Project

Hospital Management System

Sergei Rogov

U231N0051

Scenario - this application is designed to be used by 3 hospital administrators to manage the number of patients in each hospital room. When administrator (user) runs a program, he might only be the only logged user on his machine. He might login to another account as well, but one at a time.

Application instantiates General Hospital with 3 floors.

There are 3 types of rooms in this hospital: Medical, Intensive Care and Operating Room.

First floor has 5 rooms, second and third have 4 rooms each.

There are 4 doctors in this hospital: House, Forman, Cameron and Chase.

Main feature: this application shows the info about all patients located in each of hospital rooms.

When adding a new patient, he is assigned a room and a doctor.

Room is assigned by 2 criteria: 1) the type of room should suit patient needs. 2) patient is assigned a room with smallest number of patients among all the rooms of chosen type.

Doctor is assigned randomly from the set of 4 doctors.

How to use the Hospital Management System application:

In order to login, you might use one of 3 possible login – password passwords:

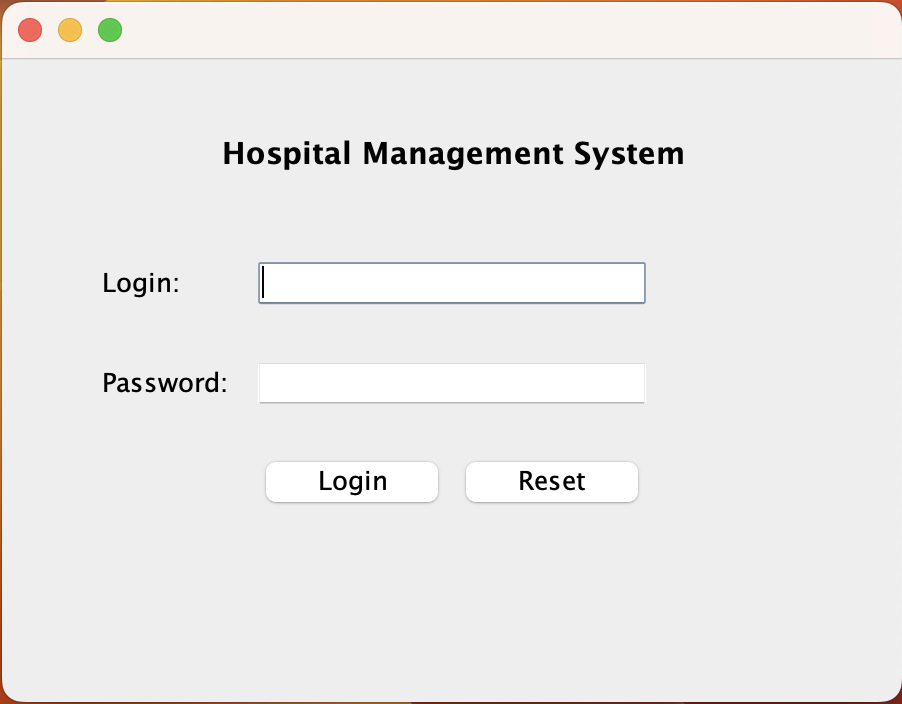
"Admin1", "qwerty"

"Admin2", "abcdefg"

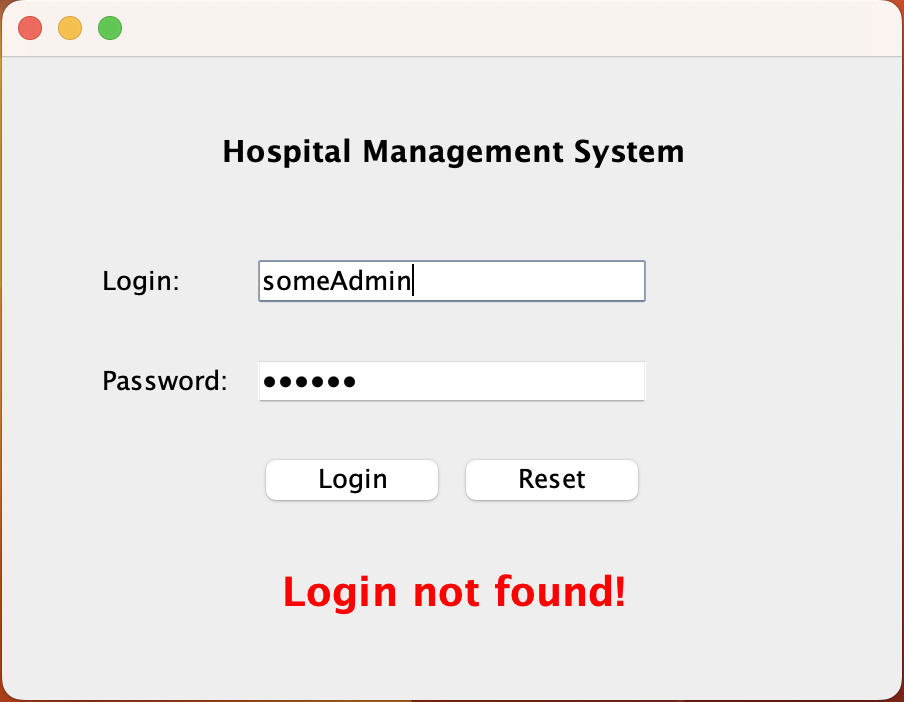
"Admin3", "123passcode"

Scenario – only 3 administrators are intitled to use the Hospital Management System, so each of them is given a login and corresponding password.

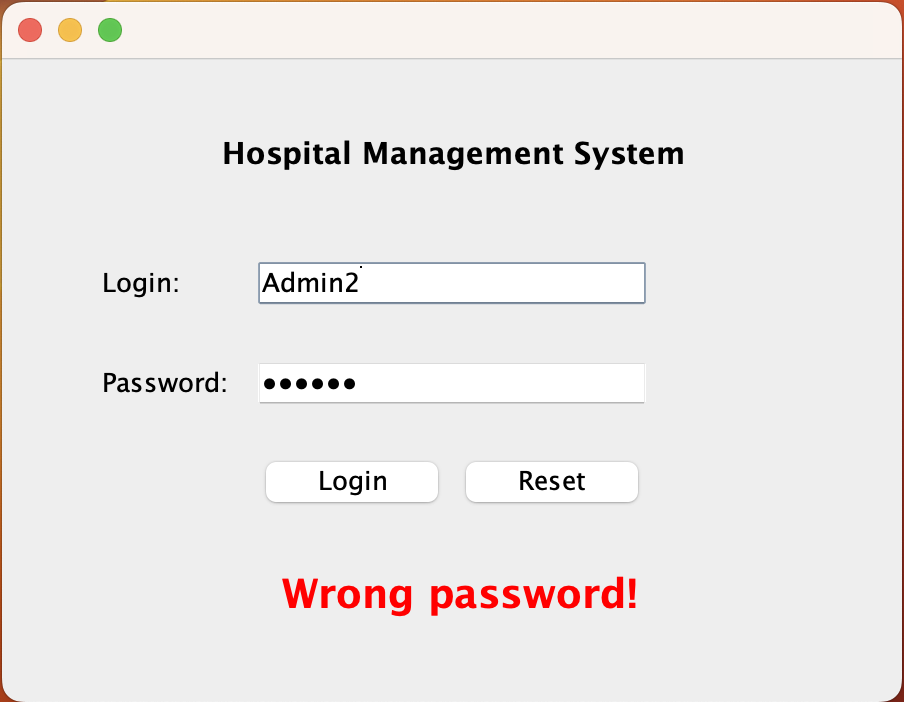
Application welcomes you with a login page:



If login is not found:



If password is not correct:

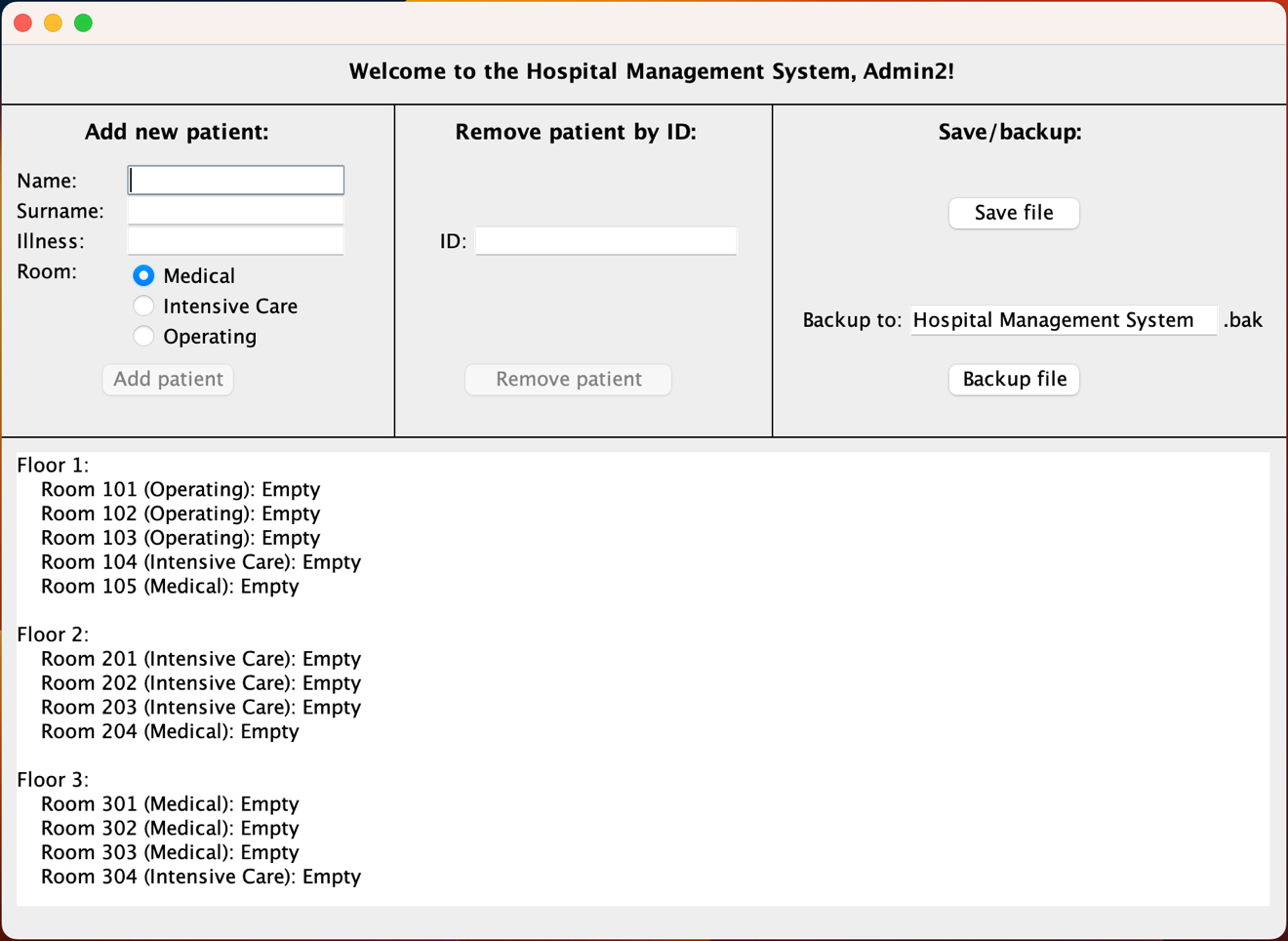


If login and password are correct, application redirects you to the System page.

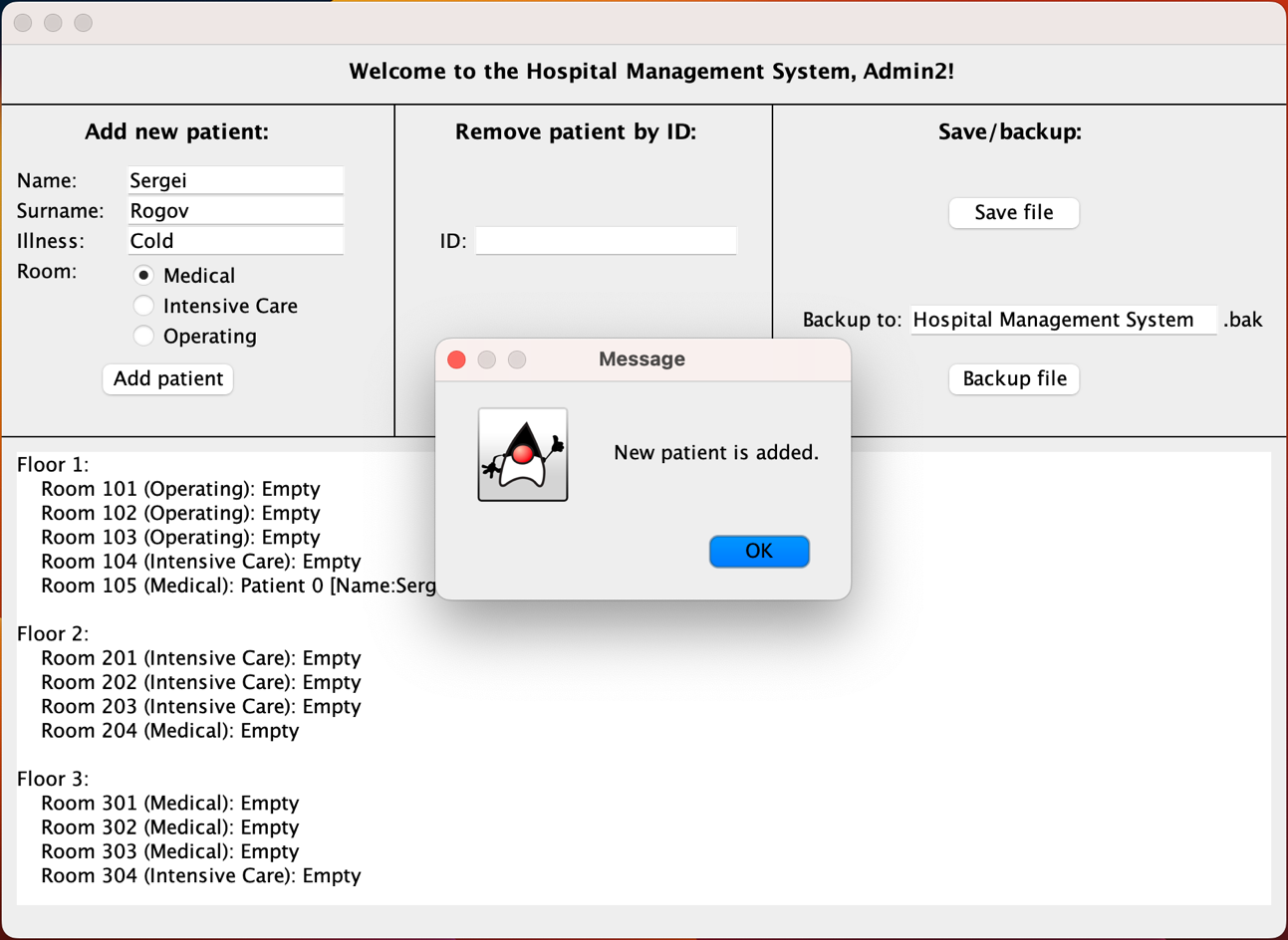
There is an upper part with available actions and bottom part with displayed information regarding the fullness of each room in a hospital.

Buttons corresponding to every action are only enabled if all necessary corresponding fields are filled with data by user.

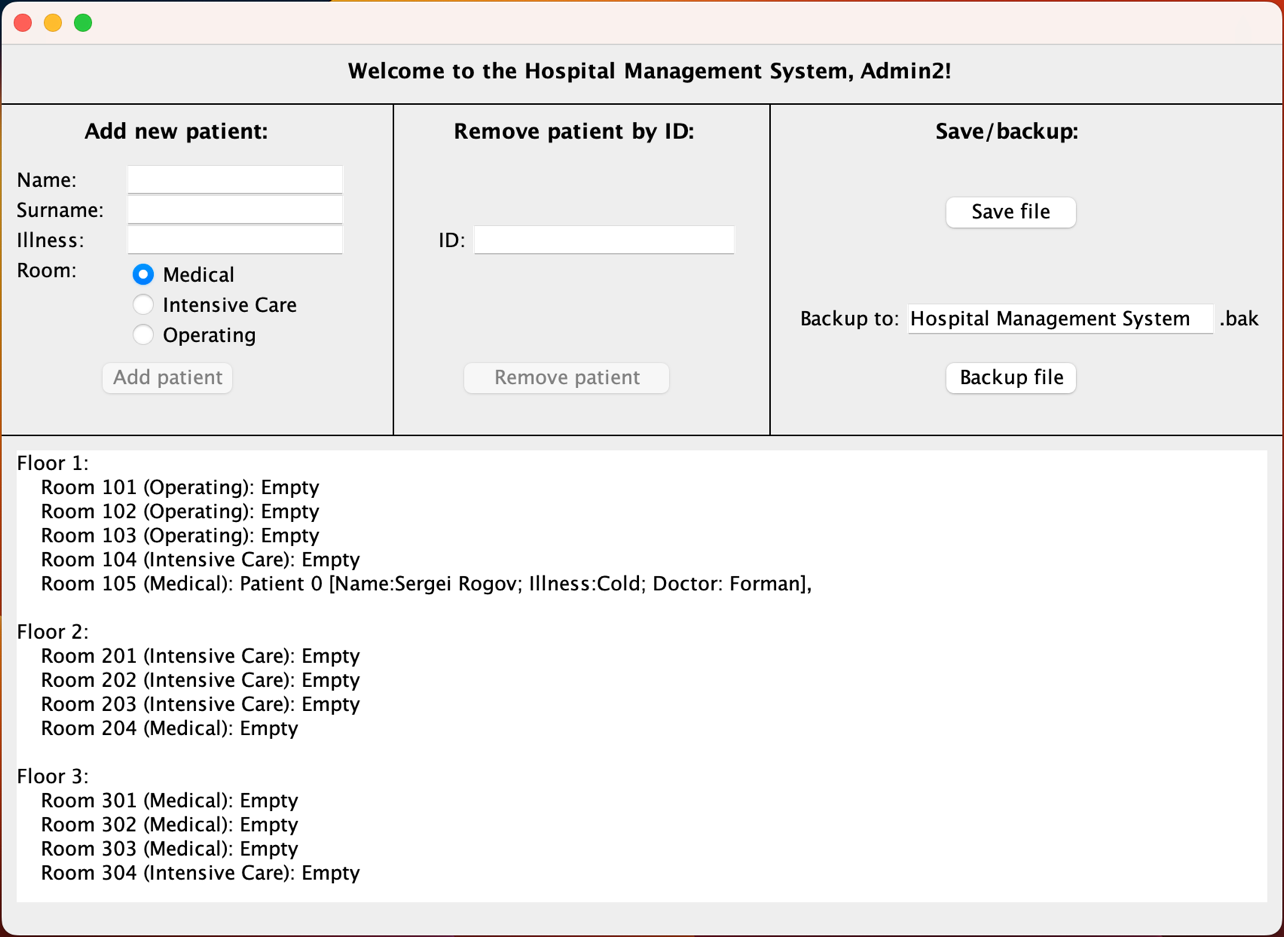
**System page – main page of the application:**



**Adding new patient:**

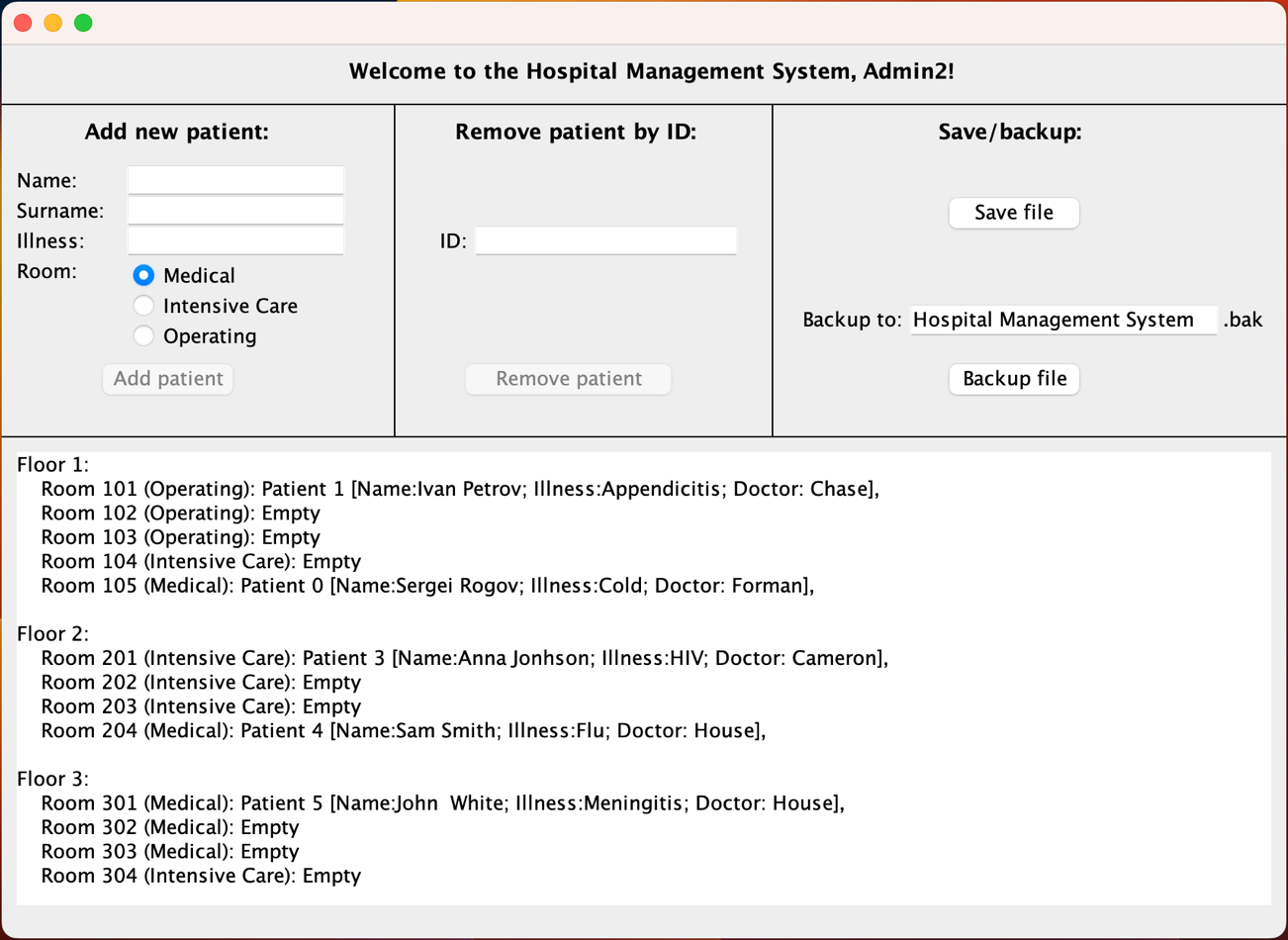


New patient is added to the system and assigned a room and a doctor:

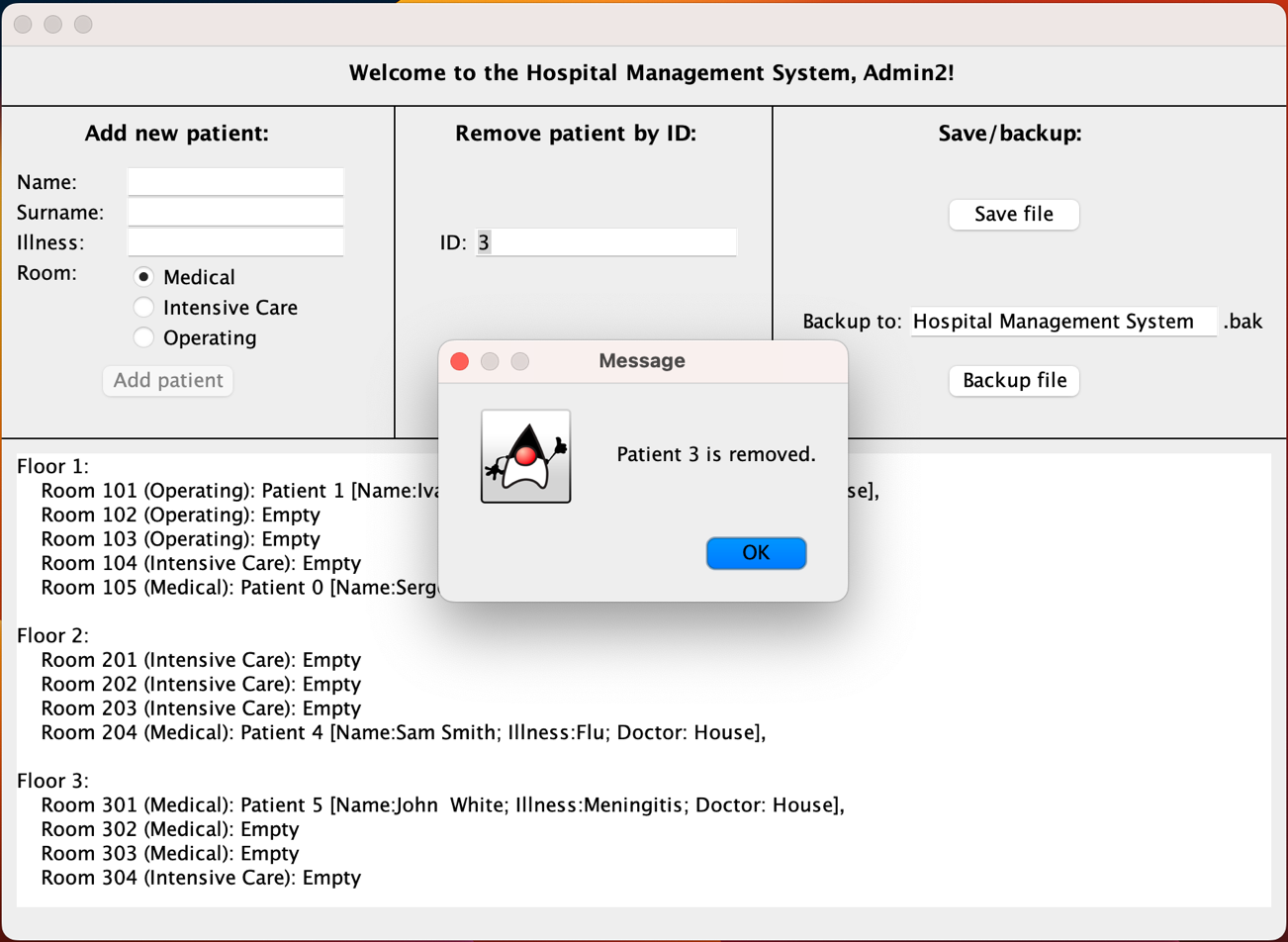


**Removing a patient by ID key:**

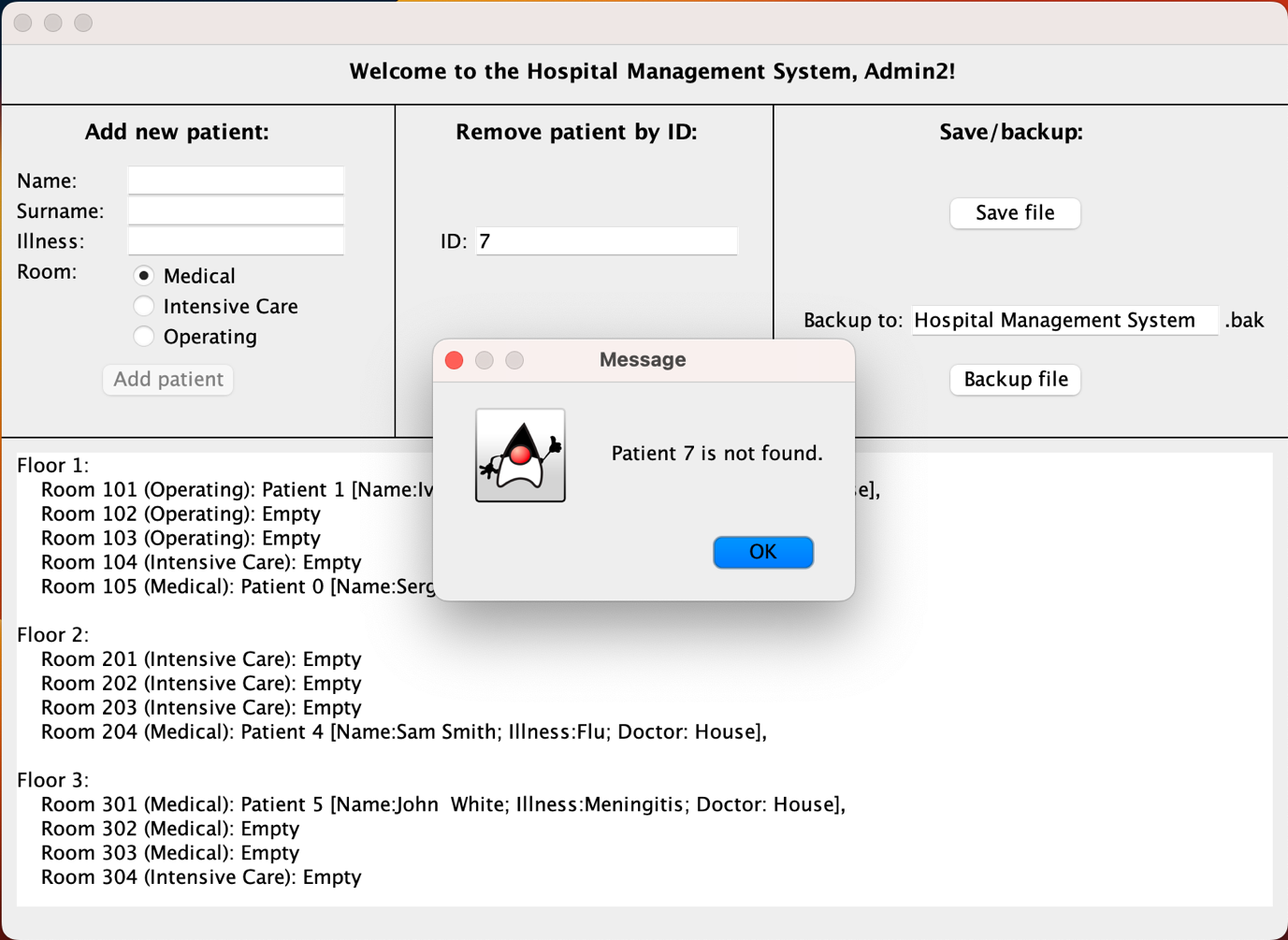
This feature is needed because when patients get well they should be discharged from hospital.



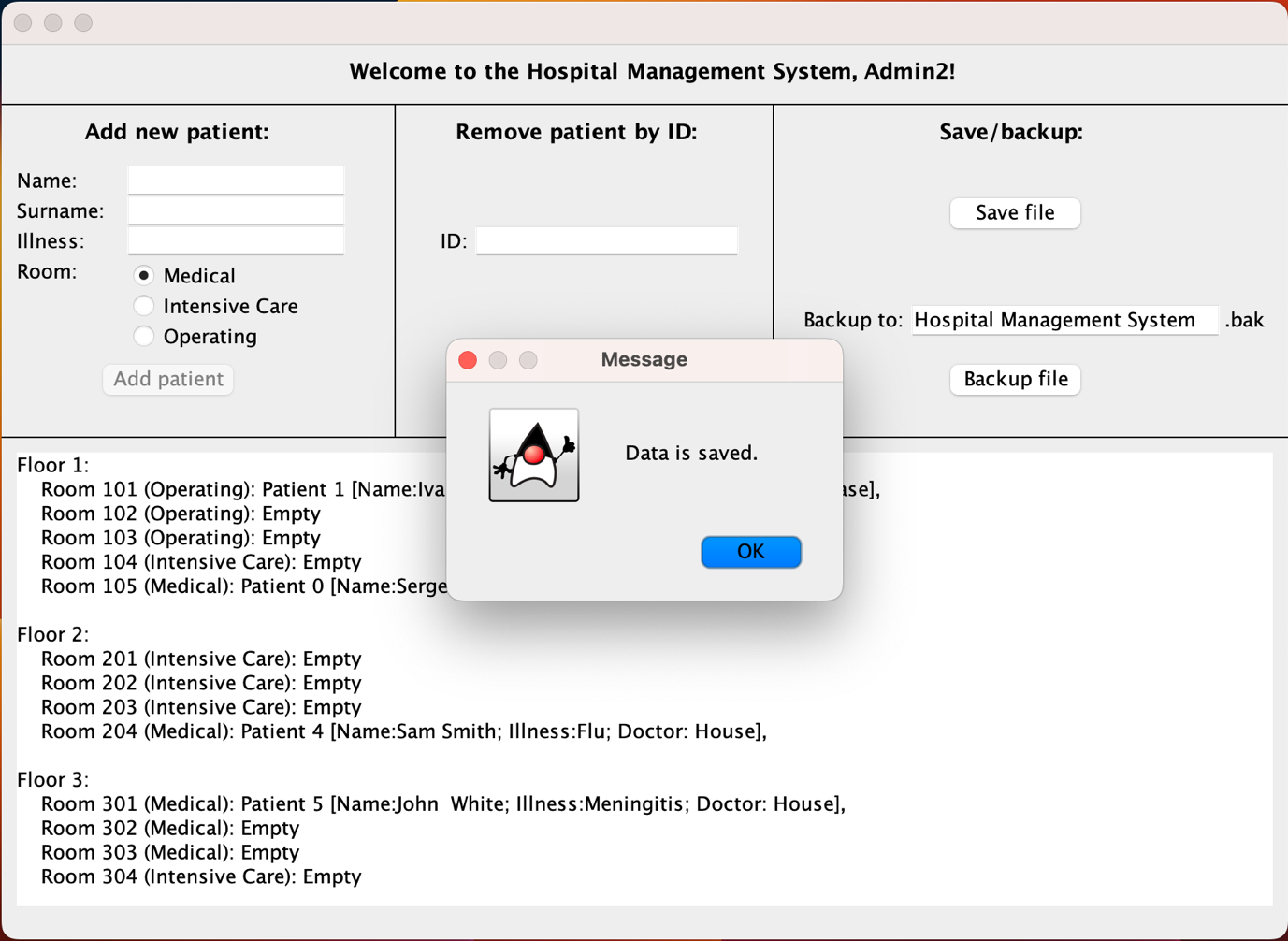
Patient 3, Anna Jonhson is discharged from hospital.



If there is no patient with given ID – the following message is shown and nothing happens:



**Saving data to a file:**

****

Hospital\_Management\_System-Database.txt

‘’’

6

0;Sergei;Rogov;Cold;105;Forman

1;Ivan;Petrov;Appendicitis;101;Chase

4;Sam;Smith;Flu;204;House

5;John ;White;Meningitis;301;House

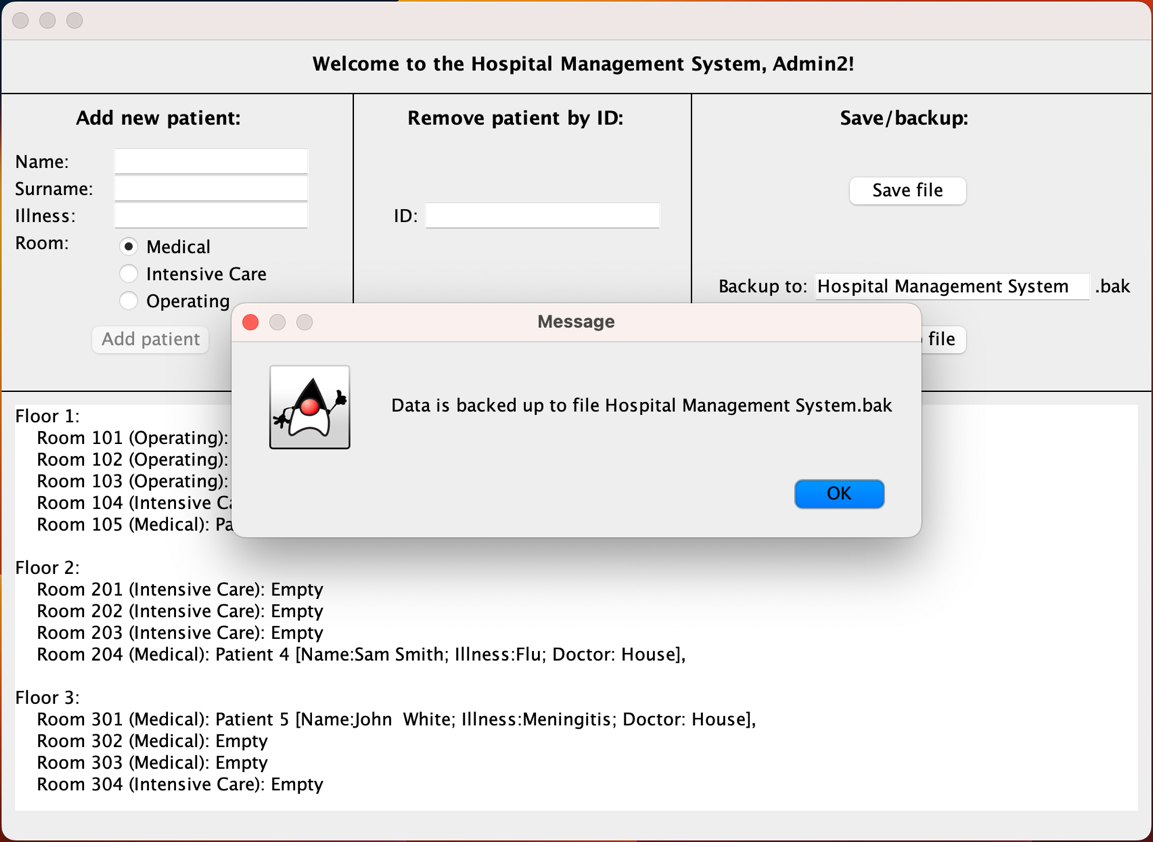
‘’’

First line serves as a total number of patients ever added to a system. Needed to assign a proper ID’s to patients.

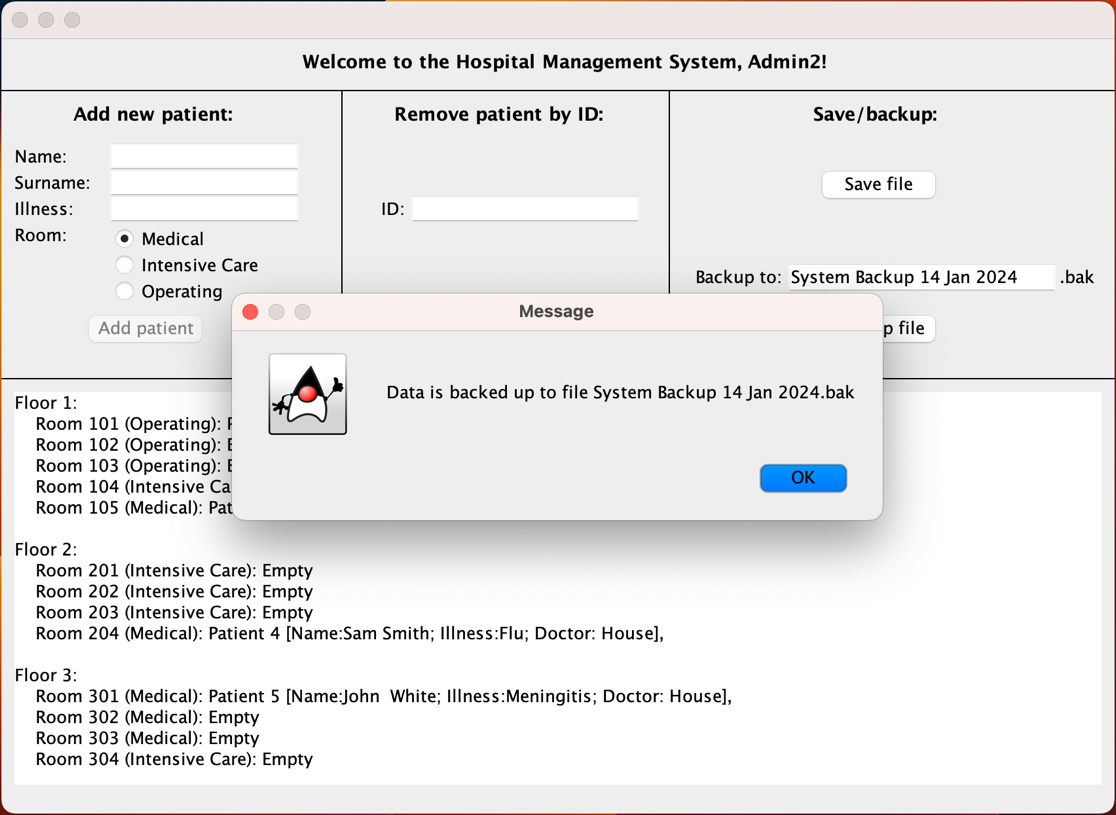
From the second line and further data is written line by line in CSV format with “;” separator.

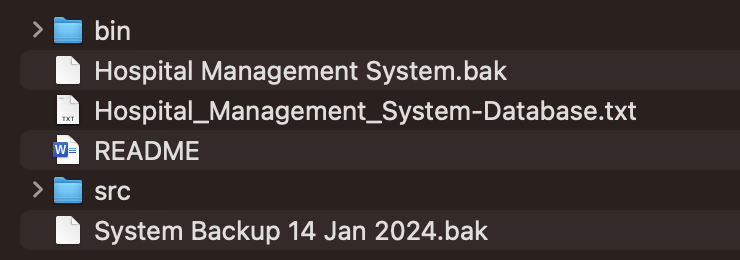
ID; name; surname; illness; roomID; doctor.

**Backup to a file called Hospital Management System.bak**



**Name-it option is supported:**





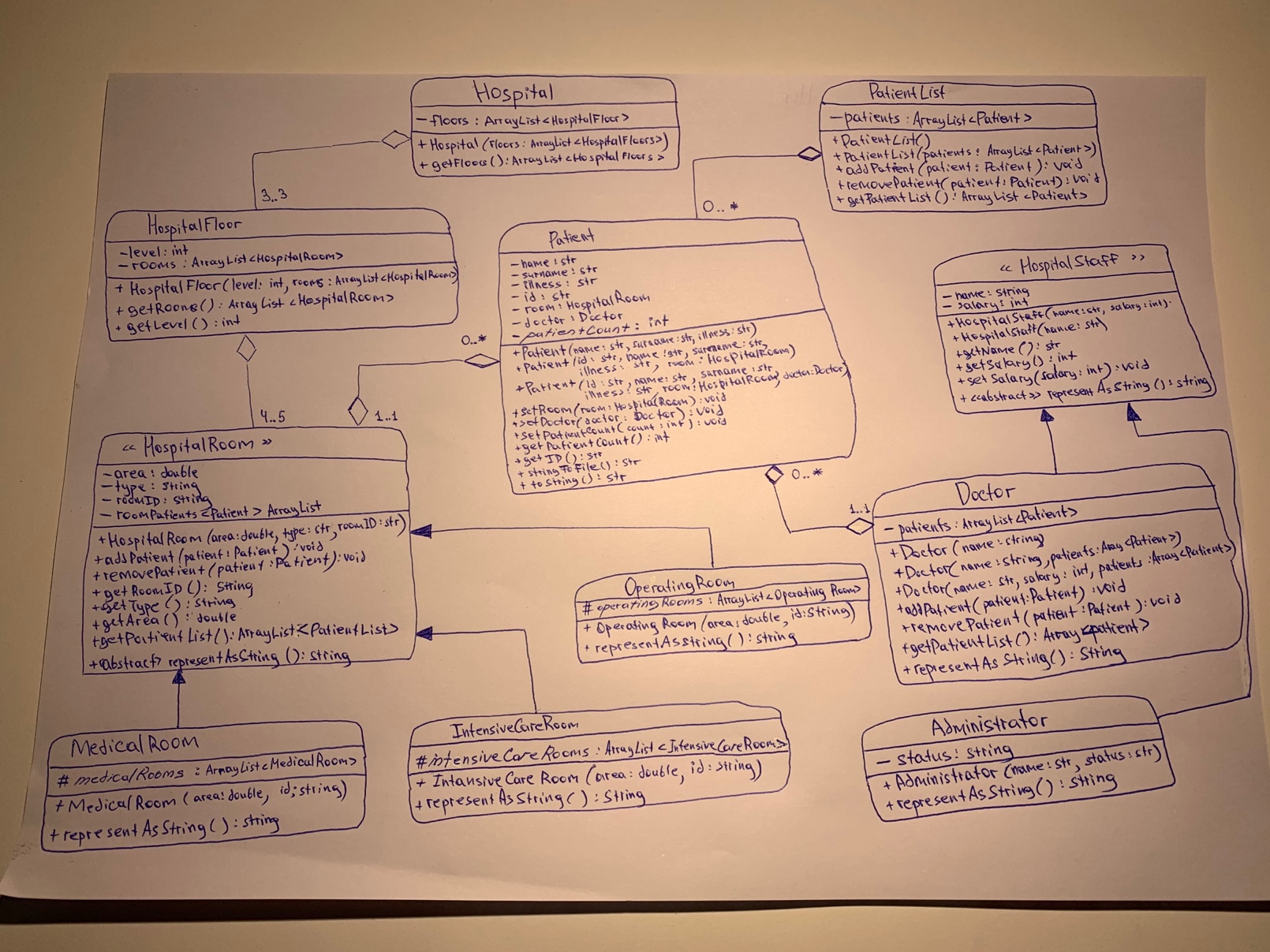
**Loading from a file:**

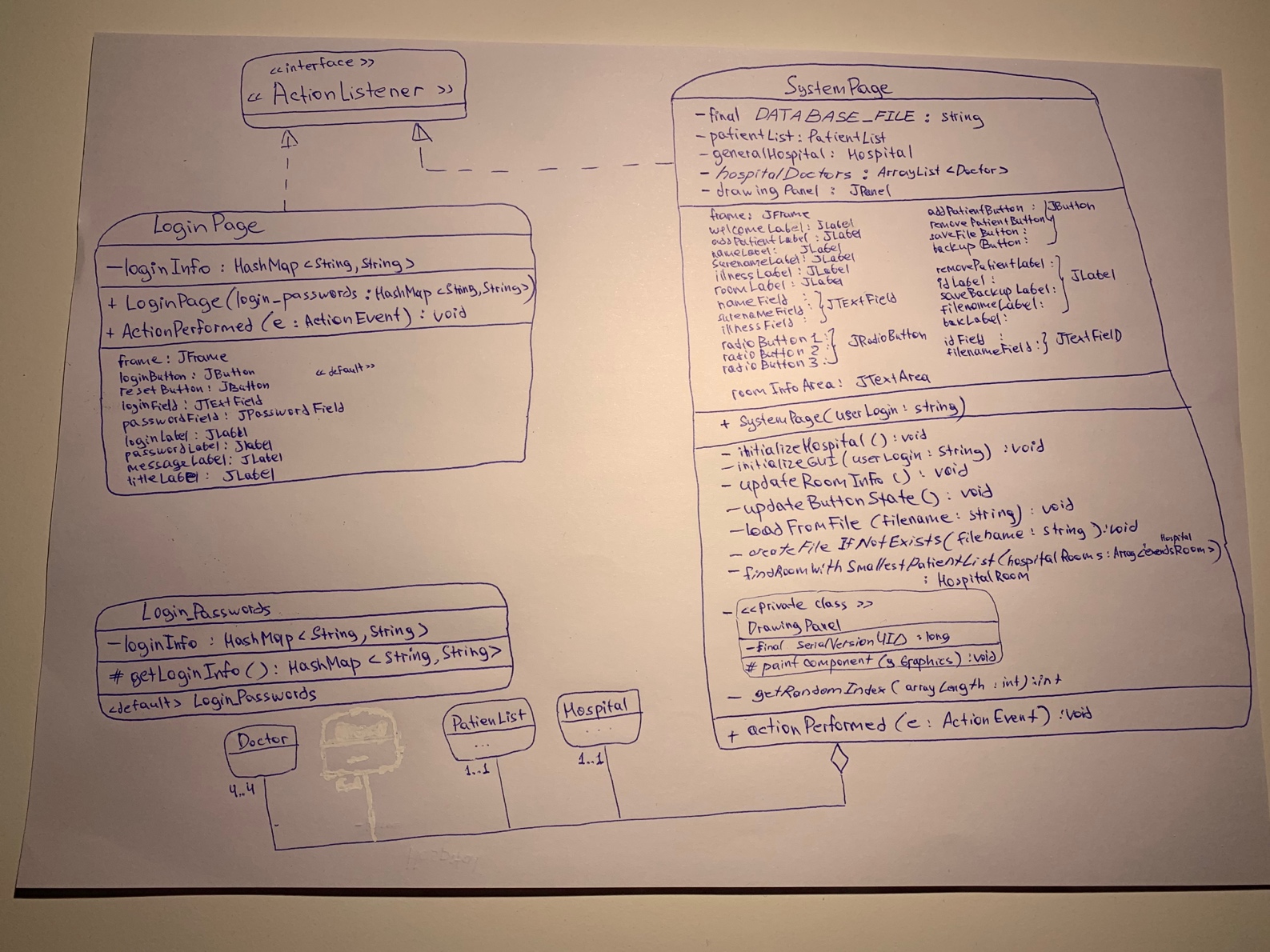
If the program is terminated, the patients record is not lost (assuming user pressed the SAVE button). When you run this program again, all the data is retrieved from Hospital\_Management\_System-Database.txt file and all the records are restored.

Starting from second run of the program, it first restores data from the database file, so all previously made records are immediately displayed in the information area.

Loading happens automatically.

**UML Class Diagrams**

****

****

**Application structure:**

1. Application starts with the Main class, where main function kick-starts the app.
2. Logins and passwords are initialized and Login Page is shown.
3. If user is successfully authenticated, Login Page is disposed and System Page is displayed.
4. After that, depending on user actions, System Page is carrying all logic regarding Hospital Management System actions.
5. If program is terminated and user runs it again, system restores all saved data from database file and instantiates all necessary objects, so nothing is lost after termination.

**State Diagram:**

