AIDERA

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

January 27th, 2017

Test Plan

Contents

1 As a User

1.1 Use Case #1: First-time user interaction with bot

- Test #1 : Functionality Test

  A. Identification and Classification

  - Introduction 001, Welcome Page
  - **Severity**: 1

  B. Instructions

  1. Find and choose Aidera from the list of bots on Facebook.

  C. Expected Result

  1. Aidera gets a callback response, gives user a first-time greeting and displays a "Get Started" button.

- Test #2 : Functionality Test

  A. Identification and Classification

  - Introduction 002, Get Started button
  - **Severity**: 1

  B. Instructions

  1. Click on "Get Started" button.

  C. Expected Result

  1. Aidera responds with a message introducing itself and a set of instructions.

- Test #3 : Equivalence Test

  A. Identification and Classification

  - Introduction 003, Get Started button
  - **Severity**: 1

  B. Instructions

  1. For the class of first-time users, the "Get Started" button is visible and clickable, regardless of the platform that the user is using the application on.

  C. Expected Result

  1. Aidera responds with a message introducing itself and a set of instructions on all platforms.

1.2 Use Case #2: Bot exchanges pleasantries with user

- Test #1 : Functionality Test

  A. Identification and Classification

  - Interaction 001, Greeting
  - **Severity**: 3

  B. Instructions

  1. Open Aidera on Messenger.
  2. Type in a greeting.

  C. Expected Result

  1. Aidera responds with an appropriate greeting message.

- Test #2 : Boundary Test

  A. Identification and Classification

  - User Alert 001, Edge Case of Greeting
  - **Severity**: 3

  B. Instructions

  1. Open Aidera on Messenger.
  2. Type in a non-greeting message.

  C. Expected Result

  1. Aidera responds with standardized error message to the users.

1.3 Use Case #3: Bot interacts with user

- Test #1 : Functionality Test

  A. Identification and Classification

  - Interaction 003, Conversation
  - **Severity**: 1

  B. Instructions

  1. Send a message to the bot.
  2. Request information from the bot.

  C. Expected Result

  1. Aidera must give the most suitable response to the user's inquiry.

- Test #2 : Equivalence Test

  A. Identification and Classification

  - Interaction 004, Conversation
  - **Severity**: 1

  B. Instructions

  1. Send a message to the bot.

2. User requests information regarding Yelp services from the bot.
3. User requests information regarding Airbnb services from the bot.

C. Expected Result

1. Aidera must give the most suitable response to the user's inquiry using the appropriate service.

- TEST #3 : BOUNDARY TEST

A. Identification and Classification

- User Alert 002, Edge Case of Conversation
- **Severity**: 1

B. Instructions

1. Send an irrelevant message to the bot.
2. Request irrelevant information from the bot.

C. Expected Result

1. Aidera must handle message appropriately and send an error response to the user including possible functionalities.

1.4 Use Case #4: Bot repeats instructions when user asks for help

- TEST #1 : FUNCTIONALITY TEST

A. Identification and Classification

- Help 001, Helper
- **Severity**: 2

B. Instructions

1. Type in "help" on the Aidera chat window.

C. Expected Result

1. Aidera gets the request and sends back standardized helpful information (functionalities of APIs and how to use the bot).

- TEST #2 : EQUIVALENCE TEST

A. Identification and Classification

- Help 002, Helper
- **Severity**: 2

B. Instructions

1. The bot will understand specific set of messages like "help", "need help" etc. on the Aidera chat window.

C. Expected Result

1. Aidera understands the request and sends back standardized helpful information (functionalities of APIs and how to use the bot).

- TEST #3 : BOUNDARY TEST

A. Identification and Classification

- User Alert 003, Edge Case of Helper
- **Severity**: 2

B. Instructions

1. Type in non-descriptive help statement.

C. Expected Result

1. Aidera responds with standardized error message to the users.

1.5 Use Case #5: Facilitate a forum-like environment on Aidera Facebook page

- Test #1 : Equivalence Test

A. Identification and Classification

- Help 002, Resource Page
- **Severity**: 3

B. Instructions

1. Discuss bot feedback with other users.

C. Expected Result

1. Aidera Facebook page information is available to all users. Developers provide responses to problematic situations.

1.6 Use Case #6: Login to Facebook to use Aidera

- Test #1 : Functionality Test

A. Identification and Classification

- Login 001, Usage
- **Severity**: 1

B. Instructions

1. Facebook users search for “Aidera” in Messenger text window.

C. Expected Result

1. The user is able to access Aidera’s chat window.

1.7 Use Case #7: Bot is cross-platform

- Test #1 : Functionality Test

A. Identification and Classification

- Platform 001, Cross Platform
- **Severity**: 1

B. Instructions

1. Interact with Aidera on mobile and seamlessly switch to desktop.

C. Expected Result

1. Aidera works and responds to user's questions.

- Test #2 : Equivalence Test

A. Identification and Classification

- Platform 002, Cross Platform
- **Severity**: 1

B. Instructions

1. Open Aidera bot on any mobile Operating System.
2. Open Aidera bot on any desktop Operating System.

C. Expected Result

1. Aidera works and responds to user's questions.

1.8 Use Case #8: Bot always acts in character

- Test #1 : Functionality Test

A. Identification and Classification

- Context 001, Understanding Message Intent
- **Severity**: 2

B. Instructions

1. Send a greeting message.
2. Send a message requesting for nearby restaurants.

C. Expected Result

1. Aidera should understand the intent of each message and reply accordingly.

- Test #2 : Equivalence Test

A. Identification and Classification

- Context 002, Understanding Message Intent
- **Severity**: 2

B. Instructions

1. Send a message like, "Show me Chinese restaurants nearby".

C. Expected Result

1. Aidera should understand what is being requested and should respond appropriately with the Chinese restaurants nearby.

- Test #3 : Boundary Test

A. Identification and Classification

- Context 003, Understanding Message Intent
- **Severity**: 2

B. Instructions

1. Type in general questions like, “What do you think of Siri?”.

C. Expected Result

1. Aidera responds appropriately by having a standard error reply to irrelevant texts.

1.9 Use Case #9: Settings and log of all messages across the conversation

- Test #1 : Functionality Test

A. Identification and Classification

- Functionality 001, History of messages
- **Severity**: 1

B. Instructions

1. Click on button to view previous messages.

C. Expected Result

1. Aidera shows the previous texts between user and the bot.

- Test #2 : Equivalence Test

A. Identification and Classification

- Functionality 002, Settings
- **Severity**: 3

B. Instructions

1. Change settings of chat bot like color, emoticons etc.

C. Expected Result

1. Aidera changes color, emoticons etc. of user messages with the bot.

1.10 Use Case #10: First-time user tutorial on how the bot works

- Test #1 : Functionality Test

A. Identification and Classification

- Help 003, Tutorial
- **Severity**: 2

B. Instructions

1. First-time user clicks on tutorial button.

C. Expected Result

1. Aidera provides tutorial details to help user get better acquainted with our chat bot.

- Test #2 : Boundary Test

A. Identification and Classification

- Help 004, Edge Case of Tutorial
- **Severity**: 2

B. Instructions

1. Returning user starts interacting with the bot.

C. Expected Result

1. Aidera won't provide usage instructions. It will instead start interaction with the user.

1.11 Use Case #11: Bot provides quick reply

- Test #1 : Functionality Test

A. Identification and Classification

- Performance 001, Speed of Response
- **Severity**: 3

B. Instructions

1. Request for information from bot.

C. Expected Result

1. Aidera immediately provides an appropriate response to user's text.

- Test #2 : Equivalence Test

A. Identification and Classification

- Performance 002, Speed of Response
- **Severity**: 2

B. Instructions

1. Any type of user (first-time or returning) interacts with the bot.

C. Expected Result

1. Aidera immediately provides an appropriate response to user's text to every user in a timely fashion.

1.12 Use Case #12: Server downtime alerts

- Test #1 : Functionality Test

A. Identification and Classification

- User Alert 004, Server downtime
- **Severity**: 2

B. Instructions

1. Text the bot when the server is down.

C. Expected Result

1. Before server is down, Aidera sends automated text informing all users about the server downtime.

- Test #2 : Boundary Test

  A. Identification and Classification

  - User Alert 005, Server up
  - **Severity**: 2

  B. Instructions

  1. Text the bot when the server is up.

  C. Expected Result

  1. Once server is running again, Aidera sends automated text informing all users about the server.

1.13 Use Case #13: Bot provides cuisine of the month

- Test #1 : Functionality Test

  A. Identification and Classification

  - Functionality 003, Cuisine of the month
  - **Severity**: 2

  B. Instructions

  1. Request recommended cuisine of the month.

  C. Expected Result

  1. Aidera responds with the recommended cuisine of the month.

1.14 Use Case #14: User sends request for cuisine based restaurant search

- Test #1 : Functionality Test

  A. Identification and Classification

  - Yelp Functionality 001, Cuisine based search
  - **Severity**: 2

  B. Instructions

  1. Request for restaurants based on cusine.

  C. Expected Result

  1. Aidera uses Yelp API to retrieve restaurants that serve that specific cuisine and responds appropriately.

- Test #2 : Boundary Test

  A. Identification and Classification

  - Yelp Functionality 002, Edge Case of Cuisine based search

- **Severity**: 2

B. Instructions

1. Request for restaurants based on cuisine that doesn't exist.

C. Expected Result

1. Aidera uses Yelp API to retrieve restaurants that serve that specific cuisine and returns an error message if no restaurants serving that particular cuisine exist.

1.15 Use Case #15: User sends request for a restaurant near the user

- Test #1 : Functionality Test

A. Identification and Classification

- Yelp Functionality 002, Restaurants near user
- **Severity**: 2

B. Instructions

1. Request for a restaurant based on vicinity.

C. Expected Result

1. Aidera uses Yelp API to retrieve restaurants near the user's current location and responds appropriately.

1.16 Use Case #16: User sends request for public ratings and reviews of a restaurant

- Test #1 : Functionality Test

A. Identification and Classification

- Yelp Functionality 003, Restaurant rating request
- **Severity**: 2

B. Instructions

1. Request for public rating of a restaurant.

C. Expected Result

1. Aidera uses Yelp API to retrieve public ratings of a restaurant and responds appropriately.

- Test #2 : Functionality Test

A. Identification and Classification

- Yelp Functionality 004, Restaurant reviews request
- **Severity**: 2

B. Instructions

1. Request for reviews of a restaurant.

C. Expected Result

1. Aidera uses Yelp API to retrieve reviews of a restaurant and responds appropriately.

- TEST #3 : BOUNDARY TEST

  A. Identification and Classification

  - Yelp Functionality 005, Edge Case of Restaurant reviews request
  - **Severity**: 2

  B. Instructions

  1. Request for reviews or a public rating of a restaurant that does not exist.

  C. Expected Result

  1. Aidera uses Yelp API to retrieve restaurants in that location and returns an error message if no restaurants near the user exists.

1.17 Use Case #17: User sends requests for price based restaurant search

- TEST #1 : EQUIVALENCE TEST

  A. Identification and Classification

  - Yelp Functionality 005, Price based search
  - **Severity**: 2

  B. Instructions

  1. Request for restaurants based on price.

  C. Expected Result

  1. Aidera leverages Yelp API to retrieve restaurants within the price range and responds appropriately.

- TEST #2 : BOUNDARY TEST

  A. Identification and Classification

  - User Alert 006, Edge case of Price Based Search
  - **Severity**: 2

  B. Instructions

  1. Request for restaurants based on price that are invalid numbers (e.g price <= $0).

  C. Expected Result

  1. Aidera responds with standardized error message to the users.

1.18 Use Case #18: User sends request for PDF of menu card of specific restaurant

- TEST #1 : FUNCTIONALITY TEST

A. Identification and Classification

- Yelp Functionality 006, Request for menu card
- **Severity**: 2

B. Instructions

1. Choose postback button to request PDF of menu card of a particular restaurant.

C. Expected Result

1. Aidera leverages Yelp API to retrieve PDF of menu card of specific restaurant and responds appropriately.

- TEST #2 : BOUNDARY TEST

A. Identification and Classification

- User Alert 007, Edge Case of Request for menu card
- **Severity**: 2

B. Instructions

1. Choose postback button to request PDF of menu card of a particular restaurant that doesn't have a menu.

C. Expected Result

1. Aidera responds with standardized error message to the users.

1.19 Use Case #19: User sends request for business information of a restaurant

- TEST #1 : FUNCTIONALITY TEST

A. Identification and Classification

- Yelp Functionality 007, Request for Business Information
- **Severity**: 3

B. Instructions

1. Choose postback button to request business information of a restaurant.

C. Expected Result

1. Aidera uses Yelp API to retrieve business information (contact number, email etc.) and responds appropriately.

- TEST #2 : BOUNDARY TEST

A. Identification and Classification

- User Alert 008, Edge Case of Request for Business Information
- **Severity**: 3

B. Instructions

1. Choose postback button to request business information of a restaurant that doesn't have it's details on Yelp.

C. Expected Result

1. Aidera responds with standardized error message to the users.

1.20 Use Case #20: User sends request for date based listing search

- Test #1 : Equivalence Test

  A. Identification and Classification

  - Airbnb Functionality 001, Date based search
  - **Severity**: 2

  B. Instructions

  1. Request for accommodations based on dates.

  C. Expected Result

  1. Aidera leverages Airbnb API to retrieve accommodations based on dates and responds appropriately.

- Test #2 : Boundary Test

  A. Identification and Classification

  - User Alert 009, Edge Case of Date based search
  - **Severity**: 2

  B. Instructions

  1. Request for accommodations based on dates that are invalid on Airbnb.

  C. Expected Result

  1. Aidera responds with standardized error message to the users.

1.21 Use Case #21: User sends request for public ratings and reviews for a particular listing

- Test #1 : Functionality Test

  A. Identification and Classification

  - Airbnb Functionality 002, Request for public ratings
  - **Severity**: 2

  B. Instructions

  1. Request for listing based on public ratings.

  C. Expected Result

  1. Aidera uses Airbnb API to retrieve listing based on public ratings and responds appropriately.

- Test #2 : Functionality Test

  A. Identification and Classification

  - Airbnb Functionality 003, Request for reviews
  - **Severity**: 2

  B. Instructions

  1. Request for listing based on reviews

C. Expected Result

1. Aidera uses Airbnb API to retrieve listing based on reviews and responds appropriately.

- Test #3 : Boundary Test

A. Identification and Classification

- User Alert 010, Edge Case of Request for public rating
- **Severity**: 2

B. Instructions

1. Ask the bot for a public rating of an accommodation that is invalid on Airbnb.

C. Expected Result

1. Aidera responds with standardized error message to the users.

- Test #4 : Boundary Test

A. Identification and Classification

- User Alert 011, Edge Case of Request for reviews
- **Severity**: 2

B. Instructions

1. Ask the bot for a review of an accommodation that is invalid on Airbnb.

C. Expected Result

1. Aidera responds with standardized error message to the users.

1.22 Use Case #22: User sends request for location based listing search

- Test #1 : Functionality Test

A. Identification and Classification

- Airbnb Functionality 004, Location based search
- **Severity**: 2

B. Instructions

1. Request for accommodations based on locations.

C. Expected Result

1. Aidera uses Airbnb API to retrieve listing based on locations and responds appropriately.

- Test #2 : Boundary Test

A. Identification and Classification

- User Alert 012, Edge Case of Location based search
- **Severity**: 2

B. Instructions

1. Request for accommodations based on locations that are invalid on Airbnb.

C. Expected Result

1. Aidera responds with standardized error message to the users.

1.23 Use Case #23: User sends request for host based listing search

- Test #1 : Functionality Test

  A. Identification and Classification

  - Airbnb Functionality 005, Host based search
  - **Severity**: 2

  B. Instructions

  1. Request for accommodations based on hosts.

  C. Expected Result

  1. Aidera leverages Airbnb API to retrieve listing for a specific host and responds appropriately.

- Test #2 : Boundary Test

  A. Identification and Classification

  - User Alert 013, Edge Case of Host based search
  - **Severity**: 2

  B. Instructions

  1. Request for accommodations based on hosts that are invalid on Airbnb.

  C. Expected Result

  1. Aidera responds with standardized error message to the users.

1.24 Use Case #24: User sends request for specific listing

- Test #1 : Functionality Test

  A. Identification and Classification

  - Airbnb Functionality 006, Request for specific listing
  - **Severity**: 2

  B. Instructions

  1. Click the postback button to get more information on a particular listing.

  C. Expected Result

  1. Aidera uses Airbnb API to retrieve listing based on a specific name and responds appropriately.

- Test #2 : Boundary Test

  A. Identification and Classification

  - User Alert 014, Edge Case of Request for Specific Listing
  - **Severity**: 2

  B. Instructions

  1. Click the postback button to get more information on a particular listing that is invalid on Airbnb.

  C. Expected Result

  1. Aidera responds with standardized error message to the users.

2 As a Developer

2.1 Use Case #1: Bot allows bot-to-bot interactions

- Test #1 : Functionality Test

  A. Identification and Classification

  - B2B 001, Bot Interaction
  - **Severity**: 3

  B. Instructions

  1. User types in a request for more information of a particular service.

  C. Expected Result

  1. Aidera connects to a chatbot of the service and gets more information if possible.

- Test #2 : Boundary Test

  A. Identification and Classification

  - Developer Alert 001, Edge Case of Bot Interaction
  - **Severity**: 3

  B. Instructions

  1. User types in a request for more information of a particular service that doesn't exist.

  C. Expected Result

  1. Aidera tries to connect to a chatbot of the service and gets more information if possible.

2.2 Use Case #2: Regular updates to the chatbot

- Test #1 : Functionality Test

  A. Identification and Classification

  - Update 001, Update Bot
  - **Severity**: 2

  B. Instructions

  1. User types in feedback of the bot for the developers to improve the functionality and add new features.

  C. Expected Result

  1. Aidera stores all ratings and reviews in our database.
  2. Developers can used suggested ratings and reviews to add new features and functionality.

- Test #2 : Boundary Test

  A. Identification and Classification

  - Developer Alert 002, Edge Case of Update Bot

  - **Severity**: 2

  B. **Instructions**

  1. User types in inappropriate/irrelevant response.

  C. **Expected Result**

  1. Aidera automatically discards reviews that don't provide any criticism and doesn't store that review.

2.3 Use Case #3: Bot should be scalable

- Test #1 : Functionality Test

  A. **Identification and Classification**

  - Scalable 001, Scalability Check
  - **Severity**: 3

  B. **Instructions**

  1. User types in a request for any service provided and wait for a reply.

  C. **Expected Result**

  1. Aidera should quickly dispatch requests and get replies even when the number of users is high.

- Test #2 : Equivalence Test

  A. **Identification and Classification**

  - Scalable 002, Scalability Check
  - **Severity**: 3

  B. **Instructions**

  1. User types in a request for any service provided and wait for a reply.

  C. **Expected Result**

  1. Aidera should be able to classify the user class (for instance, whether the user is asking for Yelp or Airbnb service) correctly on scale and respond appropriately.

2.4 Use Case #4: Reduce downtime for maintenance

- Test #1 : Functionality Test

  A. **Identification and Classification**

  - Downtime 001, Maintenance
  - **Severity**: 3

  B. **Instructions**

  1. User can type in a request at any time whatsoever.

  C. **Expected Result**

  1. Aidera should be robust and useful enough to always be able to handle requests at any time.
  2. The server downtime for maintenance should be minimal.

2.5 Use Case #5: Appropriate User Interface across all platforms

- TEST #1 : FUNCTIONALITY TEST

  A. Identification and Classification

  - UI/UX 001, User Experience
  - **Severity**: 3

  B. Instructions

  1. User types in a request on a mobile device and follows up with the conversation on desktop.

  C. Expected Result

  1. Aidera will handle the user interface across all platforms and create a seamless user experience.

2.6 Use Case #6: Follow the usage of the user

- TEST #1 : FUNCTIONALITY TEST

  A. Identification and Classification

  - Usage 001, Activity Tracking
  - **Severity**: 3

  B. Instructions

  1. Keep track of the user activities and usage without storing user credentials.

  C. Expected Result

  1. Aidera will store the user telemetry for further understanding of user requirements.

  2. Team Aidera will comprehend the user requirements and add new features and functionality.