CMPSC 290
Principles of Software Development
Fall 2013

Final Project Assignment: Implementing and Releasing a Twitter Analytics System

Introduction

Throughout this semester, you learned how to use tools that enable you to to specify, design, implement, test, and document Java programs. You also had several experiences with working in progressively larger teams to complete the phases of the software life cycle, with a focus on the elicitation of software requirements, the planning of software projects, and the analysis of a project's design, implementation, and test suite. In this final project assignment you will follow the phases of the software development life cycle to create and release Twitter analytics software.

Using the Twitter Service

Twitter is a microblogging service that allows you to send and receive short, 140-character messages. While many Twitter clients allow you to send and receive tweets and search for trending topics, few give you the ability to search and analyze all of the Tweets that you have sent over the entire time you have used Twitter. If you have tweeted many times in the past, it is particularly difficult to implement a Twitter client that downloads your entire history of tweets because of the way in which Twitter rate limits its application programming interface. Since many people want to search through all of their tweets, Twitter recently announced a mechanism for downloading them all to your computer, as described at the following Web site `https://blog.twitter.com/2012/your-twitter-archive/`. However, manually searching through this archive is very cumbersome.

Requirements for Twitter Analytics

For this final project, your team will implement a complete Twitter analytics system. To start, your program should be able to accept as input the Zip file that a Twitter user downloads by following the instructions in the aforementioned Web site. Next, your program must parse all of the tweets in the Zip file and store them in a relational database so that they can be subject to further analysis. Moreover, your analytics tool must allow the user to connect to the Twitter servers and refresh the program's tweet database so that it includes all of the new tweets since the download of the Zip file from Twitter. Your system should also allow the user to both specify a new Zip file for reloading into the local database and reinitialize the database by clearing out any existing tweets.

Beyond being able to store the tweets in a local relational database, your program should furnish several analyses that users can perform to learn more about their tweets and tweeting behavior. For instance, the system should accept a pattern from the user and then return all of the tweets that match this pattern. Another feature could involve an analysis of the days of the week or the times on which the user most frequently tweets. You should also implement features that identify common words or phrases in a user's tweets. At minimum, your system should offer at least five useful ways to analyze the full history of a user's tweets. Throughout the entire project, you should interact with your customer to best make decisions about what analysis features to include.

At minimum, one version of your system must be implemented in the Java programming language and use JCommander to parse command-line arguments. Please use SQLite version 3 to store all of the tweets and the information about the tweets. Your entire system should be tested with JUnit test cases that achieve a high level of coverage according to the JaCoCo coverage monitor. You should also regularly analyze the source code and test suite of your system using tools such as JDepend, JavaNCSS, and MAJOR. Each separate analysis of your program should be accessible through an Ant build system that also supports compilation, documentation, and cleaning tasks.

In order to implement your Twitter analytics system, you must create a project and register it with the Twitter developer network. Please take care to follow all of Twitter's rules in order to ensure that your program is not prohibited from accessing Twitter. Then, you will need to learn how to write a Java program that can access Twitter through the Twitter4J system described at `http://twitter4j.org/en/index.html`. See the instructor if you have questions about Twitter.

Students may earn extra credit if they can implement alternative interfaces to their analytics program. For instance, in addition to the command-line Java program, you could implement a mobile application or a Web site that allows users to upload their Twitter-provided .zip file and perform various types of analyses. Your team should talk with your customer to better understand either requirements for alternative interfaces or enhancements to the basic command line.

Summary of the Required Deliverables

This assignment invites your team to submit one printed version of the following files:

1. A description of and justification for your team's chosen organization, roles, and tool support
2. A description of the internal structure of a downloaded Twitter archive
3. A description of the format of the file that stores all of the tweets for a single user
4. Requirements, architecture, and design documents for your Twitter analytics system
5. The schema of the relational database that will store tweets and other relevant information
6. Reports from running analysis tools (e.g., JaCoCo, JavaNCSS, JDepend, and MAJOR)
7. Usage and output examples that clearly demonstrate the correct functioning of the system
8. The source code of the build system, test cases, and program

You must also release your complete project on Google Code so that it includes:

1. A distinctive name and logo
2. Easy-to-understand user documentation for display on a Web site
3. The source code of your system's program and test suite
4. A build system that can document, build, test, and analyze your program's source code

You must also ensure that the instructor has read access to your Bitbucket repository that is named according to the convention `cs290F2013-fp-team`*k*, with *k* representing the number of your assigned team. Your repository should contain all of the deliverables that you produced during the completion of this assignment. Once your project is complete, you should place the final version of all public-facing deliverables on a publicly visible Google Code project that uses a Git repository. Please see the instructor if you have any questions about these requirements and deliverables.