

---

# Test Plan

## Web Chat Client with Chatbot

4<sup>th</sup> February 2017

Team 8

Atul Aneja, Jalaleldeen Aref, Tarang Khanna, Wyatt Larkey, and Joel Van Auken

### Design Inspection

<b>Product</b>	<b>Chatbot Design Inspection</b>
<b>Date</b>	<b>2/4/2017</b>
<b>Author</b>	<b>Chatbot Team (Team 8)</b>
<b>Moderator</b>	<b>Wyatt Larkey</b>
<b>Inspectors</b>	<b>Atul Aneja, Tarang Khanna, Joel Van Auken, Jalal Aref</b>
<b>Recorder</b>	<b>Wyatt Larkey</b>

<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How Corrected</b>
1	HTML front end should use sockets to notify the server when a user selects "send" and also send the string.	1	Creat specifically for sending user text to the python server.
2	Front End would call the server even if the message is not a valid one for example if the message box is empty. This would just waste resources.	2	Before server is called, some preliminary javascript will check to make sure the message has a valid length (0 - MAX) Before being sent to the server via checkmessage.js.
3	Server is not currently memory efficient	2	Make a database to

	enough to hold all the user data.		which data gets sent from the server and is then stored.
4	Server does not have efficient resources to run the chat bot that will answer to user's queries	2	Create an API using which server could contact the chatbot, which is running separately.
5	Server is not currently reporting users connected to a chatroom.	1	Using sockets, server would be updated with the number of users in the chatroom every x seconds.
6	Login page won't throw an error in case someone uses invalid username or use a previously taken username.	3	Add front end error checking for length and other invalidities as well as server side would check the database to make sure the username is not already in use.
7	Chatbot backend does not have a broad understanding of the english language so would need natural language processing .	2	API.AI's natural language processing can be used to convey user request to the chat bot. We will use webhooks to make the connections between App.py and API.AI
8	Chatbot backend (App.py) needs access to machine learning algorithm for stock predictions in file-predictStocks.py.	2	Used python importing to import predictStocks.py and create an object from its class which I am using in App.py
9	Chatbot backend (App.py) needs access to twitter feelings analyzer algorithm for stock tweets in file twitter_analyze	2	Used python importing to import twitter_analyze.py and create an object from its class which I am using in App.py

## Code Inspection

<b>Product</b>	<b>Module Code Inspection</b>
<b>Date</b>	<b>2/4/2017</b>
<b>Author</b>	<b>Chatbot Team (Team 8)</b>
<b>Moderator</b>	<b>Wyatt Larkey</b>
<b>Inspectors</b>	<b>Atul Aneja, Tarang Khanna, Joel Van Auken, Jalal Aref</b>
<b>Recorder</b>	<b>Atul Aneja</b>

<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How Corrected</b>
1	Function checkmessage() in checkmessage.js. Max char limit for messages would vary to be a different number than assigned.	3	Variable's values were being changed because it got spelled like the other similarly spelled variable.
2	Function getStockPrediction(req) in app.py. Communication to app.py file over a local server were getting messed up.	3	The api call had an extra argument that was messing with the format of the data being sent.
3	Function loadContent() in fails to completely load the data.	1	The function was overwriting the array indexes which is causing incomplete load of data.
4	Sign in and forget password button on login page to be were overshadowing the other one.	3	Front End code had poor precision which caused wrong placements.
5	To extract the room id we passed the variable u as the index instead of the actual character "u".	1	Pass the character as an index instead.

---

6	checkmessage() in checkmessage.js does not receive users text from the the UI. (onclick="checkmessage()" doesnt work as expected)	1	Modify the code to allow users text to be processed in checkmessage.js and sent to server.
7	Function getStockCurrentPrice(req) in app.py. Needs to return meaningful message if stock parameter given is not found.	3	Display a message to user telling them the stock was not found.
8	Main landing page does not properly take user id or passwords, according to best practice	3	Page now captures important information and processes it for giving to service
9	Sign in and forgotten password page do not properly take user id or email, according to best practice	3	Page now captures important information and processes it for giving to service

## Unit Test Inspection

We automated our unit testing using a Python module called "unittest": A unit testing framework.

<b>Product</b>	<b>Module Unit Test Inspection</b>
<b>Date</b>	<b>2/4/2017</b>
<b>Author</b>	<b>Chatbot Team (Team 8)</b>
<b>Moderator</b>	<b>Wyatt Larkey</b>
<b>Inspectors</b>	<b>Atul Aneja, Tarang Khanna, Joel Van Auken, Jalal Aref</b>
<b>Recorder</b>	<b>Atul Aneja</b>

<b>Defect #</b>	<b>Description</b>	<b>Severity</b>	<b>How Tested</b>
1	Unit testing the getDividendDate for a stock to see if it is not null. Checks if API is working.	1	Created "testDivedendPayDate" method in "TestMLMethods.py" file.
2	Unit testing the machine learning prediction result. Checking if it is a float, since we expect to get a float.	2	Created "testCurrentPriceType" method in "TestMLMethods.py" file.
3	Unit testing the machine learning getCurrentPrice results.	3	Created "testCurrentPriceSign" method in "TestMLMethods.py" file.
4	Unit testing the getPredictedPrice for a given time interval.	2	Created "testPredictionTime" method in "TestMLMethods.py" file.
5	Unit testing signup manually by providing it pairs of usernames and password.	1	Manually passed in a certain number of pairs

---

			of usernames and passwords and made sure the database got populated as expected.
6	Unit testing login by using the manually created pairs those were signed up earlier.	1	Used the same old pairs, the ones in database to login. Logged in successfully when a right pair was entered.
7	Unit tested the chatting feature of the chat room.	1	Manually sent messages to the socket message receiving function which populated the front end with the given message rightfully.
8	Unit tested the bot responsiveness.	2	Connected bot to a facebook messenger account and had an hour long conversation with to test the bot functioning.
9	Unit tested the method that populates the name of users in the chat room.	2	Provided add people to the chat room function with a list of users and checked if it populated appropriately.