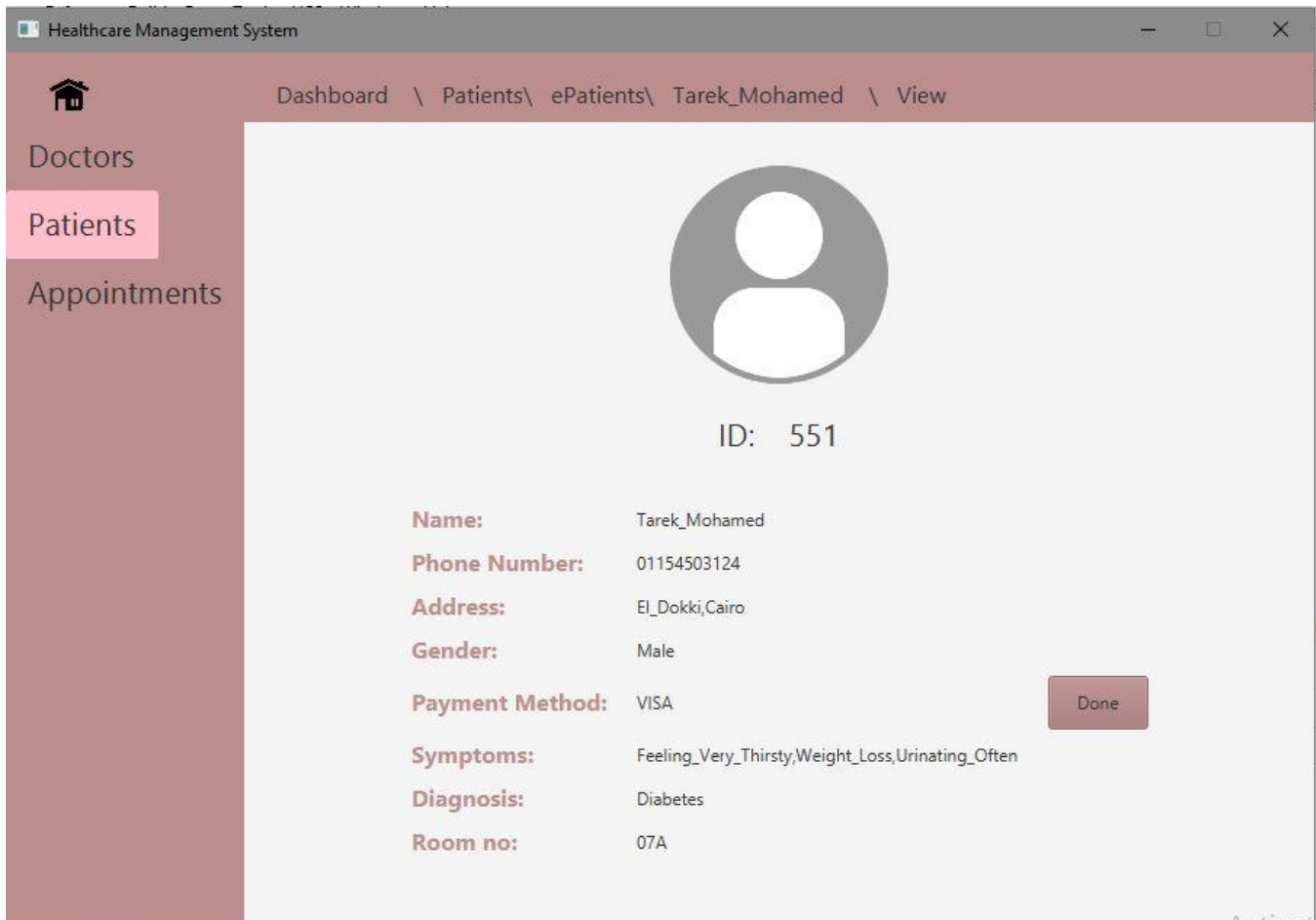
### 3.8 After clicking emergency patients

This window has the same functionalities of the doctor window except that the data stored in each patient object differs from that of the doctor.



### 3.9 After clicking view a specific emergency patient

The same functionalities apply here and to normal patients. However, I show here only this screen shot for convenience.



The screenshot displays the 'Healthcare Management System' interface. The top navigation bar includes 'Dashboard', 'Patients', 'ePatients', 'Tarek\_Mohamed', and 'View'. The left sidebar contains 'Doctors', 'Patients' (highlighted), and 'Appointments'. The main content area shows a patient profile for 'Tarek\_Mohamed' with ID 551. The profile includes a placeholder for a profile picture and a list of details: Name, Phone Number, Address, Gender, Payment Method, Symptoms, Diagnosis, and Room no. A 'Done' button is located next to the Payment Method field.

<b>Name:</b>	Tarek_Mohamed
<b>Phone Number:</b>	01154503124
<b>Address:</b>	El_Dokki,Cairo
<b>Gender:</b>	Male
<b>Payment Method:</b>	VISA
<b>Symptoms:</b>	Feeling_Very_Thirsty,Weight_Loss,Urinating_Often
<b>Diagnosis:</b>	Diabetes
<b>Room no:</b>	07A

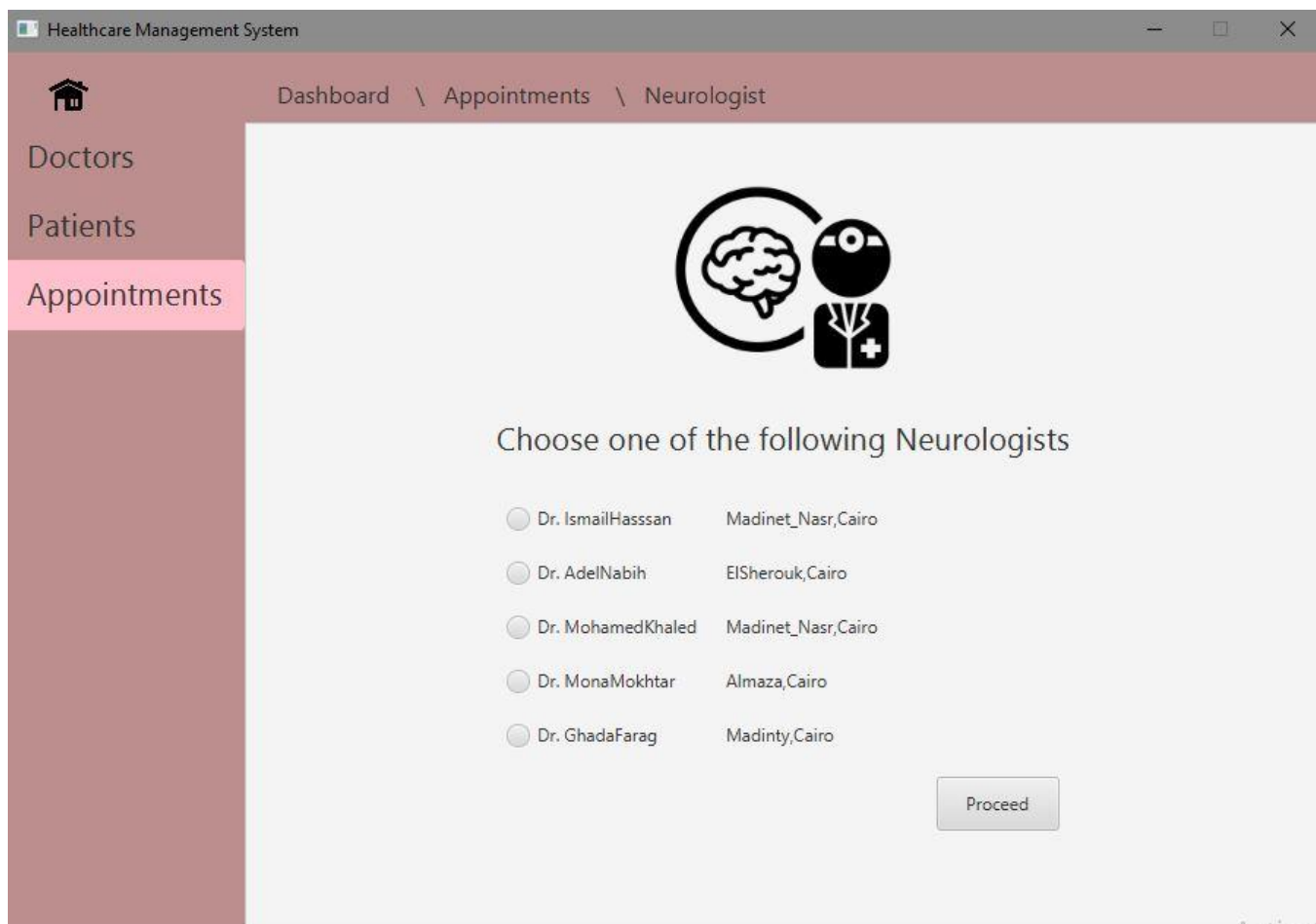
### 3.10 After clicking the appointments button

In this window, the user is prompted to choose the specialty that best suits their illness.



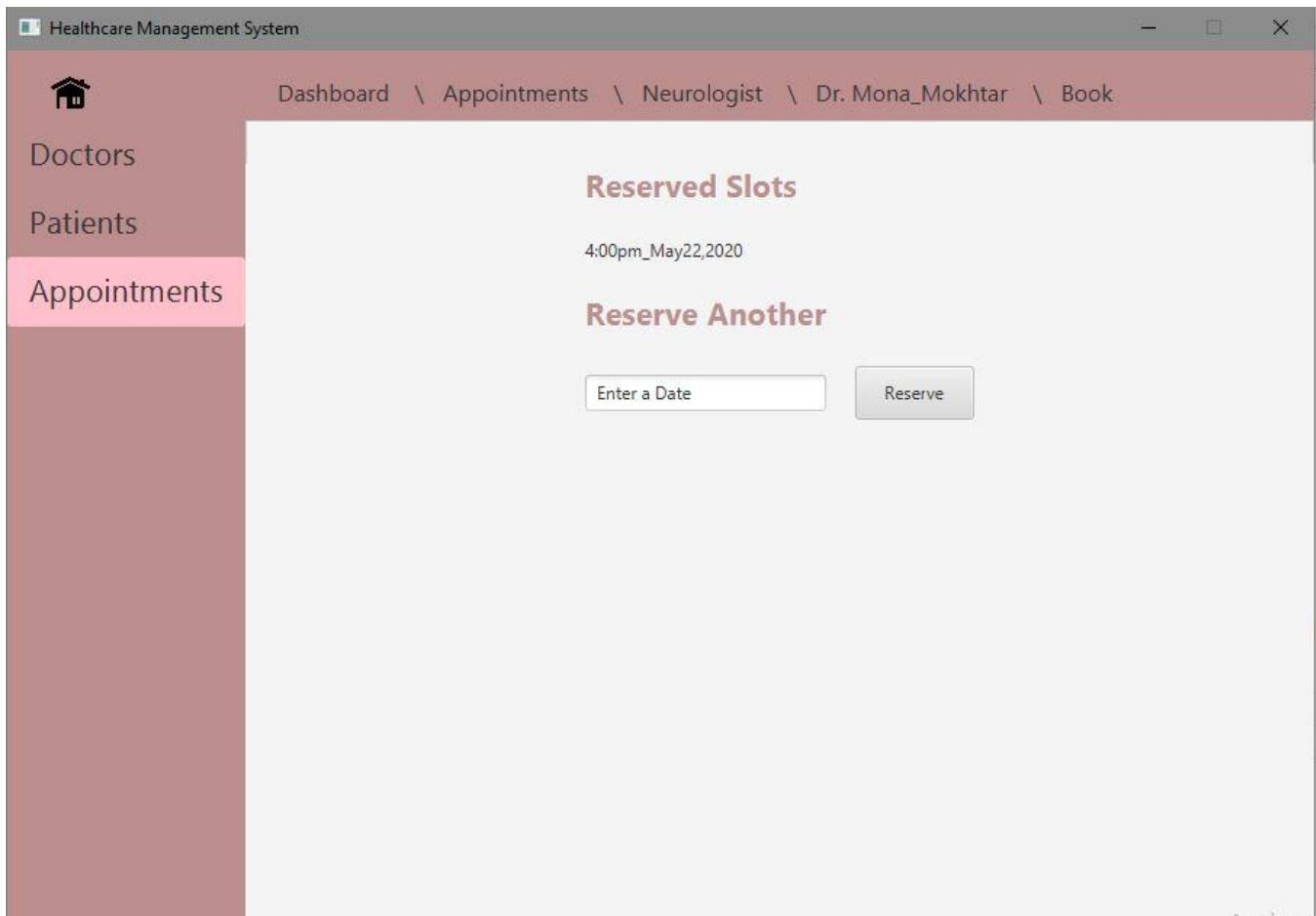
### 3.11 After clicking on neurologist

A list of all neurologists that are stored on the system show up. The list is composed of several doctors and next to each is their address so that the user can choose the address that is closest to them. Each of the other specialties have the same functionality but the neurologist is chosen as an example.



### 3.12 After selecting a specific doctor and clicking proceed

The reserved slots show up and the user is prompted to choose a time that suits them then click reserve.



The screenshot displays the 'Healthcare Management System' interface. The top navigation bar includes 'Dashboard', 'Appointments', 'Neurologist', 'Dr. Mona\_Mokhtar', and 'Book'. The left sidebar contains 'Doctors', 'Patients', and 'Appointments', with 'Appointments' highlighted. The main content area shows 'Reserved Slots' for '4:00pm\_May22,2020'. Below this, there is a 'Reserve Another' section with an 'Enter a Date' input field and a 'Reserve' button.