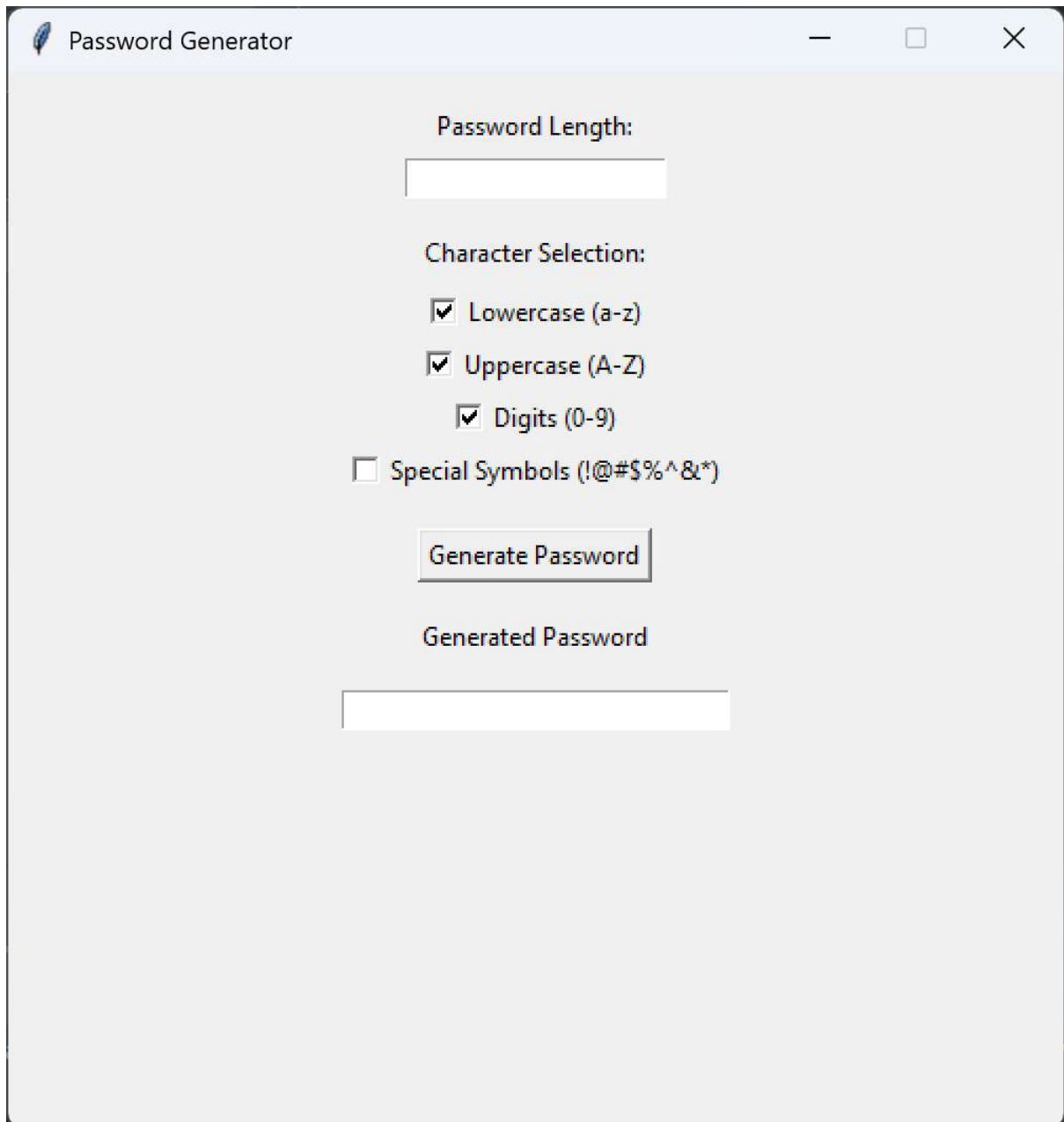


### Aufgabe 1:

Enthält die Dateien: „Aufgabe1.py“, „gui\_Aufgabe1“ und „password\_generator.py“.

- „Aufgabe1.py“ : Main entrypoint, ruft die main Funktion der „gui\_Aufgabe1.py“ Datei auf. Diese initialisiert die Gui und setzt Funktionalitäten wie Knöpfe, Eingabefelder etc auf. In die Gui logik geht dann noch die Funktionalität der password generation aus der Datei „password\_generator.py“ ein. Sie wird mit dem aktivieren eines GUI buttons verbunden.



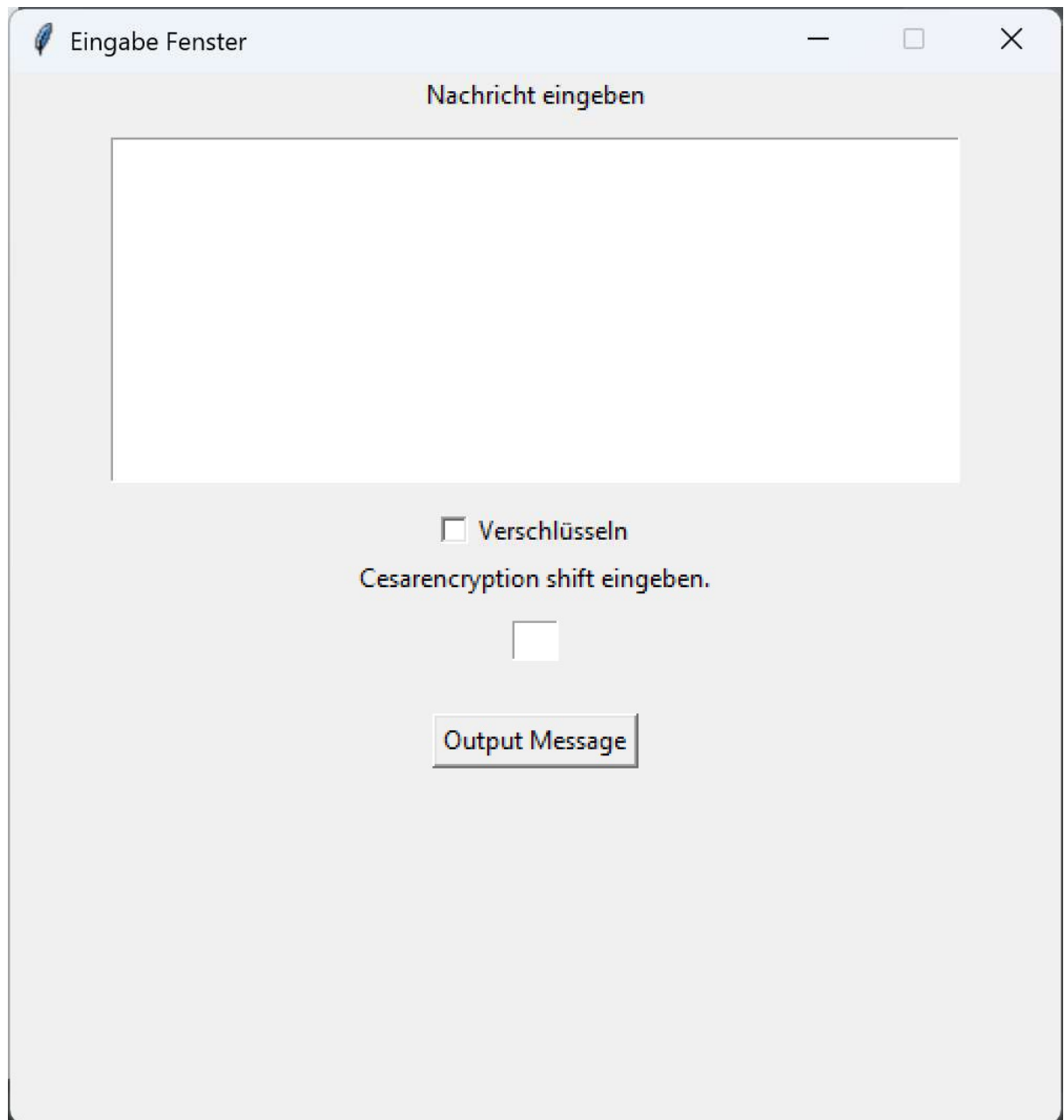
The screenshot shows a window titled "Password Generator" with a standard macOS-style title bar (blue with a feather icon, and grey with minimize, maximize, and close buttons). The window has a light grey background. The content is centered and includes the following elements from top to bottom:

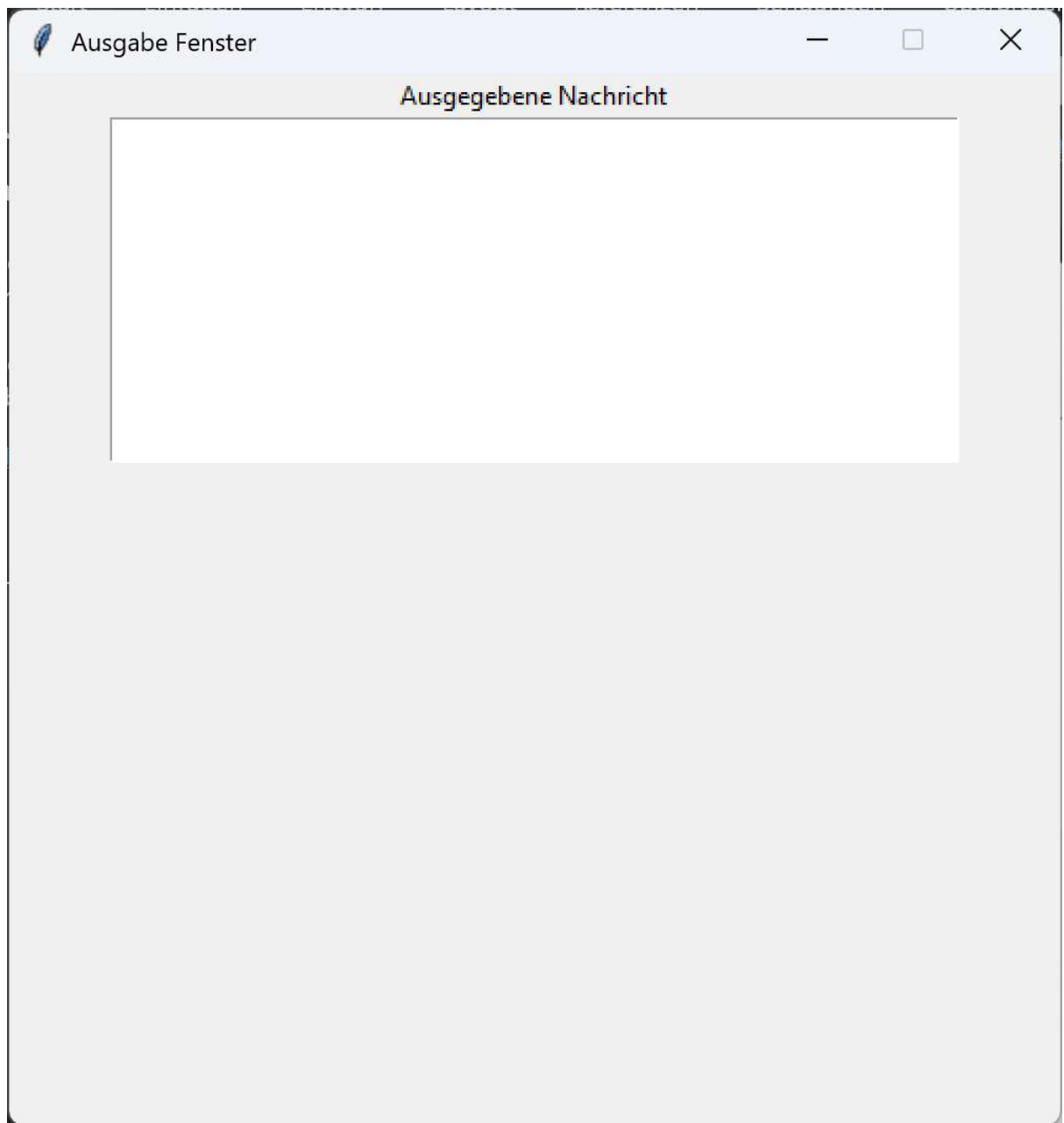
- The label "Password Length:" followed by a white text input field.
- The label "Character Selection:" followed by four checked checkboxes: "Lowercase (a-z)", "Uppercase (A-Z)", and "Digits (0-9)".
- A single unchecked checkbox labeled "Special Symbols (!@#\$%^&\*)".
- A button labeled "Generate Password".
- The label "Generated Password" followed by a white text input field.

## Aufgabe 2:

Enthält die Dateien: „Aufgabe2.py“, „gui\_Aufgabe2“ und „encryption.py“.

- „Aufgabe2.py“: Main entrypoint, ruft wie zuvor die main Funktion der „gui\_Aufgabe2.py“ Datei auf, diese setzt das root Fenster und das Ausgabefenster. Die Erstellung dieser wird in 2 unterschiedlichen Funktionen gehandhabt. Anschließend wird noch die Verbindung zwischen den beiden Fenstern mittels eines Buttons zum Verschlüsseln der Nachricht hergestellt. Die Verschlüsselung wird in „encryption.py“ gehandhabt diese greift auf vorher eingerichtete Parameter der gui zu und führt entsprechend die Caesar verschlüsselung durch.





Aufgabe 3 & 4: Abarbeiten der Aufgabenstellung, diese Dokumentiert ziemlich gut was passiert.