Specifications Document

<u>Project Overview</u>
This project implements a text encryption and decryption system using three classical cipher algorithms: Caesar Cipher, Vigenère Cipher, and One-Time Pad (OTP) Cipher. The user selects the cipher method, provides the text, and the corresponding key/shift value. The system processes the text accordingly, providing either an encrypted or decrypted output.

Feature 1: Caesar Cipher

Purpose:
The Caesar Cipher provides encryption and decryption by shifting characters in the text by a set number of positions in the English alphabet.

Assumptions:
- The input consists of alphabetic characters and possibly spaces, which will be converted to uppercase.
- Non-alphabetic characters will be ignored during encryption and decryption.
- The user provides an integer as the shift key.

Inputs:
Mode: `1` for encryption, `2` for decryption.
Input Text: The plaintext (for encryption) or ciphertext (for decryption).

Shift: An integer representing the number of positions each letter is shifted in the alphabet.

Outputs:
Ciphertext: The encrypted text when in encryption mode.
Plaintext: The decrypted text when in decryption mode.
The result is saved in a file called `ciphertext.txt`.
The shift key is saved in a file called `shift_text.txt`.

State Changes:
Each letter in the input string is shifted by the given amount during encryption or in reverse during decryption. Spaces and non-alphabet characters remain the same.

Cases and Expected Behavior:
Encryption Case:
Input: `"HELLO"`, Shift: `3`
Expected Output: `"KHOOR"`

Decryption Case:
Input: `"KHOOR"`, Shift: `3`
Expected Output: `"HELLO"`

Feature 2: Vigenère Cipher

Purpose:
The Vigenère Cipher provides encryption and decryption by using a keyword that determines the shift for each character in the text. Unlike the Caesar Cipher, the Vigenère Cipher uses a repeating key, making it more resistant to frequency analysis.

Assumptions:
- The input consists of English alphabet characters and possibly spaces, which will be converted to uppercase.
- Non-alphabetic characters are ignored during encryption and decryption.
- The user provides a string of English alphabet characters as the key.

Inputs:
Mode: `1` for encryption, `2` for decryption.
Input Text: The plaintext (for encryption) or ciphertext (for decryption).
Key: A string of English alphabetic characters provided by the user.

Outputs:
Ciphertext: The encrypted text when in encryption mode.
Plaintext: The decrypted text when in decryption mode.
The result is saved in a file called `ciphertext.txt`.
The key is saved in a file called `key_text.txt`.

State Changes:
Each character in the input string is shifted by the value of the corresponding character in the key during encryption. During decryption, the reverse shift is applied based on the key.

Cases and Expected Behavior:
Encryption Case:
Input: `"HELLO WORLD"`, Key: `"KEY"`
Expected Output: `"RIJVS UYVJN"`

Decryption Case:
Input: `"RIJVS UYVJN"`, Key: `"KEY"`
Expected Output: `"HELLO WORLD"`

Feature 3: One-Time Pad (OTP) Cipher

Purpose:
The One-Time Pad (OTP) Cipher uses a random key that is the same length as the input text. Each character in the plaintext is encrypted using a corresponding character in the randomly generated key.

Assumptions:
- The input consists of alphabetic characters and possibly spaces, which will be converted to uppercase.
- The key is randomly generated for encryption and must be the same length as the input text.
- The key contains only alphabetic characters (A-Z).

Inputs:
Mode: `1` for encryption, `2` for decryption.
Input Text: The plaintext (for encryption) or ciphertext (for decryption).
Key:
For encryption: A randomly generated key matching the length of the input text.
For decryption: The same key used during encryption, provided by the user.

Outputs:
Ciphertext: The encrypted text when in encryption mode.
Plaintext: The decrypted text when in decryption mode.
The result is saved in a file called `ciphertext.txt`.
The key is saved in a file called `key_text.txt` during encryption.

State Changes:
During encryption, each letter in the input text is shifted based on a corresponding randomly generated letter in the key. During decryption, the reverse shift is applied using the same key.

Cases and Expected Behavior:
Encryption Case:
Input: `"HELLO"`
Generated Key: `"XMCKL"`
Expected Output: `"EQNVZ"`

Decryption Case:
Input: `"EQNVZ"`
Key: `"XMCKL"`
Expected Output: `"HELLO"`

Common System Features

File Handling:

- The system saves the results of encryption or decryption in a file named `ciphertext.txt`.
- The system also saves the key or shift value in a separate file (`key_text.txt` for Vigenère and OTP, `shift_text.txt` for Caesar).

User Interaction:

- The system prompts the user for the mode (encrypt/decrypt), cipher type, input text, and the necessary key or shift.
- After processing, the system displays the output and saves it to a file.
- The user is asked if they want to perform another operation, allowing for repeated use without restarting the program.