**CS3431 A16: Project Description**

**Phase 3**

**Application Development with a Complete Database**

Due Date: Friday, Oct. 7 at 11:59pm

Late Policy: 10% off until Saturday, Oct. 8 at 11:59pm

Teams: The project is done in the same teams of two as in the previous project phases.

Submission: One teammate will upload the p3.java file to myWPI using the Project Phase 3 link. Although you may develop the code using either Eclipse or IntelliJ, **the TA will compile your file on a CCC machine so make sure your file compiles and runs on the CCC machines**! Any comments or assumptions that you have, you can include them in a separate .doc or .pdf file. Also make sure to use the same interface, filename, and class name or we will not be able to grade your project!

Description:

For the first two phases of this term project, you were a database developer in charge of creating the database component that application developers would use. For this final phase of the project, you will get experience from the other side - you are now a Java application developer who will use and modify a database created by someone else for your use.

Download and use the provided p3start.sql file to do the final phase of the project. The sql file contains the Faulkner Hospital directory and map location tables. Review the design of this database before you begin because it may be different than the one you have been working on. For your information, the relational schema, the latest version of the Faulkner directory and the map images with hallway locations are included.

You are required to write a java program that accesses and modifies the database, and prints results on the screen. Your program will be the interface to perform some simple functionality over the database. Pay careful attention to the naming of your file and class so we can run them and give you credit for the project phase!

Create a program named “p3.java”. The program will always take two parameters, e.g., username and password to connect to the DB. (Pass them as parameters such that the TAs can easily set them as needed without recompilation).

1. When your program is executed without any additional arguments, e.g.,  
   > java p3 <username> <password>

Then the program should output the following options, and then terminate:

1 – Report Health Provider Information

2 – Report Health Service Information

3 – Report Path Information

4 – Update Health Service Information

1. When the program is executed with an argument 1 as follows:  
   > java p3 <username> <password> 1

The program enters the “Report Customer Information” mode. The program should print out the following line:

Enter Provider ID: <and wait for user’s input>

When the user enters the Provider ID, the program should execute a query of the provider and related tables and print on the screen the following health provider information as follows:

Health Provider Information

Provider ID: …

First Name: …

Last Name: …

Title: ... (acronyms - there may be more than one)

Office Location: … (location names - there may be more than one)

Then the program terminates.

1. When the program is executed with an argument 2 as follows:  
   > java p3 <username> <password> 2

The program now enters the “Report Health Service Information” mode. The program should print out the following line:

Enter Health Service Name: <and wait for user’s input>

When the user enters the health service name, the program should execute a query of the Services and related tables and print on the screen the following health service information and then terminate.

Health Service Information

Service Name: …

Health Type: …

Location: … (location name)

Floor: ... (FloorID)

1. When the program is executed with an argument 3 as follows:  
   > java p3 <username> <password> 3

The program now enters the “Path Information” mode. The program should print out the following line:

Enter Starting Location: <and wait for user’s input>

Enter Ending Location: <and wait for user’s input>

When the user enters the starting and ending locations, the program should display the following information for the **shortest** path (fewest intermediate locations) found in the PathContains table:

Start Location: …

End Location: …

Path ID for shortest path: …

Order LocationName FloorID

Order LocationName FloorID

Etc.

For example, the output should look like the following:

Start Location: Starbucks

End Location: Hillside Elevators 3

PathID for shortest path: 247

1 Starbucks F1

2 F1H1 F1

3 Hillside Elevators 1 F1

4 Hillside Elevators 3 F3

and then the program will exit.

If there is no path available in the database between the two locations the output should state “There is no path between these locations in the database” and then the program will exit.

1. When the program is executed with an argument 4 as follows:  
   > java p3 <username> <password> 4

The program now enters the “Update Health Service Information” mode. The program should print out the following line:

Enter Health Service Name: <and wait for user’s input>

Enter the new LocationID: <and wait for user’s input>

Then your program should update the location for the specified service name in the database. Now if you execute option 2 again, you should get the new location.