## **Informationstechnisches XML-Projekt**

## **Projekt Vokabeltrainer**

Präsentiert von Lazar, Yousif, Simon & Jan

## Inhaltsverzeichnis des Pflichtenhefts

Inhalt Seite

1. Titelblatt 1
2. Inhaltsverzeichnis 2
3. Anforderungen / Ziele 3
4. Ideen / Aufbau 4
5. Machbarkeitsprüfung 5-6
6. Projektstrukturplan 7
7. Software - Design 8
8. GUI -und UML-Bilder 9-11
9. Code 12

**Anforderungen / Ziele**

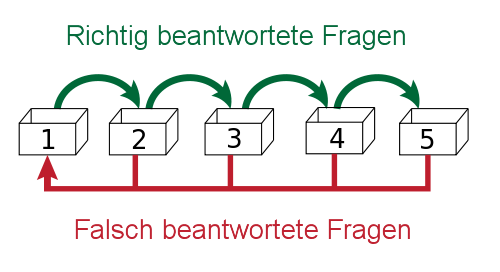
Was soll das Programm leisten?

Im Wesentlichen soll der Vokabeltrainer dazu dienen, Vokabeln effizient zu lernen und zu verinnerlichen.

Das heißt also, dass nicht nur neue Vokabeln gelernt werden sollen, sondern ebenfalls bereits gelernte Vokabeln wiederholt und vertieft werden.

Wie soll es funktionieren?

Man kann es sich im Grunde wie ein Karteikartensystem vorstellen, mit unterschiedlichen Fächern.



**Ideen / Aufbau**

Zunächst einmal sollte unser Programm die Grundfunktionen erledigen können. Dazu zählen für uns zuallererst die Abfrage von Vokabeln und ein Kastensystem (siehe oben). Die Vokabeln sollen aus CSV Dateien in das Programm importiert werden und nachträglich auch verändert werden können. Außerdem soll man beliebig viele Vokabelbestände aus beliebigen CSV Dateien anlegen können, in welchen jeweils anderer Lernstand gespeichert, sodass eine Mehrbenutzerverwendung garantiert werden kann. Unser Programm soll das Leben erleichtern und das Lernen einfacher machen. Es soll schnell und leicht zu bedienen sein.

**Machbarkeitsanalyse**

**-Welche Software wird für den Vokabeltrainer benötigt?**

Visual Studio

Git Hub

.Net Framework

Metro Framework

Windows Umgebung

**-Welche Hardware wird für den Vokabeltrainer benötigt?**

Hardwaretechnisch wird ein herkömmlicher Windows-PC benötigt.

**-Haben wir durch den Vokabeltrainer tragende Kosten?**

Für den Vokabeltrainer kostet uns die Programmierung, durch beispielsweise Software-Lizenzen keinen Cent, somit sollte uns der Erfolg des Programms garantiert sein.

**Machbarkeitsanalyse**

**Was sind unsere Team-Kompetenzen?**

Jeder aus der Gruppe hat selbstverständlich schon einmal ein paar kleinere Programme geschrieben, wenn auch nicht jeder mit der gleichen Geschwindigkeit.

Allerdings wenn es um größere und komplexere Programme geht, sticht eine Person aus der Gruppe heraus.

Simon ist aus unserer Gruppe mit Abstand der kompetenteste Programmierer, da dieser auch in der Freizeit gerne einmal vor dem Rechner sitzt, um dort seine Programmierfähigkeiten weiter zu verbessern.

Lazar legt großen Wert auf Details und deshalb haben wir uns als Gruppe entschlossen, dass Lazar das Design macht.

Yousif und Jan hingegen sind sehr fleißige Schreiberlinge, es sollte nun klar sein, dass diese beiden Gruppenmitglieder die Dokumentation anfertigen.

**Machbarkeitsanalyse - Fazit**

Durch die Verteilung der Arbeitsrollen ist unsere Gruppe nun festen Herzens entschlossen, diese Aufgabe mit Bravour und Standhaftigkeit zu meistern.

Mit diesen Voraussetzungen kann uns kein Unheil auf der gesamten Welt davon abbringen, dieses programmierte Meisterwerk zu konzipieren.

**Projektstrukturplan**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **SOLL** | | | |  | | | | **IST** | | |
| **Nr.** | **Vorgang** | **Anfang** | **Ende** | **Dauer** | **Hauptverantwortliche** | | **Vorgänger** | **Erledigt** | **Anfang** | | **Ende** | **Dauer** |
| 1 | - Aufgabenplanung und Einteilung | 18.09 | 18.09 | 30 Minuten | Simon, Lazar, Jan, Yousif | | / | ✓ | 18.09 | | 18.09 | 30 Minuten |
| 2 a) | - Ideenfindung | 18.09 | 18.09 | 30 Minuten | Simon, Lazar, Jan, Yousif | | / | ✓ | 18.09 | | 18.09 | 30 Minuten |
| 2 b) | - Designentwurf | 20.09 | 20.09 | 2 Stunden | Lazar | | 2 a) | ✓ | 20.09 | | 20.09 | 2 Stunden |
| 3 | - XML-Strukturentwurf | 21.09 | 22.09 | 2 Stunden | Simon | | / | ✓ | 22.09 | | 22.09 | 1 ½ Stunden |
| 4 | - Erstellung und Verknüpfung der Views mit MVC | 24.09 | 24.09 | 2 Stunden | Simon | | / | ✓ | 24.09 | | 24.09 | 2 Stunden |
| 5 | - Dokumentation (Pflichten und Lastenheft) | 18.09 | 27.10 | 5 Stunden | Jan, Yousif | | / | ✓ | 18.09 | | 24.09 | 4 ½ Stunden |
| 6 a) | - Realisierung der CSV zu XML Konvertierung | 26.09 | 26.09 | 1 Stunde | Simon | | / | ✓ | 26.09 | | 26.09 | 1 Stunde |
| 6 b) | - Realisierung der Abfrage | 01.10 | 1.10 | 4 Stunden | Lazar, Jan | | 6 a) | ✓ | 01.10 | | 1.10 | 4 Stunden |
| 6 c) | - Realisierung des Bearbeitungsmodus | 4.10 | 4.10 | 4 Stunden | Simon, Yousif | | 6 b) | ✓ | 4.10 | | 4.10 | 4 Stunden |
| 7 | - View-Überarbeitung und Bugbeseitigung | 7.10 | 7.10 | 2 Stunden | Simon, Lazar, Jan, Yousif | | / | ✓ | 7.10 | | 7.10 | 2 Stunden |
| 8 | - Erstellen der Abschlusspräsentation | 27.10 | 27.10 | 2 Stunden | Simon, Lazar, Jan, Yousif | | / |  | 27.10 | | 27.10 | 2 Stunden |

**Software-Design**

**Benutzerrollen:**

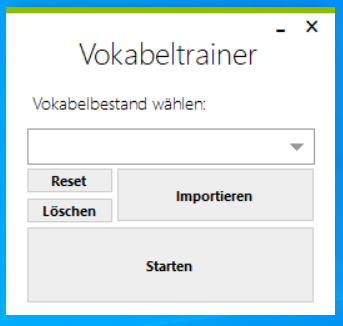
Jeder Schüler besitzt ein eigenes individuelles Profil, damit die Lernstände und Fortschritte sinnvoll für jeden gespeichert werden.

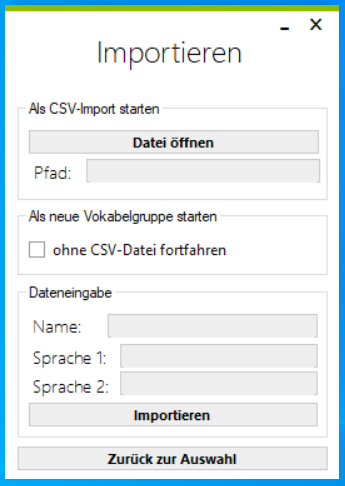
**Besondere Features:**

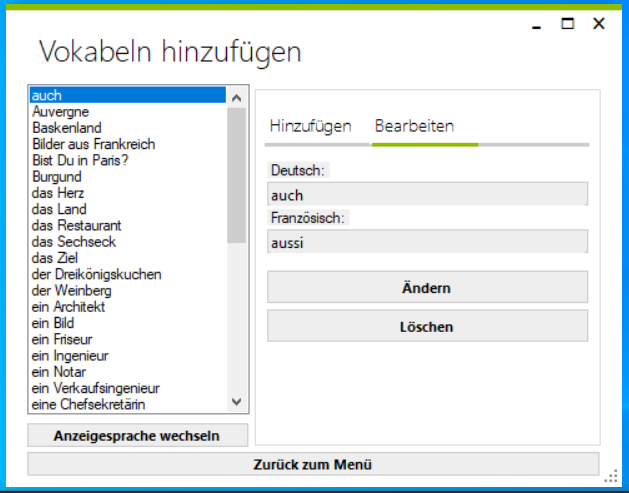
1. Mehrbenutzerfähigkeit durch verschiedene Vokabelbestände

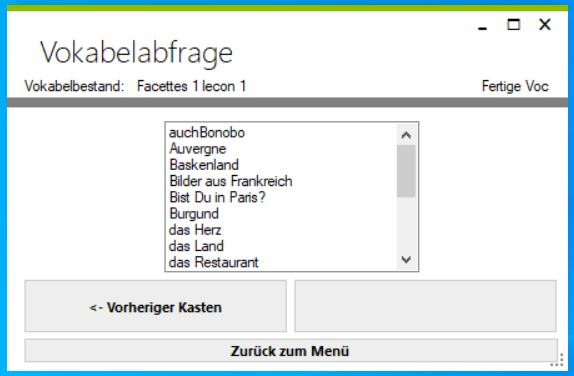
2. Anzeige der fertig gelernten Vokabeln

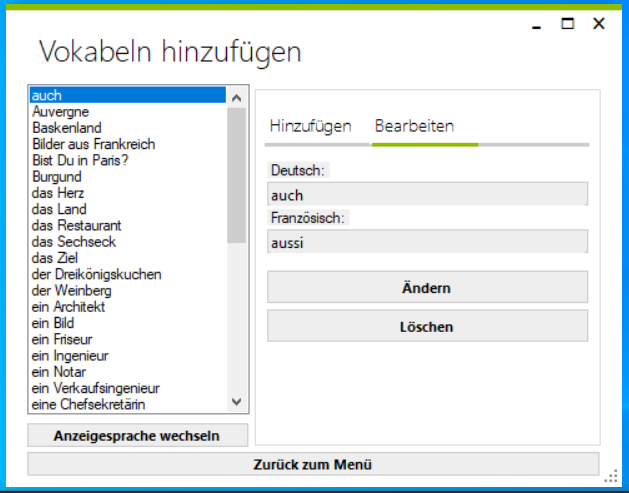
**GUI**



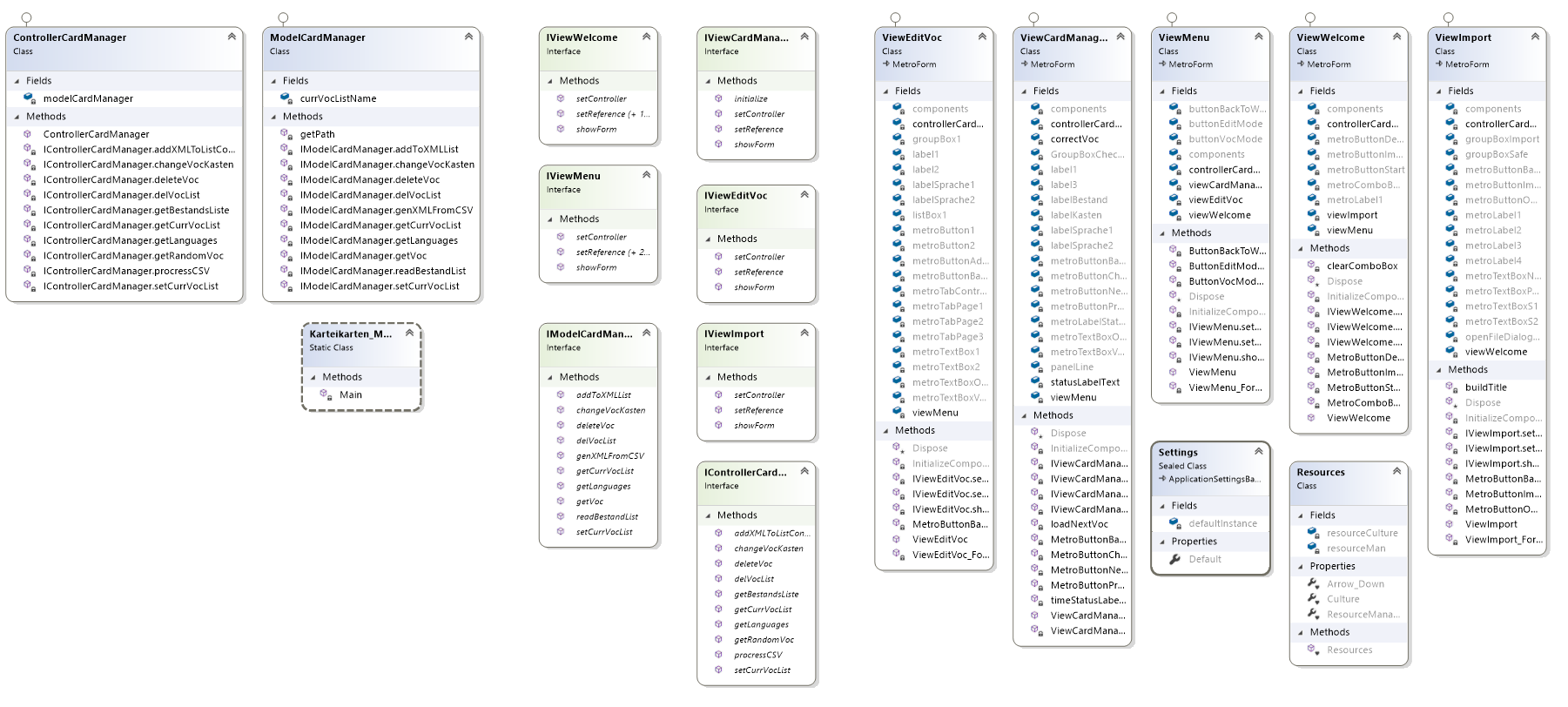
****

****



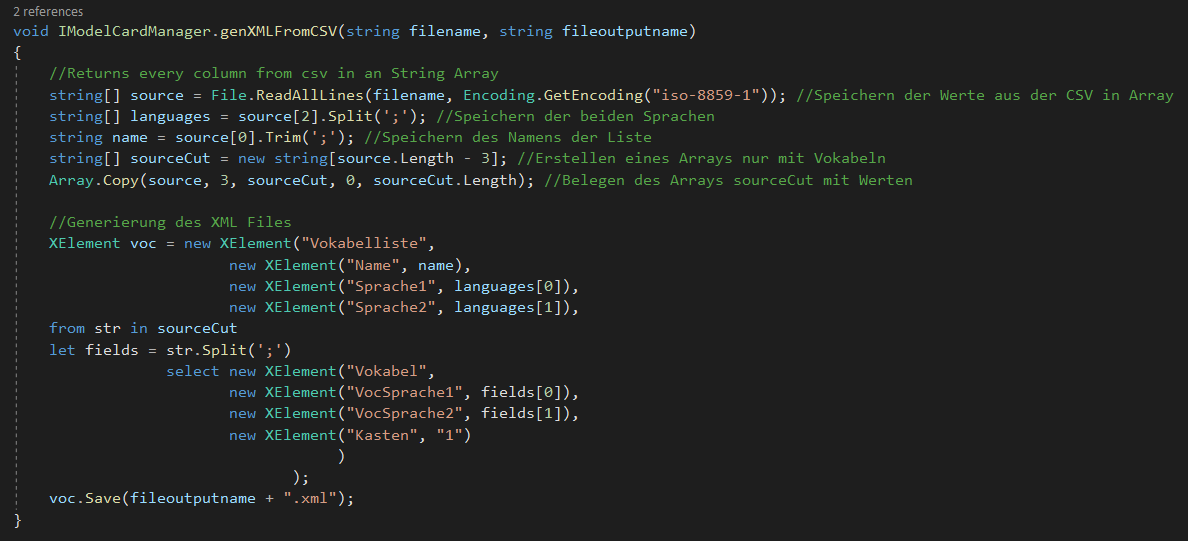


**UML**



**Code**

https://github.com/TheS1mon/Karteikarten\_Manager

**Codebeispiel: CSV Import**

**Bug Report:**

- Bei Eingabe von „\t“ als Dateinamen gibt es einen WindowsIO Error, da Windows keine Dateien damit abspeichern kann (Jedenfalls nicht aus dem Code heraus