National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

SE312: Software Construction
Class: BESE – 5 AB

Lab 6: Data Analysis with Hibenate
Date: April 12th, 2017
Time: Wednesday (10:00 – 13:00), Wednesday (14:00 – 17:00)

Instructor: Fahad Ahmed Satti

Lab 7: Data Analysis with Hibernate

Introduction

In this lab, you will be solving a real world problem, where by data collected with 3 novel graphical password authentication schemes (CHC and its variants, CO-CHC and Rogue CHC) will be used to build a simple data storage and analysis application in Java.

You will need to read the CSV file and build hibernate based data storage and analysis application.

Objectives

After performing this lab, students will be able to understand:

- Hibernate
- Data Analysis
- HQL and HCQL
- Parsing CSV files

Tools/Software Requirement

- Solutions should be made using Java and must use the ORM Hibernate.
- You can take help from internet but remember **no plagiarism.**

Description

Graphical password schemes are unconventional types of passwords in which a user's password is in the form of images or some points on an image. They are in use around us in the form of the picture password in Windows and the pattern recognition password in Android. A graphical password scheme named Convex Hull Click password is a potential graphical password under research.

The following three graphical password schemes were recently used for a usability study:

1) Convex Hull Click Graphical Password Scheme (CHC)
2) Centroid-Oriented Convex Hull Click Graphical Password Scheme (CO-CHC)
3) Rogue Convex Hull Click Graphical Password Scheme (Rogue CHC)

In this study, a user had to complete 5 successful logins of the scheme, called rounds. Each round is represented by a row in the CSV File. Each round consists of 5 challenges or rounds in which the password had to be entered correctly, with at most 2 wrong tries. On 3rd wrong try the user would not be authenticated in that particular round and the state would be set as False. For

successful authentication, in any round, the user had to get atleast 5 challenges correct. For example, if a user got a successful login with 2 wrong tries, he or she will have succeeded in 7 challenges. The data of the challenges of a single login is given in front of it. Please note that the column headers, in the CSV file, indicate the following.

1) SCHEME: The type of scheme, experimented with
2) Time taken per input: Time taken to complete a single login of the scheme
3) STATE: TRUE if a login was successful and FALSE if a login was unsuccessful
4) T-C[number]: Time taken for a challenge
5) STATE (For the challenges): True if a challenge was successful and FALSE if a challenge was not successful.

Lab Task

Your lab task will be comprised of the following 2 parts:

1. Create a DAO (Data Access Object) layer for interaction with the DB.
2. You must use parameterized insert HQL queries.
3. Data Storage
    a. You will have to read the CSV file for the 3 password schemes and build a Hibernate based data storage application.
    b. Ignore all entries against username with the word “test” (case insensitive) in it.
    c. Ignore all entries, where “time taken per input” is 0.
4. You must make a separate package called BO (Business Objects), where you will create classes with the data analysis functionality, for each scheme.
5. Your unit tests will check the functions in the BO layer.
6. For DB interaction, your BO will call DAO which in turn will only use HCQL for retrieving data.
7. Data Analysis (for all 3 schemes)
    a. Find all users, who have not participated in the current scheme.
    b. Mean percentage of correctness for a round (5 correct logins to be considered) {EXAMPLE: For every distinct participant = (Total No. of rounds/Total No. of successful rounds)*100}
    c. Standard Deviation of the percentage correctness of a password login
    d. Mean percentage of correctness for challenges(within each round) {EXAMPLE: For a single participant’s 1 correct login in a round = (5/Total No. of challenges)*100}
    e. Standard Deviation of the percentage correctness of challenges

f. Mean time taken for a correct password input (Also, find its standard deviation and range)
g. Mean time taken for every correct login of all the users (Mean time taken for the 1st successful login occurrence of all the users, mean time taken for the 2nd successful login occurrence of all the users and so on)
h. Mean percentage error of every scheme {EXAMPLE: For a single user = (No. of false logins/No. of total logins)*100}

8. Using a Version Control System (VCS) to manage your solutions.
9. Do not change the CSV file.

Deliverables

- Each submission is individual with the following composition:
  - Source Code
  - Unit Tests
  - Documentation(Introduction, ERD, How to Run and Analysis+Results)
  - Link to the public repo on GitHub
- Convert your submission files into a zip folder and name it as given below, finally upload the zip folder to LMS.
  - Name – Registration No. – Section

Grade Criteria

This lab will be graded on the following rubric, with minimum marks 0 and maximum marks of 24:

Criteria	0	1	2	3	4
R1 Completeness and Accuracy	The system failed to produce the right accurate results	The system execution led to inaccurate or incomplete results. It was not correctly functional or not all the features were implemented.	The system was correctly functional and most of the features were implemented	The system was correctly functional and all of the features were implemented	The system was correctly functional and all of the features were implemented. It was demonstrated how the real world problem was solved
R2 Coding Standards	Coding standards, best programming practices are not followed. Students cannot understand the code.	Coding standards, best programming practices are not followed.	Coding standards, best programming practices are rarely followed.	Coding standards, best programming practices are followed appropriately	Coding standards, best programming practices are followed extensively
R3 Ways of Demonstration	The system does not fulfill the functional requirements.	It is not clearly demonstrated how the system fulfills its functional requirements	It is demonstrated how the system fulfills some of its functional requirements	It is demonstrated how the system fulfills most of its functional requirements	It is clearly and effectively demonstrated how the system fulfills all of its functional requirements
R4 Quality	Student is unaware of System's non-functional requirements	System's non-functional requirements (as mentioned in SRS) are not demonstrated	Some of the system's non-functional requirements are demonstrated	Most of the system's non-functional requirements are demonstrated	All of the system's non-functional requirements are clearly demonstrated
R5 Originality	Most part of the working product is copied.	Working product is uninspired and straightforward work with little to no creative potential.	Working product has some potential for making a creative contribution.	Working product has some creative /original /inventive element and a potential for making a creative contribution	Working product has several creative /original /inventive /innovative elements and a clear potential for making a creative contribution.
R6 Modern Tool Usage	Modern engineering software were not used, where applicable, to solve complex engineering problems.		Computer-based tools and technical software were used, but more could have been used to solve the problem.		Modern computer-based tools and software were used extensively in the project. New software/language was learned as needed