**Technical Design Document**

Kristina Chorna

June 2, 2025

**Program Description:**

The goal of this code is to create a program that prompts the user to enter an email and tests that message for keywords. It will display the number of keywords found and notify the user of the likelihood that the email is spam.

**Function Name:** calculate\_spam\_score()

**Description:** The goal of this function is to calculate the number of spam words found in the email. It converts the message to lowercase and loops through the list of spam words, adding each detected keyword.

**Parameters:** message – it stands for the input email message that will be checked for possible spam words.

**Variables:**

1. score – holds the number of spam keywords detected in the email message.
2. matched\_keyword – holds all the spam keywords found in the message.

|  |
| --- |
|  |

1. lower\_message – changes input message to lowercase for case sensitive matching.
2. keyword – represents each spam word from SPAM\_KEYWORDS

**Logical Steps:**

1. convert the message to lowercase
2. initialize score to 0
3. create the matched\_keywords list to store spam words found in the message
4. Create a loop and loop through the keywords in SPAM\_KEYWORDS and add 1 to the score every time a keyword is detected.
5. Return the score and matched\_keywords after the loop is finished

**Returns:** score and matched\_keywords

**Function Name:** evaluate\_spam\_likelihood()

**Description:** This function analyzes the spam score and determines the spam likelihood.

**Parameters:** score – it stands for the spam score and calculates the likelihood of an email being spam.

**Variables:**

1. score – determines the likelihood

**Logical Steps:**

1. if score is 0, return “Highly unlikely to be spam.”
2. If score is 1-3, return “It could be spam.”
3. If score is 4-6, return “Likely spam.”
4. If score is greater than 6, return “Highly likely to be spam.”

**Returns:** likelihood message

**Function Name:** main()

**Description:** This function prompts the user to enter an email and calls on the other two functions. It then displays the results – spam score, keywords found, and likelihood of it being a spam message.

**Parameters:** none

**Variables:**

1. message – stores the input email message from the user
2. score – holds the number of spam keywords found in the email
3. matches – holds the keywords detected in the email message
4. likelihood – holds the spam likelihood message
5. word – temporary variable used in a for loop to print each spam word found in the message

**Logical Steps:**

1. prompt user to enter an email message
2. Call the calculate\_spam\_score() function and save the returned score and matched\_keywords
3. Call the evaluate\_spam\_likelihood() function and score and save the returned likelihood message.
4. Print the spam score, likelihood, and keywords found; otherwise print that no keywords were found in the email

**Returns:** spam results and keywords

[COP2373 Repository](https://github.com/kristinachorna/COP2373)

A screenshot of a computer program

Description automatically generated