**Technical Design Document Template**

**Name:** Richard Kudrya

**Date Created:** 09/17/2025

**Program Description:**

This program will scan for 30 different words/phrases that are most common in spam emails/messages and will display the results of the message including the spam score, likelihood it is a spam, and the words/phrases that were detected.

**Functions used in the Program (list in order as they are called):**

1. **Function Name:** compile\_patterns(keywords

**Description:** Converts spam words/phrases into regex patterns

**Parameters:**

* keywords : list of str

**Variables:**

* esc : str
* pattern : str
* patterns : list

**Logical Steps:**

1. Starts with empty list
2. Loops through every keyword that was in the list
3. Escape keyword for regex
4. Make a pattern with word boundaries
5. Compile pattern with IGNORCASE
6. Returns a list of regex patterns

**Returns:** List of regex patterns

2. **Function Name:** score\_massage(message, patterns, keywords)

**Description:** Scans the message for each keyword/phrase and adds one point to the spam score

**Parameters:**

* message : str
* patterns : list
* keywords : list of str

**Variables:**

* counts : dict
* matches : list
* score : int

**Logical Steps:**

1. Creates a dictionary of keywords
2. Loops over each keyword
3. Finds and records the count of keywords detected
4. Adds the total score of keywords found

**Returns:** Number of keywords found and spam count

3. Function Name: interpret\_score(score, total\_keywords=len(KEYWORDS))  
Description: Returns the likelihood of spam according to the score  
Parameters:

- score : int

- total\_keywords : int  
Variables:

- pct : float  
Logical Steps:

1. If the spam score is 0 it will return a very unlikely that the message is spam

2. Divides number of keywords detected by total number of keywords to get the likelihood of spam   
Returns: The likelihood of the message being spammy.

Function Name: print\_results(score, interpretation, counts)  
Description: Displays the spam score, likelihood of the message being a spam and the words detected as spammy  
Parameters:

* score : int
* interpretation : str
* counts : dict

Variables:

Triggered : dict  
Logical Steps:

1. Print the spam score and likelihood
2. If no spammy words/phrases were found prints “no spammy words or phrases have been found”
3. Prints the words/phrases that were detected as spammy  
   Returns:

Prints results

**Link to your repository:** <https://github.com/Roboriko/PythonCOP2373>

**Output Screenshot: (make sure big enough so I can see)**

A screen shot of a computer

AI-generated content may be incorrect.