Projektbeschrieb – Gruppe 20 TZT

1. *Projekttitel*

Kontaktbuch-Manager

2. Projektübersicht

Diese Anwendung dient als einfaches Verwaltungstool für Kontakte und ermöglicht es Nutzern, ihre persönlichen sowie geschäftlichen Kontaktinformationen zentral zu organisieren. Nutzer können neue Kontakte mit Angaben wie Name, Telefonnummer und E-Mail-Adresse hinzufügen, bestehende Einträge bearbeiten oder löschen und ihre gespeicherten Daten gezielt durchsuchen. Alle Kontaktinformationen werden dauerhaft lokal in Dateien gespeichert, sodass sie jederzeit zuverlässig verfügbar sind.

3. Motivation / Begründung

Wir haben dieses Projekt gewählt, weil die Verwaltung von Kontakten ein alltägliches und praxisnahes Szenario darstellt, das jeder aus dem privaten oder beruflichen Umfeld kennt. Ein digitales Kontaktbuch bietet die Möglichkeit, grundlegende Funktionen wie das Anlegen, Bearbeiten, Löschen und Durchsuchen von Daten umzusetzen. Dadurch können wir die im Kurs erlernten Konzepte der Dateiverarbeitung, Datenstrukturen und Benutzerinteraktion in einem nützlichen und realistischen Kontext anwenden.

4. Zielgruppe

Die Anwendung richtet sich an Nutzer, die ihre persönlichen oder geschäftlichen Kontakte digital speichern, verwalten und schnell wiederfinden möchten.

5. Hauptfunktionen und Benutzerablauf

5.1 Funktionsübersicht

- **Kontakte speichern**: Alle Daten werden dauerhaft in einer Datei abgelegt.
- **Hinzufügen**: Neue Kontakte mit Namen, Telefonnummer, E-Mail eintragen.
- **Entfernen**: Kontakte anhand einer eindeutigen ID, Telefonnummer oder E-Mail-Adresse löschen.
- **Bearbeiten**: Bestehende Kontakte ändern, ohne sie neu anlegen zu müssen
- **Suchen**: Kontakte nach Namen, Telefonnummer oder E-Mail-Adresse finden (auch Teiltreffer).
- **Auflisten**: Alle gespeicherten Kontakte anzeigen, bei der nach Vornamen, Nachnamen oder Teiltreffern (z.B nur Anfangsbuchstabe) gesucht werden kann. (Filtern)
- **Anrufe**: Anrufe tätigen über Eingabe von Kontakt oder Telefonnummer. Vergangene Anrufe werden abgespeichert und können abgerufen werden.
- Daten speichern und Programm beenden

5.2 Programmablauf

Programmstart → HAUPTMENÜ (zeigt Optionen 1-5 an)
↓
Benutzer wählt Option → Ausführung der gewählten Funktion
↓
Anzeige von Ergebnissen / Bestätigung
↓
Rückkehr zum HAUPTMENÜ (Schleife läuft bis zum Beenden)
↓
Speichern aller Daten → Programmende

6. Erforderliche Kriterien

6.1 Interaktive Anwendung

Die Kunden interagieren über eine menügesteuerte Konsolenoberfläche. Sie wählen Optionen durch Eingabe von Zahlen (1-5) aus und geben Daten ein, wenn sie dazu aufgefordert werden (Telefonnummer und Personendaten). Das Programm durchläuft kontinuierlich eine Schleife durch das Menü, bis der Benutzer sich für das Beenden entscheidet. Bei jeder Eingabe erfolgt eine Validierung und bei ungültigen Eingaben wird der Kunde mit klaren Fehlermeldungen zur erneuten Eingabe aufgefordert.

6.2 Datenvalidierung

Das Programm validiert alle Benutzereingaben umfassend. Telefonnummern werden auf numerische Zeichen und minimale Länge geprüft (+41 000 00 00). E-Mail-Adressen werden auf das korrekte Format überprüft (enthält @ und Domäne). Für die Fehlerbehandlung werden try-except-Blöcke verwendet, um ungültige Eingaben abzufangen und den Kunden mit klaren Fehlermeldungen zur erneuten Eingabe aufzufordern.

6.3 Dateiverarbeitung

Die Anwendung verwendet mehrere Dateien zur Datenspeicherung. In der Datei "contacts.txt" werden alle Kontakte mit den Zeilen User-ID, Name, Vorname, Telefonnummer, E-Mail gespeichert. Diese Datei wird beim Programmstart geladen. Die Datei "calls.txt" speichert alle vergangenen Anrufe mit Datum, Zeit und Telefonnummer. Nach jedem neuen Telefonat oder Änderung werden die aktualisierten Daten sofort in die Datei geschrieben, um Datenverlust zu vermeiden.

7. Technische Anforderungen

Das Projekt verwendet folgende Python-Konzepte:

- **Datentypen:** Strings, Floats, Integers, Listen, Dictionaries
- **Kontrollstrukturen:** if/elif/else, while-Schleifen (Menü), for-Schleifen (Kontaktsuche)
- **Funktionen:** Separate Funktionen für jede Hauptfunktion (create_contact, check_contact, show_contacts, etc.)
- **Dateiverarbeitung:** open(), read(), write()
- **Exception-Handling:** try-except für Eingabevalidierung und Dateioperationen
- **String-Operationen:** Formatierung, E-Mail- und Telefonnummer-Validierung

8. Erwartete Herausforderungen

Die Validierung von E-Mail-Adressen und Telefonnummern erfordert String-Operationen und Pattern-Matching mit grundlegenden Checks (enthält @, Punkt nach @). Um Datenverlust zu vermeiden, wird nach jeder Änderung sofort in die Datei gespeichert. Für die Eingabevalidierung werden wiederverwendbare Validierungsfunktionen mit Schleifen erstellt, die bei ungültiger Eingabe zur erneuten Eingabe auffordern. Beim Umgang mit Dateien muss überprüft werden, ob Dateien existieren, bevor sie gelesen werden, und neue Dateien müssen bei Bedarf erstellt werden.

9. Teammitglieder

David Kopp
Numana Sajid
Naomi Uwnesuyi
Arthur Saryan