GEBZE TECHNICAL UNIVERSITY

CSE222 PROJECT REPORT 2

Health System Application

Authors:

Merve Torun

Navruz Yusuf Boy

Abdullah Çelik

Oğuzhan Sezgin

Esra Nur Arıcan

Dilara Karakas

Haktan Bilgehan Dilber

Supervisors:

Prof. Dr. Fatih Erdoğan Sevilgen

Mehmet Burak Koca

June 24, 2020

Contents

Chapter 1

Group Members

1.1 Group Members

- Merve Torun
- Navruz Yusuf Boy
- Abdullah Çelik
- Oğuzhan Sezgin
- Esra Nur Arıcan
- Dilara Karakaş
- Haktan Bilgehan Dilber

Chapter 2

Problem Definition

2.1 Problem Definition

From past to present, the field of health is a leader in being one of the areas with the most data input and output worldwide. The coexistence of relevant data in this rapidly developing area, which constantly improves itself with new information, is an important issue both for health services and as it can be a light for the development of new treatment methods.For this reason, we thought developing a software in our project, where all this wide range of data can be grouped and stored within itself, and that every segment of the health sector can easily access information that concerns it.

The manual handling of the recording informations on this area is time consuming and highly prone to error. Our purpose in this project is to automate data usages and the process of day-to-day activities such as asking appointment , checking prescriptions , displaying patient's medical records. With the health clinic management system we designed, we aim to enable citizens and healthcare professionals to access and organize data collected from healthcare institutions and users. Another purpose of this system is to collect all health records, such as prescriptions, test results, personal information, user history in one place and make them available for use. Thus, we hope to ensure that the diagnosis and treatment processes are managed quickly and in good quality, before and after.Also these services will be provided in an efficient, cost effective manner with goal of reducing the time and resources currently required for such tasks.Our aim at this project is easy use of data, consolidating data and ensuring data integrity. With this project, we aim to store ever-increasing and diverse data according to common areas of use and to provide it at a level that is easily accessible to users.

Chapter 3

Users of the System

3.1 Administrative Users

Administrator: Administrators can view all organizational data and user data that the users' consent to be seen. Administrators are the most privileged users of the system. They can add or remove doctors, patients, nurses, lab employees, clinics, pharmacies and pharmacists to the system.

3.2 End Users

3.2.1 Internal End Users

Doctor: Doctors can add and remove patients to the system. Doctors can view medical records of their patients. They can add new records and prescriptions to the system.
Nurse: Nurses can add and remove patient appointments to the system.
Lab employee: Lab employees can add and remove test results to the system.

3.2.2 External End Users

Patient: Patients can get doctor and nurse appointments. They can check their existing prescriptions. Patients can see their information on three different fields. The first field contains the patient's personal information such as name-surname, birth date, address and phone number. The second field contains the patient's medical record. And the last field contains patients test results.
Pharmacist: Pharmacists are allowed to see the prescription data of the patients. Prescriptions of patients from near hospitals are automatically shown to pharmacists.
They can also remove prescriptions once the patient gets their medications.

Chapter 4

Requirements

4.1 User Requirements

4.1.1 Administrator

Updating the data on the system and granting access differs for each user. The administrator logs the patient into the system. First of all, if the patient has never registered in the system before, administrator creates a new patient object and enters the necessary information. It adds to the list of patients in the doctor that the patient should be. If the patient ends treatment, he or she will be removed from the list of his or her doctor. But all information remains registered in the system. In this way, the patient can access his or her past information (analysis results etc.) at any time.

4.1.2 Patient

Patient data is kept in the system. These data are variable in five different areas: Personal information (name, surname, date of birth, address, phone number), medical information(allergies, diseases passed, old examination information, vaccination records), medical documents(previously obtained MR, analysis, x-ray, documents related to test results), appointment history(this data is kept here if there is a current appointment) and recipe information(recipe code).

4.1.3 Doctor

The doctor has access to all the patient's data. But it can only update its medical data and prescription information. After the treatment of the patient is finished, doctor request the nurse to be removed from the patient's own list.

4.1.4 Pharmacist

Pharmacy staff has access to prescription information of patients.It provides the drugs that the patient should take with the prescription information data and deletes the provided prescription from the system.

4.1.5 Nurse

The nurse will be able to update the patient appointments in the system. It adds the patient's appointment to the system and deletes the appointment of the patient whose treatment is finished.

4.1.6 Laboratory Employee

Laboratory employee adds patient's analysis results to the system. It deletes unnecessary test results from the system.

4.2 System Requirements

4.2.1 Functional Requirements

* This part of the requirements changed, since our design is more clear and detailed than the second project report, we detailed and extended the functional requirements.*

This system will bring simple solutions to the problems that patients experience during the treatment process in healthcare. It will facilitate the patient-doctor-hospital relationship and allow the patient to adapt easily to the system. It will also allow treatment to complete as soon as possible.

4.2.1.1 Class of Administrator

The administrator is the first step of accessing the system. The clinic, pharmacy and employees added by the administrator will have formed the remaining part of the system. Each person will have a unique id and password.
There are six positions in the system. These are administrator, nurse, pharmacist, doctor, patient and lab employee.
Administrator can add or delete contacts in these positions. In addition, it will speed up the treatment process by adding or deleting clinics and pharmacies close to the hospital and these data are kept in own class type list structure. Administrator can add or remove doctors to graph structure if they are friends.
Also admin can upload new announcements to the system. These announcements are important information messages given by hospital to doctors. Admin of the hospital, randomly assigns a doctor to a patient when a new patient signs up (like family doctor).

4.2.1.2 Class of Doctor

Doctors will be able to get patients to the system and easily direct their positions in the system. In addition, they will be able to write electronic prescriptions and accelerate the patient-pharmacy relationship.
The doctor has access to their own appointments. These data are stored in the Arraylist data structure, which is type of the Appointment class.
When a doctor enters the system, he/she can see the announcements added by admin of the hospital. These announcements are kept in stack structure because when doctor see that announcement, it should be removed. Doctors can add and remove friend doctors recorded in that hospital.

4.2.1.3 Class of Nurse

The nurse changes the medical data object we keep in the patient class. This object is the medical data class. You can add operation data in this class.
We keep the operation data in the arrayList data structure.

4.2.1.4 Class of Patient

Own medical data and appointment data are kept in this class. The patient can see them at any time. Uses arraylist as data structure for appointment. When new appointment added to the patient, appointments automatically sorted for clearity.

4.2.1.5 Class of Lab Employee
The lab employee adds tests to the patient's medical data. Uses array list as data structure.

4.2.1.6 Class of Pharmacist
Pharmacist class keeps an pharmacy information inside. This field represents where the pharmacist works. When a patient comes, pharmacists uses that patient and print his/her latest prescription recorded in the system.

4.2.1.7 Class of User
This abstract class implements HealtySystemUsers interface. Personal data, login name, password and hospital are kept as data. User class a method to verify users while entering the system. The classes mentioned above are also derived from this class.

4.2.1.8 Class of Personal Data
In this class, name, surname and ID are kept. Each user on the system gets an ID when their accounts are created. User ID's are generally used in selections of the system. Changes can be made at any time.

4.2.1.9 Class of Medical Data
Keeped Data: Tests(arrayList), bloodType(string), notes(arrayList), operations(arraylist), prescriptions(arrayList).

4.2.1.10 Class of Appointment
Keeped Data: Doctor, patient and date(GregorianCalendar type). We used java's GregorianCalender class to keep localized appointment times for the patients. To correctly print the times we used java's DateTimeFormatter.

4.2.1.11 Class of Hospital
Keeped Data: Name, admin, patients(red black tree(tree map)), doctors (Skip list(ConcurrentSkipListSet)), nurses (arrayList), pharmacist (arrayList), lab employees (arrayList), family doctor(red black tree(tree map)), friend doctor(graph).

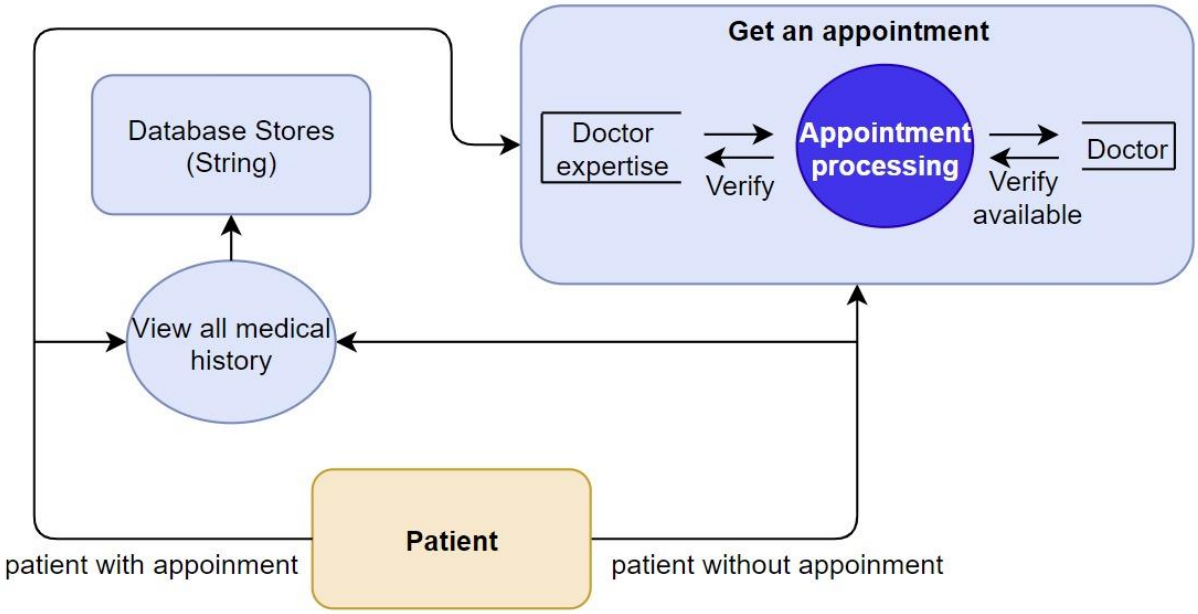
4.2.2 Non-functional Requirements

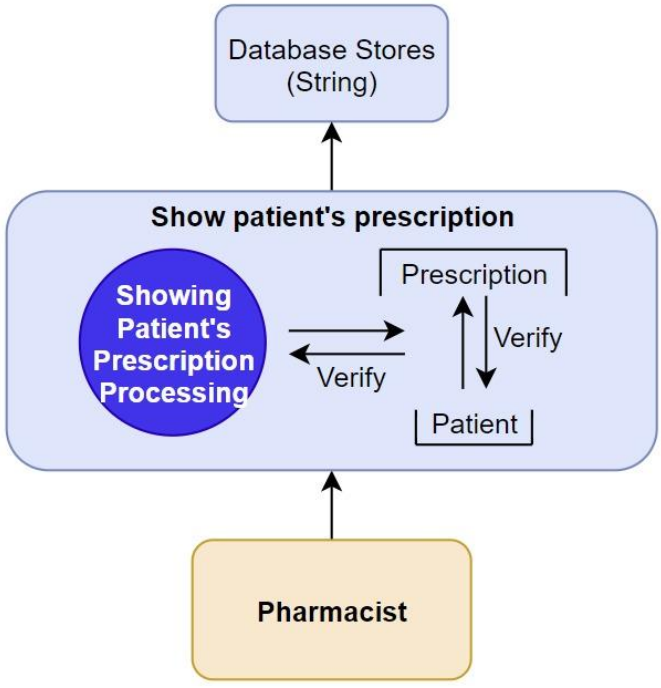
The software should run on any operating system with JVM installed.
This system should support English characters.
Software’s language:Java.
While writing the system, JDK 13.02 used.
Using programming technique: Object Oriented Programming.
Any modification (insert, delete, update) for the Database shall be synchronized.
System stores data session wisely.
Any user who uses the system must have an enter ID and password.
Administrators shall be able to view and modify all information in system.
Patients whose registered in the system,should confirm that their doctors and administrators can see their information due to ethical requirements.
Every method on the program such as adding and removing patient,doctor or appointment, controls the operation . If an error occurs, program throws exceptions due to reliability requirements.

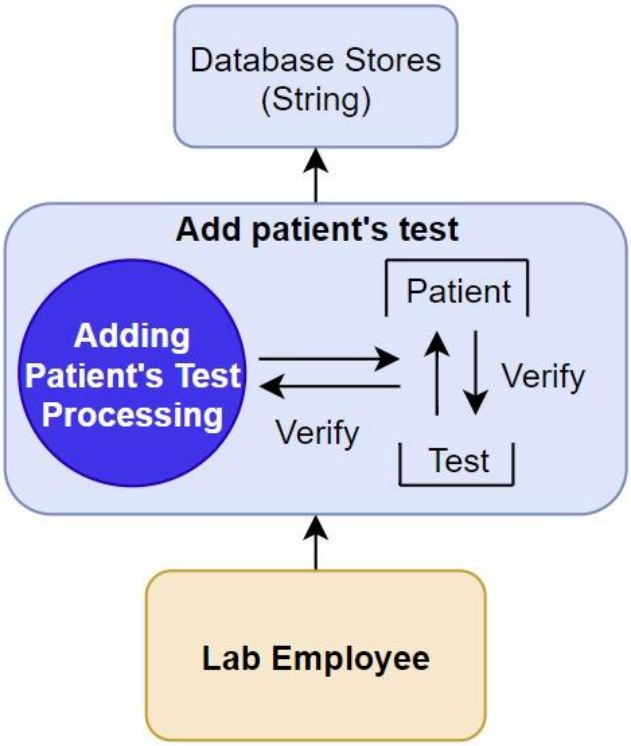
The system shall keep a log file of all the errors.

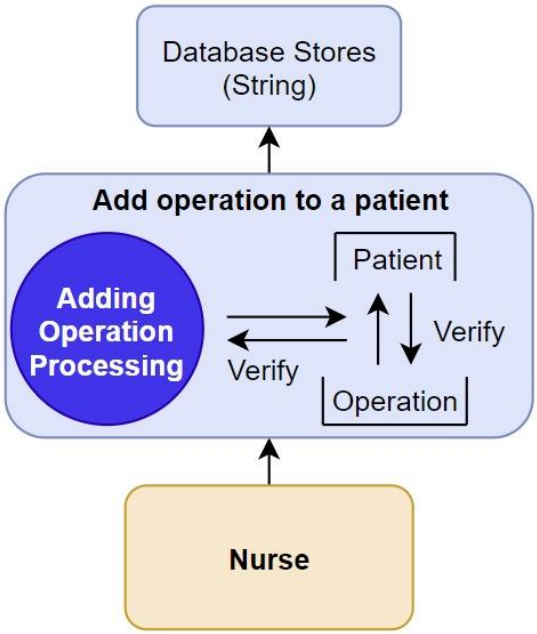
Chapter 5

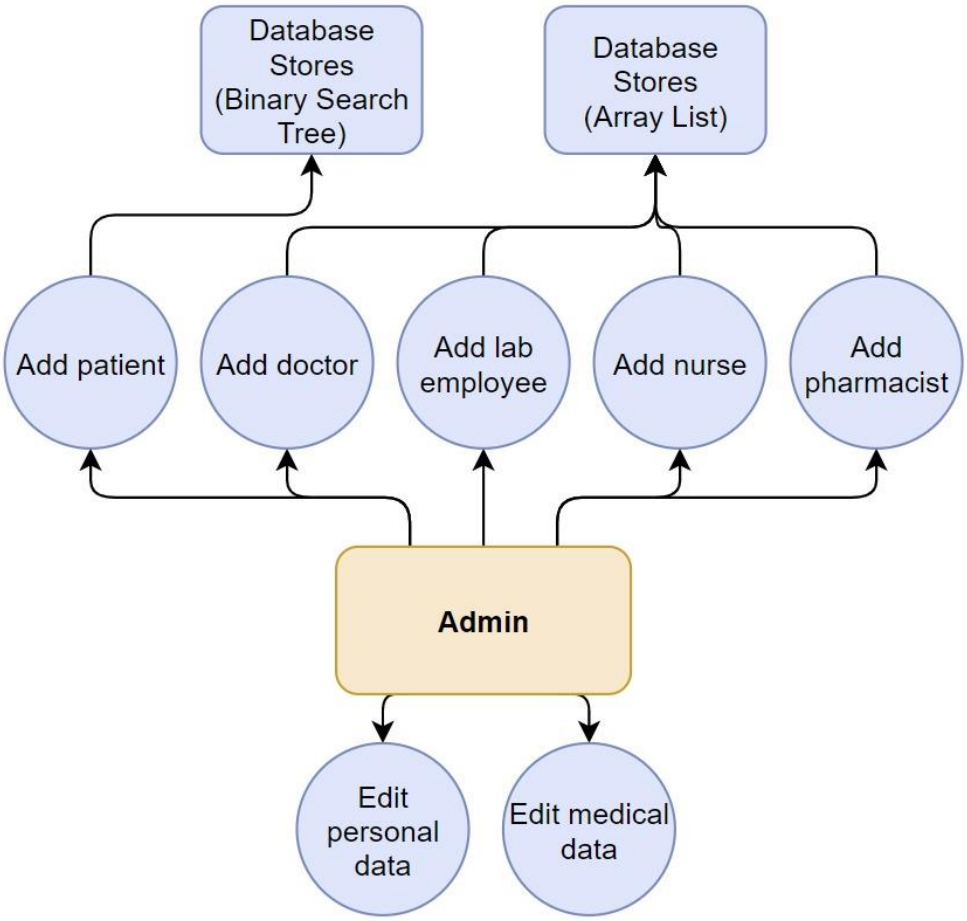
System Modules

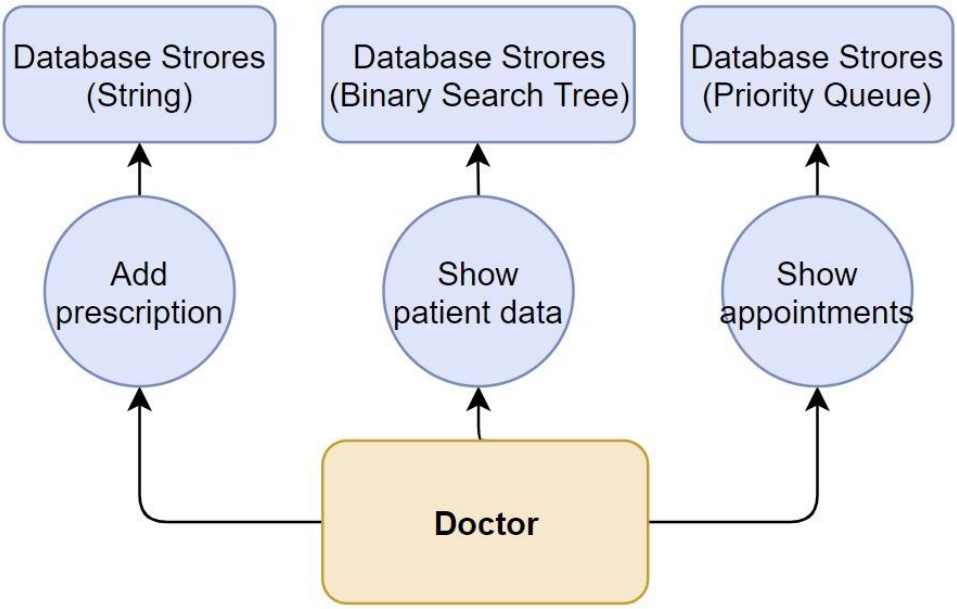





Database Stores
(String)
Add patient's test
Adding
Patient's Test
Processing
Patient
Verify
Verify
Test
Lab Employee

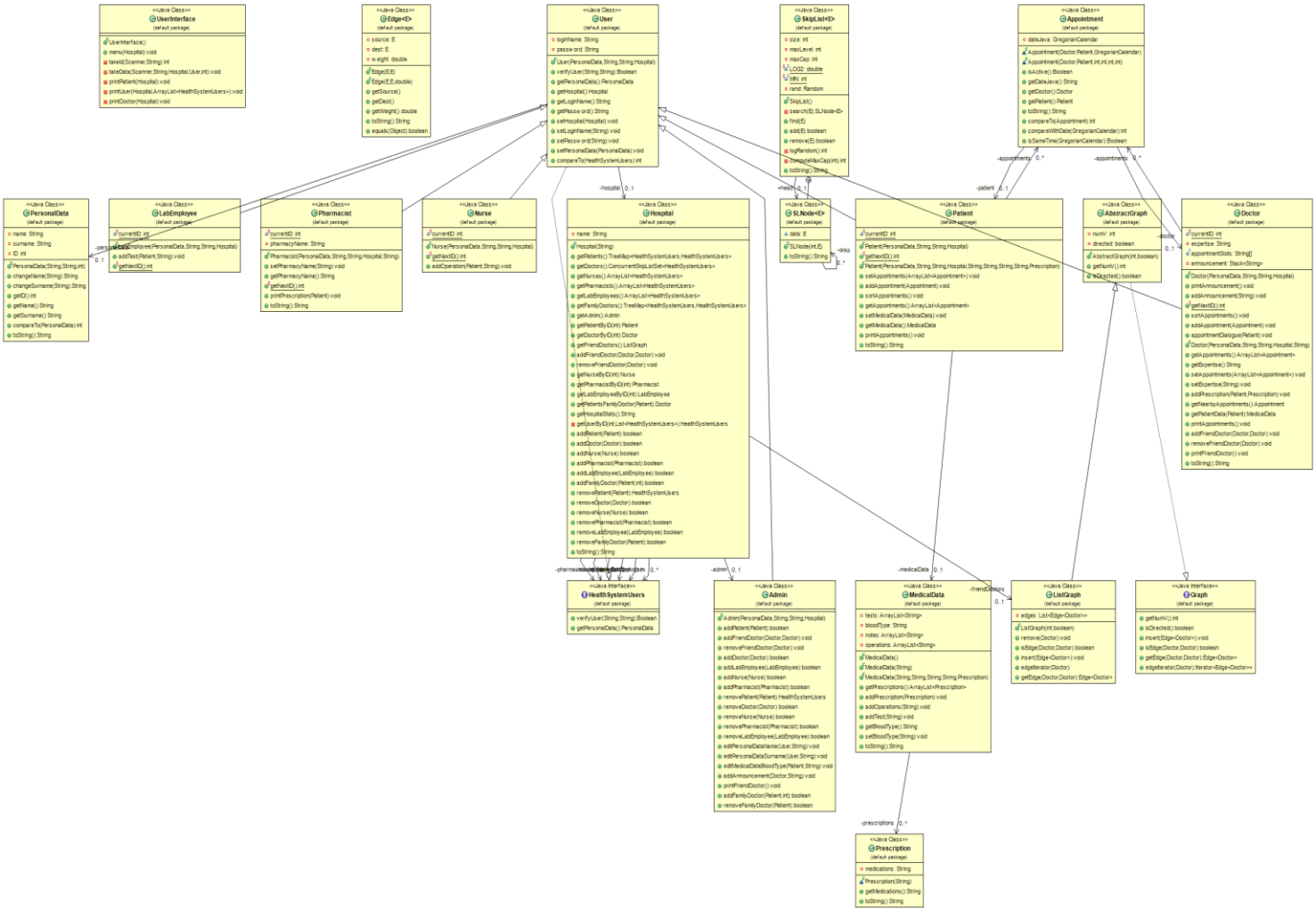

Database Stores
(String)
Add operation to a patient
Adding
Operation
Processing
Patient
Verify
Verify
Operation
Nurse


Database Stores (Binary Search Tree)
Database Stores (Array List)
Add patient
Add doctor
Add lab employee
Add nurse
Add pharmacist
Admin
Edit personal data
Edit medical data


Database Strores (String)
Database Strores (Binary Search Tree)
Database Strores (Priority Queue)
Add prescription
Show patient data
Show appointments
Doctor

Chapter 6

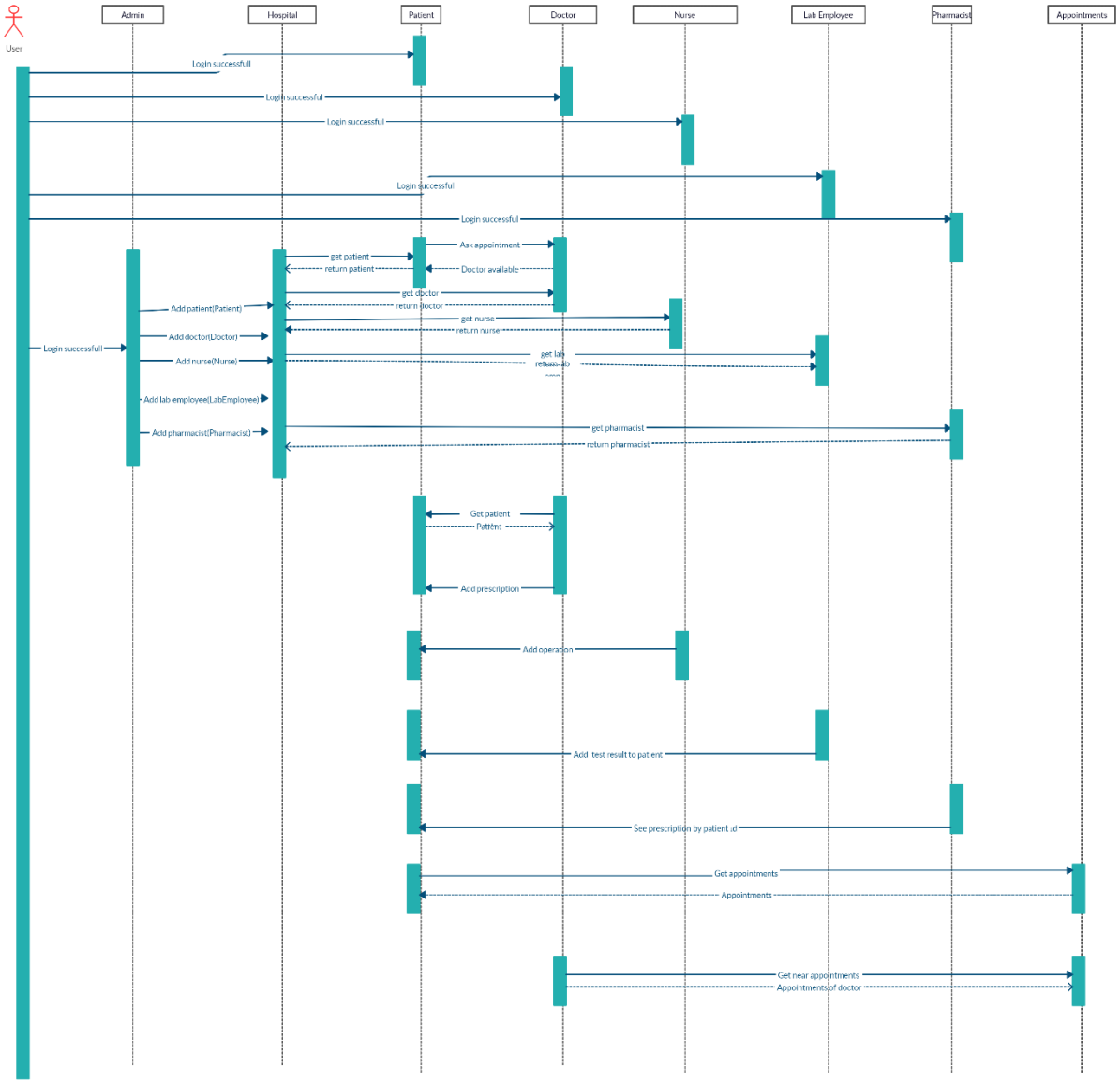
Class Diagrams



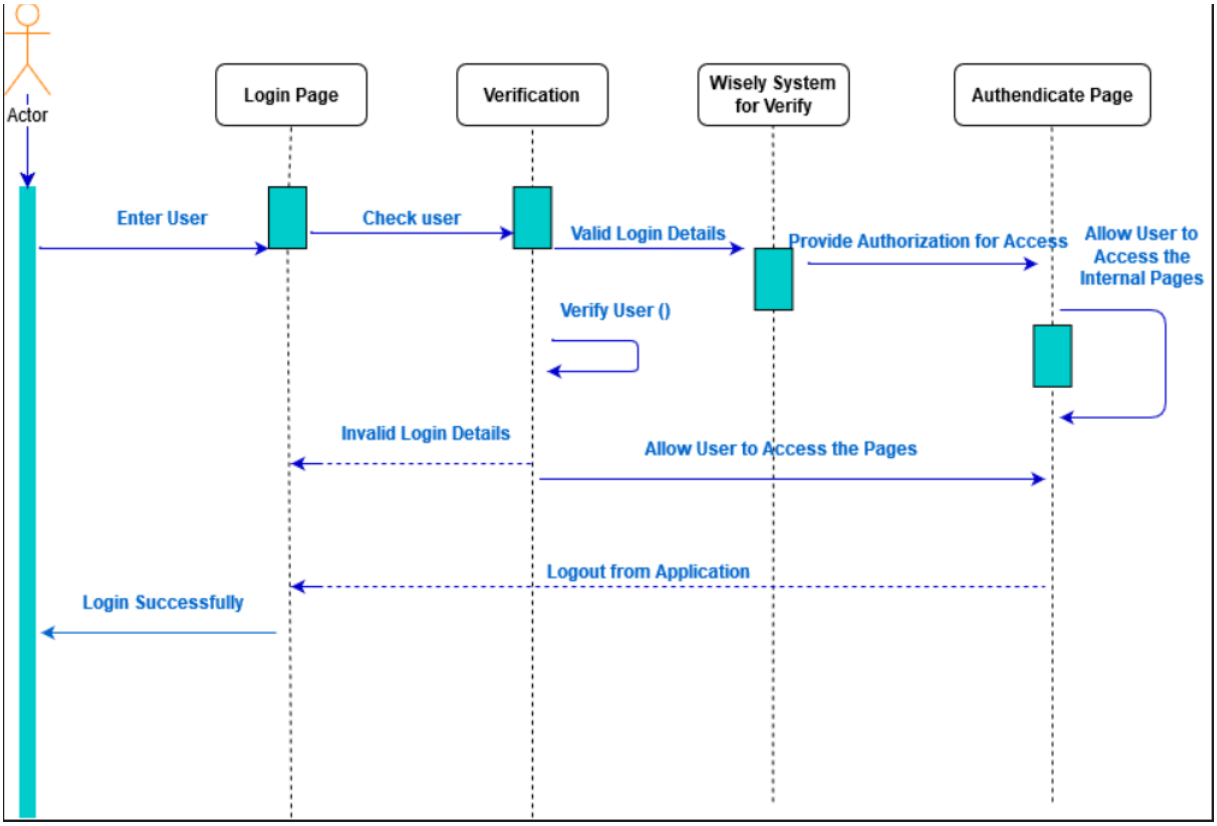
Chapter 7

Sequence Diagrams

This is the UML Sequence Diagram of Health System which shows the interaction between objects of Admin, Hospital, Patient, Lab employee, Doctor, Nurse, Pharmacist and Appointment. Sequence diagram updated.



Login Sequence Diagram :



Chapter 8

Implementation Details

Depending on the hospital management system software features, it can deal with a lot of tasks. It helps to automate routine tasks, design the patient-oriented workflows . One of the main requirements of the clinic management system is security. All medical records have to be protected and only accessible for the allowed users. The convenient and informative interfaces should correspond to their roles and responsibilities in order to protect the confidential data. For this reason, the system is designed to prevent access of stored information outside of the users.

This health system is specified in a manner that dictates accommodation of certain implementation details. The implementation environment of the developed system facilitates multiple users to use this system simultaneously. The user interfaces are designed keeping in mind that the users of this system are familiar to using console-based systems. Thus, we restricted ourselves to developing a console-based system so that it becomes easier for the end user to get acquainted to the developed system.

Our program written in Java . And we used object orianted programming concept. Because of our system is an console application, we did not use any framework,GUI etc. And we did not use any database. All of the informations about users kept in efficient data structures.

To arrange appointments we used Java's format library, also we used GregorianCalendar library which is subclass of Calender which provides the standart calendar system used by most of the world. For data structures we used to store user's informations and themselves in hospital , Java libraries used such as ArrayList, HashMap, Stack, Iterator, SkipList . Also we implement Edge.java, Graph.java, ListGraph.java ,AbstractGraph.java classes by ourselves to use in our system by modifiying them to our needs.

The user interface designed to easy to use and understand for users. Workers of hospital and pharmacists will be added by admin and at the first step of program, users will need to enter the system by their username and password. After the login step, each user will have their own menu that shows actions they are allowed to do. Also, we chose to store every test, operations and prescriptions that a patient had in their record in a data structure and we chose specific data structures for specific data fields in order to get the most efficient running results. For example, doctors nurses etc. needed faster time to get them by their index, so we used arraylists there. Every data structure that we used is chosen to serve customer needs in a most efficient way.

Also, we have changed the storage of patient's in our system. In second report were keeping them in "Binary Search Tree" because patients needed faster time to get them by their name and surname (their data fields), but now we used balanced binary search trees to spend less time when trying to find them. but now we are keeping the patients in "Red-Black Tree". Because of red-black tree is a self balancing tree, when binary search tree is able to form long chains of nodes that can cause searches to take linear time, a red-black tree guarantees a search operation takes logarithmic time (O(logn)). We changed it to Red-Black

tree because Red black tree is useful when we need insertion and deletion relatively frequent which we do a lot of insertion of patient's. Also Red-Black trees have relatively low constans in a wide variety of scenarios.

When we were sorting the appointments by date, we used merge sort algorithm. Because hospital system is all about speed and merge sort solved this need. We have choosen merge sort because:

1. MergeSort is stable by design, equal elements keep their original order.
2. MergeSort is well suited to be implemented parallel (multithreading).
3. MergeSort uses (about 30%) less comparisons than QuickSort. This is an often overlooked advantage, because a comparison can be quite expensive (e.g. when comparing several fields of database rows).

Chapter 9

Test Cases

9.1 User Operation Test

Test Case Name	Test Step	Action	Test Data	Expected Result	Test Result	Note
User Login Hospital System	1	Enter username in the username field.	Admin	System should accepts username	Passed the Test	User accepted
User Login Hospital System	2	Enter password in the password field.	XXXXX	System should accepts password	Passed the Test	Password accepted
User Enter Invalid Login Name-Password	1	Enter invalid username in the username field.	InvalidName	System should warn user – ask again	Passed the test	Warning is given
User Enter Invalid Login Name-Password	2	Enter wrong password in the password field.	XXXX	System should warn user – ask again	Passed the test	Warning is given

User Change Personal Data	1	User login system	LoginName Password	System should accepts user	Passed the test	User change saved
User Change Personal Data	2.1.a	Choose operation	Change Name	Change name menu should be opened	Passed the Test	User new Operation saved
User Change Personal Data	2.1.b	Enter new name	NewName	Name should be changed	Passed the test	User new Name saved

User Change Personal Data	2.2.a	Choose operation	Change Surname	Change surname menu should be opened	Passed the Test	User surname change entered
User Change Personal Data	2.2.b	Enter new name	NewSurname	Surname should be changed	Passed the Test	User surname Change saved
User Change Personal Data	2.3	Choose operation	Personal Data	Personal data should be printed	Passed the Test	Data printed

9.2 Admin Operation Test

Test Case Name	**Test Step**	**Action**	**Test Data**	**Expected Result**	**Test Result**	**Note**
Admin Add User to Hospital System	1	Admin login system	LoginName Password	System should accepts admin	Passed the test	Admin login name and Surname accepted
Admin Add User to Hospital System	2.1.a	Choose user which will add	Patient	Add patient menu should be opened	Passed the test	Admin adds patient

Admin Add User to Hospital System	2.1.b	Enter patient information	LoginName Password Hospital	Patient should be added	Passed the test	Admin adds Patient accepted
Admin Add User to Hospital System	2.2.a	Choose user which will add	Doctor	Add doctor menu should be opened	Passed the test	Admin adds Doctor
Admin Add User to Hospital System	2.2.b	Enter doctor information	LoginName Password Hospital	Doctor should be added	Passed the test	Admin adds Doctor accepted
Admin Add User to Hospital System	2.3.a	Choose user which will add	Lab Employee	Add lab employee menu should be opened	Passed the test	Admin adds Lab employee

Admin Add User to Hospital System	2.3.b	Enter lab employee information	LoginName Password Hospital	Lab employee should be added	Passsed the test	Admin adds Lab employee accepted
Admin Add User to Hospital System	2.4.a	Choose user which will add	Nurse	Add patient menu should be opened	Passed the test	Admin adds nurse
Admin Add User to Hospital System	2.4.b	Enter nurse information	LoginName Password Hospital	Nurse should be added	Passed the Test	Admin adds Nurse accepted
Admin Add User to Hospital System	2.5.a	Choose user which will add	Pharmacist	Add pharmacist menu should be opened	Passed the Test	Admin adds pharmacist
Admin Add User to Hospital System	2.5.b	Enter pharmacist information	LoginName Password Hospital	Pharmacist should be added	Passed the test	Admin adds Pharmacist accepted

9.3 Lab Employee Operation Test

Test Case Name	Test Step	Action	Test Data	Expected Result	Test Result	Note
Lab Employee Add Test	1	Lab employee login system	LoginName Password	System should accepts lab employee	Passed the Test	Lab employee Login accepted
Lab Employee Add Test	2.	Enter patient information	PName PSurname	Patient profile should be find	Passed the test	Patient İnformation accepted
Lab Employee Add Test	3.	Enter test result	Result	Test should be added	Passed the Test	Added test successfully

Lab Employee Enter Invalid Patient	1	Lab employee login system	LoginName Password	System should accepts lab employee	Passed the Test	Lab Employee Login accepted
Lab Employee Enter Invalid Patient	2.	Enter wrong patient information	PName PSurname	System should warn user – ask again	Passed the Test	Warning is given

9.4 Nurse Operation Test

Test Case Name	Test Step	Action	Test Data	Expected Result	Test Result	Note
Nurse Add Operation	1	Nurse login system	LoginName Password	System should accepts nurse	Passed the test	Nurse Login accepted
Nurse Add Operation	2.	Enter patient information	PName PSurname	Patient profile should be find	Passed the Test	Patient İnformation accepted
Nurse Add Operation	3.	Enter test result	Operation	Test should be added	Passed the test	Patient tests added

Nurse Enter Invalid Patient	1	Nurse login system	LoginName Password	System should accepts lab employee	Passed the Test	Warning is given
Nurse Enter Invalid Patient	2.	Enter wrong patient information	PName PSurname	System should warn user – ask again	Passed the test	Warning is given

9.5 Doctor Operation Test

Test Case Name	Test Step	Action	Test Data	Expected Result	Test Result	Note
Doctor Add Prescription	1	Doctor login system	LoginName Password	System should accepts doctor	Passed the test	Doctor Login accepted
Doctor Add Prescription	2.	Choose operation	Add Prescription	Add prescription menu should be opened	Passed the Test	Add Prescription selected
Doctor Add Prescription	3.	Enter patient information	PName PSurname	Patient profile should be find	Passed the test	Patient login information

Doctor Add Prescription	4.	Enter prescription information	Medicines	Prescription should be added	Passed the test	Added succesfully

Doctor Add Prescription Test to Invalid Patient	1	Doctor login system	LoginName Password	System should accepts doctor	Passed the test	Doctor Login accepted
Doctor Add Prescription Test to Invalid Patient	2.	Choose operation	Add Prescription	Add prescription menu should be opened	Passed the Test	Add Prescription selected
Doctor Add Prescription Test to Invalid Patient	3.	Enter wrong patient information	PName PSurname	System should warn user – ask again	Passed the test	Warning given

Doctor Check Nearby Appointment	1	Doctor login system	LoginName Password	System should accepts doctor	Passed the test	Doctor Login accepted

Doctor Check Nearby Appointment	2.	Choose operation	Nearby Appointment	Doctor appointment should be printed	Passed the test	Selected operation

Doctor Check Patient Data	1	Doctor login system	LoginName Password	System should accepts doctor	Passed the test	Doctor Login accepted
Doctor Check Patient Data	2.	Choose operation	Patient data	Patient data menu should be opened	Passed the test	Selected operation

Doctor Check Patient Data	3.	Enter patient information	PName PSurname	Patient profile should be find and print	Passed the test	Patient İnformaition accepted

9.6 Patient Operation Test

Test Case Name	Test Step	Action	Test Data	Expected Result	Test Result	Note
Patient Check Medical Data	1	Patient login system	LoginName Password	System should accepts patient	Passed the test	Patient Login accepted
Patient Check Patient Data	2.	Choose operation	Medical Data	Medical data menu should be opened	Passed the test	Selected operation
Patient Check Patient Data	3.1	Choose data	Tests	Patient tests and results should be printed	Passed the test	Patient test printed
Patient Check	3.2	Choose data	Blood Type	Patient blood type	Passed the	Patient Blood type

Patient Data				should be printed	test	Prined
Patient Check Patient Data	3.3	Choose data	Operations	Patient operations should be printed	Passed the test	Patient Operations printed

Patient Check Appointment	1	Patient login system	LoginName Password	System should accepts patient	Passed the test	Patient login accepted
Patient Check Appointment	2.	Choose operation	Appointment	Patient appointment should be printed	Passed the test	Selected operation

9.7 Pharmacist Operation Test

Test Case Name	Test Step	Action	Test Data	Expected Result	Test Result	Note
Pharmacist Check Prescription	1	Pharmacist login system	LoginName Password	System should accepts pharmacist	Passed the test	Pharmacist Login Accepted
Pharmacist Check Prescription	2.	Enter prescription id	123456	Prescription should be printed	Passed the test	İd accepted

Pharmacist Check Prescription With Invalid ID	1	Pharmacist login system	LoginName Password	System should accepts pharmacist	Passed the test	Warning is given
Pharmacist Check Prescription With Invalid ID	2.	Enter invalid prescription id	111111	System should warn user – ask again	Passed the test	Warning is given

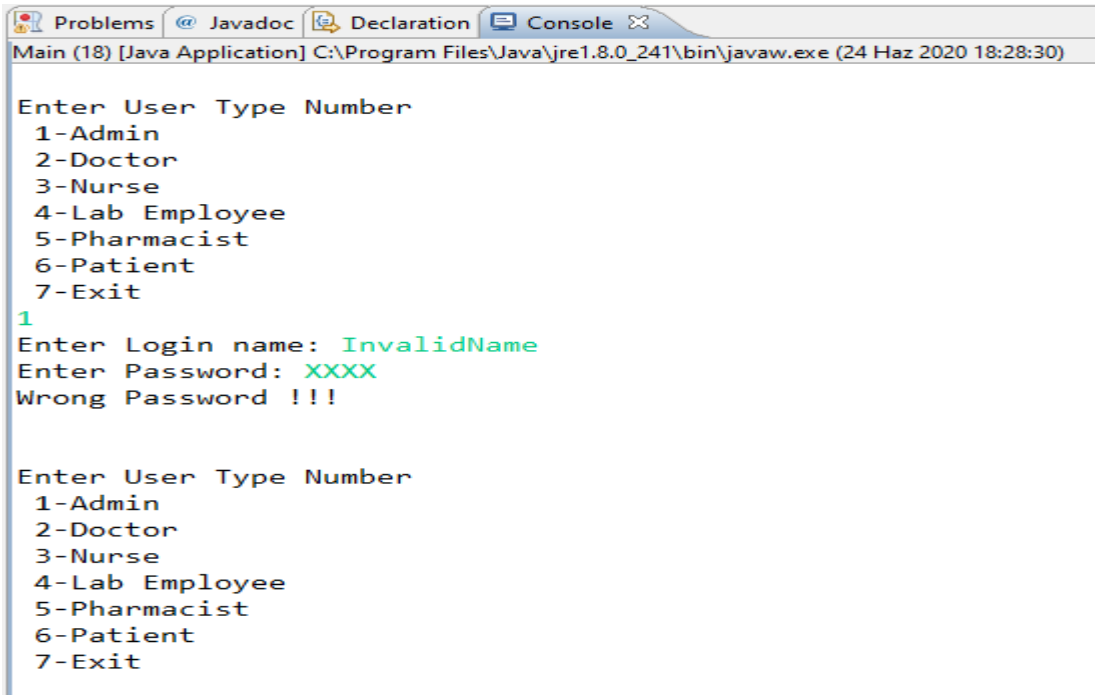
Chapter 10

Results

10.1 Test Case Results

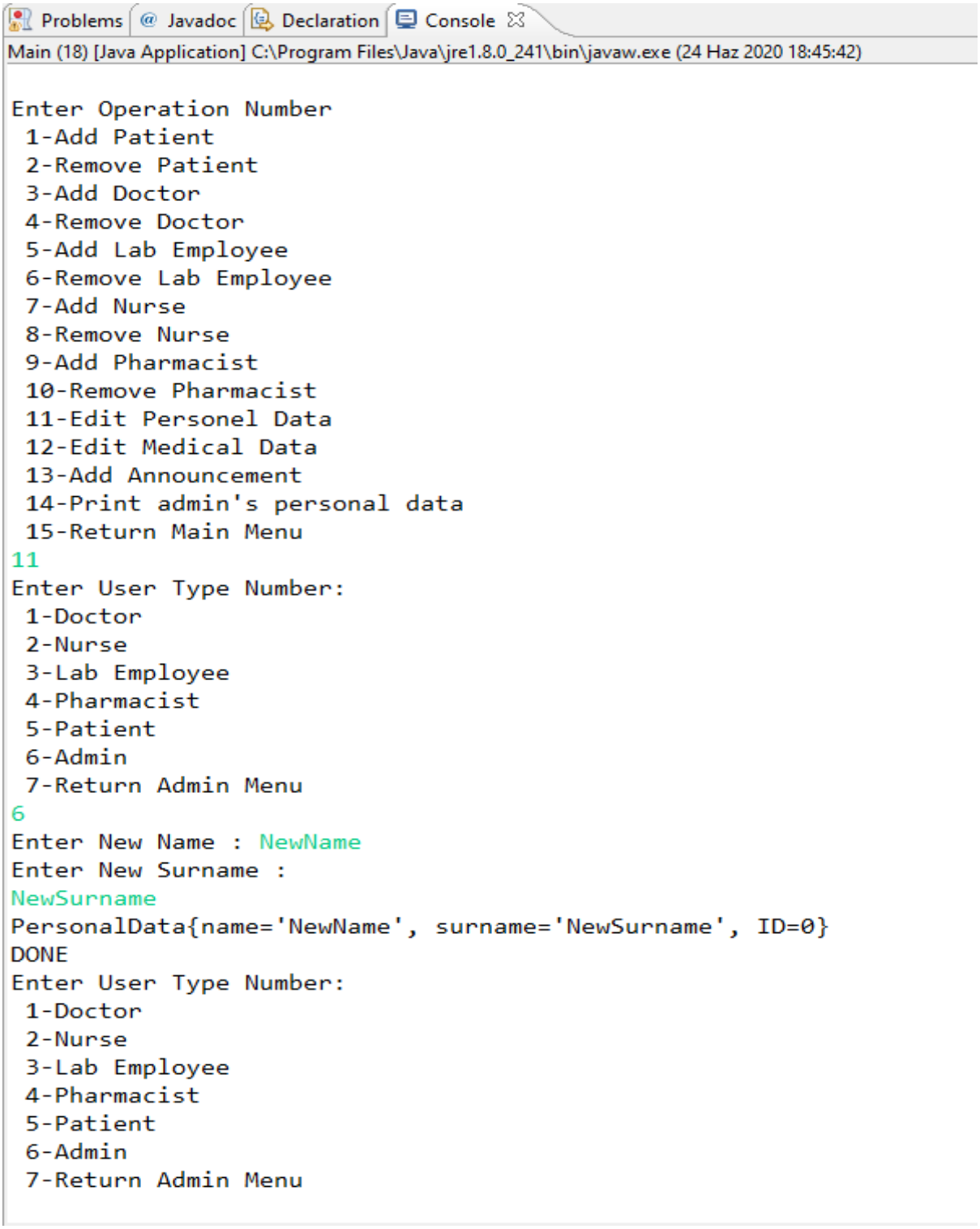
10.1.1 Admin Operation Test Result

10.1.1.1 Admin Login Invalid Test Result



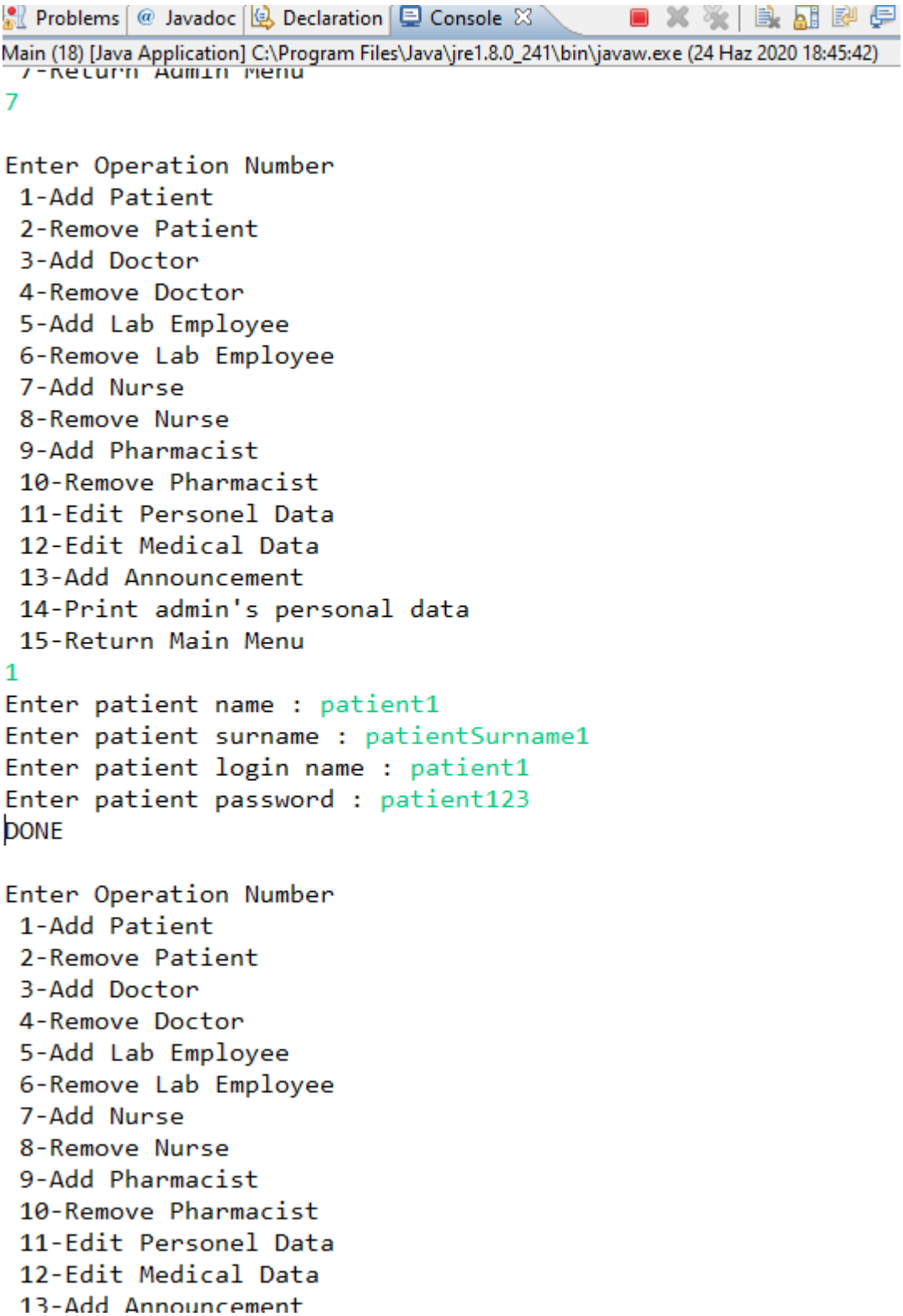
- When the program is run, we will see a menu. When an invalid login name and password is entered for Admin, it gives "wrong password" warning.

10.1.1.2 Admin Login Change Data Test Result

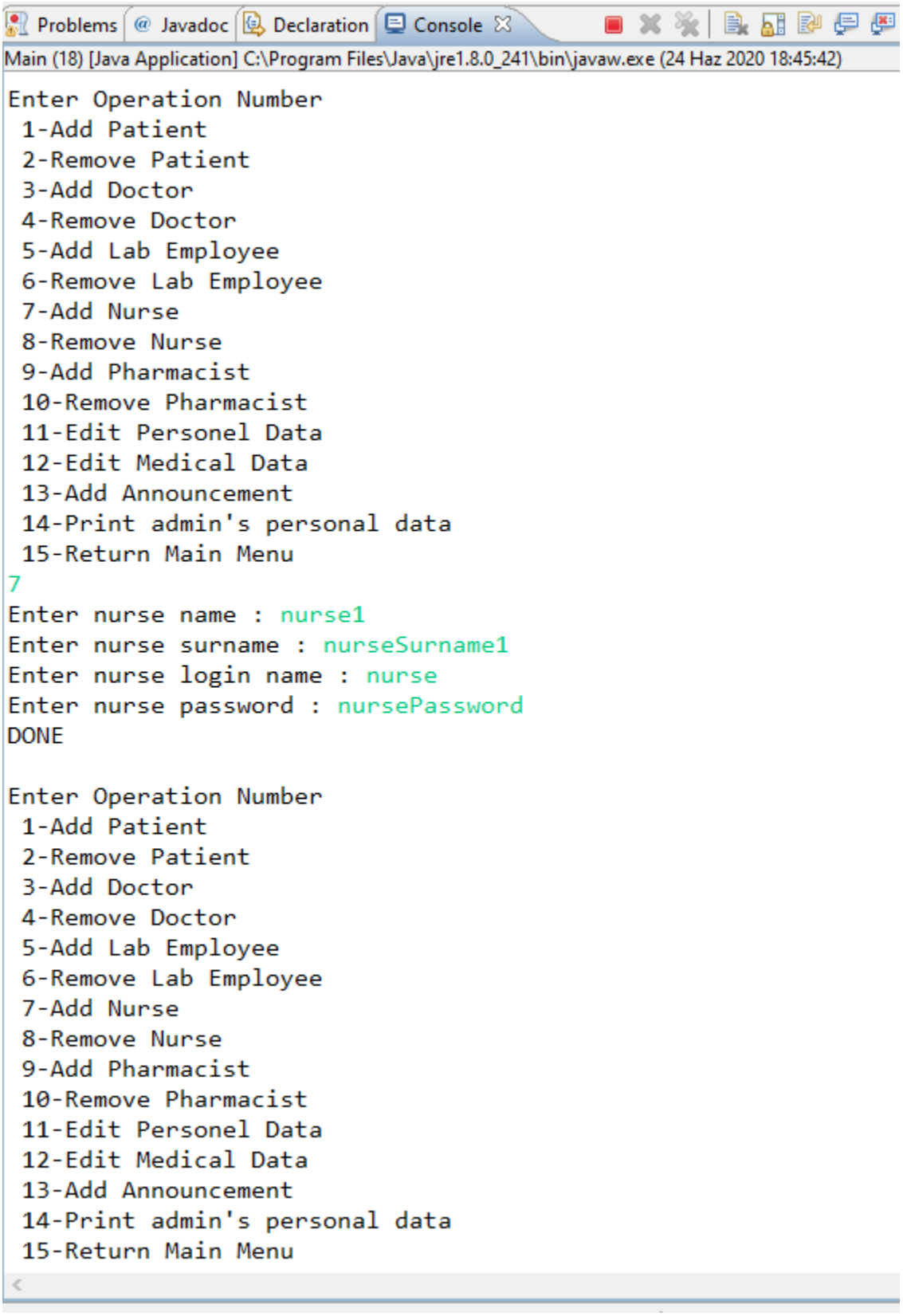


- Only admin can change login data. After the admin login, data exchange results for admin are as above

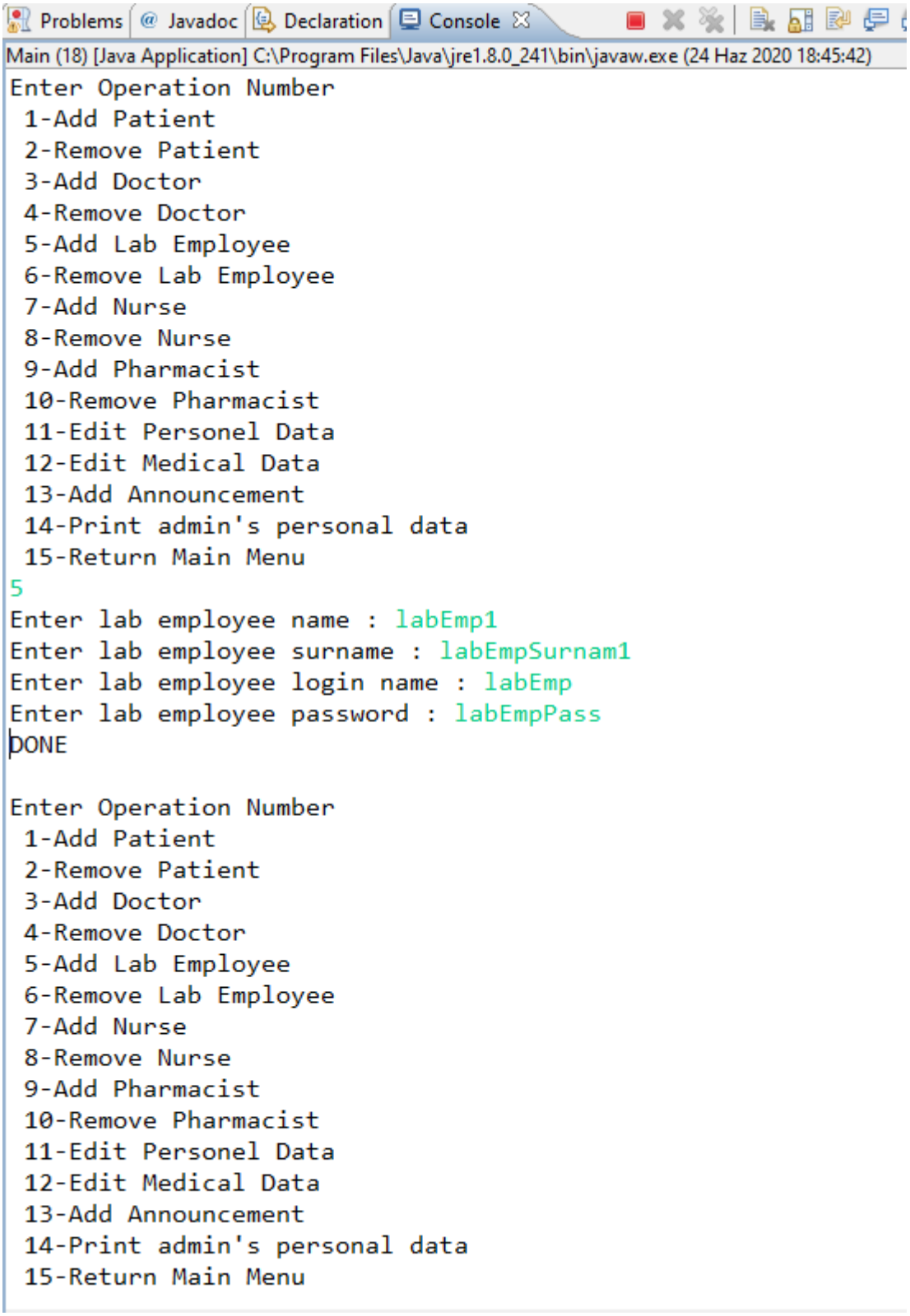
10.1.1.3 Admin Add Patient Test Result



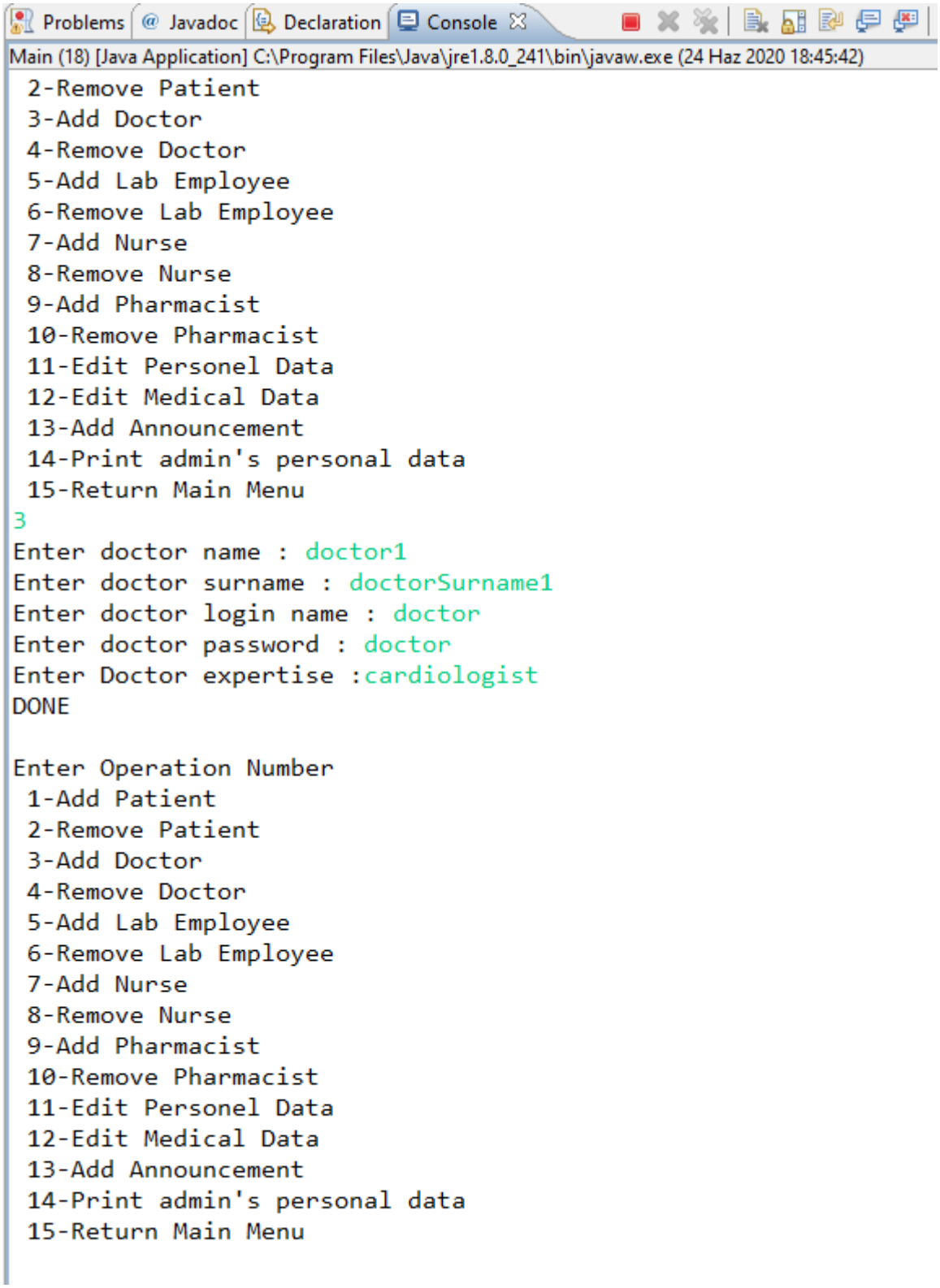
10.1.1.4 Admin Add Nurse Test Result



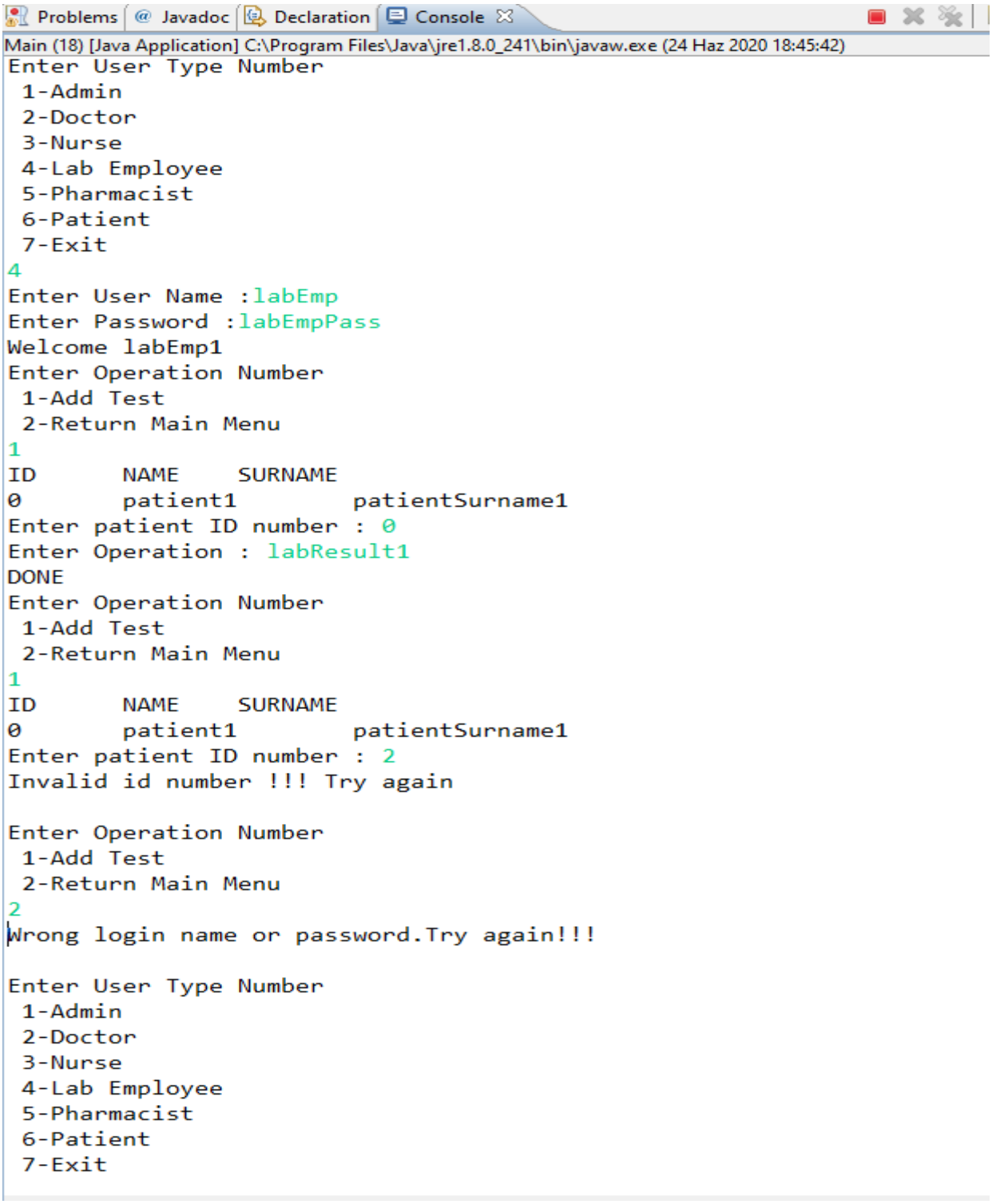
10.1.1.5 Admin Add Lab Employee Test Result



10.1.1.6 Admin Add Doctor Test Result

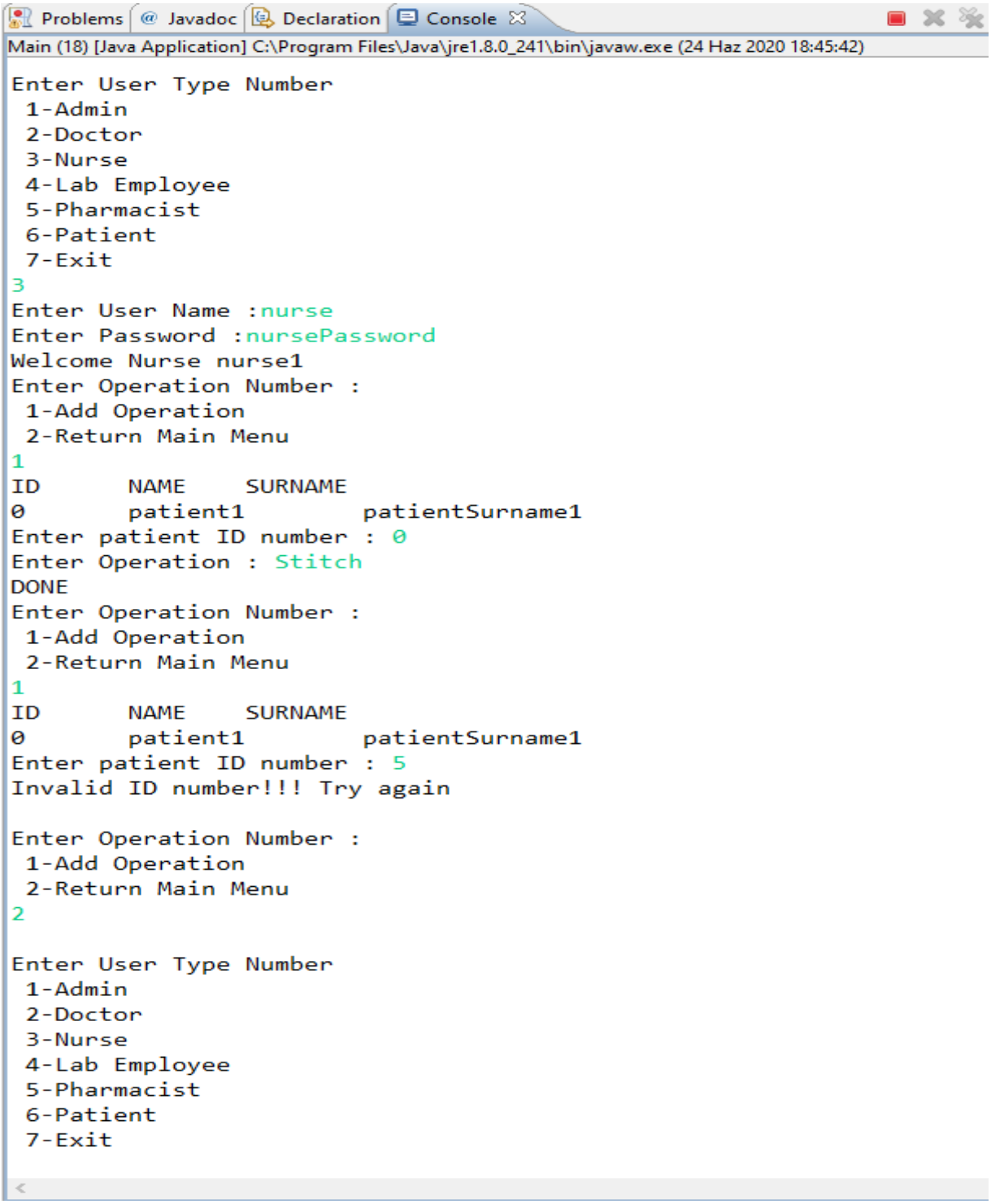


10.1.2 Lab Employee Operation Test Result



- After logging in for the lab employee from the menu, by entering the correct user and password, the necessary test is added with the required patient ID number.

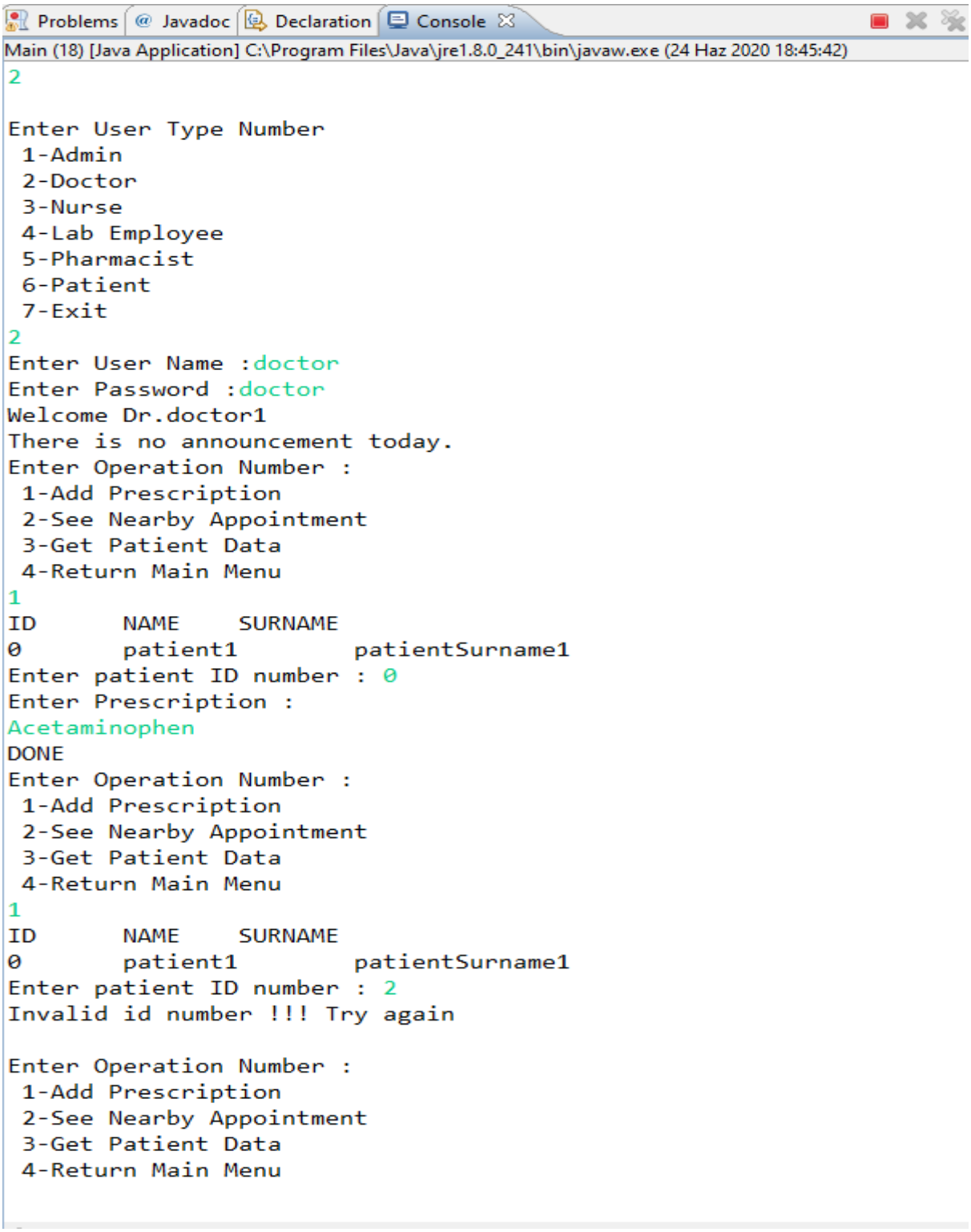
10.1.3 Nurse Operation Test Result



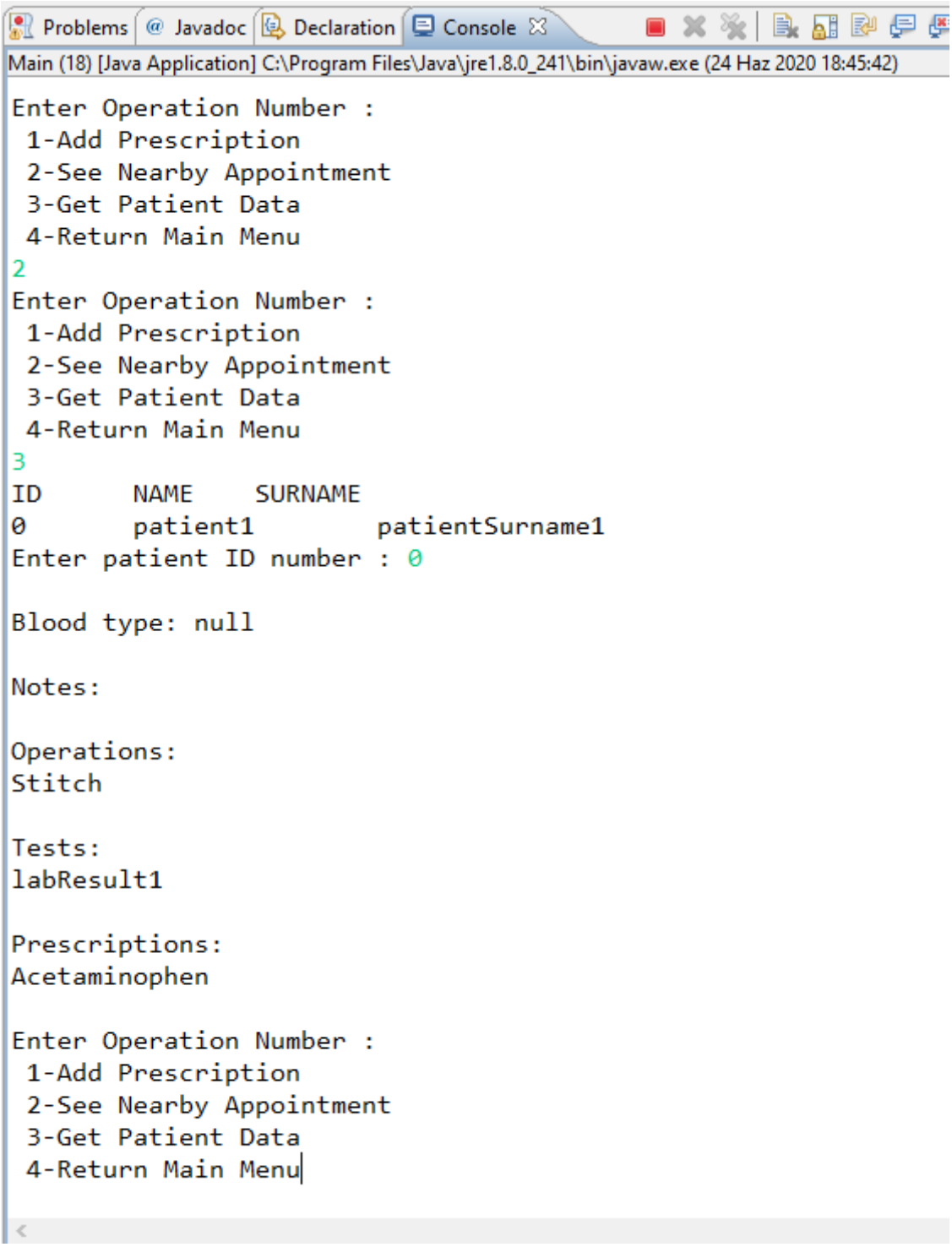
Ü

- After logging in for the nurse from the menu, by entering the correct user and password, the necessary test is added with the required patient ID number.

10.1.4 Doctor Operation Test Result

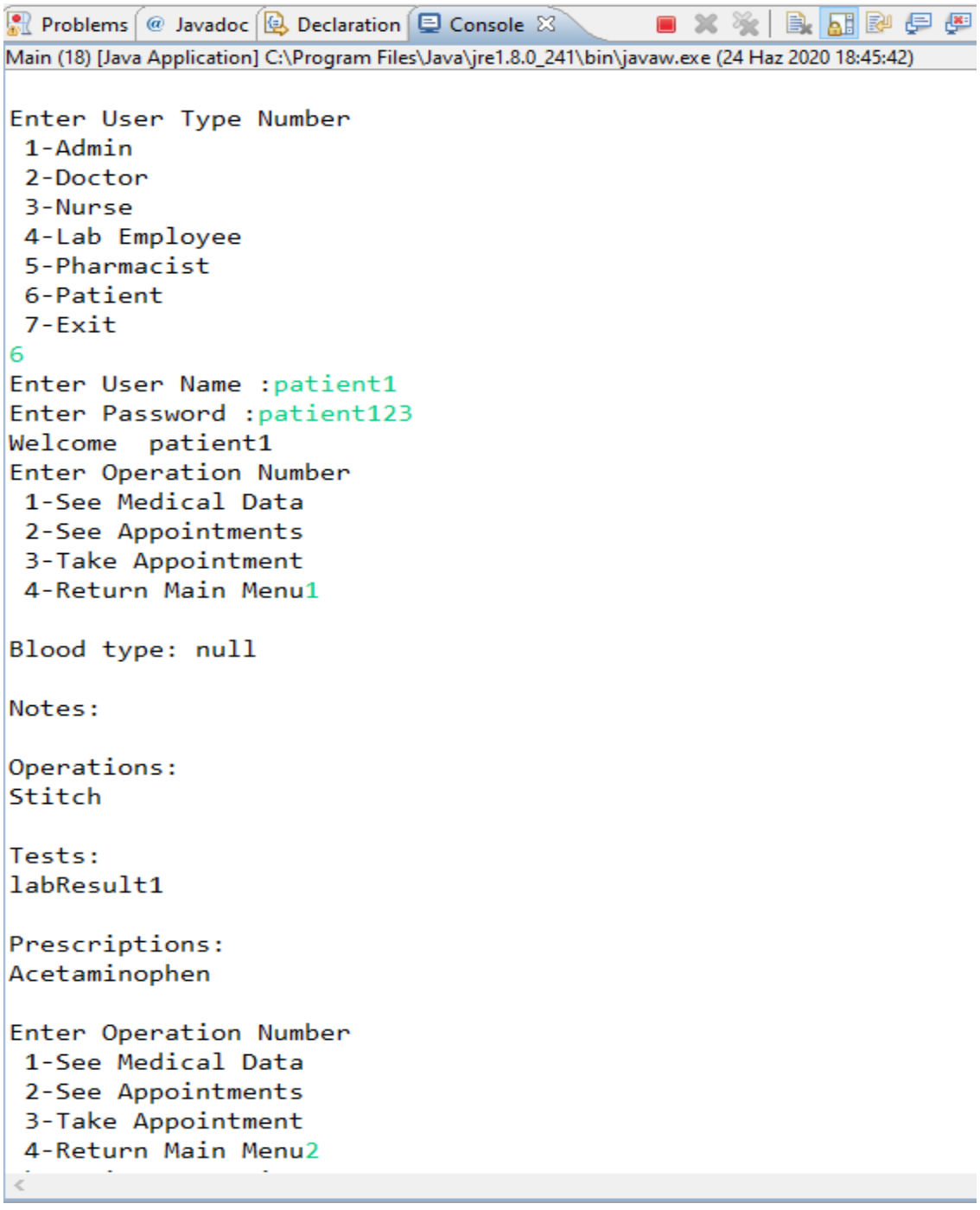


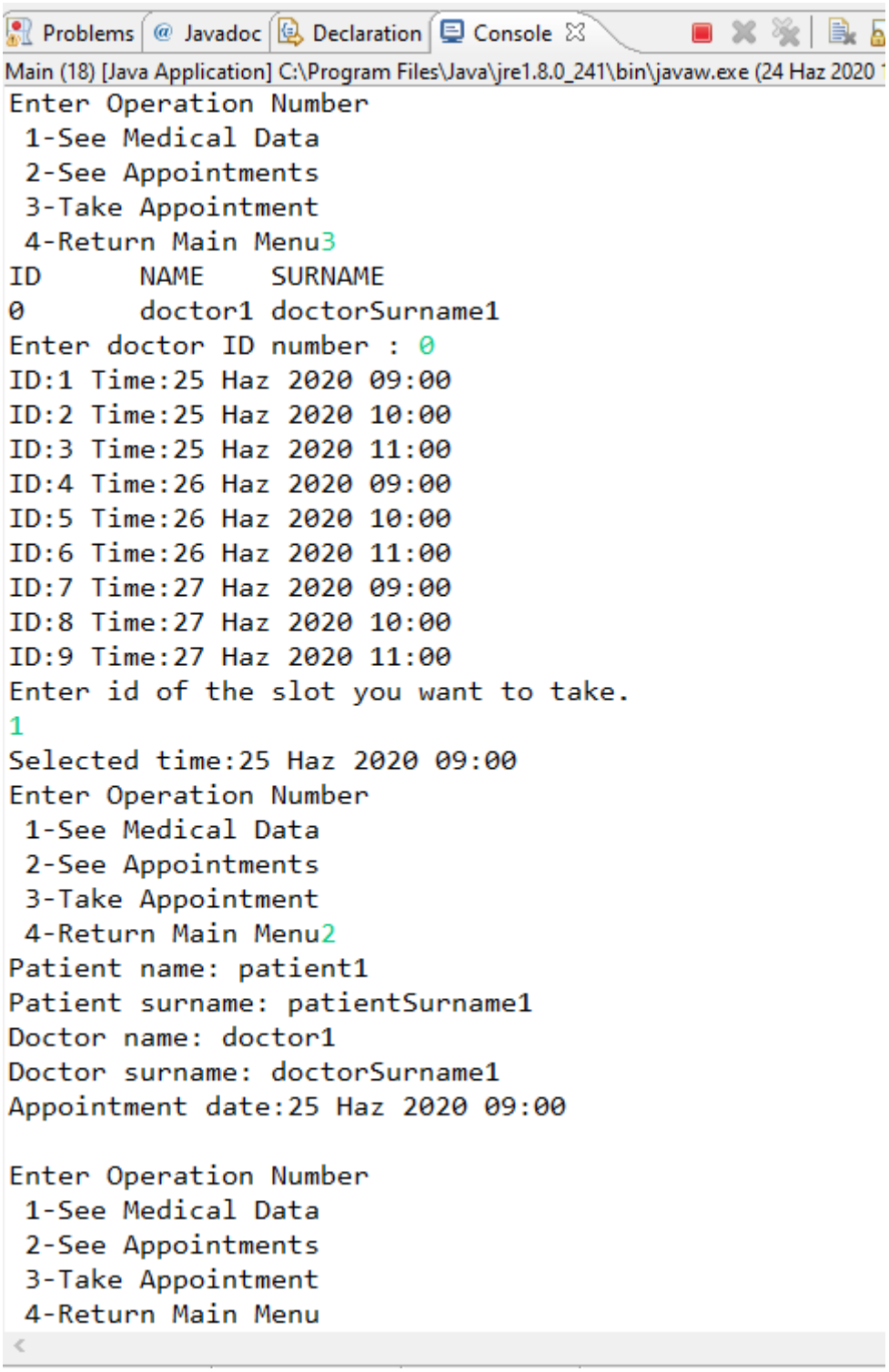
cdffgng



- After entering the doctor, the doctor performs the operations he wants with the directions in the menu.

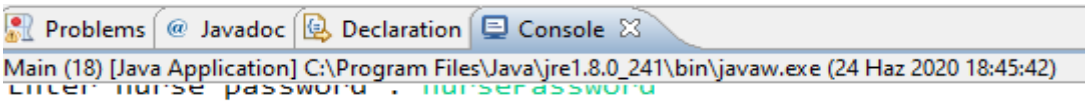
10.1.5 Patient Operation Test Result




Problems
Javadoc
Declaration
Console
Main (18) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (24 Haz 2020
Enter Operation Number
1-See Medical Data
2-See Appointments
3-Take Appointment
4-Return Main Menu3
ID NAME SURNAME
0 doctor1 doctorSurname1
Enter doctor ID number : 0
ID:1 Time:25 Haz 2020 09:00
ID:2 Time:25 Haz 2020 10:00
ID:3 Time:25 Haz 2020 11:00
ID:4 Time:26 Haz 2020 09:00
ID:5 Time:26 Haz 2020 10:00
ID:6 Time:26 Haz 2020 11:00
ID:7 Time:27 Haz 2020 09:00
ID:8 Time:27 Haz 2020 10:00
ID:9 Time:27 Haz 2020 11:00
Enter id of the slot you want to take.
1
Selected time:25 Haz 2020 09:00
Enter Operation Number
1-See Medical Data
2-See Appointments
3-Take Appointment
4-Return Main Menu2
Patient name: patient1
Patient surname: patientSurname1
Doctor name: doctor1
Doctor surname: doctorSurname1
Appointment date:25 Haz 2020 09:00
Enter Operation Number
1-See Medical Data
2-See Appointments
3-Take Appointment
4-Return Main Menu

10.1.6 Pharmacist Operation Test Result

10.1.6.1 Add Pharmacist Test Res*ult*




```
Enter nurse password : nurserassword
DONE

Enter Operation Number
 1-Add Patient
 2-Remove Patient
 3-Add Doctor
 4-Remove Doctor
 5-Add Lab Employee
 6-Remove Lab Employee
 7-Add Nurse
 8-Remove Nurse
 9-Add Pharmacist
 10-Remove Pharmacist
 11-Edit Personel Data
 12-Edit Medical Data
 13-Add Announcement
 14-Print admin's personal data
 15-Return Main Menu
9
Enter pharmacist name : pharmacist
Enter pharmacist surname : pharmacistSurname
Enter pharmacist login name : pharmacist
Enter pharmacist password : pharmacist
Enter pharmacy name
Pharmacy1
DONE

Enter Operation Number
 1-Add Patient
 2-Remove Patient
 3-Add Doctor
 4-Remove Doctor
 5-Add Lab Employee
 6-Remove Lab Employee
 7-Add Nurse
 8-Remove Nurse
 9-Add Pharmacist
 10-Remove Pharmacist
 11-Edit Personel Data
 12-Edit Medical Data
 13-Add Announcement
 14-Print admin's personal data
 15-Return Main Menu
```

Github Link of the Project

https://github.com/cse222grup3/healthsystemapplication

Password: Hj7IUGRS