AIDERA

Team 2: Arjun Bagla, Abhijit Edlabadkar, Rajalakshmy Iyer,
Eehita Parameswaran, Aakash Ranga, Akanksha Tripathy

March 3rd, 2017

Sprint 2: Design Inspection, Code Inspection and Unit Testing

Contents

1 Design Inspection Defects

Product	Aidera Design Inspection		
Date	February 21st, 2017		
Authors	Arjun Bagla, Aakash Ranga		
Moderators	Rajalakshmy Iyer, Eehita Parameswaran		
Inspectors	Abhijit Edlabadkar, Akanksha Tripathy		
Recorders	Arjun Bagla, Aakash Ranga		
Defect #	**Description**	**How Corrected**	**Severity**
1	The files containing the details for configuration of Yelp should have been in a separate directory/folder and not in the file where the setup of the application is.	The files containing the yelp service details were put in a separate folder called services.	1
2	The file containing system specific commands and functionality should be in a separate directory to avoid unnecessary exposure, create maintainable design structure and to avoid confusion.	The system functionality containing files were put in a new directory called system.	2
3	The file that holds the node modules connection settings (all dependencies) should be in a separate directory/folder and not in the file where the setup of the application is.	The npm tools and other dependencies were put in a folder called node modules.	2
4	The files that holds the intents (recommended-cuisine.js, search nearby restaurants.js, cuisine-restaurants.js, reviews-restaurants, etc.) should be in a separate directory/folder and not in the file where the setup of the application is.	The intents file recommended-cuisine.js, search nearby resturants.js, cuisine-restaurants.js, reviews-restaurants and other files were put in a new directory/folder called yelpIntents.	2
5	The files that are used to connect to Yelp should be kept as a separate module and not in the main server.js file. This module should not be exposed as it uses sensitive information like page access token etc.	The Yelp module with yelp connectivity files were put in a separate module of their own.	2

Defect #	Description	How Corrected	Severity
6	The main server.js file that was used to connect our application to ngrok uses body-parser, a node module for node.js based servers. The environment should never specify a port because that may hinder connection to ngrok or facebook.	The environment never contains a variable with port value in it. In that manner, body-parser doesn't get confused.	1
7	Large number of automated tests were pushed that made the code base unnecessarily heavy. A solution was needed so that the automated tests aren't pushed every time code is committed.	A file named .gitignore had to be modified to avoid automated tests from being pushed to the code base.	3
8	Mocha and Chai, the testing frameworks we are using for local testing should never be shut down since it generates a dummy url for query forwarding. The url is used to get and send information from and to our application.	Mocha and Chai are never re-initialized on each of our laptops to keep testing url the same.	2
9	The access token for the two services: Yelp, Airbnb and, Facebook page should be put in a separate file to maintain modularity, usability and maintainability.	A separate file config.js was created in the main directory that contains all the necessary tokens.	1
10	The simplified HTTP request promises comes with Promise support and it should be put in a separate file for safety and encapsulation. Thus, ensuring that sensitive information isn't exposed unnecessarily and all the information for using different API's is in one place.	A separate config.js file was made for the various access tokens and urls.js was made for the call urls.	1
11	In the processor of requests i.e. request, request-promise, user input expressions that are Yelp specific should be put under their respective intents.	Various intents were made to recognise what the user is trying to say along with the suitable expression.	3

2 Code Inspection Defects

Product	Aidera Code Inspection		
Date	February 24th, 2017		
Authors	Arjun Bagla, Aakash Ranga		
Moderators	Abhijit Edlabadkar, Rajalakshmy Iyer, Eehita Parameswaran, Aakash Ranga		
Inspectors	Abhijit Edlabadkar, Akanksha Tripathy		
Recorders	Arjun Bagla, Aakash Ranga		
Defect #	**Description**	**How Corrected**	**Severity**
1	When the user is trying request a service that is not yet supported by the application (the chatbot supports functionality provided by Airbnb and Yelp), the error received for the intent was null.	The error of null intent was gracefully handled by giving the user an appropriate message to try asking something else and continue the conversation.	2
2	Yelp, Airbnb or Facebook Graph API call would lead to asynchronous behaviour and subsequent code would be executed before reply from the any of the API was received.	A simplified HTTP request client 'request' with Promise (an object used for asynchronous computations) support was used to deal with asynchronous behaviour.	2
3	Due to the asynchronous nature of Node.js, problems occur when a variable is used before a query is finished that will assign a value to that variable.	Promises were used to represent the eventual result of an asynchronous operation. The object was used as a placeholder in which the successful result value or reason for failure occurs.	3
4	A database schema was created to store attributes like location, price range, request type etc and in order to support proper functionality of the integrated API's. MongoDB database crashed if any of the values in the schema are null at any point of time.	If it is known that the value will be null at any point of time, then initialize it as default=null while declaring it.	3
5	After integrating DB with the Yelp API, the application was still using user's last location (latitude and longitude) whenever user asked the bot to recommend places "near me"	The problem was fixed by making the last used latitude and longitude position null after the user's request is served once.	1

Defect #	Description	How Corrected	Severity
6	On integrating Recast.AI with the Yelp API, the chatbot was not identifying the location sent with the message itself. For instance, in the request "suggest food places near Kirkland", the location Kirkland wasn't identified.	The issue was that the NLP API wasn't confidently catching the appropriate intent and the location. It was fixed by first parsing the string received from the user in a manner that the Recast.AI could understand. The request was then sent to Recast.AI.	2
7	The main issue faced during the integration of the Airbnb API with the chatbot is not being able to get the required information like listing of a place just with GET requests.	Solved the issue by making a POST requests for each session and each unique token as well.	2
8	After the integration of the Airbnb API, implementation of the planned features wasn't working successfully. This was because correct endpoints were not so easily found.	The solution was to find correct endpoints by the method of trial and error and manually printing and scanning the JSON files to get the appropriate and relevant data.	1
9	For the proper functionality of the API's, certain features required nested requests that gave asynchronous errors with promises. In other words, the promise would resolve before completing the second request.	Fixed the issue by adding quite a few lines of apparent redundant code so that promise wouldn't resolve before completing all the nested requests.	1
10	The cards that support API functionality, do not provide the functionality 2 buttons of type 'web urls'	The difficulty was addressed by just providing one button that will redirect to the webpage containing more information about the location.	2
11	Had trouble with the authorization header that is essentially is a parameter used for sending information through a request.	The solution was to add the authorization header to every function that is called every time the Yelp service is requested.	3
12	Faced issues with connecting the business array to get the top 10 objects from all the businesses that the Yelp API sends us.	The data about all the businesses received from the API was first converted to JSON then it was sliced to an array holding the first 10 objects returned.	1

Defect #	Description	How Corrected	Severity
13	Incorrect fallback url with the Facebook Generic templates' was causing problem.	Correct fallback url that used the ngrok was given and the expected functionality was achieved.	1
14	The quick reply buttons added were not providing the functionality as desired.	The issue was with the payload property not being able to call the right function to calculate the price. Fixed by calling the appropriate method and received the desired result.	1

3 Unit Testing Defects

Team Aidera automated most of the Incremental and Regression tests for our bot built in Node.js. We used a tool called 'Mocha' which is a feature-rich JavaScript test framework running on Node.js and in the browser. This tool helped us make asynchronous tests in a simple manner. These Mocha tests run serially thus, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. So, with Mocha we had an environment to make our tests but to actually test our HTTP calls, we required an add-on library.

Hence, to add the necessary logic, we utilized 'Chai' which is an assertion library. One of the main reasons we chose Chai over other assertion libraries was because it allowed us to choose the type of assertion style we'd like to use. This library comes with three different assertion types. It has the should style, the expect style and the assert style. Mocha essentially provides traction for unit testing as there are a lot of libraries that are built on the expect package such as supertest. Supertest allows developers to test API endpoints very easily by querying the api directly and asserting the responses. The expect package can then be used to test the responses in more detail. We decided to use the expect style and by using chai-http, we were able to make the actual HTTP requests. And then, we tested the responses with the expected results.

Product	Aidera Unit Testing		
Date	March 1st, 2017		
Authors	Eehita Parameswaran, Rajalakshmy Iyer		
Moderators	Arjun Bagla, Abhijit Edlabadkar, Aakash Ranga, Akanksha Tripathy		
Defect #	**Description**	**How Corrected**	**Severity**
1	Tedious process to host Facebook services on Heroku cloud-platform for local testing	Decided to not use Heroku for middleware testing because multiplatform tunnelling is possible using other servers	1
2	Unable to connect to a public endpoint (i.e. internet) using a locally running network service	Used ngrok to host a server on which we run Aidera	1
3	Aidera provided no response on on being asked for a recommended cuisine. Input: 'recommended cuisine' Output: No Output	Added Natural Language Processing and provided varied cuisines every time. Correction: 'try ...!'	1

Defect #	Description	How Corrected	Severity
4	Aidera was unable to respond with generic template cards when user sent location using messenger application Input: (send location via messenger application) Output: No Output	Implemented algorithm that can parse latitude-longitude from sent location Correction: pinned location which then shows appropriate restaurants	1
5	Aidera replied with an output that was far off from the user's actual location Input: 'restaurants near me' Output: (bot would send restaurants near Boston)	Modified algorithm such that the user can get resturants near them. Correction: Bot replies with a list of restaurants near the user (i.e. West Lafayette)	3
6	Restaurants output was in a list and was non-interactive.	Re-wrote the algorithm such that the list of restaurants now shows up in a card's format.	2
7	Bot was unable to provide appropriate responses to restaurant reviews Input: 'reviews of Basil Thai' Output: error because the id of the restaurant could not be passed to the reviews method	Modified algorithm such that a restaurant search autocomplete function called the reviews method with the restaurant id Correction: Bot was able to print responses to the console but unable to actually provide reviews	2
8	Bot was unable to provide appropriate responses to restaurant reviews Input: reviews of Basil Thai Output: 'Didn't really get that. Could you try something else?'	Added reviews intents to Recast.ai collection and updated algorithm to respond appropriately Correction: Bot responds with a list of reviews of all thai restaurants	1
9	Bot would provide reviews of all thai restaurants when asked for a specific restaurant Input: reviews of Basil Thai Output: Bot responds with a list of reviews of all thai restaurants	Updated algorithm to respond appropriately such that a query like this would provide responses only for Basil Thai Correction: Bot now responds with several review as a list just for Basil Thai	1

Defect #	Description	How Corrected	Severity
10	Restaurants reviews output was in a list and was non-interactive.	Re-wrote the algorithm such that the list of restaurants reviews now shows up in a card's format.	3
11	Bot was unable to provide appropriate response to price texts Input: 'cheap restaurants near Kirkland' Output: 'Didn't really get that. Could you try something else?'	Added price intents to Recast.ai collection and updated algorithm to respond appropriately Correction: Bot would provide responses to price queries but answers were in a list	2
12	Bot would provide responses to price queries but only for '$' restaurants Input: 'cheap restaurants near Kirkland' Output: bot would provide responses only to '$' button options	Updated algorithm such that quick reply buttons would work for '$', '$$', '$$$' buttons Correction: Bot would provide responses to all the price queries(according to '$', '$$' or '$$$') in a list	2
13	Bot would provide responses to price queries but everything was in a list Input: 'cheap restaurants near Kirkland' Output: bot would provide appropriate responses in list format	Updated algorithm such that responses were provided in a more interactive-user friendly interface Correction: Bot would provide appropriate responses to all the price queries in a card's format now	3
14	Bot was unable to respond to queries of categorization restaurants Input: 'Chinese restaurants near me' Output: 'Didn't really get that. Could you try something else?'	Added categorization intents to Recast.ai collection and updated algorithm to respond appropriately Correction: Bot would then list all the chinese restaurants nearby	1
15	Bot wasn't able categorize restaurants. Input: 'chinese restaurants' Output: Bot would output all restaurants nearby	Updated algorithm to recognize category inputted and and output the correct list of restaurants. Correction: Bot lists all the chinese restaurants nearby.	1

Defect #	Description	How Corrected	Severity
16	The categorized restaurants output was in a list and was non-interactive	Re-wrote the algorithm such that the list of categorized restaurants shows up in a card's format.	2
17	Bot would provide reviews of all accommodations when asked for a specific site. Input: 'reviews of Charlie Chaplin Accommodation' Output: Bot responds with a list of reviews of all accommodations	Updated algorithm to respond appropriately such that a query like this would provide responses only for Charlie Chaplin Correction: Bot now responds with several review as a list just for Charlie Chaplin	2
18	Restaurants reviews output was in a list and was non-interactive.	Re-wrote the algorithm such that the list of accommodation reviews now shows up in a card's format.	2
19	Bot was unable to respond to queries of listings of places near user Input: 'accommodations near me' Output: 'Didn't really get that. Could you try something else?'	Added listing intents to Recast.ai collection and updated algorithm to respond appropriately Correction: Bot would then list all the listings near user	2
20	Bot would list the accomodations but was unable to provide an interactive list Input: 'accommodations near me' Output: Bot would show listings in list format versus cards format	Re-wrote the algorithm such that the list of nearby listings shows up in a card's format. Correction: Bot would show listings in card format now such that user could get more information	2
21	Bot would provide public ratings of all accommodations when asked for a specific listing. Input: 'reviews of Charlie Chaplin Accommodation' Output: Bot responds with a list of reviews of all accommodations	Updated algorithm to respond appropriately such that a query like this would provide responses only for Charlie Chaplin listing Correction: Bot now responds with several public ratings as a list just for Charlie Chaplin	2
22	Restaurants ratings output was in a list and was non-interactive.	Re-wrote the algorithm such that the list of accommodation ratings now shows up in a card's format.	3