LAB211Assignment

Type:	**Short Assignment**
Code:	**J1.S.P0055**
LOC:	**73**
Slot(s):	**1**

Title

Doctor management program.

Background

N/A

Program Specifications

Write a program to manage the doctor information below:

Display the menu:

1. Add Doctor
2. Update Doctor.
3. Delete Doctor
4. Search Doctor.
5. Exit.

- If the user chooses 1 then add one record to data including Code(String), Name(String), Specialization(String), Availability(int).
- If user chooses 2: then request enter the code. If it does not exist Code, the notification "Doctor code does not exist". Otherwise user can edit of the remaining information. If new information is blank then not change old information.
- If user chooses 3: the Code requires the user to enter Code and delete information if Code exists, if Code does not exist, the notification "code does not exist Doctor".
- If user chooses 4: require user to enter search strings, search and returns the list for users.
- If user chooses 5: exit the program.

Function details:

Function 1: Display GUI and Input Data.

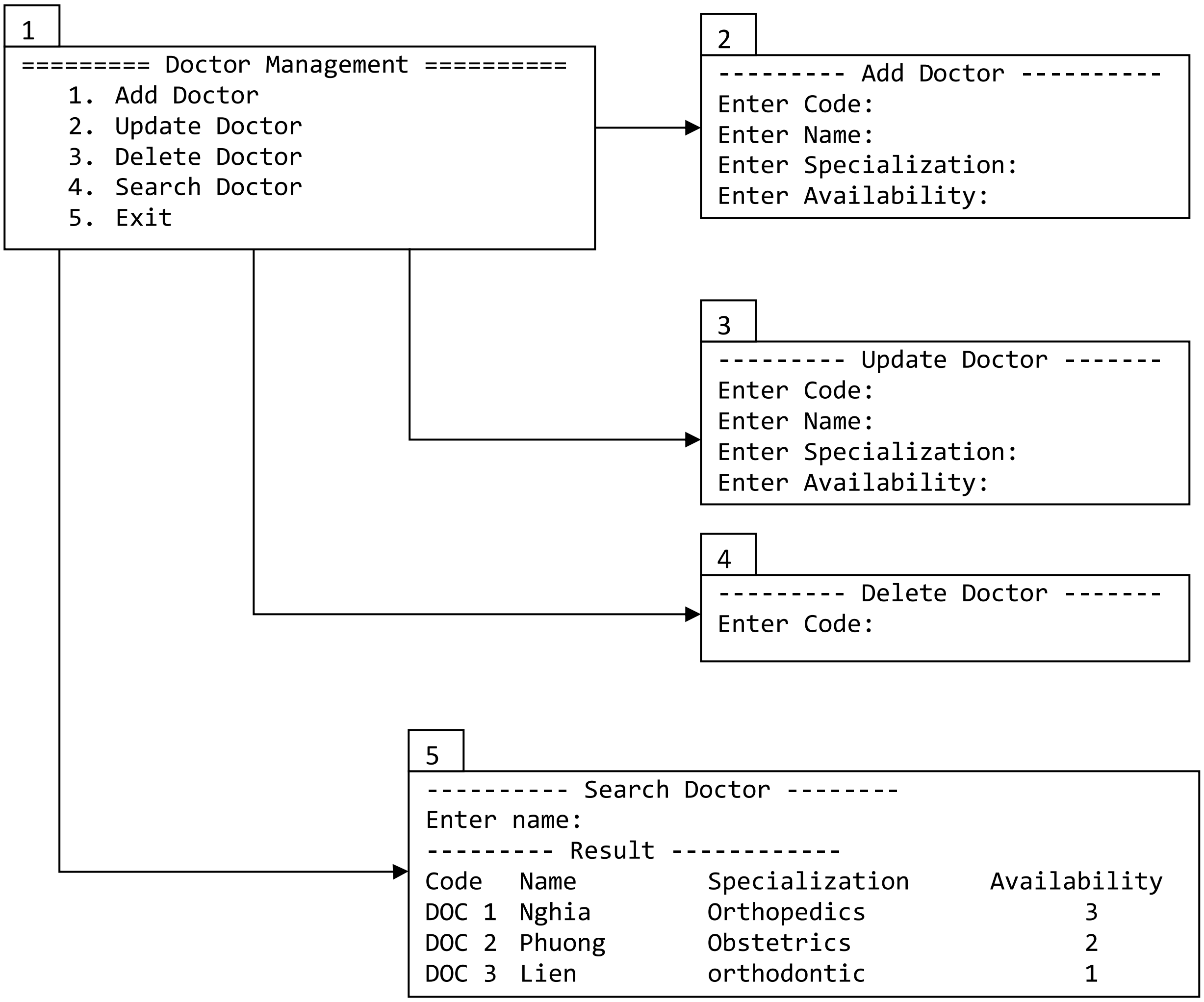
- User runs the program. The program prompts users input Data.
- Auto next **Function2**.

Function 2: Perform function based on the selected option.

- Option 1: Add Doctor
  - Require to enter task information including “code, name, specialization, availability”.
  - Check the valid data following conditions:
    - Code is not null or duplicate in the DB.
  - Add Doctor to the program.
  - Return to the main screen.
- Option 2:Update Doctor

    - Require to enter Code (id) and the data to be modified.
    - Check the valid data with the conditions below:
        - Code (id) must exist in the DB.
    - Update Doctor in the program.
    - Return to the main screen.
- Option 3: Delete Doctor
    - The requirement to enter Code (id).
    - Check valid data with the conditions below:
        - Code(id) must exist in the DB.
    - Delete the doctor information.
    - Return to the main screen.
- Option 4: Search Doctor.
    - Require to enter strings to search.
    - Search data and display on the screen
    - Return to the main screen.
- Option 5: Exit the program.

Expectation of User interface:



```
========= Doctor Management ==========
   1. Add Doctor
   2. Update Doctor
   3. Delete Doctor
   4. Search Doctor
   5. Exit
```

```
--------- Add Doctor ----------
Enter Code:
Enter Name:
Enter Specialization:
Enter Availability:
```

```
--------- Update Doctor -------
Enter Code:
Enter Name:
Enter Specialization:
Enter Availability:
```

```
--------- Delete Doctor -------
Enter Code:
```

```
---------- Search Doctor --------
Enter name:
--------- Result ------------
Code   Name         Specialization      Availability
DOC 1  Nghia        Orthopedics                3
DOC 2  Phuong       Obstetrics                 2
DOC 3  Lien         orthodontic                1
```

Guidelines

Student must implement methods

- addDoctor

- updateDoctor
- deleteDoctor
- searchDoctor

in startup code.

Suggestion:

- Class **DoctorHash** contains **adding, editing, deleting and searching** functions with doctor information.
- write **checkAvailability** function to check the validation of Availability, know that Availability> = 0.
- Use the Throw function to output the exceptions in the functional requirements
- Use the Put function to add an element into HashMap
- Use the update function to replace or put HashMap
- Use the delete function to remove an element in the HashMap.
- Use the value of HashMap to get the values and then use the Foreach Code, Name .... then use the “contains function” to find the value.

Function 1: Add a doctor

- ○ Implement function: public boolean add Doctor (Doctor Doctor) throws Exception
  - Input:
    - ➢ doctor: information of doctor.
  - Return Value:
    - ➢ Status doctor added.
    - ➢ Exception ("Database does not exist") case was null HashMap information
    - ➢ Exception("Doctor code [Code] is duplicate") if identical code.
    - ➢ Exception("Data does not exist") If the parameters are null doctor.

Lists 2: Edit the information of doctors.

- ○ Implement the function: public Boolean updateDoctor (Doctor Doctor) throws Exception
  - Input:
    - ➢ doctor: information of doctors.
  - Return Value:
    - ➢ Status doctor fix.
    - ➢ Exception ("Database does not exist") case was null HashMap information
    - ➢ Exception("Data doesn't exist") If the parameters are null doctor.
    - ➢ Exception("Doctor code doesn’t exist") if no code exists.

Function 3: Delete the information of doctors.

- ○ Implement the function: public Boolean deleteDoctor (Doctor Doctor) throws Exception

- Input:
  - doctor: information doctors.
- Return Value:
  - Status doctor fixed.
  - Exception ("Database does not exist") case was null HashMap information
  - Exception("Data doesn't exist") If the parameters are null doctor.
  - Exception("Doctor code doesn’t exist") if no code exists

Function 4: Find information of doctors.

- Implement the function: public HashMap<String, Doctor> searchDoctor (String input) throws Exception
  - Input:
    - input: information of doctors.
  - Return value:
    - HashMap<String, Doctor> list of doctors found.
    - Exception ("Database does not exist") case was null HashMap information