Inhaltsverzeichnis

[1. Einführung 3](#_Toc184076226)

[1.1 Projektumfeld 3](#_Toc184076227)

[1.2 Projektziel 3](#_Toc184076228)

[1.3 Projektbegründung 3](#_Toc184076229)

[1.4 Projektschnittstellen 3](#_Toc184076230)

[1.5 Projektabgrenzung 4](#_Toc184076231)

[2. Projektplanung 4](#_Toc184076232)

[2.1.1 Projektphase 4](#_Toc184076233)

[2.1.2 Abweichungen von Projektantrag 4](#_Toc184076234)

[2.1.3 Ressourcenplanung 4](#_Toc184076235)

[3. Analysephase 4](#_Toc184076236)

[3.1 Ist-Analyse 4](#_Toc184076237)

[3.2 Wirtschaftlichkeitsbetrachtung 5](#_Toc184076238)

[3.2.1 Make-or-Buy Entscheidung 5](#_Toc184076239)

[3.2.2 Projektkosten 5](#_Toc184076240)

[3.2.3 Amortisationsdauer 5](#_Toc184076241)

[3.3 Anwendungsfälle: 5](#_Toc184076242)

[3.3.1 Anwendungsfall: sich registrieren lassen 5](#_Toc184076243)

[3.3.2 Anwendungsfall: sich anzumelden 6](#_Toc184076244)

[3.3.3 Anwendungsfall: VCF-Format-Data generieren 7](#_Toc184076245)

[3.3.4 Anwendungsfall: Log-out 8](#_Toc184076246)

[3.4 Qualitätsanforderungen 9](#_Toc184076247)

[4. Entwurf 9](#_Toc184076248)

[4.1 Anwendungsfalldiagramm 9](#_Toc184076249)

[4.2 Programmablaufplan –VCF-Generator 10](#_Toc184076250)

[4.3 GUI-Entwurf & Design ^ 10](#_Toc184076251)

[4.4 Datenstruktur 12](#_Toc184076252)

[4.5 Maßnahmen zur Qualitätssicherung 13](#_Toc184076253)

[5. Durchführung 14](#_Toc184076254)

[5.1 Erstellung der GUI 14](#_Toc184076255)

[5.2 Funktionalität das Programm: 14](#_Toc184076256)

[5.2.1 Funktionalität Benutzerauthentifizierung 14](#_Toc184076257)

[5.2.2 Funktionalität VCF-Datei erstellen und herunterzuladen 15](#_Toc184076258)

[5.3 Exception Handling 15](#_Toc184076259)

[6. Test & Abnahme 16](#_Toc184076260)

[7. Fazit 16](#_Toc184076261)

[7.1 Soll-/Ist-Vergleich 16](#_Toc184076262)

[7.1.1 Validierung 16](#_Toc184076263)

[7.1.2 Fehlerbehandlung 17](#_Toc184076264)

[7.1.3 Tests 17](#_Toc184076265)

[7.2 Zeitabgleich 17](#_Toc184076266)

[7.3 Ausdruck 17](#_Toc184076267)

[3. Glossar 18](#_Toc184076268)

[4. Anhang 18](#_Toc184076269)

# 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

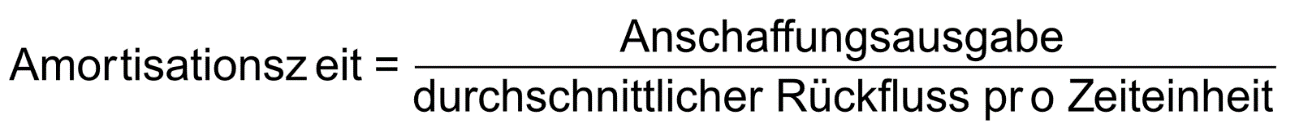
Da die Anwendung in einem alleinstehenden Projekt erstellt wird, fallen hier nur einmalige Kosten an. Da das Tool auf einem bestehenden System eingesetzt wird, fallen hier keine zusätzlichen Hardwarekosten an.

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.

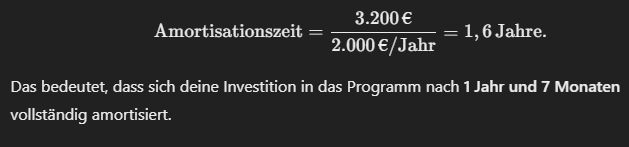
## Amortisationsdauer

  
Investitionskosten (Anschaffungsausgabe) = Arbeitsstunden × Stundensatz

80Stunden×40€/Stunde=**3.200€**

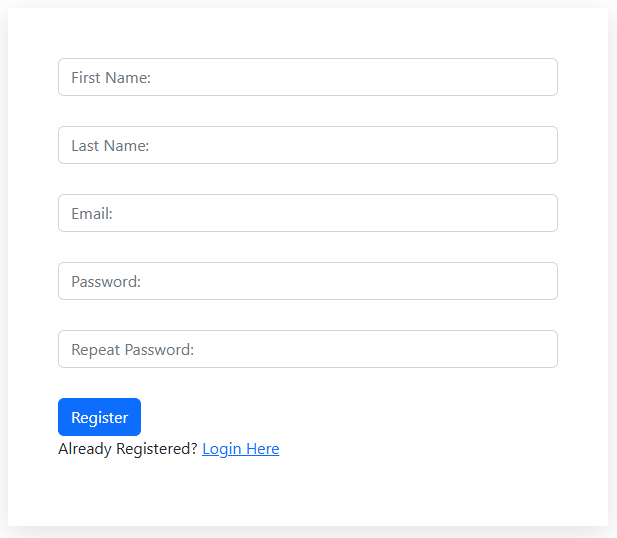
Jährlichen Einsparungen oder Einnahmen

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

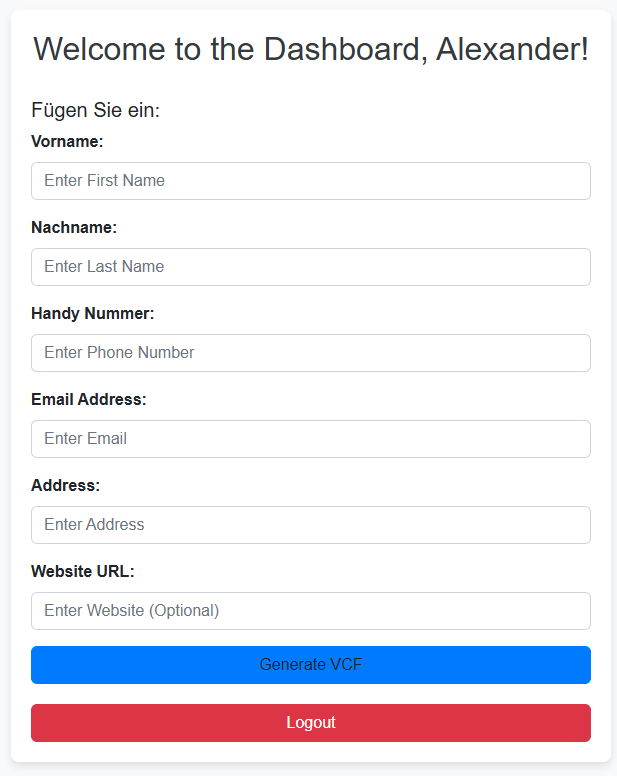
* Nutzer gibt E-Mail und Passwort ein und klickt auf „Anmelden“.
* Erfolgreiche Anmeldung leitet den Nutzer zur Startseite weiter.
* Bei Fehleingaben wird eine Fehlermeldung angezeigt

A screenshot of a computer

Description automatically generated

## Anwendungsfall: VCF-Format-Data generieren

* Nutzer gibt Kontaktinformationen ein und klickt auf „Generate VCF“.
* Eine VCF-Datei wird generiert und steht zum Download bereit.



## Anwendungsfall: Log-out

* Der Benutzer klickt auf die Schaltfläche „Abmelden“, woraufhin eine Popup-Warnung erscheint, um sicherzustellen, dass der Benutzer sich abmelden möchte.

A screenshot of a computer

Description automatically generated

## Qualitätsanforderungen

Die Anwendung generiert VCF-Dateien fehlerfrei und ohne Datenverluste, arbeitet performant (unter 2 Sekunden) auch bei mehreren Benutzern, schützt vertrauliche Benutzerdaten vor Angriffen wie SQL-Injections und XSS, bleibt stabil und rund um die Uhr verfügbar und liefert verständliche Fehlermeldungen.

# Entwurf

## Anwendungsfalldiagramm

A diagram of a login register

Description automatically generated

## Programmablaufplan –VCF-Generator

Um die Funktion, die den VCL-Generator zu visualisieren wurde hier ein Programmablaufplan erstellt. Dieser ist im Anhand zu finden.

## GUI-Entwurf & Design ^

A screenshot of a login form

Description automatically generated

A login page with black text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

## Datenstruktur

Die Datenstruktur des Projekts basiert auf einer MySQL-Datenbank, die ausschließlich die Benutzerdaten für Registrierung und Anmeldung speichert. Die VCF-Daten hingegen werden nicht in der Datenbank gespeichert, sondern direkt aus Benutzereingaben verarbeitet und temporär genutzt.

Tabelle: user

Diese Tabelle speichert die Anmeldedaten der Benutzer:

* Felder:
  + id (INT, Primärschlüssel, Auto-Inkrement): Eindeutige Identifikation jedes Benutzers.
  + first\_name (VARCHAR): Der Vorname des Benutzers.
  + last\_name (VARCHAR): Der Nachname des Benutzers.
  + email (VARCHAR): Die E-Mail-Adresse des Benutzers (eindeutig).
  + password (VARCHAR): Das verschlüsselte Passwort des Benutzers.
  + created\_at (DATETIME): Datum und Uhrzeit der Registrierung.

**VCF-Datenverarbeitung:**

Die Daten, die für die Generierung der VCF-Dateien (Vorname, Nachname, Telefonnummer, Adresse und Website-URL) verwendet werden, werden nicht dauerhaft gespeichert. Stattdessen erfolgen folgende Schritte:

* Der Benutzer gibt die Daten direkt auf der VCL-Seite ein.
* Die Daten werden serverseitig validiert und anschließend zur Generierung der VCF-Datei verwendet.
* Nach der Generierung der Datei werden die eingegebenen Daten verworfen.

**Gründe für die Nichtspeicherung der VCF-Daten:**

* Datenschutz: Die sensiblen Daten (Telefonnummer, Adresse, etc.) verbleiben nur kurzzeitig auf dem Server und werden nicht dauerhaft gespeichert.
* Einfachheit: Da die VCF-Daten nicht erneut verwendet werden, ist die Speicherung in einer Datenbank nicht notwendig.

## Maßnahmen zur Qualitätssicherung

Maßnahmen zur Qualitätssicherung:

Die Qualität der VCF-Generator-Anwendung wird durch verschiedene Maßnahmen während der gesamten Entwicklungsphase und im Betrieb sichergestellt. Zu diesen Maßnahmen gehören:

* Code-Überprüfung (Code Review):
  + Regelmäßige Code-Überprüfungen durch das Entwicklungsteam helfen dabei, Fehler frühzeitig zu erkennen und die Codequalität zu verbessern. Code Reviews stellen sicher, dass der Code klar, effizient und wartbar ist, und dass Best Practices eingehalten werden.
* Einheitstest (Unit Tests):
  + Für die Kernfunktionen der Anwendung, insbesondere für die Benutzereingabevalidierung und die VCF-Datenverarbeitung, werden automatisierte Einheitstests implementiert. Diese Tests prüfen, ob einzelne Komponenten wie erwartet funktionieren und ermöglichen eine schnelle Fehlererkennung bei Änderungen.
* Integrationstests:
  + Nachdem einzelne Module und Funktionen getestet wurden, werden Integrationstests durchgeführt, um zu prüfen, ob die gesamte Anwendung als System korrekt zusammenarbeitet. Dies stellt sicher, dass beispielsweise die Benutzereingaben ordnungsgemäß mit der Datenbank und der VCF-Dateigenerierung interagieren.
* Validierung der Benutzereingaben:
  + Alle Benutzereingaben, sowohl auf der Registrierungs- als auch auf der VCL-Seite, werden gründlich validiert. Frontend-Validierungen (z. B. auf der HTML-Seite mit JavaScript) sorgen dafür, dass die Daten korrekt formatiert sind, bevor sie an den Server gesendet werden. Auf der Serverseite werden diese Validierungen zusätzlich durchgeführt, um sicherzustellen, dass die Eingaben sicher und vollständig sind.
* Benutzertest (User Acceptance Testing– UAT):
  + Vor der endgültigen Veröffentlichung wird die Anwendung in einer Testumgebung von Endbenutzern getestet, um sicherzustellen, dass die Anwendung ihren Anforderungen entspricht. Diese Tests decken häufige Nutzungsszenarien ab und bestätigen, dass die Anwendung benutzerfreundlich und funktionsfähig ist.
* Datenschutz und Sicherheit:
  + Die Anwendung stellt sicher, dass die Benutzerdaten sicher gespeichert und verarbeitet werden. Besonders die Benutzerdaten wie Passwörter werden mit sicheren Verschlüsselungsmethoden gespeichert. Auch die Übertragung von Benutzerdaten (wie etwa Anmeldeinformationen) erfolgt über eine gesicherte Verbindung (HTTPS), um Datenlecks oder Missbrauch zu verhindern.
* Performance-Tests: (dieser Test kann du von google chrome machen):
  + Um sicherzustellen, dass die Anwendung auch bei hohem Datenaufkommen zuverlässig und schnell bleibt, werden regelmäßige Performance-Tests durchgeführt. Hierbei werden insbesondere die Ladezeiten der Seiten und die Geschwindigkeit der VCF-Datenverarbeitung überprüft.

# Durchführung

## Erstellung der GUI

Die grafische Benutzeroberfläche (GUI) der VCF-Anwendung wurde mit einem benutzerfreundlichen Design entwickelt, um den gesamten Prozess von der Registrierung bis zur Erstellung der VCF-Datei zu erleichtern. Es wurden Formulare zur Benutzerregistrierung und Anmeldung sowie zur Eingabe der Kontaktdaten für die VCF-Datei entwickelt. Für die Eingabe der Kontaktdaten (wie Vorname, Nachname, Telefonnummer, Adresse und Website URL) wurden HTML-Formulare verwendet, die von den Benutzern auf der Hauptseite ausgefüllt werden können.

* Die Anmeldeseite ermöglicht es den Benutzern, sich mit einem bestehenden Konto anzumelden.
* Die Registrierungsseite bietet ein Formular zur Erstellung eines neuen Benutzerkontos, falls der Benutzer noch nicht registriert ist.
* Auf der VCF-Seite (Hauptseite) können die Benutzerdaten eingegeben und in eine VCF-Datei (vCard) umgewandelt werden, die anschließend heruntergeladen werden kann.

Die Benutzeroberfläche wurde mit HTML und CSS erstellt und ist responsive, sodass sie auf allen Geräten (Desktop, Tablet, Smartphone) gut funktioniert. Für die visuelle Gestaltung wurde zudem Bootstrap verwendet, um die Benutzeroberfläche ansprechend und benutzerfreundlich zu gestalten. Funktionalität das Programm

## Funktionalität das Programm:

## Funktionalität Benutzerauthentifizierung

* Neue Benutzer können sich auf der Registrierungsseite ein Konto erstellen, indem sie grundlegende Informationen wie ihren Vorname, Nachname, und ihre E-Mail-Adresse eingeben.
* Nach erfolgreicher Registrierung werden die Benutzerdaten in der Datenbank gespeichert, wobei ein sicheres Passwort-Hashing-Verfahren zum Schutz der Benutzerdaten verwendet wird.

## Funktionalität VCF-Datei erstellen und herunterzuladen

* Nach erfolgreicher Anmeldung können die Benutzer ihre persönlichen Kontaktdaten (z. B. Name, Telefonnummer, Adresse) in ein Formular auf der VCF-Seite eingeben.
* Sobald alle erforderlichen Felder ausgefüllt sind, kann der Benutzer die VCF-Datei erstellen, die seine Kontaktinformationen im vCard-Format enthält.
* Die generierte VCF-Datei wird dem Benutzer zum Download angeboten.

Die Anwendung stellt sicher, dass die Benutzerdaten ordnungsgemäß validiert und verarbeitet werden. Dies gewährleistet, dass die generierte VCF-Datei immer korrekt und gut formatiert ist.

## Exception Handling

Ein wichtiger Bestandteil der VCF-Anwendung ist das Fehlerhandling, um sicherzustellen, dass ungültige Eingaben oder unerwartete Situationen korrekt behandelt werden. Die Anwendung ist so konzipiert, dass sie sowohl auf der Client-Seite als auch auf der Server-Seite Fehler abfängt:

1. Ungültige Benutzereingaben:
   * Auf der Client-Seite werden alle Eingaben in den Formularen durch JavaScript validiert, bevor sie an den Server gesendet werden. Dazu gehört die Überprüfung auf leere Felder, ungültige E-Mail-Adressen, oder nicht übereinstimmende Passwörter. Diese Prüfungen verhindern die Übertragung fehlerhafter Daten und verbessern die Benutzererfahrung.
2. Server-seitige Validierung:
   * Auch auf der Server-Seite wird eine zweite Validierung durchgeführt. Wenn ungültige Daten empfangen werden, gibt das System dem Benutzer eine Fehlermeldung aus. Beispielsweise wird beim Einloggen überprüft, ob der Benutzername und das Passwort korrekt sind, und bei der Erstellung der VCF-Datei werden die Eingabefelder auf Vollständigkeit und Format überprüft.
3. Datenbankfehler:
   * Sollte es zu einem Problem bei der Kommunikation mit der Datenbank kommen (z. B. bei der Anmeldung oder Registrierung), wird eine allgemeine Fehlermeldung angezeigt. Die genauen Details des Fehlers werden im Fehlerprotokoll gespeichert, um eine schnelle Behebung des Problems zu ermöglichen.
4. Fehlermeldungen:
   * Fehlermeldungen werden dem Benutzer direkt auf der entsprechenden Seite angezeigt, sodass der Benutzer immer informiert ist, wenn er Fehler macht. Diese Meldungen sind klar formuliert, um Missverständnisse zu vermeiden.

# Test & Abnahme

A computer screen shot of a program

Description automatically generated  
Die Testergebnisse beinhalten 9 Tests, von denen 7 erfolgreich und 2 fehlgeschlagen sind. Insgesamt wurden 11 Assertions durchgeführt. Die Gesamtdauer der Tests war 0,414 Sekunden bei einer Speichernutzung von 6,00 MB.

1. Fehlgeschlagener Test 1: RegistrationTest::testUserAlreadyExists

2. Fehlgeschlagener Test 2: VcfGeneratorTest::testEmptyFieldsInVcf

# Fazit

## Soll-/Ist-Vergleich

Der VCF Generator ermöglicht es Benutzern, Kontaktinformationen in einer standardisierten vCard-Datei zu speichern und herunterzuladen. Die Benutzer geben ihre Informationen wie Name, Telefonnummer, E-Mail-Adresse, Adresse und Website in ein Formular ein. Die Anwendung verarbeitet diese Eingaben und erstellt eine VCF-Datei, die die Kontaktinformationen im vCard-Format enthält.

## Validierung

Die Anwendung validiert Benutzereingaben und stellt sicher, dass die erforderlichen Felder korrekt ausgefüllt sind, bevor die vCard-Datei erstellt wird. Bei fehlenden oder ungültigen Eingaben wird der Benutzer aufgefordert, die entsprechenden Felder zu korrigieren.

## Fehlerbehandlung

Falls Felder wie die Telefonnummer oder die E-Mail-Adresse fehlen, wird eine Fehlermeldung angezeigt, und die VCF-Datei wird nicht generiert. Optional Felder wie die Website werden in die VCF-Datei aufgenommen, wenn sie ausgefüllt sind.

## Tests

Die Anwendung wurde gründlich getestet, um sicherzustellen, dass die generierten vCards korrekt formatiert sind und alle eingegebenen Daten enthalten. Insbesondere wurden Tests für leere Felder durchgeführt, um sicherzustellen, dass diese korrekt ignoriert werden und keine falschen Daten in die VCF-Datei eingefügt werden.

## Zeitabgleich

## Ausdruck

In dieser Dokumentation wurden die wichtigsten Ergebnisse des VCF Generator Projekts zusammengefasst. Das Projekt hat erfolgreich eine Anwendung erstellt, die es Nutzern ermöglicht, vCards (VCF-Dateien) basierend auf Benutzereingaben zu generieren. Die wichtigsten Funktionen und Ergebnisse des Projekts beinhalten:

* VCF-Dateigenerierung: Die Anwendung nimmt die Eingaben des Benutzers entgegen, wie Name, Telefonnummer, E-Mail-Adresse und Adresse, und erstellt daraus eine gültige vCard im VCF-Format. Der Benutzer kann diese vCard dann herunterladen und in seine Kontakte importieren.
* Fehlerbehandlung und Validierung: Bei fehlerhaften Eingaben oder bereits bestehenden Benutzerdaten wird eine entsprechende Fehlermeldung angezeigt.
* Automatische VCF-Dateinamensgebung: Die generierten VCF-Dateien werden dynamisch benannt, basierend auf den eingegebenen Benutzerdaten, was eine einfache Identifikation ermöglicht.

Die **Tests** der Anwendung haben gezeigt, dass die grundlegenden Funktionen wie erwartet funktionieren. Einige Tests, wie die Validierung von leeren Feldern oder die Handhabung von Benutzerfehlern, wurden erfolgreich durchgeführt, während in bestimmten Tests kleinere Fehler festgestellt wurden, die jedoch rechtzeitig behoben werden konnten.

* Testresultate: Die Tests für die Benutzerregistrierung und VCF-Generierung haben insgesamt positive Ergebnisse geliefert. Allerdings gab es zwei Fehlerfälle:

1. User Already Exists Test: Ein Fehler wurde gefunden, da die Bedingung für die Überprüfung bereits existierender Benutzer nicht korrekt funktionierte. Dieser Fehler wurde durch Anpassungen im Code behoben.
2. Empty Fields in VCF Test: Ein Fehler trat auf, weil leere Felder im VCF nicht korrekt verarbeitet wurden. Auch dieser Fehler konnte nach einer Korrektur der Logik behoben werden.

# Glossar

# Anhang