Spring 2018 - CMPS 341 Group Project

For this assignment, you will learn how to design a database application called “MyHikes” in Java using JDBC. MyHikes is a standalone Java application which allows users to create and update hiking journal entries and share them with the hiking community.

Each user needs to register as a hiker with the system with his/her email address, name, and a self-selected password. Each hiker can enter the details of specific hikes they’ve completed. A hiker can also enter any number of journal entries for each hike, each of which has a title, a journal body, and a list of tags. The system also keeps track of the date at which the journal was posted.

A hiker can search his/her hikes based on the tags they’ve entered for that particular journey. For example, if the hiker saw a bear, some moose, and a marmot, he/she could enter each of those animal sightings as a tag. At a later time, if the hiker decided to display all hikes during which bear and moose were sighted, the system could perform a search based on the tags. Tags are automatically recorded in the system when a word or phrase is surrounded by curly braces in a journal entry--e.g. Today I saw a {bear} in the woods.

A menu-driven user interface, either text-based or graphical, is recommended for this team assignment. Please note that additional points will not be granted for the fanciest looking UI. The most important aspect of any system is that it is functional and user friendly. In other words, the user should intuitively know how to use the system.

Data Requirements:

1. Each hiker has a name, password, and a unique email address.
2. A hiker can enter any number of hikes. (In fact, a hiker may choose to hike the same trail any number of times).
3. A hiker can enter any number of journals per hike (day 1, day 2, etc.), but it must have at least one.
4. Each hike and corresponding journal are created by at most one hiker.
5. Each hike has a location name, beginning set of coordinates (latitude and longitude), an ending set of coordinates, a date and time the hiker began the journey, an optional trail name, total distance (in miles), the total time it took to complete the trail, and a list of tags.
6. A journal has a title, a journal body, and the date and time the user entered it into the system.
7. A journal body can contain any number of tags associated with the hike.
8. Each tag has a name, which may be a word or phrase. (See examples above).

Functional Requirements:

1. Register Hiker. A new user enters his/her email, name, and password.
2. User Login. An existing user enters his/her email and password for authentication.
    a. Passwords should be encrypted.
3. Change Password: An existing user updates his/her password.
4. Create Hike. A hiker creates a new hike.
5. Add or Update Journal Entry. A hiker adds a new or updates an existing journal entry.
    a. This use case requires that the same menu option be used for both add and update. If an existing journal entry is specified, the system should prompt the user to verify that he/she wants to overwrite the existing journal entry.
6. View Hikes. A hiker can view all of his/her hikes ordered either chronologically or by location name.
7. View Journal. A hiker can view all of his/her journal entries for a particular hike.
8. Search Hikes. A hiker can search hikes posted from any and all hikers by date, location name, or tag.

Submissions:

1. ERD due by April 16, 2018
2. Database Schema due by April 18, 2018
3. Functional Requirements 1, 2, 3, and 4 due by May 2nd, 2018
4. User Manual, Final System, and Peer Review due by May 11, 2018

You will be submitting your code to a git repository located on kepler.cs.scranton.edu. Instructions for submitting your code will be added by April 8th. For now, focus on the data requirements, ERD, and schema. It is highly recommended that you produce the queries for retrieving the data before any Java code is written.