Inhaltsverzeichnis

[1. Einführung 3](#_Toc184048517)

[1.1 Projektumfeld 3](#_Toc184048518)

[1.2 Projektziel 3](#_Toc184048519)

[1.3 Projektbegründung 3](#_Toc184048520)

[1.4 Projektschnittstellen 4](#_Toc184048521)

[1.5 Projektabgrenzung 4](#_Toc184048522)

[2. Projektplanung 4](#_Toc184048523)

[2.1.1 Projektphase 4](#_Toc184048524)

[2.1.2 Abweichungen von Projektantrag 4](#_Toc184048525)

[2.1.3 Ressourcenplanung 4](#_Toc184048526)

[3. Analysephase 4](#_Toc184048527)

[3.1 Ist-Analyse 4](#_Toc184048528)

[3.2 Wirtschaftlichkeitsbetrachtung 5](#_Toc184048529)

[3.2.1 Make-or-Buy Entscheidung 5](#_Toc184048530)

[3.2.2 Projektkosten 5](#_Toc184048531)

[3.2.3 Amortisationsdauer 5](#_Toc184048532)

[3.3 Anwendungsfälle: 6](#_Toc184048533)

[3.3.1 Anwendungsfall: sich registrieren lassen 6](#_Toc184048534)

[3.3.2 Anwendungsfall: sich anzumelden 7](#_Toc184048535)

[3.3.3 Anwendungsfall: VCF-Format-Data generieren 7](#_Toc184048536)

[3.3.4 Anwendungsfall: Log-out 7](#_Toc184048537)

[3.4 Qualitätsanforderungen 7](#_Toc184048538)

[4. Entwurf 7](#_Toc184048539)

[4.1 Anwendungsfalldiagramm 7](#_Toc184048540)

[4.2 Programmablaufplan –VCF-Generator 7](#_Toc184048541)

[4.3 GUI-Entwurf & Design 7](#_Toc184048542)

[4.4 Datenstruktur 7](#_Toc184048543)

[4.5 Maßnahmen zur Qualitätssicherung 7](#_Toc184048544)

[5. Durchführung 7](#_Toc184048545)

[5.1 Erstellung der GUI 7](#_Toc184048546)

[5.2 Funktionalität das Programm 7](#_Toc184048547)

[5.2.1 Funktionalität Benutzerregistrierung 7](#_Toc184048548)

[5.2.2 Funktionalität Benutzerauthentifizierung 7](#_Toc184048549)

[5.2.3 Funktionalität VCF-Datei erstellen und herunterzuladen 7](#_Toc184048550)

[5.3 Exception Handling 7](#_Toc184048551)

[6. Test & Abnahme 7](#_Toc184048552)

[7. Fazit 7](#_Toc184048553)

[7.1 Soll-/Ist-Vergleich 7](#_Toc184048554)

[7.1.1 Validierung 7](#_Toc184048555)

[7.1.2 Fehlerbehandlung 7](#_Toc184048556)

[7.1.3 Tests 7](#_Toc184048557)

[7.2 Zeitabgleich 7](#_Toc184048558)

[7.3 Ausdruck 7](#_Toc184048559)

[8. Glossar 7](#_Toc184048560)

[9. Anhang 7](#_Toc184048561)

# Einführung

## Projektumfeld

Der VCF-Generator wurde als Webanwendung entwickelt und nutzt PHP, MySQL und HTML. Das Projekt ist für Nutzer, die eine einfache Methode zur Erstellung von VCF-Dateien für ihre Kontakte benötigen.

## Projektziel

Das Ziel ist die Entwicklung einer benutzerfreundlichen Webanwendung, die registrierten Nutzern erlaubt, VCF-Dateien zu erstellen und herunterzuladen.

## Projektbegründung

VCF-Dateien sind eine Standardmethode zur Speicherung und Weitergabe von Kontaktdaten. Diese Anwendung vereinfacht den Prozess der Erstellung solcher Dateien.

## Projektschnittstellen

In dem Projekt sind zwei Schnittstellen gegeben. Zu den personellen Schnittstellen zählen unter anderem der Lehrer, als Kunde und Ich als Auftragnehmer.

## Projektabgrenzung

Das Projekt beschränkt sich auf die Funktionen der Registrierung, Anmeldung und VCF-Generierung, ohne komplexe Verwaltungs- oder Berechtigungssysteme.

# Projektplanung

## Projektphase

Das Projekt wurde in mehrere Phasen unterteilt. In der Analyse- und Planungsphase werden alle benötigten Analysen, Diagramme und Entwürfe erstellt. Hier wird beispielsweise die Ist-/Soll-Analyse und die Wirtschaftlichkeitsbetrachtung durchgeführt. In der Durchführungsphase werden zunächst Informationen (zu den Programmbibliotheken) beschafft. Dann werden die einzelnen Anforderungen programmiert. Nach der Durchführungsphase werden noch Unit- und Black-Box Tests durchgeführt. Zuletzt wird die Übernahme/Abnahme durchgeführt und es wird eine Dokumentation erstellt. Hier wird die Zeitplanung mit ihren Phasen tabellarisch dargestellt.

## Abweichungen von Projektantrag

Es wurden vereinfachte Funktionen ohne erweiterte Nutzerrollen implementiert.

## Ressourcenplanung

Für die Entwicklung wurde ein Windows 10 Rechner mit zwei Bildschirmen genutzt. Verwendete Programme waren Visual Studio Code als IDE mit der Programmiersprache PHP, der Google Chrome Browser (für Recherchen), die Webseite „draw.io“ für die Erstellung von Diagrammen (Programmablaufplan), das Programm „Draw.io“ für den Entwurf der GUI und Microsoft Word für die Erstellung der notwendigen Dokumente wie Projektdokumentation.

# Analysephase

## Ist-Analyse

Vor Beginn des Projekts war die Organisation von Kontaktdaten ein zeitaufwändiger und ineffizienter Prozess. Nutzer mussten Kontaktdaten manuell in verschiedenen Formaten pflegen, was oft zu Inkonsistenzen und Fehlern führte. Besonders problematisch war, dass es keine einfache Möglichkeit gab, Kontaktdaten schnell in ein einheitliches und universelles Format, wie das VCF-Format (vCard), zu exportieren. Dies führte dazu, dass der Austausch von Kontaktdaten zwischen Geräten und Plattformen häufig Schwierigkeiten bereitete.

Zusätzlich fehlte eine zentrale Plattform, über die Nutzer ihre Daten in einem strukturierten und nutzerfreundlichen Workflow verwalten konnten. Der Prozess war fragmentiert, wobei unterschiedliche Tools und Anwendungen erforderlich waren, die keine nahtlose Integration boten.

Die Sicherheitsanforderungen für sensible Kontaktdaten waren ebenfalls unzureichend. Es bestand die Gefahr, dass unverschlüsselte Daten leicht kompromittiert werden könnten, was speziell bei geschäftlichen Kontakten problematisch war.

Ein weiterer Schwachpunkt war die fehlende Möglichkeit, die Anwendung auf verschiedenen Geräten (z. B. Desktops, Tablets, Smartphones) responsiv und benutzerfreundlich zu nutzen. Viele bestehende Lösungen hatten ein veraltetes Design und waren nicht auf die Bedürfnisse einer modernen, mobilen Benutzergruppe ausgelegt.

Die Zielgruppe bestand hauptsächlich aus Nutzern, die regelmäßig Kontakte austauschen müssen, wie z. B. Geschäftsleute, Veranstaltungsorganisatoren oder Vertriebsteams. Es zeigte sich, dass diese Nutzergruppen eine einfache, sichere und schnelle Möglichkeit benötigen, ihre Kontakte in einem standardisierten Format zu generieren, herunterzuladen und zu teilen.

Eine technische Analyse ergab, dass die bestehende Infrastruktur der Zielgruppe nicht optimal auf die Integration eines solchen Tools ausgerichtet war. Viele Nutzer arbeiteten mit veralteter Software oder hatten keinen Zugriff auf cloudbasierte Lösungen, was die Nachfrage nach einer lokalen und leicht zugänglichen Anwendung erhöhte.

## Wirtschaftlichkeitsbetrachtung

## Make-or-Buy Entscheidung

Durch Internetrecherchen habe ich herausgefunden, dass es kein Programm gibt, welches die Anforderungen des Projektes eindeutig abdeckt. Aufgrund dessen gibt es hier keine andere Möglichkeit, als die Anwendung selbstständig zu programmieren.

## Projektkosten

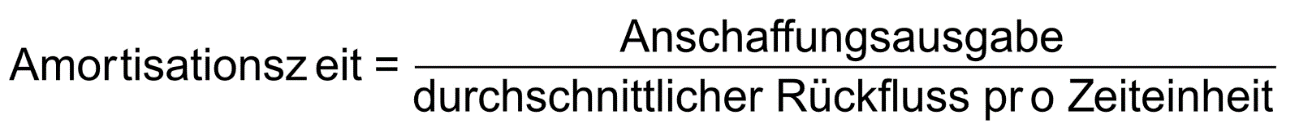
Da die Anwendung in einem alleinstehenden Projekt erstellt wird, fallen hier nur einmalige Kosten an. Da das Tool auf einem bestehenden System eingesetzt wird, fallen hier keine zusätzlichen Hardwarekosten an.

Rechnung der Projektkosten:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Personal** | **Hinweis** | **Kosten/h** | **Stunden** | **Kosten gesamt** |
| **Auszubildender** | *Erstellung der*  *Software* | *40 $* | *80* | *3200 $* |

Die Berechnung ergibt, dass sich die Projektkosten einmalig auf 3200 $ belaufen.

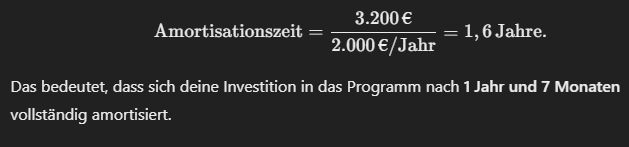
## Amortisationsdauer

  
Investitionskosten (Anschaffungsausgabe) = Arbeitsstunden × Stundensatz

80Stunden×40€/Stunde=**3.200€**

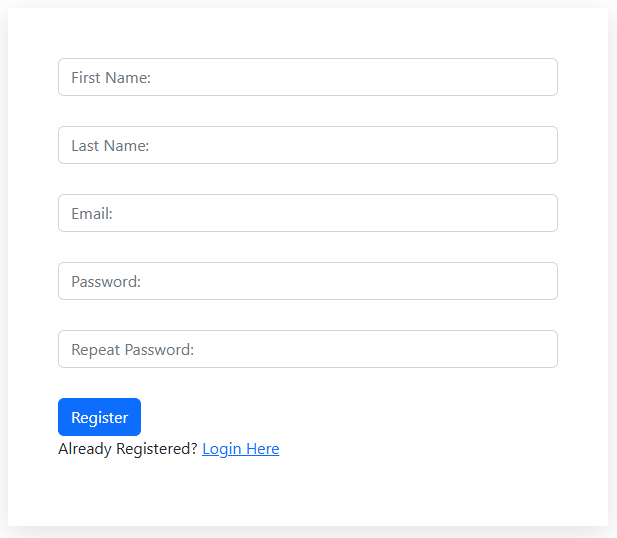
Jährlichen Einsparungen oder Einnahmen

50Stunden×40€/Stunde=2.000€ jährliche Einsparungen



## Anwendungsfälle:

## Anwendungsfall: sich registrieren lassen

* Nutzer ruft die Registrierungsseite auf und füllt die erforderliches Felder aus, einschließlich:
  + Vorname, Nachname, E-Mail-Adresse, Password und Password-Bestätigung.
* Das System überprüft die Eingaben mit folgenden Kriterien:
  + Alle Felder müssen ausgefüllt sein, Sie E-Mail-Adresse wird auf ein gültiges Format geprüft.
  + Das Password muss mindestens eine Zahl, einen Groß- und Kleinbuchstaben und mindestens 8 oder mehr Zeichen enthalten. Die Passwörter müssen übereinstimmen. Wenn die Validierung erfolgreich ist: Der Benutzer wird in der Datenbank gespeichert und Das Password wird verschlüsselt.
  + 

## Anwendungsfall: sich anzumelden

## Anwendungsfall: VCF-Format-Data generieren

## Anwendungsfall: Log-out

## Qualitätsanforderungen

# Entwurf

## Anwendungsfalldiagramm

## Programmablaufplan –VCF-Generator

## GUI-Entwurf & Design

## Datenstruktur

## Maßnahmen zur Qualitätssicherung

# Durchführung

## Erstellung der GUI

## Funktionalität das Programm

## Funktionalität Benutzerregistrierung

## Funktionalität Benutzerauthentifizierung

## Funktionalität VCF-Datei erstellen und herunterzuladen

## Exception Handling

# Test & Abnahme

# Fazit

## Soll-/Ist-Vergleich

## Validierung

## Fehlerbehandlung

## Tests

## Zeitabgleich

## Ausdruck

# Glossar

# Anhang