Social Network Search Engine

Yow-Bang Wang

piscesfantasy@gmail.com

August 7, 2012

1 Introduction

This is my final project of grad-school class "Design Pattern and Software Development" of National Taiwan University (NTU) in Spring 2012. This project is done in pair-programming. My partner (larry61410) was mainly in charge of the implementation of API calls to social networks. And I was in charge of framework design the remaining implementations. In this project we implemented a social network search engine with C++ and Python. The first version support search on Plurk and Twitter. Main functions include keyword/friend search, browsing search results, recommendation and reply.

2 Compilation

The makefile in the main file directory have been tested for g++ complier under Linux and MacOS.

1. *make* or *make all*: constructs the binary executable "SocialNetworkSearchEngine.exe".
2. *make clean*: remove the "build" folder and all the .o files inside.
3. *make doc*: if you have installed Doxygen and GraphViz, this command will generate HTML document inside "doc" folder.

3 Execution

In this section we will demonstrate how to run this program *SocialNetwork-SearchEngine.exe* if you already have a Plurk account. For Twitter the same procedure should apply too.

3.1 Preparation

The folders "Plurk" and "Twitter" contains python scripts that will be called. Therefore to run this program, the folders "Plurk" and "Twitter" must be in the same directory.

We use the following Python libraries for Social Network API:

1. Plurk: plurk-oauth by clsung, available at http://www.plurk.com/API
2. Twitter: tweepy by joshthecoder, available at https://dev.twitter.com/docs/api

3.2 Login

This program adopts interactive command line interface. For the first execution of *SocialNetworkSearchEngine.exe* you will see the messages below:

```
Initializing Twitter
Initializing Plurk
Warning: can't load query history log
Choose Social Network (type "exit" to quit):
>
```

Just type "Plurk" If you want to login Plurk (the initial must be upper case!) and press enter. Then you can input your id and password. If this is your first login with this program, after entering ID and password you will see something like this:

```
Open the following URL and authorize it
http://www.plurk.com/OAuth/authorize?oauth_token=.....
Input the verification number:
```

Browse the authentication webpage to get the verification number. This procedure will run twice to get required access token, and the program would store the access token in the file "API_plurk_all.keys" and "API_twitter.keys" in current directory.

3.2.1 Password Masking

Note we have also implemented Password Masking. However, this function was done by POSIX, which means this function is valid only under Linux environment.

3.3 Search

After logging in you will see this:

You can choose "n" to logout, or choose "y" to begin a search. The program will ask if you need "Basic Search" or "Advanced Search":

```
Start the next query? [y/n]:
>
```

```
Choose query mode (1.Basic, 2.Advanced):
>
```

1. Basic Search: Search over all the public messages with a keyword.
2. Advanced Search: Search over all the messages of your friends with a keyword and a friend's ID.

3.3.1 Auto-complete

While entering either keyword or a friend's ID, this program also support auto-complete. For keyword the program will search and iterate through your most recent 100 queries stored in the file "queryHistory.log" in current folder; and for friend's ID the program will search and iterate through your friend lis stored in the file "FriendsList" in current folder.

Yet for now this function is normal only for English words. There may be some unexpected error if you try to auto-complete non-English words.

3.4 Browse, Recommend and Reply

After entering your query, the program will retrieve relevant posts and store them in the file "POST.json" in current folder. The first post will be shown on the screen followed by some options:

```
Whats next?
- Pu this post [l]
- Reply this post [r]
- Go to next post [n]
- Quit browsing [q]
```

If you are browsing posts except the first one, the option will also includes:

```
- Go to previous post [p]
```

Just enter the option in the bracket to execute.

3.5 Terminate

To terminate the program, just type "exit" when you are not logged in any social network.

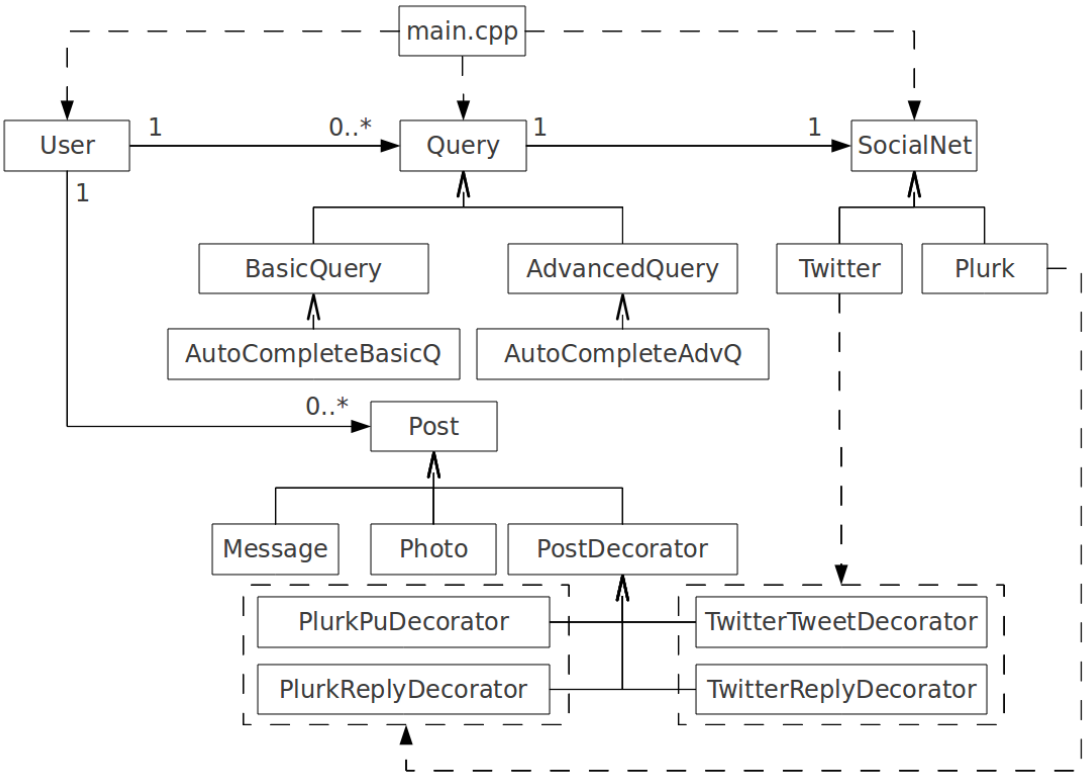

Figure 1: *The framework of our program*

4 System Framework and Design Patterns

Fig. 1 shows the overall framework of this program. There are several design patterns involved:

1. class User: Singleton
2. class SocialNet: Singleton + Prototype Manager + Abstract Factory
3. class Query: Command + Template Method
4. class Post: Decorator

For more details please refer to "doc/designPattern.pdf".