INTRODUCTION TO PROGRAMMING

COIT20245

Week-8

Submitted By:

1. CHAKRADHAR GUBBALA-12255158
2. LAVANYA RAJULAPATI-12256008
3. VALIVELA VEERA VENKATA PAVAN-12260354

Table of Contents

[Introduction 3](file:///C:\Users\Chakri\AppData\Local\Temp\a880d8e2-25a4-4345-b708-1eb46ace59d6_BAHT_Chintu_303%20(1).zip.9d6\BAHT_Chintu_303\BAHT_Chintu_303.docx#_Toc166232747)

[Limitations 3](file:///C:\Users\Chakri\AppData\Local\Temp\a880d8e2-25a4-4345-b708-1eb46ace59d6_BAHT_Chintu_303%20(1).zip.9d6\BAHT_Chintu_303\BAHT_Chintu_303.docx#_Toc166232748)

[Known Bugs 3](file:///C:\Users\Chakri\AppData\Local\Temp\a880d8e2-25a4-4345-b708-1eb46ace59d6_BAHT_Chintu_303%20(1).zip.9d6\BAHT_Chintu_303\BAHT_Chintu_303.docx#_Toc166232749)

[Test Plan 4](file:///C:\Users\Chakri\AppData\Local\Temp\a880d8e2-25a4-4345-b708-1eb46ace59d6_BAHT_Chintu_303%20(1).zip.9d6\BAHT_Chintu_303\BAHT_Chintu_303.docx#_Toc166232750)

[Conclusion 5](file:///C:\Users\Chakri\AppData\Local\Temp\a880d8e2-25a4-4345-b708-1eb46ace59d6_BAHT_Chintu_303%20(1).zip.9d6\BAHT_Chintu_303\BAHT_Chintu_303.docx#_Toc166232751)

# Introduction

The goal of this project is to use Python to make an app for managing wildlife. The program's goals are to have an easy-to-use menu system, set up functions for different animal species and their sightings in different places, and connect to outside APIs to get data. The goal of this assignment is to help the student learn basic programming concepts like functions, user input, and API usage. It will also teach them how important it is to write clean code and test it. By putting their skills to use and facing the problems that come up when making real apps, this project might help students learn more about software development.

# Limitations

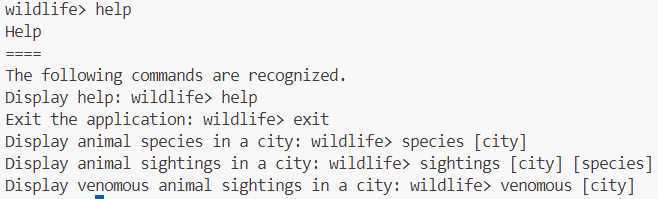
One of the problems with the program, though, is that it doesn't have a lot of tools for checking the data. The program reviews for basic input errors like making sure the commands and parameters are correct, but it fails to inspect for other errors like entering the data in the incorrect format or making sure the API replies are correct. The lack of this limitation will change how the program works and how it is organized. This is because it enables wrong or completely unexpected data to get through the system, which can cause it to crash or give the wrong result. Typing in the wrong name of a place or species, for example, could lead to API requests returning wrong or completely different results, which would make the app less reliable. In addition, assuming how the external service will act may also be against the program. For instance, an error might happen if the application thinks that it will always get answers in a certain format or that the external API service will always be available, but it doesn't. So, even though the program is only partially working right now, focusing on validating data and keeping an eye on the status of external services will make it much more stable.

# Known Bugs

There is a well-known bug in the way the program handles user input when they give an invalid order along with more invalid arguments. The program doesn't respond well when the user types in commands like "species" or "sightings" without naming a city or species, for example. This could cause an unexpected result. This bug makes the app less useful because it doesn't give clear feedback about the right way to enter data, which could be confusing for the user. Another bug happens when the external API can't return any data because of a server or network failure. When this happens, the program doesn't handle exceptions properly and ends quickly without telling the user what went wrong. These problems show that the application needs better error handling and data validation to make it more reliable and give users a better experience.

# Test Plan

1. Functionality: Display Help Menu
   * Input Values: User executes “help” command.
   * Expected Output: Help menu is displayed.
   * Assert Statement: Ensure that the help menu is printed when "help" command is entered.
   * Test Screenshots:



* + Test Status: Passed

1. Functionality: Exit the Program
   * Input Values: User executes “exit” command.
   * Expected Output: Program exits.
   * Assert Statement: Confirm that the program exits when the "exit" command is entered.
   * Test Screenshots:



* + Test Status: Passed

1. Functionality: Display Animal Species in a City
   * Input Values: User executes "species Cairns" command.
   * Expected Output: List of animal species in the specified city is displayed.
   * Assert Statement: Verify that the correct species are printed based on the provided city.
   * Test Screenshots:

A close up of a text

Description automatically generated

* + Test Status: Passed

1. Functionality: Display Animal Sightings in a City
   * Input Values: User executes "sightings Cairns snake" command.
   * Expected Output: List of animal sightings for the specified species in the specified city is displayed.
   * Assert Statement: Ensure that the correct sightings are printed based on the provided city and species.
   * Test Screenshots:

A close up of text

Description automatically generated

* + Test Status: Passed

1. Functionality: Display Venomous Animal Species in a City
   * Input Values: User executes “venomous Cairns” command.
   * Expected Output: List of venomous animal species in the specified city is displayed.
   * Assert Statement: Verify that the correct venomous species are printed based on the provided city.
   * Test Screenshots:



* + Test Status: Passed

1. Functionality: Invalid Command Handling
   * Input Values: The user executes an invalid command.
   * Expected Output: Error message displaying invalid command.
   * Assert Statement: Ensure that an error message is printed for invalid commands.
   * Test Screenshots:



* + Test Status: Passed

# Conclusion

The results have shown over the course of development how important it is to test and handle errors in a way that is both correct and applied. Regarding the program's current speed, even the function that has been proven to have bugs should focus on error handling and input validation. Validation can be made more thorough, and user feedback can be made more useful. The usefulness can also grow with the help of data visualization and user authentication. This shows that because software development is iterative, the quality of the products that are delivered is directly related to how willing people are to make improvements and change with the times.