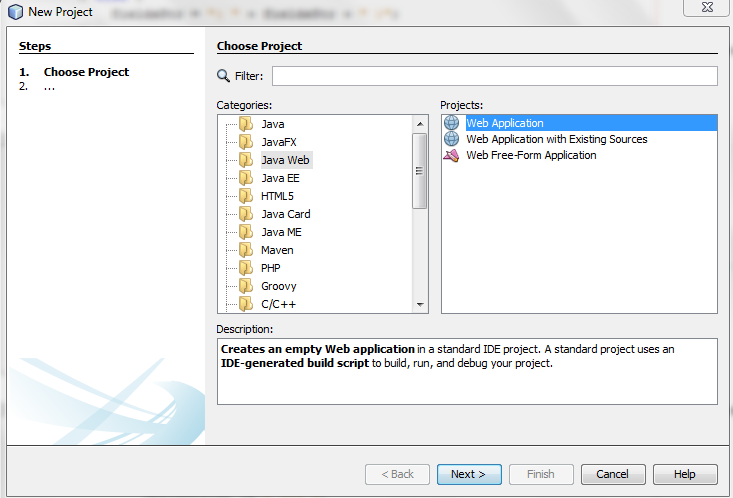
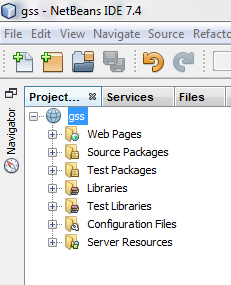
**GSS**

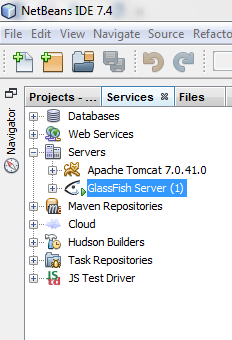
**GESTION DU STOCK**

La base de données contient vingt-trois tables pour bien gérer les attributs et aboutir l’objectif du projet de gestion de stock. Le diagramme UML ci-dessus est préparé sur MySQL Workbench et de suite exporté sous forme d’un SQL script : create.sql



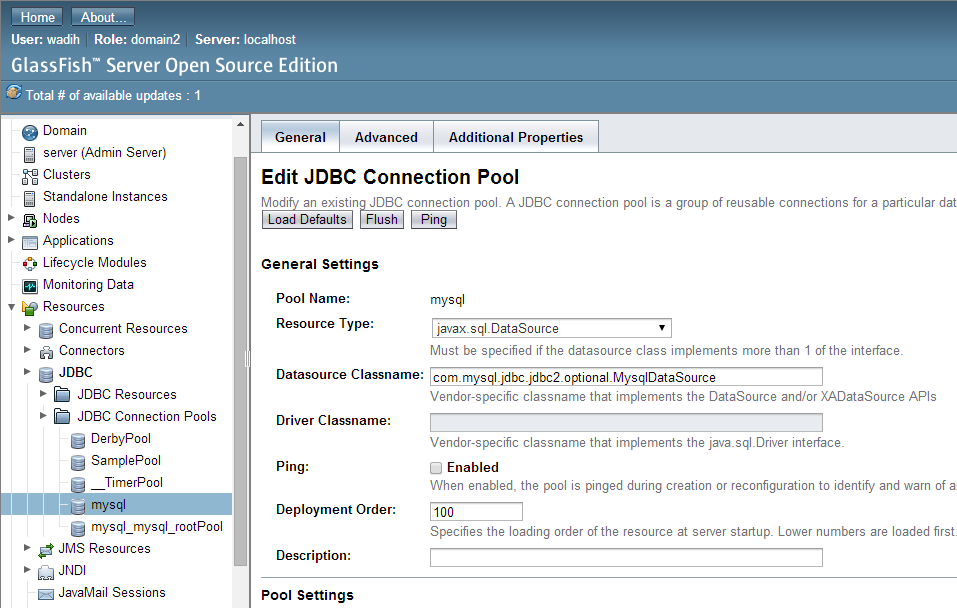
Sur Netbeans nous avons créé un nouveau projet Web Application nommé « gss » dans la catégorie Java Web.

Le projet est déjà créer avec un package complet : de page web de source de test et les librairies convenables.

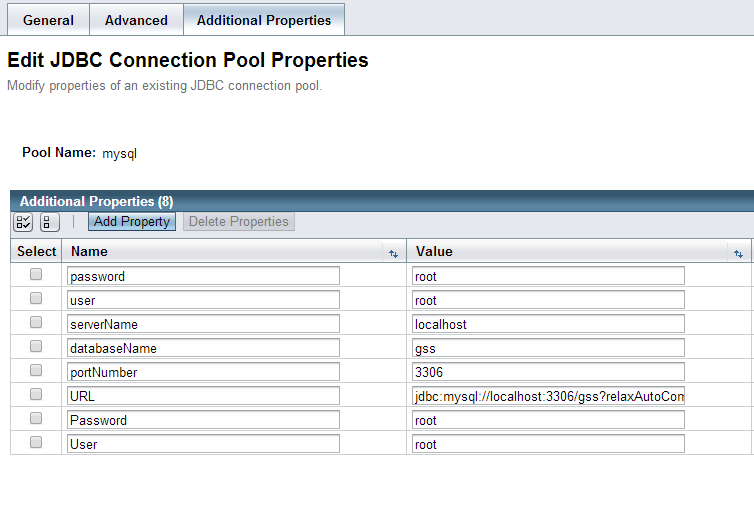


Dans les services, lancer le serveur convenable tout en vérifiant ces propriétés : location avec le port, le nom du domaine et le fichier contenant le domaine dans lequel sera déployé ensuite le projet.

Le serveur choisi dans notre cas est le GlassFish, par la suite nous allons créer un connecteur JDBC sur la console d’administration du serveur pour accéder à la base de données.



Dans le console GlassFish nous ajoutons un nom de pool « mysql » au ressource du JDBC et au JDBC connexion pool nous avons choisi le type «  javax.sql.DataSource » comme type de ressource et « com.mysql.jdbc.jdbc2.optional.MysqlDataSource » comme nom de la Datasource.

Au même niveau, dans les propriétés additionnelles nous avons ajoutés des propriétés avec leurs valeurs :

L’étape suivante consiste a modifié le fichier glassfish-ressources.xml qui est localisé dans les ressources du serveur dans Netbeans ; nous devons enlever les guillemets de la valeur true :

Le fichier « mysql-connector-java-5.1.23-bin » de type jar sera ajouté dans le fichier lib du domaine qui interprète le serveur et le fichier «ext » aussi.

