In this assignment we will compare performance of different operations on online reservation system.

You are developing an online reservation system for getting vaccination following the Object Oriented Design, and also with a focus on the system performance, for an emerging pharmacy DWT (CVS) in the market. Following are some of the requirements for you to start your design:

1. The pharmacy offers a variety of common vaccinations, such as flu, covid, chicken pox, whooping cough, etc.
2. Each type of vaccination should specify its specifications in terms of the target age groups, allergens, etc.
3. Some vaccinations should be offered in multiple doses with a certain period of time in between doses.
4. Different types of vaccinations may have different insurance eligibility, depending on the patient's age as well.
5. The system should keep a record of a patient's prior vaccination status.
6. The system should allow a patient to reserve a timeslot for a vaccination based on:
    1. The availability of the vaccination,
    2. The selected location,
    3. The selected time,
    4. The vaccination history and status,
    5. Balance should be estimated based on information from the respective insurance.

Following are tasks to follow:

1. Describe the Software Performance Engineering (SPE) process based on this system, without the model-based evaluation part, which should be the focus of 345. **Refer to lecture 'Software Performance for Object-Oriented Design' discussed on Week 11.**
2. Compare the different Java I/O APIs with regards to their performance with the provided CSV files of 10M patients (patients.csv) with vaccination records

(vaccination-history.csv) for each of the 10M patients (the two CSV files are linked using patient IDs which are primary keys:), read from input and write to an output file with a different file name. Measure the processing time, resource utilization as the number of processed records increase

    1. Allergen
    2. Current conditions
    3. Vaccination name
    4. Dose # if multiple
    5. Date of vaccination for each dose
    6. Name, SSN, DOB, Email, Phone,
    7. Insurance Company, Insurance Member ID
    8. Patient history
    9. Vaccination history

3. Load data into memory. Dataset: https://stevens0-my.sharepoint.com/:f:/g/personal/ealomar_stevens_edu/EuMQ7HBZFHNKpaXUciFd04YBEzb3Tw0QuvPIcKQSQGi0lw?e=GOoo5VLinks to an external site.

1. The provided files (patients.csv and vaccination-history.csv) are large and will most likely exceed your laptop's physical memory, resulting in an out-of-memory runtime error.

1. One workaround is to sequentially load the files in chunks. For example, if loading 10 million lines of patient.csv is too much, consider loading it in 10 chunks, 1M lines at a time.

2. For the Vaccination files with 290 million lines, you can do the same. But you need to increase the number of chunks, since it is a much larger file.

3. Compare inheritance based vs. composition based implementation

    1. Program implementation
    2. Track the data and draw diagrams, i.e. as the number of records increases, track the execution time of each program
    3. Reason about the findings/observations

4. Fill out the survey: https://stevens.qualtrics.com/jfe/form/SV_0wegvFLnsvwTQ7sLinks to an external site.

Notes:

- When uploading the results of your test run, make sure to include a snapshot of your code and test results within IDE.
- You can continue updating your work and submitting newer versions before the deadline. Only the last version before the deadline will be graded.
- If 100,000 records is taking about 0 sec in multiple steps then increase the number of records to 1,000,000.

Submission:

- Java source code named **OnlineReservationSystem.java**
- Screenshot of your test run as pdf or image (jpg, jpeg, png)
- Write-up to describe approach and comparison named **Report.pdf**