
Incremental and Regression Testing

Web Chat Client with Chatbot

9th March 2017

Team 8

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

Components

During the first sprint we created all the modules needed for the project, but did not completely implement all the features we wanted. So for the second sprint, We focused on implementing these features in their entirety. The modules listed below are the same from sprint 1, but we have added some functionality to them.

UI - Application runs in a browser and uses HTML and CSS to display the user interface. Text from on screen is processed and sent to different components using Javascript and Sockets.

UI Input - addMessage() - Adds Responses from other users

notifyMe() - Notifies user when tagged

joinRoom() - moves a user to a different room

UI Output - sendMessage() - Sends message to Server Runtime

setName() - Tells server which user is sending messages

getAllRooms() - shows a user the available rooms they can join

Server Runtime - Using Node.js, the server receives the input over socket io from the frontend and determines where the input should be processed as well as sending information back to the UI.

Server Input - getMsg() - Gets messages from the UI

getRooms() - Gets data about the message rooms.

createRoom() - Initializes new room when a user wants to create one

`createRoomWithPassword()` - Initializes new room when a user wants to create one with password option

Server Output - `io.sockets.on` - Strings and Requests from the UI, messages and requests for signing up, number of users in a room, user names, etc.

`webhook()` - communicates with the API to get ML data and Natural Language parsing from separate service running on a python server on heroku.

Natural Language Parser- Used for chatbot NLP. The server calls API.AI to parse messages from natural language to action and entities. This is forwarded to our custom webhook which determines what the correct response should be. The correct function is called from the machine learning module over API.AI sdk.

NLP Input - UI sends messages to server.

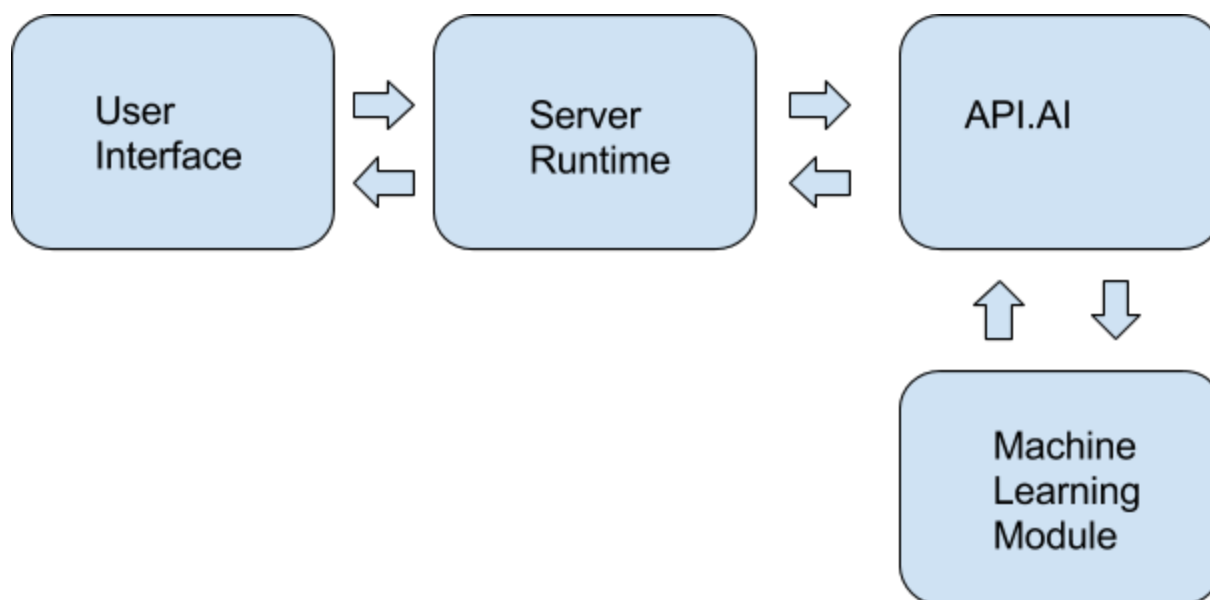
NLP Output - Server invokes code in `server.js` to call upon API.AI for language parsing when we

Machine Learning Module - Processes stock data and outputs stock metrics that the user can view in the UI. This is done by use of hosted python server on heroku.

MLM Input - Request from python script `app.py`

MLM Output - Response back to Server.

- Diagram of Application -



Incremental and Regression Testing Log

- For our Incremental Testing we used a Bottom-up approach, testing each basic component first and working our way up.

Defect #	Description	Severity	How Corrected
1	When a user tries to create a new room with password, the input just reloads the webpage.	1	The button used the same input = "submit" tag that the "change user" button did. This caused both inputs to do the same thing, so we changed the how the form was structured to keep the inputs separate.
2	The 3rd created room (4th in total) will not initialize, and the user will not be able to make any more rooms.	2	We initialized a static array to hold the "rooms". This array had only 3 indexes. We corrected this by making the array dynamically resize when new rooms are added.
3	Tagging the bot makes it active but was not getting the right query.	2	Problem in regex that captures the query for the stockbot from the message. Fixed regex to capture query right after "@stockbot".
4	The module which provides stock predictions was not returning a float as expected for the current price of a given stock	3	fixed API call to Yahoo API to get current price in float, and also added automated testing to make sure this works in the future too.
5	Link was not getting highlighted when it is combined with other text.	3	Updated regex to find link in between other words and highlight only the link
6	Message sent to incorrect room, problem while switching the room. When user enters a new room and sends a message, their message went to the first room they entered.	1	This is a big problem, when user selects to switch room through the UI, the "switchRoom" method in the server was not updating the current room for that user.

Regression Testing

Defect #	Description	Severity	How Corrected
101	When a user tries to enter a new chat room, the webpage is simply reloaded, user ends up in same chatroom.	1	These buttons were also linked to “submit” tag as “change user”. Changed the forms structure to so the input is handled accordingly.
104	Returned float values for stock prices were not the right amount of decimal places (\$102.23333 instead of \$102.23)	2	Added simple check to make sure printed value makes sense by rounding floated value to 2 decimal places.
105	Link was being generated when the text was not supposed to be a link at all.	3	Updated regex to help minimize false positives when deciding what a link is.