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. 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.