Birzeit University

Department of Electrical & Computer Engineering

Summer Semester, 2023/2024

ENCS3130 Linux Laboratory

Python Project – Medical Test Management System

Objective:

Develop a system to efficiently store, manage, and retrieve medical test data for individual patients. This system acts as a basic patient record management system focusing on test results. The system will include features for adding new test results, updating existing records, deleting outdated or incorrect entries, and retrieving specific test results based on patient ID or test type. The system should be built using object-oriented Python code. Define a class for patients with appropriate attributes and methods. Store all patients in a dictionary, where the patient ID is used as the key.

File Format:

The medical test will be stored in text file named midecalRecord. Each line in the text file represents a single medical test. The representation will include fields for:

- Patient ID (integer: 7 digits)
- Test name (string - consider a fixed length)
- Test date and time (format like YYYY-MM-DD hh:mm)
- Numeric result value
- Results unit
- Status (string - fixed length, e.g., "Pending", "Completed", "Reviewed")
- Results date and time (format like YYYY-MM-DD hh:mm) only if the status is Completed,

For example, the following file has two medical tests:

```
1300500: RBC, 2024-01-01 14:10, 13.4, mg/dL, completed, 2024-01-01 15:30
1300500: RBC, 2024-03-01 05:20, 13.5, mg/dL, completed, 2024-03-01 05:30
1300511: LDL, 2024-03-2 07:30, 110, mg/dL, pending
1300520: systole, 2024-03-4 04:40, 150, mm Hg, pending
```

Medical Tests:

There is another file that store the information of each test. Below is the list of medical tests with their normal range and nominal test duration. The turnaround time is the time it takes from when a medical test is added until the test results are reported and the test status is changed to completed. The turnaround time in the file should be written according to the following format: DD-hh-mm (i.e., if a test needs 2 days, 5 hours, and 10 minutes to be completed and results are obtained, then the test turnaround time is 02-05-10):

1. Name: Hemoglobin (Hgb); Range: > 13.8, < 17.2; Unit: g/dL, 00-03-04
2. Name: Blood Glucose Test (BGT); Range: > 70, < 99; Unit: mg/dL, 00-12-06
3. Name: LDL Cholesterol Low-Density Lipoprotein (LDL); Range: < 100; Unit: mg/dL, 00-17-06

4. Name: Systolic Blood Pressure (systole); Range: < 120; Unit: mm Hg, 00-08-04
5. Name: Diastolic Blood Pressure (diastole); Range: < 80; Unit: mm Hg, 00-10-00

The format of the second file named medicalTest contains four information about each test: name, range, unit.

System Functionality:

Develop a text-based menu that allows users to:

- Add new medical test: the system will allow user to insert new type of medical test and save it in the medicalTest file. The system will check the validity of the input data.
- Add a new medical test record: the system will allow the user to store a new medical test with the required data. The system will check the validity of the input data.
- Update patient records including all fields.
- Update medical tests in the medicalTest file.
- Filter medical tests: The system will retrieve and display tests based on one or more of the following criteria (the user might select a combination of these criteria, such as generating a report about a specific test within a period, or abnormal tests of a specific patient within a period, or tests with execution time between 10 and 20 minutes):
    - Patient ID,
    - Test Name,
    - Abnormal tests,
    - Test added to the system within a specific period (start and end dates),
    - Test status,
    - Test turnaround time within a period (minimum and maximum turnaround time),
- Generate textual summary reports: use the same search and filter test options to generate descriptive statistics for the filtered records including:
    - Minimum test values, maximum test value, average test value,
    - Minimum turnaround time, maximum turnaround time, average turnaround time.
- Export medical records to a comma separated file,
- Import medical records from a comma separated file.

In addition to the above functionality, the system has capability for:

- **Error Handling:** Implement error handling for invalid file name, searching for non-existent tests, searching for non-existent patient, ….
- **Data Validation:** Validate user input to ensure proper data types (e.g., integers for ID, valid dates) and handle potential errors.

Submission:

Please submit the following:

1. Code
2. At least 2 testing examples.

Notes:

- Write the code for the python program to satisfy the requirements described above and name the file as MedicalTest.
- Make sure your code is clean and well indented; variables have meaningful names, etc.
- Make sure your code has enough comments inserted to add clarity.
- Work in groups of at most two students
- Deadline: **Monday,** 22 **August, 2024 at 11:59pm.** Please submit your project (code + test cases) through Ritaj as a reply to this message.
- This project is per group effort: instances of cheating will result in you failing the course.

Grading Policy:

Item	Points
Add new medical test	5
Add new medical record	5
Update medical tests	5
Update patient records	5
Filter medical tests using single criterion	5
Filter medical tests using two criteria (other than test period and turnaround period)	5
Filter medical tests based on test period	5
Filter medical tests based on test turnaround period	5
Generate textual summary reports - test values	5
Generate textual summary reports - test turnaround time	5
Export medical records to a comma separated file	5
Import medical records from a comma separated file	5
Error Handling	10
Code structure and Comments	10
Discussion	20
Total	100