BluePrint Requirements

BluePrint was tasked to develop a tweet analyzer that will serve clients by being used for general purposes in order to perform a complex form of analysis for technical customers, and a basic reading of a file of regular expressions for average customers. This Tweet analyzer will collect the information and can be used to interpret the tweets and create various tools of interpretation.

It has 3 components a general-purpose analyzer component with methods used to create and manipulate streams of Twitter’s API.

It has a sophisticated analyzer class equipped with a software development kit that will allow users to use the objects created from the general-purpose analyzer to manipulate tweets with their own methods and classes. They can create and customize their own weather map using this component.

The 3rd component is for non-technical customers. It’s a regular expression application that will use methods created by the general purpose analyzer to call searchable expressions and further manipulate them using methods from the general-purpose analyzer. This can be used to generate a list of tweets that exhibit terroristic qualities based off a list of regular expressions.

The app is customizable with use of the SDK, which will allow the user to be able to create his own implementation of the software. To make usage easier there will be dialogue boxes that have basic instructions for non-technical users.

The target market is Bruce Chittenden and Dr. Muhammad. It is good for advertising and research firms to analyze tweets and determine marketing strategies from how their demographic tweets.

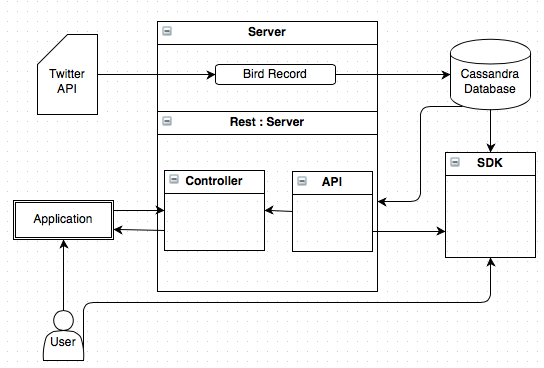
Open Close principle: All software entities should be available for extension in the future, but closed for any modifications.

Dependency Inversion Principle: the details of an application should depend on abstractions, but the opposite should not be true.

Interface segregation principle: Users of the application should not be required to depend on interfaces that they do not use.

Single Responsibility Principle: A class should not be accountable for more than one responsibility. If a class handles two responsibilities, it should be divided into two separate classes.

Transaction model



The BirdWatch application must provide tweets to the end user. These tweets can be used for several different purposes and the tweets need to be stored so they can be accessed from previous times. It will need to implement a database to that communicates with the real time server and stores the tweets in the database.

They will use an Apache Cassandra database to handle storing tweets that will be streaming because of how fast and efficient it is. Twitter and other companies that operate similar to our application already use it. It is also durable and allows for the data to be stored permanently in case the server goes down.

It is designed to have a simple interface that operates like a search bar. You enter the keywords and view a list of tweets that match the keyword. Then it will execute a step-by-step process for custom analysis of the tweets.

Blueprint should implement the application with a server that has a connection to support 15 megabytes per second. This will allow the database to return tweets to the user in a timely fashion. BirdWatch will stream until the capacity is reached and then it will remove tweets from the database starting with the oldest tweets.

Development requirements Intellij IDEA Java SE 7 JavaScript Object Notation JSON Java FX scene Builder

Compatibility Requirments 64 bit Windows Computer and a stand- alone executable file downloaded onto their machine. IntelliJ for the technical user.