Technical Design Document

CSCI 3830 Advanced Java Programming

Anna Rue, Kyle Harrison

# Tech Stack

1. Database:
   1. MySQL
2. Front-End
   1. Jsp/HTML
3. Back-End
   1. Java + JPA/Persistence
4. Glassfish Server for running the application

# Application Architecture Diagram

A diagram of a glass fish server

Description automatically generated

# Features

1. Backend
   1. Built a persistence Entity and façade to handle working with the User Objects within the java code
   2. Implemented methods to validate users’ credentials as passed from the front end of the website.
2. Frontend
   1. Users have the ability to login, register a new account, update their account info, and view a show and tell page.
   2. Users may also logout as needed when they are done browsing the site.

# Set Up Instructions

1. Pre-Requisites
   1. User must have Maven installed and useable
   2. User must have access to a Glassfish Server running at least version 7
   3. User must have a SQL database that is connected to their glassfish server
   4. User must have some working knowledge of Jakarta applications
2. Steps
   1. First, the user must get the code, either from a repository or from a shared download location
   2. Second, within the “entity” package there is a sql script, in whichever mysql database that is connected to the glassfish server, run this script
      1. To connect a mySql database to your glassfish server, first download and install MySql to whatever machine is running your glassfish server. NOTE: Remember the user credentials you set
      2. Once your mySql is running, run the script as above within the schema that is auto created. This is most commonly called mysql.
      3. After completing the above step you need to get the correct jdbc-mysql-connector,jar and set that up as a library for your glassfish server.
      4. Next open up your glassfish admin panel and navigate to the Resources -> JDBC -> JDBC Connection Pools page. From here you should be able to create a new Connection Pool.
         1. Name the pool whatever you like, set the Resourse Type to “javax.sql.DataSource”, Set the database driver vendor to MySql, and then select Next at the top right
         2. On this page set the Datasource Classname to the same classname as whatever mysql-connector,jar you added as a library, in the additional properties section add the following properties
            1. user = your MySql Database username
            2. password = your MySql Database password
            3. useSSL = false
            4. port = 3306
            5. databaseName = whatever the name of the schema you are using is
            6. serverName = localhost or the IP of wherever your database is running
         3. Once this is done you can click save, if everything was configured correctly you should be able to Ping the database successfully
      5. After the connection pool is created in the admin panel navigate to JDBC -> JDBC Resources and create a new resource
         1. Name the resource whatever you like, and select the connection pool you created in the previous step. This resource name is the name that should be set in your persistence.xml file
         2. In the <jta-data-source> property
      6. Once the above steps are done you should be able to connect the
      7. This script creates the necessary tables to hold the users objects
   3. Third, the user needs to open a command prompt tool in the folder with the code.
      1. Once the command prompt is open, Run the following command: “mvn clean install”
      2. If this is successful a new folder will be generated called “target” that will contain a .war file for the application
   4. Fourth, if the pom.xml was configured to automatically deploy the code as part of the build process then you should be able to access the application. Else, the user will need to navigate to their glassfish server and follow whatever deployment instructions they have for that specific server.
   5. Finally, the user will navigate to the url (likely localhost:8080) to view our website.