M1: Konzeptvorschlag (SecureAuthApp)

Projektname: SecureAuthApp

Programmierwerkzeug

- Sprache: Java
- Framework: JavaFX
- Datenbank: SQLite (Viewer Web App)
- Entwicklungsumgebung: VS Code

Verschlüsselungsalgorithmus für das Passwort

- Algorithmus: BCrypt

BCrypt ist ein moderner Hashing-Algorithmus, der speziell für die sichere Speicherung von Passwörtern entwickelt wurde. Er verwendet Salt und Pepper, um zusätzliche Sicherheit zu gewährleisten und Brute-Force-Angriffe zu erschweren.

Datenbank-Tabellen:

- Tabelle 1: User
    - Feldnamen:
        - id (INTEGER, PRIMARY KEY, AUTOINCREMENT)
        - email (TEXT, UNIQUE)
        - password_hash (TEXT)
        - signup_date (DATE)
- Tabelle 2: Security
    - Feldnamen:
        - user_id (INTEGER, FOREIGN KEY)
        - salt (TEXT)

Hinweis: Verschlüsseltes Passwort, Salt und Pepper (geheim) sind in separaten Tabellen gespeichert, um die Sicherheit zu erhöhen. (Pepper wird in einer ***.env*** *Datei gespeichert, um ihn geheim zu halten!)*

Tests zur Überprüfung von Methoden

Test 1: Überprüfung eines schwachen Passworts

Es wird überprüft, ob das Passwörter, den festgelegten Anforderungen entspricht (z. B. weniger als 8 Zeichen, keine Sonderzeichen, usw.)

Test 2: Validierung der E-Mail

Der Test validiert, dass die E-Mail-Adresse korrekt formatiert ist (z. B. name@domain.com).

Kürteil – Auswahl der Funktionen

- Kürteil 4: Passwortstärke überprüfen

  Die Passwortstärke wird anhand vordefinierter Kriterien überprüft, wie mindestens 8 Zeichen, Gross- und Kleinbuchstaben, Zahlen und Sonderzeichen. Die Anforderungen sollen sicherstellen, dass das Passwort ausreichend komplex ist.

- Kürteil 5: Passwortsperre nach mehreren Fehlversuchen (20 Sek. warten)

  Nach drei fehlgeschlagenen Login-Versuchen wird der Benutzer für 20 Sekunden gesperrt, bevor ein neuer Anmeldeversuch unternommen werden kann. Diese Massnahme soll Brute-Force-Angriffe erschweren.