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

entry.py

Attribute:

password

url

notice

Funktionen:

- _init_(password,url,notice)
  Konstruktormethode der Klasse
- getPassword(): string
  gibt das aktuelle Passwort zurück
- setPassword(password)
  setzt ein neues Passwort
- getUrl(): string
  gibt die aktuelle URL zurück
- setUrl(url)
  setzt eine neue URL
- getNotice(): string
  gibt die aktuelle Notiz zurück
- setNotice(notice)
  setzt eine neue Notiz

5. Nutzerinterface

Das Nutzerinterface ist so aufgebaut, dass immer nur das nötigste auf der Konsole angezeigt wird, damit die Ansicht für den Nutzer übersichtlich bleibt und nicht irreführend ist. Dafür wird der Inhalt der Konsole regelmäßig mit dem Befehl os.system(‘cls‘) gelöscht.

Dem Nutzer werden je nachdem, auf welcher Seite er sich befindet verschiedene Option vorgeschlagen, zwischen denen er wählen kann. Dabei wird fast immer jeder Option eine Zahl zugewiesen, über die der Nutzer eine bestimmte Option auswählen kann. Nur bei der Passwortgenerierung wird zur Auswahl der Buchstabe (g) für generate benutzt. Abbrüche und Bestätigungen werden über die Enter Taste getätigt. Dadurch, dass der Nutzer nie mehr als ein Zeichen zur Interaktion auswählen muss, ist ein schneller und leicht verständlicher Programmablauf möglich.

6. Programmablauf

Nach Start des Programms öffnet sich die Konsole und der Nutzer bekommt die Möglichkeit sich anzumelden, einen neuen Account zu erstellen oder das Programm zu beenden. Möchte er sich anmelden, wird er dazu aufgefordert seinen Benutzername und sein Masterpasswort einzugeben. Existiert dieses Konto wird er auf die Startseite des Kontos weitergeleitet. Existiert dieses Konto nicht, bekommt der Nutzer die Meldung, dass Nutzername oder Passwort möglicherweise falsch sind und er wird aufgefordert seine Daten erneut einzugeben. Hat der Nutzer noch gar kein Konto, kann er einen neuen Account erstellen. Um dies zu tun, wählt er einen Benutzernamen aus. Wenn dieser noch von keinem anderen User benutzt wird, wird er aufgefordert ein Passwort mit der Mindestlänge 10 einzugeben. Hier

kann der Nutzer entweder entscheiden manuell ein Passwort zu wählen oder ein Passwort generieren zu lassen. Lässt er es generieren, hat er die Möglichkeit eine bestimmte Länge und eine gewünschte Stärke anzugeben. Gefällt ihm das generierte Passwort, kann er es verwenden oder den Vorgang wiederholen. Gibt der Nutzer ein Passwort manuell ein, wird erstmal überprüft, ob die Passwortlänge mindestens 10 beträgt. Falls ja wird überprüft, wie sicher dieses Passwort ist um den User gegebenenfalls zu warnen. Hat alles funktioniert wird ein neuer Account erstellt und der User wird auf die Login Seite weitergeleitet. Dort kann er seine Daten eingeben und erhält nach erfolgreicher Anmeldung vollen Zugriff auf sein Konto. Nun hat er die Möglichkeit neue Einträge hinzuzufügen, nach einem bestimmten Eintrag zu suchen, Einträge zu löschen oder zu bearbeiten und Änderungen am Konto vorzunehmen.

Möchte der Nutzer einen neuen Eintrag hinzufügen, wird er aufgefordert eine URL einzugeben. Existiert ein Eintrag mit dieser URL bereits, wird der Nutzer darauf aufmerksam gemacht und gefragt, ob er den alten Eintrag überschreiben möchte oder den Vorgang wiederholen möchte. Nachdem er eine URL eingegeben hat, wird nach dem zu speichernden Passwort gefragt. Hier kann der Nutzer bereits wie bei der Anmeldung wieder zwischen einem manuell und einem automatisch generierten Passwort auswählen. Zusätzlich wird hier jedoch überprüft, ob der Nutzer dieses Passwort jemals für einen seiner Einträge benutzt hat. Falls nein kann er dieses Passwort verwenden und wird aufgefordert Bemerkungen bzw. Notizen dem Eintrag hinzuzufügen. Diese Eingabe ist allerdings optional. Hat alles funktioniert, wird der neue Eintrag gespeichert. Nun hat der Benutzer die Möglichkeit weitere Einträge hinzuzufügen oder zurück zur Startseite zu gelangen. Ist er wieder auf der Startseite kann er auswählen nach einem Eintrag zu suchen. Dies kann er tun, indem er ein oder mehrere Wörter eingibt, an die er sich erinnern kann und welche eventuell im Eintrag vorkommen könnten. Gibt es einen Treffer wird dieser dem Nutzer angezeigt. Außerdem gibt es die Option Einträge zu bearbeiten oder zu löschen. Dafür werden alle existierenden Einträge gelistet und dem Nutzer zusammen mit einer fortlaufenden Nummer gezeigt. Über diese Nummer kann er einen bestimmten Eintrag auswählen und kann dann entscheiden den Eintrag zu löschen oder zu überarbeiten. Gelangt der Nutzer auf seine persönliche Seite, hat er hier die Möglichkeit sein Masterpasswort oder seinen Nutzernamen zu ändern. Dazu muss er die Änderung aber nochmal bestätigen, bevor sie endgültig gespeichert wird. Hier kann der Nutzer außerdem seinen Account unwiderruflich löschen. Dies geschieht ebenfalls nur nach zusätzlicher Bestätigung. Ein User kann sich auch ausloggen und kommt dann auf die Anmeldeseite, von der aus er den Passwortmanager schließen kann