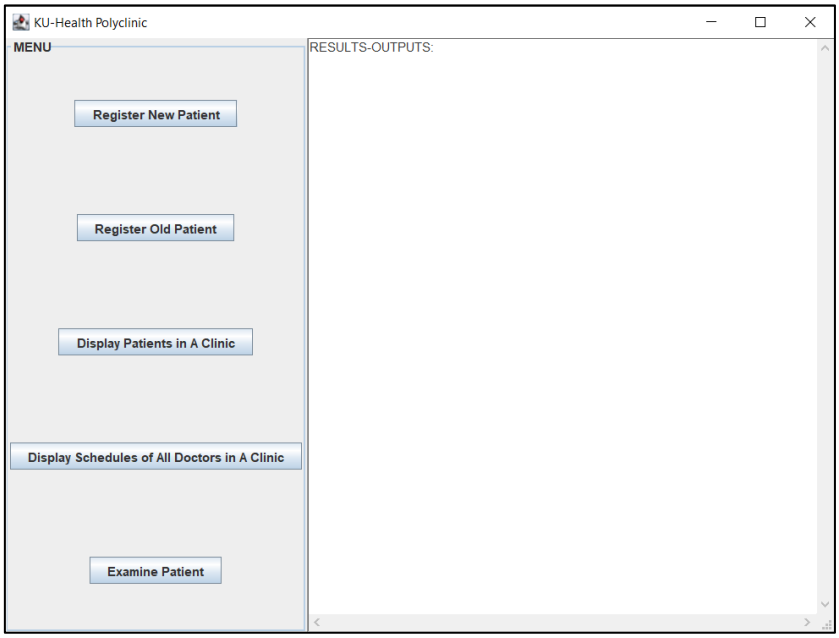
COMP132 Assignment-1 Report

Name: İsmail Ozan Kayacan
ID: 69103

GUI



BUTTONS:

1- **"Register New Patient":** Takes the inputs "name, gender, age, TCKN, old drugs, old illnesses, clinic, doctor and hour" from user, respectively. Then, register patient to the selected doctor-hour, meanwhile updates necessary data such as doctor's schedule, doctor's available hour.
2- **"Register Old Patient":** If a patient's information already exists in the policlinic system, this button should be used. It lists old patients, and asks user to select one of them. Then asks clinic, doctor and hour. Then, register the patient. As previous button, it updates necessary data.
3- **"Display Patients in A Clinic":** It asks user to select a clinic, then prints all registered patients in this clinic, sorted by age (youngers first), if ages are equal then females are first.
4- **"Display Schedules of All Doctors in A Clinic":** It asks user to select a clinic, then then show all doctors schedules in this clinic.
5- **"Examine Patient":** It asks user to choose clinic and doctor. Then, it lists all patient of selected doctor, user select one, then, shows the information of selected patient, ask user to enter examination results (diagnosed illness and suggested drugs.) It checks whether the suggested drugs conflicts with the patient's old drugs. It gives appropriate drugs to patient, then calculate the cost patient must pay, income of doctor, profit of clinic and income of drug's firms. It also updates necessary data such as patient's drug and illness history.

TEXTAREA: The TextArea at the east of the GUI, shows the results and history after the buttons are used.

CLASSES

infoUpdatable.java: (For polymorphism (interface))

Interface containing *public void addOldDrugs(String drugNames)* and *public void addOldIllnesses(String illnesses)* methods, which Patient.java implements to update data during registration and after examination.

Named.java: (For inheritance)

Superclass containing protected *String name* field, *public String getName()* and *public void setName(String name)* methods (getter-setter), which are all common in Patient, Doctor, Clinic, Drug classes. By extending this SuperClass in those classes, I get rid of the need to implement getters and setters for 'name' in those classes.

Patient.java:

Concrete class which extends Named, implements infoUpdatable and Comparable<Patient> (For sorting patients). It has data fields *String gender*, *int age*, *int tckn*, *String oldDrugs*, *String oldIllnesses* and getters and setters for them. It has constructor *public Patient(String name, String gender, int age, int tckn)*. It has *String toString()* method. It also has *public void addOldDrugs(String drugNames)* which adds the input (drugNames) to patient's old drugs and *public void addOldIllnesses(String illnesses)* which adds the input (illnesses) to patient's old illnesses. Since it implements Comparable, it has CompareTo method which sorts patients as wanted - sorted by age (youngers first), if ages are equal then females are first.

Doctor.java:

Concrete class which extends Named. It has data fields *int ID*, *int visitCost*, *TreeMap<String,String> schedule*, *TreeMap<String,Patient> registeredPatients*, *double doctorsProfitPercentage* and getters and setters for them. It has constructor *Doctor(String name, int ID, int visitCost,double doctorsProfitPercentage)*. It has method *public ArrayList<String> getAvailableHours()* which returns an arraylist consisting of available hours of doctor to show in registration.

Drug.java:

Concrete class which extends Named. It has data fields *double price*, *LinkedList<Drug> conflictingDrugs*, *double clinicsProfitPercentage* and getters and setters for them. It has constructor *Drug(String name, double price,double clinicsProfitPercentage, LinkedList<Drug> conflictingDrugs)*.

Clinic.java:

Concrete class which extends Named. It has data field *LinkedList<Doctor> doctorList* and getters and setters for them. It has two type of constructor which are Clinic(String name) and Clinic(String name, LinkedList<Doctor> doctorList). (Note: First one is used in main). It also has a method *public void addDoctor(Doctor d)* which adds input d to doctorList of the clinic.

MainFrame.java: (inheritance, since extends JFrame)

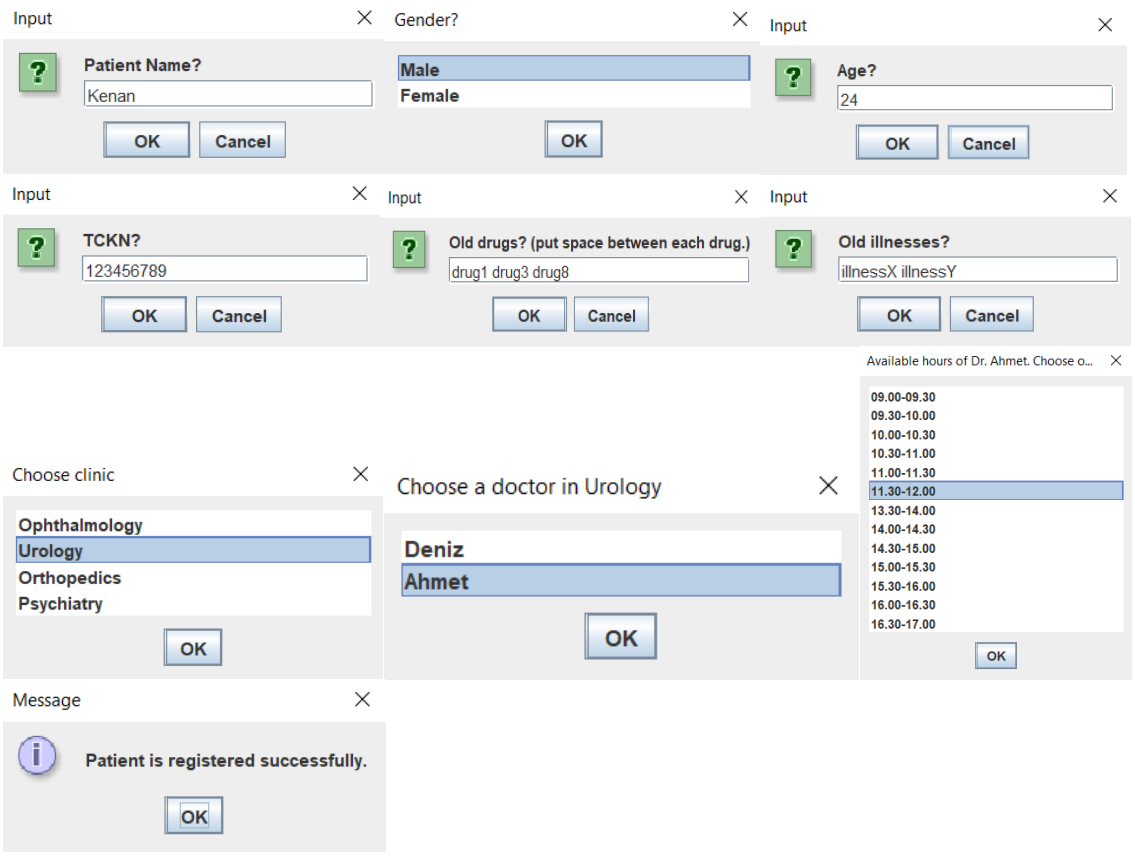
Concrete class which extends JFrame, It has data fields *JPanel panel*, *TextArea textArea*, and getters and setters for them. It has constructor *MainFrame(String title).*

MainClass.java:

Class including main method. In this class, clinics, doctor and drugs created for test. Also, panel's components (buttons) are created here. Moreover, ActionListener for each button is created. (Button's functions are already explained in first page of report.)

Example Scenario with Screenshots

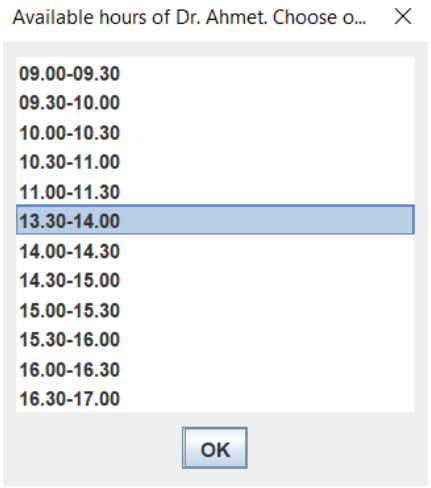
1-Register a new patient (using "Register New Patient" button):



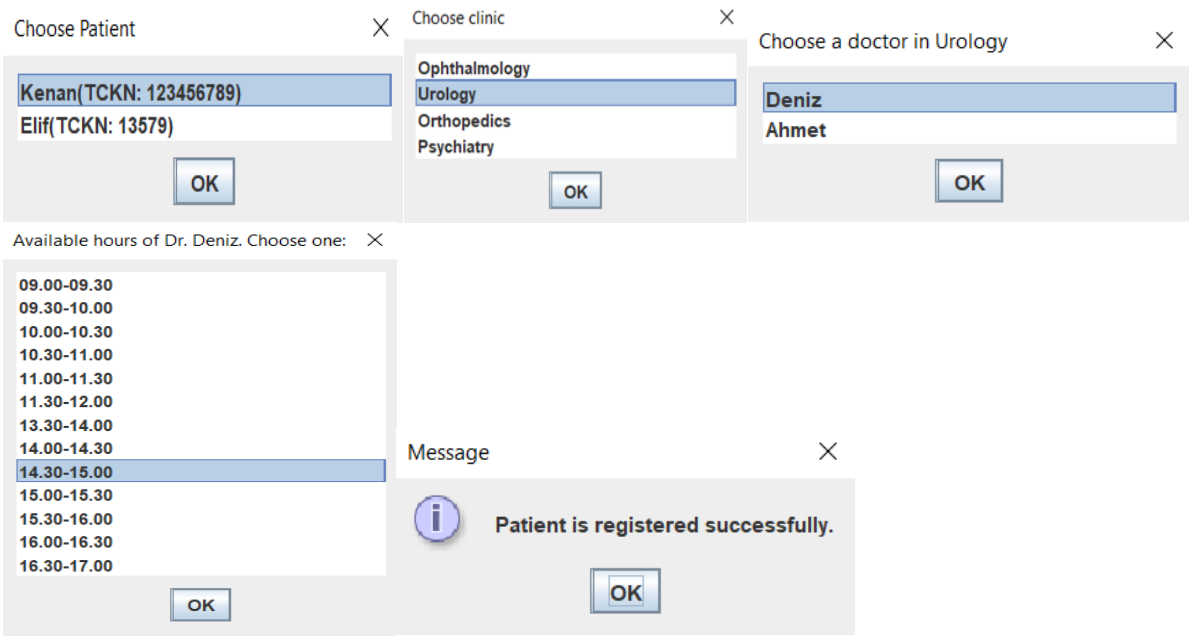
(Input windows appear one by one.)

2-Register new patient with the information *[Name=Elif, Gender=Female, Age=24, TCKN=13579, Old Drugs= drug5, Old Illnesses= illnessZ]* **to the same doctor (using "Register New Patient" button):**

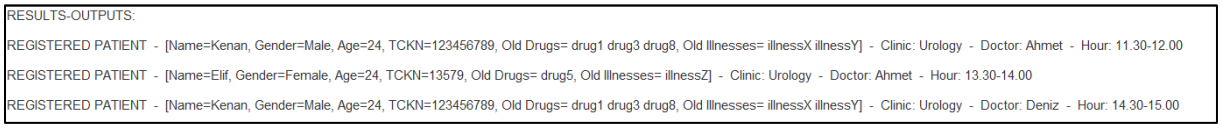
(I did not add screenshots of getting input for name, age etc. because they are same with previous one. But I add available hours of the doctor to show that previous patient's hour (11.30-12.00) is not visible for the next patient.)



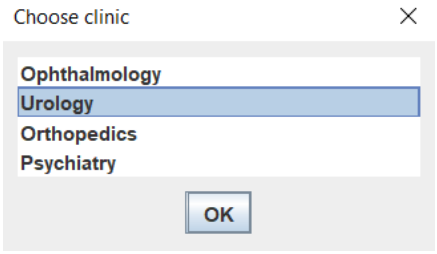
3-Register old patient (Kenan -first patient in the scenario-) to another doctor. (using "Register Old Patient" button):



AFTER FIRST 3 STEPS, SCREENSHOT OF TEXTAREA (EAST OF FRAME):



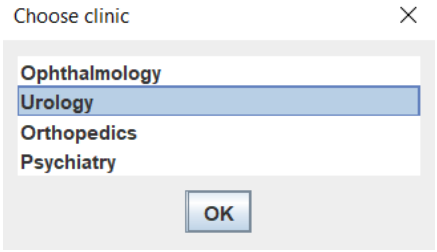
4- Display patients in urology clinic (using “Display Patients in A Clinic” button):



OUTPUT OF 4TH STEP IN THE TEXT AREA:

```
Patients Registered In Urology:
[Name=Elif, Gender=Female, Age=24, TCKN=13579, Old Drugs= drug5, Old Illnesses= illnessZ]
[Name=Kenan, Gender=Male, Age=24, TCKN=123456789, Old Drugs= drug1 drug3 drug8, Old Illnesses= illnessX illnessY]
[Name=Kenan, Gender=Male, Age=24, TCKN=123456789, Old Drugs= drug1 drug3 drug8, Old Illnesses= illnessX illnessY]
```

5-Display all doctor’s schedules in urology clinic (using “Display Schedules of All Doctors in A Clinic” button):



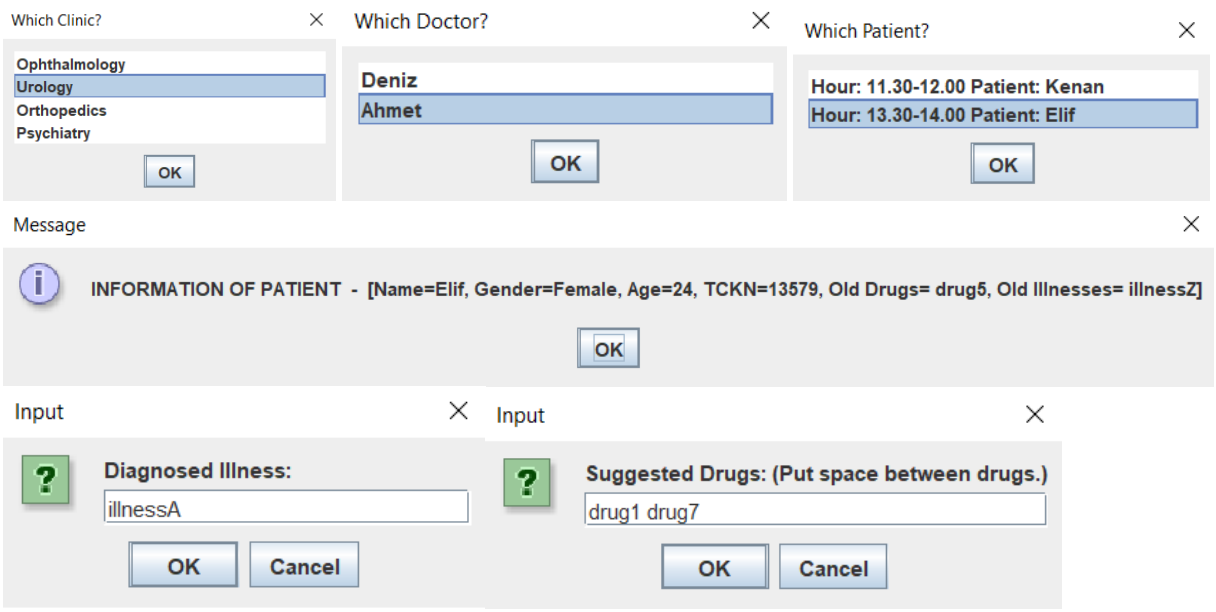
OUTPUT OF 5TH STEP IN THE TEXT AREA:

```
Schedule Of Doctors In Urology Clinic:

Deniz:
09.00-09.30 : -
09.30-10.00 : -
10.00-10.30 : -
10.30-11.00 : -
11.00-11.30 : -
11.30-12.00 : -
13.30-14.00 : -
14.00-14.30 : -
14.30-15.00 : [Name=Kenan, Gender=Male, Age=24, TCKN=123456789, Old Drugs= drug1 drug3 drug8, Old Illnesses= illnessX illnessY]
15.00-15.30 : -
15.30-16.00 : -
16.00-16.30 : -
16.30-17.00 : -

Ahmet:
09.00-09.30 : -
09.30-10.00 : -
10.00-10.30 : -
10.30-11.00 : -
11.00-11.30 : -
11.30-12.00 : [Name=Kenan, Gender=Male, Age=24, TCKN=123456789, Old Drugs= drug1 drug3 drug8, Old Illnesses= illnessX illnessY]
13.30-14.00 : [Name=Elif, Gender=Female, Age=24, TCKN=13579, Old Drugs= drug5, Old Illnesses= illnessZ]
14.00-14.30 : -
14.30-15.00 : -
15.00-15.30 : -
15.30-16.00 : -
16.00-16.30 : -
16.30-17.00 : -
```

6- Examine a patient (using “Examine Patient” button):



OUTPUT OF 6TH STEP IN THE TEXT AREA:

```
Doctor Ahmet Examining a Patient.
INFORMATION OF PATIENT - [Name=Elif, Gender=Female, Age=24, TCKN=13579, Old Drugs= drug5, Old Illnesses= illnessZ]

EXAMINATION FINISHED. RESULTS:

drug1 CANNOT be given because it conflicts with patients old drug: drug5
drug7 is given to patient.

TOTAL EXAMINATION COST: 23.0 (Visit Cost: 13, Drugs cost:10.0)
Clinic's Profit:12.1
Doctor's Income:3.9
Drug Firms' Income:7.0
```

7-To show that the examined patient’s information is updated, register her to another doctor (using “Register Old Patient” button).

(I did not add screenshots because they are same as 2nd step.)

OUTPUT OF 7TH STEP IN THE TEXT AREA:

```
REGISTERED PATIENT - [Name=Elif, Gender=Female, Age=24, TCKN=13579, Old Drugs= drug5 drug7, Old Illnesses= illnessZ illnessA] - Clinic: Orthopedics - Doctor: Ege - Hour: 15.00-15.30
```

As seen, old drugs and old illnesses are updated automatically of the patient after the examination which is on 6th step.