Britney Johnson

Software used to implement BirdWatch

* IntelliJ IDEA is used to create the twitter analyzer platform. It provides a comprehensive feature set of tools and integrations with the most important modern technologies and frameworks for enterprise and web dev with Java, Scala, Groovy, etc.
* Apache Cassandra database is used for when scalability and high availability are needed without compromising performance. The benefits it offers were considered better compared to traditional SQL implementations due to the amount of information that will be digested from Twitter.

Product Features

* Searching Tweets
* Searching the stream of information going through the Twitter API is done to choose the appropriate tweets
* Storing Data
* Consumes the JSON schema, inserts info from the relevant fields into a .CSV file and inserting each of those into a database
* Report Generation
* Final step; for users utilizing the SDK to take the info from the database and generate a visual way of interpreting the data (for example – the weather map)

**Use Case Descriptions**

**Use Case 1: Simple Tweet Search**

**Actors:** Any User

**Preconditions:** User must have:

 an internet connection

**Description:** The user types a keyword or keywords into the search bar to filter

tweets stored in the BirdWatch database. When the user presses the search button,

a call is made to the server database to return all tweets that match the search.

These are then displayed on the user interface for the user to view.

**Exceptions:** N/A

**Post‐conditions:** The user is now able to view all tweets that match their search.

**Use Case 2: Regular Expression Search**

**Actors:** Any User

**Preconditions:** User must have:

 an internet connection

**Description:** The user clicks a button to browse for a file that includes a list of

regular expressions. Once the file is uploaded to the server, the file is parsed. This

parse simply reads the regular expressions from the file and puts it into a .csv. Once

the regular expressions are in a .csv file, the Java Regex parser looks at each regular

expression and interprets which java operation should occur on that regular

expression by way of whichever operators are attached to it. The regular

expressions are now saved into the database. Once Java knows exactly what

operation should be performed on the expression, a database query is sent to

Cassandra so that the appropriate matches can be found. Once the matches are

found, the tweets are returned and displayed to the user.

**Exceptions:** N/A

**Post‐conditions:** The user is now able to view a list of tweets that is specific to the

regular expressions uploaded.

**Use Case 3: Custom Tweet Analysis via the SDK**

**Actors:** Highly Technical User

**Preconditions:** User must have:

 developer tools such as IntelliJ IDE

 Java 7 SE installed

 an internet connection

 experience with development in Java

**Description:** The highly technical user will be able to use the predefined methods in

the provided BirdWatch SDK in order to access tweets stored by the BirdWatch

server. The SDK will provide the highly technical user with various methods that

will allow them access tweets stored in the database. These tweets will be returned

as an array of Strings. Each element of the array will be a String that contains all the

pertinent information of the tweet, such as tweet text, username, and date/time. The

7

highly technical user is then able to parse these returned Strings and use the tweet

data as they choose. Once the user has created a complete tweet analysis based on

the SDK, they can create an executable JAR for their analysis that can be run from

within the BirdWatch application.

**Exceptions:** N/A

**Post‐conditions:** Highly Technical User will have unique, customized classes to

view data from Twitter which can be manipulated into views such as a weather map

for example.

Architectural Analysis

System Overview

* BirdWatch application will consist of several different layers of software and interfaces to the environment
* The top layer is the GUI, which is followed by UI Controllers, Data Handlers, Communicators, and the Server/Database.
* UI – the top layer which allows the user to search the entire database with general regular expressions by entering the expression into a text field. It will allow users to upload a text file with a list of regular expressions to create a custom search of tweets by being prompted to do so within the UI.
* UI Controllers – Controllers will handle the data entered within the UI. There will be a controller for keywords entered into the search text box, a controller to handle text files uploaded to the app, and controllers to handle connections between the UI and the server.
* Local Data Handler – will manipulate and modify data.
* Communicator – handles all communication between the server, database, and UI. This includes reading data from database to the UI and storing searches.
* Server/Database – The server is used to stream tweets from Twitter using the Twitter API. Then, the tweets will be stored to the database, which will keep record of all tweets for a specific amount of time due to limited resources.
* Analysis Object Model

* UI Tasks

**UI Tasks**

**Task List**

1. Create simple user interface

2. Create look and feel of interface

3. Create wireframe for interface

4. Develop event handlers for buttons

5. Create look of pre-results

6. Create look of post-results

**Task Specifications**

1. Develop sleep interface for user to interact with. Transitions from one

view to another will be smooth and prompts and notices will pop up

promptly for user.

2. Design colors, scroll bars, font, authentic Twitter experience.

3. Model the applications flow control for action handlers.

4. Develop handlers to control navigation from one pane to another and

to handle actions of various buttons.

5. Develop look to give user show user that the application is loading

tweets to be displayed on the screen.

6. Create readable results showing fields from tweets that include

username, date, time, location, hashtags, and the body.