QProjektdokumentation

Matteo Jakob

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 15.8.2023 | 0.0.1 | Erste Version | Jakob |
|  | 1.0.0 | Finale Version | Jakob |

# Informieren

## Ihr Projekt

EduConfig – Ein Schulverwaltungssystem

Das Projekt sollte, ist aber nicht darauf beschränkt, ein Schulverwaltungssystem beinhalten, mit welchem man Schüler, Lehrer und selbst definierte Rollen erstellen und «Personen» hinzufügen kann. «Personen» sollten verschiedene Eigenschaften haben wie Name, Geburtsdatum, Klasse (optional) und weiteres.

Es wird mit C++ als CLI oder GUI erstellt.

Die Daten werden Lokal auf einer Datei gespeichert, damit man diese aus jeder Session abrufen kann und sie nicht gelöscht werden.

Zudem sollte das Projekt auf GitHub hochgeladen werden und «Actions» beinhalten.

Falls noch viel Zeit übrig wäre ( mind. 10 Lektionen), werde ich noch das Projekt mit einem Stundenplan-Manager erweitern. Dieser sollte automatisch Stundenpläne für mind 3 Klassen erstellen können.

## Quellen

-

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| Nummer | Muss / Kann? | Funktional? Qualität? Rand? | Beschreibung |
| 1.1 | Muss | Rand | Visual Studio C++ als Programmierumgebung |
| 2.1 | Muss | Funktional | Der Benutzer soll Personen in eine Personen-Liste eintragen können. |
| 3.1 | Muss | Funktional | Der Benutzer soll die Personen-Liste abfragen können |
| 4.1 | Muss | Funktional | Der Benutzer kann eigene Funktionen den Personen hinzufügen (z.B. Lehrer oder Putzdienst) |
| 5.1 | Kann | Funktional | Der Benutzer kann Stundenpläne erstellen |
| 6.1 | Kann | Funktional | Die Stundenpläne können automatisch generiert werden |
| 7.1 | Kann | Funktional | Der Benutzer kann spezifische Details von einer Person ansehen. |
| 8.1 | Muss | Funktional | Die Personen-Liste wird in einer Datei auf dem Computer gespeichert. |
| 9.1 | Kann | Funktional | Die Rollen werden in einer Datei auf dem Computer gespeichert. |

## 1.4 Diagramme

keine

## 1.5 Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| Nummer | Vorbereitung | Eingabe | Erwartete Ausgabe |
| 1.1 | - | Sln öffnen | C++ Projekt |
| 2.1 | Personen-Liste leer | Neue Personendaten (Name, Geburtsdatum, Klasse etc.) | Erfolgreiche Hinzufügung zur Personen-Liste |
| 3.1 | Personen-Liste mit Einträgen | - | Ausgabe der gesamten Personen-Liste |
| 4.1 | Personen-Liste mit Einträgen, Neue Funktion | Person, Funktion | Erfolgreiche Zuweisung der Funktion zur Person |
| 5.1 | - | Daten für Stundenplan (Klassen, Fächer etc.) | Stundenplan erstellt und bestätigt |
| 6.1 | Stundenplan-Manager bereit | Klassen und Parameter | |  | | --- | | Automatisch generierter Stundenplan erstellt | |
| 7.1 | Personen-Liste mit Einträgen | Person auswählen | |  |  | | --- | --- | |  | Anzeige spezifischer Details der ausgewählten Person | |
| 8.1 | Personen-Liste mit Einträgen | - | Personen-Liste erfolgreich in Datei gespeichert |
| 9.1 | Rollen definiert | - | Rollen erfolgreich in Datei gespeichert |

1. Planen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nummer | Frist | Mitglied | Beschreibung | | Zeit (geplant) |
| 0.1 | 15.8 | Matteo | Informieren, Planen, Entscheiden | | 5 |
| 1.1 | 22.8 | Matteo | Code schreiben: Anf. 1.1 | | 1 |
| 2.1 | 22.8 | Matteo | Code schreiben: Anf. 2.1 | | 3 |
| 3.1 | 22.8 | Matteo | Code schreiben: Anf. 3.1 | | 1 |
| 4.1 | 29.8 | Matteo | Code schreiben: Anf. 4.1 | | 5 |
| 5.1 | 5.9 | Matteo | Code schreiben: Anf. 5.1 | | 3 |
| 6.1 | 5.9 | Matteo | Code schreiben: Anf. 6.1 | | 4 |
| 7.1 | 5.9 | Matteo | Code schreiben: Anf. 7.1 | | 3 |
| 8.1 | 12.9 | Matteo | Code schreiben: Anf. 8.1 | | 2 |
| 9.1 | 12.9 | Matteo | Code schreiben: Anf. 9.1 | | 2 |
| 10.1 | 19.9 | Matteo | Testen | | 1 |
| 11.1 | 19.9 | Matteo | Portfolio | | 5 |
|  | | | | TOTAL: | 35 |

1. Entscheiden

Keine

1. Realisieren

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nummer | Datum | Name | Beschreibung | Zeit (geplant) | Zeit (effektiv) |
| 0.1 | 15.8 | Matteo | Informieren, Planen, Entscheiden | 5 | 5 |
| 1.1 | 22.8 | Matteo | Code schreiben: Anf. 1.1 | 1 | 1 |
| 2.1 | 22.8 | Matteo | Code schreiben: Anf. 2.1 | 3 | 5 |
| 3.1 | 22.8 | Matteo | Code schreiben: Anf. 3.1 | 1 | 15 |
| 4.1 | 29.8 | Matteo | Code schreiben: Anf. 4.1 | 5 | 3 |
| 5.1 | 5.9 | Matteo | Code schreiben: Anf. 5.1 | 3 | X |
| 6.1 | 5.9 | Matteo | Code schreiben: Anf. 6.1 | 4 | X |
| 7.1 | 5.9 | Matteo | Code schreiben: Anf. 7.1 | 3 | 5 |
| 8.1 | 12.9 | Matteo | Code schreiben: Anf. 8.1 | 2 | 3 |
| 9.1 | 12.9 | Matteo | Code schreiben: Anf. 9.1 | 2 | 2 |
| 10.1 | 19.9 | Matteo | Testen | 1 | 1 |
| 11.1 | 19.9 | Matteo | Portfolio | 5 | 5 |
|  |  |  | Total | 35 | 45 |

1. Kontrollieren

## **Testprotokoll**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TestfallNr | Nr | Datum | Resultat | Durchgeführt |
| 1 | 1 | 15.10 | **Ok** | Jakob |
| 2 | 1 | 15.10 | **Ok** | Jakob |
| 3 | 1 | 15.10 | **NOk** | Jakob |
| 4 | 1 | 15.10 | **Ok** | Jakob |
| 5 | 1 | 15.10 | **Nok** | Jakob |
| 6 | 1 | 15.10 | **Nok** | Jakob |
| 7 | 1 | 15.10 | **Nok** | Jakob |
| 8 | 1 | 15.10 | **ok** | Jakob |
| 9 | 1 | 15.10 | **ok** | Jakob |
|  |  |  |  |  |

Nicht alle Test verliefen wie erwartet.

Problem lag bei dem lesen der Inputs, es liest die daten nicht korrekt von der Datei und gibt immer Fehler.

Dieses Problem muss behoben werden.

Man kann das Projekt so nicht weitergeben.

Die Tests wurden auf Windows 11 durchgeführt

1. Auswerten

*Portfolio-Eintrag auf Mahara*