Class Project

CS 410 Spring 2018

100 points[1] - Due on 04/29/2018 (Sunday)

In this project, you will design and implement a database-driven to-do list manager in Java. This will be a command-line application; you can look at TaskWarrior (https://taskwarrior.org/) for inspiration.

You may work with a partner on this assignment.

Project Requirements

Your to-do list manager needs to support the following commands:

- View currently-active tasks - list the task IDs, labels, create dates, and due dates (if assigned):
  `active`
- Add new tasks (e.g., add a new task with the label "*Finish Assignment*"; it should print the task ID once it has added the task)
  `add Finish Final Project`
- Associate due dates with tasks - to make task 7 due on April 1:
  `due 7 2018-04-01`
- Associate tags with tasks - to tag task 7 with 'school' and 'homework':
  `tag 7 school homework`
- Mark tasks as completed
  `finish 7`
- Mark tasks as canceled
  `cancel 7`
- Show details for a task
  `show 7`
- Show active tasks for a tag
  `active school`
- Show completed tasks for a tag
  `completed school`
- Show tasks overdue (due date in the past but not completed)
  `overdue`

[1] This assignment is 15% of the final grade.

- Show tasks due today, or due in the next 3 days
  ```
  due today
  due soon
  ```
- Change the label of a task
  ```
  rename 7 Finish Final Project
  ```
- Search for tasks by keyword (e.g. search for tasks having the word “project” in their label)
  ```
  search project
  ```

You do **not** need to support multiple user accounts or authentication - this is a single-user task list. You **do** need to properly use transactions so that multiple commands can run simultaneously.

Technical Requirements

Implement your project as a Java program that uses the MySQL database

- Implement commands for the various operations in an *interactive shell*, e.g. by using the Cliche library.
- Accept a MySQL URL string (the part after ‘jdbc:’, e.g. ‘mysql://me:pass@localhost:3298/db’) on the command line.

Submission Details

Submit 6 files:

- A PDF containing your E-R model.
- An SQL file called ‘schema.sql’ that contains your SQL DDL for your database.
- An SQL file called ‘example-data.sql’ that populates your database with some example data.
- A zip file of your source code
- An executable jar file that lets us run your assignment with ‘java -jar’ and includes all dependencies; the Maven ‘assembly’ plugin can help you generate this.
- A README file describing your implementation.