Glasgow Caledonian University

Summer Internship Program

August 9th, 2011

Project On

QWallet: An Electronic Wallet for Generating and Storing Security Questions and Answers

Submitted To
Mike Just

Submitted By
Vy Thuy Nguyen

1. ABSTRACT

Design and develop a standalone Java application that will let user create a QWallet, which stores the user's security questions and answers that are used by the user to authenticate different websites.

2. PROJECT OBJECTIVES

The following description of the objectives of QWallet Project has been taken from the Full Project Synopses and Allocations 2011.

Some key objectives for the QWallet Application include the following:

(I) Functional.

QWallet needs to meet its prime requirement: Storing security questions and answers that can be used to authenticate to web sites.

(II) Secure

Access to a user's QWallet needs to be protected with a password. And the security question generation tool should try to avoid generating very insecure questions, such as "What is your favorite color?"

(III) User-friendly

Creating and using a QWallet account should be as easy and convenient as possible for users.

(IV) Creative

Use your imagination and creativity in the design, especially in designing a tool for generating security questions!

3. PROJECT METHODOLOGIES, RESULTS AND ACHIEVEMENTS

The purpose of this section is to summarize the developments that took place within QWallet project.

a) Application Development and Obstacles

My approach to the project consists of four main steps:

- ***Designing the interface of the application***

- ***Creating the MySQL database and text files***.
  My MySQL database is named **QWallet** and contains the following tables:
  + **UserInfo**: this table stores users' identification, and has five columns: User, Password, Name, Passort/ID Number, and DOB. Name, Passport/ID Number and DOB make up the composite primary key of the table
  + **QandA**: this tabled stores users' security questions and answers.
  + **SavedUandP**: this table stores user and password of user who has specified to save his/her User and Password or chosen Automatically Sign in.

  Aside from the MySQL database, there are two more text files which are essential to the execution of the application. The first text file is **QuestionList.txt**. This file contains all the sample questions that users can choose from. More questions can be added to this file if the user/future developer so desires.
  The second file is **Words.txt**. This file contains all the words that user can pick to make up a question. *(See* **b) Feature Description** – ***Fridge-Magnet Method)*** Similar to the previous file, more words can also be added to this file for future improvement of the application.

- ***Designing and develop a Java class which has all the methods needed to interact with the MySQL database through the QWallet application.***

- ***Coding the main part of the application***

During the development of the application, it was fortunate that I did not come across any major obstacles. This is mostly due to the familiarity with Java Swing based components, which I have been working with a lot in the last two years.

There was, however, one minor problem at the end which had to do with transferring my MySQL database to the CD, so that it can be copied to another machine. After a few failed attempts, I was able to find out the solution. The solution I chose was to dump my MySQL database (named QWallet) to a file by executing the following command:

```
mysqldump –-quick -u <MySQL User> -p QWallet > QWallet.sql
```

And then copy the file QWallet.sql to the CD.

b) Feature Description

This section briefly describes some interesting features of the QWallet application.

- ***The Login Screen:*** User has the option to let the application remember his/her User and Password, or even Automatically Sign in, the next time he/she runs the application

- ***Password Strength Check:*** When creating a new account, the user will be able to see the strength of the password he/she has just chosen. The criteria for measuring the strength of the password is specified below:
  - ➢ Very Strong Password:
    1) Has at least 6 characters.
    2) Has uppercase letter(s) AND number(s) AND special character(s)

  - ➢ Strong Password:
    1) Has at least 6 characters.
    2) Has uppercase letter(s) AND number(s).
       OR has uppercase letter(s) AND special character(s).
       OR has number(s) AND special character(s).

  - ➢ Fair Password:

1) Has at least 6 characters.
2) Has upper case letter(s) OR number(s) OR special character(s)

➢ Weak Password: <the rest>

- ***Completion Method:*** as the user types his/her question, a list of questions starting with whatever the user has typed will be displayed. (This text field works almost like Google's search box)

- ***Fridge Magnets Method:*** this feature allows user to pick words from a pool to create a question. The user selects a word simply by clicking on the button containing that word. The button will then be moved to another designated area. To remove a word, simply click on that word again.

4. CONCLUSION

This section summarizes the progress with respect to the main objectives of the project, namely, functional, secure, user-friendly, and creative.

i) ***Functional***: apparently, the application meets this goal. It has the ability to store users' questions and answers, as well as let users retrieve those questions and answers.

ii) ***Secure***: user needs a password to access to his/her QWallet. There is, however, one limitation of the application. Since the application stores user's username and password in MySQL tables, whoever has access to MySQL will be able retrieve this information.

iii) ***User-friendly***: the application has a simple and easy-to-use interface.

iv) ***Creative:*** the application provides users with three different ways to generate a security questions. Personally, I find the "Fridge-Magnet" method most interesting.