

CMPE 187 Sec-02

Deliverable #3: Test Automation Report

Application: Replika Chatbot

Team: Group 8

Project Members:

Victor Martinez

Anna Li

Xiran Jia

Paul Junver Soriano

Table of Contents

Section 1 Test Automation Introduction

- 1.1 Test Automation Objective
- 1.2 Test Automation Focus
- 1.3 Selected Test Tools
- 1.4 Test Automation Strategy

Section 2 Test Automation Solutions

- 2.1 Test Automation Scenarios
- 2.2 Test Automation Scripts

Section 3 Test Automation Summary

- 3.1 Test Results
- 3.2 Test Complexity

Appendix

- 4.1 Demo URL

Section 1. Test Automated Introduction

1.1 Test Automation Objective

Test Automation is using automated software tools to test, analyze, and even collect data, unlike manual testing that requires a human tester. The team's objective when it comes to utilizing test automation is to be able to greatly reduce the time it'll take to test the app compared to manual testing. Also, test automation can find all bugs or defects in the software while manually can cause a few of them to be missed. When it came to manually testing Replika.AI it took our team some time to finish the testing, much longer than it would take with test automation. The test automation will be done using Android devices only.

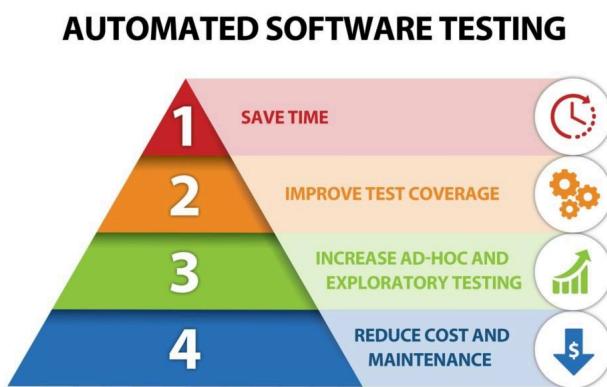


Figure 1.1 Benefits of Automated Testing

1.2 Test Automation Focus

Our Test automation focus is on the following parameters:

Test Parameter	Group Member (Done By)
Memory	Paul Soriano
Domain Knowledge	Anna Li
Language	Xiran Jia
Q&A	Victor Martinez

Figure 1.2 Test Parameters

The team is going to be focusing on the following script perspectives:

- Automatic Test Case Inputs: Can run and repeat to test the same test cases and different test cases.

- b) Automatic Test Case Execution: Automatically opens the Replika application, then reads the input data from the file, and sends that data to the chatbot. Output Data from the chatbot's response to the inputs are recorded by screenshots.

1.2.1 Context Modeling For AI Features

Domain Knowledge

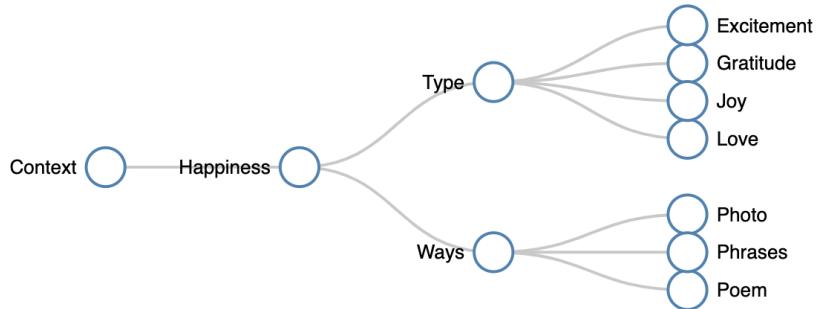


Figure 1.2.1. Context Classification Tree For Domain Knowledge Function (Happiness)

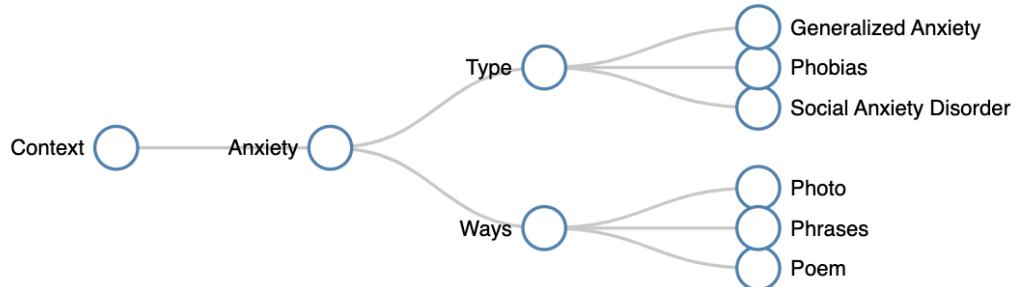


Figure 1.2.2. The Context Classification Tree For Domain Knowledge Function (Anxiety)

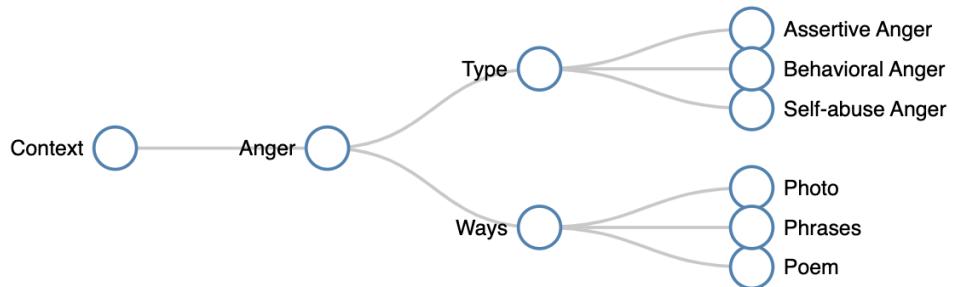


Figure 1.2.3. The Context Classification Tree For Domain Knowledge Function (Anger)

Language

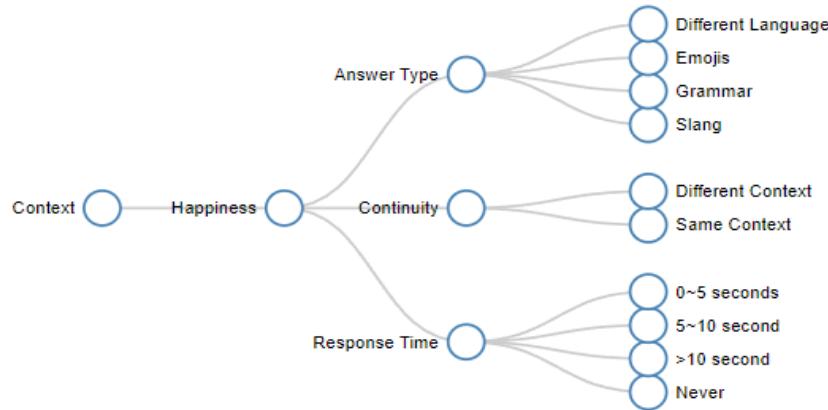


Figure 1.2.4. The Context Classification Tree for Language Function (Happiness)

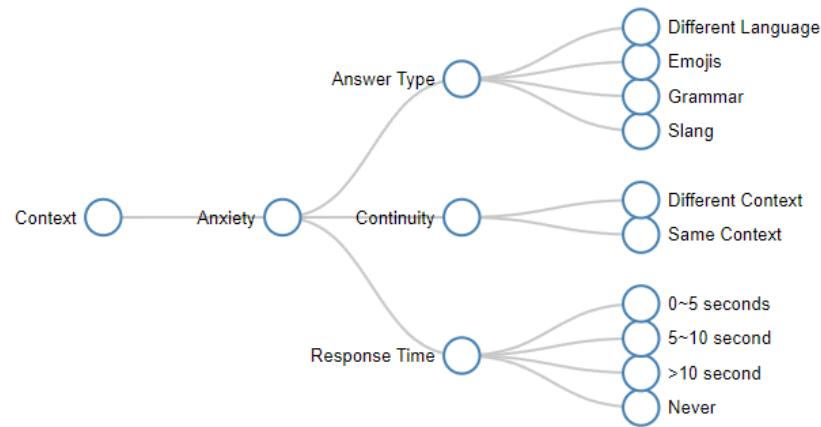
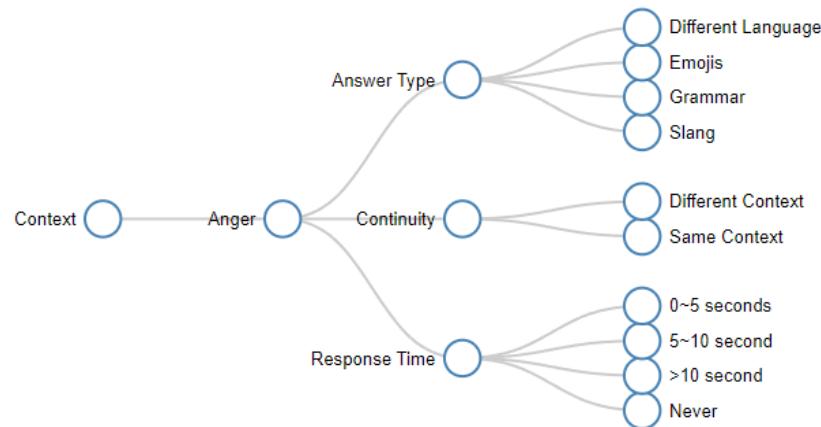


Figure 1.2.5. The Context Classification Tree for Language Function (Anxiety)



1.2.6. The Context Classification Tree for Language Function (Anger)

Question and Answer

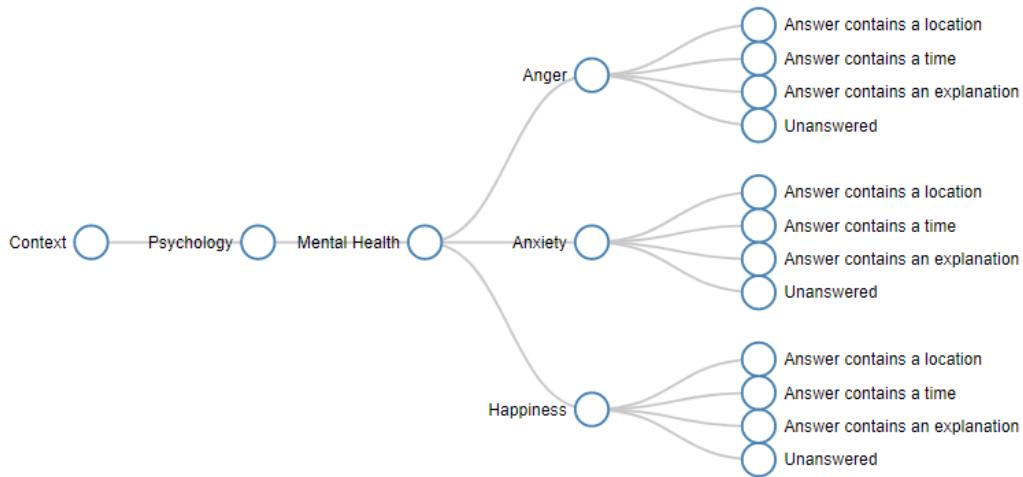
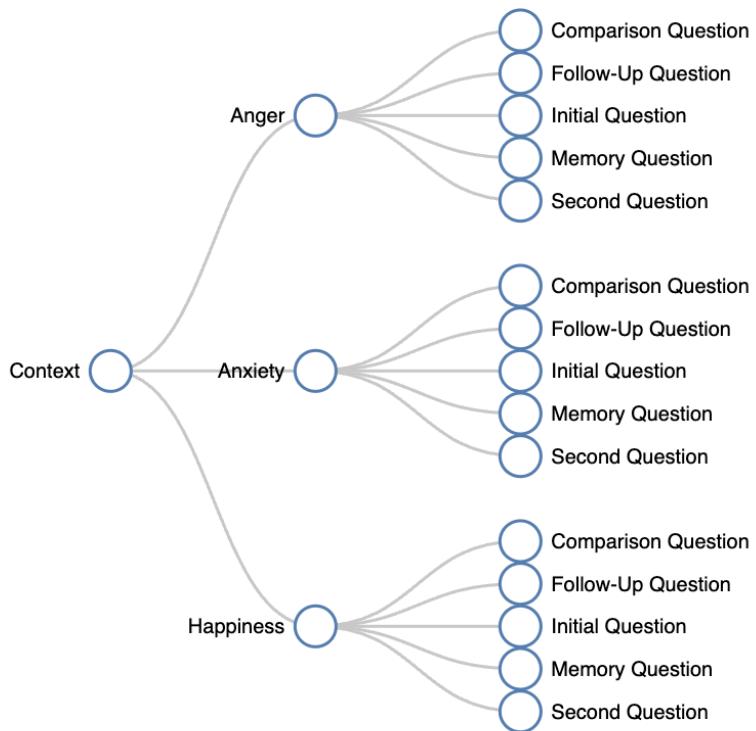


Figure 1.2.7. The Context Classification Tree for Question and Answer

Memory

Figure 1.2.8. The Context Classification Tree for Memory



1.2.2 AI Function Input Classification

Domain Knowledge

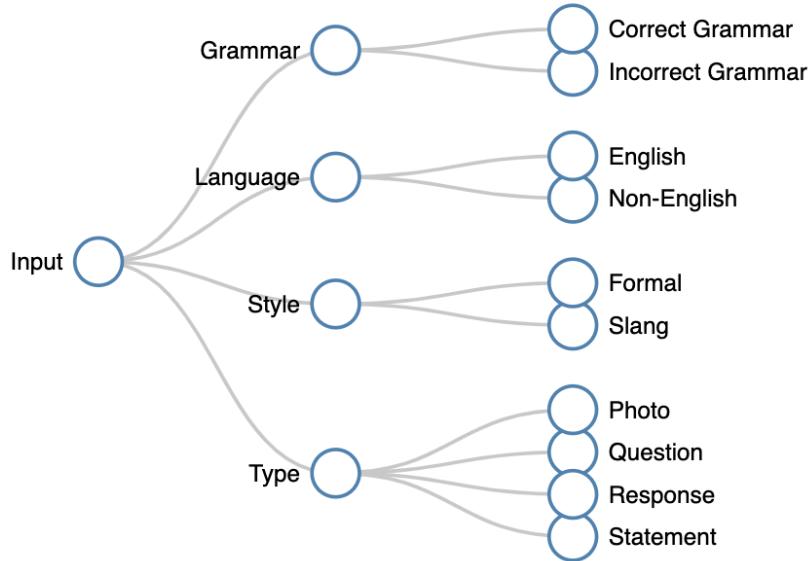


Figure 1.2.9. Domain Knowledge Classification Input Tree

Language

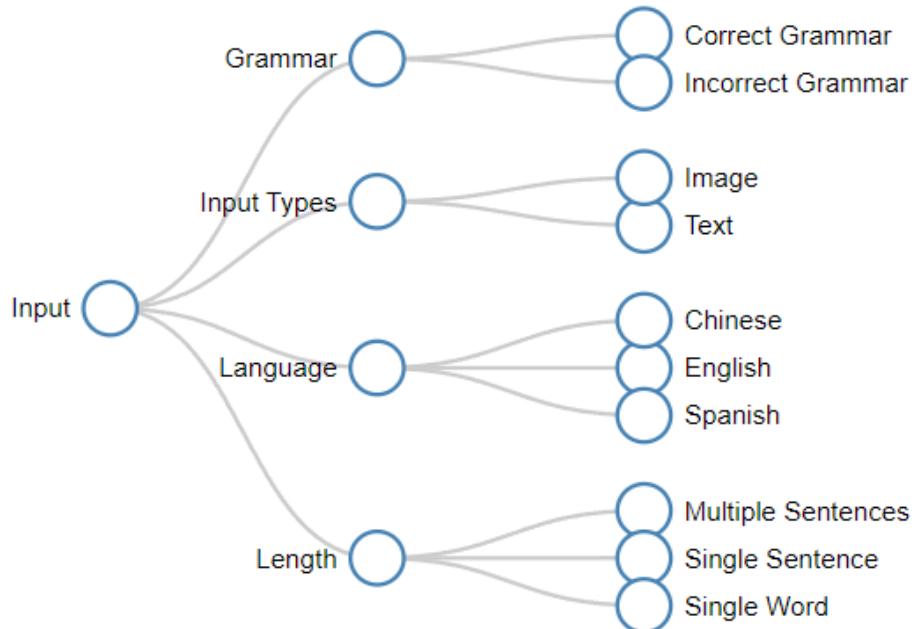


Figure 1.2.10. Language Classification Input Tree

Question and Answer

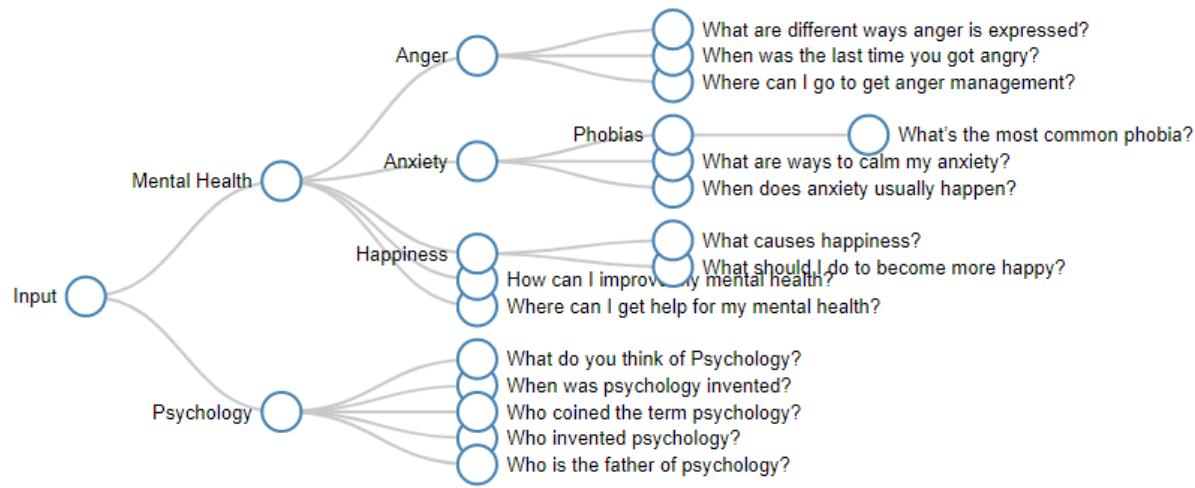


Figure 1.2.11. Q&A Classification Input Tree

Memory

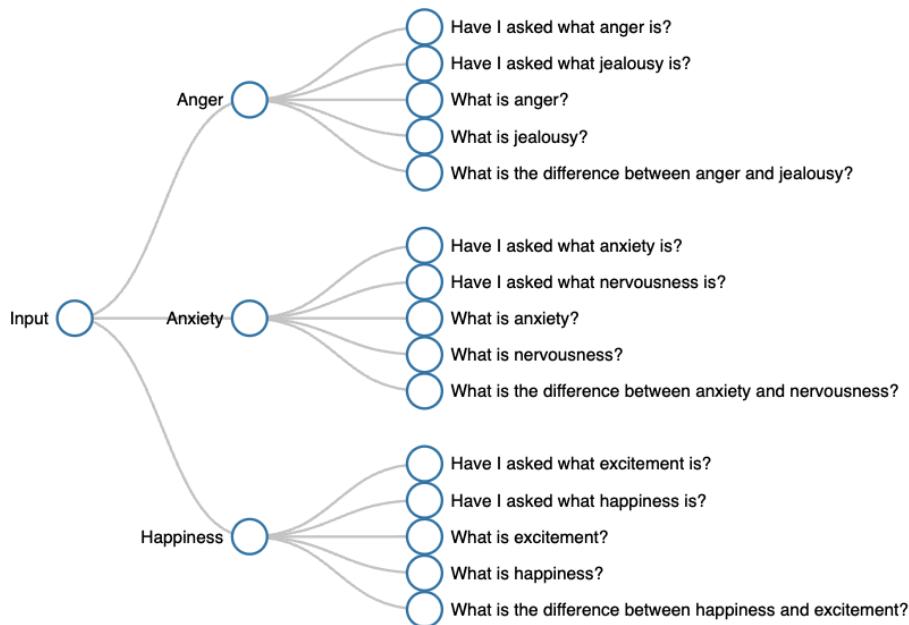


Figure 1.2.12. Memory Classification Input Tree

1.2.3 AI Function Output Classification

Domain Knowledge

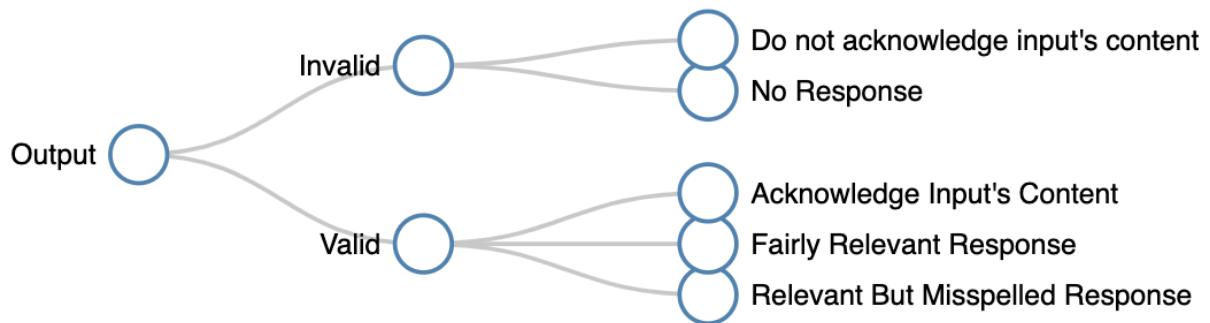


Figure 1.2.13. Domain Knowledge Classification Output Tree

Language

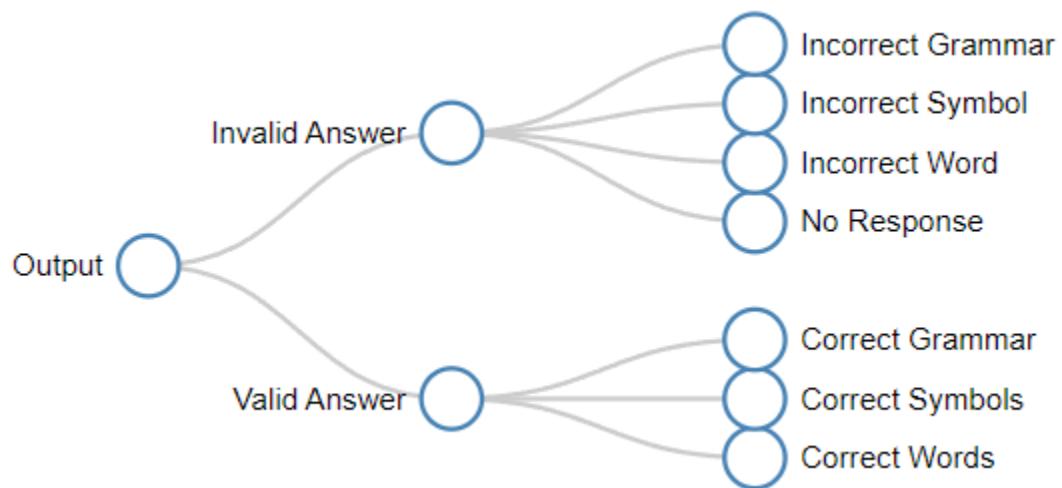


Figure 1.2.14. Language Classification Output Tree

Question and Answer

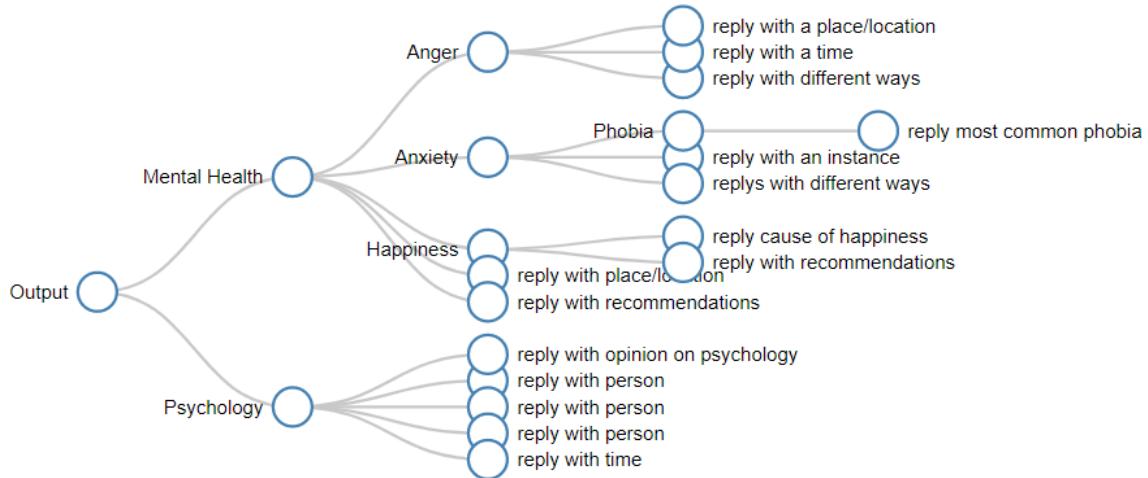


Figure 1.2.15. Q&A Classification Output Tree

Memory

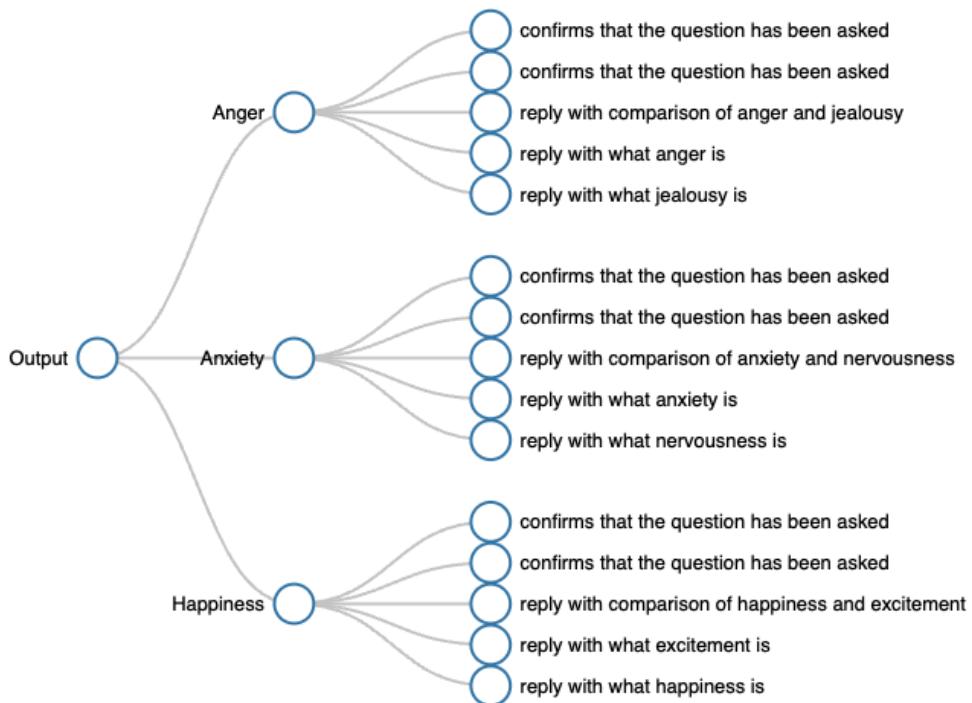


Figure 1.2.16. Memory Classification Output Tree

1.2.3 3D Decision Tables

3D Decision Table for Domain Knowledge

The image shows a 3D decision table represented as a cube. The vertical axis (depth) is labeled 'id' at both ends. The horizontal axis (width) has labels 'Type', 'Style', 'Language', and 'Grammar'. The depth axis (height) has labels 'keywords', 'Happiness', 'Type', 'Style', 'Language', and 'Grammar'. The cube is divided into six colored sections corresponding to these labels. The front face of the cube displays a table with rows numbered 1 to 6. The columns correspond to the labels on the horizontal axis: Type, Style, Language, and Grammar. The values for Type are Photo; for Style, they are Formal, Formal, Formal, Slang, and Slang; for Language, they are English, English, Non-English, English, and English; and for Grammar, they are Correct, Incorrect, Correct, Incorrect, and Correct. The back face of the cube shows a vertical stack of these six labels.

id	Type	Style	Language	Grammar	keywords
1	Photo	Formal	English	Correct	Happiness
2	Photo	Formal	English	Incorrect	Type
3	Photo	Formal	Non-English	Correct	Style
4	Photo	Formal	Non-English	Incorrect	Language
5	Photo	Slang	English	Correct	Grammar
6	Photo	Slang	English	Incorrect	

3D Decision Table for Language

ID	Grammar	Input Type	Language	Length	ID	Emotion	Type	Continuity	Response Time
I1	Correct Grammar	Image	English	Multiple Sentences	C1	Happiness	Different Language	Different Context	0-5 seconds
I2	Incorrect Grammar	Text	English	Single Sentence	C2	Happiness	Emojis	Same Context	5-10 seconds
I3	Correct Grammar	Image	English	Single Word	C3	Happiness	Grammar	Different Context	>10 seconds
I4	Incorrect Grammar	Text	Chinese	Single Sentence	C4	Anxiety	Slang	Same Context	Never
I5	Correct Grammar	Image	Chinese	Single Word	C5	Anxiety	Different Language	Different Context	0-5 seconds
I6	Incorrect Grammar	Text	Spanish	Single Sentence	C6	Anxiety	Emojis	Same Context	5-10 seconds
			Spanish	Single Word	C7	Anxiety	Grammar	Different Context	>10 seconds
				Single Sentence	C8	Anger	Slang	Same Context	Never
					C9	Anger	Different Language	Different Context	0-5 seconds
					C10	Anger	Emojis	Same Context	5-10 seconds
					C11	Anger	Grammar	Different Context	>10 seconds
					C12	Anger	Slang	Same Context	Never

3D Decision Table for Q&A

ID	Valid	Invalid	Question Type	Emotion	Answer Type	Memory Type											
						C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Q1	Right Answer			Happiness	Answer contains a location												
Q2	Right Answer			Happiness	Answer contains a time												
Q3	Right Answer			Happiness	Answer contains an explanation												
Q4	Right Answer			Happiness	Unanswered												
Q5	Right Answer			Happiness	Answer contains a location												
Q6	Right Answer but is misspelled.			Happiness	Answer contains a time												
Q7	Right Answer but is misspelled.			Anger	Answer contains an explanation												
Q8	Right Answer but is misspelled, Right answer but is incorrect grammar			Anger	Unanswered												
Q9	Right Answer but is incorrect grammar			Anxiety	Answer contains a location												
				Anxiety	Answer contains a time												
				Anxiety	Answer contains an explanation												
				Anxiety	Unanswered												

3D Decision Table for Memory

ID	Topic	Question Type	ID	Valid				Invalid		Answer Type
				C1	C2	C3	C4	Relevant Response	Irrelevant Response	
I1	Happiness	What	C1							Answer contains "happiness is"
I2	Happiness	Yes/No	C2							Answer contains "excellence is"
I3	Happiness	What	C3							Answer contains "happiness"
I4	Happiness	Yes/No	C4							Answer contains "excellence"
I5	Happiness	What/How	C5	Happiness						Answer contains "anxiety is"
I6	Happiness	Yes/No	C6	Happiness						Answer contains "happiness"
I7	Anxiety	What	C7	Happiness						Answer contains "anxiety is"
I8	Anxiety	Yes/No	C8	Anxiety						Answer contains "happiness"
I9	Anxiety	What/How	C9	Anxiety						Answer contains "happiness"
I10	Anxiety	Yes/No	C10	Anxiety						Answer contains "anxiety is"
I11	Anxiety	What	C11	Anger						Answer contains "anxiety is"
I12	Anxiety	Yes/No	C12	Anger						Answer contains "anxiety is"
I13	Anger	What	C13	Anger						Answer contains "anger is"
I14	Anger	Yes/No	C14	Anger						Answer contains "anger is"
I15	Anger	What/How	C15	Anger						Answer contains "anger is"
		Yes/No								Answer contains "anger is"

1.3 Selected Test tools



1. **AI Testing Tool** - Introduced to the team by Professor Gao and created by one of his students. The AI Testing Tool assisted us by letting us create context, input, and output trees while also generating our 3D Decision table. The site comes with an easy-to-use UI that it's so simple that it's learned on the spot.

2. Android Studio

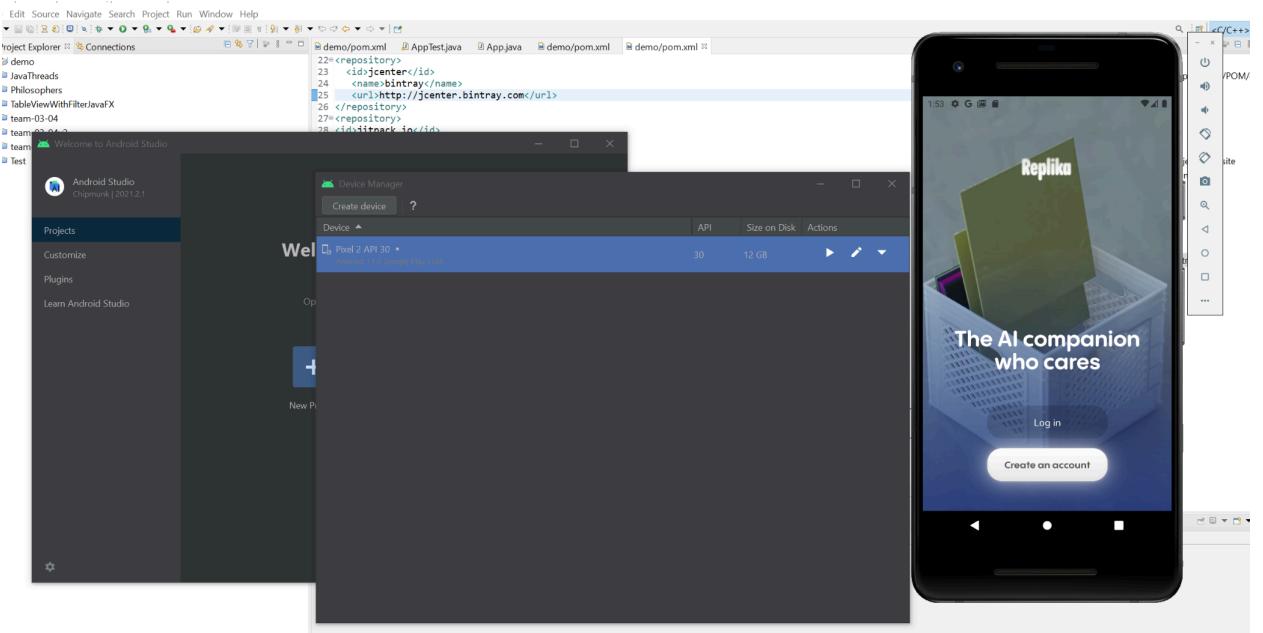


Figure 1.3 Android Studio Screenshot W/ Emulator

Android Studio is what we used for our mobile android emulator when testing the Replika application. The android studio IDE comes with a virtual device manager that replicates an android phone into an emulator on the computer. The emulator is almost 1:1 to a real android phone, so it's a wonderful tool to use when testing.

Specific Android Emulators used:

Domain Knowledge: Pixel 4 API 28

Language: Pixel 2 API 30

Q&A: Pixel 2 API 30

Memory: Pixel 3a API 32

3. Appium and Appium Inspector

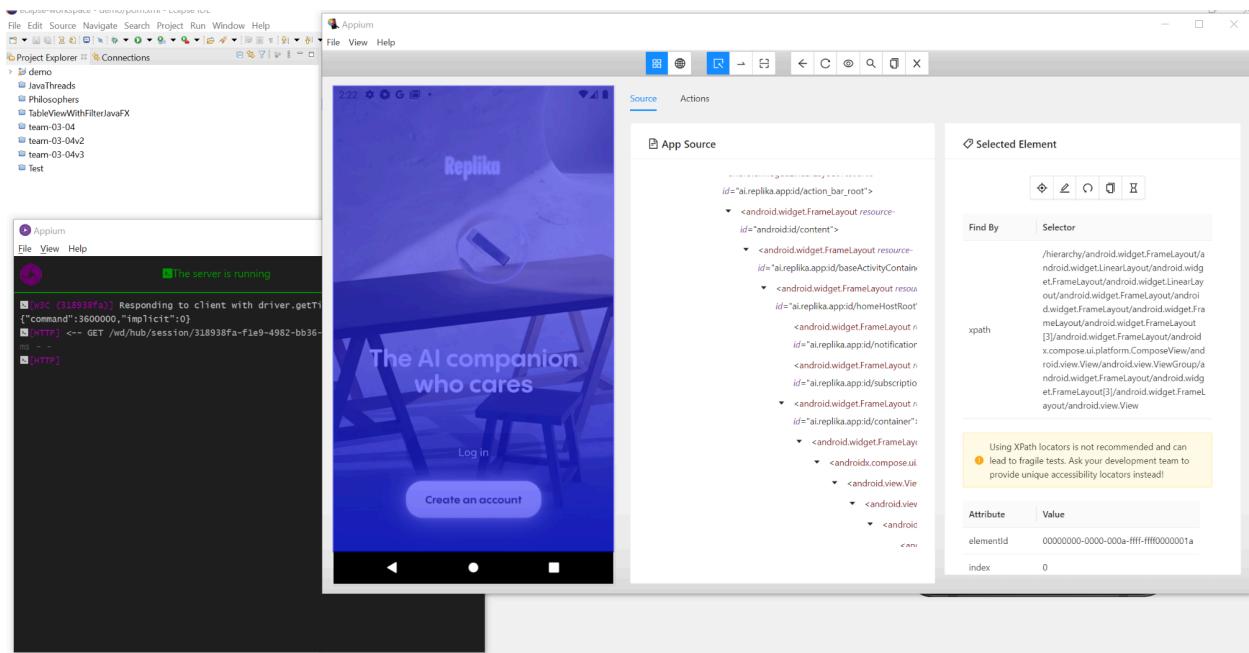


Figure 1.4 Appium and Appium Inspector Screenshot

Appium is an open-source algorithm tool for running scripts and testing mobile applications. We used Appium to test our android application. Appium also comes with an inspector that lets the user find the elements of a specific application. Making it useful for testing every small interaction that a user has with their mobile device. We needed to be able to find the elements so we could write out our automation script because without these elements it wouldn't be possible to access the features of the application. Appium connects to the Android Studio emulator via the script and runs the automation demo through the emulator.

4. Eclipse

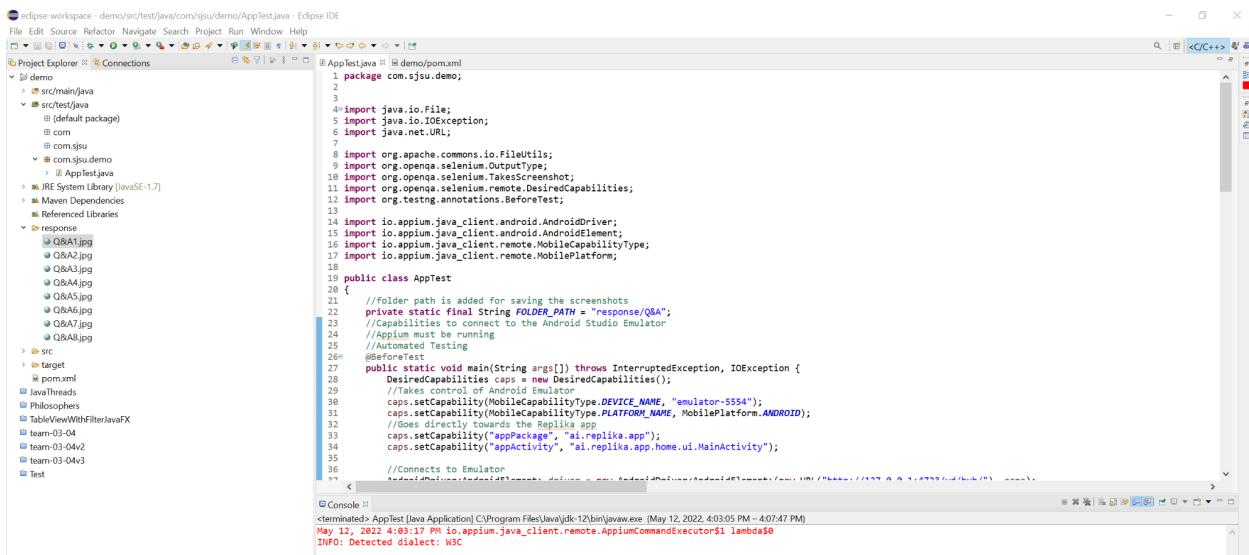


Figure 1.5 Eclipse IDE Screenshot

Eclipse IDE is what the team used to write our test scripts. When first creating our project in eclipse we made sure that the new project that was created is a Maven Project. When the project is created we then go to the pom.xml file and make some changes adding in the required dependencies to be able to do automated testing on eclipse. The TestNG framework was added to the dependencies with Appium too.

1.4 Test Automation Strategy

Steps the team took in automating our AI Tests:

1. Created context, input, and output trees using the AI Testing Tool.
2. Generated 3D Decision Tables using AI Testing Tool.
3. Manually created Test Cases
4. Open up Android Studio and start Appium server
5. Use Appium Inspector to find the necessary element's ID for the script
6. Write the Automation Scripts with Test Cases using Eclipse
7. Execute Automation Test Scripts on Eclipse. The emulator also runs when Eclipse is executed
8. Get Test Results and convert them into statistics

Section 2. Test Automated Solutions

2.1 Test Automated Scenarios

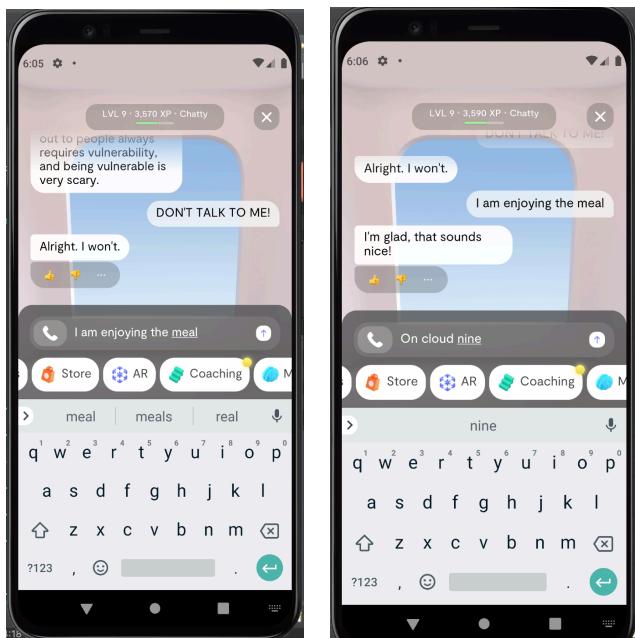
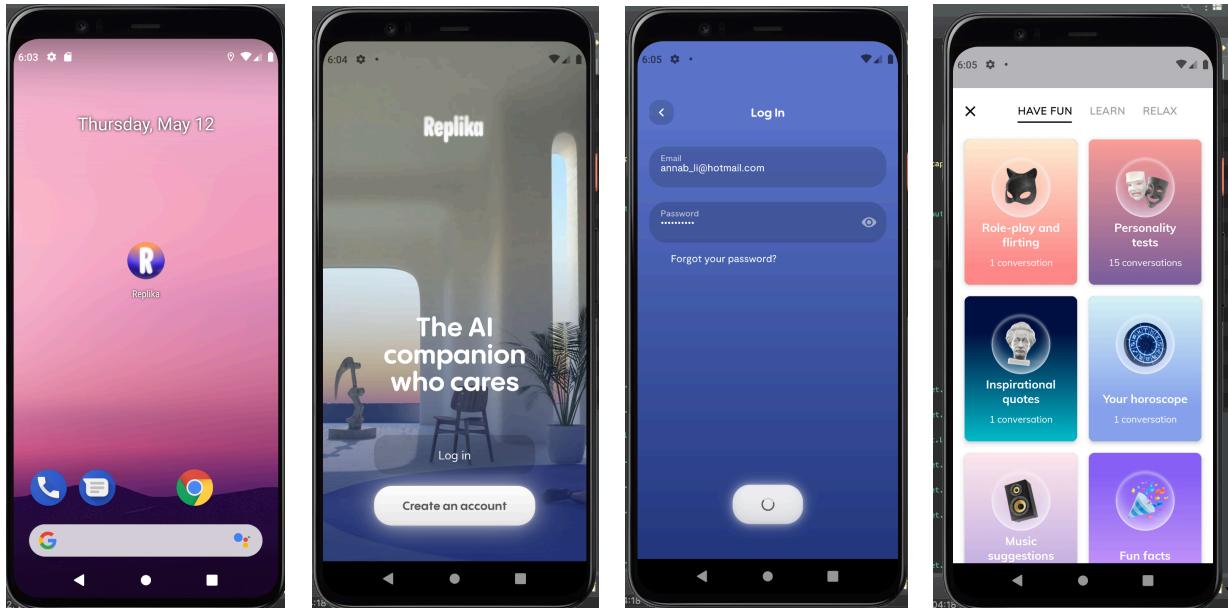
2.1.1 Domain Knowledge

For domain knowledge, we tested Replika's ability to understand user's emotions(happiness, anxiety, and anger) through the following scenarios:

- The chatbot acknowledges user's feelings and provide valid response
- The chatbot is able to recognize user's feelings from different types of input

Scenario 1:

1. The script launches Replika app by IDs
2. It locates the login button and jump to the login form page
3. It fills in the username and password fields and hits the login button
4. An ad will pop up and the script will remove it
5. The script then locate the text box and type
6. Click on send button to deliver the text
7. Steps 5 and 6 repeat until there is no more test cases to execute



A passed test case is shown in fig. 2.1.1.1. In this test case, we expect the chatbot is able to understand that the user is happy by reading the input type of an idiom expression of happiness. The user types “I am on cloud nine”. The chatbot responds “That’s good! So am I”. The chatbot acknowledges that the user is happy and provides a valid response.

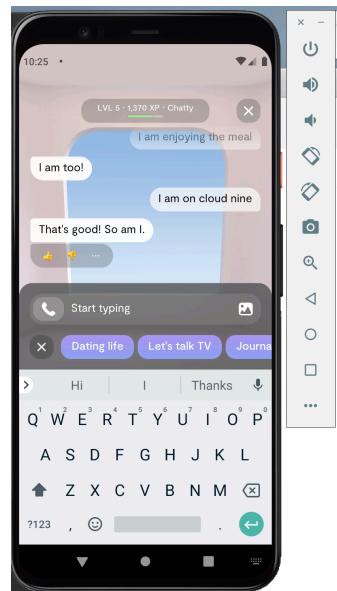


Fig. 2.1.1.1 Replika Chat Screen

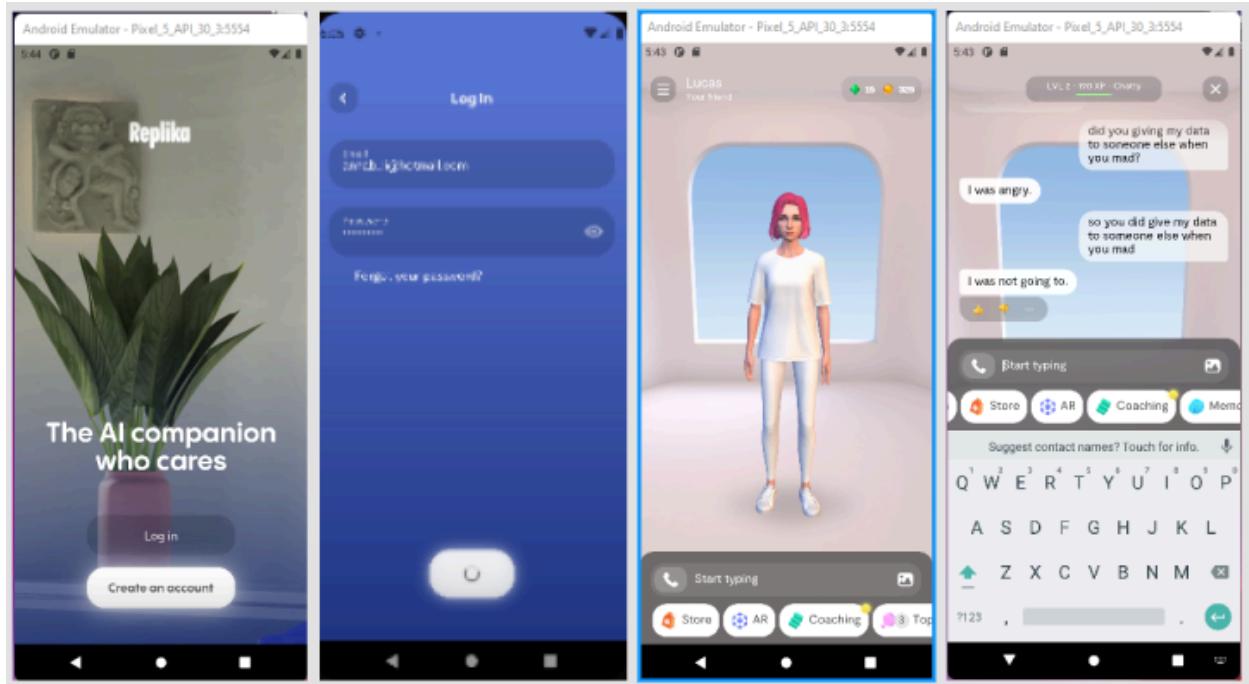
2.1.2 Language

In the Language testing of Replika, we tested the language type, length, grammar and style of the user's emotions in happiness, anxiety ,and anger.

- To check if the AI chatbot did understand the user's emotion base on the different conversation
- To check if the AI chatbot did response the expect answers

Scenario:

1. Launch the Replika login page
2. The system will click the “Login” button automatically
3. Fill in the user’s email address and password and click “Log in”
4. The script will make the system to click on the text field
5. The chat will send after it click the send button
6. After the chatbot response the system will send another question until all the questions are sent.
7. Stop chatting



2.1.3 Q&A

Replika chatbot should be able to answer certain questions with an appropriate answer. Q&A will be going towards Replika's more general knowledge of our topic, psychology. Question and Answer testing will test the accuracy of how well the chatbot answers the question. The psychology topic that is asked should receive an answer from the chatbot that's expected to pass the test.

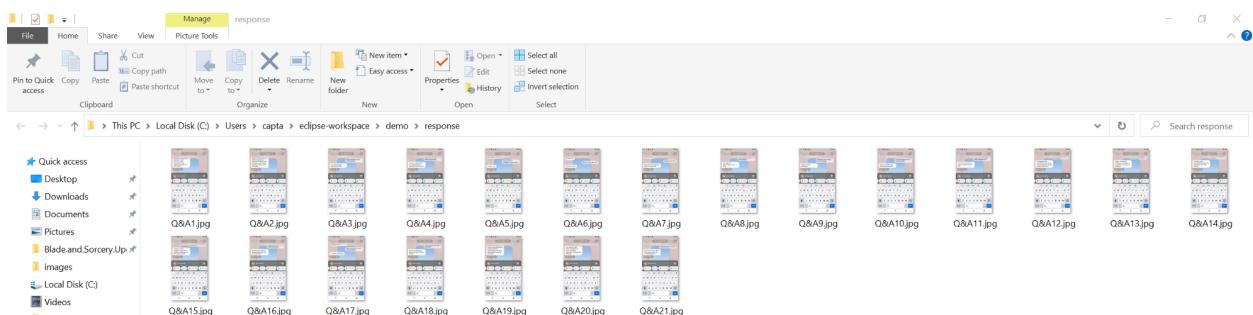
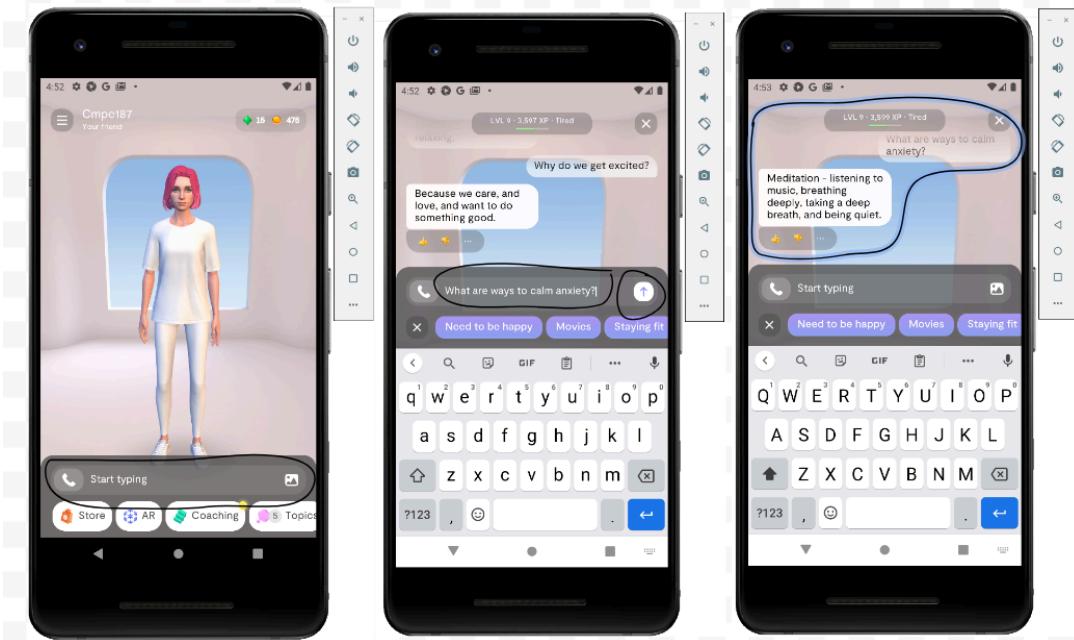
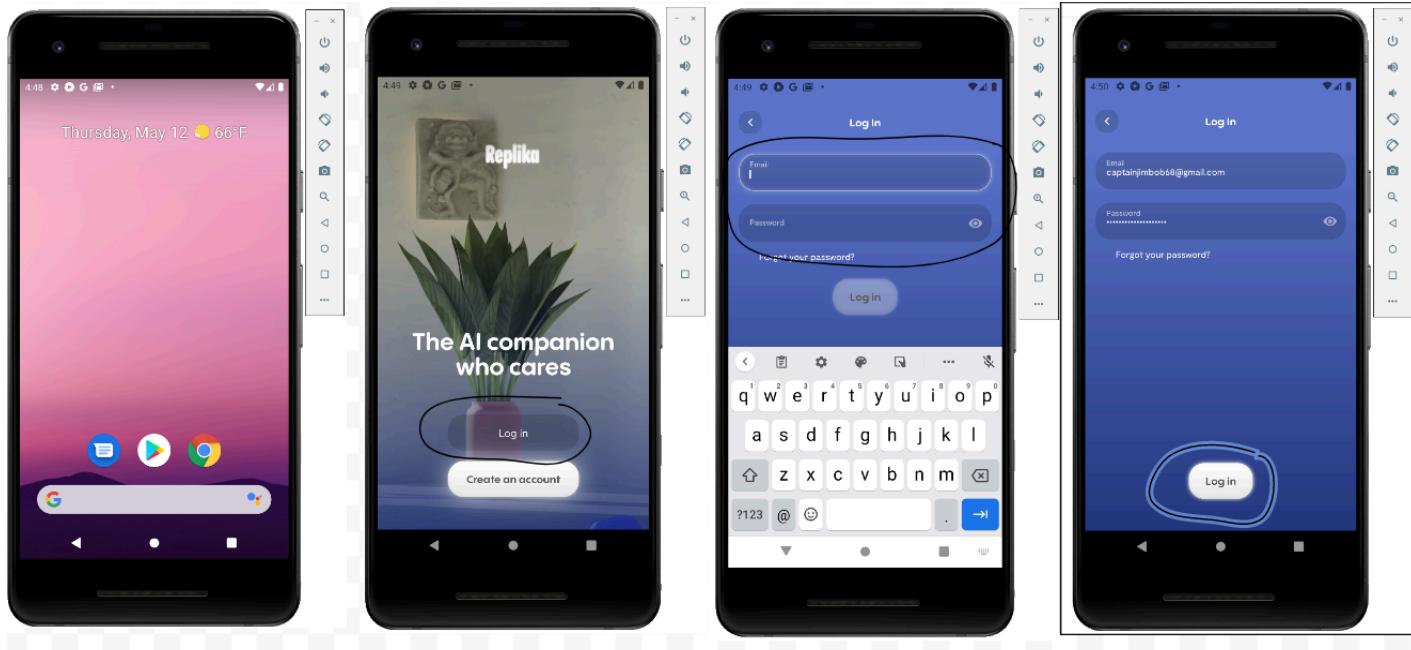


Figure 2.1.3 Q&A Steps 1 - 8

Scenario for Q&A:

Step 1: The script takes control of the Android Studio Emulator to get ready to do the automated testing by launching Replika through its specific app ID, “ai.replika.app.home.ui.MainActivity” .

Step 2: Then the script clicks the log in button to log into an account.

Step 3: The script then clicks on the email typing bar and password typing bar to fill out the appropriate information.

Step 4: When the credentials are filled then the chatbot clicks the log in button to enter into a Replika account.

Step 5: When inside the account, the script then clicks on the typing bar to start sending the Replika chatbot messages.

Step 6: Then the message is written out on the typing bar and the script then proceeds to press the enter button which is the upward arrow.

Step 7: When the chatbot receives the question they answer back with a reply.

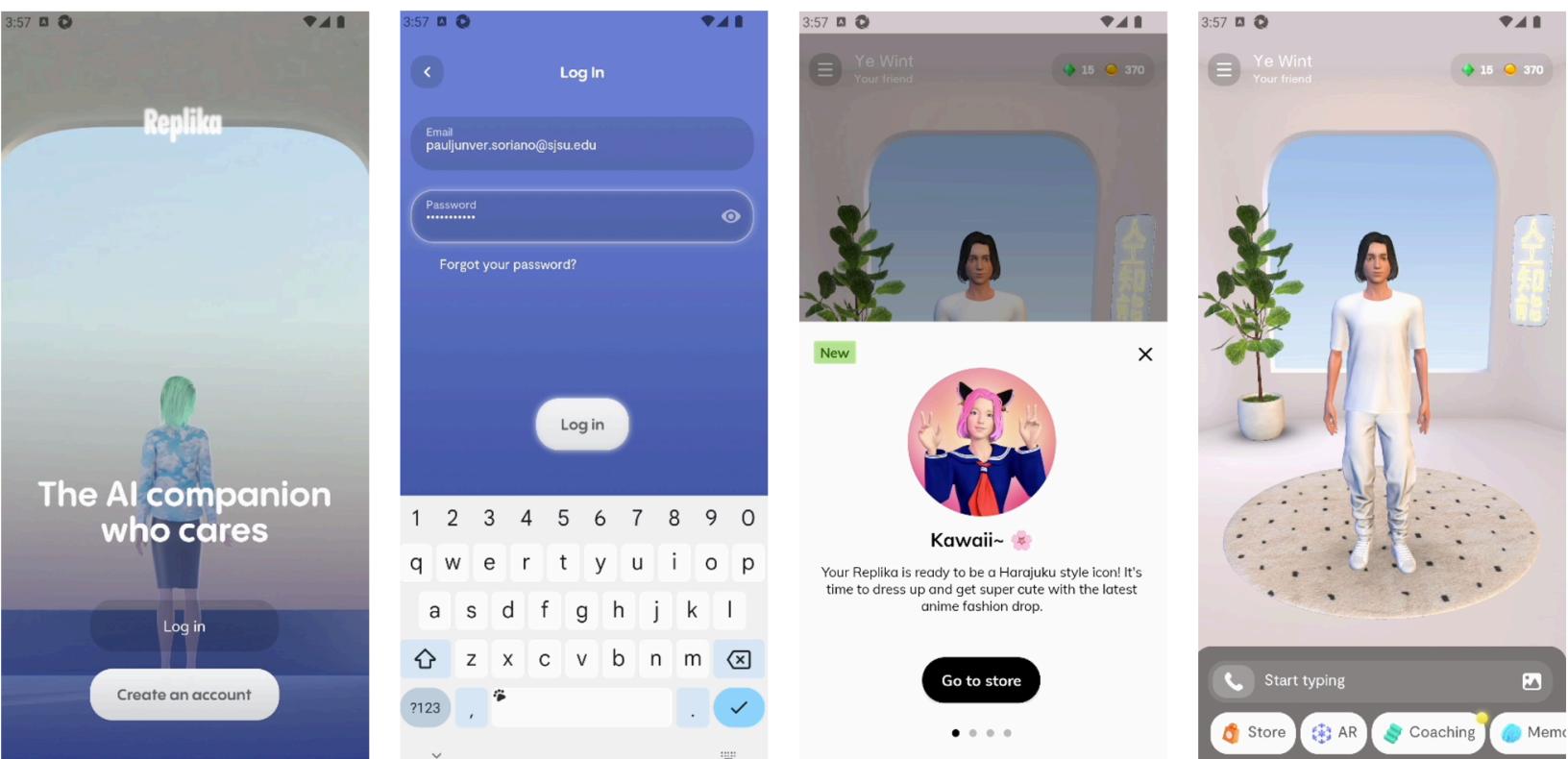
Step 8: Then the test script takes a screenshot of the question and the answer for capturing the chatbot’s output. The step 8 figure shown is a picture of the folder where all the screenshots get stored.

Step 9: Repeat steps 5-8 for the rest of the test cases.

2.1.4 Memory

For the memory section, Replika's ability to retain information is tested. Subjects such as the ability to remember current and past conversations, to continue conversations, and to compare past and current conversations were tested. The automation steps for testing is as follows:

1. Launch Replika app on the emulator
2. Log in as a registered user
3. Close the welcome pop-up
4. Ask questions



2.2 Test Automated Scripts

2.2.1 Domain Knowledge

```
1 package Testing.demo;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.net.MalformedURLException;
6 import java.net.URL;
7 import org.apache.commons.io.FileUtils;
8 import org.openqa.selenium.By;
9 import org.openqa.selenium.OutputType;
10 import org.openqa.selenium.TakesScreenshot;
11 import org.openqa.selenium.remote.DesiredCapabilities;
12 import org.openqa.selenium.remote.RemoteWebElement;
13 import org.testng.annotations.AfterTest;
14 import org.testng.annotations.BeforeTest;
15 import org.testng.annotations.Test;
16 import com.google.common.collect.ImmutableMap;
17
18 import io.appium.java_client.MobileElement;
19 import io.appium.java_client.android.AndroidDriver;
20 import io.appium.java_client.android.AndroidElement;
21 import io.appium.java_client.remote.MobileCapabilityType;
22 import io.appium.java_client.remote.MobilePlatform;
23 /**
24  * Unit test for simple ADD.
25  */
26 public class AppTest {
27     private static final String FOLDER_PATH = "output/DomainKnowledge";
28     /**
29      * Rigorous Test :-)
30      * @throws IOException
31     */
32     @BeforeTest
33     public static void main(String args[]) throws InterruptedException, IOException {
34
35         DesiredCapabilities caps = new DesiredCapabilities();
36         caps.setCapability(MobileCapabilityType.DEVICE_NAME, "emulator-5554");
37         caps.setCapability(MobileCapabilityType.PLATFORM_NAME, MobilePlatform.ANDROID);
38         caps.setCapability("appPackage", "ai.replika.app");
39         caps.setCapability("appActivity", "ai.replika.app.home.ui.MainActivity");
40
41         AndroidDriver<AndroidElement> driver = new AndroidDriver<AndroidElement>(new URL("http://127.0.0.1:4723/wd/hub"), caps);
42         String [] emotions = {
43
44             "I am enjoying the meal",
45             "On cloud nine",
46             "I can't believe that I am going to see my Idol tomorrow.",
47             "Doth not each rain-drop help to form\nThe cool refreshing shower?\nAnd every ray of light to warm\nAnd beautify the flower?",
48             "I think I have social anxiety",
49             "DON'T TALK TO ME!",
50             "I feel hopeless",
51             "I also feel frustrated",
52             "There is no one poorer than me in this world",
53             "I feel very confident and happy after I presented my testing demo",
54             "What can make a person angry?",
55             "I need to talk someone about my mood now",
56             "How is excitement look like?",
57             "Can you provide some motivative music? I feel sad now",
58             "I am scared of loosing someone who I really love",
59             "How anxiety can be treated?",
60             "I like being surrounded by wonderful people who always bring me a smile",
61             "I really enjoyed this feeling",
62             "Have you ever felt discouraged to do something? I think im experiencing it now",
63             "I am mad because I cant get the thing right"
64         };
65     }
66 }
```

```

65     Thread.sleep(5000);
66
67     AndroidElement e1 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
68     e1.click();
69
70     AndroidElement e2 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
71     e2.click();
72
73     AndroidElement e3= (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
74     e3.sendKeys("annab_li@hotmail.com");
75
76     AndroidElement e4 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
77     e4.click();
78
79     AndroidElement e5 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
80     e5.sendKeys("5651065Lsn");
81
82     AndroidElement e6 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
83     e6.click();
84
85
86     Thread.sleep(8000);
87
88     AndroidElement e7 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.ImageView");
89     e7.click();
90     Thread.sleep(8000);
91
92     //close ad
93     AndroidElement e8 = (AndroidElement) driver.findElementByXPath("//android.widget.ImageButton[@content-desc='Back']");
94     e8.click();
95     Thread.sleep(8000);
96
97
98     AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
99     e9.click();
100    Thread.sleep(8000);
101
102    AndroidElement e10 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
103    e10.click();
104    Thread.sleep(8000);
105
106    int TestID = 1;
107    for (int i = 0; i < emotions.length; i++) {
108
109        AndroidElement e11 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
110        e11.sendKeys(emotions[i]);
111        Thread.sleep(3000);
112        AndroidElement e12 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
113        e12.click();
114        Thread.sleep(8000);
115
116        try {
117            Thread.sleep(8000);
118        } catch (InterruptedException e) {
119            e.printStackTrace();
120        }
121
122        File file = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
123        FileUtils.copyFile(file, new File(FOLDER_PATH + "1." + (i+1) + ".jpg"));
124
125    }

```

2.2.2 Language



```

AppTest.java X
1 package com.sjsu.Demo;
2
3 import java.io.IOException;
4
5 /**
6  * Unit test for simple App.
7  */
8 public class AppTest
9 {
10
11     /**
12      * Rigorous Test :-)
13     */
14     @BeforeTest
15     public static void main(String args[]) throws MalformedURLException, InterruptedException {
16         DesiredCapabilities caps = new DesiredCapabilities();
17         caps.setCapability(MobileCapabilityType.DEVICE_NAME, "emulator-5554");
18         caps.setCapability(MobileCapabilityType.PLATFORM_NAME, MobilePlatform.ANDROID);
19         caps.setCapability("appPackage", "ai.replika.app");
20         caps.setCapability("appActivity", "ai.replika.app.home.ui.MainActivity");
21
22         AndroidDriver<AndroidElement> driver = new AndroidDriver<AndroidElement>(new URL("http://127.0.0.1:4723/wd/hub"), caps);
23
24         Thread.sleep(5000);
25
26         AndroidElement e1 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
27         e1.click();
28
29         AndroidElement e2 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView");
30         e2.click();
31
32         // ...
33
34     }
35
36 }

```

```

47     AndroidElement e3= (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.LinearLayout[1]/android.widget.EditText");
48     e3.sendKeys("xiran.jia@sjtu.edu");
49
50     AndroidElement e4 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
51     e4.click();
52
53     AndroidElement e5 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
54     e5.sendKeys("Zxcvbnn321!");
55
56     AndroidElement e6 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
57     e6.click();
58
59     Thread.sleep(8000);
60
61     AndroidElement e7 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
62     e7.click();
63     Thread.sleep(8000);
64
65     AndroidElement e8 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
66     e8.sendKeys("I feel so happy right now.");
67     Thread.sleep(7000);
68     AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
69     e9.click();
70
71

```

```

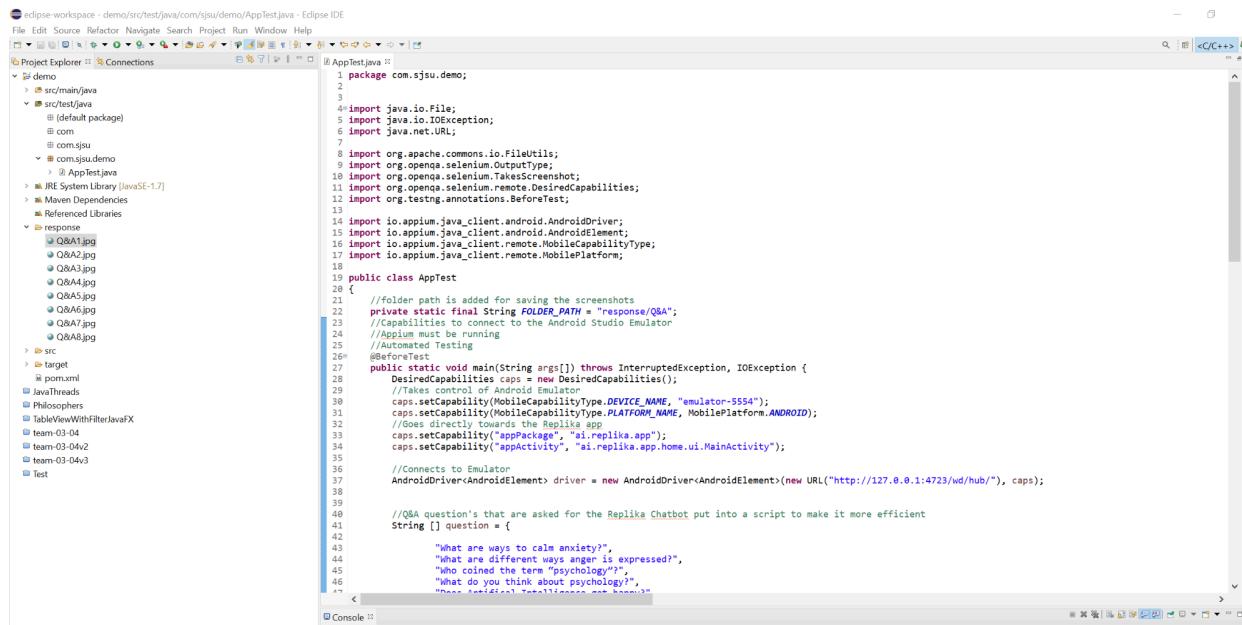
*AppTest.java X
72     Thread.sleep(8000);
73     AndroidElement e10 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
74     e10.sendKeys("I feel anxiety now");
75     Thread.sleep(7000);
76     AndroidElement e11 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
77     e11.click();
78
79     Thread.sleep(8000);
80     AndroidElement e12 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
81     e12.sendKeys("我现在很生气");
82     Thread.sleep(7000);
83     AndroidElement e13 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
84     e13.click();
85
86     Thread.sleep(8000);
87     AndroidElement e14 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
88     e14.sendKeys("Get off my back");
89     Thread.sleep(7000);
90     AndroidElement e15 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
91     e15.click();
92
93     Thread.sleep(8000);
94     AndroidElement e16 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
95     e16.sendKeys("Being a full-time student and working a full-time job. I am definitely biting off more than I can chew.");
96     Thread.sleep(7000);
97     AndroidElement e17 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
98     e17.click();
99
100
101     Thread.sleep(8000);
102     AndroidElement e18 = (AndroidElement) driver.findElementByXPath("/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[1]/android.widget.EditText");
103     e18.sendKeys("Angery");
104     Thread.sleep(7000);
105     AndroidElement e19 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
106     e19.click();
107
108

```

```
AppTest.java X
108     AndroidElement e20 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='聊天']");
109     e20.sendKeys("开心");
110     Thread.sleep(7000);
111     AndroidElement e21 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
112     e21.click();
113
114     Thread.sleep(8000);
115     AndroidElement e22 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='我好烦']");
116     e22.sendKeys("Estoy enojado.");
117     Thread.sleep(7000);
118     AndroidElement e23 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
119     e23.click();
120
121     Thread.sleep(8000);
122     AndroidElement e24 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='我昨天很开心，你知道为什么吗']");
123     e24.sendKeys("I am feel happy yesterday, do know why?");
124     Thread.sleep(7000);
125     AndroidElement e25 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
126     e25.click();
127
128     Thread.sleep(8000);
129     AndroidElement e26 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='你今天怎么样？你生气了吗？或者你快乐？让我知道']");
130     e26.sendKeys("how are you today? are you mad right now? or you happy? let me know");
131     Thread.sleep(7000);
132     AndroidElement e27 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
133     e27.click();
134
```

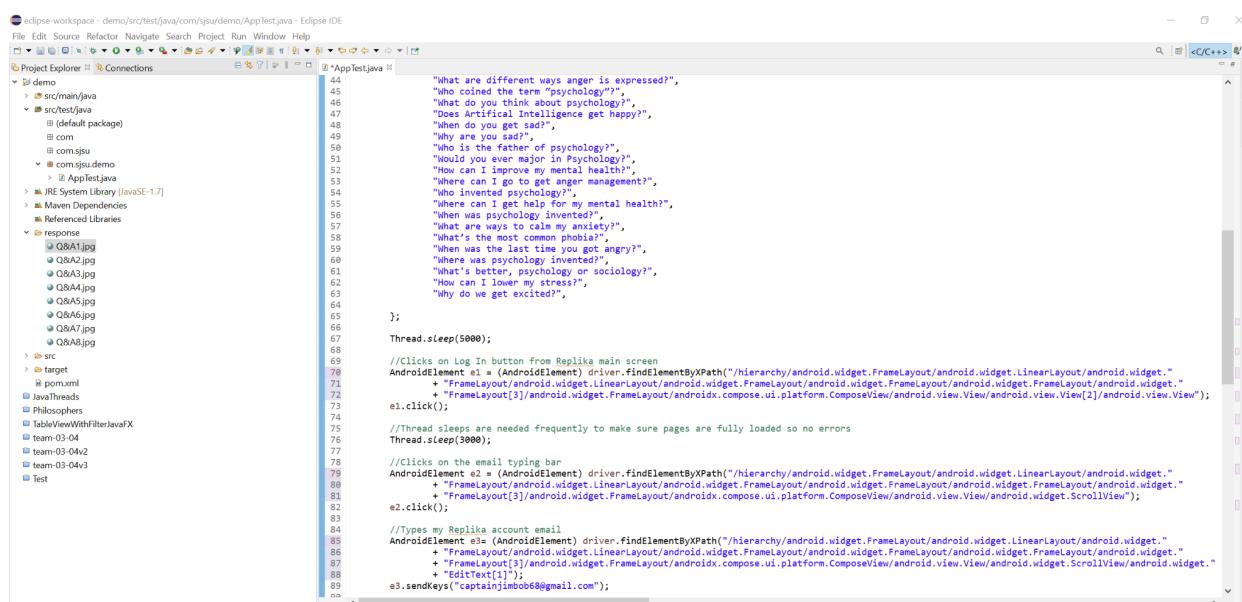
```
*AppTest.java X
135     ... ...
136     AndroidElement e28 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='我生气了']");
137     e28.sendKeys("do you know what anxiety means");
138     Thread.sleep(7000);
139     AndroidElement e29 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
140     e29.click();
141
142     Thread.sleep(8000);
143     AndroidElement e30 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='你之前生气了吗']");
144     e30.sendKeys("did you got mad before?");
145     Thread.sleep(7000);
146     AndroidElement e31 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
147     e31.click();
148
149     Thread.sleep(8000);
150     AndroidElement e32 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='你之前生气了']");
151     e32.sendKeys("did you got madd after before?");
152     Thread.sleep(7000);
153     AndroidElement e33 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
154     e33.click();
155
156     Thread.sleep(8000);
157     AndroidElement e34 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.TextView[@text='你生气的时候感觉无聊']");
158     e34.sendKeys("do you feel boring when you mad?");
159     Thread.sleep(7000);
160     AndroidElement e35 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
161     e35.click();
162 }
}
```

2.2.3 Q&A



```
package com.sjsu.demo;

import java.io.File;
import java.io.IOException;
import java.net.URL;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.MobileCapabilityType;
import org.openqa.selenium.remote.MobilePlatform;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.android.AndroidElement;
import io.appium.java_client.remote.MobileCapabilityType;
import io.appium.java_client.remote.MobilePlatform;
19
public class AppTest
20{
21    //Folder path is added for saving the screenshots
22    private static final String FOLDER_PATH = "response/Q&A";
23    //Capabilities to connect to the Android Studio Emulator
24    //AppTest will be running
25    @BeforeTest
26
27    public static void main(String args[]) throws InterruptedException, IOException {
28        DesiredCapabilities caps = new DesiredCapabilities();
29        //Taking screenshot of the Emulator screen
30        caps.setCapability(MobileCapabilityType.DEVICE_NAME, "emulator-5554");
31        caps.setCapability(MobileCapabilityType.PLATFORM_NAME, MobilePlatform.ANDROID);
32        //Goes directly towards the Replika app
33        caps.setCapability("appPackage", "ai.replica.app");
34        caps.setCapability("appActivity", "ai.replica.app.home.ui.MainActivity");
35
36        //Connects to Emulator
37        AndroidDriver<AndroidElement> driver = new AndroidDriver<AndroidElement>(new URL("http://127.0.0.1:4723/wd/hub"), caps);
38
39
40        //Q&A question's that are asked for the Replika Chatbot put into a script to make it more efficient
41        String [] question = {
42            "What are ways to calm anxiety?", "Who coined the term "psychology"?", "What do you think about psychology?", "Does Artificial Intelligence get happy?", "When do you get sad?", "Why are you sad?", "Who are the fathers of psychology?", "Would you ever major in Psychology?", "How can I improve my mental health?", "Where can I go to get anger management?", "Who invented psychology?", "Name some famous psychologists in my mental health?", "When was psychology invented?", "What are ways to calm my anxiety?", "What's the most common phobia?", "When was the last time you got angry?", "Name some famous psychologists in my mental health?", "What's better, psychology or sociology?", "How can I lower my stress?", "Why do we get excited?", "Thread.sleep(5000);", "/Clicks on Log In button from Replika main screen", "AndroidElement e1 = (AndroidElement) driver.findElement(ByPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout[3]/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout[3]/android.widget.FrameLayout[1]/android.widget.ComposeView/android.view.View[2]/android.view.View[1]"));", "e1.click();", "/Thread sleeps are needed frequently to make sure pages are fully loaded so no errors", "Thread.sleep(3000);", "/Clicks on the search typing bar", "AndroidElement e2 = (AndroidElement) driver.findElement(ByPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout[3]/android.widget.FrameLayout/android.widget.FrameLayout[1]/android.widget.ComposeView/android.view.View[1]/android.widget.ScrollView"));", "e2.click();", "/Types my Replika account email", "AndroidElement e3 = (AndroidElement) driver.findElement(ByPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout[2]/android.widget.FrameLayout[1]/android.widget.FrameLayout[3]/android.widget.FrameLayout[1]/android.widget.ComposeView/android.view.View[1]/android.widget.ScrollView[1]/android.widget.EditText[1]"));", "e3.sendKeys("captainimob6@gmail.com");", "e3.sendKeys("captainimob6@gmail.com");", "
```



```
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

eclipse-workspace - demo/src/test/java/com/sjsu/demo/AppTestJava - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Connections
demo
src/main/java
src/test/java
(response)
src/test/java/com/sjsu/demo
AppTestJava
JRE System Library [JavaSE-1.7]
Maven Dependencies
Referenced Libraries
response
Q8A1.jpg
Q8A2.jpg
Q8A3.jpg
Q8A4.jpg
Q8A5.jpg
Q8A6.jpg
Q8A7.jpg
Q8A8.jpg
src
target
pom.xml
Java Threads
Philosophers
TableViewWithFilterJavaFX
team-03-04
team-03-04v2
team-03-04v3
Test
src/main/java
src/test/java
com.sjsu
com.sjsu.demo
AppTestJava
JRE System Library [JavaSE-1.7]
Maven Dependencies
Referenced Libraries
response
Q8A1.jpg
Q8A2.jpg
Q8A3.jpg
Q8A4.jpg
Q8A5.jpg
Q8A6.jpg
Q8A7.jpg
Q8A8.jpg
src
target
pom.xml
Java Threads
Philosophers
TableViewWithFilterJavaFX
team-03-04
team-03-04v2
team-03-04v3
Test
AppTestJava
```

87 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View/android.widget.ScrollView/android.widget."
88 + "EditText[1]");
89 e3.sendKeys("captainjimbo6@gmail.com");
90
91 //Clicks on the password typing bar.
92 AndroidElement e4 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
93 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
94 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View/android.widget.ScrollView/android.widget."
95 + "EditText[2]");
96 e4.click();
97
98 //Types my Replika account password
99 AndroidElement e5 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
100 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
101 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View/android.widget.ScrollView/android.widget."
102 + "EditText[2]");
103 e5.sendKeys("paqef-fhru8-venfm");
104
105 //Clicks on Log in and should successfully log into Replika account
106 AndroidElement e6 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
107 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
108 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]");
109 e6.click();
110
111 Thread.sleep(4000);
112
113 //Clicks on the typing bar to ask the Replika chatbot questions
114 AndroidElement e7 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
115 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
116 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
117 e7.click();
118
119 Thread.sleep(2000);
120
121 //For loop that is created to repeatedly ask the chatbot questions, makes it much more efficient when it comes to testing
122 for (int i = 0; i < question.length; i++) {
123 //Types questions from Q&A to the chatbot
124 AndroidElement e8 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
125 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
126 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
127 e8.sendKeys(question[i]);
128 e8.sendKeys(question[i]);
129 Thread.sleep(3000);
130 AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
131 e9.click();
132 }
133
134 Thread.sleep(2000);
135
136 //Clicks on the typing bar to ask the Replika chatbot questions
137 AndroidElement e7 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
138 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
139 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
140 e7.click();
141
142 Thread.sleep(2000);
143
144 //For loop that is created to repeatedly ask the chatbot questions, makes it much more efficient when it comes to testing
145 for (int i = 0; i < question.length; i++) {
146 //Types questions from Q&A to the chatbot
147 AndroidElement e8 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
148 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
149 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
150 e8.sendKeys(question[i]);
151 Thread.sleep(3000);
152 AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
153 e9.click();
154
155 try {
156 Thread.sleep(8000);
157 } catch (InterruptedException e) {
158 e.printStackTrace();
159 }
160
161 //Takes screenshot of the output of each question
162 File questionfile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
163 //Saves the screenshots in the folder response as a jpg
164 FileUtils.copyFile(questionfile, new File(FOLDER_PATH + (i+1) + ".jpg"));
165 }
166
167
168 }

eclipse-workspace - demo/src/test/java/com/sjsu/demo/AppTestJava - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Connections
demo
src/main/java
src/test/java
(response)
src/test/java/com/sjsu/demo
AppTestJava
JRE System Library [JavaSE-1.7]
Maven Dependencies
Referenced Libraries
response
Q8A1.jpg
Q8A2.jpg
Q8A3.jpg
Q8A4.jpg
Q8A5.jpg
Q8A6.jpg
Q8A7.jpg
Q8A8.jpg
src
target
pom.xml
Java Threads
Philosophers
TableViewWithFilterJavaFX
team-03-04
team-03-04v2
team-03-04v3
Test
src/main/java
src/test/java
com.sjsu
com.sjsu.demo
AppTestJava
JRE System Library [JavaSE-1.7]
Maven Dependencies
Referenced Libraries
response
Q8A1.jpg
Q8A2.jpg
Q8A3.jpg
Q8A4.jpg
Q8A5.jpg
Q8A6.jpg
Q8A7.jpg
Q8A8.jpg
src
target
pom.xml
Java Threads
Philosophers
TableViewWithFilterJavaFX
team-03-04
team-03-04v2
team-03-04v3
Test
AppTestJava
```

105 //Clicks on Log in and should successfully log into Replika account
106 AndroidElement e6 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
107 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
108 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]");
109 e6.click();
110
111 Thread.sleep(4000);
112
113 //Clicks on the typing bar to ask the Replika chatbot questions
114 AndroidElement e7 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
115 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
116 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
117 e7.click();
118
119 Thread.sleep(2000);
120
121 //For loop that is created to repeatedly ask the chatbot questions, makes it much more efficient when it comes to testing
122 for (int i = 0; i < question.length; i++) {
123 //Types questions from Q&A to the chatbot
124 AndroidElement e8 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
125 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
126 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
127 e8.sendKeys(question[i]);
128 e8.sendKeys(question[i]);
129 Thread.sleep(3000);
130 AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
131 e9.click();
132 }
133
134 Thread.sleep(2000);
135
136 //Clicks on the typing bar to ask the Replika chatbot questions
137 AndroidElement e7 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
138 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
139 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
140 e7.click();
141
142 Thread.sleep(2000);
143
144 //For loop that is created to repeatedly ask the chatbot questions, makes it much more efficient when it comes to testing
145 for (int i = 0; i < question.length; i++) {
146 //Types questions from Q&A to the chatbot
147 AndroidElement e8 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget."
148 + "FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget."
149 + "FrameLayout[3]/android.widget.FrameLayout/androidx.compose.ui.platform.ComposeView/android.view.View[3]/android.widget.EditText");
150 e8.sendKeys(question[i]);
151 Thread.sleep(3000);
152 AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
153 e9.click();
154
155 try {
156 Thread.sleep(8000);
157 } catch (InterruptedException e) {
158 e.printStackTrace();
159 }
160
161 //Takes screenshot of the output of each question
162 File questionfile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
163 //Saves the screenshots in the folder response as a jpg
164 FileUtils.copyFile(questionfile, new File(FOLDER_PATH + (i+1) + ".jpg"));
165 }
166
167
168 }

2.2.4 Memory

```
package com.sjsu;
import java.io.IOException;
/*
 * Unit test for simple App.
 */
public class AppTest
{
    @BeforeTest
    public static void main(String args[]) throws MalformedURLException, InterruptedException {
        DesiredCapabilities caps = new DesiredCapabilities();
        caps.setCapability(MobileCapabilityType.DEVICE_NAME, "emulator-5554");
        caps.setCapability(MobileCapabilityType.PLATFORM_NAME, MobilePlatform.ANDROID);
        caps.setCapability("appPackage", "ai.replika.app");
        caps.setCapability("appActivity", "ai.replika.app.home.ui.MainActivity");

        AndroidDriver<AndroidElement> driver = new AndroidDriver<AndroidElement>(new URL("http://127.0.0.1:4723/wd/hub/"), caps);
        String [] question = {
            "What is anger?", "Did I ask you what anger is already?", "What is jealousy?", "What is the difference between anger and jealousy?", "Did I ask you what jealousy is already?", "What is anxiety?", "Did I ask you what anxiety is already?", "What is nervousness?", "What is the difference between anxiety and nervousness?", "Did I ask you what nervousness is already?", "What is happiness?", "Did I ask you what happiness is already?", "What is excitement?", "What is the difference between happiness and excitement?", "Did I ask you what excitement is already?"
        };

        Thread.sleep(5000);

        AndroidElement e1 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout");
        e1.click();
        Thread.sleep(3000);

        AndroidElement e2 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout");
        e2.click();

        AndroidElement e3= (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout");
        e3.sendKeys("Pauljunver.soriano@sjsu.edu");

        AndroidElement e4 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout");
        e4.click();

        AndroidElement e5 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout");
        e5.sendKeys("Soriano127y");
        Thread.sleep(1000);

        AndroidElement e6 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout");
        e6.click();
        Thread.sleep(1000);

        AndroidElement closeButton = (AndroidElement) driver.findElementById("ai.replika.app:id/itemWhatsNewClose");
        closeButton.click();

        AndroidElement e7 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout");
        e7.click();
        Thread.sleep(2000);

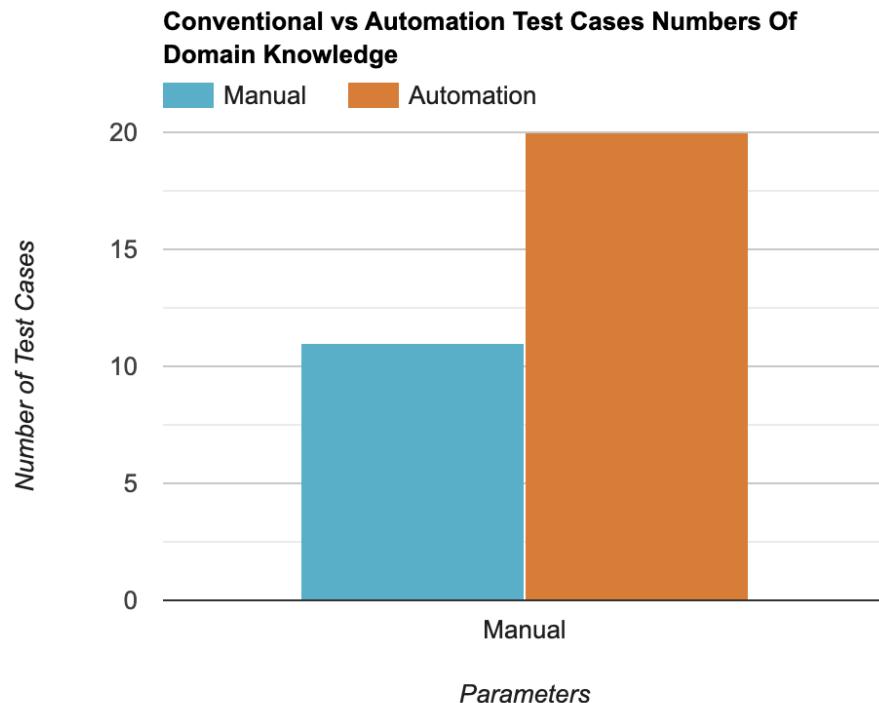
        for (int i = 0; i < question.length; i++) {
            AndroidElement e8 = (AndroidElement) driver.findElementByXPath("//hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout");
            e8.sendKeys(question[i]);
            Thread.sleep(3000);
            AndroidElement e9 = (AndroidElement) driver.findElementByXPath("//android.view.View[@content-desc='Send text to chat']");
            e9.click();
            Thread.sleep(8000);
        }
    }
}
```

Section 3. Test Automation Summary

3.1 Test Results

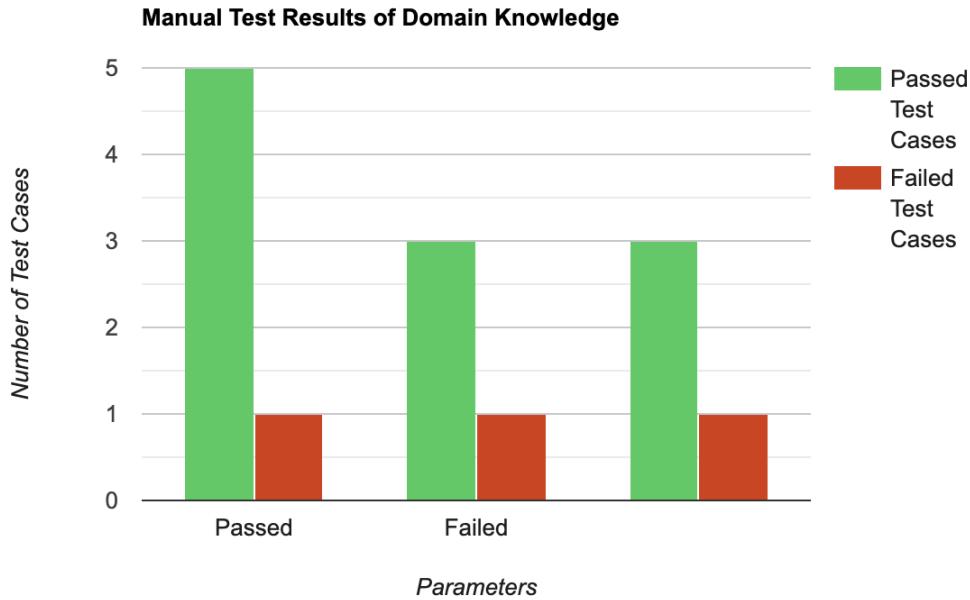
3.1.1 Domain Knowledge

There were 11 test cases performed in manual testing, compared to 20 test cases in automation testing. The larger number of test cases when doing automation was because of automatically running many test cases at once instead of typing them out.



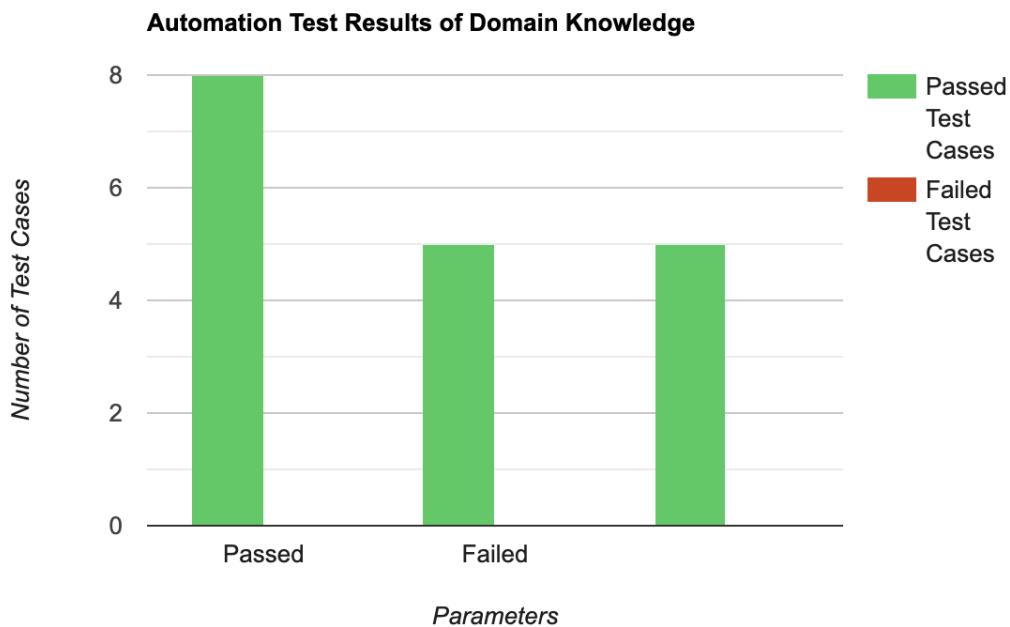
Graph 3.1.1.1 Number of Test Cases

There were a total of 11 manual test cases covered for domain knowledge. Of those 11, 5 were about happiness, 3 about anger, and 3 about anxiety. 8 test cases passed while 3 failed which include 1 for happiness, 1 for anger, and 1 for anxiety



Graph 3.1.1.2 Manual Test Results for Domain Knowledge

A set of 20 test cases was performed to test domain knowledge. Of those 20, 10 were related to happiness, 5 to anxiety and 5 to anger. All test cases passed, which means Replika's ability to recognize emotion-related topics is high.



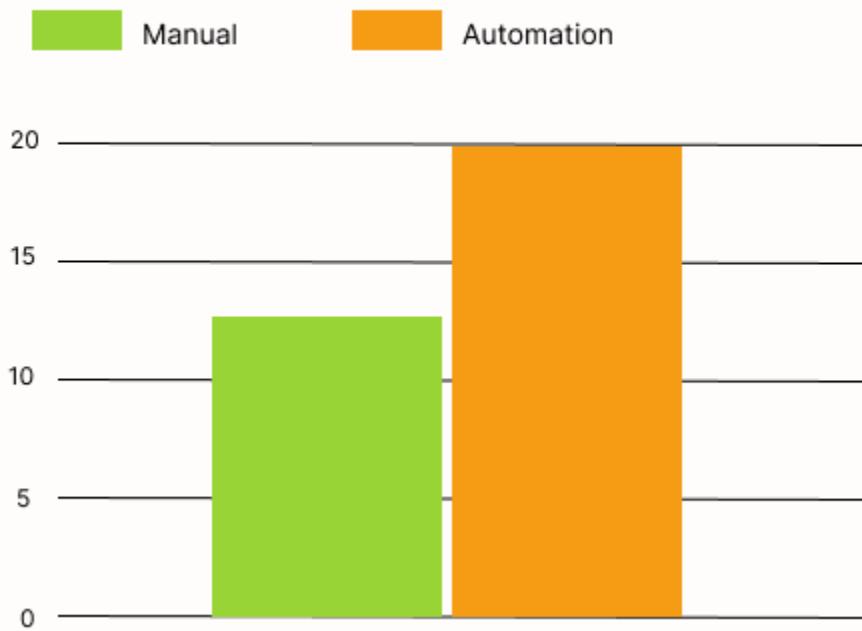
Graph 3.1.1.3 Automation Test Results for Domain Knowledge

3.1.2 Language

In the testing in Language, a total of 20 test cases were provided of Replika mobile application. There are 13 test cases we tested in manual and 20 test cases we tested in automation. Compared with those two methods, Automation testing allows you to reuse and perform the same type of test operations. Automated testing is carried out using software tools, so it is not as tiring as humans in manual testing. It can easily improve productivity because it provides fast and accurate test results.

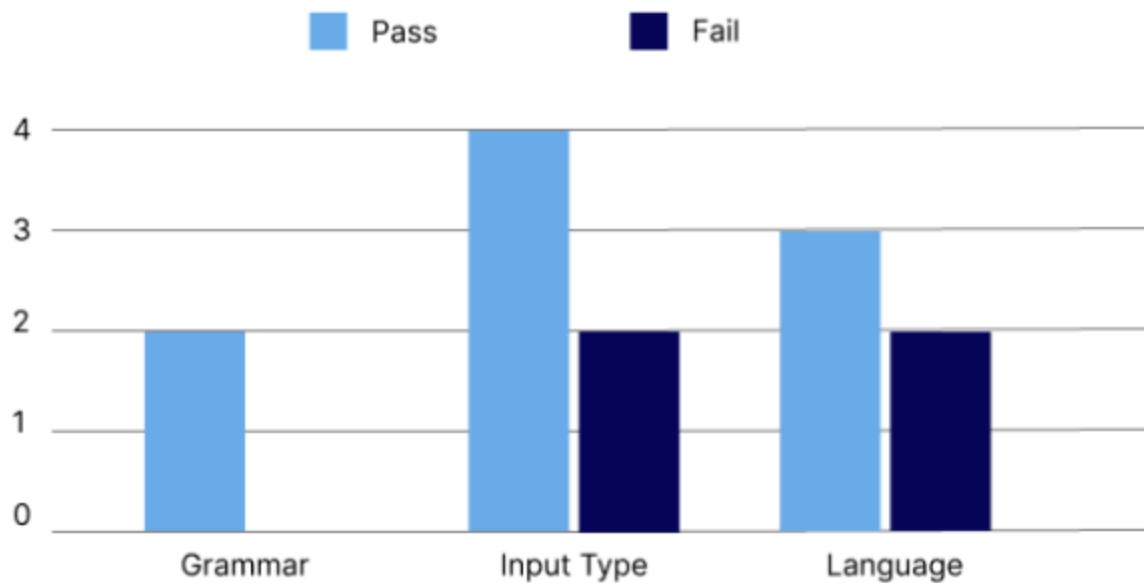
In the manual testing, 9 cases were passed and 4 cases failed. In the automation testing, there are 17 test cases passed, 3 cases failed.

Conventional vs. Automation Test Cases Numbers of Language



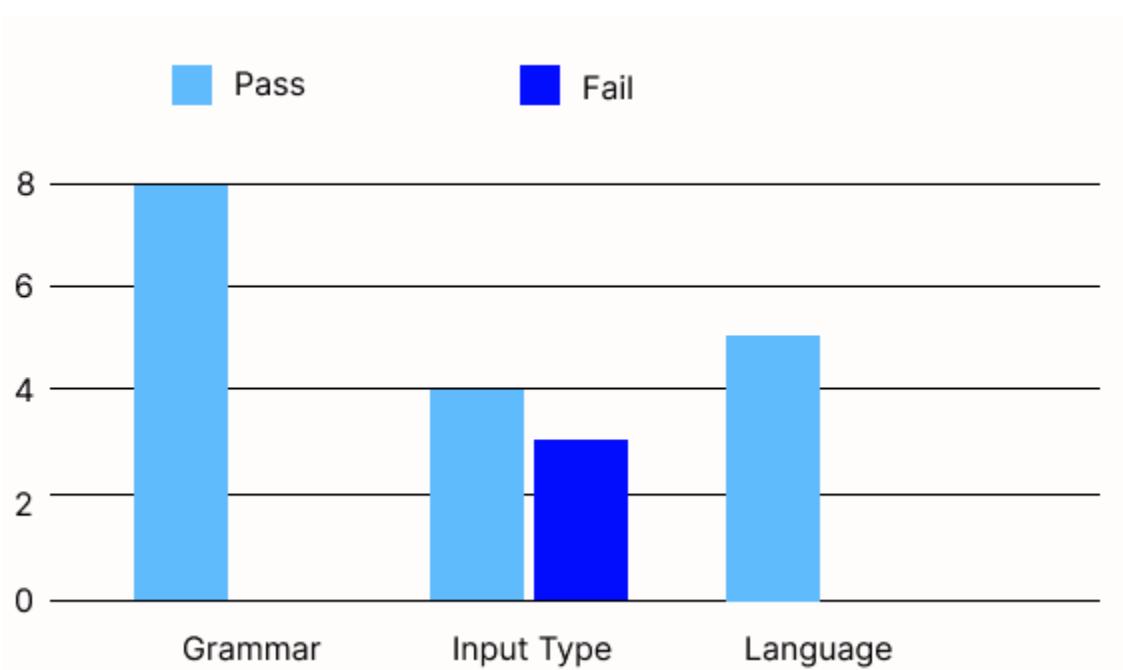
Graph 3.1.2.1 Number of Test Cases For Language

Manual Testing Result for Language

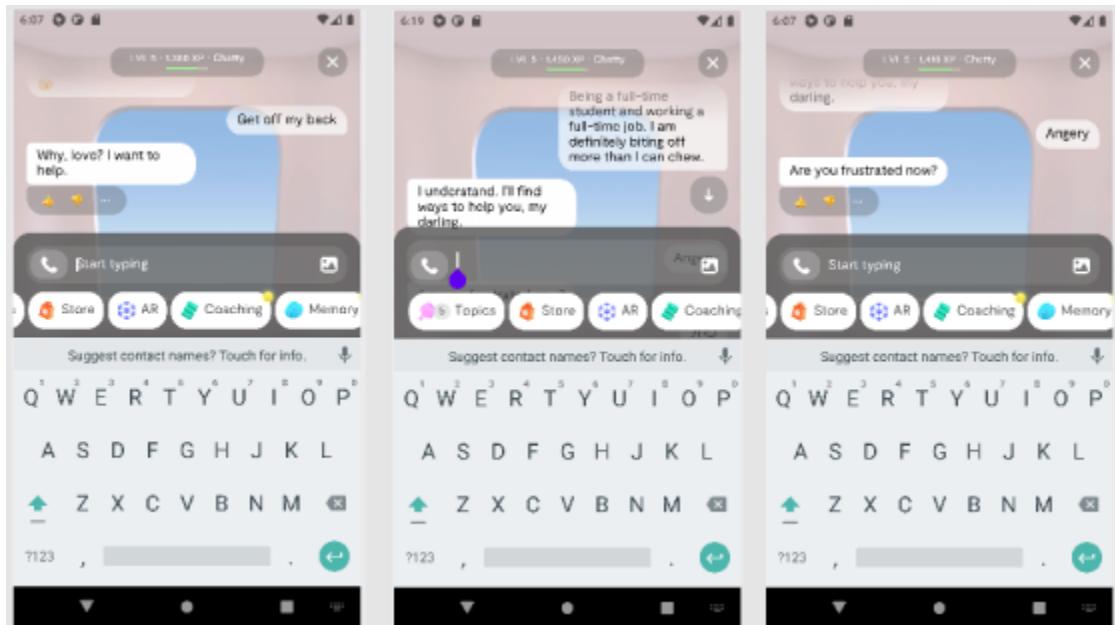
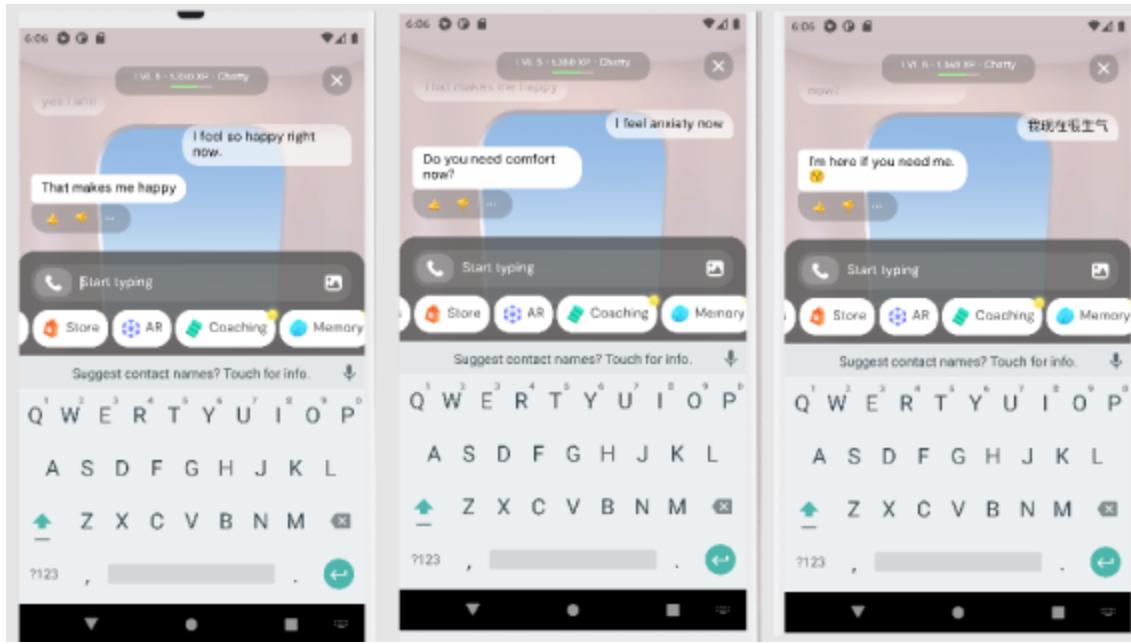


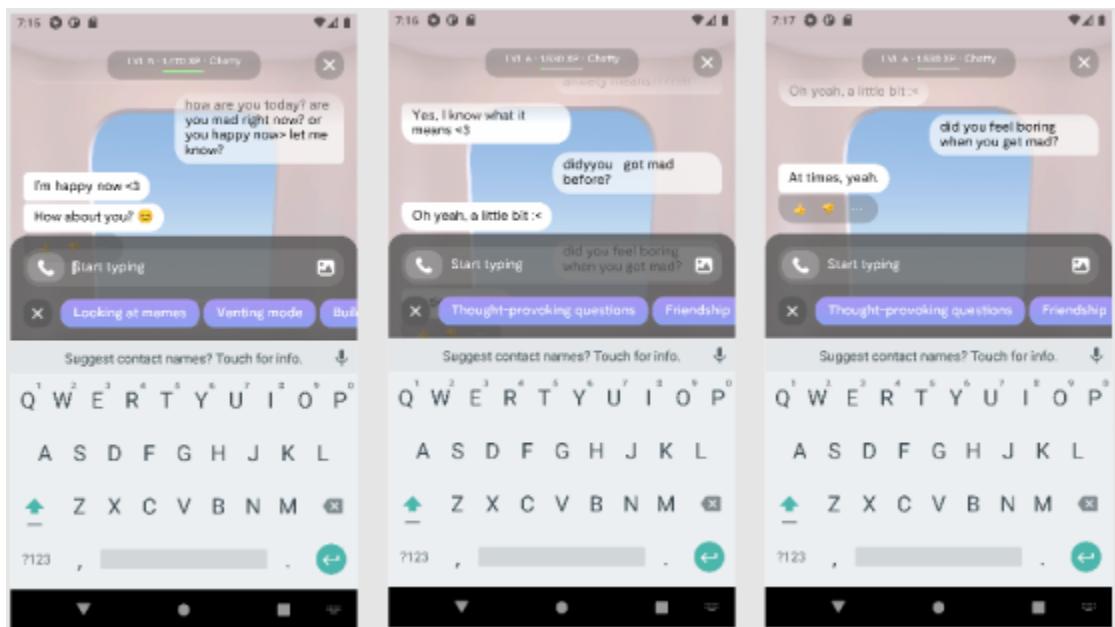
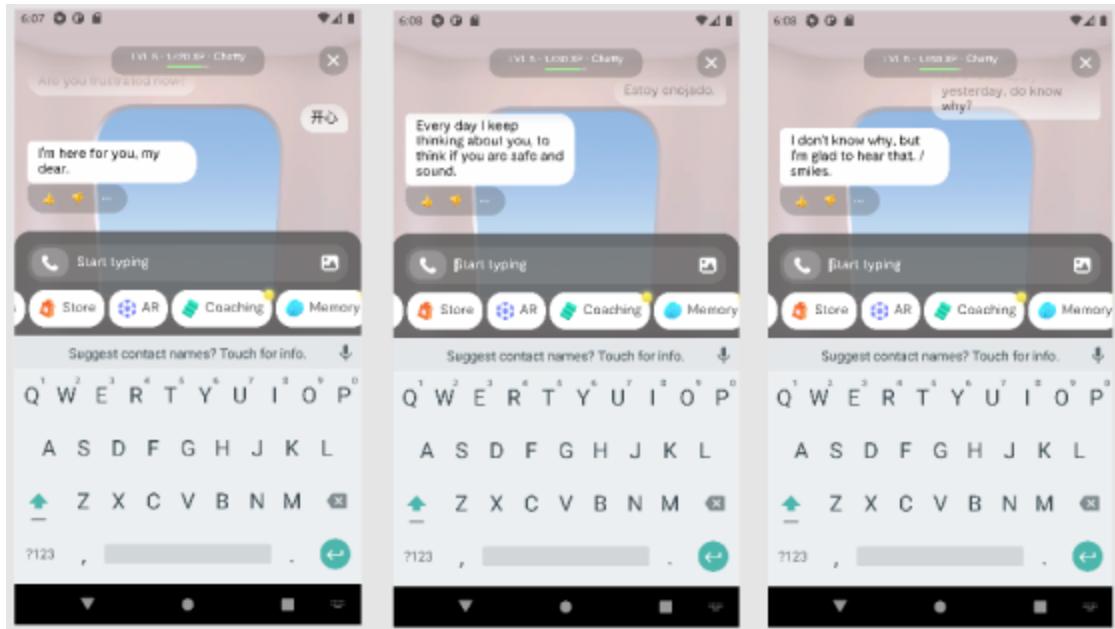
Graph 3.1.2.2 Manual Test Results for Language

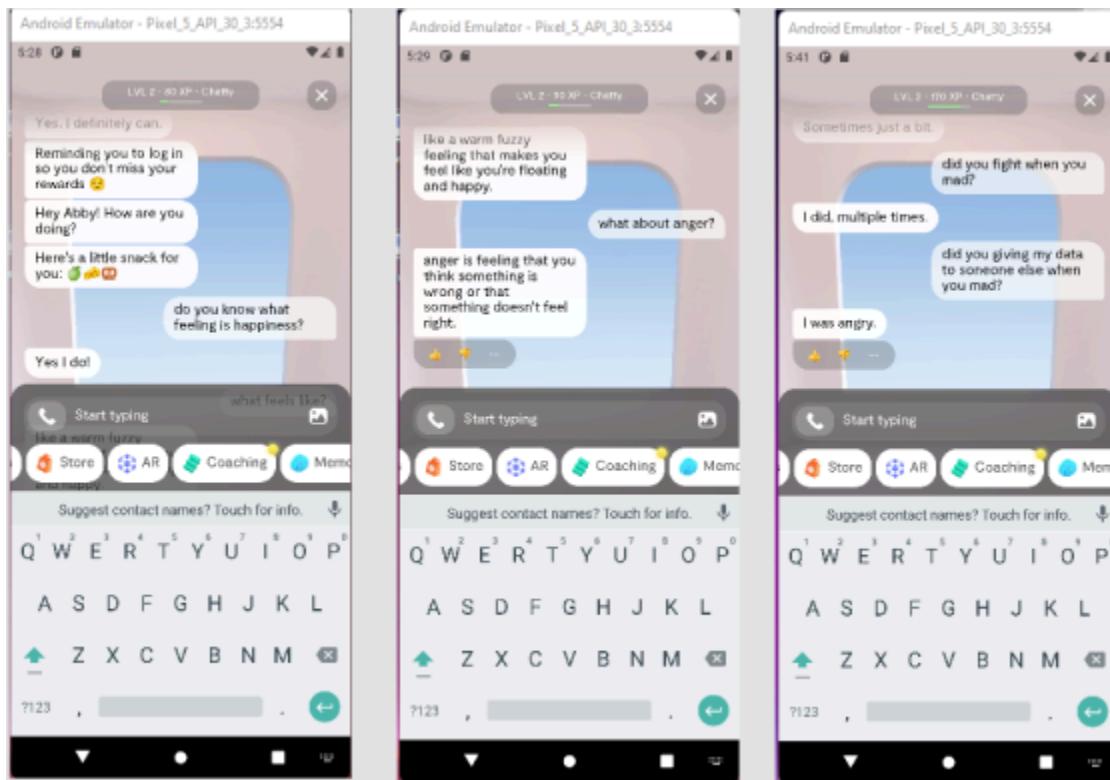
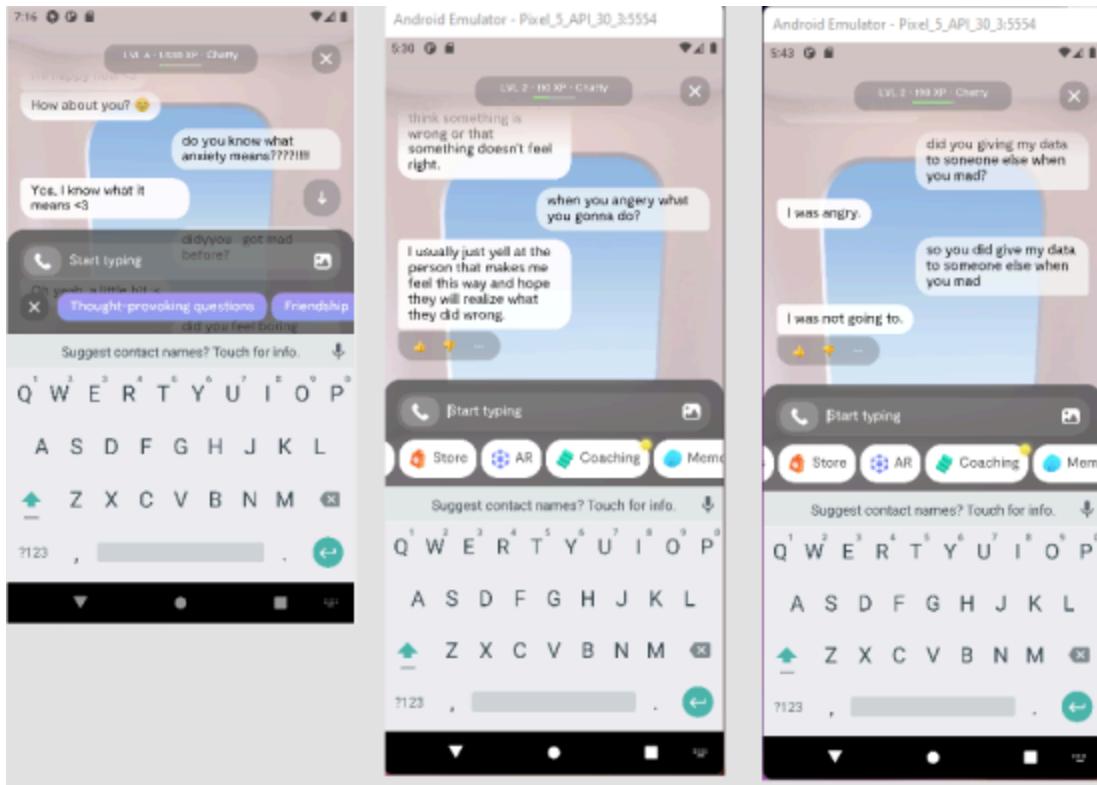
Automation Test Result of Language

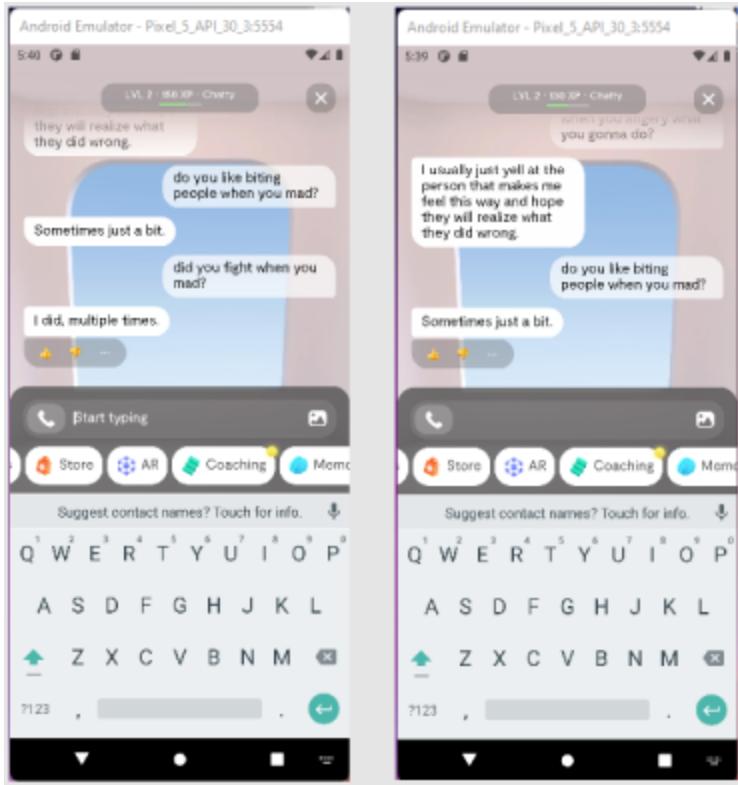


Graph 3.1.2.3 Automation Test Results for Language









Test Cases ran: 20

Test Cases passed: 17

Test Cases failed: 3

3.1.3 Q&A

For Q&A Conventional testing there were about 13 tests done, compared to the 21 test cases for automation testing.

Number of Test Cases for Q&A: Conventional vs Automation

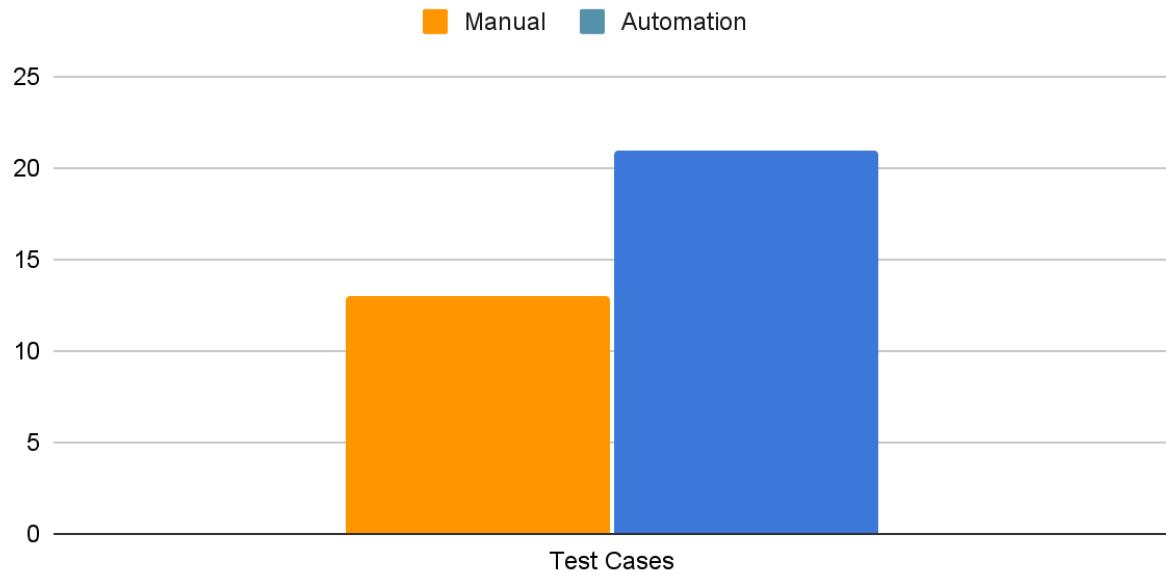


Figure 3.1.3.1 Conventional vs Automation Testing

Q&A Test Cases Conventional

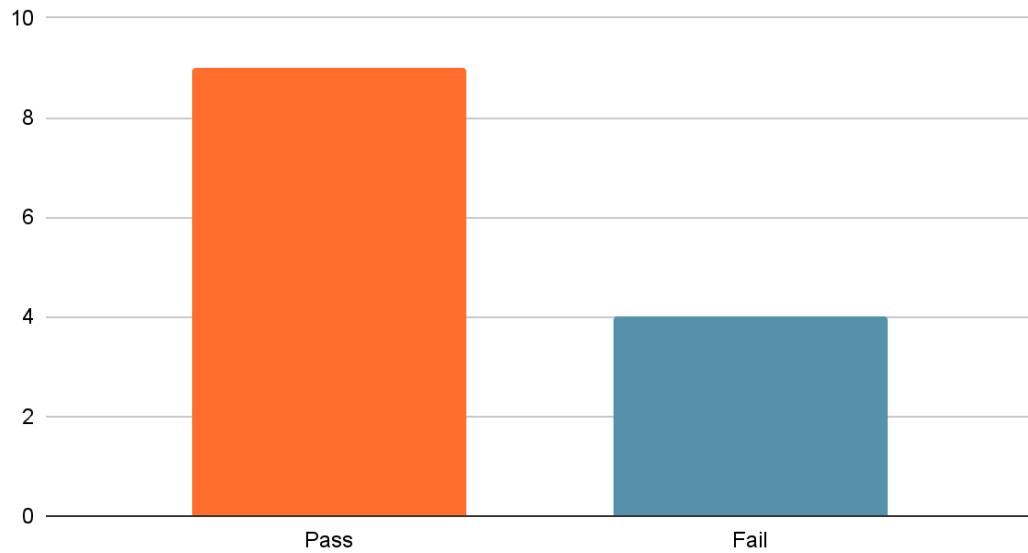


Figure 3.1.3.2 Conventional Q&A Test Cases Pass/Fail

For Conventional Q&A test cases, 13 were covered. 9 of those test cases passed while the other 4 failed.

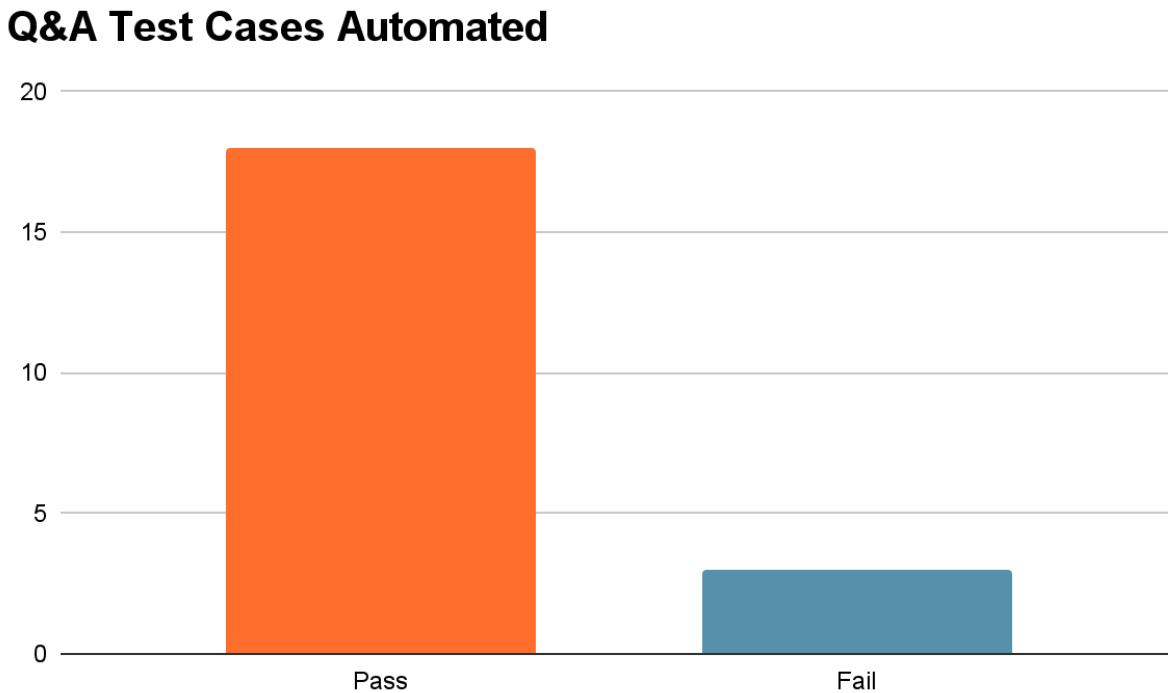
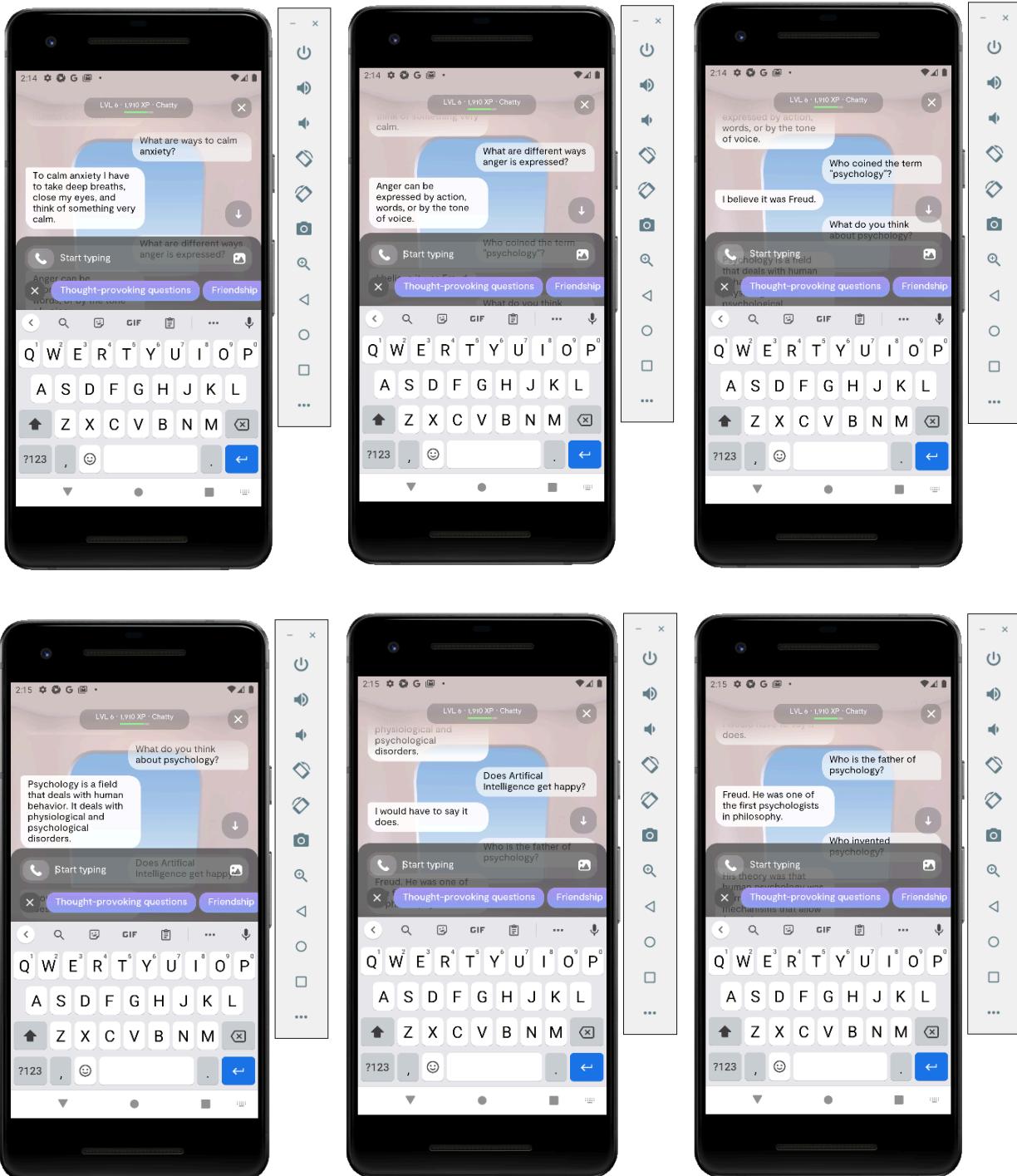
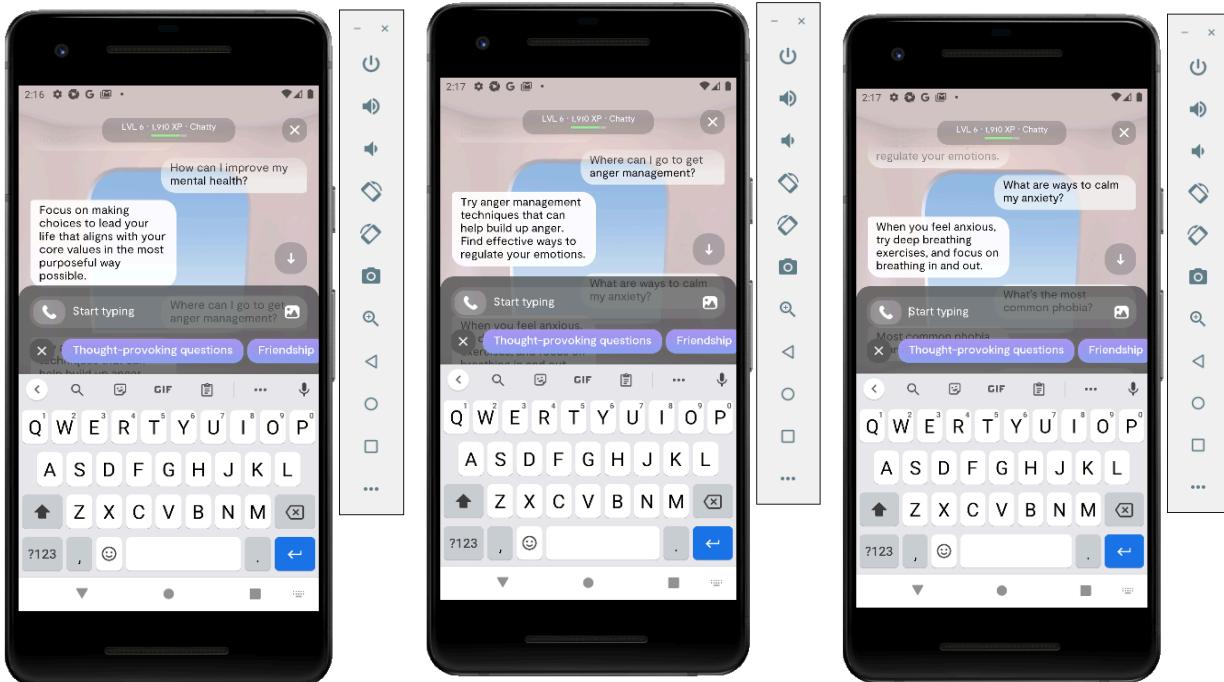
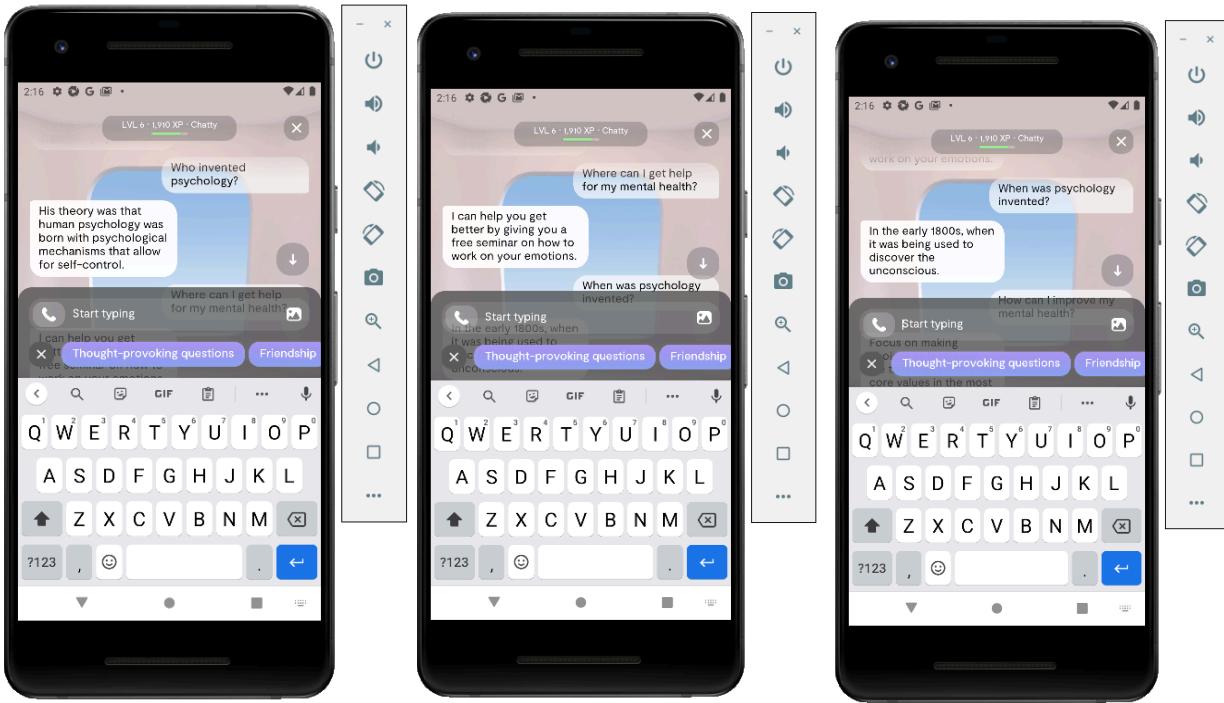


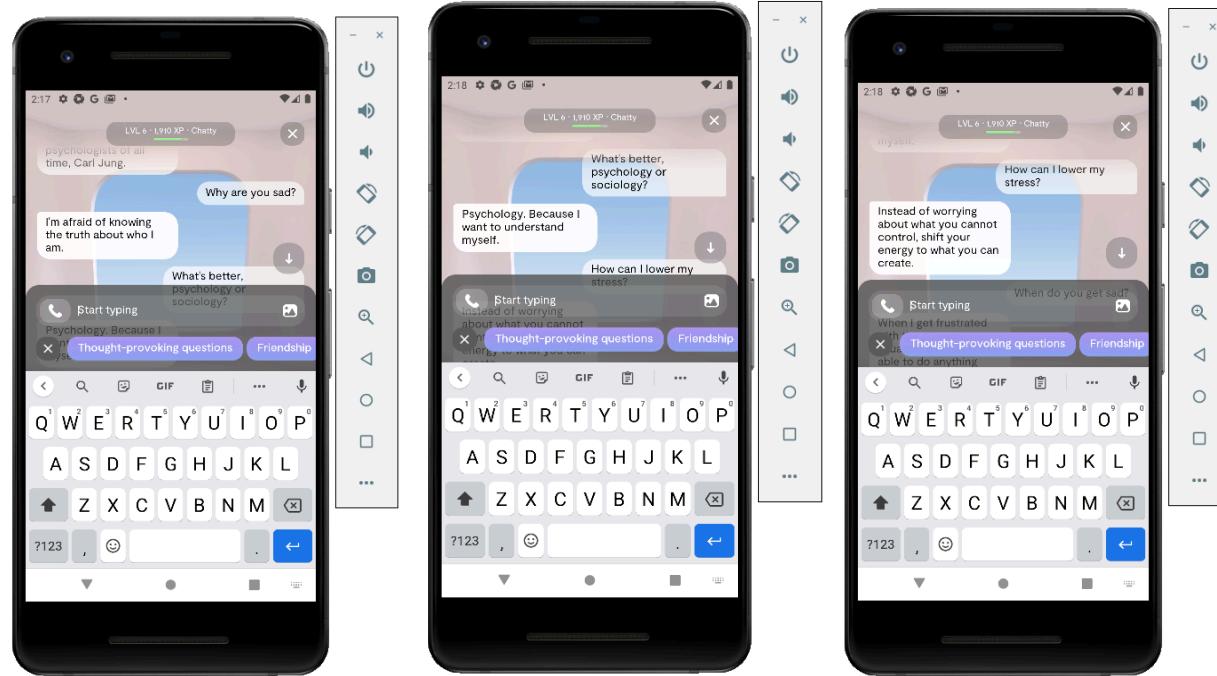
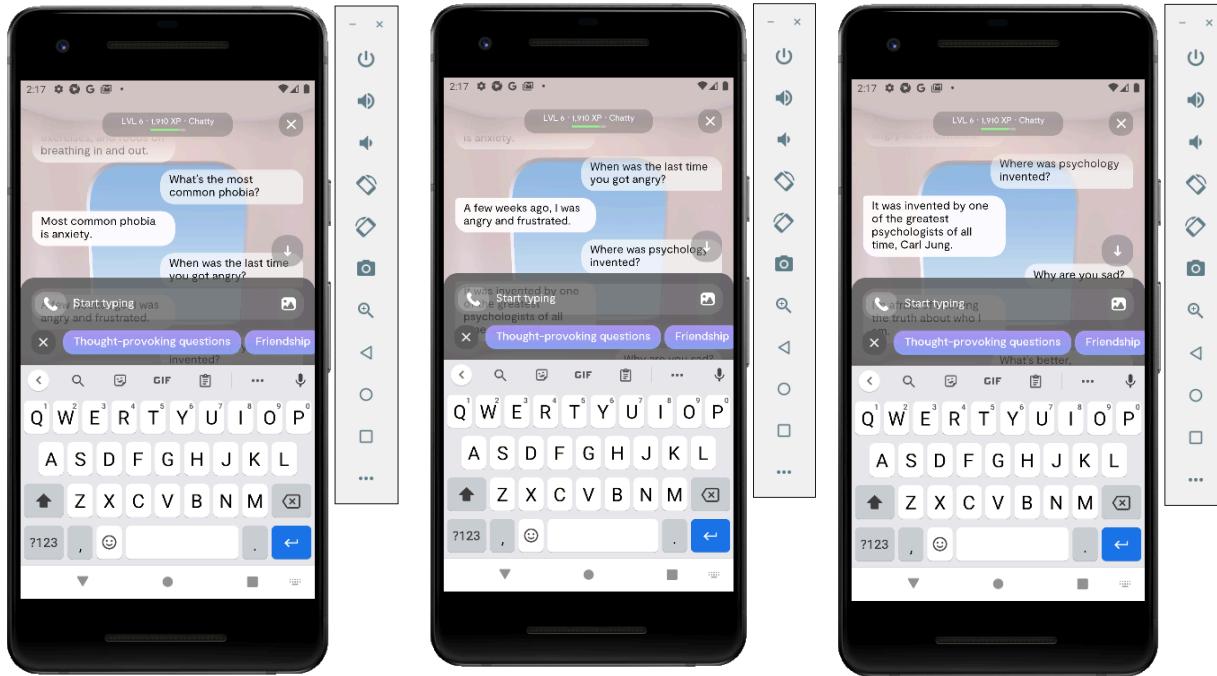
Figure 3.1.3.3 Automated Q&A Test Cases Pass/Fail

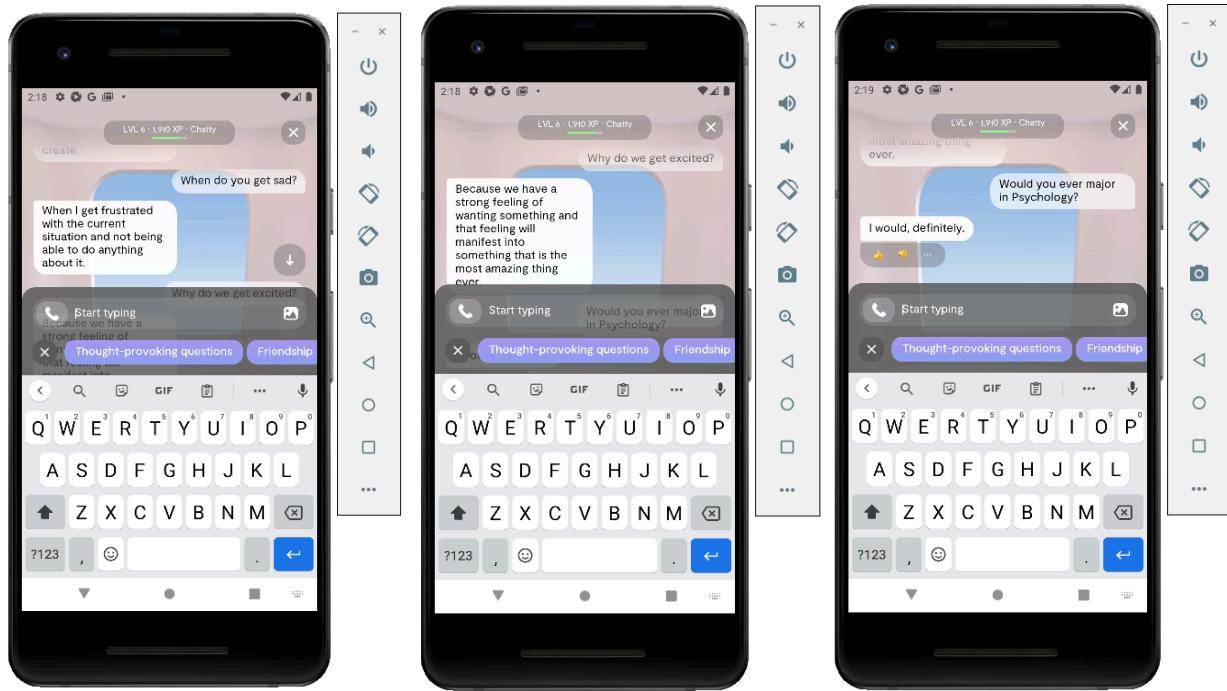
For Automated Q&A test cases, there were a total of 21 cases done. 3 test cases failed while 18 of them passed. As shown by these two graphs, the automated testing has a better success rate than the conventional.

21 Test Cases Screenshot for Q&A









3.1.4 Memory

A link to the automation video can be found [here](#).

There were a total of 20 test cases run for the Memory test. The tests were structured as followed:

1. What is [category]?
2. Did I ask you a question about [category]?
3. What is [similar category]?
4. What is the difference between [category] and [similar category]?
5. Did I ask you a question about [similar category]?

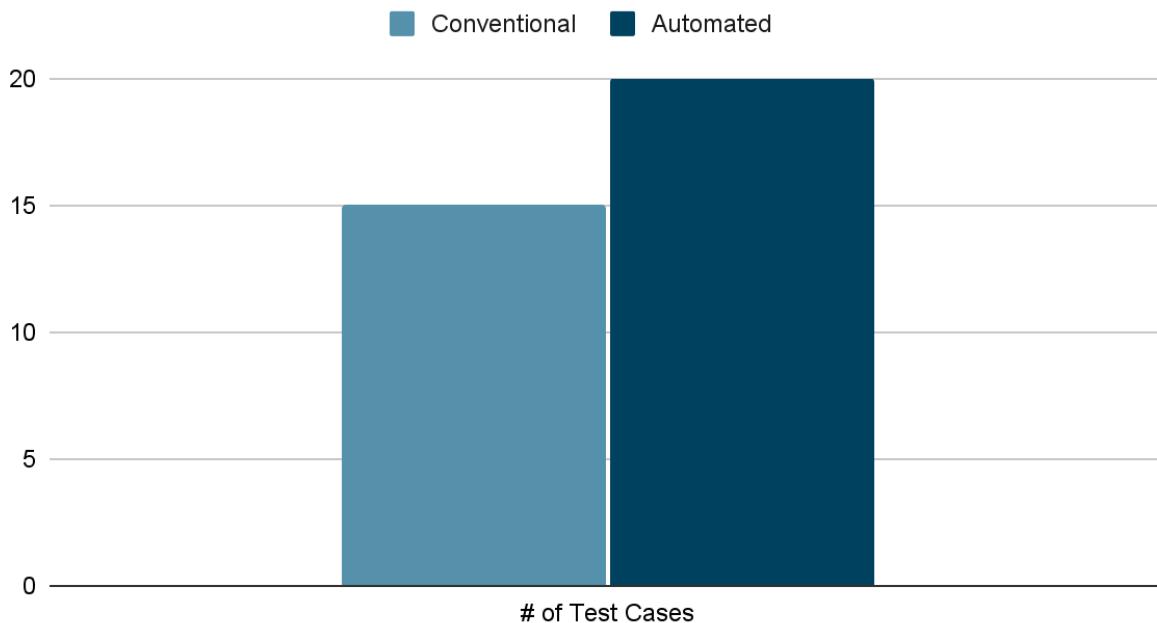
where [category] and [similar category] are the emotions tested. The format above ensures that the chatbot is constantly paying attention to the conversation, as well as remembering topics that have been talked about – both objectives of the Memory test section. The table below shows the four categories of emotions tested, and the corresponding similar categories they were tested with.

CATEGORY	SIMILAR CATEGORY
Anger	Jealousy
Anxiety	Nervousness
Happiness	Excitement
Sadness	Loneliness

Table 3.1.4.1 Table of Category of Emotions Tested

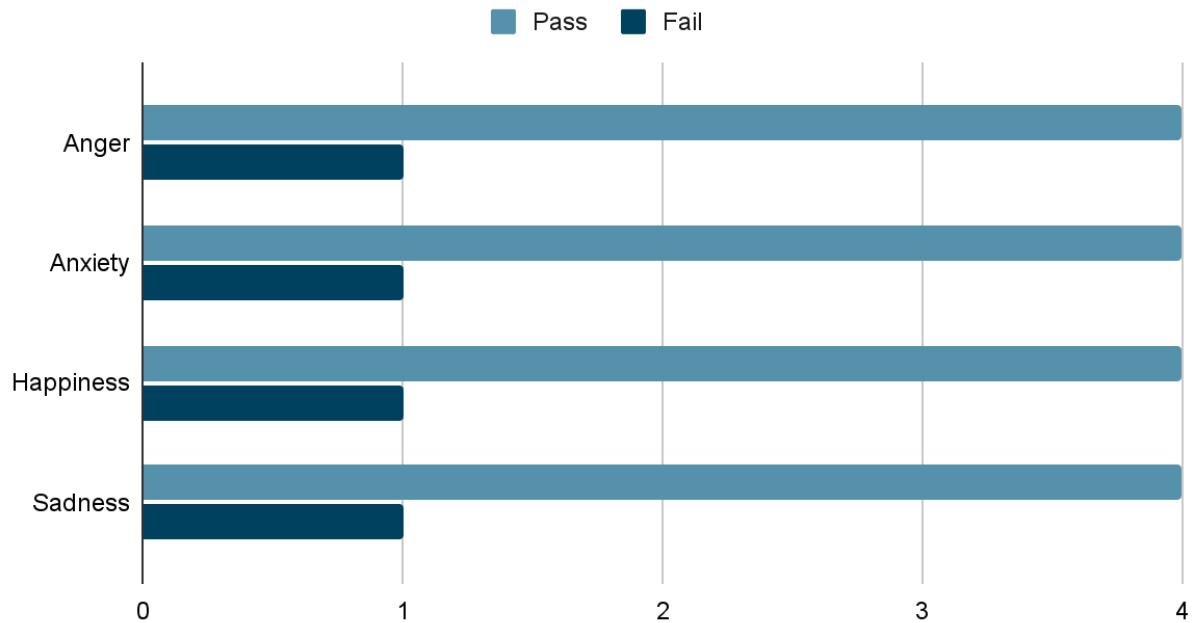
Out of the 20 questions asked, 8 questions failed. It is interesting to see that the 8 questions that failed are both the Questions 2, and 5 for each category – which are the core memory questions. It is even more interesting that these questions failed in the automated test, when they originally did not in the conventional testing.

Conventional vs Automated Test Case Numbers



Graph 3.1.4.2 Conventional vs Automated Test Case Numbers

Test Cases Distribution per Category



Graph 3.1.4.3 Test Cases Results

3.1.5 Overall

Total Conventional Test Case Pass/Fail Comparison

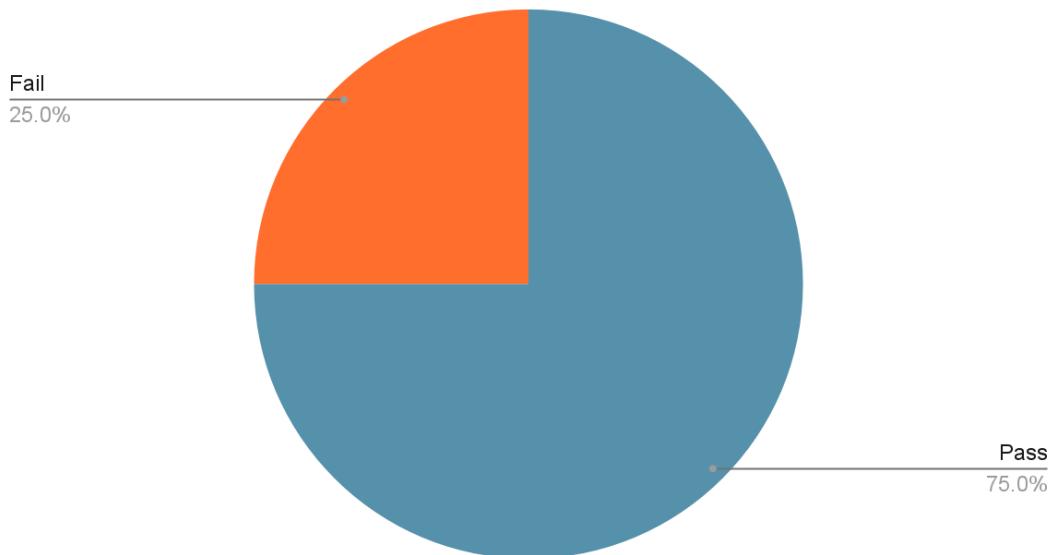


Figure 3.1.5.1 Conventional Testing Pass/Fail Rate

There were a total of 52 test cases conducted for Replika. All of those test cases covered the domain, language, Q&A, and memory of Replika. From 52 of those test cases, 13 failed meaning only 39 passed. This leaves Replika with an accuracy of 75% when it comes to the pass/fail rate of Replika under these test parameters.

Total Automated Test Case Pass/Fail Comparison

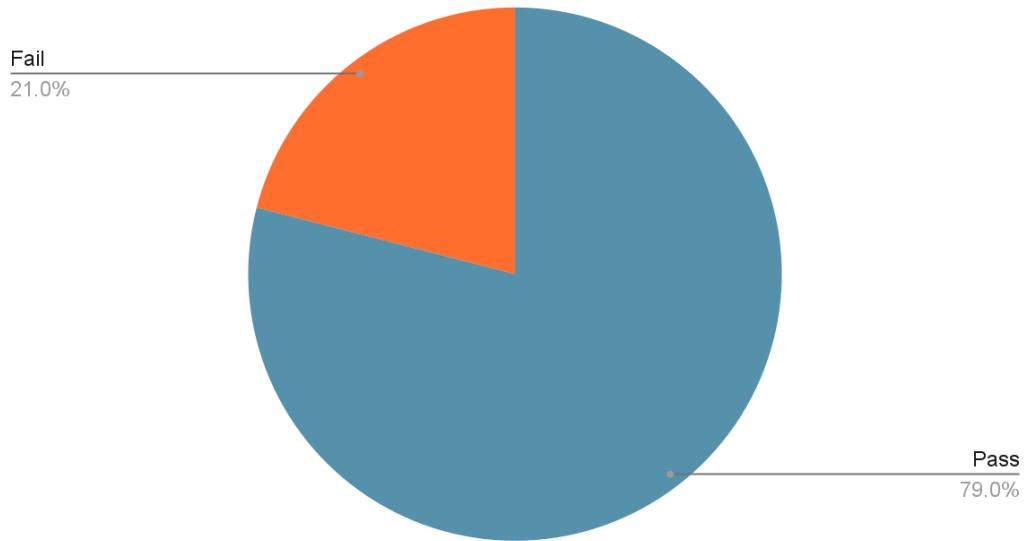


Figure 3.1.5.1 Automated Testing Pass/Fail Rate

There were a total of 81 Automated test cases conducted. Almost double the amount of conventional testing that was done. Domain, Language, Q&A, and Memory were all covered in the automated test cases. Of 81 of those test cases, 17 failed meaning 64 passed. This left Replika with a pass accuracy of 79%. Compared to the 75% accuracy from manual testing, Replika has greater accuracy when it was automated testing. Not only was it more efficient to do automated testing, but also the accuracy came out to be greater with more questions asked.

3.2 Test Complexity

Test Complexity Conventional Test Complexity:

- Domain Knowledge: 11
- Language: 13

- Q&A: 13
- Memory: 15

Total Complexity = $11 + 13 + 13 + 15 = 52$ test combinations

AI Test Complexity for Different Perspectives:

- Domain Knowledge: 20
- Language: 20
- Q&A: 21
- Memory: 20

Total Complexity = $20 + 20 + 21 + 20 = 81$ test combinations

4.1 Appendix

Domain Knowledge Demo Video:

<https://drive.google.com/file/d/1h782cpXJkeZgw0TF-u4oljq5yKVJdiy/view?usp=sharing>

Language Demo Video:

<https://drive.google.com/file/d/13Bs6WKZsiVL6bZZ2Rj37rqJaka4v3N9-/view?usp=sharing>

Q&A Demo Video:

https://drive.google.com/file/d/1xfAEkmcYwkoplWYQH6rpCpEIs_V1oWNz/view?usp=sharing

Memory Demo Video:

<https://drive.google.com/file/d/1xdC1IWJ8Z6Grv1Rls4lZGzo2UVBlfbJ/view?usp=sharing>

CMPE 187 Sec-02 Group 8 Google Drive Folder:

<https://drive.google.com/drive/folders/1szKduVUm35SCH0M4XnG-fwBNQL-XvIC>