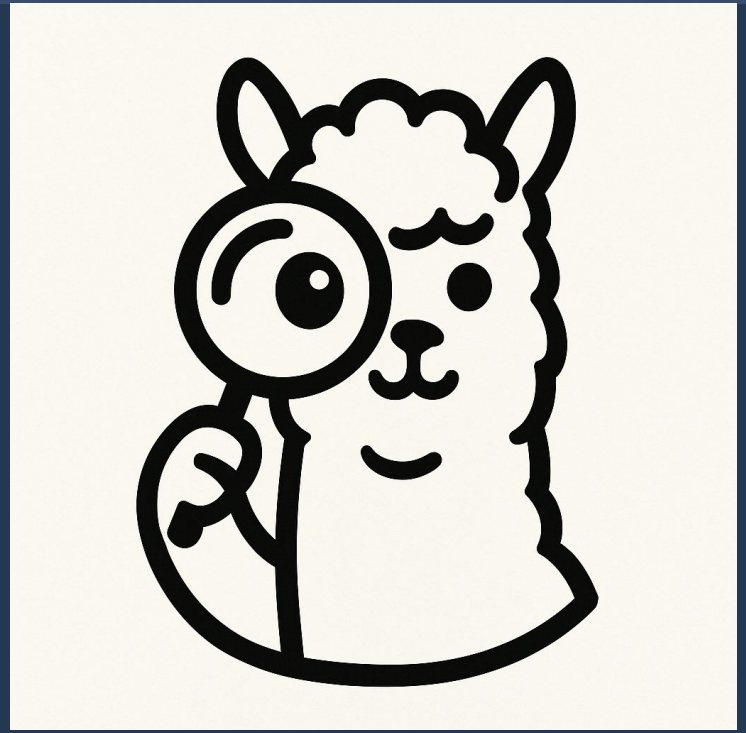
Aperture Labs Package Accountability Computer Assistant



By Hammud Haq

General Structure

Query User if they need to find a package [01]

Validate input [02]

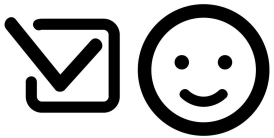
Ask if they have any information to find it with [03]

Validate input! [04]

Get information from DB [05]

Reset or Terminate [06]

3 Types of User Input categorization



Compliant Positive

User follows prompts and program flow so parsing and everything works fine

Compliant Negative

User follows prompts and program flow but decides to not use services or resets program

Malicious Negative

User enters garbage or malicious text which is either used to reset or forces an interpretation by the program

Our solution

The way I approached user input is essentially forcing it into the previously described categories so that the system doesn't crash or can trigger a reset. To do this we check for the length of responses, strip extra spaces, make sure they are following the correct format (integers where needed) and attempt to provide feedback on status of their input

```
def yesNoCheck(checkMe):
  # Assuming the person writes out a sentence or some phrase that contains yes or no
   if checkMe and len(checkMe) < 500:
      checkMe = checkMe.strip()
      checkMe = checkMe.lower()
      # Responses met parameters
      if "yes" in checkMe or "oui" in checkMe or "true" in checkMe or "affirmative" in checkMe or "si" in checkMe:
          return True
      elif "no" in checkMe or "non" in checkMe or "false" in checkMe or "negative" in checkMe or "nyet" in checkMe:
          print("No worries then! I hope you have a wonderful day!")
          return False
      # incase of a single letter response
      elif len(checkMe) == 1:
         if checkMe == "y":
            print("Kinda getting a yes vibe from your response")
            return True
         elif checkMe == "n":
            print("Kind of got a no vibe from your response")
            print("No worries then! I hope you have a wonderful day!")
            return False
      else:
          print("Hmm I'm not sure I understood that can we try that again?")
          return False
```