Objectives

1. Develop a simple and easy-to-use User Interface (UI) for the clinical data warehouse.
2. Use a version control system to store and maintain the program.
3. Follow good coding practices to ensure that the code is readable and easy to maintain.

Description

You are working as an Information Technology staff for a local hospital. As the hospital continues to grow, the clinical data warehouse you developed is used by more users, who do not necessarily have the technical expertise to directly start and interact with it through command lines on a terminal. Therefore, you are asked to develop a simple and easy-to-use UI, through which users can interact with your program to add, remove, retrieve patient information, or perform other tasks. In addition, as the program grows, you are also asked to properly store and maintain the program code using a version control system so that your team members can access and modify the code.

Developing a simple UI

If you have not used Python to develop UI before, you can use the TKinter package (https://docs.python.org/3/library/tkinter.html). However, you are free to use any other packages or frameworks to develop the UI, either as a stand-alone program or as a web application.

Essential functionalities through the UI

1. **Validating credentials:** when the program starts and the UI is shown to users, it should show a log-in page and prompt users to enter their username and password and click log in.
2. **Reading in existing patient information:** if a user's username and password match the records in the credential file, the UI should read in a file that contains existing patient information and show action buttons. The menu shown to a user should depend on the type of the user. For example, if the user that logged in is in the management or admin type, then the only two buttons shown should be "Generate key statistics" and "Exit". If the user that logged in is in the nurse or clinician type, then five buttons should be displayed to them: "Retrieve_patient", "Add_patient", "Remove_patient", "Count_visits" and "Exit".
3. **Retrieving, adding, and removing patient information:** these functionalities remain the same as in previous assignments. After the user clicks on a button, the UI should prompt the user to enter information corresponding to the action required. For example, if the user clicks on the "Retrieve_patient" button, the UI should then prompt the user to enter a Patient_ID and display the patient information in an easily readable format. After a user performs the action and is shown results, the user should be returned to the menu where they can select one of the actions again.
4. **Generating key statistics:** If a user logged in is in the management or admin role, then the only action the user can perform is to generate key statistics. After the user clicks on this button, your program should either display key statistics on the screen through the UI, or write and generate key statistics to a file and alert the user about the file.
5. **Exiting the program** from UI by clicking on a "Exit" button.

Table 1: Four user roles

Value	Description

“admin”	Administrative staff. They can access non-PHI but not patient-related information. Therefore, the only possible action is count_visits.
“clinician”	Healthcare professionals. They can access all patient information and perform all actions.
“nurse”	Healthcare professionals. They can access all patient information and perform all actions.
“management”	Human resource managers who do not interact with patients. They cannot access PHI. The only action they can perform is to request key statistics from the program

Once a user can access patient information, your program should allow the interactions from Programming Assignment 2 & 3 (add_patient, remove_patient, retrieve_patient, count_visits). The interaction patterns should also remain the same. Note that the possible data and interactions for the user should be strictly based on their roles, as described above. For example, the program should never prompt a user with “admin” or “management” role to access PHI.

Generating usage statistics

Now that more users are using the software, the IT department plans to track usage of the software so that they have more insights into which types of users are frequently using the software and for what types of tasks. After each user finishes using the software, your program should store the information such as the username and role for the user, the action the user performed, and time of log-in. This information should be stored in a file and updated after a user has used the program.

Writing patient information to file

After users perform actions such as removing or adding patient information, your program should store these changes to existing patient information and update the patient information file accordingly.

Patient Data (same as in Programming Assignment 2 & 3)

Data element	Type	Description	Example
Patient_ID	String	A unique identifier of a patient	‘238610’
Visit_ID	String	A unique identifier of a patient’s visit. A patient may have multiple visits, therefore multiple Visit_ID.	‘729471’
Visit_time	String	The date time for the patient visit.	‘’2024-02-24”
Visit_department	String	The department that the patient visited. Each visit may have a different department.	‘ED’
Gender	String	A patient’s gender will be recorded for each visit.	‘Female’, ‘Male’, ‘Non-binary’

Race	String	A patient's race will be recorded for each visit.	'White', 'Black', 'Asian', 'Pacific islanders', 'Native Americans', 'Unknown'
Age	Integer	For simplicity, we assume age will be an integer. Each visit will have an age recorded.	78
Ethnicity	String	A patient's ethnicity will be recorded for each visit.	'Hispanic', 'Non-Hispanic', 'Other', 'Unknown'
Insurance	String	A patient's insurance will be recorded for each visit	'Medicare', 'Medicaid', 'None', 'Unknown'
Zip code	String	A patient's zip code will be recorded for each visit	'53211'
Chief complaint	String	A patient's major reason for the visit	"Chest pain"
Note_ID	String	A unique identifier to a patient' s clinical note	"38471937"
Note_type	String	The type of the clinical note	"Oncology", "Discharge"

Program implementation requirements

Your program for this assignment should be based on your code for Programming Assignment 2 & 3. Building upon the classes you already designed and implemented in Programming Assignment 2 & 3, you should also add another class to model the UI application.

Input files

Input file 1: credential file

The file Project_credentials will contain usernames, passwords, and user roles of authorized users. Your program should read in this file first so that it can check users' credentials. The possible values included in the file are described on Page 1.

Input file 2: existing patient file

This is the file you used from Programming Assignment 2 & 3. Your program should also read it in so that it has all existing patient information.

The possible input from users include:

1. `Add_patient`: this adds a new patient to the program. Your program should then prompt the user to enter Patient_ID. If the Patient_ID already exists, you should prompt the user to enter the Visit_time and randomly create a unique visit_ID and add the information to the patient records.
2. `Remove_patient`: this removes all information about the patient from your program. Your program should then prompt the user enter Patient_ID. If the Patient_ID does not exist, your

program should print an alert to the user to show this. Otherwise, remove all records associated with the Patient_ID.

3. `Retrieve_patient`: your program should then prompt the user to a Patient_ID. Your program should search if this Patient_ID exists. If not, show an alert. If yes, ask the user what information is needed and then display the information on the UI.
4. `Count_visits`: this function allows users to check how many patients were seen at a given date. Your program should prompt the user to enter a date following the format "yyyy-mm-dd" and then count the number of total visits (if a patient has multiple visits, count them all) on that day and display to the patient.

Grading (400 points in total)

The submission will be graded based on the following criteria. The assignment will be 400 points in total.

Program (100 points)

20 points: I can start your program through terminal and interact with the program through a UI.

50 points: For successfully implementing 5 essential actions (retrieve patient, add patient, remove patient, count visits, generate key statistics).

30 points: For successfully implementing the log in function and verifying user credentials.

Output files (50 points)

50 points: There are two expected output files. The first one is the patient information file. The final file should be updated to reflect the patient information after changes made by the users such as adding new patients and removing patients. The final file should follow the format of the original patient information file I provided on Canvas. The second output file is the usage statistics file. This file should record each log in made by users. This file should record the user's username, role, log in time, and actions performed. If a user fails to login by providing incorrect user credentials, this file should also record this event and mark it as failed log in attempt.

Coding style (50 points)

50 points: Your program should be easy to read. You should follow the general coding styles introduced in class, such as good naming practices, small and concise functions, etc.

Design (100 points)

30 points: The UI for your program should be simple, easy to use, and intuitive. I should be able to interact with the UI to perform the tasks easily.

30 points: You should submit a UML diagram that illustrates the design of your classes. The granularity of the diagram is up to you, but at least it should clearly show the classes, relationships among the classes, and attributes and methods of each class. The UML diagram should match the implementation of your program.

20 points: Your program should be structured into multiple .py files and a main file (main.py) that imports modules from other files. Putting all codes into one file will lose points. The structure of these files should also be carefully thought out, e.g., patients.py should contain patient-related functions. However, the main file is the one that I will execute to start the program. All other files will not be directly run but should be imported into the main file to be used.

20 points: Your program should implement at least 4 classes using object-oriented programming, including the User class and a class for modeling the UI application. Each class has a set of attributes and methods and an appropriate constructor.

Documentation (50 points)

50 points: you should write a ReadMe file that explains: (1) how to run your file, such as the command used, and packages required; (2) a short description of your program, for other programmers who are not familiar with it; (3) other information that you think is important for users and programmers to be aware of. There are many good resources for how to write a ReadMe file online, e.g, https://dev.to/merlos/how-to-write-a-good-readme-bog.

Version control (50 points)

20 points: You use GitHub to store and maintain the code base. Your submission should include a link to your GitHub repository that contains the code for this assignment. You need to make the repository a public repository so that I can view and download it. **You should show the code, not just the zip file on GitHub.**

20 points: Your GitHub repository structure should be clear and easy to understand. The repository should show your ReadMe on the front page and store the UML diagram and source code in the repository.

10 points: Your GitHub repository should show at least one commit that you made to make changes of your code. Simply uploading your code to a GitHub repository does not count. You have to make a commit with a commit message to make a modification to your code. Any changes are acceptable. This is intended to test if you can use the basic functionality of GitHub for version control.

Submission

Your submission should be a zip file with the name: **FIRSTNAME_LASTNAME_Project.zip**. The zip file should contain the python files (you can use one file or multiple ones, depending on how you design your program) and a PDF of the UML diagram. For consideration for bonus points, please add the link to your public GitHub repository in the submission comment so that I can access.

The assignment is due: 5/13/2024, 11:59pm CT

Late submission

If you submit your assignment after the deadline but within 24 hours, only 70% of the overall points (140 points) will be considered for full. After 24 hours, all submissions will receive 0 points.

If you have personal or medical conditions that may affect submitting the assignment on time, please reach out to me as soon as possible to communicate an alternative timeline for submission.

Plagiarism

Discussion with your classmates is encouraged, but you should not directly duplicate their codes. If direct duplication of codes is found, you will receive no points in this assignment. You can use AI such as ChatGPT for troubleshooting or consulting, but you cannot directly copy and paste code generated from these tools. If you refer to code from external sources such as StackOverflow, please add the link to the comment of your program.