**Homework 2**

**Jiangfeng Wang(jw3107)**

**Tianrui Peng(tp2522)**

url for github: https://github.com/janewanggg/ASE\_HW2

1. Description of the system

Our toy system is a simple user registration and login web application. Users can create accounts, and the application would save the account and password information into a local database. Once created the account, users can also sign in into the account or update their password if they forget. Users can change their password by clicking forget the password.

To run the program, from terminal type “./build.sh” to run. (NOTE: If you have two versions of python and user “python2” to differentiate compiling python version on terminal, run “./buildpython2.sh” instead. The program is written in python2.) Open the browser with url specified by the terminal output upon starting the server (<http://0.0.0.0:8111/> by default). Make sure Mysql server is running for correct database connection.

(We will try to have our mysql server is up for the following. If not, database connection will be down and web application cannot function well. We also provide our database sql files so that everyone can build the same database locally.)

2. Technologies

Programming language: python

Overall, we feel that python is a great language to work with. It has a lot of useful toolkit online that provides all kinds of functionalities. Python is also easy to set up the environment. We are using python2 for our team project because the API provided is also written in python2, which makes it easier to use.

Framework: flask, flask SQLAlchemy

UI design: template and css files from Bootstrap

Our experience with flask is pretty good. Flask is great for working with data storage and easy for setup. We were able to find a lot of tutorials and documents online to learn it and to start from scratch. By using flask and Bootstrap, we were able to build a decent front-end web application in a short amount of time.

Database: mysql

Mysql is a really convenient tool for database usage. Since we are working on different OS, it took us a while to install all the dependencies and setup the environment for it. But once the environment has been successfully set up, mysql is very easy to use.

Test cases: flask\_testing

For test cases, we used flask\_testing to test register and login functionality. We created another database schema, exactly same structure as the one used for web application, because we do not want to mess up with the data information stored in the original database. All the sample data inserted for testing will be deleted once the tests are done. The most challenging part for using flask\_testing is to configure another database connection. For insertion, updating and cleanup, flask\_testing is a very useful and convenient package to use.

3. Challenges:

Besides a long time on environment setup, we were having trouble connecting our front-end with the database at first. Flask SQLAlchemy is well-documented, but the sample on their official website does not have enough details for more complicated cases. For example, when a user create an new account and typed same username as others (username is a primary key in the table), it took us a long time to figure out how to check the error and return user appropriate message. After we went through online tutorial on youtube, we successfully connected our front-end with the database and implemented some features to prevent user error usage cases.

We also want to be able to have others, such as our project teammates, to be able to connect to the database, not locally. To achieve this feature, we added database user and password manually in database. However, it may not be the best way to do, so we will discuss about with the team and IA and try to solve this issue.