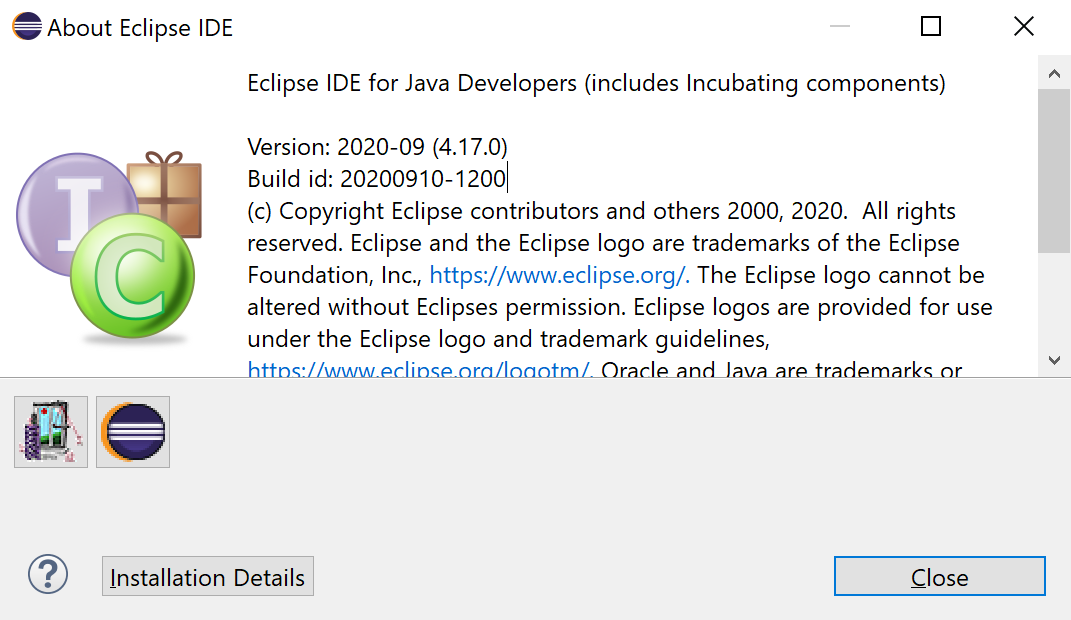
**README**

This application used MySQL for database design and Java for programming. There are some preparations need to be done before running this application.

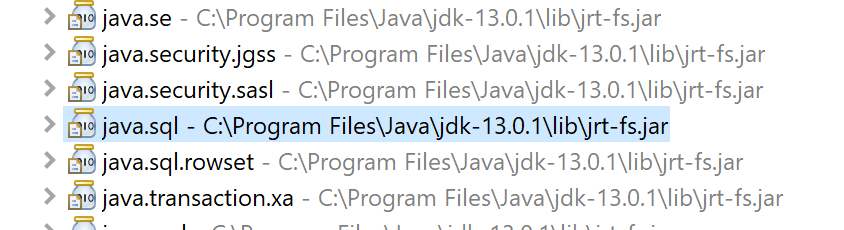
**1. Software**

This application was developed by *Eclipse IDE for Java Developers*, and the version is *2020-09 (4.17.0)*.



**2. Library**

A standard JRE System Library will serve, and it should contain *java.sql*.

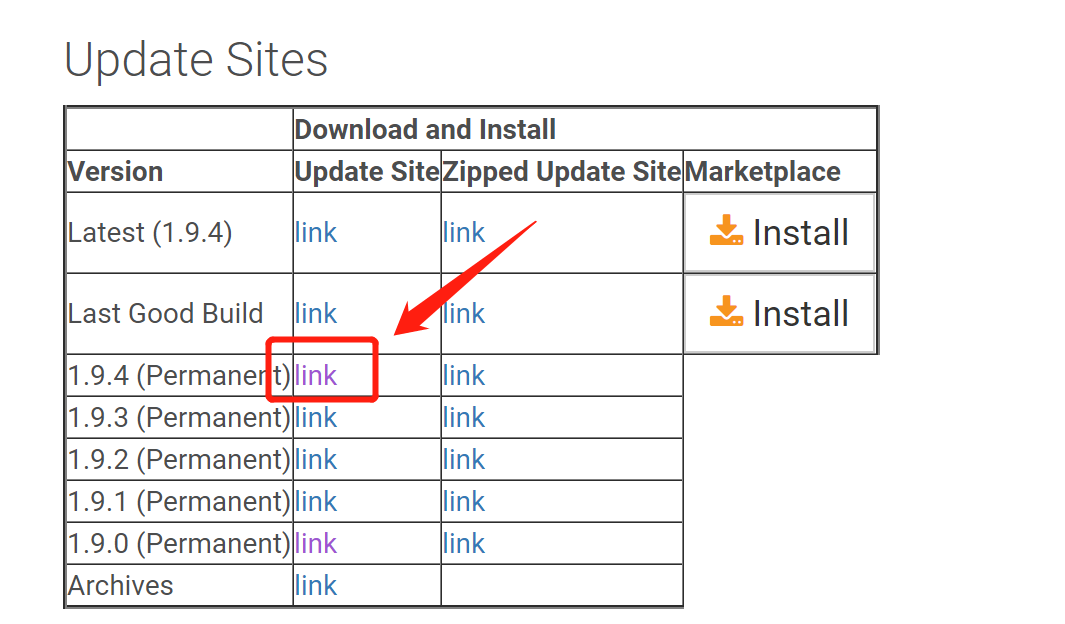


**3. Plugins**

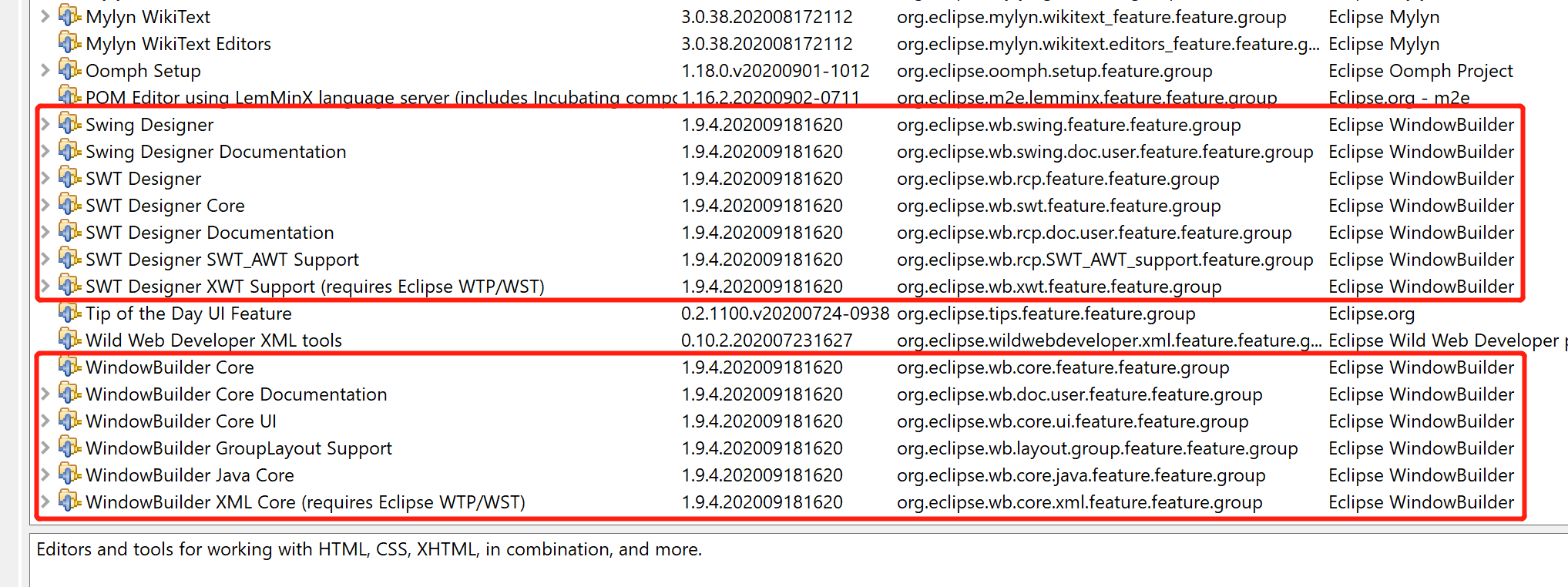
The front-end is based on *Swing* and *WindowBuilder*. Follow the links below:

<https://www.eclipse.org/windowbuilder/download.php>

Open the link and you will find a web page containing the *Update Sites* table below. Copy the link in the red rectangular and paste it into Eclipse *Install New Software* window (in the navbar, click help and then click install new software). If you need more detailed information or instructions on this installation, click the link in the red rectangular and you will get what you want.

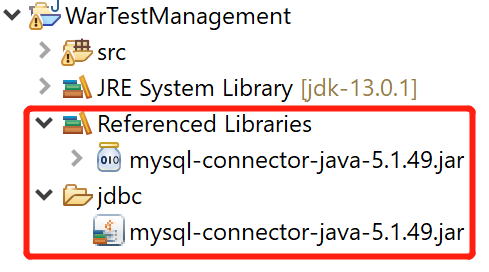


When Eclipse identify that installation link, it prompts you to select plugins. It is suggested to select all of them. When the installation finishes, restart Eclipse and check the installation details. Make sure you have those plugins in the red rectangular below installed.



**4. JDBC**

Create a folder called jdbc and add mysql-connector-java into it. Remember to add path.

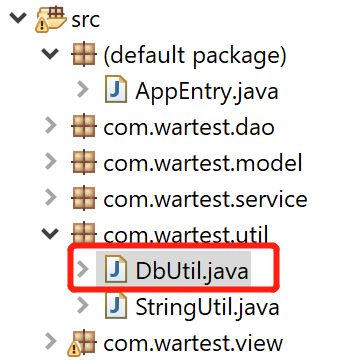


**5. Run Application**

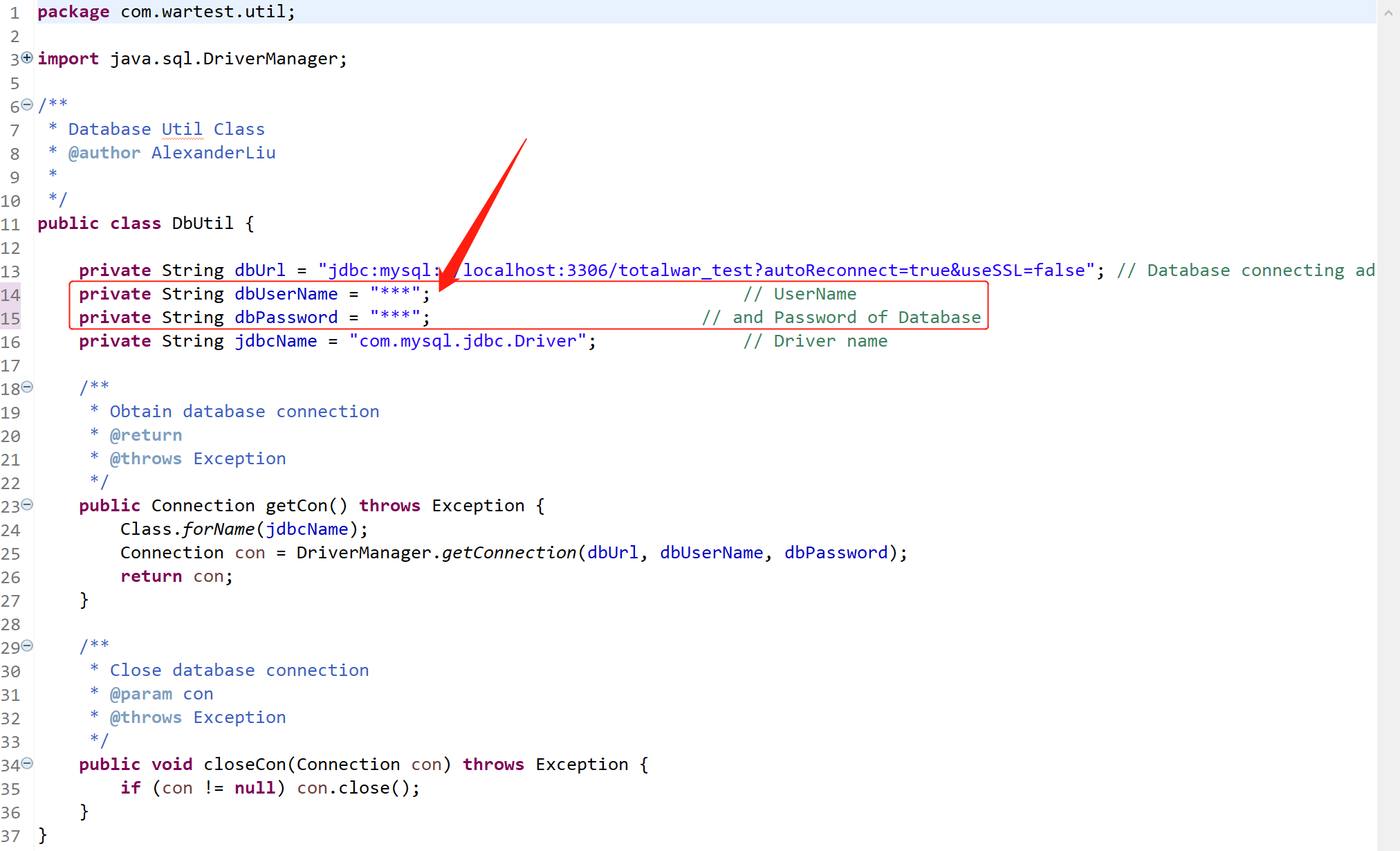
Create a Java Project. Copy those packages and .java files into the *src* folder.

/\*\* ***Important*** \*\*/ Open the file:

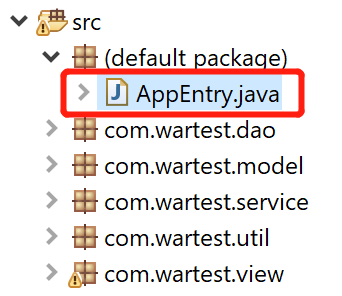
com/wartest/util/DbUtil.java



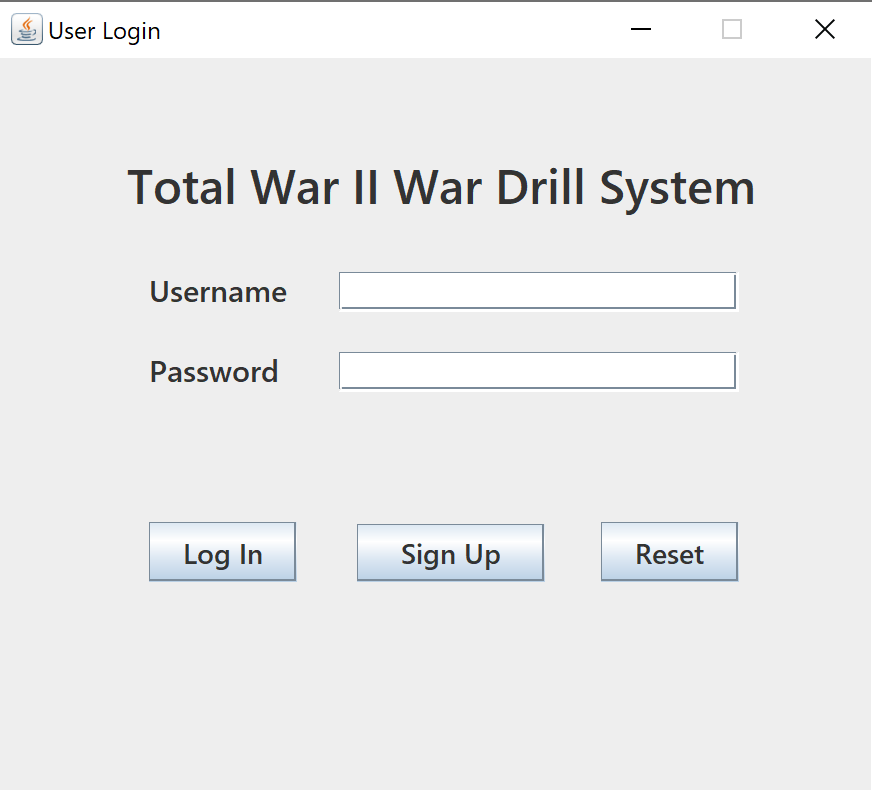
and **change the username and the password** that serve your own database.



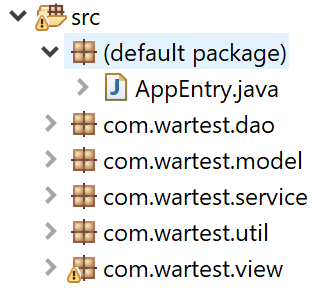
There is only one java file in the default package: AppEntry.java, which is the only entrance of the whole project. Run this file and the application starts.



When you can see the Log In form below, it means the application works well so far.



**6. Source Files Explanation**



1) AppEntry.java, the only entrance of the whole project.

2) util. It contains two files. DbUtil.java is used for the connection works; StringUtil.java is used to integrate the methods related to String operations.

3) model. It contains all the objects corresponding to the entities of the database. Those objects are used for simplifying the process of front-back-database interactions. Passing objects among files are simpler and neater than passing scattered information.

4) dao. The data accessing layer. The files in this package are used for direct interactions with the database. For example, TroopDao.java contains all SQL Statements, PreparedStatements, Procedure Callings, and their execution methods for Troop entity.

5) service. The business logic (service) layer. The files in this package are used to implement all the business logic related to the user operations, like click SUBMIT button to submit an update, or generate all troops that formed by the user who currently logged in on the front-end windows.

6) view. The front-end layer. It contains all JFrames and JInterFrames. Those files form up the whole UI. They implement all the dynamic logic and user operation responses by calling methods from the service layer.