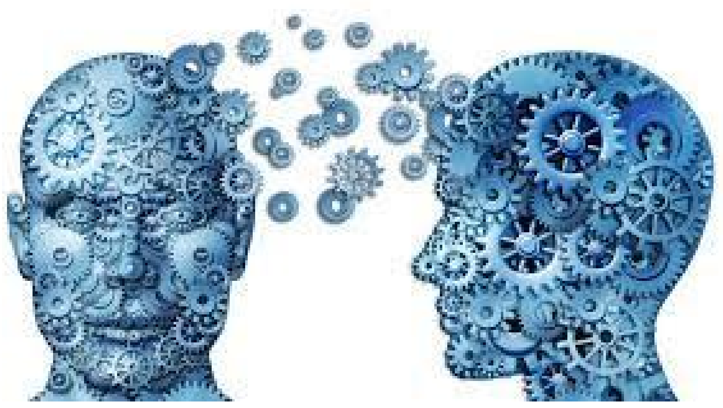
Watson and Twitter App using Bluemix



Software Design Document

Created by
Krishna Damarla
Semanti Kundu
Shalini Chellathurai Saroja

Created On
08-Jun-2015

Table of Contents

1. INTRODUCTION

Purpose

This software design document describes the architecture and system design of the application "Question Watson in Twitter". This document is intended for developers and users of this application.

Scope

The basic goal of this application is to allow user tweet a question in the category of Healthcare to IBM Watson and receive an answer as a post in Twitter.

The extended goal of this application is to allow user tweet a question in the category of Healthcare, in a language supported by this application to IBM Watson and receive an answer in the same language as a post in Twitter. The languages supported by this application are English, French, Spanish and Brazilian Portuguese.

Benefits

This application helps user to clarify doubts in healthcare from IBM Watson via Twitter.

IBM Watson contains healthcare contents from Healthfinder.gov, CDC Health Topics, National Heart, Lung, and Blood Institute (NHLBI), National Institute of Arthritis and Musculoskeletal and Skin Diseases (NIAMS), National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), National Institute of Neurological Disorders and Stroke (NINDS) and Cancer.gov (physician data query)[1].

The questions asked by user can be of the following types.

1. Condition questions like "What is X?" where X is a general medical condition such as a stroke or a specific condition such as Machado-Joseph Disease,
2. Procedure questions like "What should I expect before X?" where X is a procedure such as heart surgery or a blood transfusion,
3. General health questions like "What are the benefits of taking aspirin daily?" and
4. Action-related questions like "How can I quit smoking?"[1].

2. SYSTEM OVERVIEW

IBM Bluemix

IBM Bluemix is a cloud platform for building, running and managing applications. With Bluemix, developers could use a set of IBM and third party APIs and services [2]. Bluemix abstracts and hides most of the complexities that are associated with hosting and managing cloud-based applications [3].

Liberty for Java

Liberty for Java on IBM Bluemix is a highly composable, fast-to-start, dynamic application server runtime environment. It is part of IBM WebSphere Application Server v8.5.5. It enables rapid application development that is well suited to the cloud[4].

Question and Answer Service

The Question and Answer service on IBM Bluemix enables applications to post questions and receive responses from a corpus of information that provides domain knowledge based on a specific set of input documents and other information [5]. Applications can connect to the Question and Answer service by binding the service using functions from the Question and Answer REST API.

Language Identification Service

The Language Identification service on IBM Bluemix detects the language of text [6] posted by user or an application. Applications can connect to the Language Identification service by binding the service using functions from the Language Identification REST API .

Machine Translation Service

The Machine Translation service on IBM Bluemix translates text from one language to another [7]. Applications can connect to the Machine Translation service by binding the service using functions from the Machine Translation REST API .

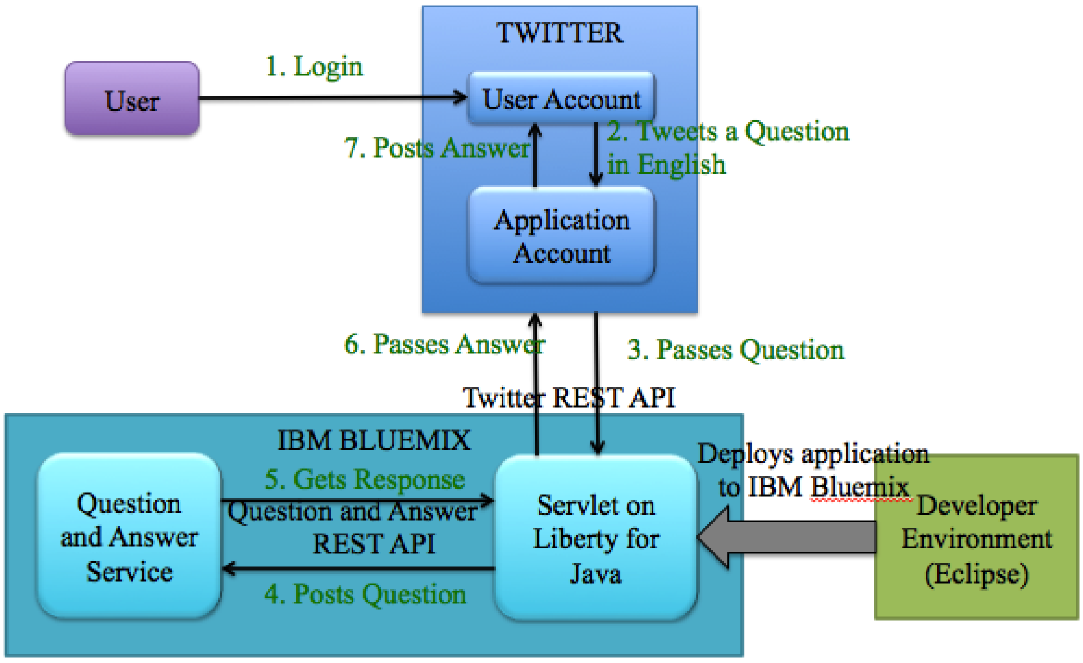
Twitter

Twitter is an online social networking service that enables users to send and read short 140-character messages called tweets [8]. Users may access Twitter through website interface, SMS or mobile device applications. Applications could use Twitter REST API to read or write Twitter data [9].

3. SYSTEM ARCHITECTURE

Architectural Design

Basic Design:

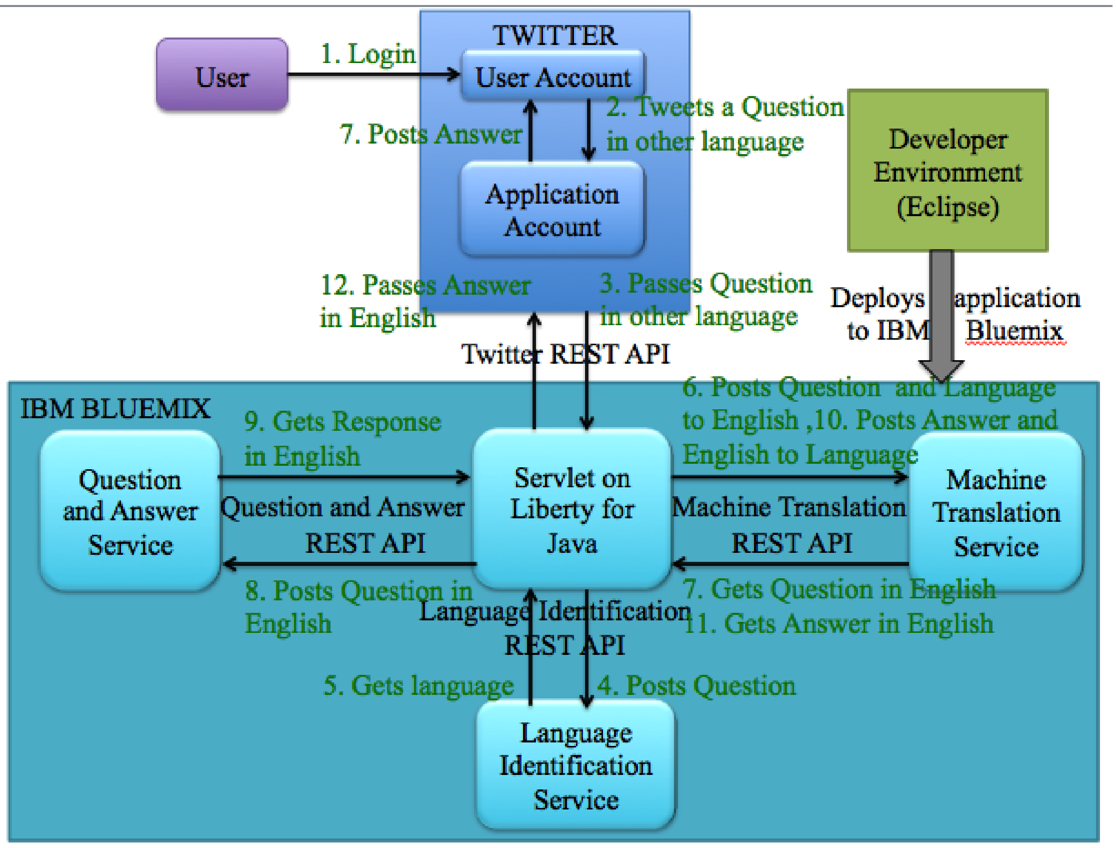

TWITTER
User
1. Login
User Account
7. Posts Answer
2. Tweets a Question in English
Application Account
6. Passes Answer
3. Passes Question
Twitter REST API
IBM BLUEMIX
Question and Answer Service
5. Gets Response
Question and Answer REST API
4. Posts Question
Servlet on Liberty for Java
Deploys application to IBM Bluemix
Developer Environment (Eclipse)

The basic functionality of this application is to tweet a question and receive an answer for the question in twitter. The question and answer are in English in this case.

Description of Basic Design:

1. User tweets a question from user account to application account.
2. This question is retrieved by the application via Twitter REST API through Twitter4j java library.
3. The application posts the question to Question and Answer Service via Question and Answer REST API.
4. The Question and Answer Service processes the question and posts the answer to the application via Question and Answer REST API.
5. The application converts the answer to fewer than 140 characters and posts this answer to the twitter user account via Twitter REST API.

Extended Design:


User
1. Login
TWITTER
User Account
2. Tweets a Question in other language
7. Posts Answer
Application Account
Developer Environment (Eclipse)
12. Passes Answer in English
3. Passes Question in other language
Deploys application to IBM Bluemix
Twitter REST API
IBM BLUEMIX
9. Gets Response in English
6. Posts Question and Language to English ,10. Posts Answer and English to Language
Question and Answer Service
Question and Answer REST API
Servlet on Liberty for Java
Machine Translation REST API
Machine Translation Service
8. Posts Question in English
Language Identification REST API
7. Gets Question in English
11. Gets Answer in English
5. Gets language
4. Posts Question
Language Identification Service

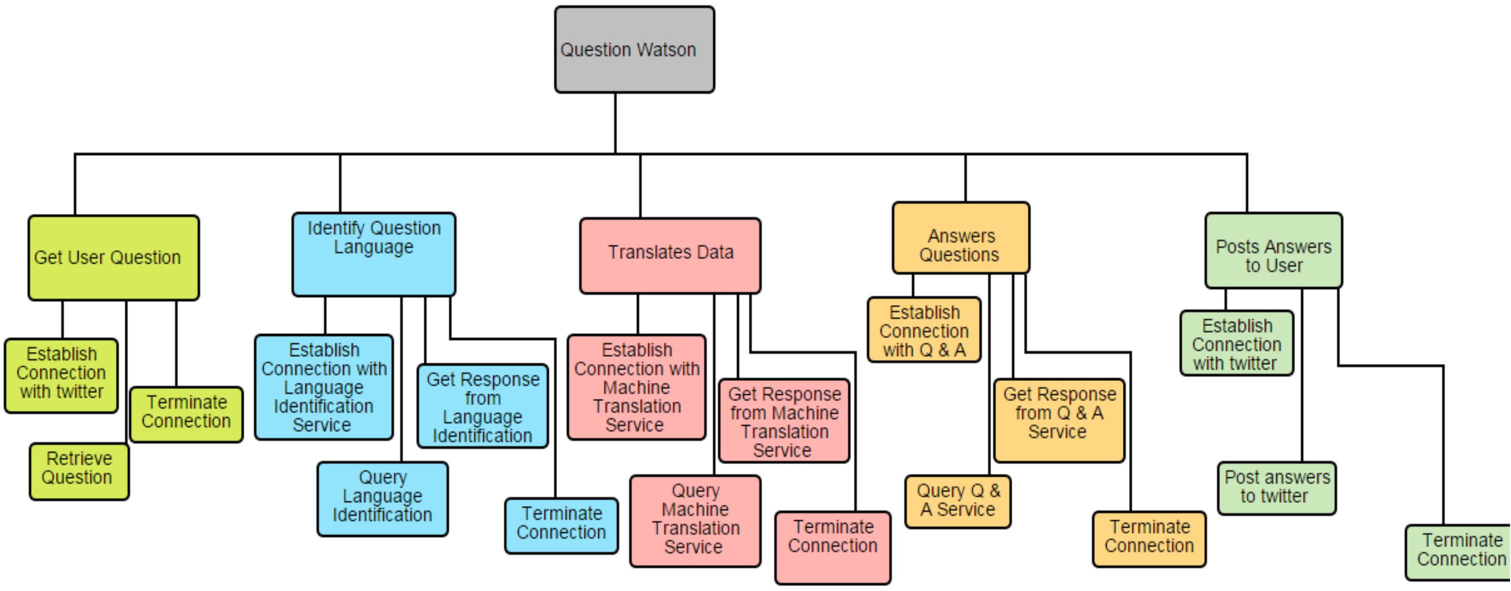
The extended design focuses on tweeting a question in a language supported by this application and receiving an answer for the question in the same language. The languages supported by this application are English, French, Spanish and Brazilian Portuguese.

Description of Additional Functionalities:

1. After fetching the question from twitter, the application posts the question to Language Identification Service via the Language Identification REST API.
2. The Language Identification Service processes the question and posts the language of the question to the application via Language Identification REST API.
3. The question and the language are posted by the application to the Machine Translation Service via the Machine Translation REST API.
4. The Machine Translation Service processes the question and posts the question in English to the application via the Machine Translation REST API.
5. This question is send to the Question and Answer Service and the application gets the answer as in basic design.

6. The answer in English and the user language is posted to the Machine Translation Service and the application gets the answer in user language.
7. This answer is converted to 140 characters and is send to the user account in twitter as in basic design.

Structural Decomposition


Question Watson
Get User Question
Establish Connection with twitter
Terminate Connection
Retrieve Question
Identify Question Language
Establish Connection with Language Identification Service
Get Response from Language Identification
Query Language Identification
Terminate Connection
Translates Data
Establish Connection with Machine Translation Service
Get Response from Machine Translation Service
Query Machine Translation Service
Terminate Connection
Answers Questions
Establish Connection with Q & A
Get Response from Q & A Service
Query Q & A Service
Terminate Connection
Posts Answers to User
Establish Connection with twitter
Post answers to twitter
Terminate Connection

1) Get User Question:

The objective of this module is to retrieve the question from Twitter to the application.

Connection is established between Twitter and the java application via Twitter REST API by using Twitter4j library in the java application. The question tweeted by the user to the application id is then retrieved to the application. Then the connection is terminated.

2) Identify Question Language:

The objective of this module is to identify the language of the question retrieved in “Get User Question” module.

Connection is established between the application and the Language Identification Service through the Language Identification REST API. The question retrieved from the "Get User Question" module is posted as an input to the Language Identification Service. The language of the question is posted as an output from the service to the application. Then the connection is terminated.

3) **Translate Data:**

There are two objectives for this module. First objective is to translate the question retrieved in "Get User Question" module to English. The other objective is to translate the answer from the "Answers Questions" module to a language, in which the question was posted.

Connection is established between the application and the Machine Translation Service through the Machine Translation REST API. The question retrieved from the "Get User Question" module or the answer from the "Answers Questions" module and the language of translation is posted as input to the Machine Translation Service. The translated question or answer is posted as an output from the service to the application. Then the connection is terminated.

4) **Answers Questions:**

The objective of this module is get answer for the question in English retrieved from "Translate Data" module.

Connection is established between the application and the Question and Answer Service through the Question and Answer REST API. The question in English retrieved from "Translate Data" module is posted as an input to the Question and Answer Service. The answer of the question is posted as an output from the service to the application. Then the connection is terminated.

5) **Posts Answers to Users:**

The objective of this module is to cut short the answer in user language retrieved from "Translate Data" module to fewer than 140 characters and post to Twitter.

The string length of the answer in user language retrieved from "Translate Data" module is reduced to fewer than 140 characters. Connection is established between the application and Twitter through the Twitter REST API. The answer in user language with fewer than 140 character length is posted to the user id in Twitter. Then the connection is terminated.

Top Level Data Flow Diagram(DFD): Basic Diagram

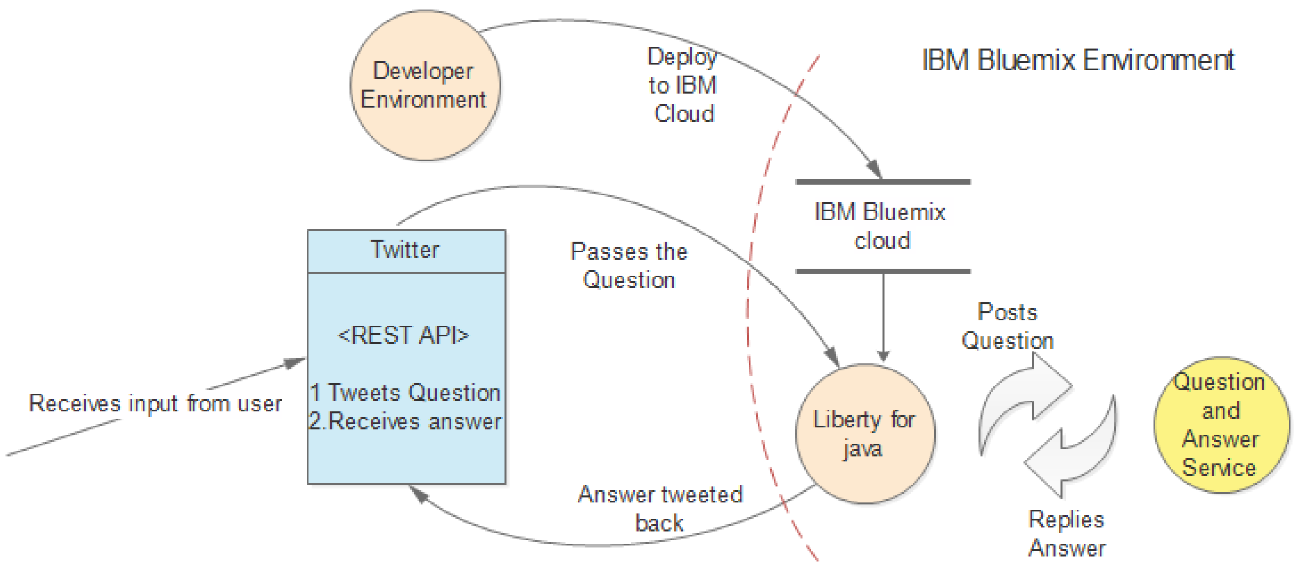
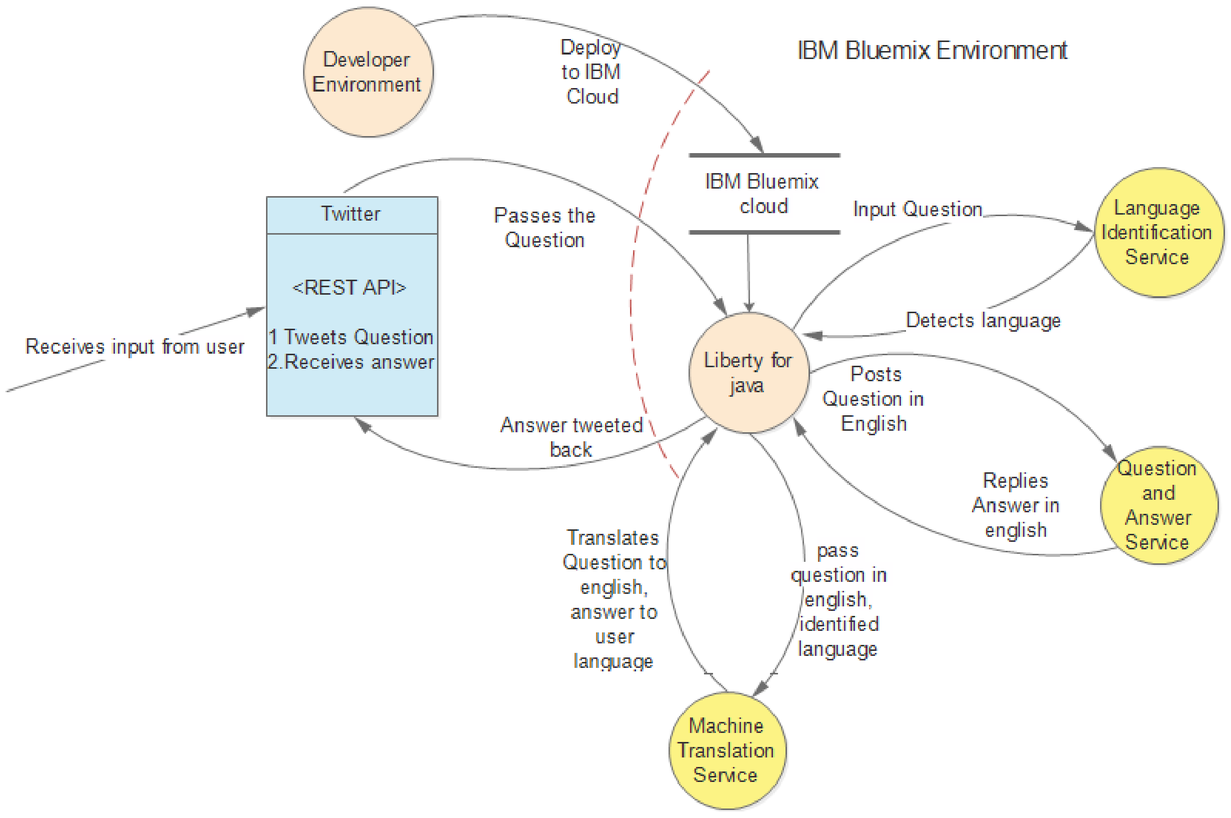


Fig 1: Basic Data Flow Diagram

The Level 0 Data Flow Diagram for basic implementation describes about data flow among the IBM Bluemix data store, external entity Twitter and data process Question and answer service in IBM Bluemix. The privilege boundary (red dotted line) separates the IBM Bluemix environment from other entities.

DFD: For Extended Functionality

The below Data flow diagram explains the further functionality of adding the ability to recognize the language, translating it and sending response back. This DFD shows the data flow between various supportive services like machine translation, Language identification, Question and Answer available in IBM Bluemix Platform.



4. **DATA DESIGN**:

The question from Twitter is passed to the application in Bluemix. This question is posted to the Language Identification Service, which outputs the language of the question to the application. The question and the language of conversion are posted to the Machine Translation Service, which outputs the translated question in English to the application. The question in English is posted to the Question and Answer Service, which outputs the answer in English to the application. The answer and the language of conversion are posted to the Machine Translation Service, which outputs the answer in user language to the application. This answer is reduced to fewer than 140 characters in length and send to user id in twitter.

The data and the application are stored in IBM Bluemix cloud.

5. COMPONENT DESIGN

Pseudo code:

```
//Starts the application
Class TweetContextListener implements ServletContextListener

    Method ContextInitialized
        Start thread TweetListener

//
Class TweetListener implements Runnable

        //Inside Constructor
        1. Configure Twitter account
            Steps:
            1.1.1. Create an object for ConfigurationBuilder class
            1.1.2. Set consumer key for the object with method
                   setOAuthConsumerKey
            1.1.3. Set consumer secret for the object with method
                   setOAuthConsumerSecret
            1.1.4. Set access token for the object with method
                   setOAuthAccessToken
            1.1.5. Set access token secret for the object with method
                   setOAuthAccessToken

            Sample Code:
```

```
            ConfigurationBuilder cb = new ConfigurationBuilder();
            cb.setDebugEnabled(true);
            cb.setOAuthConsumerKey("V0Qpoydtp6M262sTisu2y8SxJ");
        cb.setOAuthConsumerSecret("RSa1xz8vSV4oXCbpIsWhNSrJq1Qzbz
        53vxF8R9HGavoS9puO6c");
            cb.setOAuthAccessToken("3308014853Il0xQCB3tCRJznqy4AxQ6Am
        wTUifUKUSlClPmU7");
        cb.setOAuthAccessTokenSecret("80m0FxwyF7cSQ9wjxbfKvIaVSDh
        31cakCGJEHBgvGUA6C");
```

```
        2. Create an object for TwitterStream class and connect to twitter with the
           configured credentials.
```

Sample Code:

```
twitterStream = new  TwitterStreamFactory(cb.build()).getInstance();
```

//Inside method run()

3. Create an object for StatusListener class.
4. Add this listener object to TwitterStream object.

Sample Code:

```
twitterStream.addListener(listener);
```

5. Create a FilterQuery to filter TwitterStream object from a particular user.

Sample Code:

```
FilterQuery query = new FilterQuery();
query.follow(new long[]{3308076790L});//user id
twitterStream.filter(query);
```

//Inside onStatus method of StatusListener (called if filter condition is satisfied, that is if tweet is send by the user account)

6. Get the tweet from the application account

Sample Code:

```
String tweetQues=status.getText();
```

7. Call appService method from TweetWatson class and pass the tweet as an argument to the method.

//Inside appService method

8. Pass the tweet to the identify method in class Watson. This method returns the language of the tweet.

Sample Code:

```
tweetLang = Locale.forLanguageTag(wt.identify(tweetQues));
```

9. If language of tweet is in English pass the question, proceed to step 12 else continue with step 10.
10. If the language of the tweet is in Spanish, French or Portuguese choose appropriate code word for the language.

Sample Code:

```
    if(userLang!="English")
  {
  switch(userLang)
  {
    case "Spanish":
            translatetoLang="mt-eses-enus";
            break;

    case "French":
            translatetoLang="mt-frfr-enus";
            break;

    case "Portuguese":
            translatetoLang="mt-ptbr-enus";
            break;
  }
```

11. Pass the tweet and the code word of the language to the method translate in class Watson. This method returns the tweet in English.

    Sample Code:

    ```
    userQuestionEng=wt.translate(tweetQues,translatetoLang).toStr   ing();
    ```

12. Pass the tweet in English to QuestionAnswer method in class Watson. This method returns the answer of the tweet in English.

    Sample Code:

    ```
    ansEng=wt.QuestionAnswer(userQuestionEng);
    ```

13. If the tweet was in English, proceed to step 14 else continue with step 15.
14. Repeat step 10 and 11, such that the answer is translated from English to language of the tweet.
15. Divide the answer in to chunks, each of length 140 characters.
16. Post the chunks as tweet of the application id.

//Inside Watson class

17. Get the following credentials for language_identification, machine_translation and question_and_answer services.
    i) url of the service
    ii) username of the service
    iii) password of the service

    Sample Code:

    ```
    private static JSONObject getVcapServices() {
    ```

```
String envServices = System.getenv("VCAP_SERVICES");
sysEnv = JSONObject.parse(envServices);
return sysEnv;          }

private static void processVCAP_Services() {
JSONObject sysEnv = getVcapServices();
if (sysEnv.containsKey(questionAnswerService)) {
JSONArray services   =(JSONArray)sysEnv.get(questionAnswerService);
JSONObject service = (JSONObject)services.get(0);
JSONObject credentials       =(JSONObject)service.get("credentials");
baseURLQuestionAnswer = (String)credentials.get("url");
usernameQuestionAnswer    =(String)credentials.get("username");
passwordQuestionAnswer=(String)credentials.get("password"       );}}
```

//Method questionAnswer
//Parameter : tweet in English
//Returns : answer in English

18. Converts the tweet in English to JSONObject.

    Sample Code:

```
questionJson.put("questionText",question);
JSONObject evidenceRequest = new JSONObject();
JSONObject postData = new JSONObject();
postData.put("question",questionJson);
```

19. Specifies the corpus of the question. The corpus is travel by default
20. Authorizes the account by passing the user credentials of the question_and_answer service.
21. Passes the tweet, which is converted to JSONObject as input to Question and Answer service and gets the answer in the form of a list of JSONObject as the output.

    Sample Code:

```
        Executor executor =
Executor.newInstance().auth(usernameQuestionAnswer,
passwordQuestionAnswer);
        URI serviceURI = new URI(baseURLQuestionAnswer+
"/v1/question/"+dataset).normalize();
        String answersJson =          executor.execute(Request.Post(serviceURI)
        .addHeader("Accept", "application/json")
        .addHeader("X-SyncTimeout", "30")
        .bodyString(postData.toString(),       ContentType.APPLICATION_JSON)
        ).returnContent().asString();
        List<Map<String, String>> answers =         formatAnswers(answersJson);
```

22. Converts the list of answers in JSON format to string.

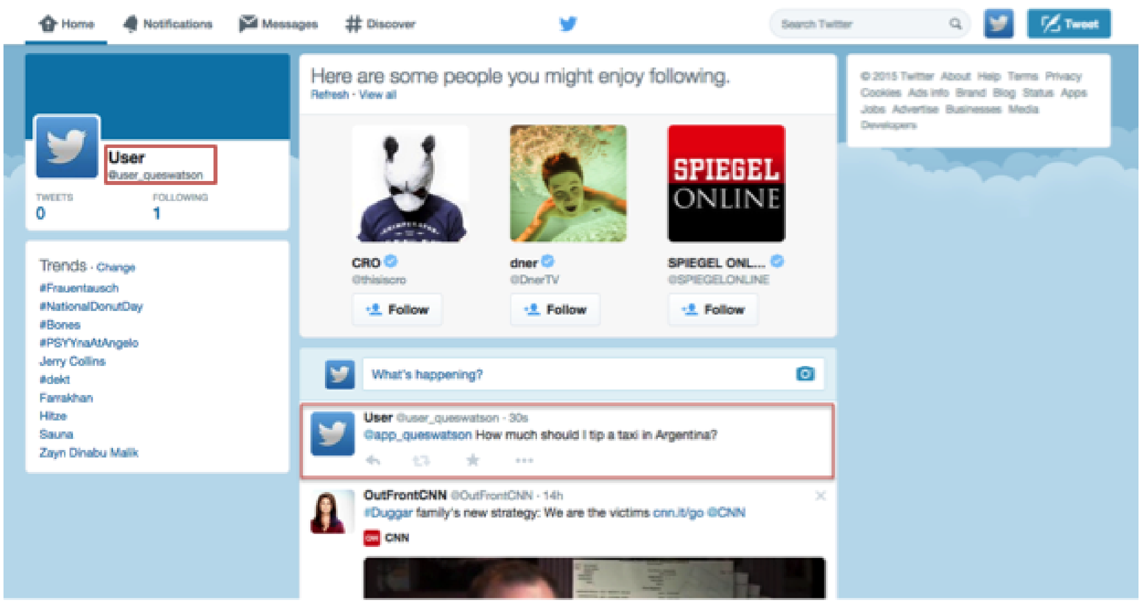
23. Retrieves the first item from the list, which is the answer with highest confidence level and returns it.

//Method identify
//Parameter : tweet
//Returns : language of tweet
24. Passes the tweet as input to language identification service and gets the language as output.

//Method translate
//Parameter : tweet/answer, language conversion code
//Returns : tweet in English/answer in user language
25. Passes the text and language conversion code as input to machine translation service and gets the text in required language as output.

6. HUMAN INTERFACE DESIGN

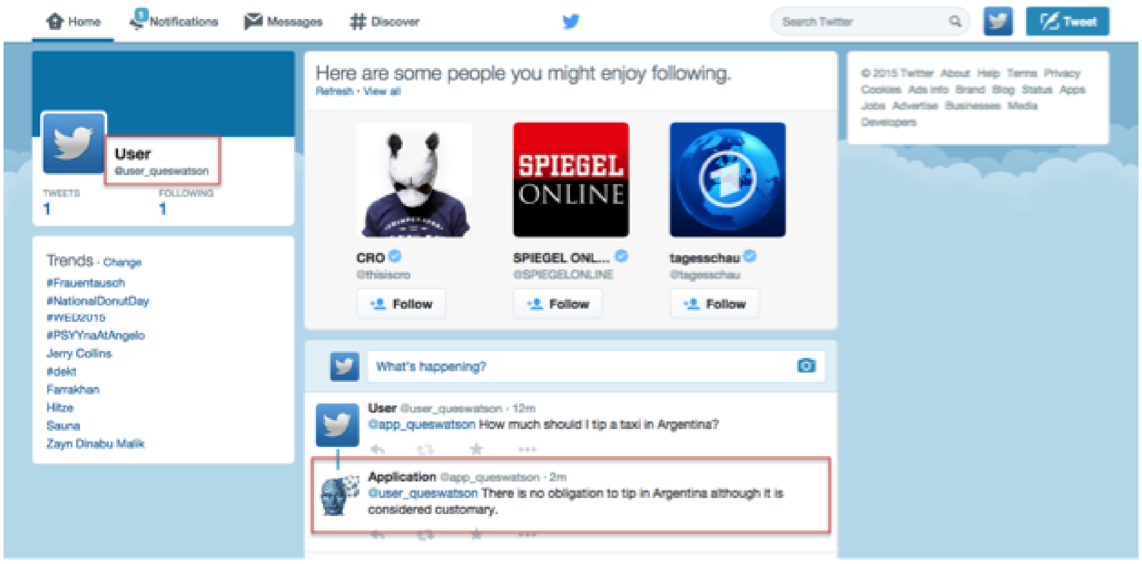
User logs into the user account and tweets a question from the user account to application account in twitter.



User @user_queswatson · 30s
@app_queswatson How much should I tip a taxi in Argentina?

OutFrontCNN @OutFrontCNN · 14h
#Duggar family's new strategy: We are the victims cnn.it/go @CNN

The answer is posted to the user account in twitter.



User @user_queswatson · 12m
@app_queswatson How much should I tip a taxi in Argentina?

Application @app_queswatson · 2m
@user_queswatson There is no obligation to tip in Argentina although it is considered customary.

7. CONTACT DETAILS

Developers	E-mail IDs
Krishna Damarla	krishnadamarla1@gmail.com
Semanti Kundu	semanti.kundu@gmail.com
Shalini Chellathurai Saroja	shalini.chellathurai@gmail.com

8. REFERENCES

1. http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/qaapi/corpora.shtml#corpora
2. https://www.ng.bluemix.net/docs/
3. https://www.ng.bluemix.net/docs/overview/overview.html
4. https://www.ng.bluemix.net/docs/starters/liberty/index.html#optionsforpushinglibertyapplications
5. https://www.ng.bluemix.net/docs/services/QuestionAnswer/index.html
6. https://www.ng.bluemix.net/docs/services/LanguageIdentification/index.html
7. https://www.ng.bluemix.net/docs/services/MachineTranslation/index.html

8. http://en.wikipedia.org/wiki/Twitter
9. https://dev.twitter.com/rest/public