Dokumentation

Inhaltsverzeichnis

1. Funktionen des Passwortmanagers

    - Nutzer können sich einen eigenen Account erstellen und sich auf diesem Account an bzw. abmelden.
    - Angemeldete Nutzer können ihre Anmeldedaten ändern oder ihren Account komplett löschen
    - Ist ein Nutzer angemeldet, kann er einen neuen Eintrag erstellen, indem er eine URL und ein Passwort, sowie optional eine Notiz angibt
    - Das Passwort kann manuell eingegeben oder automatisch erstellt werden
    - Die Stärke des Passworts wird ermittelt um den Nutzer eventuell vor einem schwachen Passwort zu warnen
    - Bereits vorhandene Einträge können auch wieder gelöscht oder vollständig überarbeitet werden
    - Außerdem kann ein Nutzer auch nach einem Eintrag, welcher bestimmte Wörter enthält suchen

2. Grundlegender Aufbau

    2.1 Architekturbeschreibung

    Das Projekt besteht aus den 4 folgenden Klassen:

    - main.py
    - logic.py
    - user.py
    - entry.py

    Dabei ist die main.py die Ausführungsklasse. Sie kontrolliert die ganze grafische Oberfläche und damit die Aus – und Eingaben auf der Konsole.

    In der logic.py findet die komplette Logik des Passwortmanagers statt.

    Die Klasse User stellt die Verallgemeinerung eines Nutzers dar.

    Von der Klasse Entry werden Eintragsobjekte erstellt.

2.2 Detaillierte Klassenbeschreibung

main.py

Attribute:

- functions = Logic()
- loggedUser = None
- generatedEntry = Entry()

Funktionen:

- _init_()
  Wird bei Programmstart ausgeführt und initialisiert die 3 Referenzattribute
- start()
  Erstellt die Startseite auf der Konsole
- login()
  Erstellt die Anmeldeseite auf der Konsole
- createAccount()
  Erstellt die Registrierseite auf der Konsole
- openMainPage()
  Erstellt Benutzer Startseite
- createEntry()
  Erstellt die Seite, auf der ein neuer Eintrag erstellt werden kann
- searchForEntry()
  Erstellt die Seite, auf der nach Einträgen gesucht werden kann
- generatePassword()
  Erstellt die Seite, auf der Passwörter generiert werden können
- openPersonalSite()
  Erstellt die Seite, auf der persönliche Daten geändert werden können
- listAllEntrys()
  Erstellt die Seite, auf der Einträge bearbeitet oder gelöscht werden können

logic.py

Attribute:

- allUser (list)
- alphabetLowercase (list)
- alphabetUppercase (list)
- numberList (list)

- symbolsList (list)
- completeList (list)

Funktionen:

- tryToLogin(username,password): User
  versucht mit den eingegebenen Daten einen Nutzer anzumelden
- alreadyUsed(username): bool
  prüft, ob es bereits einen Nutzer mit gleichem Nutzernamen gibt
- checkPasswordLength(password): bool
  prüft, ob das Passwort die erforderliche Länge hat
- addNewUser(newUsername)
  fügt der Liste allUsers einen neuen Benutzer hinzu
- deleteUser(userToDelete)
  löscht den Account und alle Daten des zu diesem Zeitpunkt angemeldeten Nutzers
- updateUser(user)
  Aktualisiert die Liste allUsers, wenn Änderungen vorgenommen wurden
- addOldPassword(oldPassword,loggedUser)
  fügt ein Passwort in die Liste aller, der vom Nutzer bereits verwendeten Passwörter
- isOldPassword(newPassword,loggedUser): bool
  prüft, ob ein Passwort bereits schon genutzt wurde
- addEntry(entry,loggedUser)
  fügt einen neuen Eintrag in die Liste myEntrys der Klasse user.py hinzu
- deleteEntry(entryToDelete,loggedUser)
  löscht einen Eintrag aus der Liste myEntrys
- updateEntry(entry,loggedUser,oldUrl)
  aktualisiert die Liste myEntrys, nach Änderungen von Einträgen
- goThroughEntry(entry,searchedFor): bool
  sucht nach einem bestimmmten Wort oder Satz in allen Einträgen
- generatePassword(length,strongness): string
  generiert ein Passwort einer bestimmten Länge und Stärke, indem je nach Stärke nur bestimmte Zeichen zugelassen werden
- checkIfSimmilarEntryExists(url,loggedUser): entry
  prüft, ob es bereits einen Eintrag mit gleicher Url gibt
- getSafetyLevel(password): int
  prüft anhand mehrere Berechnungen, wie sicher ein Passwort ist
- getAmmountOfDifferentChars(password): int
  berechnet, wie viel verschiedene Zeichen in dem eingegebenen Passwort vorkommen
- combinationPoints(password): int
  berechnet, die Punkte der Stärke der Kombinationen von unterschiedlichen Kategorien von Zeichen (z.B hat ein Passwort bestehend aus kleinbuchstaben eine geringere Sicherheit als ein Passwort aus Kleinbuchstaben und Symbolen)
- containsLower(password): bool

prüft, ob das Passwort Kleinbuchstaben enthält

- containsUpper(password): bool
  prüft, ob das Passwort Großbuchstaben enthält
- containsNumber(password): bool
  prüft, ob das Passwort Zahlen enthält
- containsSymbol(password): bool
  prüft, ob das Passwort Kleinbuchstaben enthält

user.py

Attribute:

- username
- password
- myEntrys (list)
- oldPasswords (list)

Funktionen:

- _init_(username,password)
  Konstruktormethode der Klasse
- getUsername(): string
  gibt den aktuellen Nutzernamen zurück
- setUsername(username)
  setzt einen neuen Nutzernamen
- getPassword(): password
  gibt das aktuelle Passwort zurück
- setPassword(password): setzt ein neues Passwort