

Task Manager

A database driven to-do list manager written in Java using the Cliche Library

Project Overview

We use the Cliche Library to implement a shell that allows for multiple commands/transactions can occur between the program and the database.

First we prompt the user for their ssh credentials and then the database credentials they are trying to access. This information will be stored in User.java

Second a session is established so that an ssh tunnel occurs using the users ssh credentials. This will allow the user to connect to the onyx server if they have an account. This is done in JDBC.java

Third if the session is established/exists then a connection will be made to the database using the database credentials the user have inputted. If both the session and connection are a success the user will be able to start adding tasks to the database using the main file TaskManager.java.

TaskManager.java contains the queries that the user can run based on the available functions. This would be things like add() so the user can add a task, show() so a user can see the tasks they still need to do, etc.

Project Design

Database Design

We decided to design our database with two entities with a relationship between them (See the E-R Model). We chose that the Tag entity would not be accessible to users and gave generic tags centered around college. This was to stop user error, and since our To Do List is college centered, we were able to create many tags directed toward the purpose of college centered tasks. Because of our N-to-N relationship between the two entities, we created another table to handle that relation.

Code Design

We based the majority of our code design on the given Cliche Library. All mySQL queries are processed in our TaskManager class. The JDBC class ssh's into Java and connects to the input database. We also have a User class that keeps track of the user inputs to help with organization and connecting the program to the database.

We purposely made the code very simple since we wanted to focus more on the database side.

The Project Process

Our team started off this project on the weekend and spent all day working together to design the database, code the command line interface, and work on queries. It was surprisingly fun to work together and bounce ideas off of each other as we coded.

We drew out a lot of diagrams and lists to ensure that our program has the expected behavior.

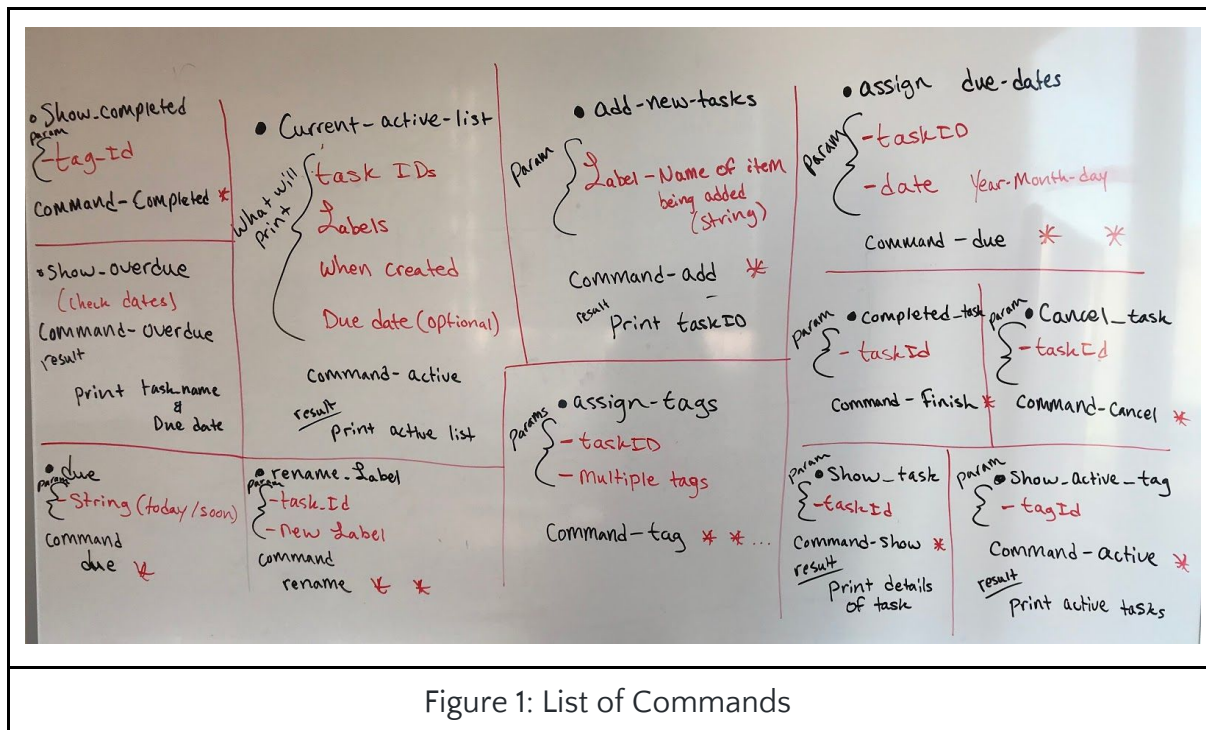


Figure 1: List of Commands

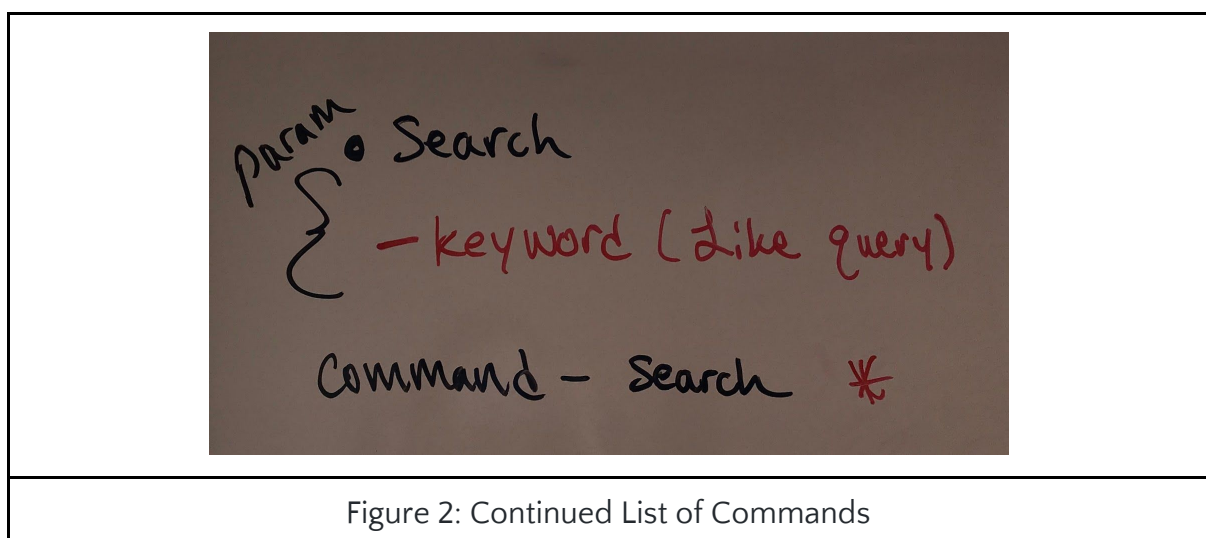
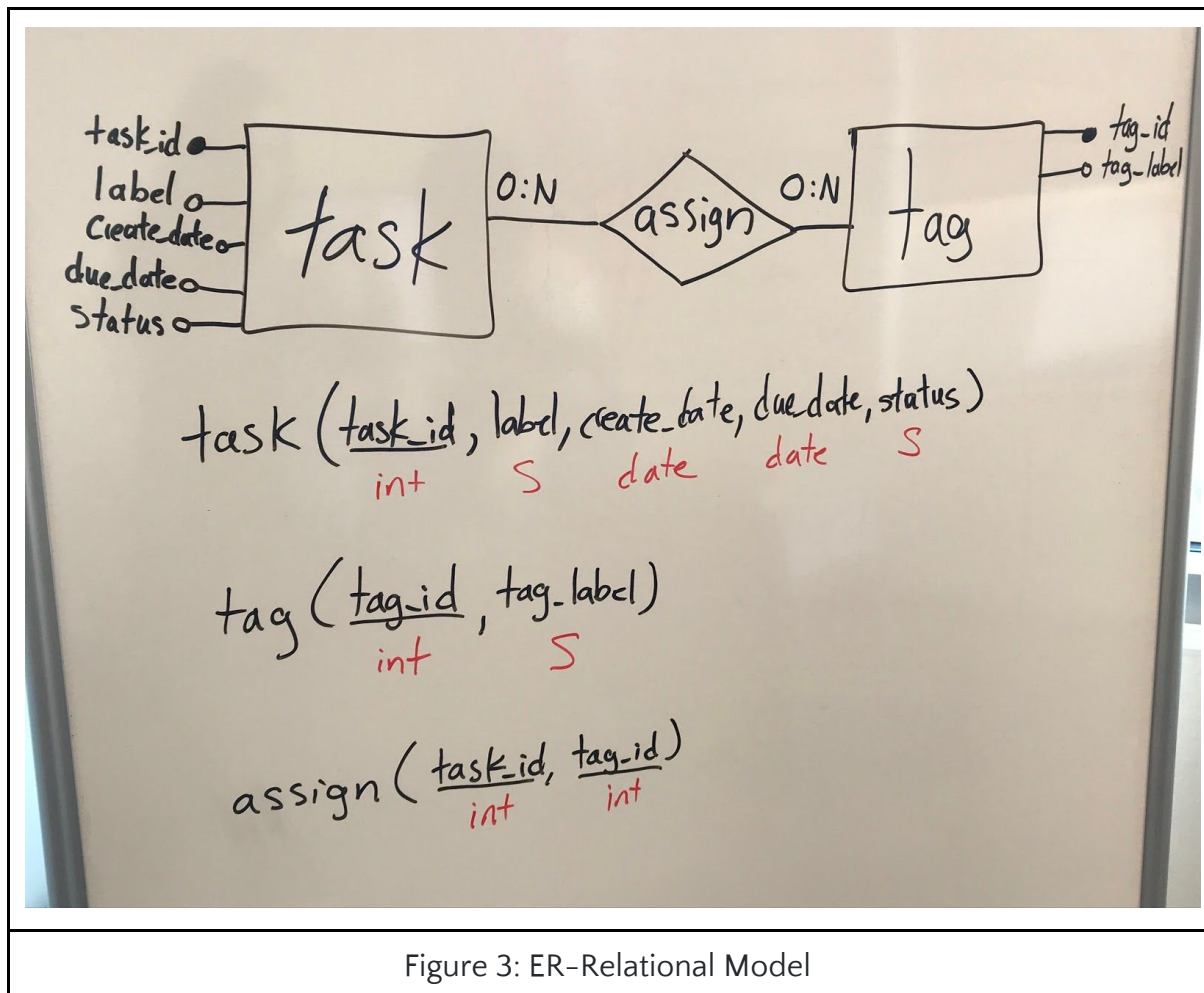


Figure 2: Continued List of Commands



We did have a lot of trouble with ssh'ing to Onyx. We spent the majority of our time getting our Java program to connect to our database on Onyx.

Our first attempt at the project, we did in Onyx which made the database connection very simple, but after examining the project specifications, we found that we'd have to be able to use this program on a local machine. So we pulled the code to a local machine and ran into a bunch of tunneling issues. It took us a long time, and we spoke to Micheal D. Ekstrand about the issue.

After a few more hours of work, the program was able to access Onyx and through Onyx access our database.

Overall, the project was fun, but connecting everything together took the most time and was the most frustrating part of the project.