Test Report
Sentiment Analysis with Twitter
Team Jazz Men

Anagh Goswami (1217426)
Meet Pandya (1214306)
Jasman Gill (1211554)
Jesse Truong (1222722)
Jia Xu (1213268)

March 26, 2017

Contents

Revision History

Date	Author	Comments
March 26, 2017	Anagh Goswami	Created first draft.

1 Introduction

The following report addresses the system tests and non-functional tests on the Social Media Sentiment Analysis web application. The system tests are reported based on each individual module. Non-functional tests include tests on usability, performance, and robustness.

2 Manual vs. Automated Testing

Note that the User Acceptability Testing was completed before the application was uploaded to the servers in the form of system tests and non-functional tests. Given this, there are a couple of reasons that manual testing was chosen over automated testing.

Since the application is using the client-server model, there are many components that need to work for the website to go live. Since these components are core requirements, testing is unnecessary due to the website not being live. There is no possibility for the code to change on its own volition, therefore once the website is live it will remain live until a server side issue occurs or a functionality bug appears. The testing required to ensure that the JavaScript and server-side python functions worked were simple enough to check manually, and security of user input fields were considered during the code implementation phase.

Apart from user acceptability functions, website functionality such as custom searches cannot be automated since they require a human's level of comprehension to check for accuracy of results. The web application was intended for business analysts to take the results and work with data collected. Even though an automated test could tell us that the data has been collected, the correctness of the sentiment score given for the words of the resulting twitter data needed to be benchmarked against human sentiments.

Since the web application collects data only when req uested, it does not have a critical need to function all the time. This allows some leeway with robustness testing, since a technical issue can be handled when reported after deployment. If it is critical for the website functions to be live, automated tests can be developed in the future to test them on a more frequent basis. The user base will have easy modes of communications with the developers since this website is a product targeted as a business tool rather than an open-market consumer tool. Due to this philosophy, the web application is not planning on tracking usage data such as the number of hits on the website, rather it is focusing on functionality and customizability.

Automated unit tests will be created, updated and modified as needed throughout the development phase, and even after the deployment phase. In practice, these unit tests will be used to ensure that updating the code base will not break the system.

3 System Tests

This section describes the test cases carried out on each individual module. The trivial cases for some modules are not described with a high level explanation. Additional details are provided when necessary. The test cases in this document relate to the requirements specified at the beginning of the project, however they are not directly mapped to the software requirements document.

3.1 User Registration

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
1.1	User Registration	Registration page. Empty fields.	Username and password entered. Clicks register.	Redirected to login page.	As expected.	PASS
1.2	User Registration	Registration page. Empty fields.	Empty field(s). Clicks register.	Stays on the same page. No registration	As expected.	PASS

3.2 User Login

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
2.1	User Login	Login page. Empty username and password fields.	Valid username and password combination. Clicks login.	Redirected to quick search query.	As expected.	PASS
2.2	User Login	Login page. Empty username and password fields.	Invalid username and password combination. Clicks login.	Stays on the same page.	As expected.	PASS

2.3	User Login	Login page. Empty username and password fields.	Empty username and/or password fields. Clicks login.	Stays on the same page.	As expected.	PASS

3.3 Main Page Menu

No.	**Test Case**	**Initial State**	**Input**	**Expected Output**	**Actual Output**	**Result**
3.1	From menu choose ’capstone’.	Home page.	User selects ’capstone’ from menu	Traverses to appropriate section on main page.	As expected.	PASS
3.2	From menu choose ’home’.	Home page.	User selects ’home’ from menu	Traverses to appropriate section on main page.	As expected.	PASS
3.3	From menu choose ’how it works’.	Home page.	User selects ’how it works’ from menu	Traverses to appropriate section on main page.	As expected.	PASS
3.4	From menu choose ’examples’.	Home page.	User selects ’examples’ from menu	Traverses to appropriate section on main page.	As expected.	PASS
3.5	From menu choose ’contact us’.	Home page.	User selects ’contact us’ from menu	Traverses to appropriate section on main page.	As expected.	PASS

3.4 Demo Selector

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
4.1	Loading up sample demo.	No demo selected.	User clicks on 'TV shows'	Load sample items choices	As expected	PASS
4.2	Loading up sample demo.	No demo selected.	User clicks on 'University'	Load sample items choices	As expected	PASS
4.3	Loading up sample demo.	No demo selected.	User clicks on 'Restaurant'	Load sample items choices	As expected	PASS
4.4	Loading up sample demo.	No demo selected.	User clicks on 'Athlete'	Load sample items choices	As expected	PASS

3.5 Selecting Individual Demo

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
5.1	Select item choice from demo.	Selected demo category.	Click item choice.	Expand item choice with twitter searches and associated positive and negative scores	As expected.	PASS

3.6 Table Sorter

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
6.1	Table sorter.	Expanded item choice.	Clicks ’users’ on first row of table.	Sorts table from lowest to highest and vice versa if clicked again.	As expected.	PASS
6.2	Table sorter.	Expanded item choice.	Clicks ’followers’ on first row of table.	Sorts table from lowest to highest and vice versa if clicked again.	As expected.	PASS
6.3	Table sorter.	Expanded item choice.	Clicks ’tweet’ on first row of table.	Sorts table from lowest to highest and vice versa if clicked again.	As expected.	PASS
6.4	Table sorter.	Expanded item choice.	Clicks ’score’ on first row of table.	Sorts table from lowest to highest and vice versa if clicked again.	As expected.	PASS

3.7 Custom Search

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
7.1	Search keyword	Custom search page. Field empty	User types keyword and clicks ’search’ button	Nothing is searched, remains on same page	As expected.	PASS
7.2	Search keyword	Custom search page. Field filled	User types keyword and clicks ’search’ button	Keyword is queried and tables are generated dynamically below search field	As expected.	PASS

3.8 Result Accuracy

No.	Test Case	Initial State	Input	Expected Output	Actual Output	Result
8.1	Results Accuracy	Dynamically generated results table	User typed keyword is checked in all table entries	User typed keyword is visible in all resulting tweets	As expected.	PASS
8.2	Results Accuracy	Dynamically generated results table	Resulting tweets have a positive/negative sentiment score	Sentiment score relates to general idea of the corresponding tweet	As expected.	PASS

4 Non-Functional Tests

4.1 Usability

The usability of the system was tested through team members creating accounts for themselves and log in on the website. They were tasked to create a customized keyword search and observe the output. The team members then checked for the keywords in the outputted table along with their respective sentiment scores. Thereafter, if the user read through the resulting tweets and judged their scores as correctly positive or negative, it was considered a pass.

4.2 Performance

To test our system performance, we looked towards the time the system spent loading the users search. Any time that was minimal, for example in a couple of seconds, it was deemed to be reasonable and within in our scope of acceptance. We had team members search up custom searches and found the time spent was indeed to our liking.

4.3 Robustness

The Social Media Sentiment Analysis web application is tested against various browsers and hardware devices to ensure that it will work in all environments.

Browser	Device	Look and feel	Functionalities	Bugs
Edge	Windows	As expected	As expected	None
Firefox	Windows	As expected	As expected	None
	Android	As expected	As expected	None
	iOS	As expected	As expected	None
Chrome	Windows	As expected	As expected	None
	Android	As expected	As expected	None
	iOS	As expected	As expected	None
Opera	Windows	As expected	As expected	None
	Android	As expected	As expected	None
	iOS	As expected	As expected	None
Safari	Windows	As expected	As expected	None
	iOS	As expected	As expected	None

Permission: When a user logs in, they can view data from their specific search history. These searches are only viewable by the user as long as he/she is currently logged in.

5 Summary of Changes

Moving forward, there is great room for improvement for this application. Through functional testing, we have identified that the test cases outlined have passed successfully. Some tests that were preliminarily considered to be automated tests were changed due to the scope of the requirements. For example, the accuracy of the sentiment scores results for tweets could not be an automated test since it can only be verified by benchmarking it against a human's perceived sentiment. This was changed to a manual test method and a sample of tweets were checked to ensure proper sentiment scores are generally assigned.

As for non-functional testing, our single page web application has adapted a modern design that is proven to have good usability following general heuristics of good design. This can be tested further with usability surveys and questionnaires. This will be done in the future to ensure proper usability of the web application. User feedback has and will continue to have great weight in the future design changes that need to be made to make this web application more user friendly.

Since this is a product that will require further customizing based on specific business requirements, there are more changes expected in the development of other functionality for this web application. These changes will be dealt with when come upon.