Software Development 2

BSCH-SD2/Dub/FT

ChatBot Project

Krishna Ramdam

Milestone: Final Submission

First review:

Goal:

Our main goal in the first review was just to run chatbot and understand the meaning and the uses of various tools provided via emulator, library and dependencies.

Designing tools:

We used JFrame in Java which is graphical user interface toolkit to create a GUI of our chatbot. For the programming language we used java programming language and for development purpose we used Eclipse and Maven project. The main thing our team used in this review was AIML (Artificial Intelligence Markup Language) library. Which comes with pre-build chatbot code.

```
<aiml>
    <category>
        <pattern>Hello</pattern>
        <template>Hi! How are you.</template>
        </category>
</aiml>
```

Progress:

- Working chatbot
- Designed GUI

Second review:

Goal:

Our main goal in second review to add weather api in our running chatbot. Our bot has successfully connected with weatherapi. We are using unirest dependency.

```
\   / |_  /\  |_  |_| |_  |_) /\  |_) |
 \/\/  |_ /--\ |_ | | |_ |  /--\ |   |_|

             powered by ActFramework r1.9.1b-0646

 version: v0.0.1-SNAPSHOT-${buildNumber}
scan pkg: com.google.weatherApi
base dir: C:\Users\crayz\eclipse-workspace\weatherApi
     pid: 14960
 profile: dev
    mode: DEV
      OS: Windows
     jdk: OpenJDK 64-Bit Server VM 15

     zen: Special cases aren't special enough to break the rules.
          Although practicality beats purity.

2021-04-15 20:58:56,261 [34mINFO [0;39m [36ma.Act@[main][0;39m - loading application(s) ...
2021-04-15 20:58:56,277 [34mINFO [0;39m [36ma.a.App@[main][0;39m - App starting ....
2021-04-15 20:58:56,293 [34mINFO [0;39m [36mo.xnio@[Thread-0][0;39m - XNIO version 3.8.0.Final
2021-04-15 20:58:56,305 [34mINFO [0;39m [36mo.x.nio@[Thread-0][0;39m - XNIO NIO Implementation Version 3.8.0.Final
2021-04-15 20:58:56,481 [34mINFO [0;39m [36mo.j.threads@[Thread-0][0;39m - JBoss Threads version 3.1.0.Final
```

To get our weather api running we user api key from rapidweatherapi.com and added some libraries to run the api.

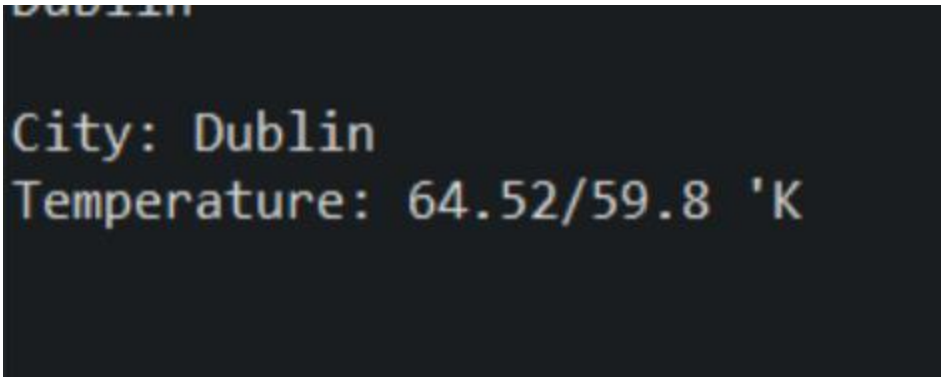
```
        </systemPath>
    </dependency>
    <dependency>
        <groupId>com.mashape.unirest</groupId>
        <artifactId>unirest-java</artifactId>
        <version>1.4.9</version>
    </dependency>
</dependencies>
```

Progress:

- Implemented an api.

```
*chatBot.java ×  chatBot/pom.xml  chatBotTest.java  AppTest.java
35              if (textLine.equals("q")) {
36                  System.exit(0);
37              } else if (textLine.equals("wq")) {
38                  bot.writeQuit();
39                  System.exit(0);
40              } else {
41                  String request = textLine;
42                  if (MagicBooleans.trace_mode)
43                      System.out.println(
44                              "STATE=" + request + ":THAT=" + ((History) chatSession.thatHistory.get(0)).get(0)
45                                      + ":TOPIC=" + chatSession.predicates.get("topic"));
46  //                  String response = chatSession.multisentenceRespond(request);
47  //                  while (response.contains("&lt;"))
48  //                      response = response.replace("&lt;", "<");
49  //                  while (response.contains("&gt;"))
50  //                      response = response.replace("&gt;", ">");
51
52                  HttpResponse<String> response1 = Unirest.get("https://community-open-weather-map.p.rapidapi.com/forecast?q=san%20francisco%2Cus")
53                          .header("x-rapidapi-key", "79ca02ff30msh652d6444a3aefcap1da589jsnd96564ed63e1")
54                          .header("x-rapidapi-host", "community-open-weather-map.p.rapidapi.com")
55                          .asString();
56                  System.out.println("Jarvis : " + response1);
57              }
58          }
59      } catch (Exception e) {
60          e.printStackTrace();
61      }
62  }
63
64  private static String getResourcesPath() {
65      File currDir = new File(".");
```

- Got the temperature.



Failure:

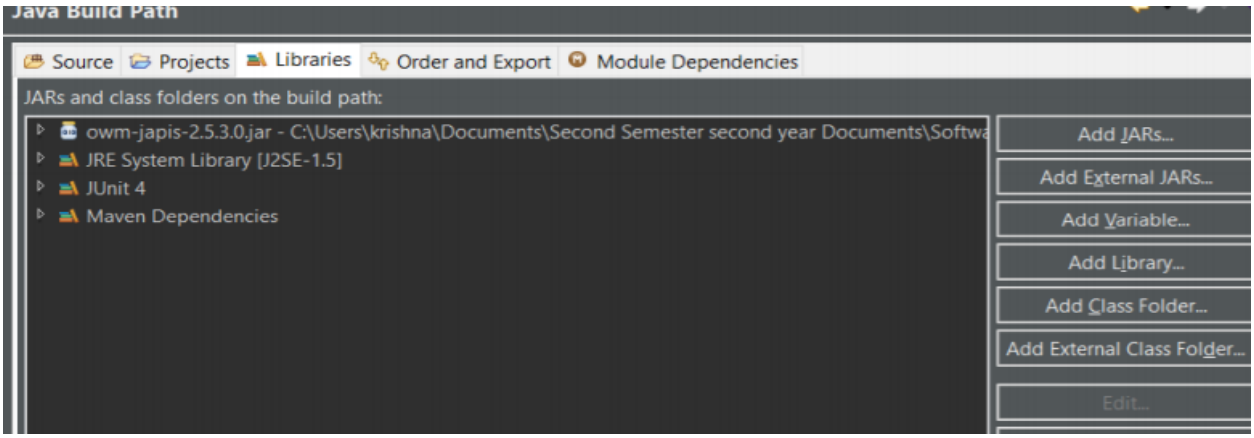
Even though the chatbot worked fine. The JUnit testing in eclipse was really hard to implement in prebuild shell of code.

Three review:

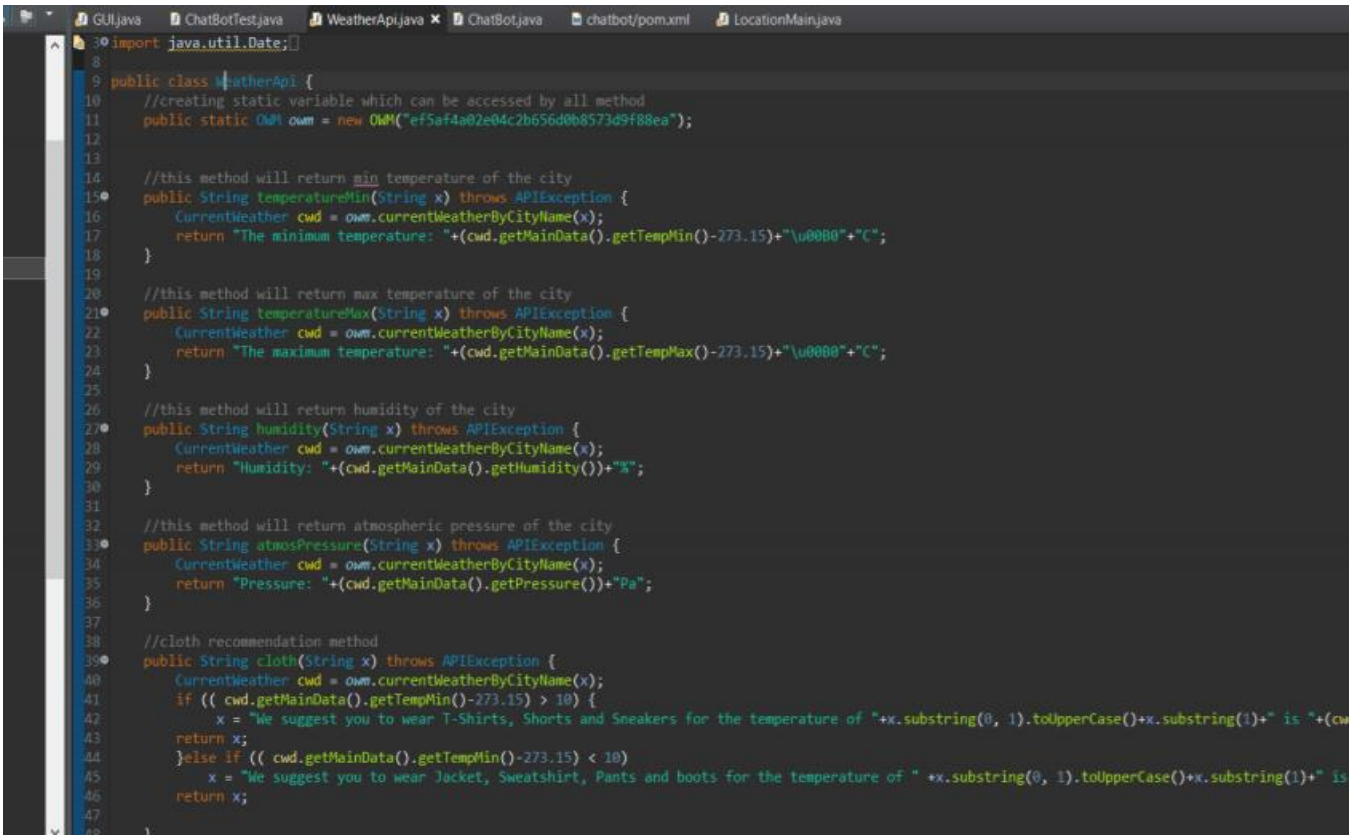
The main objective of this milestone was to fully run weather API, use weather API to our chatbot and run JUnit test to ensure that our chatbot is working properly. Here, in order to proper use weather api and get full authority of what will be the input and response and do JUnit test properly we scrap the idea of making chatbot using AIML and started making chatbot with JFrame and weather key and library provided by OpenWeatherMap.org.

Procedure:

- We added the weather api library.



- Created WeatherApi.java and added method which can extract live weather information

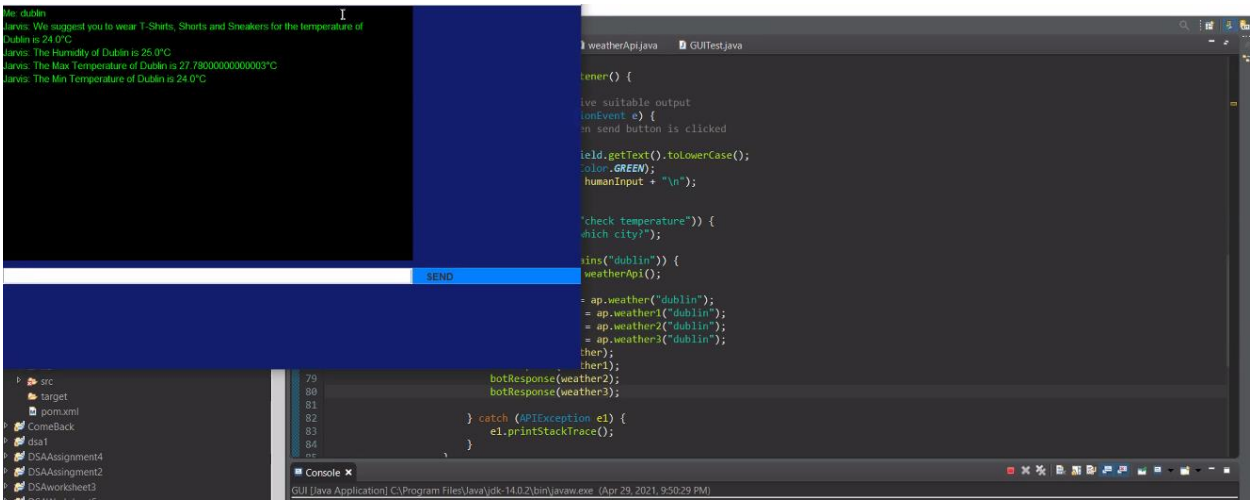


```java
import java.util.Date;

public class WeatherApi {
    //creating static variable which can be accessed by all method
    public static OWM owm = new OWM("ef5af4a02e04c2b656d0b8573d9f88ea");


    //this method will return min temperature of the city
    public String temperatureMin(String x) throws APIException {
        CurrentWeather cwd = owm.currentWeatherByCityName(x);
        return "The minimum temperature: "+(cwd.getMainData().getTempMin()-273.15)+"\u0000"+"C";
    }

    //this method will return max temperature of the city
    public String temperatureMax(String x) throws APIException {
        CurrentWeather cwd = owm.currentWeatherByCityName(x);
        return "The maximum temperature: "+(cwd.getMainData().getTempMax()-273.15)+"\u0000"+"C";
    }

    //this method will return humidity of the city
    public String humidity(String x) throws APIException {
        CurrentWeather cwd = owm.currentWeatherByCityName(x);
        return "Humidity: "+(cwd.getMainData().getHumidity())+"%";
    }

    //this method will return atmospheric pressure of the city
    public String atmosPressure(String x) throws APIException {
        CurrentWeather cwd = owm.currentWeatherByCityName(x);
        return "Pressure: "+(cwd.getMainData().getPressure())+"Pa";
    }

    //cloth recommendation method
    public String cloth(String x) throws APIException {
        CurrentWeather cwd = owm.currentWeatherByCityName(x);
        if (( cwd.getMainData().getTempMin()-273.15) > 10) {
            x = "We suggest you to wear T-Shirts, Shorts and Sneakers for the temperature of "+x.substring(0, 1).toUpperCase()+x.substring(1)+" is "+(cw
        return x;
        }else if (( cwd.getMainData().getTempMin()-273.15) < 10)
            x = "We suggest you to wear Jacket, Sweatshirt, Pants and boots for the temperature of " +x.substring(0, 1).toUpperCase()+x.substring(1)+" is
        return x;

    }
```

- We added remaining dependencies in pom.xml which is required to run our java library.

```xml
    <dependency>
      <groupId>org.apache.httpcomponents</groupId>
      <artifactId>httpclient</artifactId>
      <version>4.5.13</version>
    </dependency>
    <dependency>
        <groupId>org.jetbrains.kotlin</groupId>
        <artifactId>kotlin-runtime</artifactId>
        <version>1.2.0</version>
    </dependency>
    <dependency>
        <groupId>org.jetbrains.kotlin</groupId>
        <artifactId>kotlin-stdlib</artifactId>
        <version>1.2.0</version>
    </dependency>
    <dependency>
        <groupId>com.squareup.okhttp3</groupId>
        <artifactId>okhttp</artifactId>
        <version>3.8.1</version>
    </dependency>
    <dependency>
        <groupId>com.google.code.gson</groupId>
        <artifactId>gson</artifactId>
        <version>2.8.1</version>
    </dependency>
    <dependency>
        <groupId>com.squareup.retrofit2</groupId>
        <artifactId>retrofit</artifactId>
        <version>2.3.0</version>
    </dependency>
    <dependency>
        <groupId>com.squareup.retrofit2</groupId>
        <artifactId>converter-gson</artifactId>
        <version>2.3.0</version>
    </dependency>
  </dependencies>
</project>
```

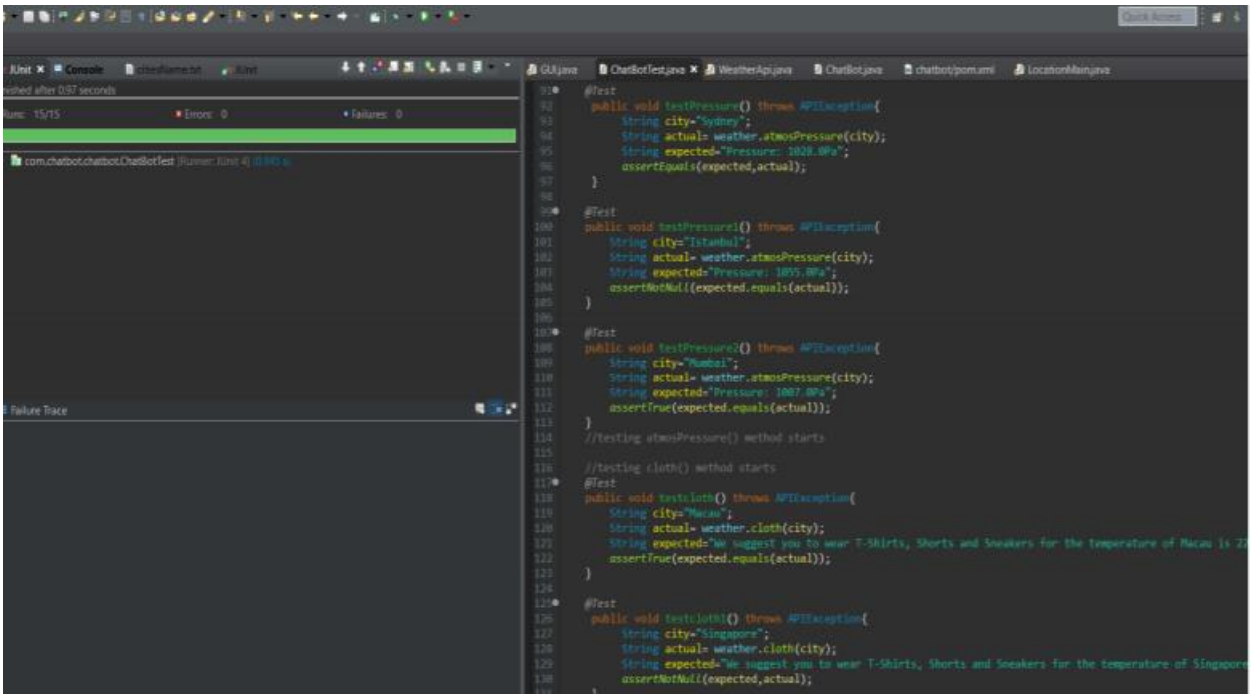
- Created attractive shell for our chatbot



- Created proper user and chatbot responses

```
boolean isValid = false;
//this will send message when send button is clicked
if(e.getSource()==btn) {
    String humanInput=chatField.getText().toLowerCase();
    chatArea.setForeground(Color.GREEN);
    chatArea.append("Me: "+ humanInput + "\n");
    chatField.setText("");


    if(humanInput.contains("hi") || humanInput.contains("hello")) {
        botResponse("Hi there, how are you today.");
        isValid = true;
    }
    else if(humanInput.contains("i am fine")) {
        botResponse("Glad to hear. How can I help you.");
        isValid = true;
    }
```

- JUnit testing



Progress:

- Our bot includes:
- Humidity,
- Max temperature,
- Min temperature
- Clothing suggestions
- Customize GUI
- Accurate responses with respect to user Input.
- Successful JUnit testing

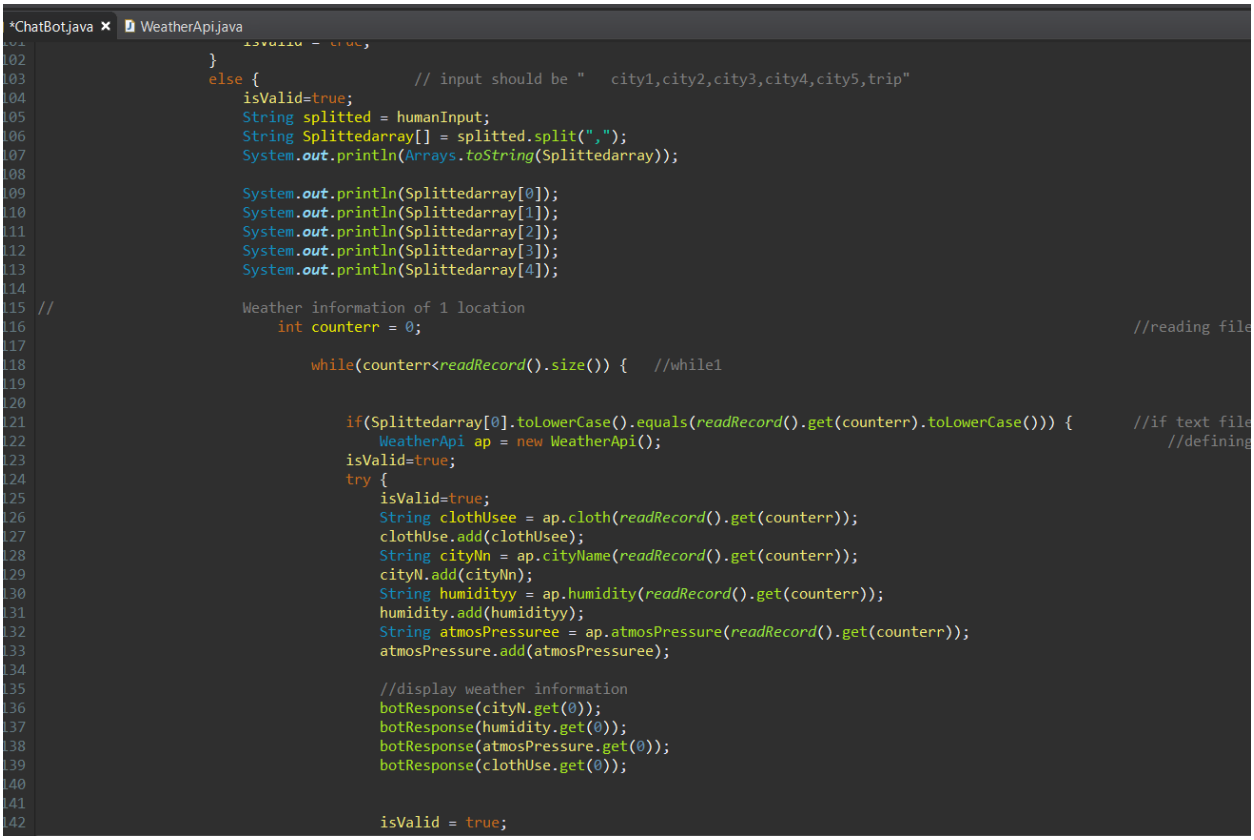
Final Submission:

Goal:

The main objective for the final submission is to make sure everything in the chatbot works perfectly. The chatbot should follow the main objective of our software development 2 project. Enhance the responses from user input and Junit test all the method.

All the steps that we followed for the final submission

- While our pervious iteration of chatbot can only give the weather information and clothing recommendation of one city at a time. Our new chatbot can give information of multiple cities at once. According to project description we are allowing 5 cities max to get weather information.



```
*ChatBot.java ×  WeatherApi.java
102                 }
103                 else {                  // input should be "   city1,city2,city3,city4,city5,trip"
104                     isValid=true;
105                     String splitted = humanInput;
106                     String Splittedarray[] = splitted.split(",");
107                     System.out.println(Arrays.toString(Splittedarray));
108
109                     System.out.println(Splittedarray[0]);
110                     System.out.println(Splittedarray[1]);
111                     System.out.println(Splittedarray[2]);
112                     System.out.println(Splittedarray[3]);
113                     System.out.println(Splittedarray[4]);
114
115 //                  Weather information of 1 location
116                         int counterr = 0;                                                                   //reading file
117
118                         while(counterr<readRecord().size()) {   //while1
119
120
121                             if(Splittedarray[0].toLowerCase().equals(readRecord().get(counterr).toLowerCase())) {       //if text file
122                                 WeatherApi ap = new WeatherApi();                                                   //defining
123                             isValid=true;
124                             try {
125                                 isValid=true;
126                                 String clothUsee = ap.cloth(readRecord().get(counterr));
127                                 clothUse.add(clothUsee);
128                                 String cityNn = ap.cityName(readRecord().get(counterr));
129                                 cityN.add(cityNn);
130                                 String humidityy = ap.humidity(readRecord().get(counterr));
131                                 humidity.add(humidityy);
132                                 String atmosPressuree = ap.atmosPressure(readRecord().get(counterr));
133                                 atmosPressure.add(atmosPressuree);
134
135                                 //display weather information
136                                 botResponse(cityN.get(0));
137                                 botResponse(humidity.get(0));
138                                 botResponse(atmosPressure.get(0));
139                                 botResponse(clothUse.get(0));
140
141
142                                 isValid = true;
```

Figure: Code which make possible to get information of 5 cities at once.

- We enhance the responses for the user input.

```
            }
            if(isValid==false) {
                int rand = (int)(Math.random()*3+1);
                if(rand==1) {
                    botResponse("Please try again invalid data input.");
                } else if(rand ==2) {
                    botResponse("Can you be more specific");
                } else if(rand == 1) {
                    botResponse("Cannot find you data in my database");
                }
            }
    }


    }
    //method to print out bot response
    public void botResponse(String s) {
        chatArea.append("Jarvis: "+s+"\n");
    }
```

Figure: More than single output from same user interaction.

- JUnit testing

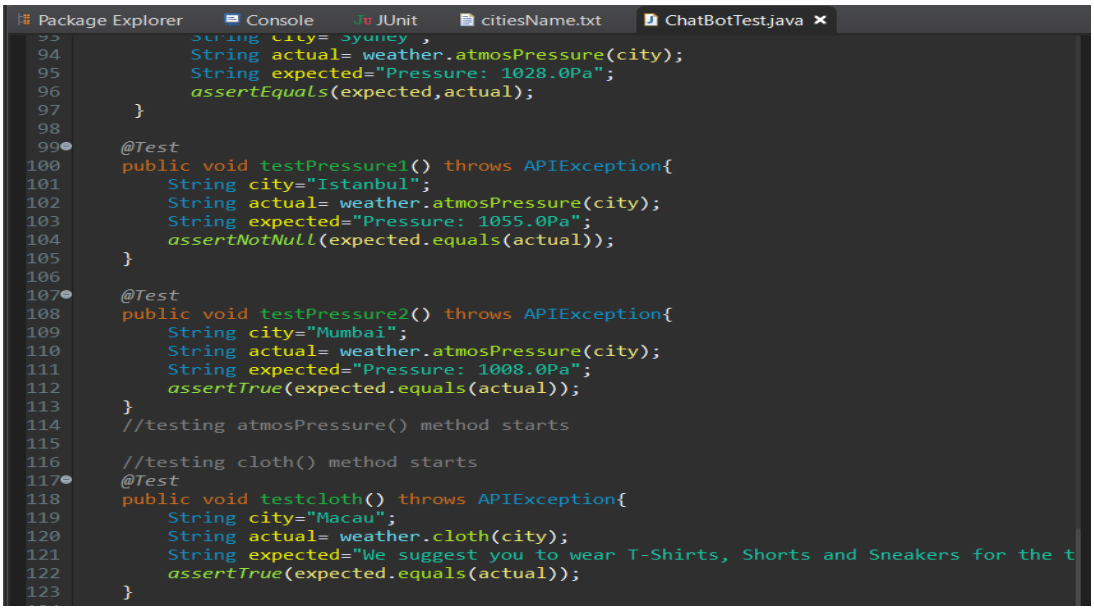


Figure: JUnit testing

- GUI update and output

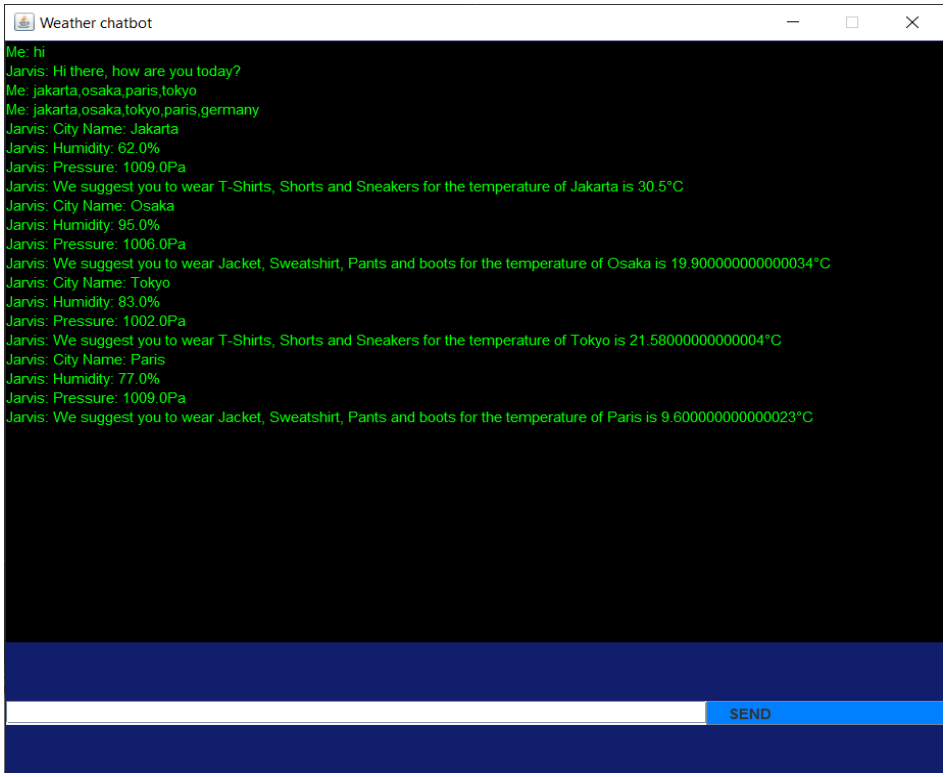


Figure: New chatbot.

For the trip User should enter 5 locations with the format of (city1,city2,city3,city4,city5)

Our console output:

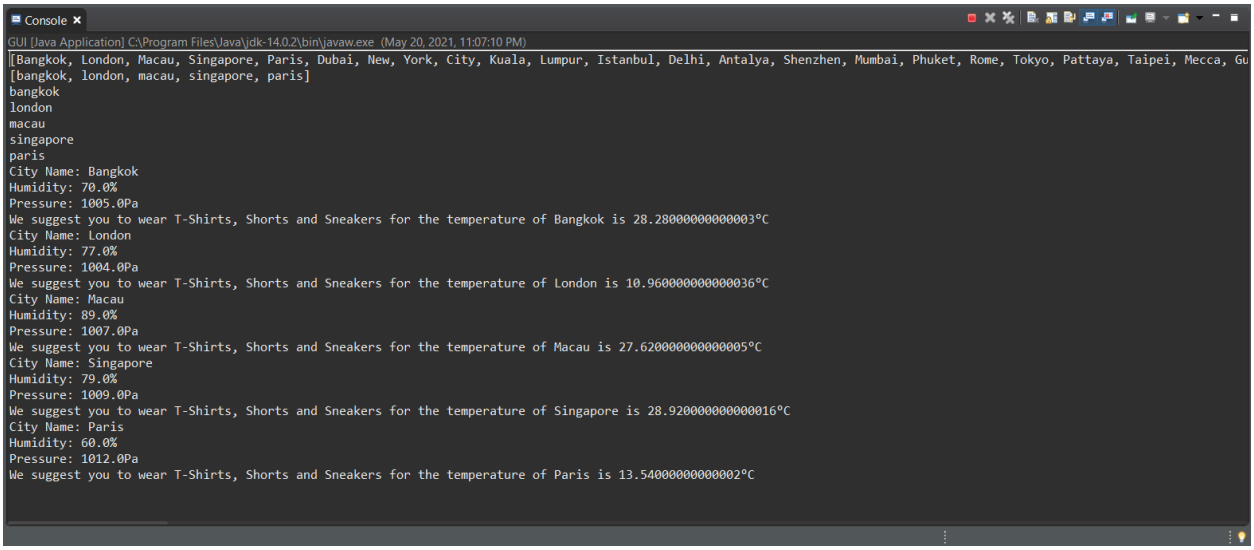


Figure: Information of 5 different cities printed in console.