Term Project Description
"*Hospital Management System*"

Submission Due date: Monday, June 6th 2022 — 10 Marks

Hospital Management System is usually designed and programmed to deal with day-to-day operations and management of the hospital activities. In this project, you are asked to create a hospital management system. The hospital is divided into many departments each department has a department ID, name and a manager (Doctor of a high rank) and a list of doctors that work for this department. The hospital also saves the doctors, nurses, and receptionists information as follow (staff ID, name, phone number, department, specialization).

#Note only doctors and nurses are assigned to departments and specialization.

Patients are asked to fill in their name, phone number, address, age, gender and later the system gives them ID. Inpatient will be given a room, while outpatient doesn't need a room. Each room will carry number of beds and has a nurser responsible for it. Once the patient (either inpatient or outpatient) receives and finish the medical service, the receptionists are responsible for generating a bill for the patient. Bills are to contain (bill number, patient name and ID, list of medical services provided, and total amount). The total amount of a bill is the sum of the price of each service provided to the patient.

The system can be entered using a valid username and password. It is accessible either by an administrator or receptionist.

Doctors should be able to:

- Prescribe medicine to a patient
- Check reports.
- Draw salary.

Nurses should be able to:

- Draw salary.
- Be assigned new rooms.

Receptionists are responsible for:

- Checking whether a room is available or not.
- Book the room for a patient.
- Generate bills.
- Draw salary.

Your system should allow the user to:

1. Add staff member: Doctor, nurse, or receptionist.
2. Delete staff member: Doctor, nurse, or receptionist.
3. Update staff member: Doctor, nurse, or receptionist.
4. Search for a staff member using ID or name: Doctor, nurse, or receptionist.
5. Get all the rooms that a certain nurse is responsible for.
6. Add a new department.
7. Delete department.
8. Enroll staff (doctors or nurses) to a certain department.
9. Delete doctors from a certain department.
10. Change the responsible doctor for a certain department.
11. Check for the physician (doctor) who medicate a specific patient.
12. Check the room number that a patient resides.
13. Search to know the current status of each room
14. Registering the patients, storing their details into the system.
15. Give a unique id for every patient automatically.
16. Check the availability of a doctor.

General Constraints:

- All your C++ code files (source and header files) should be in one folder named "SRC".
- Just ONE of the team members must submit the project deliverables on OpenLMS.

Grading Rubrics:

- Specifications: The program meets all the requirements (35%).
- Execution: The program run successfully (10%).
- Code Design: The program applies the basic concepts of OOP correctly, such as encapsulation, abstraction, inheritance, and polymorphism (20%)
- Readability: The code is well organized and easy to follow (10%).
- Documentation: The code is well documented and clearly explained (15%).
- Report: The report is well-written and meets all the requirements (10%).
- **BONUS**: Using files to save data before exiting, and load data at program start (20%)

Project Deliverables:

Each project should submit on OpenLMS a zip file that includes:

1. A README.txt file, corresponding to the "user's manual".

This file should contain: first team member names and IDs, then describes the package in a few paragraphs, how to run and use the program.

2. A "SRC" folder contains the C++ code files (source files and header files)

3. A report with the following:
   a) Team member names and IDs
   b) Application description
   c) Program structure (description of each class)
   d) Explanation of the OOP concepts applied in the project.
   e) Workload distribution within the team (the role of each member)
   f) Some test cases.
   g) Sample input and output screens.
   h) List of references (if any)

Teams:

Work in groups of 4-6 students (mixing between groups is allowed).

Plagiarism:

Plagiarism is a serious academic offence and students who share code with others or use any source code from the internet, will lose the project grade.

Best of luck!