Massmine for the Masses

Project Team: Patricia Tanzer, Morgan McIntyre, Logan Hornbuckle, Joshua Moore

May 2019

Table of Contents

1. Project Definition

Massmine-for-the-Masses is an open source, proof of concept tool intended to extend the functionality of the command line big-data gathering tool Massmine (www.massmine.org) to users unfamiliar or uncomfortable with the command line. It may be installed as a Docker image in a Docker container, or installed manually onto a compatible Linux machine. The project itself is a web application built with the Django framework, using Django's default test server and SQLite database. As a result, this project is not suitable for use outside a protected network and can only fulfill the needs of a limited number of researchers. (No more than five anticipated, and less if each attempts to use the tool to store large data sets.) The user's private Twitter Developer keys are encrypted in our database, and each user may only view and run analyses on the tweets he/she has collected.

The intended use of this tool is to create user accounts which contain the user's private Twitter Development key, and allow these users to interact with a simple graphical user interface to request a keyword or phrase and a desired number of tweets to gather, sending this information to the Massmine tool for processing. The Twitter data (consisting of zero to many tweets) received back from the tool is parsed and stored into a local database. The user may further interact with this data by running some basic analyses on the data gathered, or simply view the tweets, all from the web application.

2. Project Requirements

2.1 Functional Requirements

2.1.1 Store User Information

Required Data: username, hashed and salted password, encrypted Twitter authorization keys.

Optional Data: first name, last name

2.1.2 Store Collected Twitter Data

Data including but not limited to: tweet id, creation time, text, author name, hashtags.

2.1.3 Create and Update New Users

Users must be able to change their authorization keys, email, and password.

2.1.4 Allow Users To Gather Twitter Data

Each set of Twitter data collected by a user is called a 'study'. There must be a way to view the individual entries of this study.

2.1.5 Allow Users To Perform Basic Analysis on Twitter Data

Analyses include: Sort by data and time, view the top five most frequent associated words, see the overall sentiment associated with a key term.

2.2 Usability

2.2.1 User Interface

Login Interface: Should allow the creation of accounts, a login page, and a way to recover forgotten passwords.

User Account Interface: Should allow the user to enter or change their Twitter authentication keys, email, and password.

Query Interface: Allow the user to run a query for a given keyword or phrase, with a given number of tweets, resulting in the creation of a ‘study’.

Study Interface: Should allow user to list the studies to which they have access and allow them to select one of their choice for analysis.

Analysis Interface: Allow the user to list the possible analyses to run and run a selected analyses on a selected study.

2.2.2 Performance

Ideally, the user should not notice any significant time delay when running analyses on data already gathered. Depending on the query, data likely will not be available immediately after a query is given, but this should not prevent the user from signing out, back in, or running additional queries and analyses in the meantime. However, this feature was not implemented for the proof-of-concept project, meaning that when a query is started, other simultaneous operations are not recommended.

Multiple users should be able to use the project, however no more than five accounts are expected per project. Additional database space is required to extend this project for larger use.

2.3 System

2.3.1 Hardware

Either:

a) a computer running a compatible Linux distribution (Ubuntu 18.04 preferred) for manual installation

b) a computer capable of running a Docker container for the Docker image of this project to run on

2.3.2 Software

This is a web application for the Django framework, programmed in Python and HTML.

2.3.3 Database

SQLite, provided by Django framework. While SQLite has a theoretical database size limit at 140TB, it does store its database as a single file, which means that users may encounter size restrictions based on their file system. If more than about 4GB is desired, it may be necessary to either create a new container or use a different back-end database.

2.4 Security

Passwords are salted and hashed with the default Django libraries.

User input is guarded against cross site request forgery with a Django token. Django by default scrubs input for SQL injection.

The default server used by Django, which is implemented for this project, does not support HTTPS. As a result, this project is not suited for public or insecure networks.

3. Specification

3.1 Focus

Develop an open-source, web-based application to collect and analyze social network data using Massmine.

3.2 Domain

The intended users for this application are academic researchers and universities that may host their own version of this project. This application should appeal to those interested in social media research that lack the necessary command-line skills to fully utilize Massmine.

3.3 Area

Big Data/Data Analysis, Data Mining/Text Mining, Sentiment Analysis, Database management, Web Application

3.4 Framework

Web Framework: Django 2.1.7, a high-level Python web framework.

Database: SQLite, a small database which is the default for Django and suitable to the limited needs of a proof-of-concept project.

Server: We used the default testing server for Django, which does not support HTTPS or other advanced features.

3.5 Platform

This project is meant for desktop access on a protected network. The Docker image platform allows for usage on any operating system; a manual installation requires a Linux-based operating system (Ubuntu preferred). Users desiring to make this a public-facing web page will need to download from source and set up an HTTPS server for security, and either an SMTP server or a third part SMTP service for the forgot password functionality.

3.6 Genre

Massmine-for-the-Masses is a web application layered over the command-line tool Massmine.

3.7 Required Installations

Massmine 1.1.0

Python 3

Django 2.1.7

Pip 19.03

Numpy 1.16.2

Matplotlib 3.0.3

Plotly 3.7.1

Cufflinks 0.15

Pandas 0.24.2

Celery 4.3.0 (rhubarb)

Erlang OTP 20 (support for asynchronous expansion)

Rabbitmq-server 3.6.10 (support for asynchronous expansion)

Django-celery-results 1.0.4 (stable)(support for asynchronous expansion)

Django-encrypted-model-fields 0.5.8

Django-tables2 2.0.6

Pexpect 4.2.1

Textblob 0.15.3

3.8 Development Environment

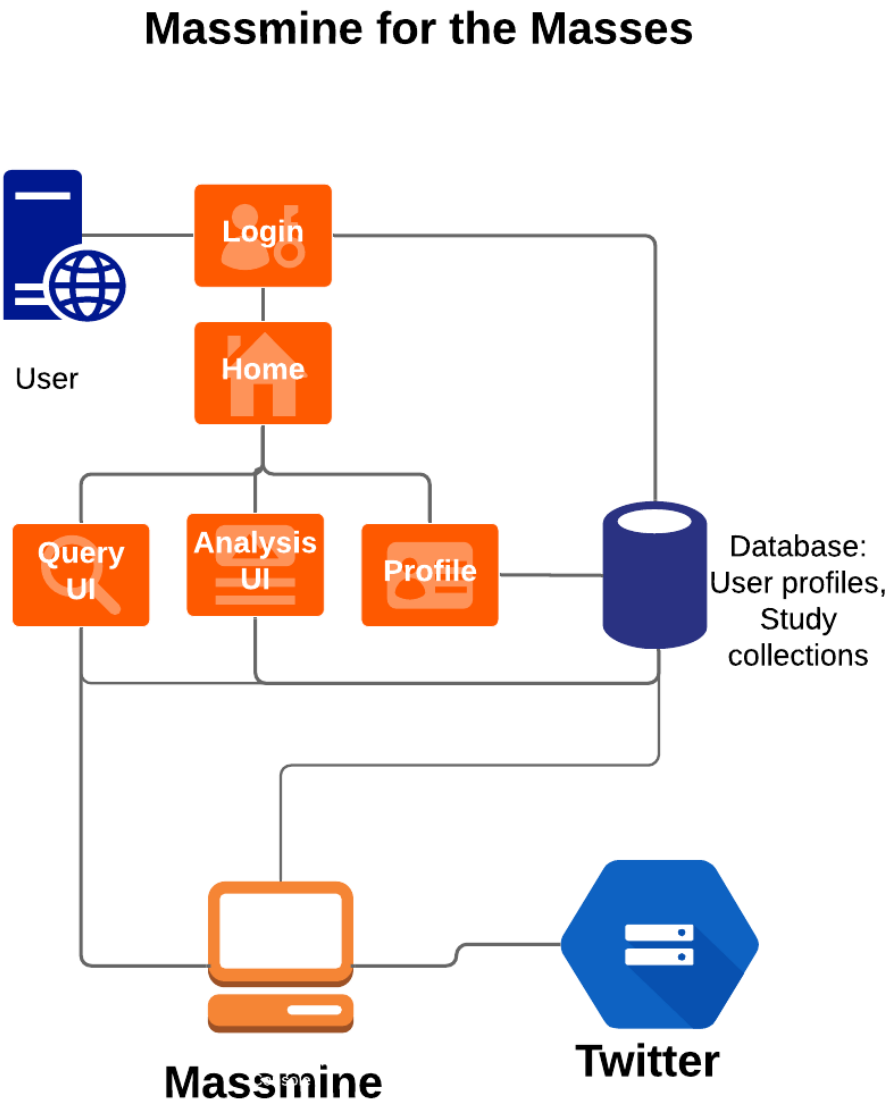
Pip virtualenv

Ubuntu 18.04 host (virtual machine)

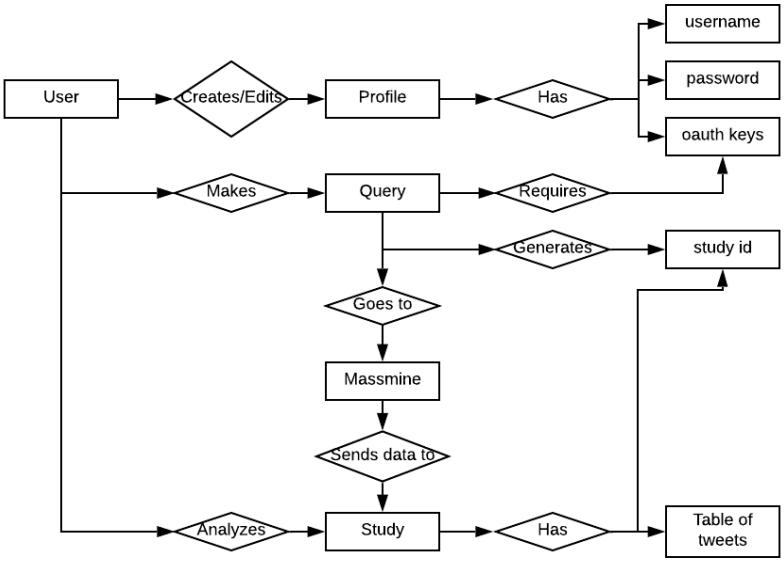
4. Overall Design

4.1 Overall Operation and Diagrams

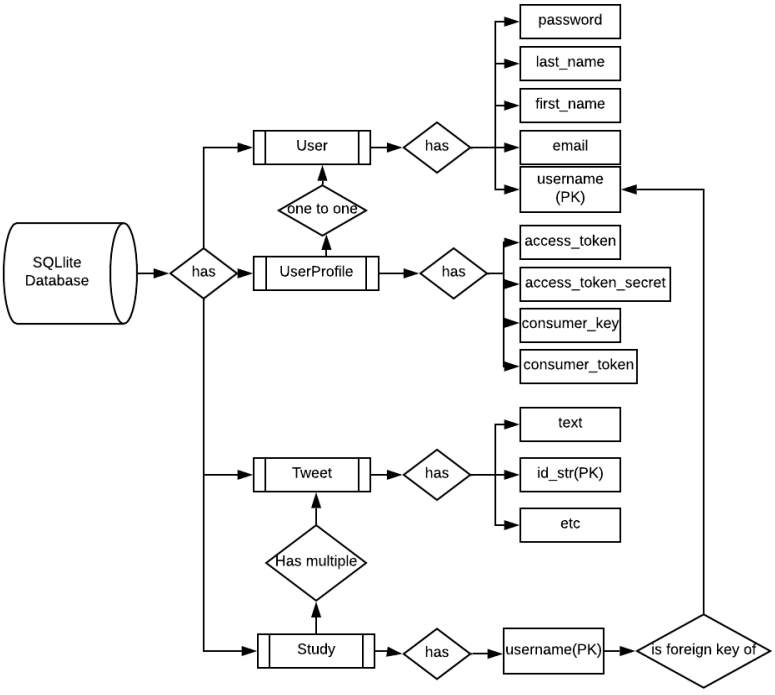
4.1.1 System Model



4.1.2 System Entity Relationship Model



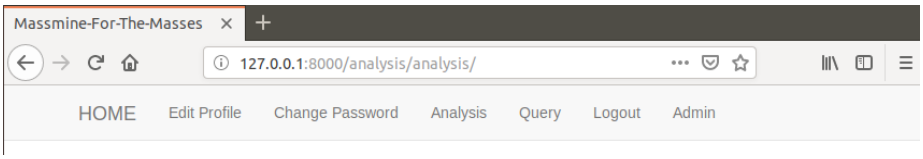
4.1.3 Database Entity Relationship Model



4.1.4 User Interface Design

Style: Using a white and gray bootstrap CSS: https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css

Navigation: Each page is accessible from a navbar at the top of the screen. This navbar works equally well full-screen or in a smaller window.



4.2 Subsystem Overview

4.2.1 User/Administration Overview

Pages: Home, User Profile, Registration, Edit Profile, Change Password, Login/Logout, Administration

Manages: User credentials and personal information

Subsystem Communication: Query, Study

4.2.2 Query Overview

Pages: Query

Manages: Starting a user-generated query with Massmine

Subsystem Communication: User/Study

4.2.3 Study Overview

Pages: Analysis (view studies)

Manages: Parsing and storing data gathered by Massmine and linking it to individual users, then displaying the study to the user who gathered it.

Subsystem Communication: User, Query

4.2.4 Analysis Overview

Pages: Analysis

Manages: Analyzing selected study with user-chosen analyses and displaying the results

Subsystem Communication: Study

5. Subsystems

5.1 User/Administration

5.1.1 Pages

Home page: Gives options to register, login, recover a forgotten password, or go to the administration page (requires authentication).

Register: Require a username, email, consumer key, consumer secret, access token, and access token secret. Requires a password that meets basic security standards (cannot be the same as username, cannot be entirely numeric, must be at least eight characters, and cannot contain restricted characters). Only available when logged out.

User Profile: Allow user to change email, password, consumer key, consumer secret, access token, access token secret, first name, and last name. The four access keys are encrypted at rest in the database. Only available when logged in.

Edit Profile: Allow user to change everything in their profile except their username and password. Only available when logged in.

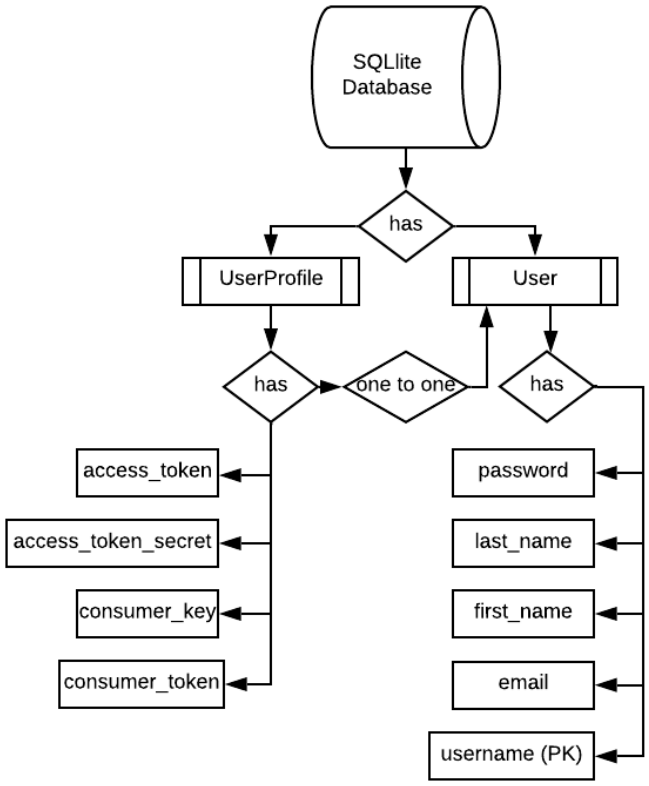
Change Password: Allow user to change their password to another that meets the same requirements as at registration. Only available when logged in.

Login/Logout: The user should be able to log out from any page. The login option is only available when the user is logged out, and the logout option is only available when the user is logged in.

Forgot Password: If a user forgets their password, they may go to this password and generate a link to allow them to reset it. In this project, this link is sent to a folder in the project directory, rather than to an email. Only available when logged out.

Administration: Be able to edit, delete, and create users, as well as see all user information. This page is only accessible when logged in as an administrator account.

5.1.2 Database design



5.1.3 Data dictionary

User model: Using default Django model.
https://docs.djangoproject.com/en/2.1/ref/contrib/auth/

Attribute	Description	Type
Username	Name identifying a user	CharField, max length 150. May contain alphanumeric, @, +, . and - characters. Primary key.
password	Hashed and salted password to confirm user login	CharField.
first_name	Optional detail if users wish to personalize their accounts.	CharField. 30 characters or fewer.
last_name	Optional detail if users wish to personalize	CharField. 150

	their accounts.	characters or fewer.
email	Optional detail if users wish to personalize their accounts. In future releases could be used to provide notifications and recover password support.	CharField, formatted as email address.
is_staff	Designates whether this user can access the admin site. Admin viewable only.	Boolean.
is_active	Designates whether user is active. May use instead of deleting user. Admin viewable only.	Boolean.
is_superuser	Designates that this user has all permissions. Admin viewable only.	Boolean.
last_login	Datetime of the user's last login. Admin viewable only.	DateTime
date_join	Datetime of when account was created. Admin viewable only.	DateTime

UserProfile: Stores the four Twitter Developer keys specific to each user.

Attribute	**Description**	**Type**
User	One-to-One link to user model	Primary Key
consumer_key	Twitter authorization key.	EncryptedCharField, max_length 50
consumer_secret	Twitter authorization key	EncryptedCharField, max_length 50
access_token	Twitter authorization key	EncryptedCharField, max_length 50
access_token_secret	Twitter authorization key	EncryptedCharField, max_length 50

5.1.4 Use Cases

Registration Scenario:

Case: Creating (registering) a new user

Actor: User named Jane, not logged in or created

Scenario: Jane would like to create an account on the Massmine-for-the-Masses website. She goes to the desktop where the application is running, and clicks ‘Register’ on the navbar at the top of the webpage. She enters a username (‘Jane’) and email, her password twice, and all four Twitter authorization keys. She leaves the first name and last name fields blank. Then she clicks ‘Submit’ at the bottom of the page, and is returned to the home page. Then she clicks ‘Login’ at the top of the navbar, and enters her username and password on the login screen. This time when she hits submit, she is returned to the Home screen but the message ‘Hello Jane’ appears on the screen and she can see new navbar links, such as Edit Profile, Change Password, Analysis, Query, Logout, and Admin.

Change Information Scenario:

Case: Changing a user’s information (editing a user’s profile)

Actor: User named Jane, currently logged in

Scenario: Jane realizes that she put the wrong numbers in for her Twitter consumer key. She clicks ‘Edit Profile’ on the navbar at the top of the screen, and the screen displays her email address, first name, last name, and all four Twitter authorization keys. She clicks on the consumer key text field, and edits the field to contain the correct information. Then she clicks ‘Submit’ at the bottom of the page, and is returned to the home screen.

Change Password Scenario:

Case: Changing a user’s information (editing a user’s password)

Actor: User named Jane, currently logged in

Scenario: Jane realizes that ‘janepassword’ is not a very secure password, and wants to change it. She clicks ‘Change Password’ on the navbar at the top of the screen, and the screen prompts her for her old password, a new password, and a confirmation of the new password. It also displays instructions that the password can’t be too similar to her other personal information, must contain at least 8 characters, can’t be a commonly used password, and can’t be entirely numeric. Jane enters ‘janepassword’ as the old password and ‘password1’ as the new password and confirmation, and is given an error page which tells her the new password does not meet the requirements. She clicks on ‘Change Password’ again, and this time enters ‘janepassword’ as the old password and ‘thisisapasswordforjane’ as the new password and password confirmation. This time, she is returned successfully to the home screen.

Deleting A User Scenario

Case: Using admin powers to delete/edit a user (deleting a specific user)

Actor: Admin user named Bob, currently logged out.

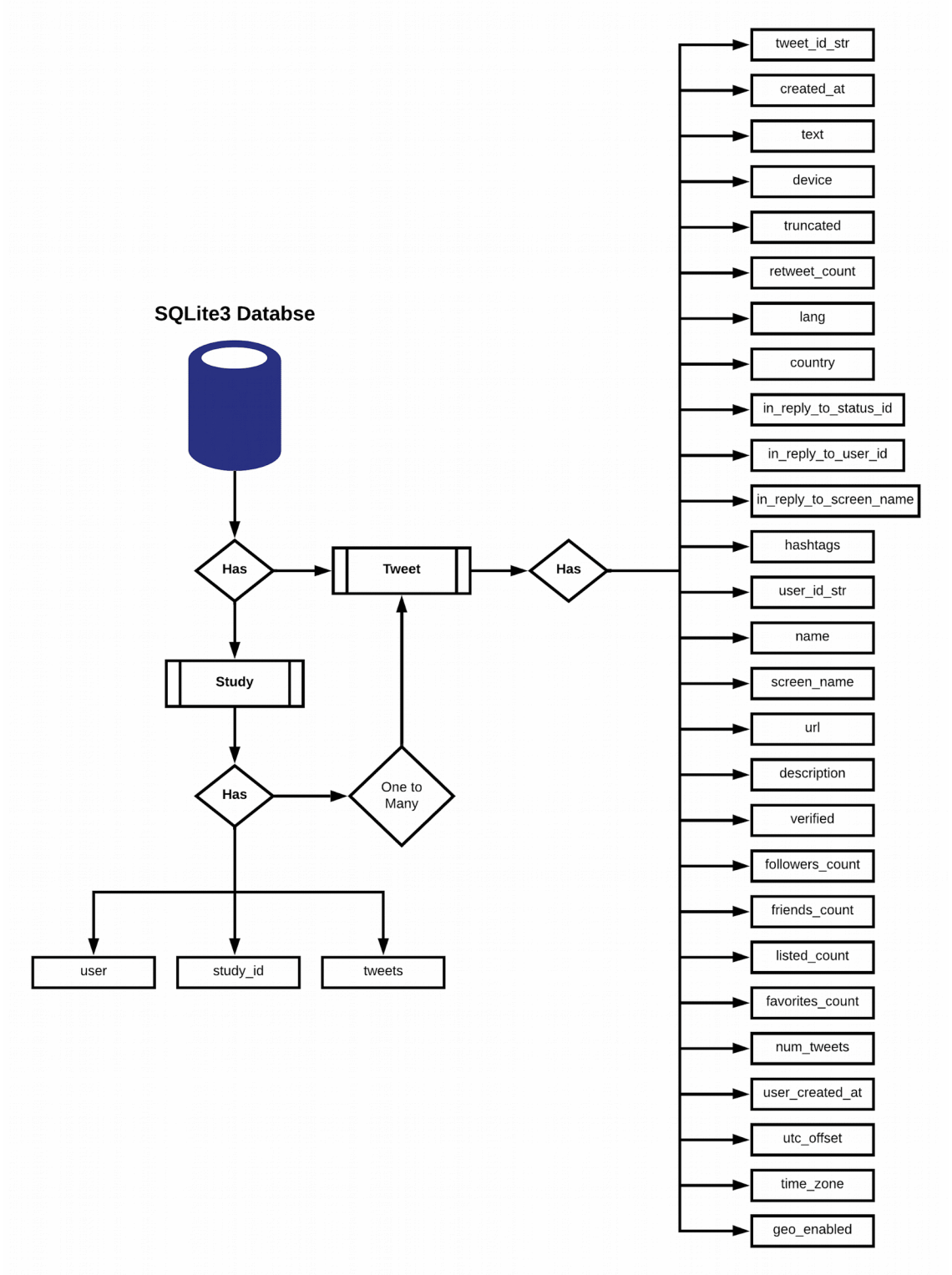
Scenario: Bob, an admin in the department, is told that Jane has graduated three years ago and is no longer using the account she made. He logs in to the webpage with his basic user account and clicks on ‘Admin’ on the navbar at the top of the page. He is prompted for an admin username and password, which he enters. He clicks on ‘Users’ under the ‘Authentication and Authorization’ tab, then clicks on the ‘Jane’ username. He scrolls to the bottom of the page, and clicks the large red button labeled ‘Delete’. The page prompts for confirmation, and he clicks ‘Yes, I’m sure’. He is then returned to the administration page, where a notification at the top of the screen says ‘The user ‘Jane’ was deleted successfully. Bob can see that Jane is no longer in the list of users, so he clicks ‘Log Out’ at the top of the page and goes to get coffee.

5.2 Query

5.2.1 Pages

Query Page: Allow user to enter a keyword or phrase of their choice, and the number of most recent tweets containing that keyword or phrase to gather. This page must also check the user’s Twitter authorization keys for validity and inform the user if the keys are invalid. If the keys are valid, the user’s keyword or phrase and number of tweets is formatted and sent to Massmine to begin the query process. Only available when logged in.

5.2.2 Database Design



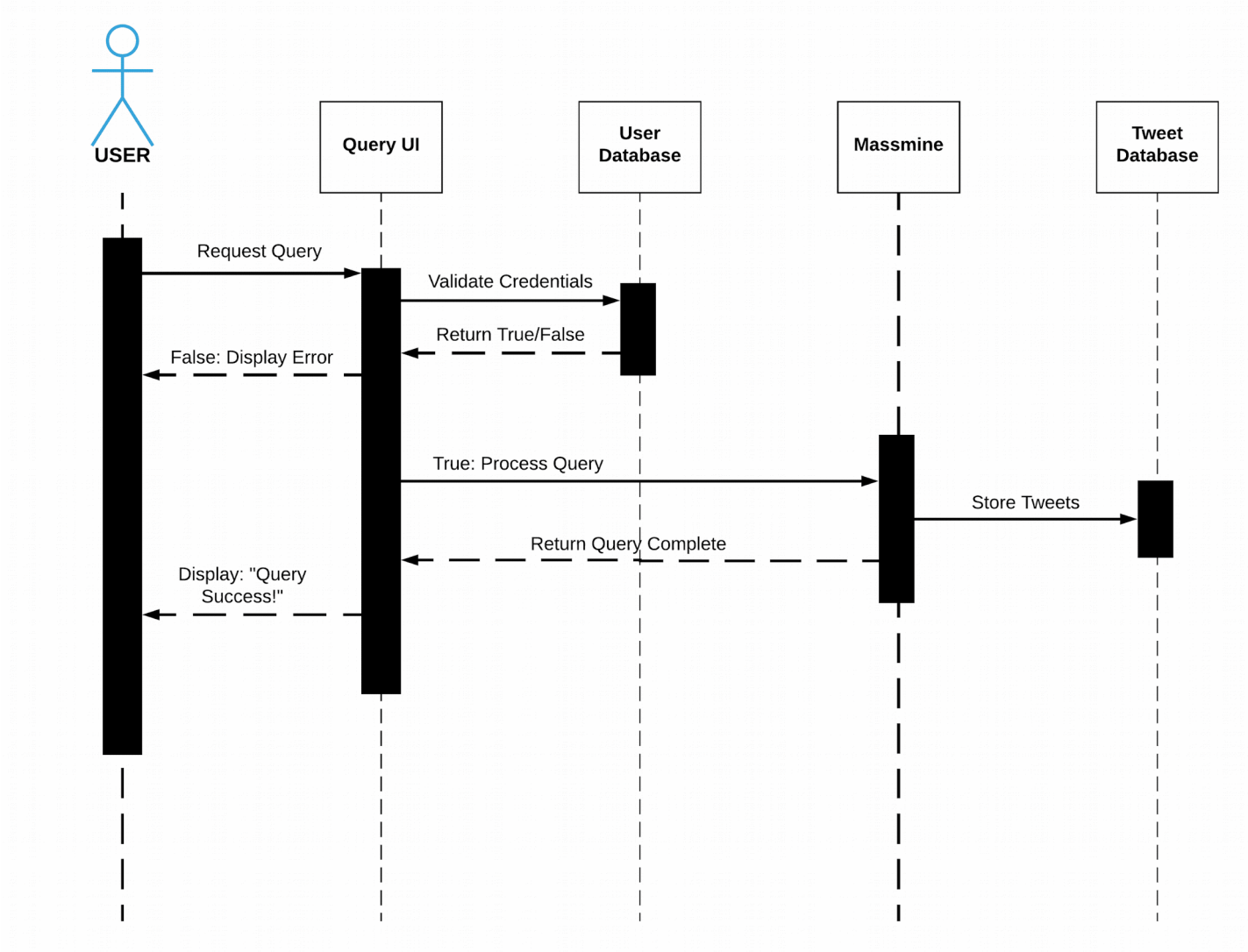
5.2.3 Data Dictionary

Tweet Model:

Attribute	Description	Type
created_at	UTC time when this Tweet was created.	CharField
tweet_id_str	The string representation of the tweet's unique identifier.	CharField
text	The actual UTF-8 text of the status update.	TextField
device	Utility used to post the Tweet.	CharField
truncated	Indicates whether the value of the text parameter was truncated.	BooleanField
retweet_count	Number of times this Tweet has been retweeted.	IntegerField
lang	Nullable. When present, indicates a BCP 47 language identifier corresponding to the machine-detected language of the Tweet text, or "und" if no language could be detected.	CharField
country	Nullable When present, indicates what country the tweet is associated.	CharField
in_reply_to_status_id_str	Nullable. If the represented Tweet is a reply, this field will contain the integer representation of the original Tweet's ID.	CharField
in_reply_to_user_id_str	Nullable. If the represented Tweet is a reply, this field will contain the string representation of the original Tweet's ID.	CharField
in_reply_to_screen_name	Nullable. If the represented Tweet is a reply, this field will contain the screen name of the original Tweet's author.	CharField
hashtags	Hashtags used. Found within entities which have been parsed out of the text of the Tweet.	CharField
user_id_str	The string representation of the unique identifier for this User.	CharField
name	The name of the user, as they've defined it. Not necessarily a person's name.	CharField
screen_name	The screen name, handle, or alias that this user identifies themselves with.	CharField
url	Nullable . A URL provided by the user in	CharField

	association with their profile.	
description	Nullable . The user-defined UTF-8 string describing their account.	CharField
verified	When true, indicates that the user has a verified account.	BooleanField
followers_count	The number of followers this account currently has. Under certain conditions of duress, this field will temporarily indicate “0”.	IntegerField
friends_count	The number of users this account is following (AKA their “followings”). Under certain conditions of duress, this field will temporarily indicate “0”.	IntegerField
listed_count	The number of public lists that this user is a member of.	IntegerField
favorites_count	The number of Tweets this user has liked in the account’s lifetime.	IntegerField
num_tweets	The number of Tweets (including retweets) issued by the user.	IntegerField
user_created_at	The UTC datetime that the user account was created on Twitter.	DateTimeField
utc_offset	Value will be set to null.	CharField
time_zone	Value will be set to null.	DateTimeField
geo_enabled	When true, indicates that the user has enabled the possibility of geotagging their Tweets. This field must be true for the current user to attach geographic data when using POST statuses / update.	BooleanField

5.2.4 Dataflow Diagram



5.3 Study

5.2.1 Pages

Analysis (view studies): On the analysis page, the user must be able to select a study which they have created, and not any which they have not created, from a drop-down menu. They should then be able to view the entire study, including all the available fields for each tweet in that study, by choosing a second drop-down menu. Only available when logged in.

5.2.2 Database Design

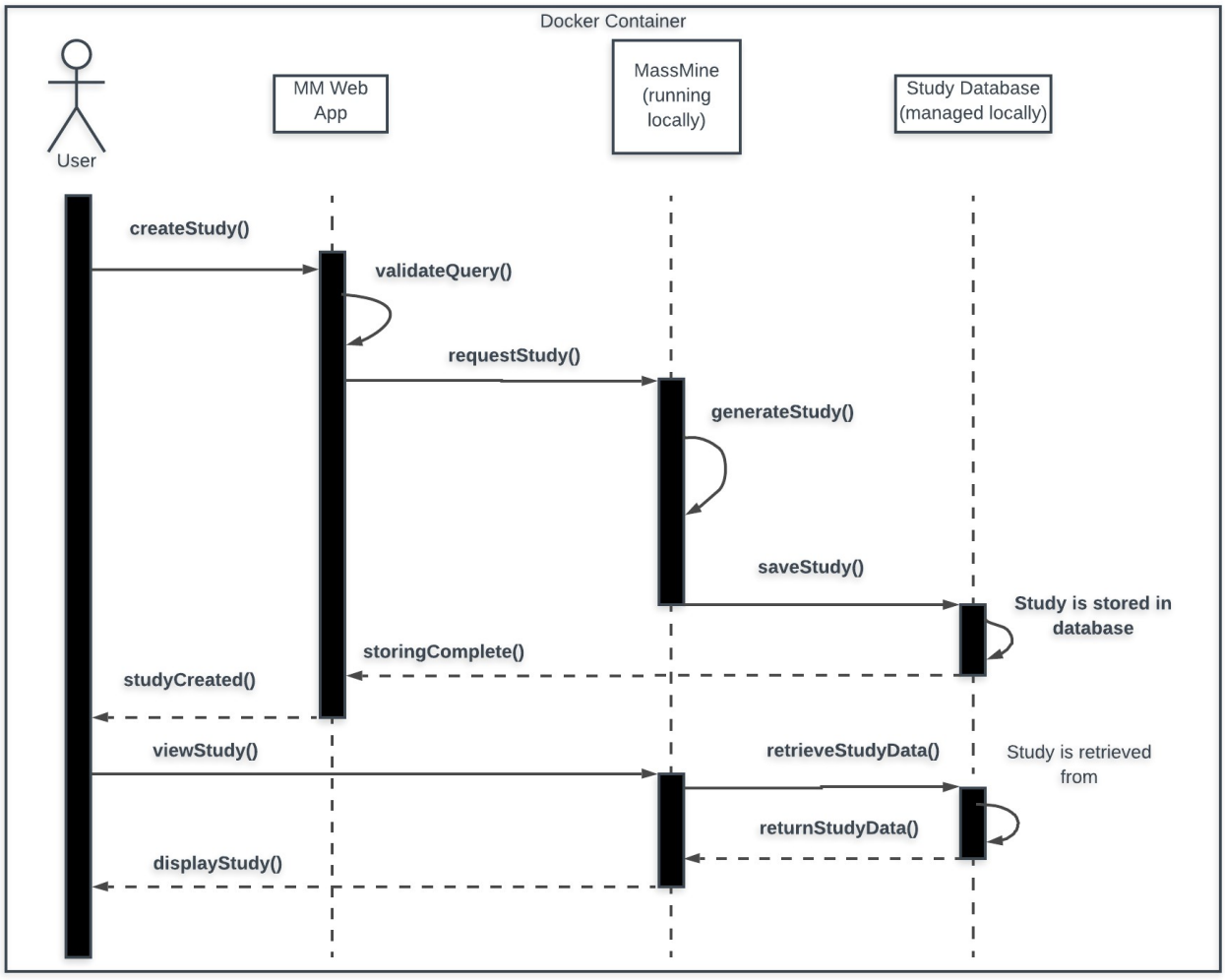
See 5.2.2

5.2.3 Data Dictionary

Study Model:

Attribute	Description	Type
user	One-to-One link to User model	CharField, max length 60
study_id	Unique id of a study	CharField, max length 100
Tweets	Table of Tweets	ManyToManyField

5.2.4 Dataflow Diagram



5.4 Analysis

5.2.1 Pages

Analysis: On this page, the user should be able to select one of the analysis options and a study on which to perform said analysis, and have the results displayed in a new tab. Only accessible when logged in.

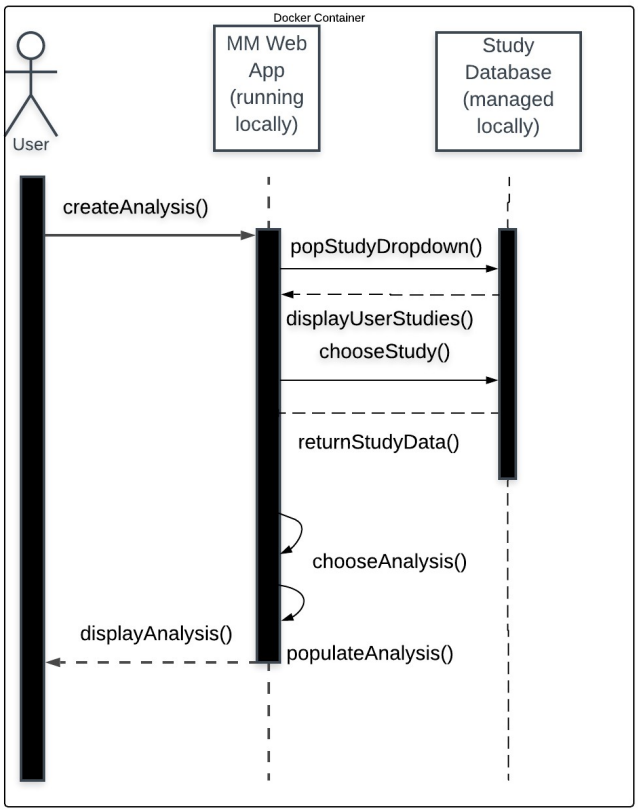
5.2.2 Analysis Options

Frequent Words Analysis: Displays the five most frequent words associated with the keyword or phrase, with the number of frequencies. Uses a list of stop words, stemming, and the bag of words technique.

Sentiment Analysis: Display the number of tweets containing the keyword or phrase that are judged to be positive, negative, or neutral in sentiment in a bar graph.

Sort By Date: Display a line graph showing on which the date and time each tweet containing the keyword or phrase was posted.

5.2.3 Dataflow



5.2.4 Use Cases

View Frequent Words Scenario:

Case: View the top five most frequently used words in a study

Actor: User who is logged in. Their username is SallyJ

Scenario: SallyJ (after logging in and creating a study) navigates to the analysis page. She first selects a study she queried for from the drop-down menu. SallyJ then selects the frequent word analysis from the second drop-down menu. She is taken to a new tab

which displays the top five most frequently used words based on the original keyword queried for. SallyJ zooms into the map to see it in more detail. She also right-clicks and saves the graph as an image so she can review it at a later date.

View Sentiment Analysis Scenario:

Case: View the overall sentiment of a study

Actor: User who is logged in. Their username is Drew95

Scenario: Drew95 (after logging in and creating a study) navigates to the analysis page. He first selects a study he queried for from the drop-down menu. Drew95 then selects the sentiment analysis from the second drop-down menu. He is taken to a new tab which displays the overall sentiment (how many tweets are positive, negative, or neutral) based on the original keyword queried for. He hovers his mouse over one of the columns to see how many tweets were labeled as ‘negative’.

View Activity Scenario:

Case: View the activity of a study

Actor: User who is logged in. Their username is CraigS

Scenario: CraigS (after logging in and creating a study) navigates to the analysis page. He first selects a study he queried for from the drop-down menu. CraigS then selects the date analysis from the second drop-down menu. He is taken to a new tab which displays the timeline of the tweets being posted based on the original keyword queried for. He hovers his mouse over one of the nodes to see how many tweets were posted at this particular date and time. He clicks and drags the graph right and left to see the timeline better.