1. Unzip the projects into the assignments directory
2. Start IntelliJ and open the projects
3. Make sure SQL Server is running
   1. Go to “Services” on your Windows computer and restart SQL Server (MSSQLSERVER) if it is off
   2. Go to SQL Server Configuration Manager > SQL Server Network Configuration > Protocols for MSSQLSERVER and enable TCP/IP.
4. Check to make sure you have created a database with tables filled with data
5. Build the Project
   1. Verify that you have two separate modules
      1. Project Structure > Modules > New Module > Java
         1. Name the Module “dblib”
   2. Create a JAR from “dblib”
      1. Project Structure > Artifacts > Add JAR > From Modules with Dependencies
         1. Make sure to click “Include in Project Build”
   3. Add dependency to asg1 module
      1. Project Structure > Modules > Dependencies > Module Dependency > dblib > apply > okay
   4. Create Client Class in main src folder module
   5. Add SQLServerDriver Dependency
      1. Project Structure > Modules > Dependencies > JAR or Directories > C:\MicrosoftJDBCDriver9.2\mssql-jdbc-9.2.1.jre8.jar
      2. Set Scope to Runtime
6. Deploy the project/Run the Client
   1. Edit Configurations (top right of IntelliJ) > Add New Configuration > Application
      1. Name it ClientApp
      2. Set the Main Class to Client and specify the module as asg1
      3. Apply and click Okay
   2. Press Hammer Symbol (“Build Project”) CTRL+F9
   3. Run the Configuration (“Run ‘ClientApp’) SHIFT+F10
7. Explore the Application

**Client**

Follow the messages/instructions provided by the application client

1. Enter the database password (Should be ‘ism6236bo’ without the single quotes)
2. You will be prompted to enter a ‘Year’ and ‘Semester’ to display a list of available, non-full classes
   1. Valid ‘Semesters’ are Fall and Spring
3. Press L to List a student’s classes. You must then enter their student number and the year and semester you are interested in seeing.
4. Press R to register a student for a class. You must enter their student number, a course number, and section number.
   1. The business logic makes sure the student is valid, not already registered for the course and that the course is not at capacity.
5. Press Q to quit the client application.

**Registration**

**List ( String year, String semester )**

1. The first List method takes two String parameters, year and semester. It formats a SQL query as a String that is executed by a method from the Statement class and stores the values that are returned as a ResultSet. The query selects the Sections of classes that are not at capacity (the available classes) during a given year and semester. These values are then retrieved and stored as strings in a while loop, and formatted into a table. The values are *courseNo, sectionNo, room, days, and time*. If no SQL Exceptions occur, then this is what will be returned as a formatted String.

**List (int StuNo, String year, String semester )**

1. The second List method takes an int parameter, StuNo, and two String parameters, year and semester. It formats a SQL query as a String that is executed by a method from the Statement class and stores the values that are returned as a ResultSet. The query returns a students course schedule for a given semester and year. These values are then retrieved and stored as strings in a while loop, and formatted into a table. The values are *courseNo, sectionNo, room, days, and time*. If no SQL Exceptions occur, then this is what will be returned as a formatted String.

**Register (int StuNo, String CourseNo, String SecNo )**

1. The Register method takes an int parameter, StuNo, and two String parameters, CourseNo and SecNo. It creates three prepared SQL Statements. The first gets the TotalEnrolled and Capacity for the course number and section number that was passed through the method. This allows the method to check to make sure the course the student wishes to be added to isn’t already at capacity. The second prepared statement query gets the student numbers for the students registered for the course number and section number provided. This allows the method to compare these students with the student number passed through the method so that there is no double registration. The third prepared statement simply gets all students enrolled in the school. This lets the method check to make sure the student number passed is a valid student. After making these checks, the method updates the Section table by adding to the TotalEnrolled column and inserts a new row into the Enrollment table. The method returns the number of rows modified or created.