MARMARA UNIVERSITY
FACULTY OF ENGINEERING

CSE2062

Object Oriented Programming

Instructure : Şakir Bingöl

	Student ID Number	Name & Surname
1	150718031	Alperen EROĞLU
2	150718052	Ravzanur TOK
3	150718056	Berat Asrın CAFEROĞLU
4	150718059	Ezgi SÜNE
5	150718060	Zeynep DOĞAN

HOSPITAL MANAGEMENT SYSTEM

In general, this project is a management system to regulate the input and output of a hospital's service, staff and patients.

In the management of a hospital in real life, there are many arrangements that are necessary for the service and income to meet each other. For example, what will happen to the patient as a result of the patient's doctor change, the transfer of nurses and patients from the doctor's departure from the hospital, the room arrangement of the bedridden patients, such a system is absolutely necessary.

This project aims to outline the management system of a particular hospital. In the main menu, there are six options numbered sequentially that are ;

1)Hospital   2)Doctor   3)Nurse   4)Patient   5)Disease   6)Close Program

The user can get information about doctor/nurse/patient 's name, location(department),disease or medicine and also utilities of hospital by different submenus. We can determine the individuals by their ID. Additionally, in submenus, there will be options for doctor, nurse and patient ;

New register   Individual info   All registration   Deregistration

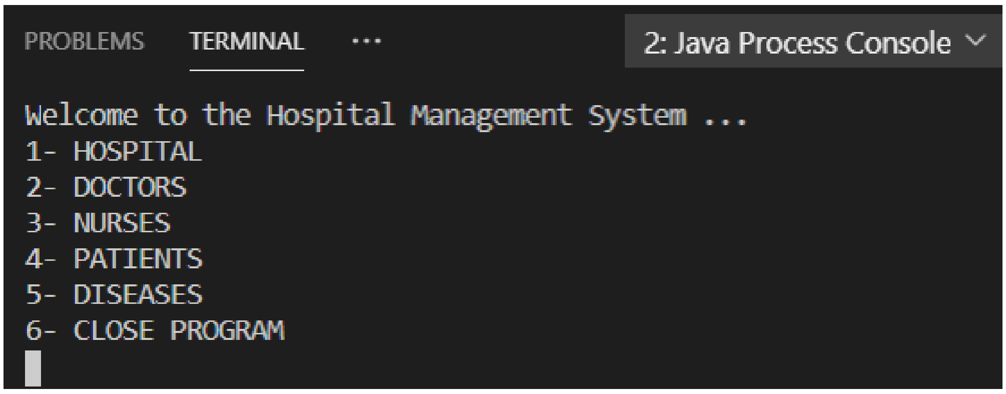
If the user wants to insert new doctor, nurse or patient, the user comes to the option to ***new*** and fills in the information about the person. If the user wants to remove doctor ,nurse or patient, the user comes to the option to ***show all*** and see all person and chooses ***remove*** then removes the person by typing person id. If the extracted doctor has patient, this patient is transferred to the different doctor who have same department with the deleted doctor. In the worst case scenario, it is requested to be transferred from the hospital.

Every nurses have related doctor. When the new nurse is inserted, the user should enter the related doctor's id and also related doctor can be changed in the window that shows all nurses and every patient have related doctor like nurses.

The part to select information about the diseases is found in the window when the user type *5* .There are five diseases is defined for each department exemplary, the medicine to be used for this disease are also shown in the window.

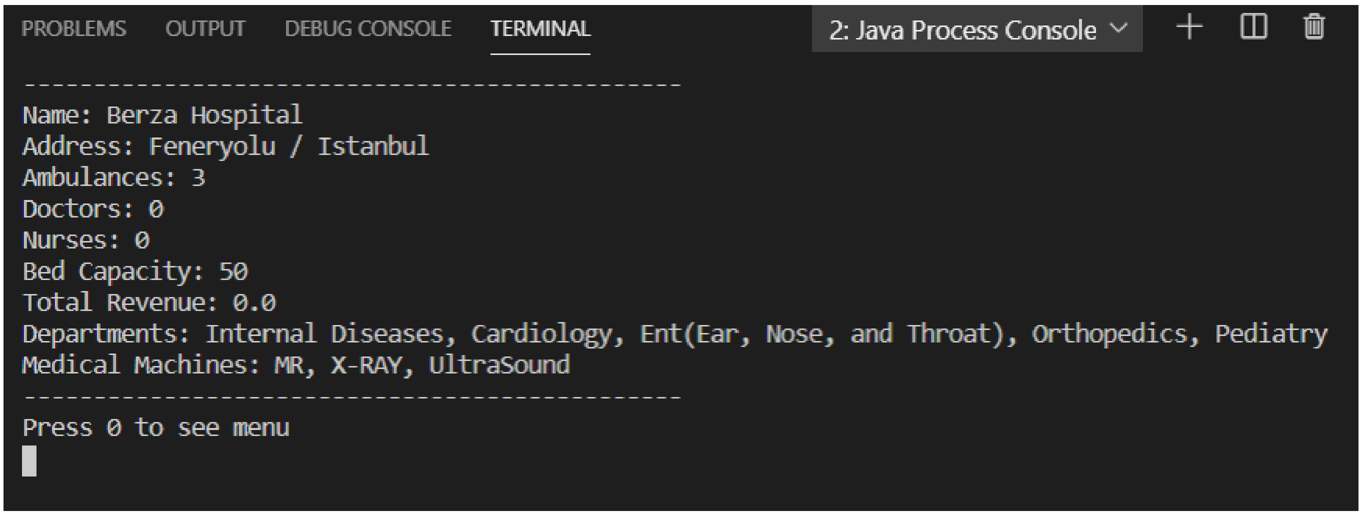
All these changes can be observed in the hospital tab in the main window. For example, when a new doctor or nurse is added or removed ,the number of doctors or nurses increases or decreases and when the inpatient is inserted, income of hospital increases depending on the position of the doctor and the day of the patient's hospitalization. Furthermore, each new window the user opens has a section to return to the main menu and also this main menu has a section to close the program.

Now let's get to know the menus and try various combinations to understand the working principle of the system.



```
Welcome to the Hospital Management System ...
1- HOSPITAL
2- DOCTORS
3- NURSES
4- PATIENTS
5- DISEASES
6- CLOSE PROGRAM
```

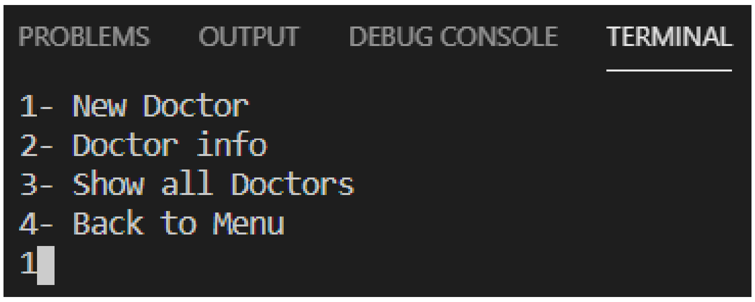
Main Menu



```
-----------------------------------------------
Name: Berza Hospital
Address: Feneryolu / Istanbul
Ambulances: 3
Doctors: 0
Nurses: 0
Bed Capacity: 50
Total Revenue: 0.0
Departments: Internal Diseases, Cardiology, Ent(Ear, Nose, and Throat), Orthopedics, Pediatry
Medical Machines: MR, X-RAY, UltraSound
-----------------------------------------------
Press 0 to see menu
```

1-) Hospital Information

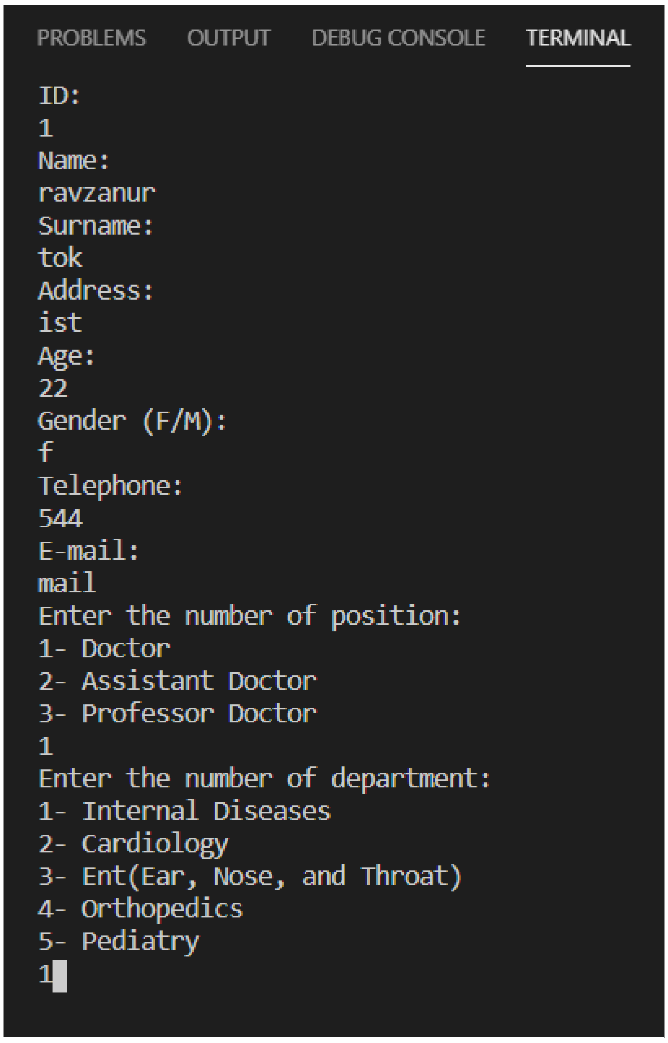
Since no doctors, nurses etc. are added to the system as a start, the total revenue is 0.



```
1- New Doctor
2- Doctor info
3- Show all Doctors
4- Back to Menu
1
```

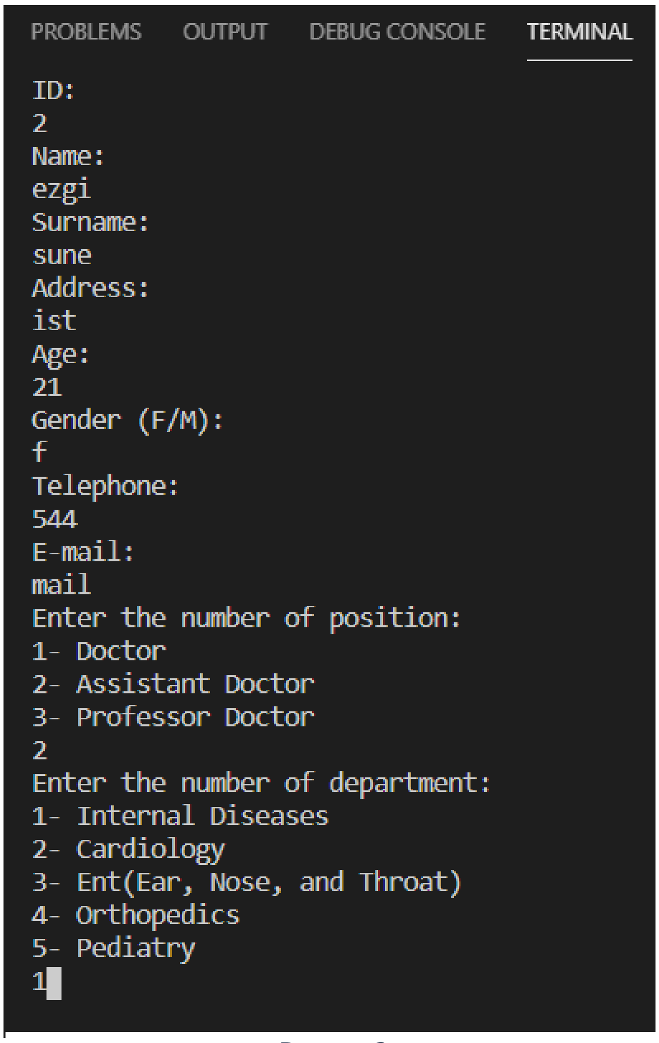
2-) Doctors Menu

Add new doctor: press *1*



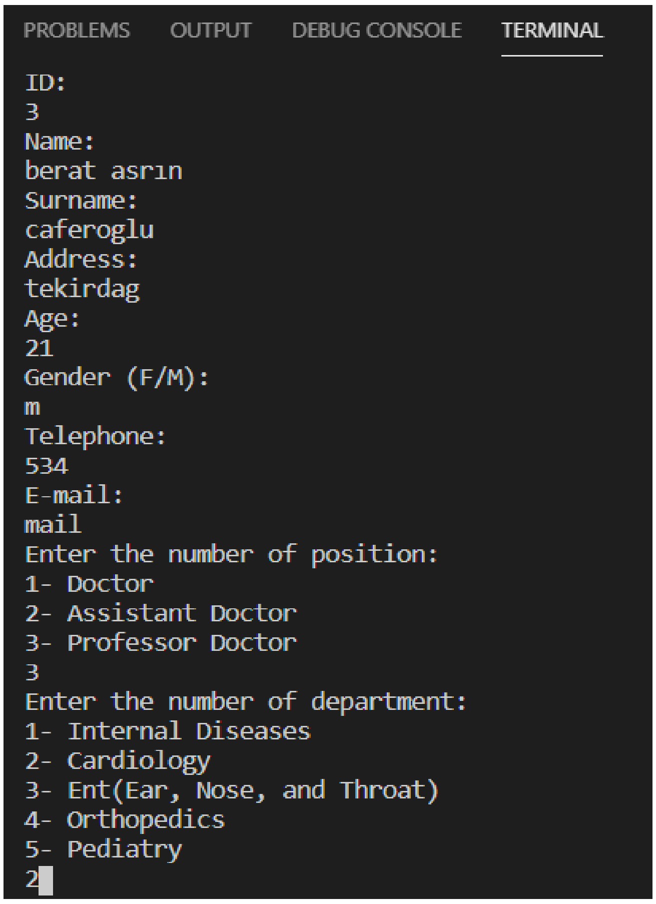
```
ID:
1
Name:
ravzanur
Surname:
tok
Address:
ist
Age:
22
Gender (F/M):
f
Telephone:
544
E-mail:
mail
Enter the number of position:
1- Doctor
2- Assistant Doctor
3- Professor Doctor
1
Enter the number of department:
1- Internal Diseases
2- Cardiology
3- Ent(Ear, Nose, and Throat)
4- Orthopedics
5- Pediatry
1
```

Doctor 1



```
ID:
2
Name:
ezgi
Surname:
sune
Address:
ist
Age:
21
Gender (F/M):
f
Telephone:
544
E-mail:
mail
Enter the number of position:
1- Doctor
2- Assistant Doctor
3- Professor Doctor
2
Enter the number of department:
1- Internal Diseases
2- Cardiology
3- Ent(Ear, Nose, and Throat)
4- Orthopedics
5- Pediatry
1
```

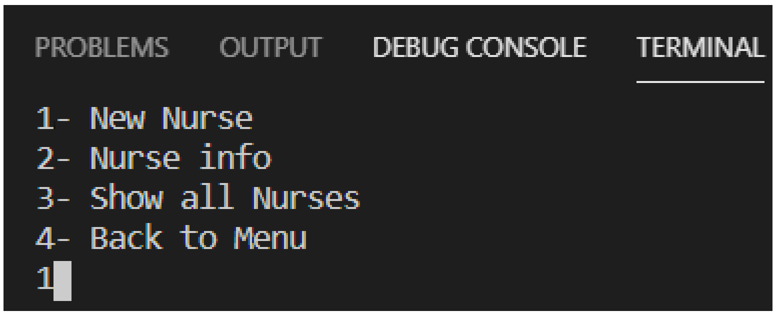
Doctor 2



```
ID:
3
Name:
berat asrın
Surname:
caferoglu
Address:
tekirdag
Age:
21
Gender (F/M):
m
Telephone:
534
E-mail:
mail
Enter the number of position:
1- Doctor
2- Assistant Doctor
3- Professor Doctor
3
Enter the number of department:
1- Internal Diseases
2- Cardiology
3- Ent(Ear, Nose, and Throat)
4- Orthopedics
5- Pediatry
2
```

Doctor 3

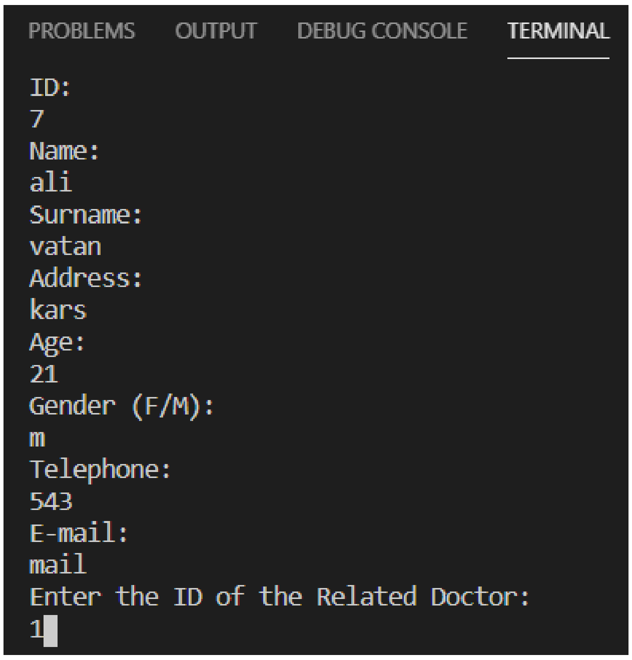
We added 3 doctor with different position.



```
1- New Nurse
2- Nurse info
3- Show all Nurses
4- Back to Menu
1
```

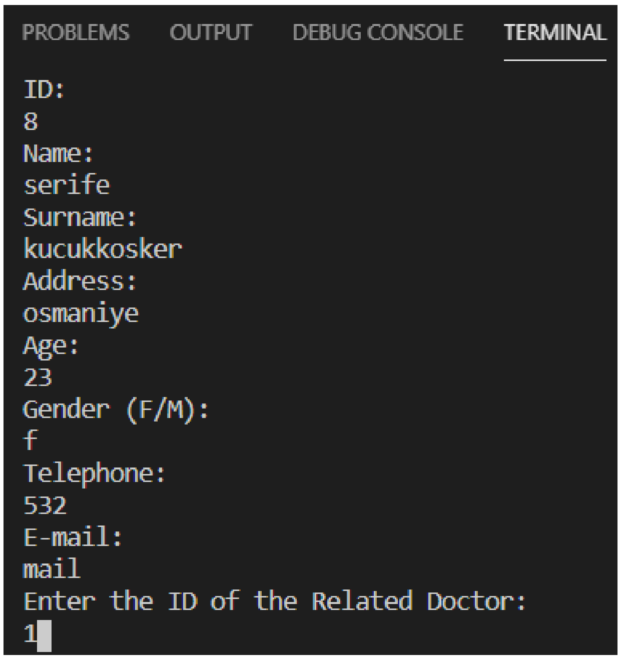
3-) Nurses Menu

Add new nurse: press *1*



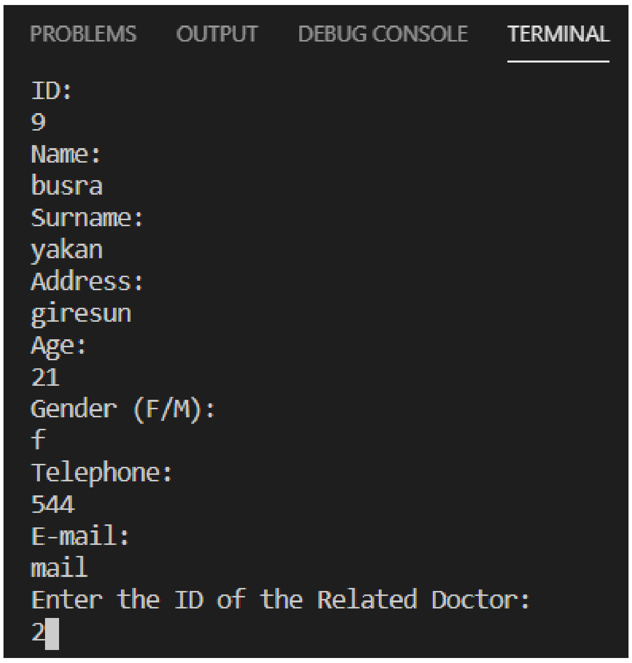
```
ID:
7
Name:
ali
Surname:
vatan
Address:
kars
Age:
21
Gender (F/M):
m
Telephone:
543
E-mail:
mail
Enter the ID of the Related Doctor:
1
```

Nurse 1



```
ID:
8
Name:
serife
Surname:
kucukkosker
Address:
osmaniye
Age:
23
Gender (F/M):
f
Telephone:
532
E-mail:
mail
Enter the ID of the Related Doctor:
1
```

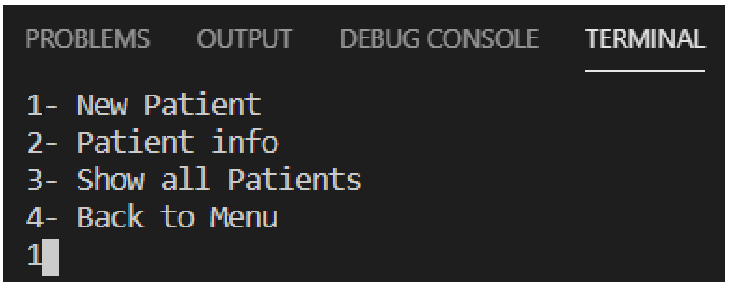
Nurse 2



```
ID:
9
Name:
busra
Surname:
yakan
Address:
giresun
Age:
21
Gender (F/M):
f
Telephone:
544
E-mail:
mail
Enter the ID of the Related Doctor:
2
```

Nurse 3

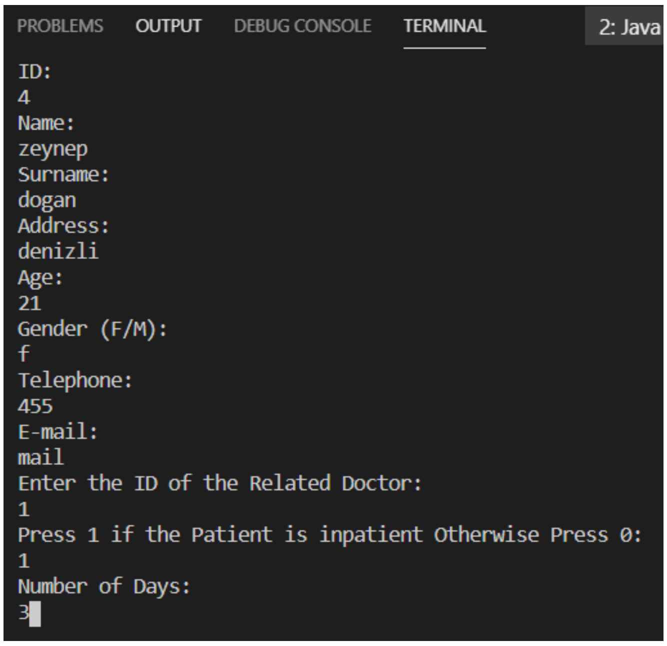
We added three nurses.



```
1- New Patient
2- Patient info
3- Show all Patients
4- Back to Menu
1
```

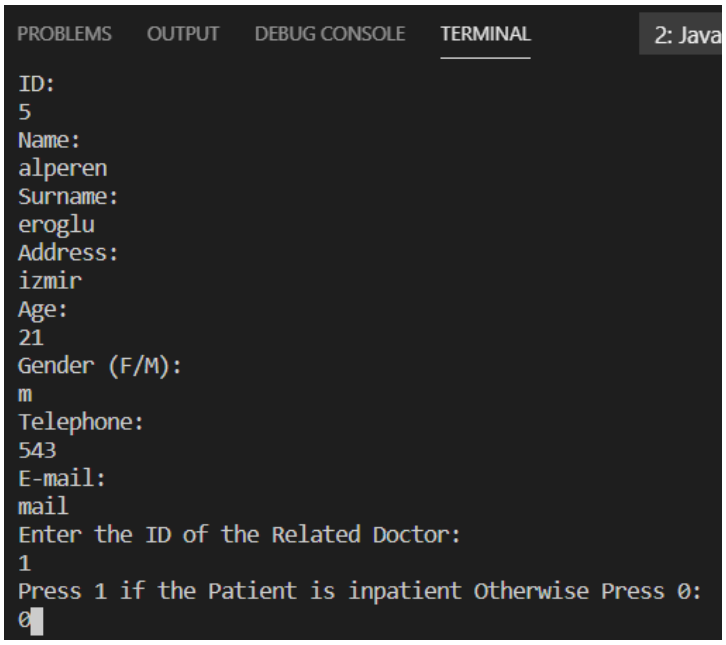
4-) Patients Menu

Add new patinet: press *1*



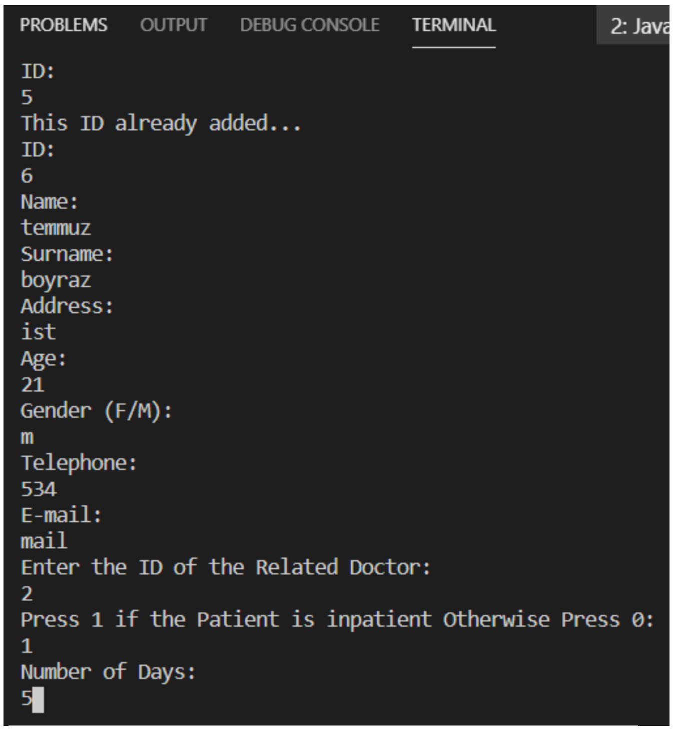
```
ID:
4
Name:
zeynep
Surname:
dogan
Address:
denizli
Age:
21
Gender (F/M):
f
Telephone:
455
E-mail:
mail
Enter the ID of the Related Doctor:
1
Press 1 if the Patient is inpatient Otherwise Press 0:
1
Number of Days:
3
```

Patient 1



```
ID:
5
Name:
alperen
Surname:
eroglu
Address:
izmir
Age:
21
Gender (F/M):
m
Telephone:
543
E-mail:
mail
Enter the ID of the Related Doctor:
1
Press 1 if the Patient is inpatient Otherwise Press 0:
0
```

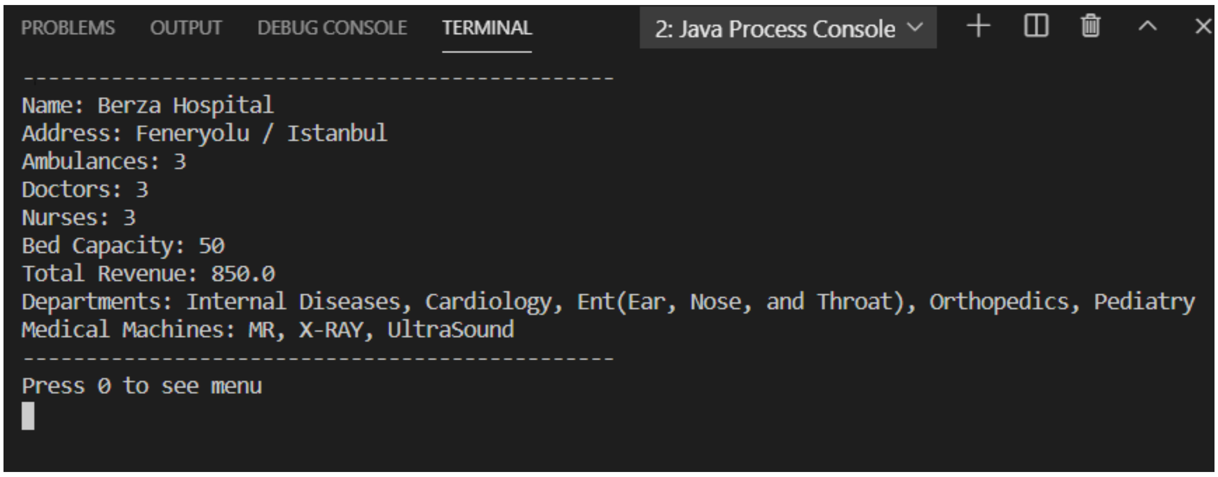
Patient 2



```
ID:
5
This ID already added...
ID:
6
Name:
temmuz
Surname:
boyraz
Address:
ist
Age:
21
Gender (F/M):
m
Telephone:
534
E-mail:
mail
Enter the ID of the Related Doctor:
2
Press 1 if the Patient is inpatient Otherwise Press 0:
1
Number of Days:
5
```

Patient 3

We added three patients.



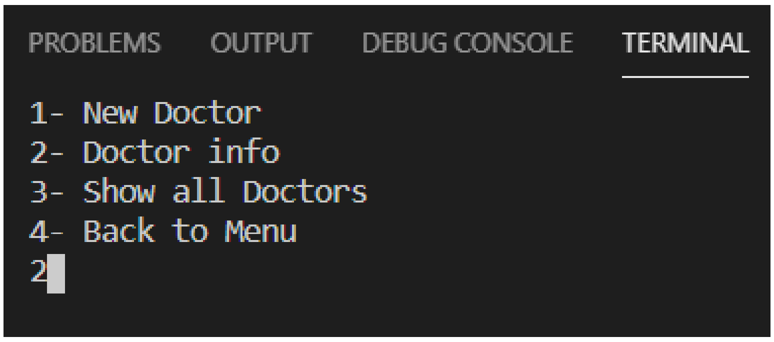
```
----------------------------------------------
Name: Berza Hospital
Address: Feneryolu / Istanbul
Ambulances: 3
Doctors: 3
Nurses: 3
Bed Capacity: 50
Total Revenue: 850.0
Departments: Internal Diseases, Cardiology, Ent(Ear, Nose, and Throat), Orthopedics, Pediatry
Medical Machines: MR, X-RAY, UltraSound
----------------------------------------------
Press 0 to see menu
```

Hospital Menu Check

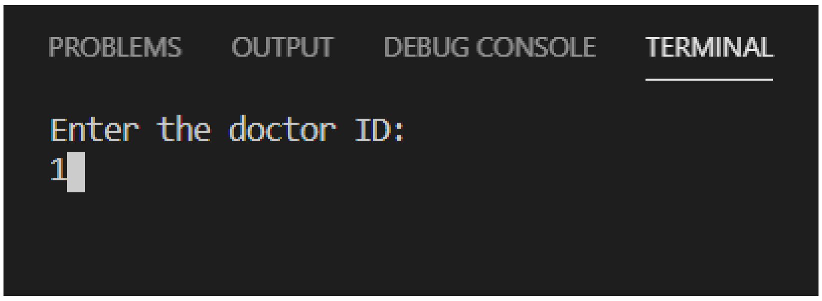
As can be seen from the hospital information, we can see the 3 doctors,3 nurses added and also the income from the patients in the total revenue.

Info **submenus in 2(Doctors)-3(Nurses)-4(Patients) menus**: In order to follow the menus more comfortably, we made a clearscreen (it will be mentioned in the code) so that the inputs are not clustered.

Doctor info menu: press *2*



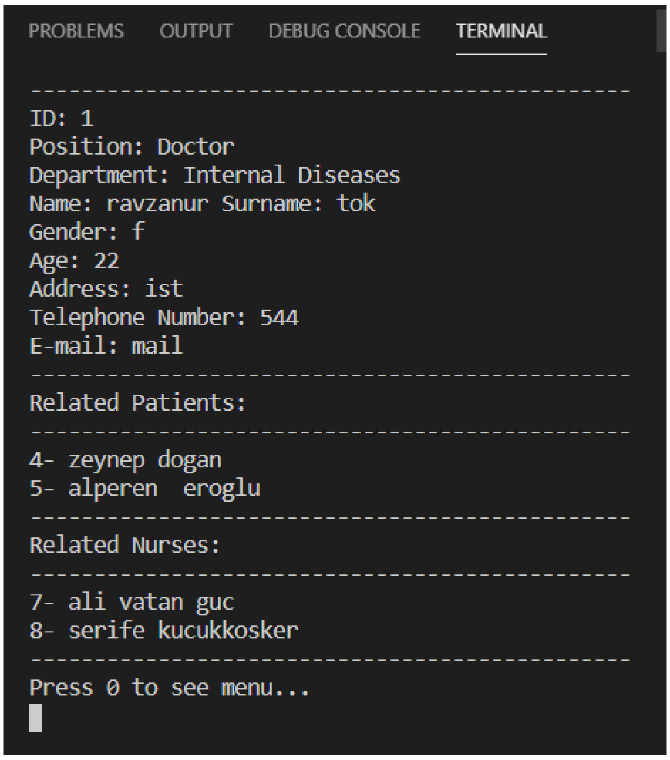
```
1- New Doctor
2- Doctor info
3- Show all Doctors
4- Back to Menu
2
```



```
Enter the doctor ID:
1
```

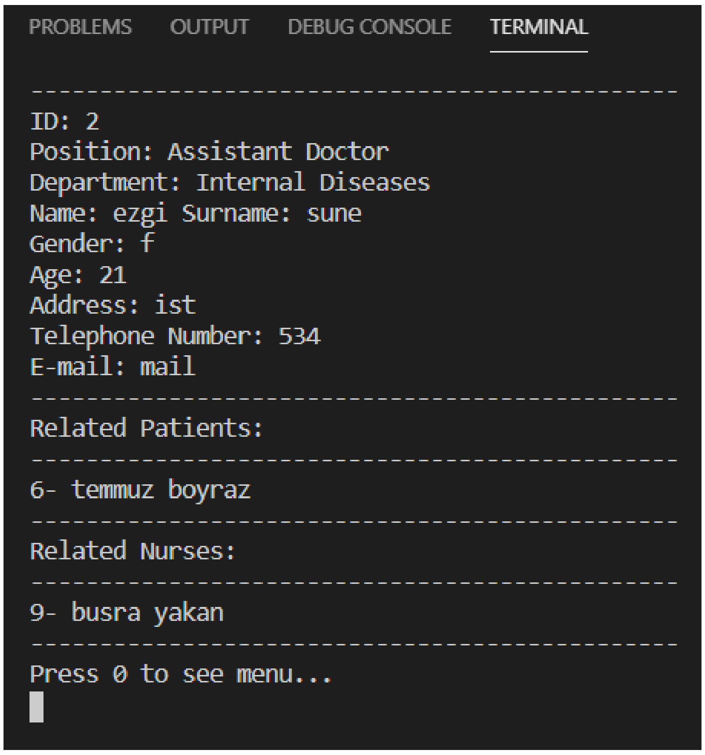
(is also done for doctor 2 and 3)

We can see each doctor's nurse(s) and their patient(s).



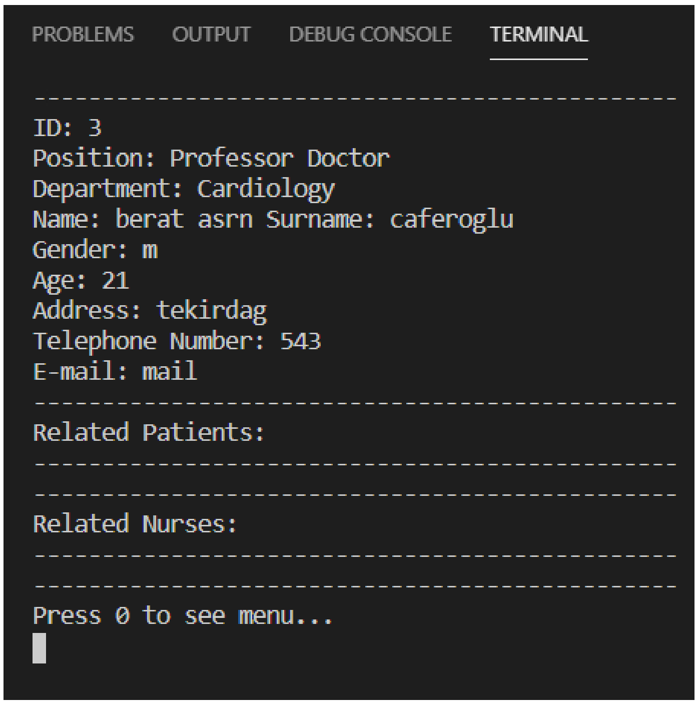
```
----------------------------------------------
ID: 1
Position: Doctor
Department: Internal Diseases
Name: ravzanur Surname: tok
Gender: f
Age: 22
Address: ist
Telephone Number: 544
E-mail: mail
----------------------------------------------
Related Patients:
----------------------------------------------
4- zeynep dogan
5- alperen  eroglu
----------------------------------------------
Related Nurses:
----------------------------------------------
7- ali vatan guc
8- serife kucukkosker
----------------------------------------------
Press 0 to see menu...
```

Doctor 1 Information



```
----------------------------------------------
ID: 2
Position: Assistant Doctor
Department: Internal Diseases
Name: ezgi Surname: sune
Gender: f
Age: 21
Address: ist
Telephone Number: 534
E-mail: mail
----------------------------------------------
Related Patients:
----------------------------------------------
6- temmuz boyraz
----------------------------------------------
Related Nurses:
----------------------------------------------
9- busra yakan
----------------------------------------------
Press 0 to see menu...
```

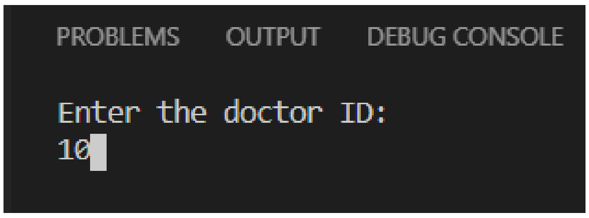
Doctor 2 Information



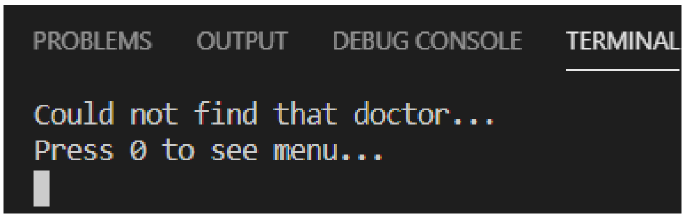
```
----------------------------------------------
ID: 3
Position: Professor Doctor
Department: Cardiology
Name: berat asrn Surname: caferoglu
Gender: m
Age: 21
Address: tekirdag
Telephone Number: 543
E-mail: mail
----------------------------------------------
Related Patients:
----------------------------------------------
----------------------------------------------
Related Nurses:
----------------------------------------------
----------------------------------------------
Press 0 to see menu...
```

Doctor 3 Information

If there is no doctor in the entered id, the below message is printed.

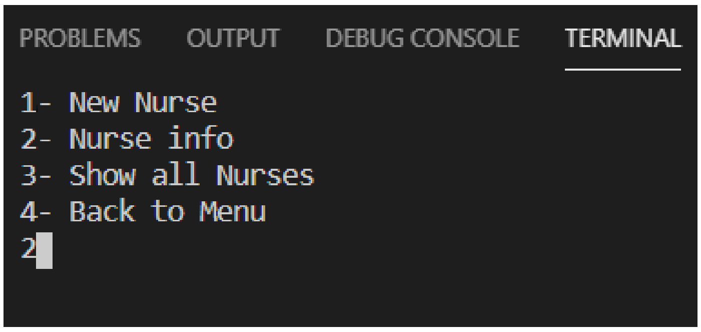


```
Enter the doctor ID:
10
```

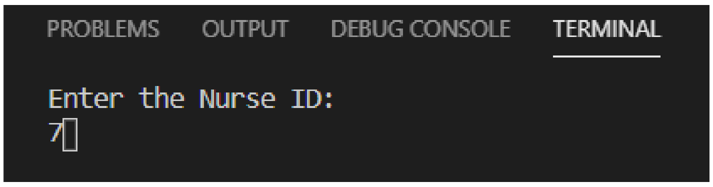


```
Could not find that doctor...
Press 0 to see menu...
```

Nurse info menu: press *2*



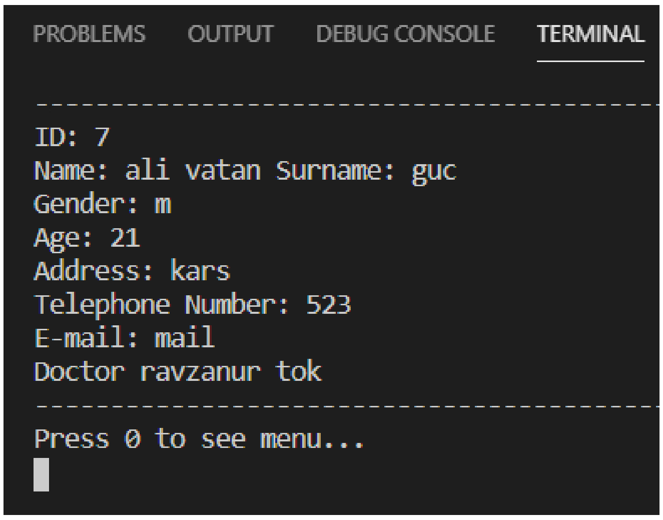
```
1- New Nurse
2- Nurse info
3- Show all Nurses
4- Back to Menu
2
```



```
Enter the Nurse ID:
7
```

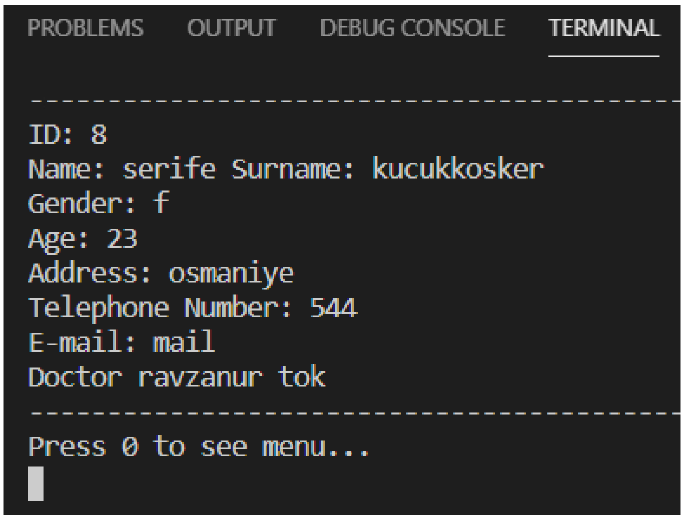
(is also done for nurse 8 and 9)

The doctor to which the nurses are attached can also be seen.



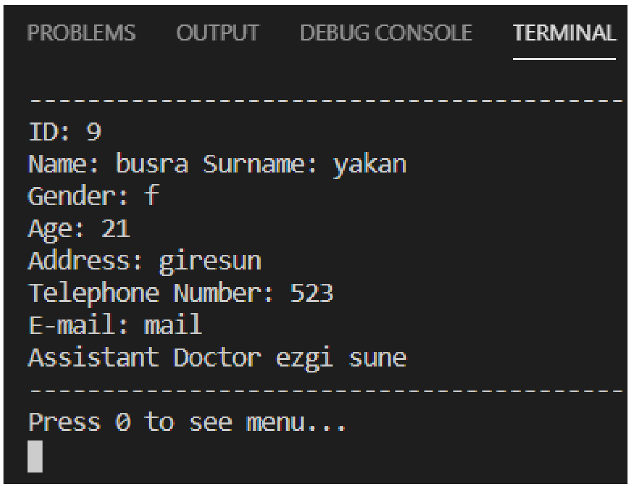
```
-----------------------------------------
ID: 7
Name: ali vatan Surname: guc
Gender: m
Age: 21
Address: kars
Telephone Number: 523
E-mail: mail
Doctor ravzanur tok
-----------------------------------------
Press 0 to see menu...
```

Nurse 1 Information



```
-----------------------------------------
ID: 8
Name: serife Surname: kucukkosker
Gender: f
Age: 23
Address: osmaniye
Telephone Number: 544
E-mail: mail
Doctor ravzanur tok
-----------------------------------------
Press 0 to see menu...
```

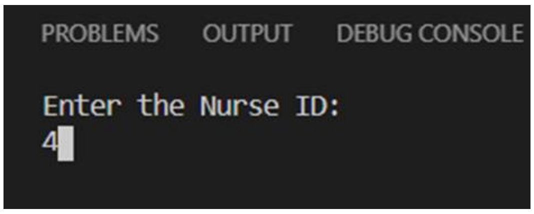
Nurse 2 Information



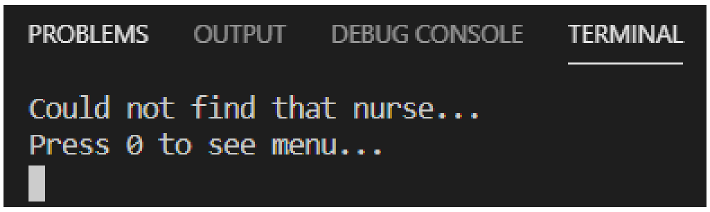
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

-----------------------------------------
ID: 9
Name: busra Surname: yakan
Gender: f
Age: 21
Address: giresun
Telephone Number: 523
E-mail: mail
Assistant Doctor ezgi sune
-----------------------------------------
Press 0 to see menu...
```

Nurse 3 Information

If there is no nurse in the entered id, the below message is printed.

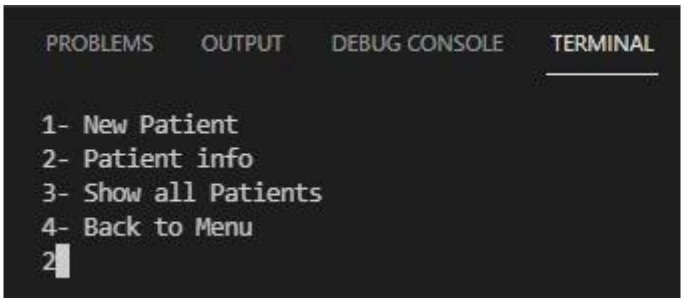


```
PROBLEMS    OUTPUT    DEBUG CONSOLE

Enter the Nurse ID:
4
```

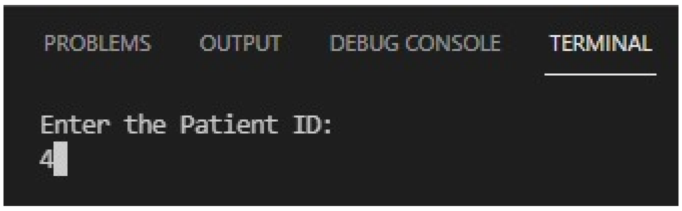


```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Could not find that nurse...
Press 0 to see menu...
```

Patient info menu: press *2*



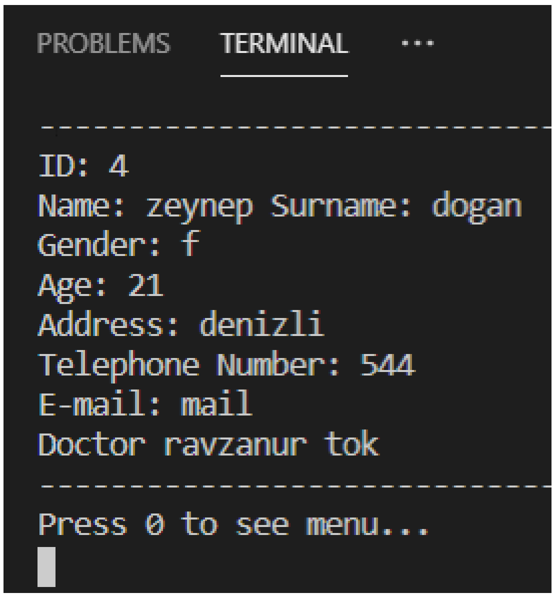
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

1- New Patient
2- Patient info
3- Show all Patients
4- Back to Menu
2
```



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Enter the Patient ID:
4
```

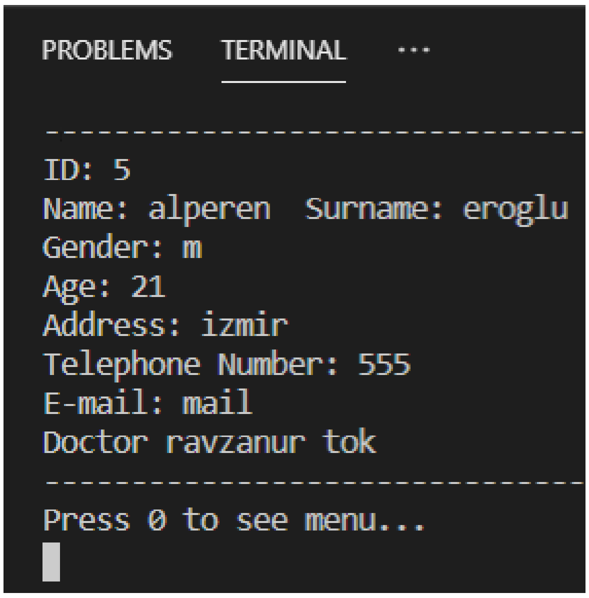
(is also done for patient 4 and 5)

The relevant doctor of the patient can also be seen in the information.



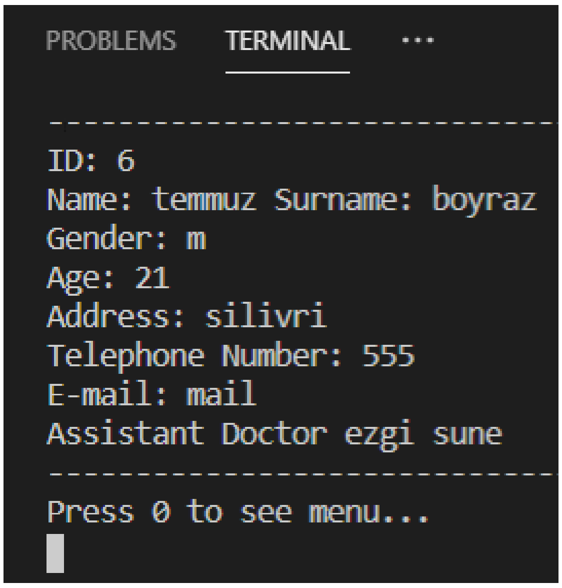
```
------------------------------
ID: 4
Name: zeynep Surname: dogan
Gender: f
Age: 21
Address: denizli
Telephone Number: 544
E-mail: mail
Doctor ravzanur tok
------------------------------
Press 0 to see menu...
```

Patient 1 Information



```
------------------------------
ID: 5
Name: alperen  Surname: eroglu
Gender: m
Age: 21
Address: izmir
Telephone Number: 555
E-mail: mail
Doctor ravzanur tok
------------------------------
Press 0 to see menu...
```

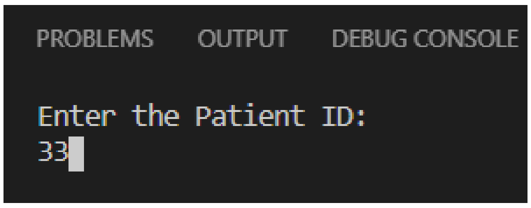
Patient 2 Information



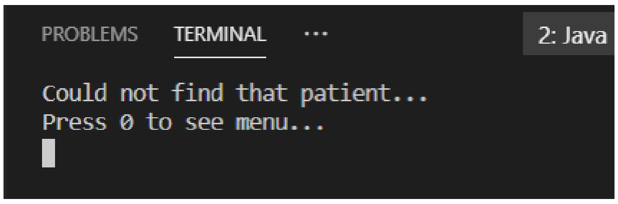
```
------------------------------
ID: 6
Name: temmuz Surname: boyraz
Gender: m
Age: 21
Address: silivri
Telephone Number: 555
E-mail: mail
Assistant Doctor ezgi sune
------------------------------
Press 0 to see menu...
```

Patient 3 Information

If there is no patient in the entered id, the below message is printed.



```
Enter the Patient ID:
33
```

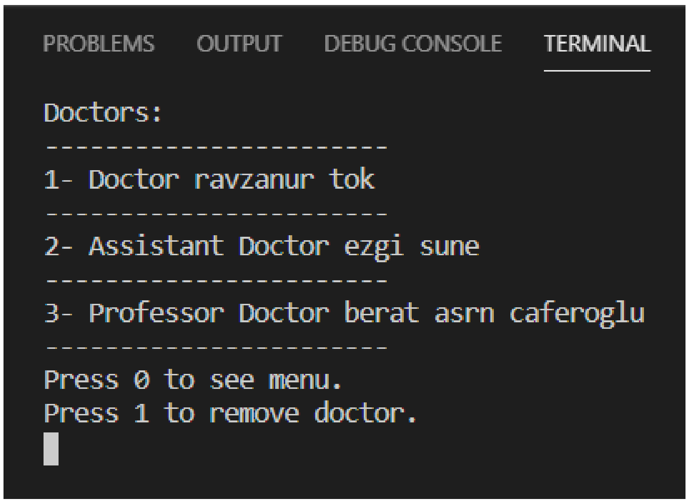


```
Could not find that patient...
Press 0 to see menu...
```

Show all **submenus in 2(Doctors)-3(Nurses)-4(Patients) menus**:

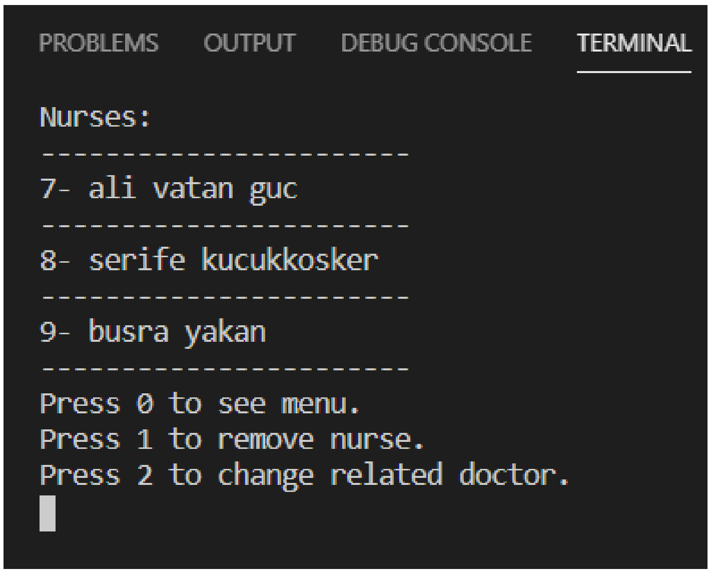
(It doesn't matter in what order the ids are entered, sorting is done on all arrays, we'll discuss this later.)

Show all doctors menu: press *3*



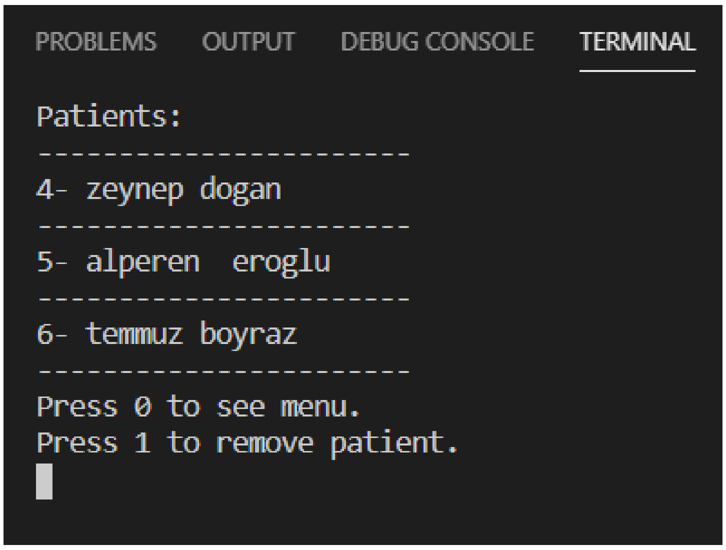
```
Doctors:
-----------------------
1- Doctor ravzanur tok
-----------------------
2- Assistant Doctor ezgi sune
-----------------------
3- Professor Doctor berat asrn caferoglu
-----------------------
Press 0 to see menu.
Press 1 to remove doctor.
```

Show all nurses menu: press *3*

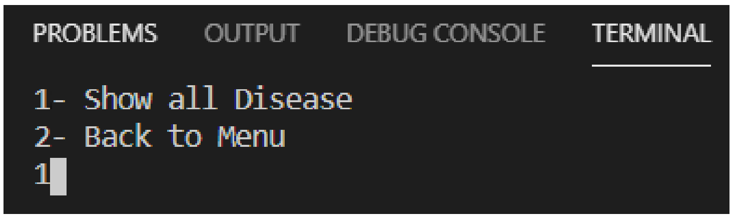


```
Nurses:
-----------------------
7- ali vatan guc
-----------------------
8- serife kucukkosker
-----------------------
9- busra yakan
-----------------------
Press 0 to see menu.
Press 1 to remove nurse.
Press 2 to change related doctor.
```

Show all patients menu: press *3*



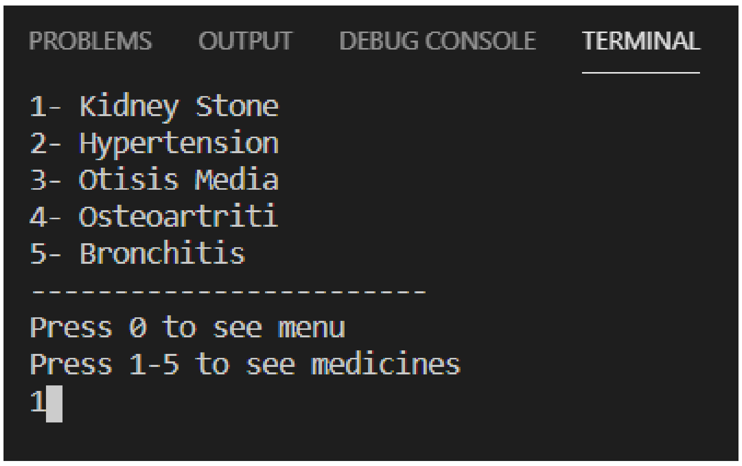
```
Patients:
-----------------------
4- zeynep dogan
-----------------------
5- alperen  eroglu
-----------------------
6- temmuz boyraz
-----------------------
Press 0 to see menu.
Press 1 to remove patient.
```



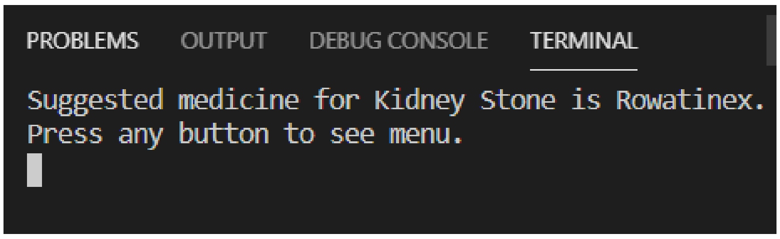
```
1- Show all Disease
2- Back to Menu
1
```

5-) Diseases Menu

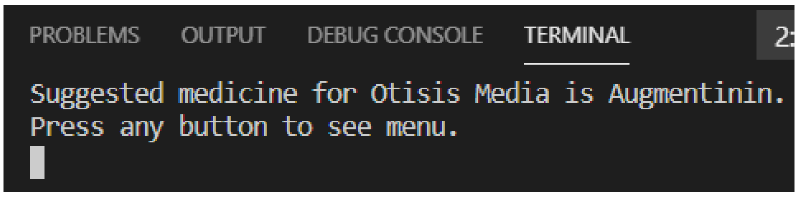
5 different disease examples were shown for 5 different departments in the system, and the recommended medicines for each disease can also be seen in the output.



```
1- Kidney Stone
2- Hypertension
3- Otisis Media
4- Osteoartriti
5- Bronchitis
------------------------
Press 0 to see menu
Press 1-5 to see medicines
1
```

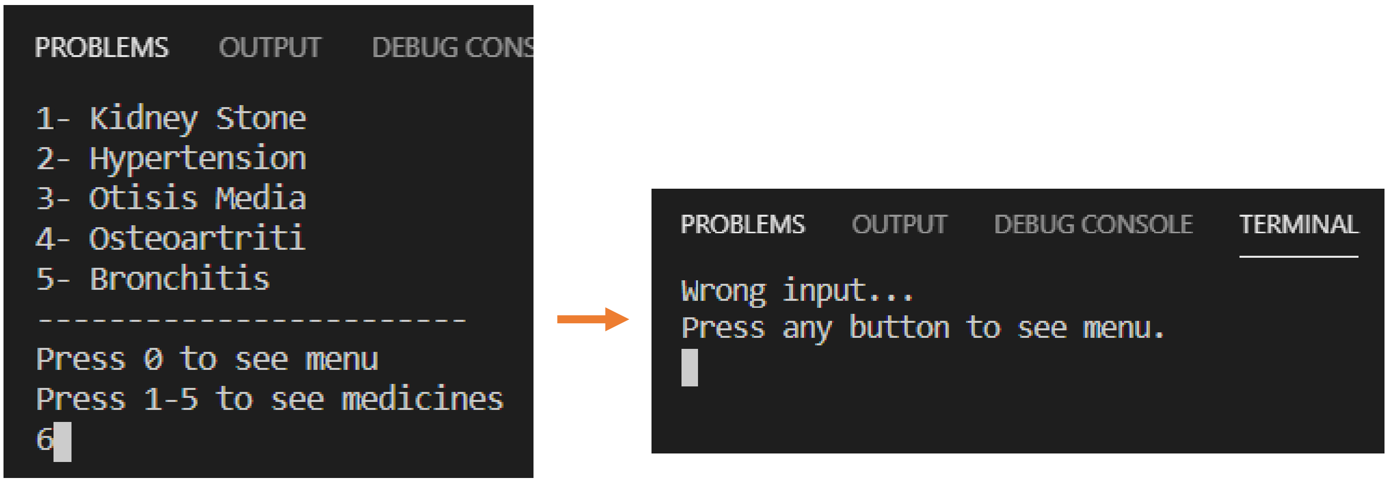


```
Suggested medicine for Kidney Stone is Rowatinex.
Press any button to see menu.
```



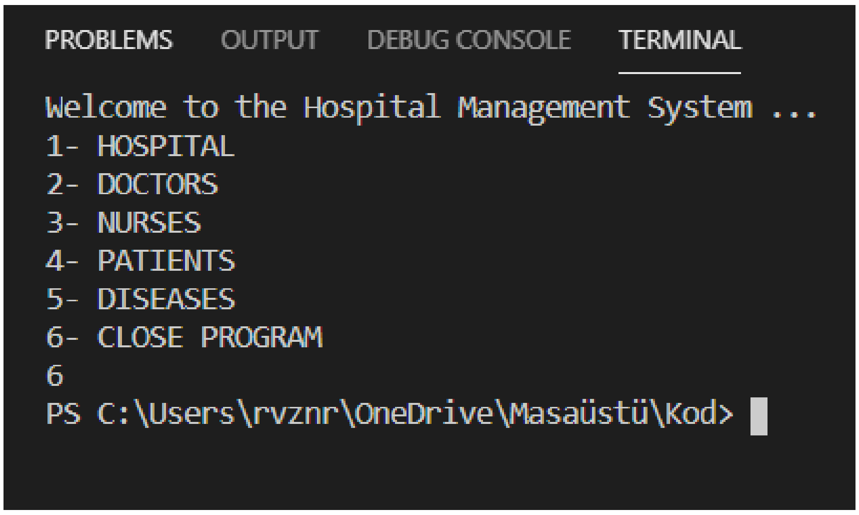
```
Suggested medicine for Otisis Media is Augmentinin.
Press any button to see menu.
```

If a disease that is not in the system is selected, the below message is printed.



```
1- Kidney Stone
2- Hypertension
3- Otisis Media
4- Osteoartriti
5- Bronchitis
------------------------
Press 0 to see menu
Press 1-5 to see medicines
6
```

```
Wrong input...
Press any button to see menu.
```



```
Welcome to the Hospital Management System ...
1- HOSPITAL
2- DOCTORS
3- NURSES
4- PATIENTS
5- DISEASES
6- CLOSE PROGRAM
6
PS C:\Users\rvznr\OneDrive\Masaüstü\Kod>
```

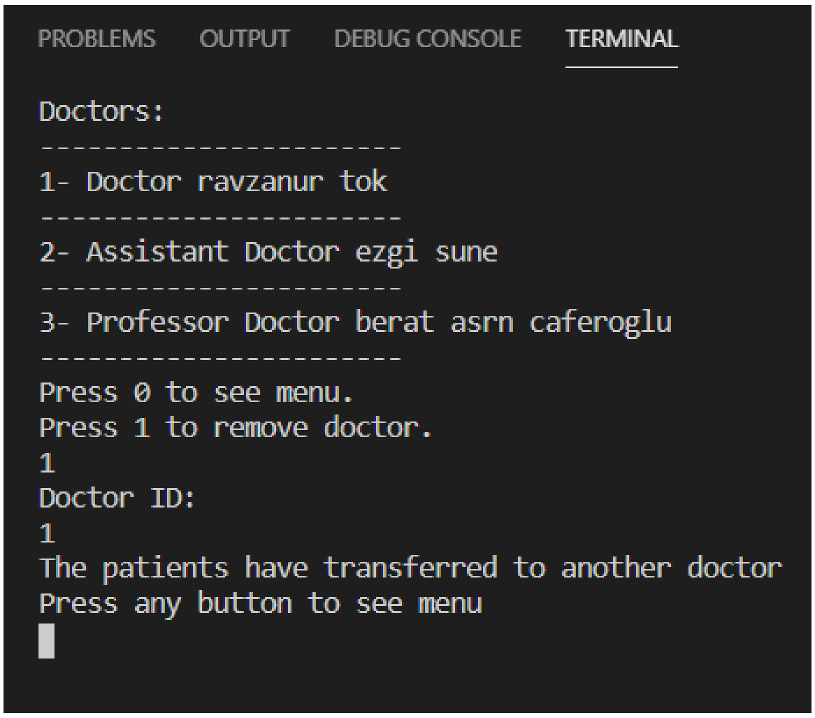
Close Program Menu

The menus in the management system are fully explained.

The operation of the system will now be examined more closely with scenarios.

- <u>What happens to patients and nurses of a doctor removed from the system?</u>

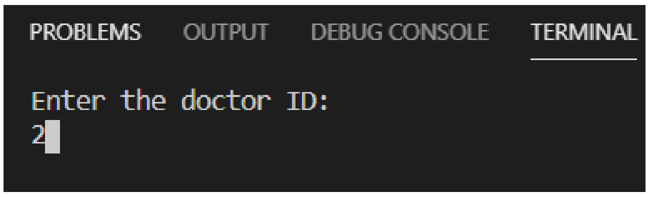
Remove doctor1 (has patients 4 and 5, nurses 7 and 8):



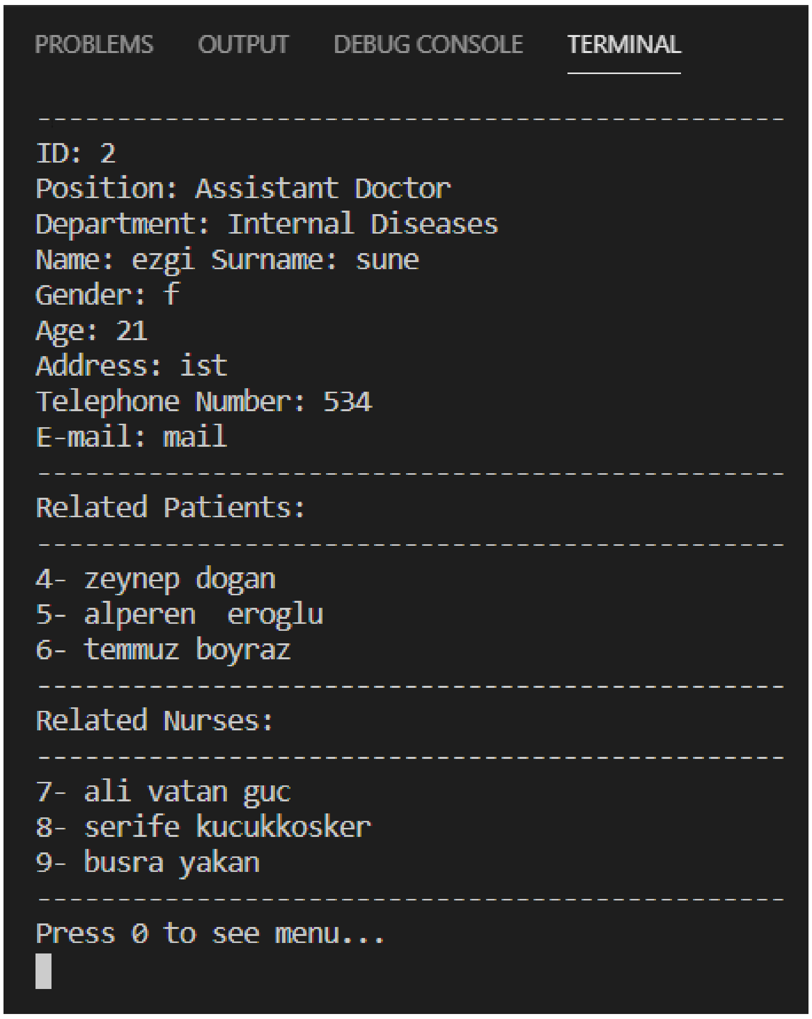
```
Doctors:
-----------------------
1- Doctor ravzanur tok
-----------------------
2- Assistant Doctor ezgi sune
-----------------------
3- Professor Doctor berat asrn caferoglu
-----------------------
Press 0 to see menu.
Press 1 to remove doctor.
1
Doctor ID:
1
The patients have transferred to another doctor
Press any button to see menu
```

Patients were appointed to another doctor in the same department, and so did nurses.

Now let's take a look at doctor 2 (ezgi) in the same department.



```
Enter the doctor ID:
2
```

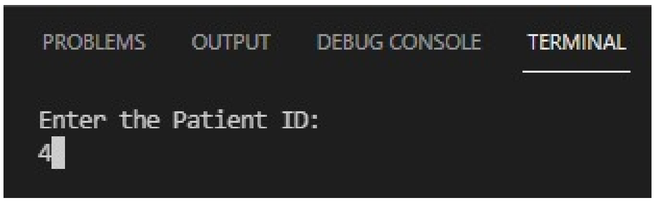


```
-----------------------------------------------
ID: 2
Position: Assistant Doctor
Department: Internal Diseases
Name: ezgi Surname: sune
Gender: f
Age: 21
Address: ist
Telephone Number: 534
E-mail: mail
-----------------------------------------------
Related Patients:
-----------------------------------------------
4- zeynep dogan
5- alperen  eroglu
6- temmuz boyraz
-----------------------------------------------
Related Nurses:
-----------------------------------------------
7- ali vatan guc
8- serife kucukkosker
9- busra yakan
-----------------------------------------------
Press 0 to see menu...
```

As can be seen, the patients (4,5) and nurses(7,8) of doctor 1(ravzanur) were transferred to doctor 2(ezgi).

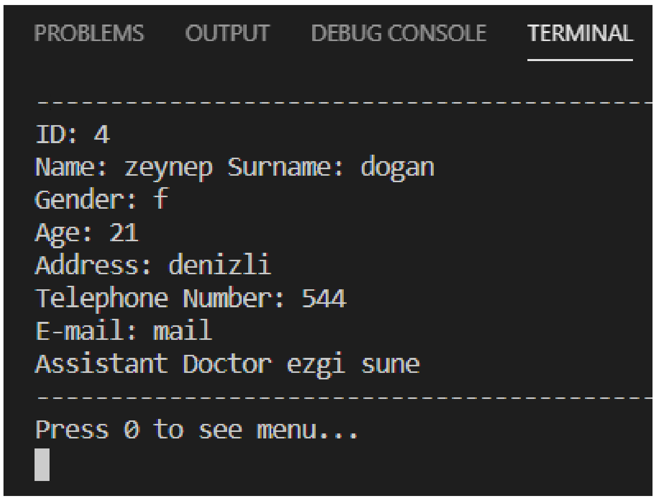
Likewise, let's check if the doctor has changed in patients and nurses.

Patient 4 and 5;

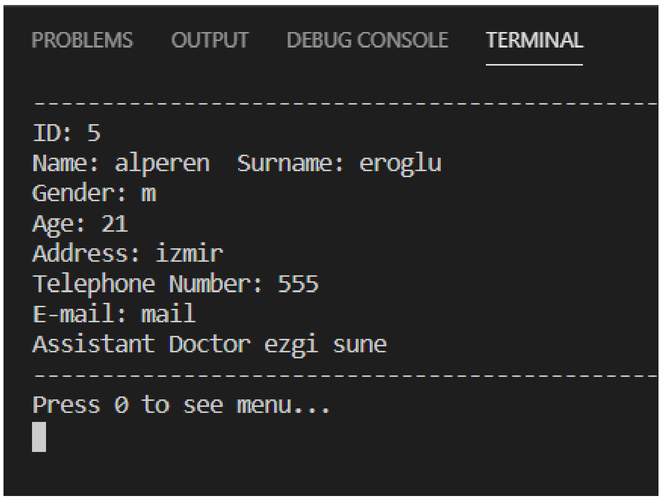


```
Enter the Patient ID:
4
```

(is also done for id 5)



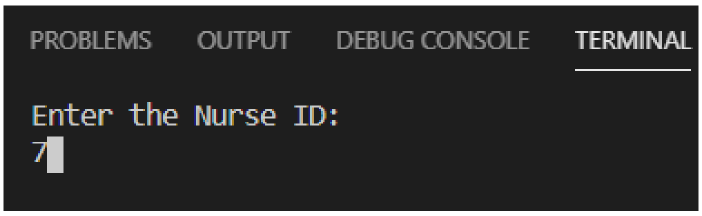
```
----------------------------------------
ID: 4
Name: zeynep Surname: dogan
Gender: f
Age: 21
Address: denizli
Telephone Number: 544
E-mail: mail
Assistant Doctor ezgi sune
----------------------------------------
Press 0 to see menu...
```



```
----------------------------------------
ID: 5
Name: alperen  Surname: eroglu
Gender: m
Age: 21
Address: izmir
Telephone Number: 555
E-mail: mail
Assistant Doctor ezgi sune
----------------------------------------
Press 0 to see menu...
```

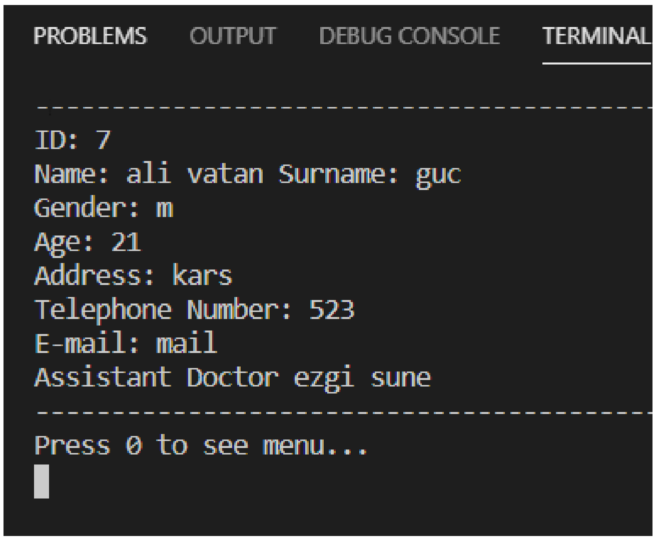
Their doctor is doctor (ezgi) as expected.

Nurses 7 and 8;

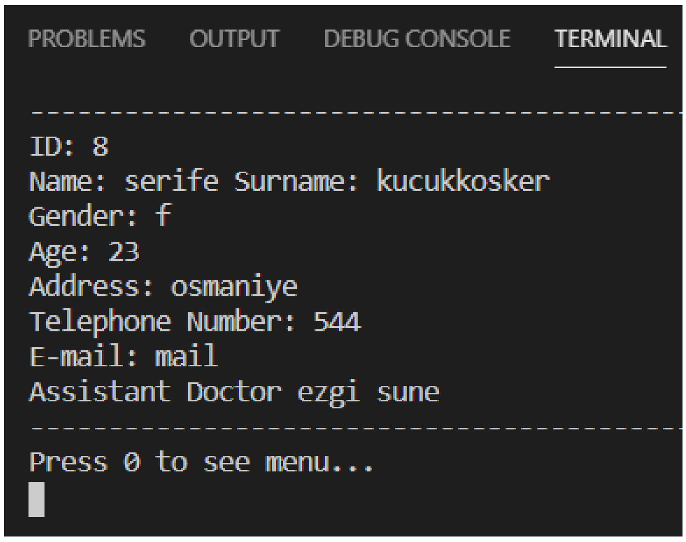


```
Enter the Nurse ID:
7
```

(is also done for id 8)



```
----------------------------------------
ID: 7
Name: ali vatan Surname: guc
Gender: m
Age: 21
Address: kars
Telephone Number: 523
E-mail: mail
Assistant Doctor ezgi sune
----------------------------------------
Press 0 to see menu...
```

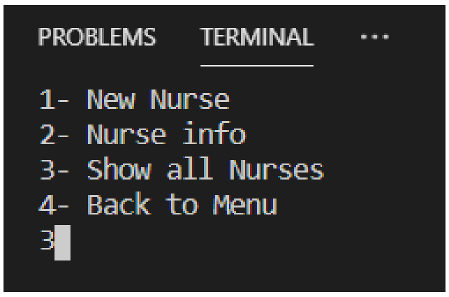


```
----------------------------------------
ID: 8
Name: serife Surname: kucukkosker
Gender: f
Age: 23
Address: osmaniye
Telephone Number: 544
E-mail: mail
Assistant Doctor ezgi sune
----------------------------------------
Press 0 to see menu...
```

Their doctor is doctor 2 (ezgi) as expected.

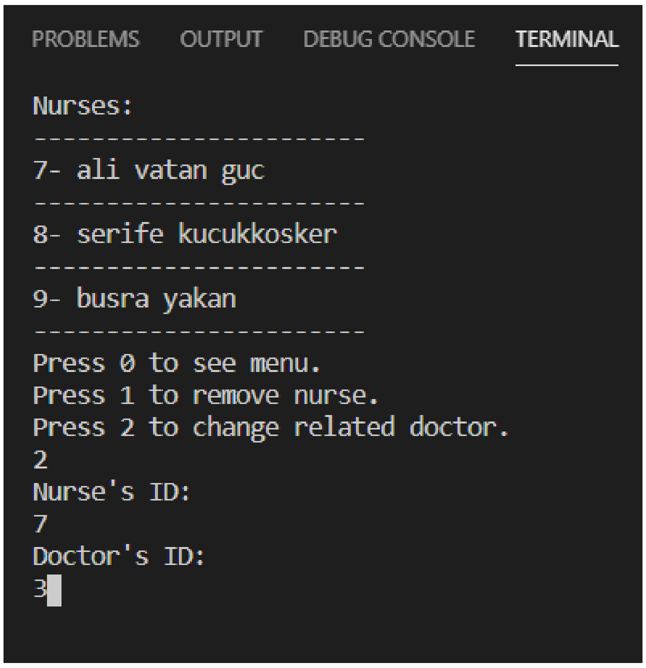
- What happens in the system if the nurse is appointed to another doctor or deleted?

Reached the nurses list from show all nurses.



```
1- New Nurse
2- Nurse info
3- Show all Nurses
4- Back to Menu
3
```

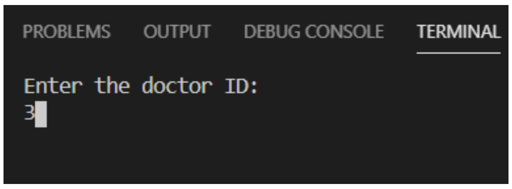
Change the related doctor; press 2



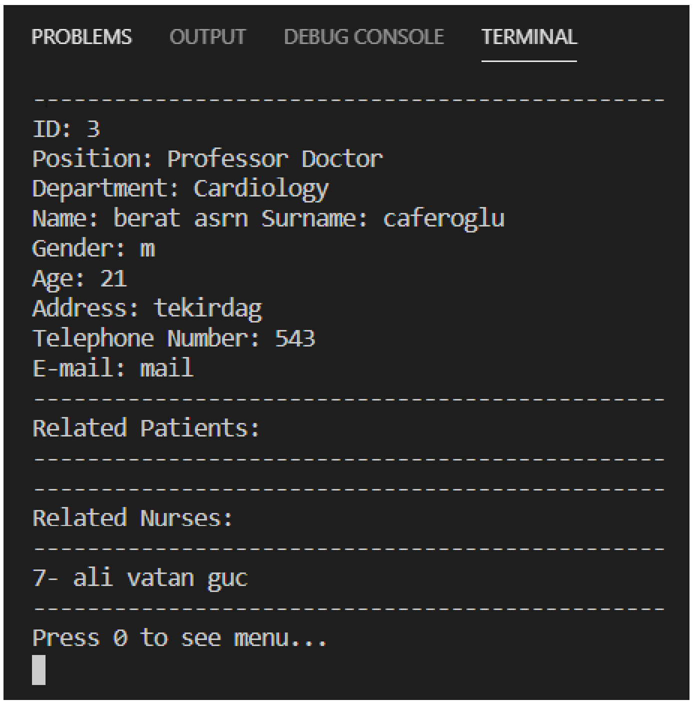
```
Nurses:
-----------------------
7- ali vatan guc
-----------------------
8- serife kucukkosker
-----------------------
9- busra yakan
-----------------------
Press 0 to see menu.
Press 1 to remove nurse.
Press 2 to change related doctor.
2
Nurse's ID:
7
Doctor's ID:
3
```

Nurse 7 is transferred to doctor 3.

Check the doctor 3;



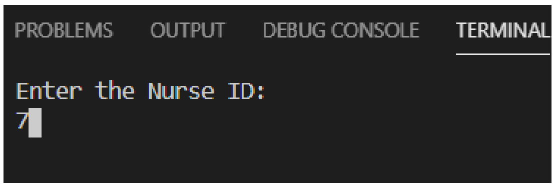
```
Enter the doctor ID:
3
```



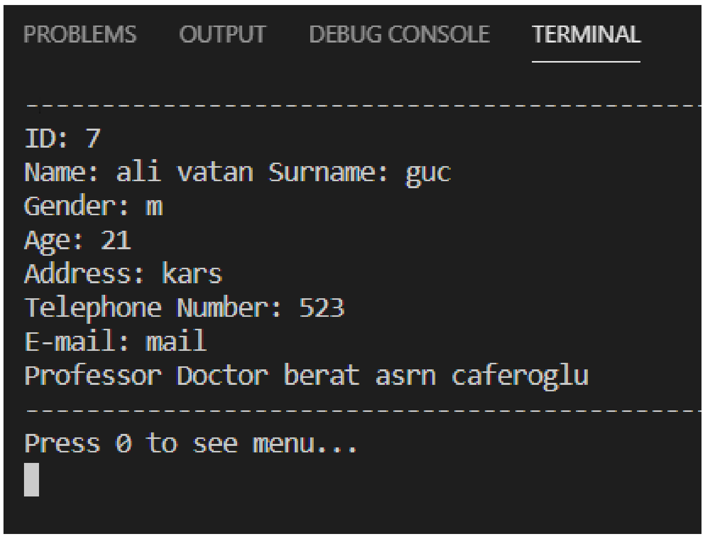
```
-----------------------------------------------
ID: 3
Position: Professor Doctor
Department: Cardiology
Name: berat asrn Surname: caferoglu
Gender: m
Age: 21
Address: tekirdag
Telephone Number: 543
E-mail: mail
-----------------------------------------------
Related Patients:
-----------------------------------------------
-----------------------------------------------
Related Nurses:
-----------------------------------------------
7- ali vatan guc
-----------------------------------------------
Press 0 to see menu...
```

Nurse 7(ali) is attached as expected.

Likewise, check the nurse 7;



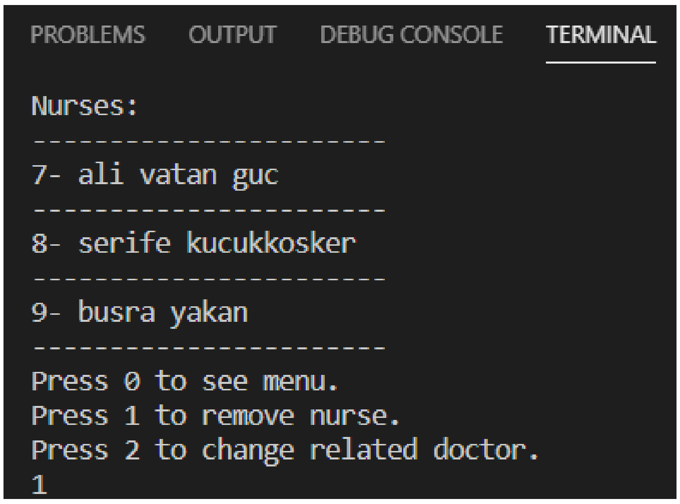
```
Enter the Nurse ID:
7
```



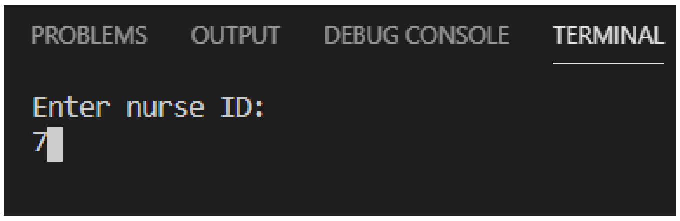
```
-------------------------------------------
ID: 7
Name: ali vatan Surname: guc
Gender: m
Age: 21
Address: kars
Telephone Number: 523
E-mail: mail
Professor Doctor berat asrn caferoglu
-------------------------------------------
Press 0 to see menu...
```

Doctor 3(berat) is attached as expected.

Now let's delete nurse 7; press 1



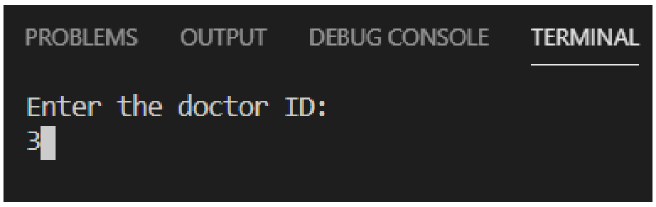
```
Nurses:
-----------------------
7- ali vatan guc
-----------------------
8- serife kucukkosker
-----------------------
9- busra yakan
-----------------------
Press 0 to see menu.
Press 1 to remove nurse.
Press 2 to change related doctor.
1
```



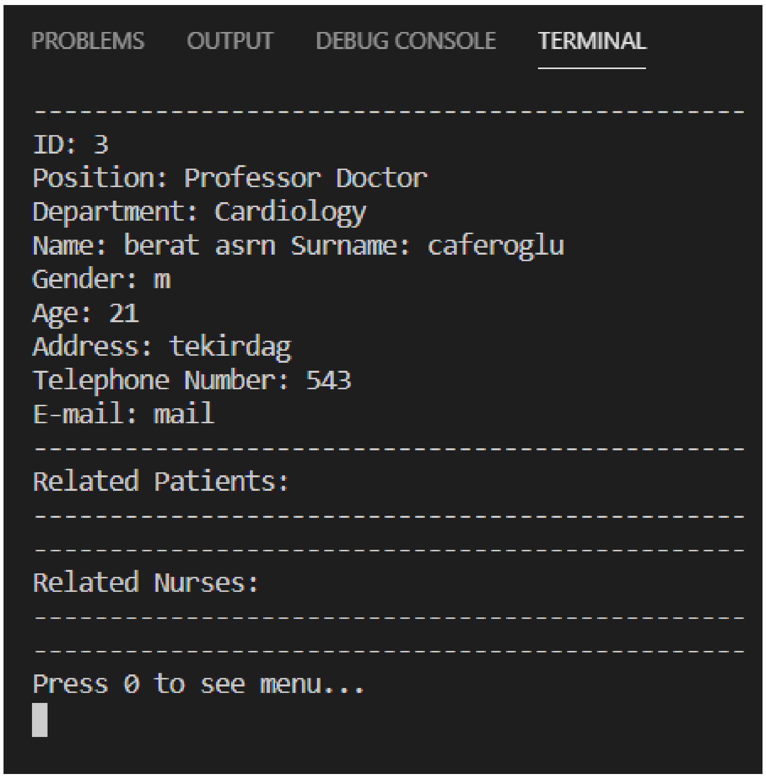
```
Enter nurse ID:
7
```

Nurse 7 is removed.

Check the doctor 3(berat) ;



```
Enter the doctor ID:
3
```

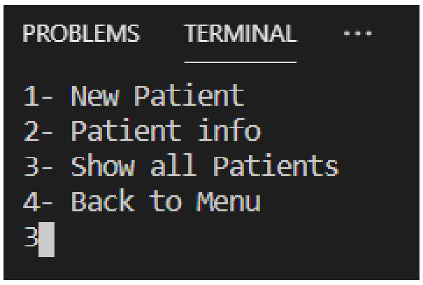


```
----------------------------------------------
ID: 3
Position: Professor Doctor
Department: Cardiology
Name: berat asrn Surname: caferoglu
Gender: m
Age: 21
Address: tekirdag
Telephone Number: 543
E-mail: mail
----------------------------------------------
Related Patients:
----------------------------------------------
----------------------------------------------
Related Nurses:
----------------------------------------------
----------------------------------------------
Press 0 to see menu...
```

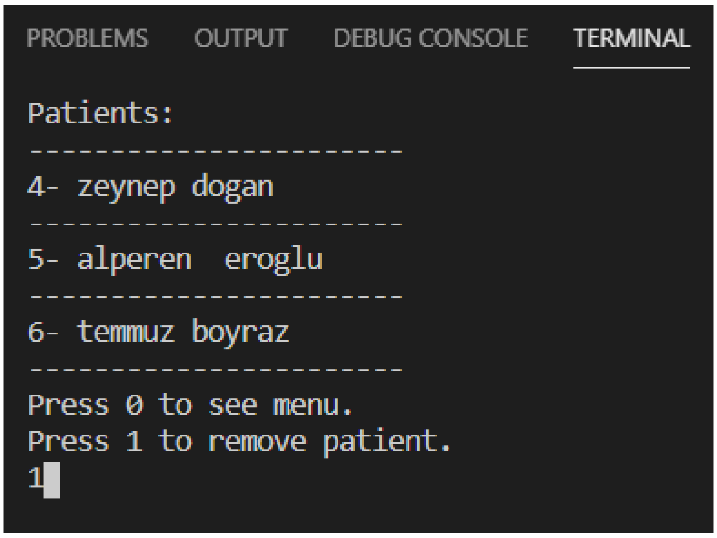
Nurse 7 (ali) is deleted.

- <u>What happens if the patient is deleted from the system?</u>

Remove patient 4 in show all patients.



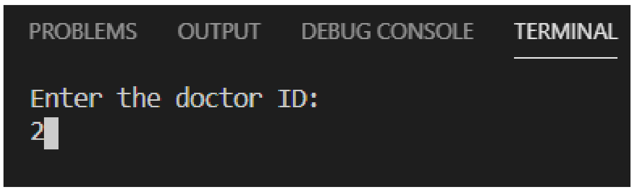
```
1- New Patient
2- Patient info
3- Show all Patients
4- Back to Menu
3
```



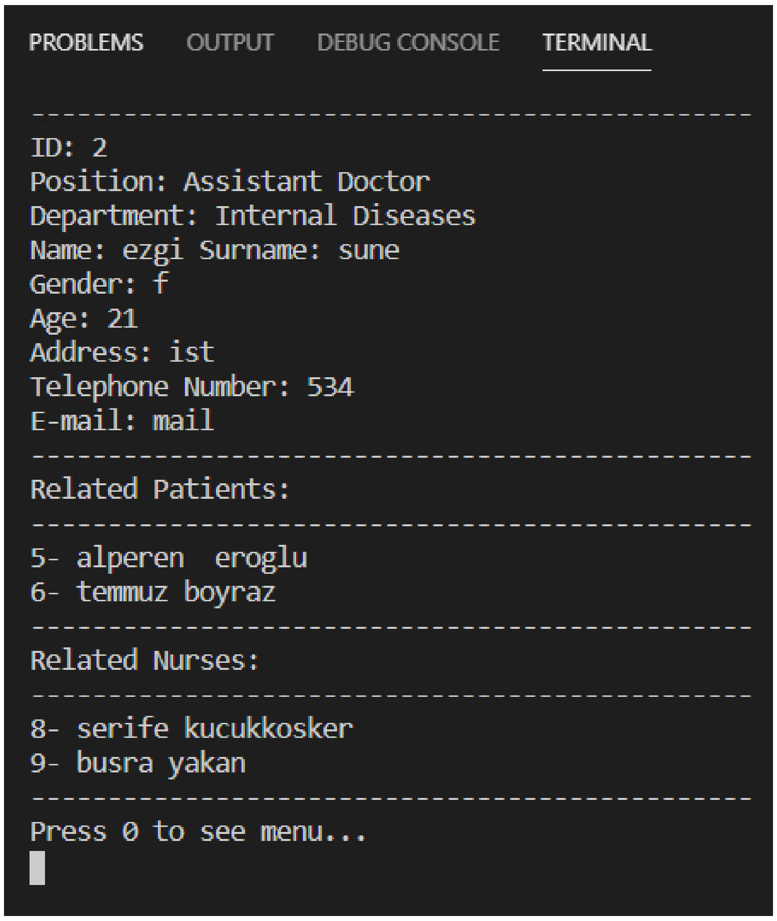
```
Patients:
------------------------
4- zeynep dogan
------------------------
5- alperen  eroglu
------------------------
6- temmuz boyraz
------------------------
Press 0 to see menu.
Press 1 to remove patient.
1
```

Patient 4 (zeynep) is removed.

Look at the doctor 2 to which patient 4 is related.



```
Enter the doctor ID:
2
```

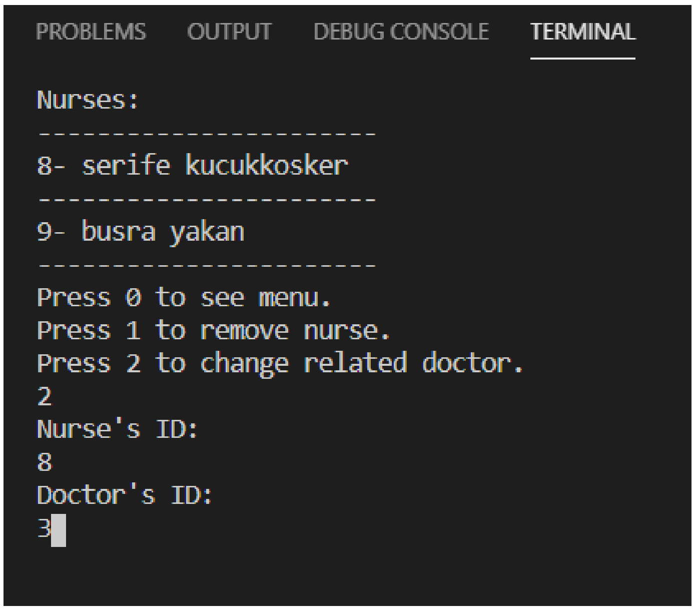


```
-----------------------------------------------
ID: 2
Position: Assistant Doctor
Department: Internal Diseases
Name: ezgi Surname: sune
Gender: f
Age: 21
Address: ist
Telephone Number: 534
E-mail: mail
-----------------------------------------------
Related Patients:
-----------------------------------------------
5- alperen  eroglu
6- temmuz boyraz
-----------------------------------------------
Related Nurses:
-----------------------------------------------
8- serife kucukkosker
9- busra yakan
-----------------------------------------------
Press 0 to see menu...
```

Patient 4 (zeynep) is now removed at doctor 2 as expected.

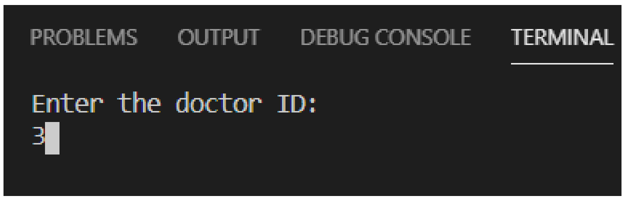
- Let's try another combination on nurse transfer.

Transfer nurse 8 (şerife) to doctor 3 (berat) from show all nurse.

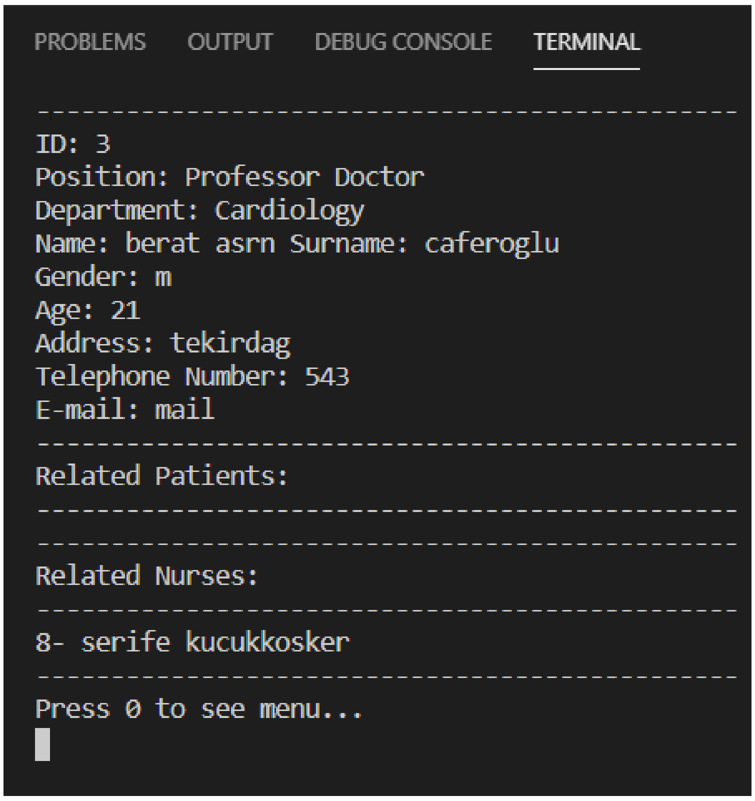


```
Nurses:
----------------------
8- serife kucukkosker
----------------------
9- busra yakan
----------------------
Press 0 to see menu.
Press 1 to remove nurse.
Press 2 to change related doctor.
2
Nurse's ID:
8
Doctor's ID:
3
```

Check the doctor 3 (berat);



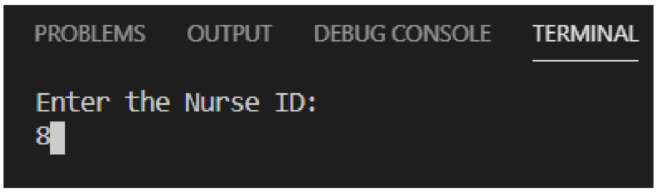
```
Enter the doctor ID:
3
```



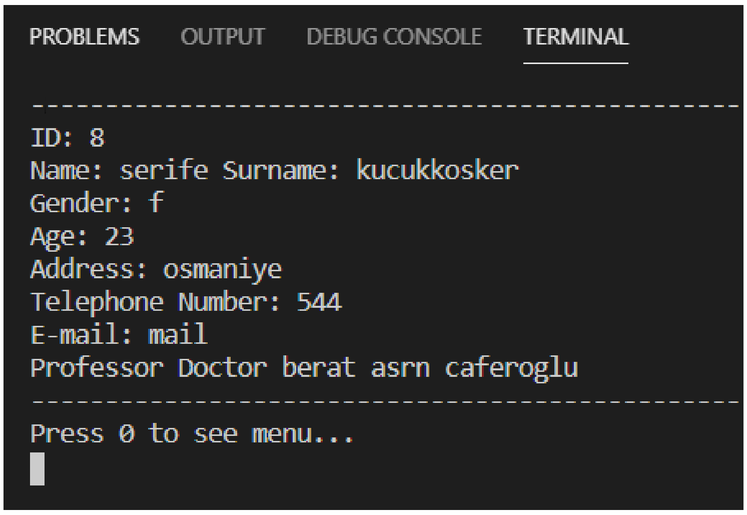
```
-----------------------------------------------
ID: 3
Position: Professor Doctor
Department: Cardiology
Name: berat asrn Surname: caferoglu
Gender: m
Age: 21
Address: tekirdag
Telephone Number: 543
E-mail: mail
-----------------------------------------------
Related Patients:
-----------------------------------------------
-----------------------------------------------
Related Nurses:
-----------------------------------------------
8- serife kucukkosker
-----------------------------------------------
Press 0 to see menu...
```

Nurse 8 (şerife) is attached as expected.

Check the nurse 8;



```
Enter the Nurse ID:
8
```

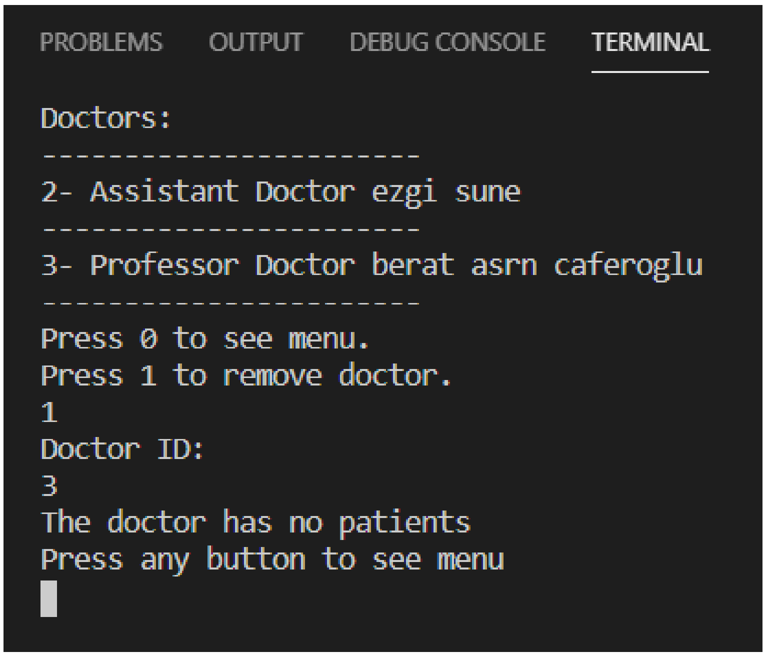


```
-----------------------------------------------
ID: 8
Name: serife Surname: kucukkosker
Gender: f
Age: 23
Address: osmaniye
Telephone Number: 544
E-mail: mail
Professor Doctor berat asrn caferoglu
-----------------------------------------------
Press 0 to see menu...
```

Also doctor 3 (berat) is attached.

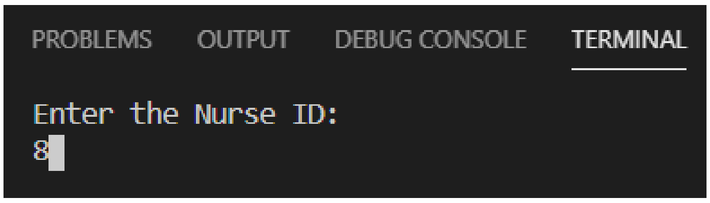
- Let's see some more outputs about system operation.

When doctor 3 (berat) is deleted, we see that he has no patients.

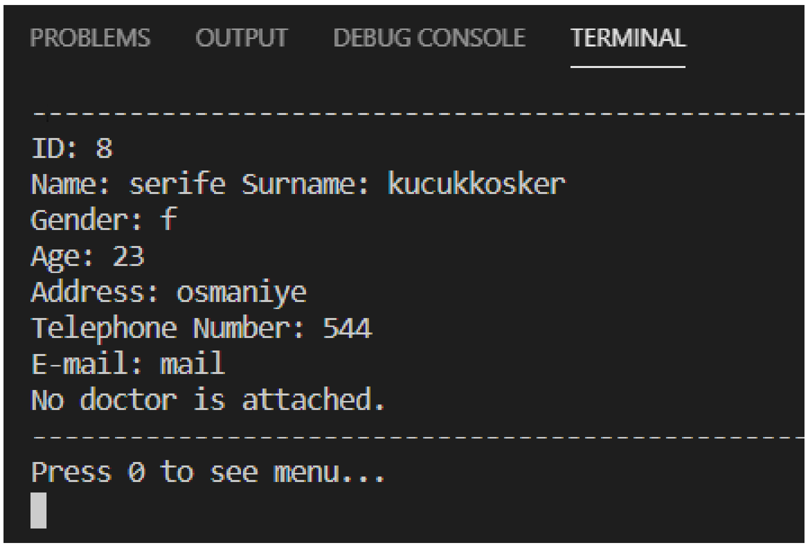


```
Doctors:
-----------------------
2- Assistant Doctor ezgi sune
-----------------------
3- Professor Doctor berat asrn caferoglu
-----------------------
Press 0 to see menu.
Press 1 to remove doctor.
1
Doctor ID:
3
The doctor has no patients
Press any button to see menu
```

But what if he has a related nurse(8) ?



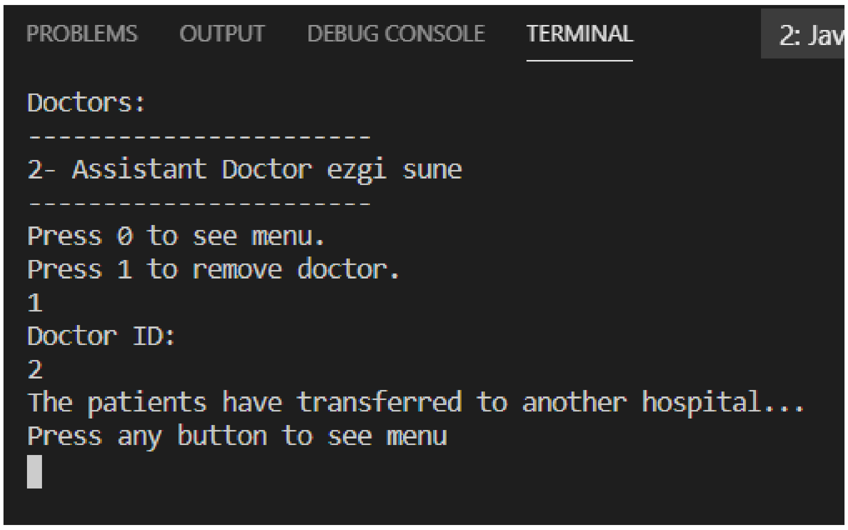
```
Enter the Nurse ID:
8
```



```
-----------------------------------------------
ID: 8
Name: serife Surname: kucukkosker
Gender: f
Age: 23
Address: osmaniye
Telephone Number: 544
E-mail: mail
No doctor is attached.
-----------------------------------------------
Press 0 to see menu...
```

Nurse 8(şerife) was kept in the open because there is no other doctors from the same department.

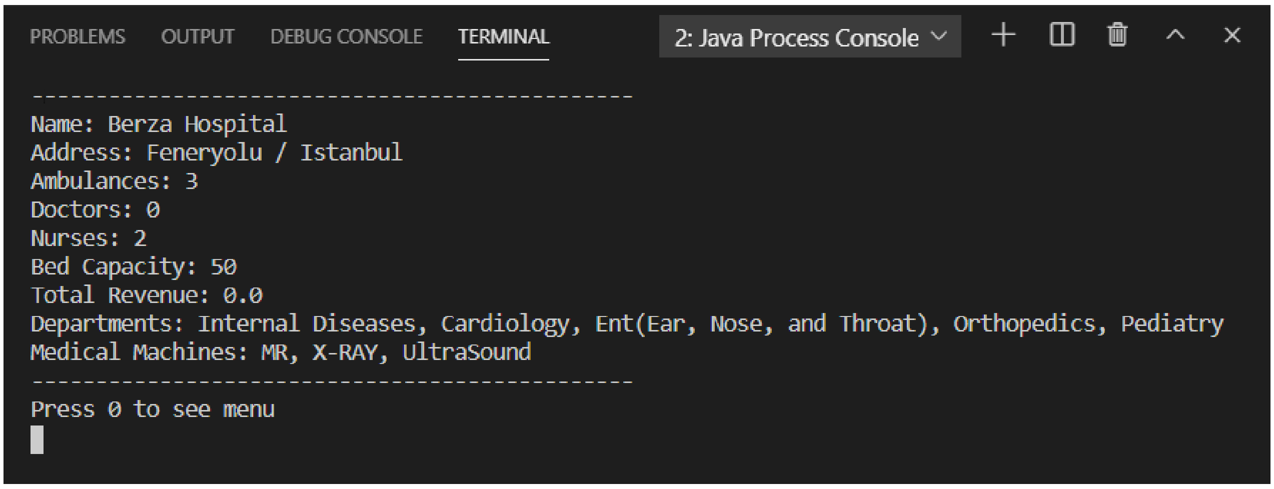
If we remove the last doctor (2) we have;



```
Doctors:
-----------------------
2- Assistant Doctor ezgi sune
-----------------------
Press 0 to see menu.
Press 1 to remove doctor.
1
Doctor ID:
2
The patients have transferred to another hospital...
Press any button to see menu
```

Since there is no doctor in the hospital, patients are asked to be transferred to another hospital.

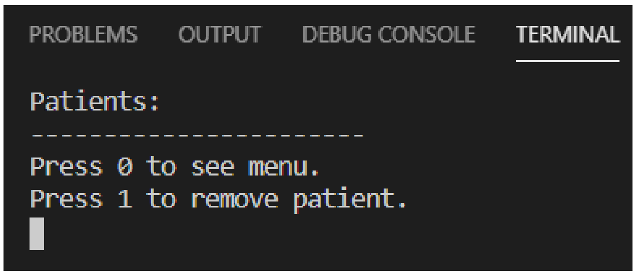
Check the hospital info;



```
----------------------------------------------
Name: Berza Hospital
Address: Feneryolu / Istanbul
Ambulances: 3
Doctors: 0
Nurses: 2
Bed Capacity: 50
Total Revenue: 0.0
Departments: Internal Diseases, Cardiology, Ent(Ear, Nose, and Throat), Orthopedics, Pediatry
Medical Machines: MR, X-RAY, UltraSound
----------------------------------------------
Press 0 to see menu
```

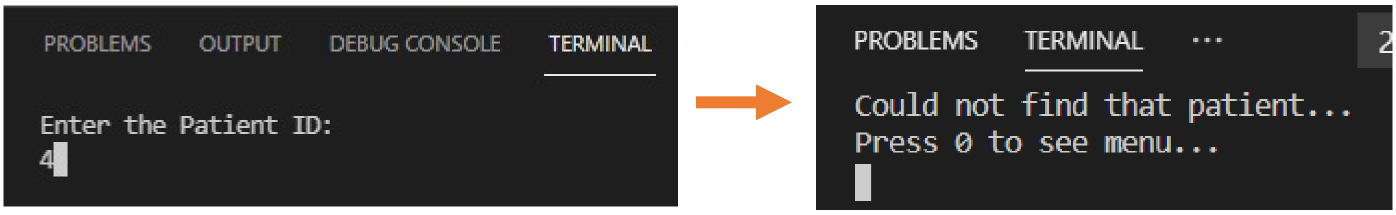
There is no doctor, no patient so total revenue is 0

Empty patient list ;



```
Patients:
-----------------------
Press 0 to see menu.
Press 1 to remove patient.
```

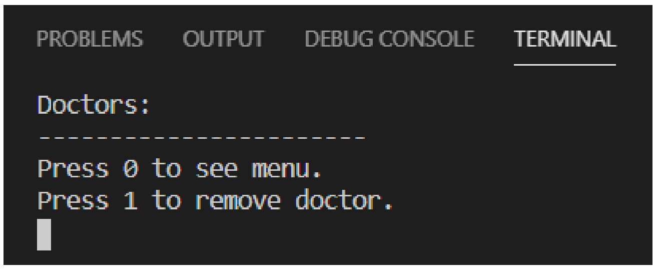
Let's search for info on the empty patient list.



```
Enter the Patient ID:
4
```

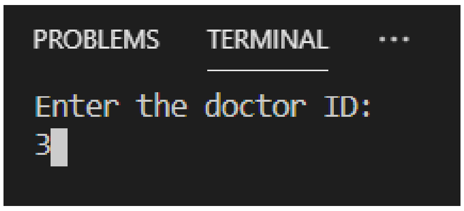
```
Could not find that patient...
Press 0 to see menu...
```

Empty doctor list ;

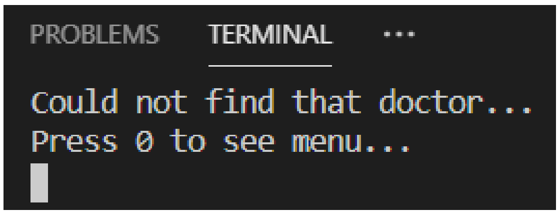


```
Doctors:
------------------------
Press 0 to see menu.
Press 1 to remove doctor.
```

Let's search for info on the empty doctor list.

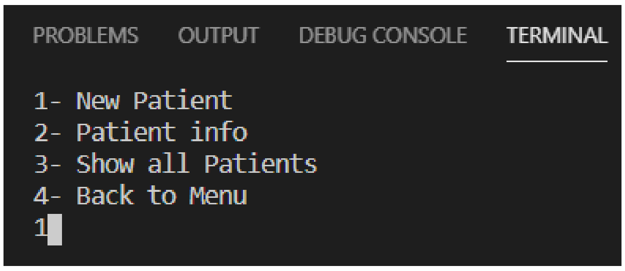


```
Enter the doctor ID:
3
```

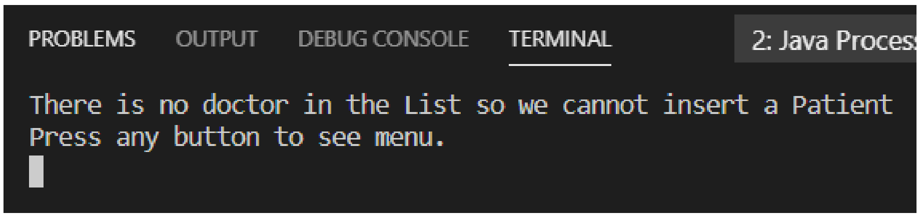


```
Could not find that doctor...
Press 0 to see menu...
```

Let's try to add patient when doctors are away.



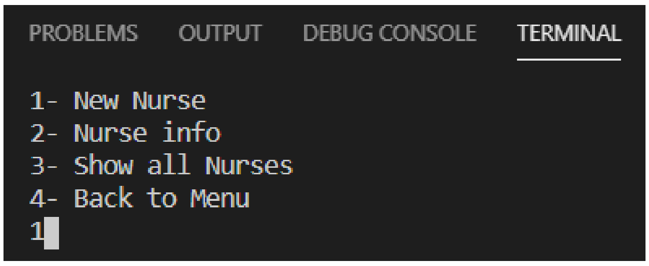
```
1- New Patient
2- Patient info
3- Show all Patients
4- Back to Menu
1
```



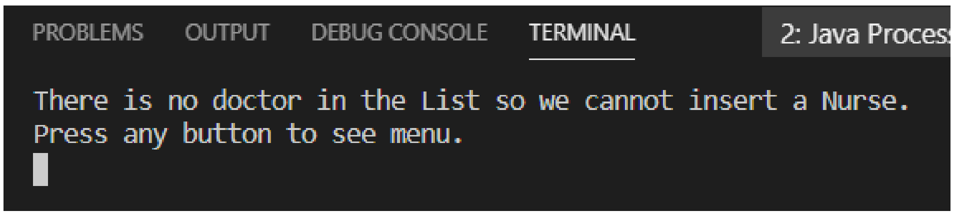
```
There is no doctor in the List so we cannot insert a Patient
Press any button to see menu.
```

We cannot add patients without a doctor.

Let's try to add nurse when doctors are away.



```
1- New Nurse
2- Nurse info
3- Show all Nurses
4- Back to Menu
1
```

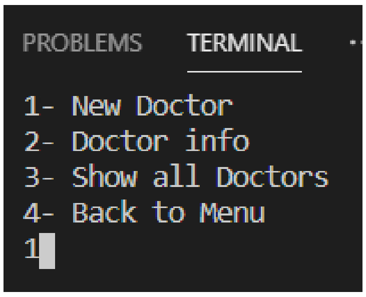


```
There is no doctor in the List so we cannot insert a Nurse.
Press any button to see menu.
```

Similarly, we cannot add a nurse without a doctor.

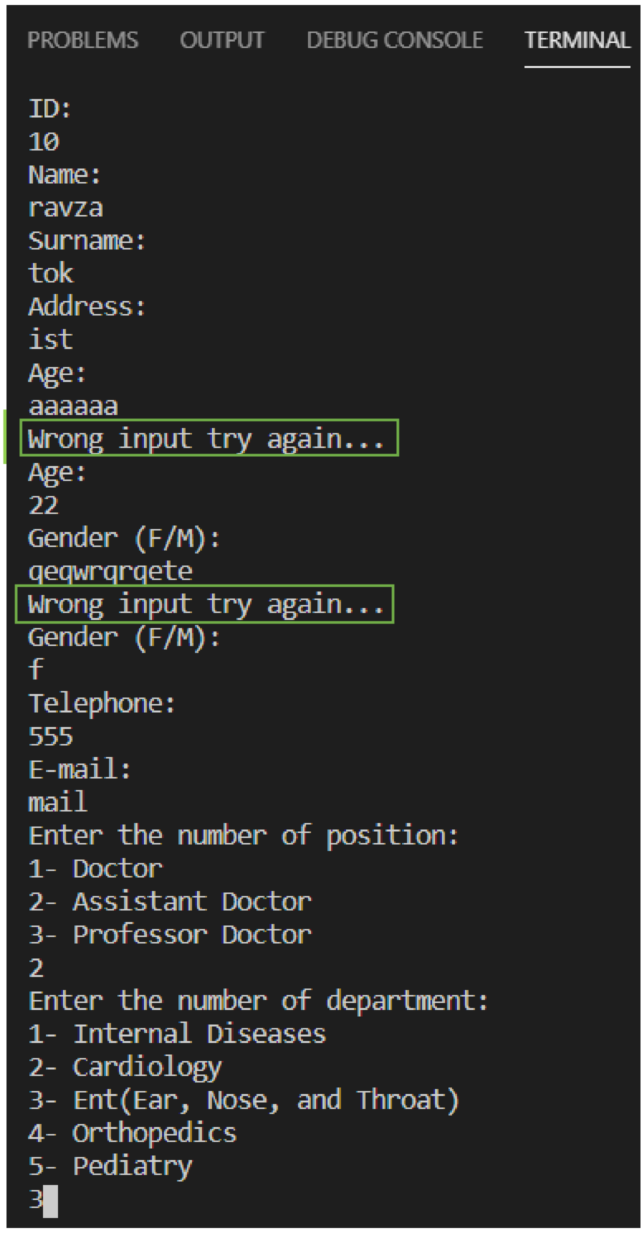
- Let’s see the id control and some wrong entering messages.

Add new doctor;



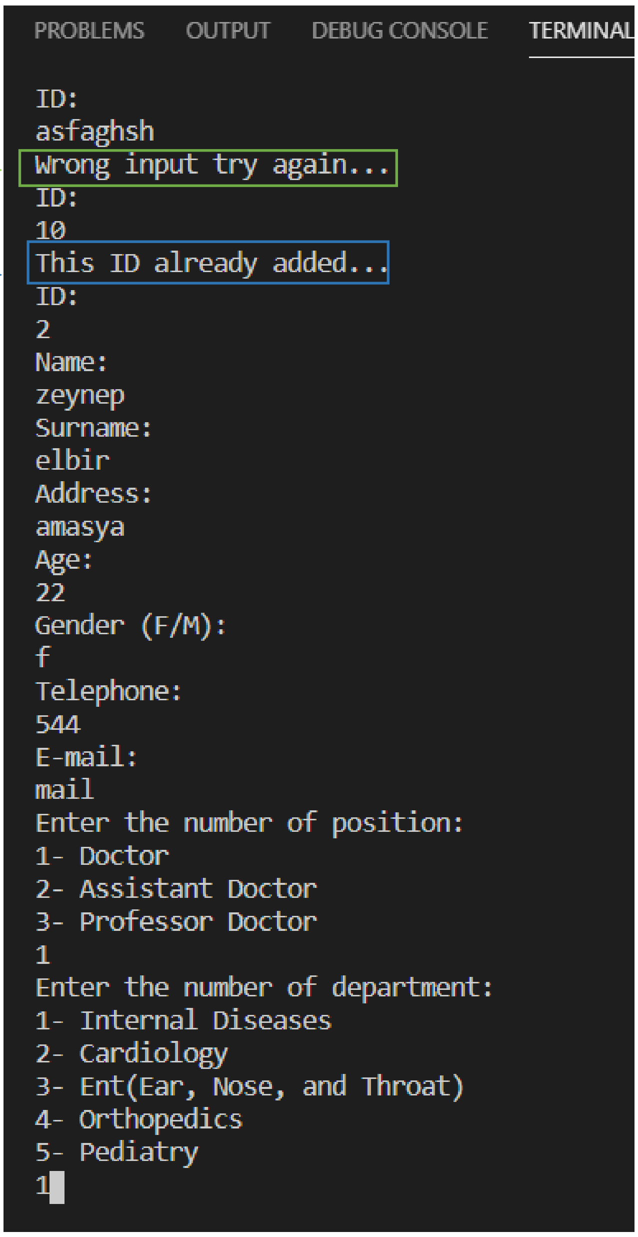
```
1- New Doctor
2- Doctor info
3- Show all Doctors
4- Back to Menu
1
```

Doctor is added with id 10.



```
ID:
10
Name:
ravza
Surname:
tok
Address:
ist
Age:
aaaaaa
Wrong input try again...
Age:
22
Gender (F/M):
qeqwrqrqete
Wrong input try again...
Gender (F/M):
f
Telephone:
555
E-mail:
mail
Enter the number of position:
1- Doctor
2- Assistant Doctor
3- Professor Doctor
2
Enter the number of department:
1- Internal Diseases
2- Cardiology
3- Ent(Ear, Nose, and Throat)
4- Orthopedics
5- Pediatry
3
```

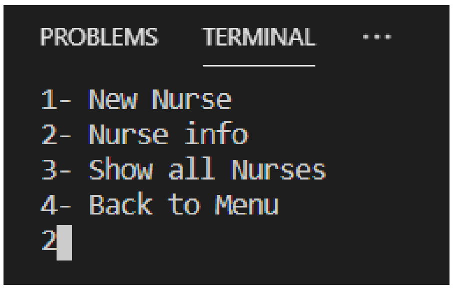
Try to add second doctor with same id;



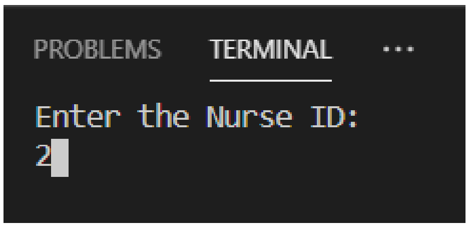
```
ID:
asfaghsh
Wrong input try again...
ID:
10
This ID already added...
ID:
2
Name:
zeynep
Surname:
elbir
Address:
amasya
Age:
22
Gender (F/M):
f
Telephone:
544
E-mail:
mail
Enter the number of position:
1- Doctor
2- Assistant Doctor
3- Professor Doctor
1
Enter the number of department:
1- Internal Diseases
2- Cardiology
3- Ent(Ear, Nose, and Throat)
4- Orthopedics
5- Pediatry
1
```

Could not add new doctor at the same id and continued until you get a different id.

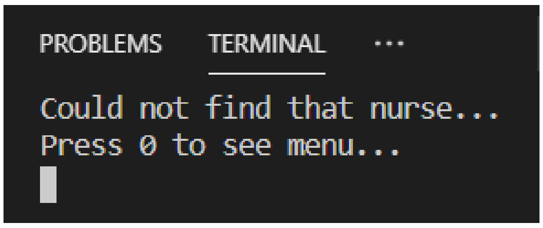
Let's look at the nurses who are not on the list



```
1- New Nurse
2- Nurse info
3- Show all Nurses
4- Back to Menu
2
```



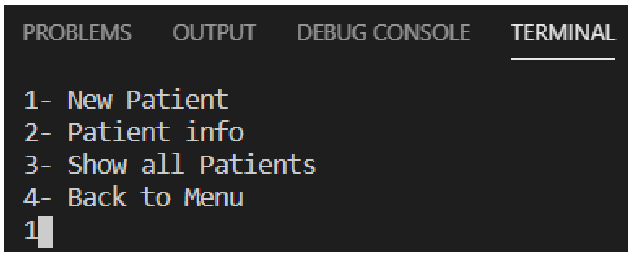
```
Enter the Nurse ID:
2
```



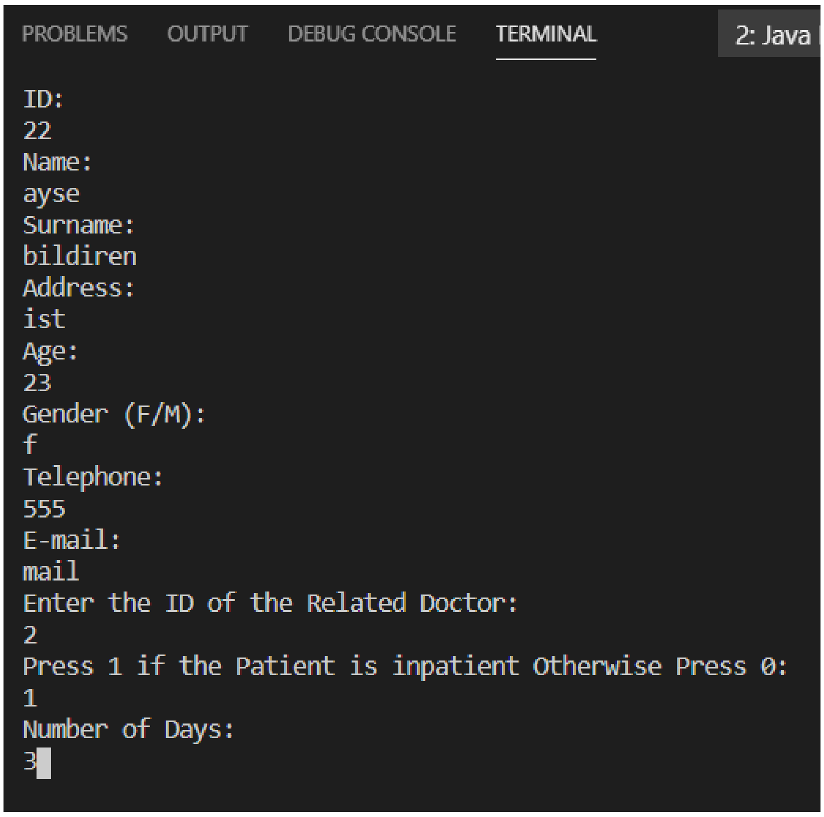
```
Could not find that nurse...
Press 0 to see menu...
```

- <u>Are ids' entered out of order displayed sequentially?</u>

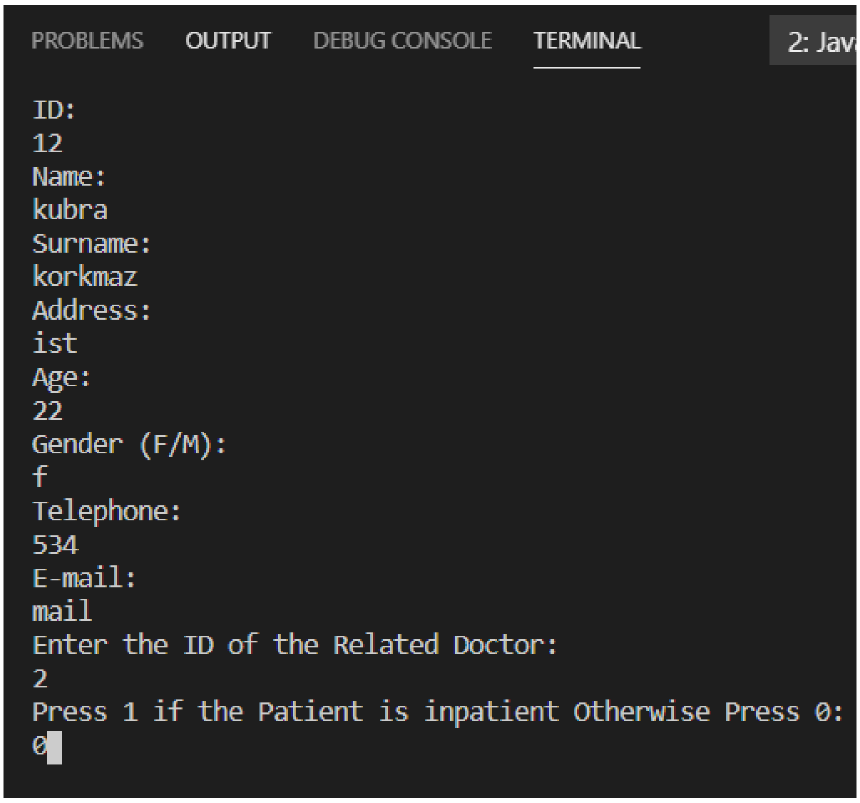
Check by adding new patients ;



```
1- New Patient
2- Patient info
3- Show all Patients
4- Back to Menu
1
```



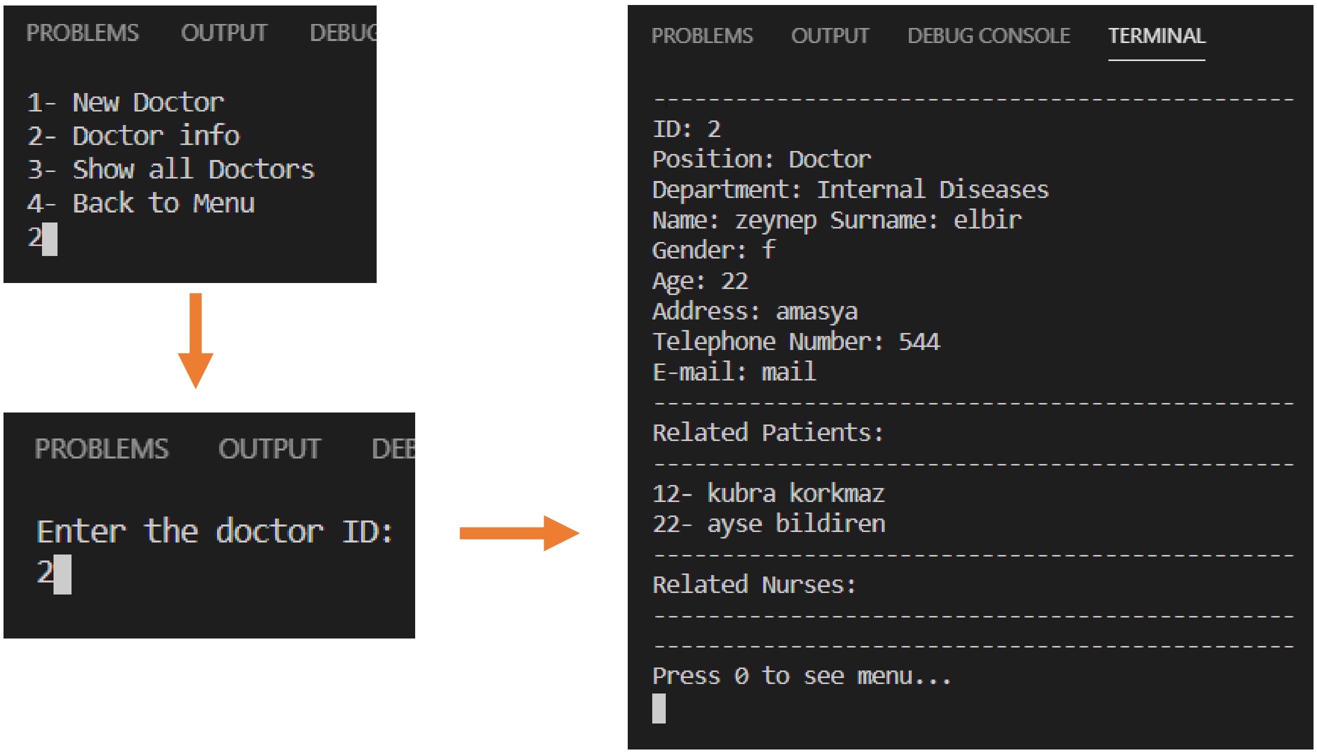
```
ID:
22
Name:
ayse
Surname:
bildiren
Address:
ist
Age:
23
Gender (F/M):
f
Telephone:
555
E-mail:
mail
Enter the ID of the Related Doctor:
2
Press 1 if the Patient is inpatient Otherwise Press 0:
1
Number of Days:
3
```



```
ID:
12
Name:
kubra
Surname:
korkmaz
Address:
ist
Age:
22
Gender (F/M):
f
Telephone:
534
E-mail:
mail
Enter the ID of the Related Doctor:
2
Press 1 if the Patient is inpatient Otherwise Press 0:
0
```

We add first id 22 then id 12 .

Check with the related doctor (we can see sequential patients list there).



```
1- New Doctor
2- Doctor info
3- Show all Doctors
4- Back to Menu
2
```

```
Enter the doctor ID:
2
```

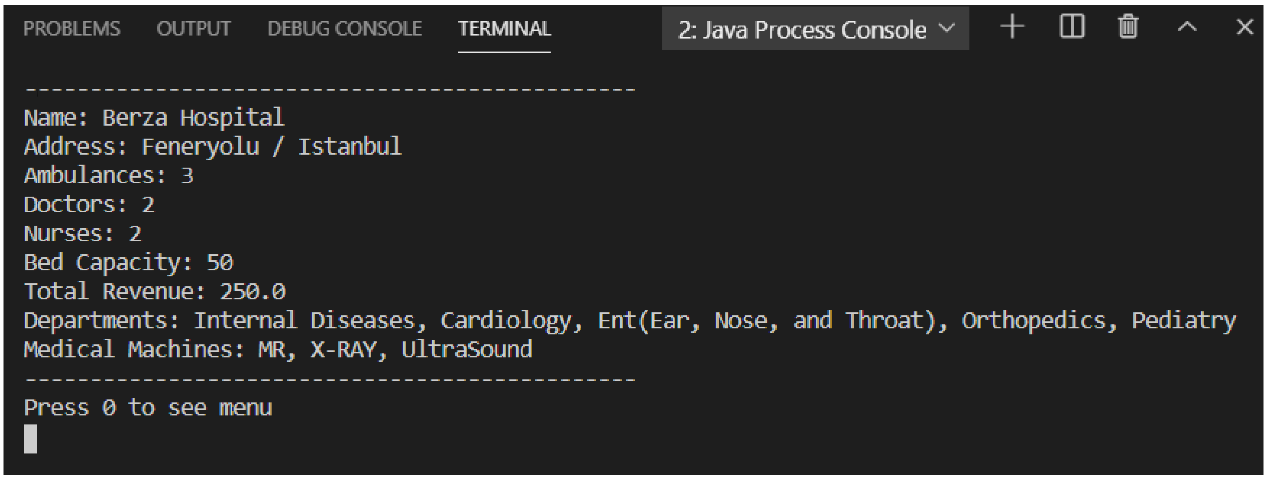
```
-------------------------------------------------
ID: 2
Position: Doctor
Department: Internal Diseases
Name: zeynep Surname: elbir
Gender: f
Age: 22
Address: amasya
Telephone Number: 544
E-mail: mail
-------------------------------------------------
Related Patients:
-------------------------------------------------
12- kubra korkmaz
22- ayse bildiren
-------------------------------------------------
Related Nurses:
-------------------------------------------------
-------------------------------------------------
Press 0 to see menu...
```

They are shown sequentally.

Now, we have 2 patients so total revenue is shown below.

The important part is that different positions in doctors and patient hospitalization effect the revenue we can get.

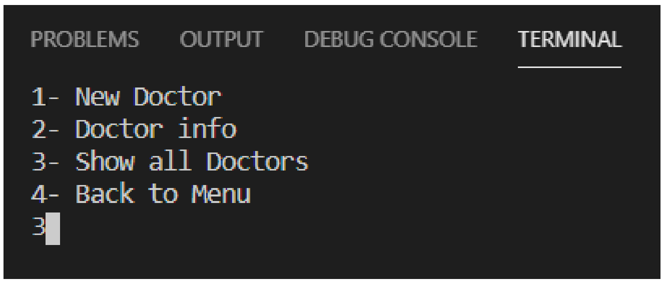
(can be also determine from the code)



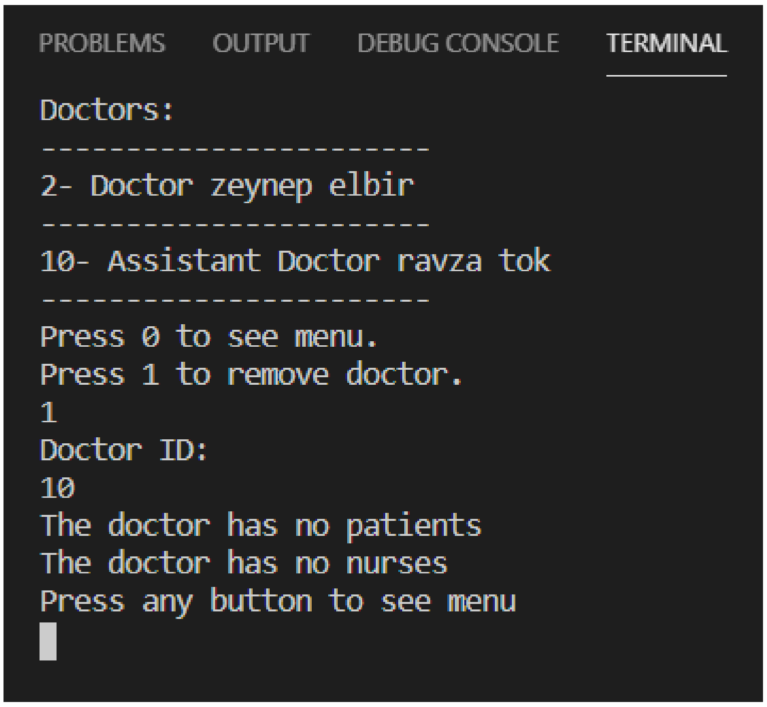
```
-----------------------------------------------
Name: Berza Hospital
Address: Feneryolu / Istanbul
Ambulances: 3
Doctors: 2
Nurses: 2
Bed Capacity: 50
Total Revenue: 250.0
Departments: Internal Diseases, Cardiology, Ent(Ear, Nose, and Throat), Orthopedics, Pediatry
Medical Machines: MR, X-RAY, UltraSound
-----------------------------------------------
Press 0 to see menu
```

- <u>Another combination for removed doctor;</u>

Remove doctor 10;



```
1- New Doctor
2- Doctor info
3- Show all Doctors
4- Back to Menu
3
```



```
Doctors:
------------------------
2- Doctor zeynep elbir
------------------------
10- Assistant Doctor ravza tok
------------------------
Press 0 to see menu.
Press 1 to remove doctor.
1
Doctor ID:
10
The doctor has no patients
The doctor has no nurses
Press any button to see menu
```

There is no related patient and nurse for doctor 10 as can bee seen the above output.

Now the classes and their methods will be explained, also at the end of the report complete code will be given.

In our project we have used two different **interfaces** names as IDate, and IPerson.

```
interface IDate // The interface that stores date data.
{
    void setter(int day, int month, int year);
    int getDay();
    int getMonth();
    int getYear();
    void showDate();
}

interface IPerson // The interface that stores person data.
{
    public void setter(int id, String name, String surname, String address, int age, char
gender, String contactNumber, String contactEmail);
    public void getFullname();
    public void getAddress();
    public void getAge();
    public void getGender();
    public void getcontactNumber();
    public void getcontactEmail();
    public void show();
    public void showshort();
    public int getID();
}
```

The information of every person that we store in the system has parent class of PersonInfo. This class implements these interfaces. PersonInfo class contains lots of **overrided** methods because of interfaces that we used.

```
class PersonInfo implements IDate,IPerson{ // The class that stores person data.
    protected int id;
    protected String name,surname;
    protected String address;
    protected int age;
    protected char gender; // F / M
    protected String contactNumber,contactEmail;
    protected int day;
    protected int month;
    protected int year;

    @Override
    public void setter(int id, String name, String surname, String address, int age, char gender, String contactNumber
, String contactEmail) { // Setter method of personal data.
        this.id = id;
        this.name = name;
        this.surname = surname;
        this.address = address;
        this.age = age;
        this.gender = gender;
        this.contactNumber = contactNumber;
        this.contactEmail = contactEmail;
    }
```

```java
    @Override
    public void getFullname() {
        System.out.println("Name: " + name + " Surname: " + surname);
    }

    @Override
    public void getAddress() {
        System.out.println("Address: " + address);
    }

    @Override
    public void getAge() {
        System.out.println("Age: " + age);
    }

    @Override
    public void getGender() {
        System.out.println("Gender: " + gender);
    }

    @Override
    public void getcontactNumber() {
        System.out.println("Telephone Number: " + contactNumber);
    }

    @Override
    public void getcontactEmail() {
        System.out.println("E-mail: " + contactEmail);
    }

    @Override
    public void show() { // Shows all the data.
        System.out.println("ID: " + id);
        System.out.println("Name: " + name + " Surname: " + surname);
        System.out.println("Gender: " + gender);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
        System.out.println("Telephone Number: " + contactNumber);
        System.out.println("E-mail: " + contactEmail);
    }

    @Override
    public int getDay() {
        return day;
    }

    @Override
    public int getMonth() {
        return month;
    }

    @Override
    public int getYear() {
        return year;
    }

    @Override
    public void setter(int day, int month, int year) { // Setter method of date.
        this.day = day;
        this.month = month;
        this.year = year;
    }

    @Override
    public void showDate() {
        System.out.println("Day: " + day + "Month: " + month + "Year: " + year);
    }

    @Override
    public int getID() {
        return id;
    }

    @Override
    public void showshort() { // Shows information shortly.
        System.out.println(id + "- " + name + " " + surname);
    }
}
```

After that we inherit this class into three different type of people such as Doctor, Nurse, and Patient. In these classes comperable interface is implemented because we would like to **override** the compareTo method. Also some of the constructors are **overloaded**, that is, in doctor class we have two different constructors. One of them does not require any parameters, in contrast the other does. So that, the program decides which one will be used, in **compile time**.

```
class Doctor extends PersonInfo implements Comparable<Doctor>{ // The class stores doctors' info.
    protected ArrayList<Patient> relatedPatients = new ArrayList<Patient>();
    protected ArrayList<Nurse> relatedNurses = new ArrayList<Nurse>();
    private String position;
    String department;
    public Doctor(){  // Empty constructor.

    }

    public Doctor(int id, String name, String surname, String address, int age, char gender, String contactNumber,
    String contactEmail, String position,String department){ // Constructor. Notice that it is Overload.
        this.position = position;
        this.department=department;
        super.setter(id, name, surname, address, age, gender, contactNumber, contactEmail);
    }

    @Override
    public void showshort() { // Shows information shortly.
        System.out.println(id + "- " + position + " " +name + " " + surname);
    }
```

```java
    public void patientenroller(Patient toenroll){ // Used to add patient to the doctor.
        relatedPatients.add(toenroll);
    }

    public void nurseenrolfler(Nurse toenroll){ // Used to add nurse to the doctor.
        relatedNurses.add(toenroll);
    }

    public void showallpatients(){ // Shows all patients that is related to the doctor.
        for(Patient p: relatedPatients){
            p.showshort();
        }
    }

    public void removepatient(int id){ // Deletes the patient from the doctor's patient list.
        for(Patient p: relatedPatients){
            if(p.id == id){
                relatedPatients.remove(p);
            }
        }
    }

    public String getPosition(){ // Returns the position of doctor in form of a string.
        return position;
    }

    public void showallnurses(){ // Shows all nurses that is related to the doctor.
        for(Nurse n: relatedNurses){
            n.showshort();
        }
    }

    public void removenurse(int id){ // Deletes the nurse from the doctor's patient list.
        for(Nurse n: relatedNurses){
            if(n.id == id){
                relatedNurses.remove(n);
            }
        }
    }

    @Override
    public void show() { // Shows information of doctor.
        System.out.println("ID: " + id);
        System.out.println("Position: " + position);
        System.out.println("Department: " + department);
        System.out.println("Name: " + name + " Surname: " + surname);
        System.out.println("Gender: " + gender);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
        System.out.println("Telephone Number: " + contactNumber);
        System.out.println("E-mail: " + contactEmail);
    }

    @Override
    public int compareTo(Doctor d) { // Compares the ids of two doctor objects.
        int compareId = d.id;
        return this.id - compareId;
    }
}

class Nurse extends PersonInfo implements Comparable<Nurse>{ // The class that stores information of nurses.
    protected Doctor relateddoctor;
    public Nurse(){ // Empty constructor.

    }
    public Nurse(int id, String name, String surname, String address, int age, char gender, String contactNumber,
                 String contactEmail, Doctor relateddoctor){ // Constructor. Notice that it is overload.
        this.relateddoctor = relateddoctor;
        super.setter(id, name, surname, address, age, gender, contactNumber, contactEmail);
    }

    public void changeDoc(Doctor newdoc){ // Changes the doctor of nurse.
        relateddoctor = newdoc;
    }

    public void showRelateddoctor(){ // Shows the doctor that the nurse is related.
        if(relateddoctor.name == null){
            System.out.println("No doctor is attached.");
        }
        else{
        System.out.println(relateddoctor.getPosition() + " " + relateddoctor.name + " " + relateddoctor.surname);
        }
    }

    @Override
    public int compareTo(Nurse n) { // Compares the ids of two nurse objects.
        int compareId = n.id;
        return this.id - compareId;
    }

}

class Patient extends PersonInfo implements Comparable<Patient>{ // The class that stores the data of patient.
    protected int isinpatient;
    protected int numberOfdays;
    protected Doctor relatedDoctor;
    private double basePrice=0;
    private double totalPrice=0;
     Patient(int id, String name, String surname, String address, int age, char gender, String contactNumber,
    String contactEmail, int isinpatient, int numberOfdays, Doctor relatedDoctor){ // Constructor.
        this.isinpatient = isinpatient;
        this.numberOfdays = numberOfdays;
        this.relatedDoctor = relatedDoctor;
        super.setter(id, name, surname, address, age, gender, contactNumber, contactEmail);
        priceCalculator();
    }

    public void baseprices(){ // Used to determine base price that patient has to pay.
```

```java
        if(relatedDoctor.getPosition().equals("Doctor")){
            basePrice = 50;
        }
        else if(relatedDoctor.getPosition().equals("Assistant Doctor")){
            basePrice = 100;
        }
        else if(relatedDoctor.getPosition().equals("Professor Doctor")){
            basePrice = 150;
        }
    }

    public void showRelateddoctor(){ // Shows the related doctor of patient.
        System.out.println(relatedDoctor.getPosition() + " " + relatedDoctor.name + " " + relatedDoctor.surname);
    }
    public double getTotalprice(){ // Returns total price (in form of double) that patient has to pay.
        baseprices();
        totalPrice = (double)(basePrice + isinpatient * basePrice * numberOfdays);
        return totalPrice;
    }
    public void priceCalculator(){ // Calculates the price that patient has to pay.
        baseprices();
        totalPrice = (double)(basePrice + isinpatient * basePrice * numberOfdays);
    }

    @Override
    public int compareTo(Patient p) { // Compares ids of two patient objects.
        int compareId = p.id;
        return this.id - compareId;
    }
}
```

Also in the project the **abstract class** feature of the JAVA has been used. HospitalInfo abstract class is used to determine properties of the hospital and the methods that will be **overrided** in the child class.

```java
abstract class HospitalInfo{ // The abstract class that stores the data of hospital.
    protected String hospitalname = "Berza Hospital";
    protected String hospitaladdress = "Feneryolu / Istanbul";
    protected int numberofambulances = 3;
    protected String [] departments = {"Internal Diseases","Cardiology","Ent(Ear, Nose, and Throat)","Orthopedics","Pediatry"};
    protected String [] machines = {"MR","X-RAY","UltraSound"};
    protected int bedcapacity = 50;
    public void showInfo(){
    }
}
```

The Hospital class inherits the HospitalInfo class and **overrides** its method. This class is used to create Hospital object and access the information about the hospital.

```java
class Hospital extends HospitalInfo{ // The class that stores the data of hospital.
    static int numberofdoctors=0;
    static int numberofnurses=0;
    protected double totalRev=0;
    public void revCal(ArrayList<Patient> clients){ // Used to calculate total revenue of hospital.
```

```java
        totalRev = 0;
        if(numberofdoctors == 0){ // If there is no doctor in hospital total revenue is 0.
            return;
        }
        else{
            for(Patient p: clients){
                totalRev += (p.getTotalprice());
            }
        }

    }

    @Override
    public void showInfo(){ // Shows the hospital information.
        System.out.println("-----------------------------------------------");
        System.out.println("Name: " + hospitalname);
        System.out.println("Address: " + hospitaladdress);
        System.out.println("Ambulances: " + numberofambulances);
        System.out.println("Doctors: " + numberofdoctors);
        System.out.println("Nurses: " + numberofnurses);
        System.out.println("Bed Capacity: " + bedcapacity);
        System.out.println("Total Revenue: " + totalRev);
        System.out.print("Departments: ");
        for(int i=0;i<departments.length;i++){
            if(i==departments.length-1){
                System.out.print(departments[i]);
            }
            else{
                System.out.print(departments[i] + ", ");
            }
        }
        System.out.print("\nMedical Machines: ");
        for(int i=0;i<machines.length;i++){
            if(i==machines.length-1){
                System.out.print(machines[i]);
            }
            else{
                System.out.print(machines[i] + ", ");
            }
        }
        System.out.print("\n");
        System.out.println("-----------------------------------------------");
    }
}
```

The Disease class is used to show diseases that can be treated in this hospital, and their recommended medicine.

```java
class Disease{ // The class that stores the information about diseases and their medicines.
    static String[][] dis_and_med= {{"Kidney Stone","Rowatinex"},{"Hypertension","Alpha Blocker"},{"Otisis Media","Augmentinin"},
    {"Osteoartriti","Parasetamol"},{"Bronchitis","Broksin"}};
    public static void showdis(){ // Shows the all diseases.
        for(int i=0;i<dis_and_med.length;i++){
            System.out.println((i+1) + "- " + dis_and_med[i][0]);
        }
    }
    public static void showmed(int disnumber){ // Shows the suggested medicine for specific disease.
        disnumber = disnumber-1;
        System.out.println("Suggested medicine for " +dis_and_med[disnumber][0]+" is "+ dis_and_med[disnumber][1] + ".");
    }
}
```

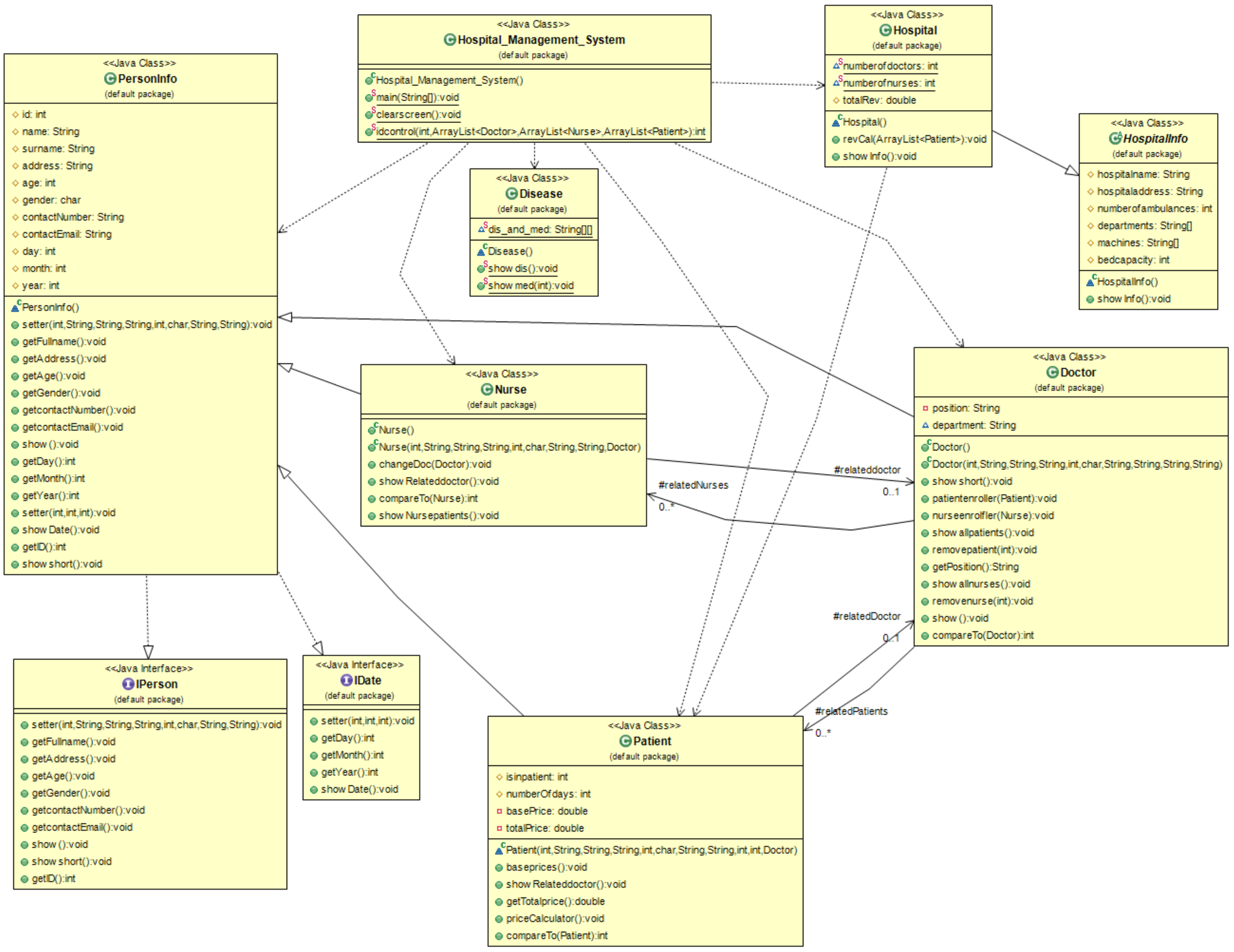
Clearscreen() method, idcontrol(int id, ArrayList<Doctor> doctors, ArrayList<Nurse> nurses, ArrayList<Patient> patients) method are defined in the Hospital_Management_System which includes the main class. Here clearscreen method is used to clear the terminal or console. Idcontrol method checks the arraylists to see if there is same id.

```
public class Hospital_Management_System{

    public static void main(String[] args){…}  // Main Class

    public static void clearscreen(){ // Clears the console screen every time it is called.
        System.out.print("\033[H\033[2J"); // ANSI ESCAPE CODE
        System.out.flush();
        //Runtime.getRuntime().exec("cls"); // For Windows Operating System.
    }
    public static int idcontrol(int id, ArrayList<Doctor> doctors, ArrayList<Nurse> nurses, ArrayList<Patient> patients){
        // Checks the all ids that is enrolled in the system, and returns 1 the id exists, returns 0 if the id does not exist.
        for(Doctor d: doctors){
            if(d.getID() == id){
                return 1;
            }
        }
        for(Nurse n: nurses){
            if(n.getID() == id){
                return 1;
            }
        }
        for(Patient p: patients){
            if(p.getID() == id){
                return 1;
            }
        }
        return 0;
    }
}
```

In the project we also have example of **polymorphism** in the main class. In case of polymorphism to access properties of child class **typecasting** is required. The polymorphism is used as follows:

```
PersonInfo toTransfered=new Doctor(); // Polymorphism
```

The **UML Diagram** of our program is given below.



The complete code is given below. Also you can find complete code in form of .java file in the RAR file.

```
import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;



public class Hospital_Management_System{ // The class that contains main

    public static void main(String[] args){ // Main Class

        Scanner input = new Scanner(System.in); // Scanner object used to take input from user

        int selection; // Used to switch between menus

        String stringselection; // Used to switch between menus

        String inputtoconvert; // Used to switch between menus

        Hospital berzahospital = new Hospital(); // The object that contains hospital's information

        ArrayList<Doctor> doctors = new ArrayList<Doctor>(); // The arraylist that stores dynamically all doctors in the hospital

        ArrayList<Nurse> nurses=new ArrayList<Nurse>(); // The arraylist that stores dynamically all nurses in the hospital

        ArrayList<Patient>patients=new ArrayList<Patient>(); // The arraylist that stores dynamically all patients in the hospital

        mainmenu: // label of main menu

        while(true) { // The while loop that goes until CLOSE PROGRAM is entered.

            clearscreen();

            System.out.println("Welcome to the Hospital Management System ...");
```

```java
System.out.println("1- HOSPITAL\n2- DOCTORS\n3- NURSES\n4- PATIENTS\n5- DISEASES\n6- CLOSE PROGRAM");

try { // To prevent errors in the program in case of wrong input.

    inputtoconvert = input.next();

    selection = Integer.parseInt(inputtoconvert);

} catch (Exception e) {

    System.out.println("Wrong input try again...");

    continue;

}

if (selection == 1) {

    clearscreen();

    berzahospital.revCal(patients); // Revenue of hospital is calculated.

    berzahospital.showInfo(); // Hospital info is showed.

    System.out.println("Press 0 to see menu");

    try {

        inputtoconvert = input.next();

        input.nextLine();

        selection = Integer.parseInt(inputtoconvert);

        if (selection == 0) {

            continue;

        } else {

            System.out.println("Wrong entrance please enter 0");

        }

    } catch (Exception e) {

        System.out.println("Wrong input try again...");

    }

} else if (selection == 2) {

    clearscreen();

    System.out.println("1- New Doctor\n2- Doctor info\n3- Show all Doctors\n4- Back to Menu");

    selection = input.nextInt();

    if (selection == 1) {

        int id;

        String name;

        String surname;

        String address;

        int age;

        char gender;

        String contactNumber;

        String contactEmail;

        String position;

        String department;

        clearscreen();

        do {

            try {

                while(true){

                    System.out.println("ID: ");

                    inputtoconvert = input.next(); input.nextLine();

                    id = Integer.parseInt(inputtoconvert);

                    if(idcontrol(id, doctors, nurses, patients) == 1){

                        System.out.println("This ID already added...");

                        continue;

                    }

                    else{

                        break;

                    }
                }
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("Name: ");
                name = input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("Surname: ");
                surname = input.next();
                input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("Address: ");
                address = input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("Age: ");
                inputtoconvert = input.next();
                input.nextLine();
                age = Integer.parseInt(inputtoconvert);
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        while (true) {
            System.out.println("Gender (F/M): ");
            gender = input.next().charAt(0);
            if (Character.toUpperCase(gender) == 'F' || Character.toUpperCase(gender) == 'M') {
                break;
            } else {
                System.out.println("Wrong input try again...");
            }
        }
```

```java
do {
    try {
        System.out.println("Telephone: ");
        contactNumber = input.next();
        input.nextLine();
        break;
    } catch (Exception e) {
        System.out.println("Wrong input try again...");
    }
} while (true);
do {
    try {
        System.out.println("E-mail: ");
        contactEmail = input.next();
        input.nextLine();
        break;
    } catch (Exception e) {
        System.out.println("Wrong input try again...");
    }
} while (true);
do {
    try {
        System.out.println("Enter the number of position: ");
        String [] positions = {"Doctor","Assistant Doctor","Professor Doctor"};
        System.out.println("1- "+ positions[0]+"\n2- "+positions[1]+"\n3- "+positions[2]);
        selection = input.nextInt(); input.nextLine();
        switch (selection){
            case 1:
                position = positions[0];
                break;
            case 2:
                position = positions[1];
                break;
            case 3:
                position = positions[2];
                break;
            default:
                position = "";
                break;
        }
        break;
    } catch (Exception e) {
        System.out.println("Wrong input try again...");
    }
} while (true);
do {
    try {
        System.out.println("Enter the number of department: ");
        String [] departments = {"Internal Diseases","Cardiology","Ent(Ear, Nose, and Throat)","Orthopedics","Pediatry"};
        System.out.println("1- "+ departments[0]+"\n2- "+departments[1]+"\n3- "+departments[2]+"\n4- "+departments[3]+"\n5- "+departments[4]);
        selection = input.nextInt(); input.nextLine();
        switch (selection){
            case 1:
                department = departments[0];
```

```java
                            break;
                        case 2:
                            department = departments[1];
                            break;
                        case 3:
                            department = departments[2];
                            break;
                        case 4:
                            department = departments[3];
                            break;
                        case 5:
                            department = departments[4];
                            break;
                        default:
                            department = "";
                            break;
                    }
                    break;
                } catch (Exception e) {
                    System.out.println("Wrong input try again...");
                }
            } while (true);
            Doctor newdoctor = new Doctor(id, name, surname, address, age, gender, contactNumber, contactEmail, position,department);
            doctors.add(newdoctor);
            Hospital.numberofdoctors++; // It increases the number of doctors in hospital using static variable.
        } else if (selection == 2) {
            try {
                int idOfdoctor;
                int flag = 0;
                clearscreen();
                System.out.println("Enter the doctor ID: ");
                inputtoconvert = input.next();
                input.nextLine();
                idOfdoctor = Integer.parseInt(inputtoconvert);
                clearscreen();
                for (Doctor d : doctors) {
                    if (d.id == idOfdoctor) {
                        System.out.println("-----------------------------------------------");
                        d.show();
                        System.out.println("-----------------------------------------------");
                        Collections.sort(d.relatedPatients); //It sorts doctor's patients.
                        System.out.println("Related Patients: ");
                        System.out.println("-----------------------------------------------");
                        d.showallpatients();
                        System.out.println("-----------------------------------------------");
                        Collections.sort(d.relatedNurses); //It sorts doctor's nurses.
                        System.out.println("Related Nurses: ");
                        System.out.println("-----------------------------------------------");
                        d.showallnurses();
                        System.out.println("-----------------------------------------------");
                        flag = 1;
                        break;
                    }
                }
```

```java
                    if (flag == 0) {
                        System.out.println("Could not find that doctor...");
                    }
                    System.out.println("Press 0 to see menu...");
                    selection = input.nextInt(); input.nextLine();
                    if(selection == 0){
                        continue;
                    }
                } catch (Exception e) {
                    System.out.println("Wrong input try again...");
                }
            } else if (selection == 3) {
                clearscreen();
                System.out.println("Doctors:");
                Collections.sort(doctors);
                for (Doctor d : doctors) {
                    System.out.println("-----------------------");
                    d.showshort();
                }
                try {
                    System.out.println("-----------------------");
                    System.out.println("Press 0 to see menu.");
                    System.out.println("Press 1 to remove doctor.");
                    inputtoconvert = input.next(); input.nextLine();
                    selection = Integer.parseInt(inputtoconvert);
                    if (selection == 0) {
                        continue;
                    }
                    else if(selection == 1){
                        try{
                            int flag=0;
                            System.out.println("Doctor ID: ");
                            inputtoconvert = input.next(); input.nextLine();
                            selection = Integer.parseInt(inputtoconvert);
                            Doctor toRemoved=new Doctor(); // The doctor that will be removed.
                            PersonInfo toTransfered=new Doctor(); // Polymorphism
                            for(Doctor d: doctors){
                                if(d.getID() == selection){
                                    toRemoved=d;
                                    Hospital.numberofdoctors--; // It decreases the number of doctors in hospital using static variable.
                                    break;
                                }
                            }
                            boolean isNurseEmpty=toRemoved.relatedNurses.isEmpty(); // Checks the nurse list if it is empty.
                            boolean isEmpty=toRemoved.relatedPatients.isEmpty(); // Checks the patient list if it is empty.
                            if(isEmpty==false) {
                                for (int i=0; i<doctors.size();i++) {
                                    if (toRemoved.department.equals(doctors.get(i).department)==true && toRemoved.id!=doctors.get(i).id) {
                                        flag = 1;
                                        toTransfered =doctors.get(i) ; // If exists that indicated the doctor to be transferred.
                                        break;
                                    }
                                }
                                if (flag == 1) {
                                    System.out.println("The patients have transferred to another doctor");
                                    for (Patient p : toRemoved.relatedPatients) {  // The patients of doctor which will be removed.
                                        ( (Doctor)toTransfered).patientenroller(p); // Type casting due to polymorphism.
                                        p.relatedDoctor=(Doctor)toTransfered; // Patients are transferred to new doctor.
                                    }
                                    System.out.println("Press any button to see menu");
                                    input.nextLine();
                                }
                                else {
                                    System.out.println("The patients have transferred to another hospital...");
                                    System.out.println("Press any button to see menu");
                                    input.nextLine();
                                    for(Patient p: toRemoved.relatedPatients){ // Patients are removed.
                                        patients.remove(p);
                                    }
                                }
                            }
                            else{
                                System.out.println("The doctor has no patients");
                            }
                            if(isNurseEmpty==true){
                                System.out.println("The doctor has no nurses");


                            }
                            else{
                                for (Nurse n : toRemoved.relatedNurses) { // The nurses of doctor which will be removed.
                                    ( (Doctor)toTransfered).nurseenrolfler(n); // Type casting due to polymorphism.
                                }
                                for(Nurse n:toRemoved.relatedNurses){
                                        n.relateddoctor=(Doctor)toTransfered; // Nurses are transferred to new doctor.
                                }
                            }
                            doctors.remove(toRemoved);
                            System.out.println("Press any button to see menu");
                            input.nextLine();
                        }catch(Exception e){
                        }
                    }
                } catch (Exception e) {
                    System.out.println("Wrong input try again...");
                }
            } else if (selection == 4) {
                continue;
            } else {
                System.out.println("Wrong entrance!!!");
            }
        } else if (selection == 3) {
            clearscreen();
            System.out.println("1- New Nurse\n2- Nurse info\n3- Show all Nurses\n4- Back to Menu");
            selection = input.nextInt(); input.nextLine();
            boolean checkDocList = doctors.isEmpty(); // If there is no doctor in hospital, nurse cannot be added.
            if (selection == 1) {
                int nurseId;
                String name;
```

```java
String surname;
String address;
int age;
char gender;
String contactNumber;
String contactEmail;
Doctor relatedDoctor = new Doctor();
clearscreen();
if(checkDocList==true){
    System.out.println("There is no doctor in the List so we cannot insert a Nurse.");
    System.out.println("Press any button to see menu.");
    stringselection = input.nextLine();
    continue mainmenu; // Goes to the mainmenu label.
}
do {
    try {
        System.out.println("ID: ");
        inputtoconvert = input.next();
        input.nextLine();
        nurseId = Integer.parseInt(inputtoconvert);
        if (idcontrol(nurseId, doctors, nurses, patients) == 1) {
            System.out.println("This ID already added...");
            continue;
        } else {
            break;
        }
    } catch (Exception e) {
        System.out.println("Wrong input try again...");
    }
} while (true);
do {
    try {
        System.out.println("Name: ");
        name = input.nextLine();
        break;
    } catch (Exception e) {
        System.out.println("Wrong input try again...");
    }
} while (true);
do {
    try {
        System.out.println("Surname: ");
        surname = input.next();
        input.nextLine();
        break;
    } catch (Exception e) {
        System.out.println("Wrong input try again...");
    }
} while (true);
do {
    try {
        System.out.println("Address: ");
        address = input.nextLine();
        break;
```

```java
                    } catch (Exception e) {

                        System.out.println("Wrong input try again...");

                    }

                } while (true);

                do {

                    try {

                        System.out.println("Age: ");

                        inputtoconvert = input.next();

                        input.nextLine();

                        age = Integer.parseInt(inputtoconvert);

                        break;

                    } catch (Exception e) {

                        System.out.println("Wrong input try again...");

                    }

                } while (true);

                while (true) {

                    System.out.println("Gender (F/M): ");

                    gender = input.next().charAt(0);

                    if (Character.toUpperCase(gender) == 'F' || Character.toUpperCase(gender) == 'M') {

                        break;

                    } else {

                        System.out.println("Wrong input try again...");

                    }

                }

                do {

                    try {

                        System.out.println("Telephone: ");

                        contactNumber = input.next();

                        input.nextLine();

                        break;

                    } catch (Exception e) {

                        System.out.println("Wrong input try again...");

                    }

                } while (true);

                do {

                    try {

                        System.out.println("E-mail: ");

                        contactEmail = input.next();

                        input.nextLine();

                        break;

                    } catch (Exception e) {

                        System.out.println("Wrong input try again...");

                    }

                } while (true);

                while(true){ // Loop iterates until the id matches with any doctor.

                        System.out.println("Enter the ID of the Related Doctor: ");

                        inputtoconvert = input.next();

                        input.nextLine();

                        int doctorId;

                        int flag=0;

                        doctorId = Integer.parseInt(inputtoconvert);

                        for (int i = 0; i < doctors.size(); i++) {

                            if (doctorId == doctors.get(i).id) {

                                relatedDoctor = doctors.get(i);
```

```
                                    flag=1;
                                    break;
                                }
                            }
                            if(flag==1)
                            break;
                    }
                    Nurse newnurse;
                    newnurse = new Nurse(nurseId, name, surname, address, age, gender, contactNumber, contactEmail, relatedDoctor);
                    nurses.add(newnurse);
                    relatedDoctor.nurseenrolfler(newnurse);
                    Hospital.numberofnurses++; // It increases the number of nurses in hospital using static variable.
                }
            else if (selection == 2) {
                    try {
                        int idOfNurse;
                        int flag = 0;
                        clearscreen();
                        System.out.println("Enter the Nurse ID: ");
                        inputtoconvert = input.next();
                        input.nextLine();
                        idOfNurse = Integer.parseInt(inputtoconvert);
                        clearscreen();
                        for (Nurse n : nurses) {
                            if (n.id == idOfNurse) {
                                System.out.println("------------------------------------------------------");
                                n.show();
                                n.showRelateddoctor();
                                flag = 1;
                                System.out.println("------------------------------------------------------");
                                break;
                            }
                        }
                        if (flag == 0) {
                            System.out.println("Could not find that nurse...");
                        }
                        System.out.println("Press 0 to see menu...");
                        inputtoconvert = input.next(); input.nextLine();
                        selection = Integer.parseInt(inputtoconvert);
                        if(selection == 0){
                            continue;
                        }
                    } catch (Exception e) {
                        System.out.println("Wrong input try again...");
                    }
                } else if (selection == 3) {
                    clearscreen();
                    System.out.println("Nurses: ");
                    Collections.sort(nurses);
                    for (Nurse n : nurses) {
                        System.out.println("-----------------------");
                        n.showshort();
                    }
                    try {
```

```java
                        System.out.println("----------------------");
                        System.out.println("Press 0 to see menu.");
                        System.out.println("Press 1 to remove nurse.");
                        System.out.println("Press 2 to change related doctor.");
                        inputtoconvert = input.next();
                        input.nextLine();
                        selection = Integer.parseInt(inputtoconvert);
                        if (selection == 0) {
                            continue;
                        }
                        else if(selection == 1){
                            clearscreen();
                            Nurse toDeleted=new Nurse();
                            System.out.println("Enter nurse ID: ");
                            selection = input.nextInt(); input.nextLine();
                            int i;
                            for(i=0; i<nurses.size();i++){
                                if(nurses.get(i).id==selection){
                                    toDeleted=nurses.get(i); // Doctor to be removed.
                                    break;
                                }
                            }
                            toDeleted.relateddoctor.relatedNurses.remove(toDeleted); // The nurse is removed from the list of nurses of the doctor.
                            nurses.remove(toDeleted); // Removed from nurse list.
                            Hospital.numberofnurses--; // It decreases the number of nurses in hospital using static variable.
                        }
                        else if(selection == 2){
                            String nurseID;
                            String doctorID;
                            System.out.println("Nurse's ID: ");
                            nurseID = input.nextLine();
                            System.out.println("Doctor's ID: "); // ID of doctor that the nurse will be transferred.
                            doctorID = input.nextLine();
                            for(Doctor d: doctors){
                                if(d.getID() == Integer.parseInt(doctorID)){
                                    for(Nurse n: nurses){
                                        if(n.getID() == Integer.parseInt(nurseID)){
                                            d.nurseenrolfler(n); // The nurse is added to the doctor's nurse list.
                                            n.relateddoctor.relatedNurses.remove(n); // Nurse is removed from previous doctor's list.
                                            n.relateddoctor=d; // Nurse's related doctor is changed.
                                            break;
                                        }
                                    }
                                }
                            }
                        }
                    } catch (Exception e) {
                    }
                } else if (selection == 4) {
                    continue;
                } else {
                    System.out.println("Wrong entrance!!!");
                }
            } else if (selection == 4) {
```

```java
clearscreen();

System.out.println("1- New Patient\n2- Patient info\n3- Show all Patients\n4- Back to Menu");

selection = input.nextInt(); input.nextLine();

boolean checkDocList=doctors.isEmpty();

if (selection == 1) {

    int id;

    int doctorId;

    String name;

    String surname;

    String address;

    int age;

    char gender;

    String contactNumber;

    String contactEmail;

    Doctor relatedDoctor = new Doctor();

    int isInpatient;

    int numberOfDays=0;

    clearscreen();

    if(checkDocList==true){ // Patient cannot be added before any doctor has been added.

        System.out.println("There is no doctor in the List so we cannot insert a Patient");

        System.out.println("Press any button to see menu.");

        stringselection = input.nextLine();

        continue mainmenu; // Goes to mainmenu label.


    }

    do {

        try {

            System.out.println("ID: ");

                inputtoconvert = input.next(); input.nextLine();

                id = Integer.parseInt(inputtoconvert);

                if(idcontrol(id, doctors, nurses, patients) == 1){

                    System.out.println("This ID already added...");

                    continue;

                }

                else{

                    break;

                }

        } catch (Exception e) {

            System.out.println("Wrong input try again...");

        }

    } while (true);

    do {

        try {

            System.out.println("Name: ");

            name = input.nextLine();

            break;

        } catch (Exception e) {

            System.out.println("Wrong input try again...");

        }

    } while (true);

    do {

        try {

            System.out.println("Surname: ");

            surname = input.next();
```

```java
                input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("Address: ");
                address = input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("Age: ");
                inputtoconvert = input.next();
                input.nextLine();
                age = Integer.parseInt(inputtoconvert);
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        while (true) {
            System.out.println("Gender (F/M): ");
            gender = input.next().charAt(0);
            if (Character.toUpperCase(gender) == 'F' || Character.toUpperCase(gender) == 'M') {
                break;
            } else {
                System.out.println("Wrong input try again...");
            }
        }
        do {
            try {
                System.out.println("Telephone: ");
                contactNumber = input.next();
                input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        do {
            try {
                System.out.println("E-mail: ");
                contactEmail = input.next();
                input.nextLine();
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        while (true){
                System.out.println("Enter the ID of the Related Doctor: ");
                inputtoconvert = input.next();
                input.nextLine();
                int flag=0;
                doctorId = Integer.parseInt(inputtoconvert);
                for (int i = 0; i < doctors.size(); i++) {
                    if (doctorId == doctors.get(i).id) {
                        relatedDoctor = doctors.get(i);
                        flag=1;
                        break;
                    }
                }
                if(flag==1)
                break;
        }
        do {
            try {
                System.out.println("Press 1 if the Patient is inpatient Otherwise Press 0: ");
                inputtoconvert = input.next();
                input.nextLine();
                isInpatient = Integer.parseInt(inputtoconvert);
                break;
            } catch (Exception e) {
                System.out.println("Wrong input try again...");
            }
        } while (true);
        if(isInpatient==1){
            do {
                try {
                    System.out.println("Number of Days: ");
                    inputtoconvert = input.next();
                    input.nextLine();
                    numberOfDays = Integer.parseInt(inputtoconvert);
                    break;
                } catch (Exception e) {
                    System.out.println("Wrong input try again...");
                }
            } while (true);
        }
        else{
            numberOfDays=0;
        }
        Patient newPatient;
        newPatient = new Patient(id, name, surname, address, age, gender, contactNumber, contactEmail,isInpatient, numberOfDays,relatedDoctor);
        patients.add(newPatient); // Patient is added to the patient list.
        relatedDoctor.patientenroller(newPatient); // Patient is added to the Doctor's patient list.
    }
    else if (selection == 2) {
        try {
            int idOfPatient;
            int flag = 0;
            clearscreen();
```

```java
                        System.out.println("Enter the Patient ID: ");

                        inputtoconvert = input.next();

                        input.nextLine();

                        idOfPatient = Integer.parseInt(inputtoconvert);

                        clearscreen();

                        for (Patient p : patients) {

                            if (idOfPatient==p.id) {

                                System.out.println("-----------------------------------------------------");

                                p.show();

                                p.showRelateddoctor();

                                flag = 1;

                                System.out.println("-----------------------------------------------------");

                                break;

                            }

                        }

                        if (flag == 0) {

                            System.out.println("Could not find that patient...");

                        }

                        System.out.println("Press 0 to see menu...");

                        selection = input.nextInt(); input.nextLine();

                        if(selection == 0 ){

                            continue;

                        }

                    } catch (Exception e) {

                        System.out.println("Wrong input try again...");

                    }

                } else if (selection == 3) {

                    clearscreen();

                    Collections.sort(patients);

                    System.out.println("Patients: ");

                    for (Patient p : patients) {

                        System.out.println("-----------------------");

                        p.showshort();

                    }

                    try {

                        System.out.println("-----------------------");

                        System.out.println("Press 0 to see menu.");

                        System.out.println("Press 1 to remove patient.");

                        stringselection = input.nextLine();

                        if (stringselection.equals("0")) {

                            continue;

                        }

                        else if(stringselection.equals("1")){

                            clearscreen();

                            System.out.println("Patient ID: ");

                            stringselection = input.nextLine();

                            for (Patient p : patients) {

                                if(p.getID() == Integer.parseInt(stringselection)){

                                    patients.remove(p); // Patient is removed from patient list

                                    p.relatedDoctor.removepatient(p.getID()); // Patient is removed from the doctor's patient list.

                                }

                            }

                        }

                    } catch (Exception e) {
                    }
                } else if (selection == 4) {
                    continue;
                }

                else {
                    System.out.println("Wrong entrance!!!");
                }
            }
            else if(selection==5){
                clearscreen();
                System.out.println("1- Show all Disease\n2- Back to Menu");
                selection = input.nextInt();input.nextLine();
                if(selection == 1){
                    clearscreen();
                    Disease.showdis();
                    System.out.println("-----------------------");
                    System.out.println("Press 0 to see menu");
                    System.out.println("Press 1-5 to see medicines");
                    selection = input.nextInt();input.nextLine();
                    clearscreen();
                    if(selection==1){
                        Disease.showmed(1);
                    }
                    else if(selection==2){
                        Disease.showmed(2);
                    }
                    else if(selection==3){
                        Disease.showmed(3);
                    }
                    else if(selection==4){
                        Disease.showmed(4);
                    }
                    else if(selection==5){
                        Disease.showmed(5);
                    }
                    else if(selection==0){
                        continue;
                    }
                    else{
                        System.out.println("Wrong input...");
                    }
                    System.out.println("Press any button to see menu.");
                    selection = input.nextInt(); input.nextLine();
                }
                else if(selection == 2){
                    continue;
                }
                else {
                    System.out.println("Wrong entrance!!!");
                }
            }
            else if(selection == 6){
                input.close();

                break;

            }

            else{

                continue;

            }

        }

    }

    public static void clearscreen(){ // Clears the console screen every time it is called.

        System.out.print("\033[H\033[2J"); // ANSI ESCAPE CODE

        System.out.flush();

        //Runtime.getRuntime().exec("cls"); // For Windows Operating System.

    }

    public static int idcontrol(int id, ArrayList<Doctor> doctors, ArrayList<Nurse> nurses, ArrayList<Patient> patients){

        // Checks the all ids that is enrolled in the system, and returns 1 the id exists, returns 0 if the id does not exist.

        for(Doctor d: doctors){

            if(d.getID() == id){

                return 1;

            }

        }

        for(Nurse n: nurses){

            if(n.getID() == id){

                return 1;

            }

        }

        for(Patient p: patients){

            if(p.getID() == id){

                return 1;

            }

        }

        return 0;

    }

}


interface IDate // The interface that stores date data.

{

    void setter(int day, int month, int year);

    int getDay();

    int getMonth();

    int getYear();

    void showDate();

}


interface IPerson // The interface that stores person data.

{

    public void setter(int id, String name, String surname, String address, int age, char gender, String contactNumber, String contactEmail);

    public void getFullname();

    public void getAddress();

    public void getAge();

    public void getGender();

    public void getcontactNumber();

    public void getcontactEmail();

    public void show();

    public void showshort();
```

```java
    public int getID();

}


class PersonInfo implements IDate,IPerson{ // The class that stores person data.

    protected int id;

    protected String name,surname;

    protected String address;

    protected int age;

    protected char gender; // F / M

    protected String contactNumber,contactEmail;

    protected int day;

    protected int month;

    protected int year;


    @Override

    public void setter(int id, String name, String surname, String address, int age, char gender, String contactNumber,

            String contactEmail) { // Setter method of personal data.

        this.id = id;

        this.name = name;

        this.surname = surname;

        this.address = address;

        this.age = age;

        this.gender = gender;

        this.contactNumber = contactNumber;

        this.contactEmail = contactEmail;

    }


    @Override

    public void getFullname() {

        System.out.println("Name: " + name + " Surname: " + surname);

    }


    @Override

    public void getAddress() {

        System.out.println("Address: " + address);

    }


    @Override

    public void getAge() {

        System.out.println("Age: " + age);

    }


    @Override

    public void getGender() {

        System.out.println("Gender: " + gender);

    }


    @Override

    public void getcontactNumber() {

        System.out.println("Telephone Number: " + contactNumber);

    }


    @Override

    public void getcontactEmail() {
        System.out.println("E-mail: " + contactEmail);

    }


    @Override

    public void show() { // Shows all the data.

        System.out.println("ID: " + id);

        System.out.println("Name: " + name + " Surname: " + surname);

        System.out.println("Gender: " + gender);

        System.out.println("Age: " + age);

        System.out.println("Address: " + address);

        System.out.println("Telephone Number: " + contactNumber);

        System.out.println("E-mail: " + contactEmail);

    }


    @Override

    public int getDay() {

        return day;

    }


    @Override

    public int getMonth() {

        return month;

    }


    @Override

    public int getYear() {

        return year;

    }


    @Override

    public void setter(int day, int month, int year) { // Setter method of date.

        this.day = day;

        this.month = month;

        this.year = year;

    }


    @Override

    public void showDate() {

        System.out.println("Day: " + day + "Month: " + month + "Year: " + year);

    }


    @Override

    public int getID() {

        return id;

    }


    @Override

    public void showshort() { // Shows information shortly.

        System.out.println(id + "- " + name + " " + surname);

    }

}


class Doctor extends PersonInfo implements Comparable<Doctor>{ // The class stores doctors' info.

    protected ArrayList<Patient> relatedPatients = new ArrayList<Patient>();
    protected ArrayList<Nurse> relatedNurses = new ArrayList<Nurse>();
    private String position;
    String department;
    public Doctor(){  // Empty constructor.


    }



    public Doctor(int id, String name, String surname, String address, int age, char gender, String contactNumber,
    String contactEmail, String position,String department){ // Constructor. Notice that it is Overload.
        this.position = position;
        this.department=department;
        super.setter(id, name, surname, address, age, gender, contactNumber, contactEmail);
    }

    @Override
    public void showshort() { // Shows information shortly.
        System.out.println(id + "- " + position + " " +name + " " + surname);
    }

    public void patientenroller(Patient toenroll){ // Used to add patient to the doctor.
        relatedPatients.add(toenroll);
    }

    public void nurseenrolfler(Nurse toenroll){ // Used to add nurse to the doctor.
        relatedNurses.add(toenroll);
    }

    public void showallpatients(){ // Shows all patients that is related to the doctor.
        for(Patient p: relatedPatients){
            p.showshort();
        }
    }

    public void removepatient(int id){ // Deletes the patient from the doctor's patient list.
        for(Patient p: relatedPatients){
            if(p.id == id){
                relatedPatients.remove(p);
            }
        }
    }

    public String getPosition(){ // Returns the position of doctor in form of a string.
        return position;
    }

    public void showallnurses(){ // Shows all nurses that is related to the doctor.
        for(Nurse n: relatedNurses){
            n.showshort();
        }
    }

    public void removenurse(int id){ // Deletes the nurse from the doctor's patient list.
        for(Nurse n: relatedNurses){
            if(n.id == id){
                relatedNurses.remove(n);
            }
        }
    }

    @Override
    public void show() { // Shows information of doctor.
        System.out.println("ID: " + id);
        System.out.println("Position: " + position);
        System.out.println("Department: " + department);
        System.out.println("Name: " + name + " Surname: " + surname);
        System.out.println("Gender: " + gender);
        System.out.println("Age: " + age);
        System.out.println("Address: " + address);
        System.out.println("Telephone Number: " + contactNumber);
        System.out.println("E-mail: " + contactEmail);
    }

    @Override
    public int compareTo(Doctor d) { // Compares the ids of two doctor objects.
        int compareId = d.id;
        return this.id - compareId;
    }
}

class Nurse extends PersonInfo implements Comparable<Nurse>{ // The class that stores information of nurses.
    protected Doctor relateddoctor;
    public Nurse(){ // Empty constructor.

    }
    public Nurse(int id, String name, String surname, String address, int age, char gender, String contactNumber,
                 String contactEmail, Doctor relateddoctor){ // Constructor. Notice that it is overload.
        this.relateddoctor = relateddoctor;
        super.setter(id, name, surname, address, age, gender, contactNumber, contactEmail);
    }

    public void changeDoc(Doctor newdoc){ // Changes the doctor of nurse.
        relateddoctor = newdoc;
    }

    public void showRelateddoctor(){ // Shows the doctor that the nurse is related.
        if(relateddoctor.name == null){
            System.out.println("No doctor is attached.");
        }
        else{
        System.out.println(relateddoctor.getPosition() + " " + relateddoctor.name + " " + relateddoctor.surname);
        }
    }

    @Override
    public int compareTo(Nurse n) { // Compares the ids of two nurse objects.
        int compareId = n.id;
        return this.id - compareId;
```

```java
    }


}


class Patient extends PersonInfo implements Comparable<Patient>{ // The class that stores the data of patient.

    protected int isinpatient;

    protected int numberOfdays;

    protected Doctor relatedDoctor;

    private double basePrice=0;

    private double totalPrice=0;

     Patient(int id, String name, String surname, String address, int age, char gender, String contactNumber,

    String contactEmail, int isinpatient, int numberOfdays, Doctor relatedDoctor){ // Constructor.

        this.isinpatient = isinpatient;

        this.numberOfdays = numberOfdays;

        this.relatedDoctor = relatedDoctor;

        super.setter(id, name, surname, address, age, gender, contactNumber, contactEmail);

        priceCalculator();

    }


    public void baseprices(){ // Used to determine base price that patient has to pay.

        if(relatedDoctor.getPosition().equals("Doctor")){

            basePrice = 50;

        }

        else if(relatedDoctor.getPosition().equals("Assistant Doctor")){

            basePrice = 100;

        }

        else if(relatedDoctor.getPosition().equals("Professor Doctor")){

            basePrice = 150;

        }

    }


    public void showRelateddoctor(){ // Shows the related doctor of patient.

        System.out.println(relatedDoctor.getPosition() + " " + relatedDoctor.name + " " + relatedDoctor.surname);

    }

    public double getTotalprice(){ // Returns total price (in form of double) that patient has to pay.

        baseprices();

        totalPrice = (double)(basePrice + isinpatient * basePrice * numberOfdays);

        return totalPrice;

    }

    public void priceCalculator(){ // Calculates the price that patient has to pay.

        baseprices();

        totalPrice = (double)(basePrice + isinpatient * basePrice * numberOfdays);

    }


    @Override

    public int compareTo(Patient p) { // Compares ids of two patient objects.

        int compareId = p.id;

        return this.id - compareId;

    }

}


abstract class HospitalInfo{ // The abstract class that stores the data of hospital.

    protected String hospitalname = "Berza Hospital";

    protected String hospitaladdress = "Feneryolu / Istanbul";
    protected int numberofambulances = 3;

    protected String [] departments = {"Internal Diseases","Cardiology","Ent(Ear, Nose, and Throat)","Orthopedics","Pediatry"};

    protected String [] machines = {"MR","X-RAY","UltraSound"};

    protected int bedcapacity = 50;

    public void showInfo(){

    }

}


class Hospital extends HospitalInfo{ // The class that stores the data of hospital.

    static int numberofdoctors=0;

    static int numberofnurses=0;

    protected double totalRev=0;

    public void revCal(ArrayList<Patient> clients){ // Used to calculate total revenue of hospital.

        totalRev = 0;

        if(numberofdoctors == 0){ // If there is no doctor in hospital total revenue is 0.

            return;

        }

        else{

            for(Patient p: clients){

                totalRev += (p.getTotalprice());

            }

        }


    }


    @Override

    public void showInfo(){ // Shows the hospital information.

        System.out.println("-----------------------------------------------");

        System.out.println("Name: " + hospitalname);

        System.out.println("Address: " + hospitaladdress);

        System.out.println("Ambulances: " + numberofambulances);

        System.out.println("Doctors: " + numberofdoctors);

        System.out.println("Nurses: " + numberofnurses);

        System.out.println("Bed Capacity: " + bedcapacity);

        System.out.println("Total Revenue: " + totalRev);

        System.out.print("Departments: ");

        for(int i=0;i<departments.length;i++){

            if(i==departments.length-1){

                System.out.print(departments[i]);

            }

            else{

                System.out.print(departments[i] + ", ");

            }

        }

        System.out.print("\nMedical Machines: ");

        for(int i=0;i<machines.length;i++){

            if(i==machines.length-1){

                System.out.print(machines[i]);

            }

            else{

                System.out.print(machines[i] + ", ");

            }

        }

        System.out.print("\n");
        System.out.println("-----------------------------------------------");

    }

}



class Disease{ // The class that stores the information about diseases and their medicines.

    static String[][] dis_and_med= {{"Kidney Stone","Rowatinex"},{"Hypertension","Alpha Blocker"},{"Otisis Media","Augmentinin"},

    {"Osteoartriti","Parasetamol"},{"Bronchitis","Broksin"}};

    public static void showdis(){ // Shows the all diseases.

        for(int i=0;i<dis_and_med.length;i++){

            System.out.println((i+1) + "- " + dis_and_med[i][0]);

        }

    }

    public static void showmed(int disnumber){ // Shows the suggested medicine for specific disease.

        disnumber = disnumber-1;

        System.out.println("Suggested medicine for " +dis_and_med[disnumber][0]+" is "+ dis_and_med[disnumber][1] + ".");

    }

}
```