Quest 1 – DM-09 – Lecture Video Implementation (10 Experience Points)

Well met hero. Your first quest will be to implement the implementation that you saw in today's video (lecture). Follow the video first before you attempt Quest 2 below.

Quest 2 – DM-09 – HealthOS (60 Experience Points)

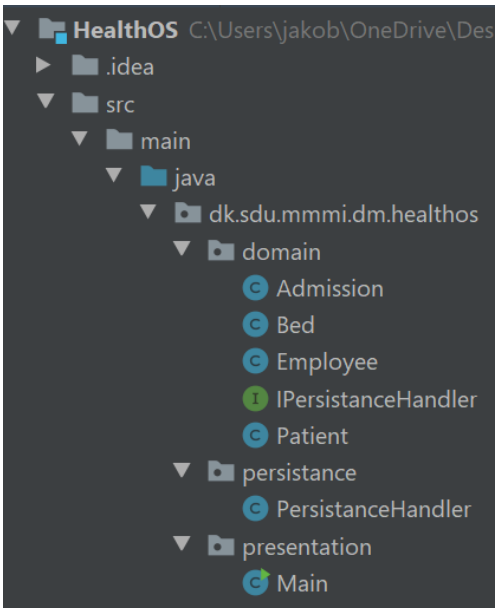
Last time we met, you helped create HealthOS to help battle the plague dispersed by the evil undead Beer Lich – Viscount Rexx Lestat. Unfortunately, he and his zombies have attacked and destroyed the hosting provider for PostgreSQL databases used in HealthOS. Fortunately, new allies have volunteered their MongoDB servers, and we are now in need of some brave adventures that can re-write HealthOS to work with MongoDB instead! Brave adventurer, please complete the application, and help stem the tide of the growing plague.

The zip file you have opened here holds the current progress on the MongoDB adaptation from a previous adventurer group (that was eaten by a Beholder). The zip file contains the following files:

1. Assignment.pdf (this file)
2. HealthOS_MongoDB.zip (Thanks goes to Simon Madsen for rewriting HealthOS to MongoDB)
3. HealthOS.sql

File 1 contains the days quests, while 2, HealthOS.zip, includes the current MongoDB progress on HealthOS. Also, 3, HealthOS.sql holds the previous relational database implementation that can no longer be used with MongoDB. Complete the application to complete this quest.

1. First, create a database, and the relevant collections in MongoDB.
2. The data from the previous installation was lost! Re-create the data from the HealthOS.sql file in the new MongoDB document structure.
3. Unzip HealthOS.zip and open it in IntelliJ IDEA. The contents of the file should look something like below:



4. Additional Quest Background Information: The HealthOS has been created with three modules: The Presentation layer (A console application), a Domain Layer, and finally, a persistence layer. Also, the MongoDB driver has already been imported using Maven. The three layers are built in a way where the presentation layer is dependent on the Domain Layer, and the Persistence Layer also depends on the Domain Layer. This is due to the interface, IPersistanceHandler, in the Domain Layer, which effectively inverts the normal dependency between the Domain Layer and the Persistence Layer. This phenomenon is aptly named Dependency Inversion. Also, the PersistanceHandler class is implemented using a Singleton Pattern and only returns the interface IPersistanceHandler.
5. Open the PersistanceHandler. Start by updating the Login Information for the database, and the database name that you created in step 1.
6. Explore the class. See that getEmployees() and getEmployee() is already implemented. Now you must implement the 10 methods below that are currently awaiting implementation.
7. Update the Main.java file in the presentation layer to support the methods createEmployee, createPatient, createBed, createAdmission, and finally deleteAdmission. Be aware that you might have to edit some classes in the domain layer.
8. Adventurer – you finally reached the end of the quest, and helped push back the Lich's plague. Enjoy, party (in safe corona isolation!!), and let everyone know in the #general chat how many experience points you accumulated today! Congratulations!