Use Case: Execute Arbitrary Python Code, and Chart Data

**Traceability ID: EXEC-CODE-CHART-TECHNICAL**

**Relevant Requirements:**

### 3.1.1  Executing Code Snippets

### 3.1.2  Executing Code Snippets with Input

### 3.1.3  Rendering Charts

### 3.4.1  Text input

### 3.4.2  Text output

### 3.4.3  Image output

### 3.1.7  Connecting to Discord messaging layer

Non-functional

### 3.1.3  Node code execution

### 3.3.1  Code execution timeout

### 3.5.2  Code execution restriction

**Primary Actor:** User

**Pre-conditions:** User is in a channel for which the ResearchRealms bot has access to messages, and that their bot instance is running.

**Post-conditions**: The user is presented with the output of their small program, if it contains **plt.show()** then they are also presented with the chart they wish to display.

**Basic Flow or Main Scenario:**

1. The user enters their program into the current channel, preceded by the **!exec** command string.
2. The user uploads up to one attachment with their program string.
3. The user receives either the output of their program or some error (see alternate flow).

**Alternate Flows:**

1. The user is presented with an error detailing what went wrong. This can be one of:
   1. Exception due to timeout (program took too long to finish)
   2. Exception due to using a forbidden package
   3. Exception due to malformed program script (user error)
2. The user is presented with a warning that their program did not write to STDOUT
3. The user is presented with only an image file (of their chart) and no returned output from STDOUT

**Exceptions:** Programmatic Exceptions are returned as formatted messages to the source channel. The types of possible exceptions are detailed in **Alternate flows**.

**Related Use Cases:** (disjoint)

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| Date | Description | By |
| Mar 17, 2020 | Added Initial Use Case | Cole Smith |