ISYS622
Assignment 3
(50 points)

Using the Mercy Pharmacy case, implement a data access layer for the new pharmacy system by completing the following:

1) Project Setup
    a. Extract the assign3.zip folder to a new folder inside your eclipse workspace.
    b. Import the assignment 3 Maven project into your workspace as instructed in class.
    c. Run the Maven test goal to execute your schema creation scripts and set up your database for development (mercy-pharm-create.sql, mercy-west-ehr.sql).
    d. All screenshots and other documentation are included in the site folder of the project.

2) Project Implementation

    a. Use the screen shots of the pharmacy system to design your data access API.
        i. You must provide some facility for getting and/or modifying the data used on every part of the screen. We will discuss this in detail in class.
        ii. You must utilize the given java object model fully. We will review in detail in class.
        iii. You must utilize the Repository pattern in your solution. Note that several repository interfaces are provided for your convenience but you must create the concrete implementations. Again, we will review these in detail in class.
        iv. Hints:
            1. Note that a Mercy East patient appears on the screenshots provided. As a result, you must provide some facility for loading Mercy East EHR data (as a proof of concept to convince the board). Mercy East data comes in JSON format and not from an RDBMS like Mercy West. A JsonUtil class has been provided to help you load that data into Record objects. However, you must use the repository pattern to hide the details of this process from the users of your data access API.
            2. Consider creating a DBHelper class to open and close database connections in a central location to avoid having to rewrite the same code in all of your repositories.
    b. Using the provided java code files, implement your test cases. Note that all methods with the exception of constructors, getters and setters must have one or more corresponding test cases.
        i. Hints:
            1. Review all of the java code before updating your API.
            2. Your test cases should instantiate your concrete repository objects and call all of their public methods to read and write data to the different data sources.

3. Consider writing your test cases before writing any concrete Repository implementations. You will have to imagine the names of these classes as they haven’t been created yet but you already know the method signatures. For example, you might write something like the following:

```
PatientRepository repository = new JdbcPatientRepository();
List<Patient> patients = repository.get(“Matt”, “%”);
assertEquals(1, patients.size());
```

3) Project Testing
   a. Use the maven test target to verify that your code functions as expected before submitting through eCampus.

4) Project Packaging
   a. Create a zipped archive of your project. Be sure to only include the src folder, .classpath, .project and pom.xml files in the zipped archive. Name your archive *[your_UIN]*-A3 (ex. 1234567890-A3).

5) Submit the zipped archive as Assignment 3 in eCampus.