LAB 211 Assignment

Type:	**Long Assignment**
Code:	**J1.L.P0014**
LOC:	**500**
Slot(s):	**N/A**

Title

Hospital Management

Background

You were asked to build a Hospital management application. The program must have the basic functions of adding, deleting and editing departments, doctors, patients, record medical examination and treatment, medical examination results, ... The doctor.dat file is used to store information of all doctor in the hospital. The department.dat file has stored department information. **The relational model is shown as follows: one doctor belongs to only one department; one department can have many doctors.**

Program Specifications

Build a management program. With the following basic functions

0. Build your data structure
1. Show information
   - 1.1 Show doctor list
   - 1.2 Show department list
2. Add new
   - 2.1 Add new doctor
   - 2.2 Add new department
3. Update information
   - 3.1 Update doctor
   - 3.2 Update department
4. Delete
   - 4.1 Delete doctor
   - 4.2 Delete department
5. Search information
   - 5.1 Search doctor by name
   - 5.2 Search department by ID
6. Store data to file

Others- Quit

Each menu choice should invoke an appropriate function to perform the selected menu item. Your program must display the menu after each task and wait for the user to select another option until the user chooses to quit the program.

Each department has the following information: departmentID, name, createDate, lastUpdateDate

Each doctor has the properties such that doctorID, name, sex, address, departmentID, createDate, lastUpdateDate

Features:

This system contains the following functions:

Display a menu and ask users to select an option.

- **Function 0: Build the data structure**: **100LOC**
  - Classes, abstract classes, Interfaces.
  - departmentID, doctorID can not change after create.
  - Must implement the polymorphism properties of object-oriented programming.

- **Function 1: Show information**
  - **Show doctor list-25LOC**
    - ✓ Show all data in the doctor.dat file into the screen.
  - **Show department list- 25LOC**
    - ✓ Show all data in the depertment.dat file into the screen.

- **Function 2:**
  - **Add new doctor- 500 LOC**
    - ✓ Create a submenu that allows the admin add new doctor.
    - ✓ createDate is current date when this doctor is added. **It can not change after created**
    - ✓ lastUpdateDate is empty.
    - ✓ **Remember that the constraints must be checked: doctorID can not duplicate, departmentID must be existed in the department list.**
    - ✓ Add the new doctor to collection.
    - ✓ Ask to continuous create new doctor or go back to the main menu.
  - **Add new department- 50 LOC**
    - ✓ Create a submenu that allows the admin add new department.
    - ✓ createDate is current date when this department is created. **It can not change after created**
    - ✓ lastUpdateDate is empty.
    - ✓ **Remember that the constraints must be checked: departmentID can not duplicate**
    - ✓ Add the new department to collection.
    - ✓ Ask to continuous create new department or go back to the main menu
- **Function 3: Update information**
  - **Update doctor- 50LOC**
    - ✓ Require enter the doctorID
    - ✓ If doctorID does not exist, the notification “Doctor does not exist”. Otherwise, admin can start input new information of doctor and update.
    - ✓ If new information is blank, then not change old information.
    - ✓ **Remember new information must be validated and departmentID must be existed in the department list**
    - ✓ lastUpdateDate is current date.
    - ✓ System must print out the result of the updating.
    - ✓ After updating, the program returns to the main screen.
  - **Update department- 50LOC**
    - ✓ Require enter the departmentID
    - ✓ If departmentID does not exist, the notification “Department does not exist”. Otherwise, admin can start input new information of department and update.
    - ✓ If new information is blank, then not change old information.
    - ✓ **Remember new information must be validated.**
    - ✓ lastUpdateDate is current date.

    - System must print out the result of the updating.
    - After updating, the program returns to the main screen.

- **Function 4: Delete**
  - **Delete doctor- 25LOC**
    - Admin can delete doctor in the hospital.
    - Before the delete system must show confirm message.
    - Show the result of the delete: success or fail.
    - After delete, the program returns to the main screen.
  - **Delete Department- 25LOC**
    - Admin can delete doctor in the hospital.
    - Before the delete system must show confirm message.
    - **Department can not delete if there is a doctor who existed in this department.**
    - Show the result of the delete: success or fail.
    - After delete, the program returns to the main screen
- **Function 5: Search information**
  - **Search doctor by name- 25LOC**
    - The admin enters search text.
    - The system searches the doctor list, and returns all doctor that has name contain the search text.
    - Show result list: all information of doctors.
    - The program returns to the main screen.
  - **Search department by ID- 25LOC**
    - The admin enters search text.
    - The system searches the department list, and returns a department that has departmentID equal the input text.
    - Show information of department.
    - The program returns to the main screen.
- **Function 6: Store data to file-50LOC**
  - Store list doctor to doctor.dat file.
  - Store list department to department.dat file.

- The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- The lecturer will explain the requirement only once on the first slot of the assignment.