Schulprojekt 2018

Fachinformatiker Anwendungsentwicklung

Entwicklung eines Tools zur Erfassung und Verwaltung von Zitaten

Projektteam:

Lars Baumgarten

Pascal Schmotz

Tim Rumrich

Auftraggeber:

Peter Grüning

Brühlwiesenschule

Inhaltsverzeichnis

Inhaltsverzeichnis I
Tabellenverzeichnis II
Abbildungsverzeichnis III
Glossar IV
1 Einleitung 1
1.1 Projektbeschreibung 1
1.2 Projektumfeld 1
1.3 Projektziel 1
1.4 Projektbegründung 1
2 Projektplanung 2
2.1 Projektphasen 2
2.2 Ressourcenplanung 2
2.3 Entwicklungsprozess 2
3 Analysephase 3
3.1 Ist-Analyse 3
3.2.1 Make-Or-Buy Entscheidung 3
3.2.2 Projektkosten 3
3.3 Lastenheft/Fachkonzept 4
4 Entwurfsphase 4
4.1 Zielplattform 4
4.2 Architekturdesign 4
4.3 Entwurf der Benutzeroberflächen 4
4.4 Maßnahmen zur Qualitätssicherung 4
4.5 Pflichtenheft 5
5 Implementierungsphase 5
5.1 Implementierung der Benutzeroberflächen 5
5.2 Implementierung der Backendlogik 5
5.3 Aufsetzen der Datenbank 5
6 Abnahmephase 5
7 Fazit 6
7.1 Soll-/Ist-Vergleich 6
7.2 Ausblick 6
Anhang 7

Tabellenverzeichnis

Table 1: Grobe Zeitplanung 2
Table 2: Kostentabelle 3

Abbildungsverzeichnis

Figure 1: Klassendiagramm 8
Figure 2:MVC Beispielhafte Darstellung..... 11

Glossar

API (Application Programming Interface)

Eine Programmierschnittstelle einer Software ist ein Programmteil, der zur Anbindung anderer Programme an das System zur Verfügung gestellt wird.

BWS

Die Berufsschule Brühlwiesenschule in Hofheim am Taunus.
Ein Hort der Kompetenz und der Freude.

Config File (Configuration File /Konfigurationsdatei)

Eine Textdatei die Konfigurationsdaten für Software bereitstellt.

CSV (Comma Separated Values)

Ein Dateiformat für Textdateien zur Speicherung oder zum Austausch einfach strukturierter Daten. In CSV-Dateien können beispielsweise Tabellen abgebildet werden.

GUI (Graphical User Interface)

Eine grafische Benutzeroberfläche bildet die Schnittstelle für Benutzer zur Interaktion mit der Anwendungssoftware.

MVC (Model View Controller)

Ein Architekturmuster zur Eingliederung von Software in die Kategorien Model(Daten/Logik), View(Ansicht/Oberfläche) und Controller (Interaktion zwischen Oberfläche und Logik), mit Ziel eines flexiblen Programmentwurfes um spätere Wiederverwendbarkeit oder Änderbarkeit zu erleichtern. Detaillierte Erklärung im Anhang.

1 Einleitung

Die nachfolgende Projektdokumentation beschreibt die Umsetzung des Schulprojektes im Rahmen der schulischen Ausbildung zum Fachinformatiker Anwendungsentwicklung von Lars Baumgarten, Pascal Schmotz und Tim Rumrich.

1.1 Projektbeschreibung

Das Projekt wurde als eines von vier Wahlthemen zur Verfügung gestellt und soll vom Projektteam als Übung für die Abschlussprüfung dienen. Gewählt wurde eine Zitatverwaltung, mit welcher Zitate, die in der Schule gefallen sind, abzuspeichern und zu sammeln, sodass die Schüler nach Abschluss des Schuljahres ein Andenken an die erquickliche Zeit haben.

1.2 Projektumfeld

Das Projekt wurde in den Lehrräumen der BWS umgesetzt. Hier stand ein Lehrer für Fragen zur Verfügung.

1.3 Projektziel

Das Projekt hat als primäres Ziel, eine komfortable, digitale Verwaltung von Zitaten und deren Aufnahme in eine Datenbank zu realisieren. Es soll Schülern möglich sein, Zitate zentral zu erfassen und verwalten und sie gegebenenfalls zu exportieren, damit Worthülsen und epische Ansprachen von Lehrern nicht vergessen gehen. Sekundäres Ziel ist die Vorbereitung der Schüler auf ihre jeweils eigenen Abschlussprüfungen im Rahmen der Ausbildung durch eine Simulation.

1.4 Projektbegründung

Das Projekt wurde als Wunsch einiger Schüler geäußert und wurde dann an die Anwendungsentwickler Klasse weitergegeben, welche sich großzügiger Weise und aus gegebenem Anlass der Aufgabe verschrieben hat.
Weiterhin war die Durchführung eines Projektes vom Klassenlehrer fest eingeplant, was an dieser Stelle als Begründung genügen soll.

2 Projektplanung

2.1 Projektphasen

Für die Umsetzung des Projektes standen 70 Stunden zur Verfügung. Zu Beginn des Projektes wurden diese auf verschiedene Phasen aufgeteilt, die während des Projektes durchlaufen werden.

Grobe Zeitplanung der Phasen	
Projektphase	**Zeitplanung**
Analysephase	16 h
Entwurfsphase	24 h
Implementierungsphase	72 h
Dokumentation	8 h
Gesamt	**120 h**

Table 1: Grobe Zeitplanung

2.2 Ressourcenplanung

In der Ressourcenübersicht sind alle Ressourcen aufgelistet, die für das Projekt verwendet wurden. Damit sind Hard- und Software sowie Personal gemeint. Bei der ausgewählten Software wurde auf bekannte Software gesetzt, um eine Einarbeitungszeit in eine bis dato fremde Software zu verhindern und so den Entwicklungsprozess zu beschleunigen.

Im Anhang findet sich eine entsprechende Auflistung.

2.3 Entwicklungsprozess

Noch vor dem Beginn der Umsetzung musste sich für einen geeigneten Entwicklungsprozess entschieden werden, um die Vorgehensweise der Umsetzung vorab zu definieren. Das Team einigte sich hier auf eine agile Entwicklung, um schnell auf auftretende Probleme reagieren zu können. Durch kurze Entwicklungszyklen können bei einer agilen Entwicklung sehr zeitnah erste Ergebnisse präsentiert, Module getestet und auf Probleme bei der Umsetzung schnell reagiert werden, was zu Zeitersparnis und einem besseren Ergebnis für die Kunden führen kann, da auch auf kurzfristige Änderungswünsche eingegangen werden kann.

3 Analysephase

3.1 Ist-Analyse

Vor Beginn des Entwicklungsprozesses wurde eine kurze Ist-Analyse durchgeführt, welche zu dem Ergebnis kam, dass eine solche Software noch nicht in der Brühlwiesenschule existiert. Bisher verwendet wurden zum Speichern von Zitaten altmodische Medien wie Papier und Stift.

3.2.1 Make-Or-Buy Entscheidung

Bei dem Projekt handelt es sich um eine recht grundlegende Problemstellung, für die es bereits Tools gibt. Daher ist es vonnöten, eine gründliche Make-Or-Buy Analyse durchzuführen, da es möglich sein kann, dass es effektiv billiger ist eine Lösung einzukaufen, als selber ein Produkt zu entwickeln.

Hier hatte das Team allerdings keine Wahl, da die selbstständige Durchführung des Projektes Vorgabe durch den Auftraggeber war, weswegen die Analyse einer Buy-Lösung von vornherein ignoriert wurde.

3.2.2 Projektkosten

Hier sollen die anfallenden Projektkosten errechnet werden. Dafür werden die anfallenden Personalkosten sowie die Ressourcenkosten ermittelt, die während der Projektumsetzung anfallen. Die genauen Personalkosten innerhalb des Projektteams variieren und die Stundesätze der zuständigen Lehrer sind nicht bekannt, weswegen die Berechnung der Personalkosten anhand von Stundensätzen von 35€ für Lehrer und 10€ für Auszubildende erfolgt.

Kostentabelle			
Vorgang	**Personal**	**Zeit**	**Kosten**
Entwicklung	3x Auszubildender	23 h	700 €
Review	3x Auszubildender	0.3 h	10 €
Abnahme	1x Lehrer	2 h	70 €
Gesamt			**780 €**

Table 2: Kostentabelle

3.3 Lastenheft/Fachkonzept

Mit dem Ende der Analysephase wurde ein Lastenheft verfasst, in welchem die Wünsche und Vorgaben des Auftraggebers festgehalten wurden.
Ein Auszug findet sich im Anhang.

4 Entwurfsphase

4.1 Zielplattform

Zielplattform für das Projekt ist das System der BWS. Es wurden bis auf die Kompatibilität zu anderen Betriebssystemen als Windows keine Vorgaben vom Auftraggeber gemacht.

4.2 Architekturdesign

Das Design Der Software erfolgt nach dem Model-View-Controller Modell. Dieses Modell besagt, dass sich jeder Teil der fertig entwickelten Software in eine der drei Gruppen Model, View und Controller klar einfügen lässt.
Das Model übernimmt die Aufgabe der Datengenerierung und Verwaltung und beinhaltet alle logischen Komponenten des Programms sowie Datenverarbeitung.
Die View ist die Ansicht auf diese Daten und umfasst alle grafischen Oberflächen die dem User präsentiert werden.
Der Controller bildet das Bindeglied zwischen den ersten zwei Gruppen und reagiert beispielsweise auf Usereingaben und füllt Oberflächen der View mit Daten aus dem Model.

Das Model-View-Control Modell erlaubt es durch lose Koppelung der Komponenten flexibel erweiterbare Software zu schreiben, sodass eine Ausweitung der Applikation oder eine Veränderung von einzelnen Modulen komfortabler wird.

4.3 Entwurf der Benutzeroberflächen

Die Benutzeroberflächen sind gemäß dem MVC-Modell reine Informationsträger. Sie bekommen die anzuzeigenden Daten von Kontrollklassen im Hintergrund und geben Benutzereingaben an diese in Form von Methodenaufrufen zurück. Die Benutzeroberflächen beinhalten keine Logik und führen keinerlei Datenmanipulation durch.

4.4 Maßnahmen zur Qualitätssicherung

Im Laufe der Entwicklung führten wir immer wieder einzelne Komponententests durch. Nach Abschluss der Entwicklung testeten wir alle Komponenten im Verbund.

4.5 Pflichtenheft

Am Ende der Entwurfsphase wurde ein Pflichtenheft verfasst, in welchem die durchzuführenden Arbeiten dokumentiert wurden.
Ein Auszug findet sich im Anhang.

5 Implementierungsphase

5.1 Implementierung der Benutzeroberflächen

Für die Erstellung der Benutzeroberflächen wurden zunächst MockUps für alle Benutzeroberflächen erstellt. Anschließend wurden diese mit dem QT-Creator zusammengestellt und mit den ersten Methoden zu Benutzerinteraktionen versehen.

5.2 Implementierung der Backendlogik

Beim Implementieren der Backendlogik war darauf zu achten die Daten korrekt von der Oberfläche entgegenzunehmen und die Datenmanipulation an die Datenbank zu vermitteln. Dabei stand besonders die richtige Verarbeitung im Vordergrund, um Fehler zu vermeiden.

5.3 Aufsetzen der Datenbank

Für das Aufsetzen der Datenbank wurde zu Beginn des Projektes ein Datenmodell erstellt, in welchem konzeptionell überlegt wurde welche Daten wie persistiert werden sollen. Im Anschluss wurden einige Skizzen gemacht, wie diese Daten sinnvoll in eine Datenbank eingepflegt werden können.
Die entsprechende SQLite Datenbank wurde vom Projektteam anschließend mit den für das Projekt nötigen Tabellen erstellt. Erste DummyDaten wurden eingepflegt, um das Projekt ersten Tests unterziehen zu können.

6 Abnahmephase

Nach Fertigstellung des Projektes konnte die Applikation den Kunden vorgestellt werden. Aufgrund der agilen Entwicklung konnten bereits während der Implementierung Zwischenergebnisse vorgestellt werden, wodurch das Produkt nicht näher erläutert werden musste, da Oberflächen und Funktionsweisen bereits vertraut waren. Dadurch konnte die Einführung reibungslos durchgeführt werden.

7 Fazit

7.1 Soll-/Ist-Vergleich

Alle in dem Projektantrag festgehaltenen Soll-Aufgaben wurden erfüllt. Die Kann-Aufgabe der Votingfunktionalität wurde nicht fertig implementiert, jedoch ist ein Hinzufügen problemlos möglich, da das Datenmodell von Beginn an dafür vorbereitet wurde.
Auch die lose Kopplung der einzelnen Module macht eine solche Änderung leicht möglich.

7.2 Ausblick

Da es Wunschkriterium war, die Applikation als Webapplikation verfügbar zu machen, könnten in dieser Richtung nachfolgende Bemühungen erfolgen. Bisher fehlende Funktionalitäten können noch hinzugefügt werden.

Anhang

A.1 Ressourcenplanung 7
A.2 Klassendiagramm (Konzept) 7
A.3 Tabellen zur Zeitplanung 9
Detaillierte Zeitplanung 9
A.4 Lastenheft (Auszug) 10
A.5 Pflichtenheft (Auszug) 10
A.6 Erklärung MVC 11
A.7 Installationsbeschreibung 11
A.8 Benutzerhandbuch 12

A.1 Ressourcenplanung

Hardware

- Laptop

Software

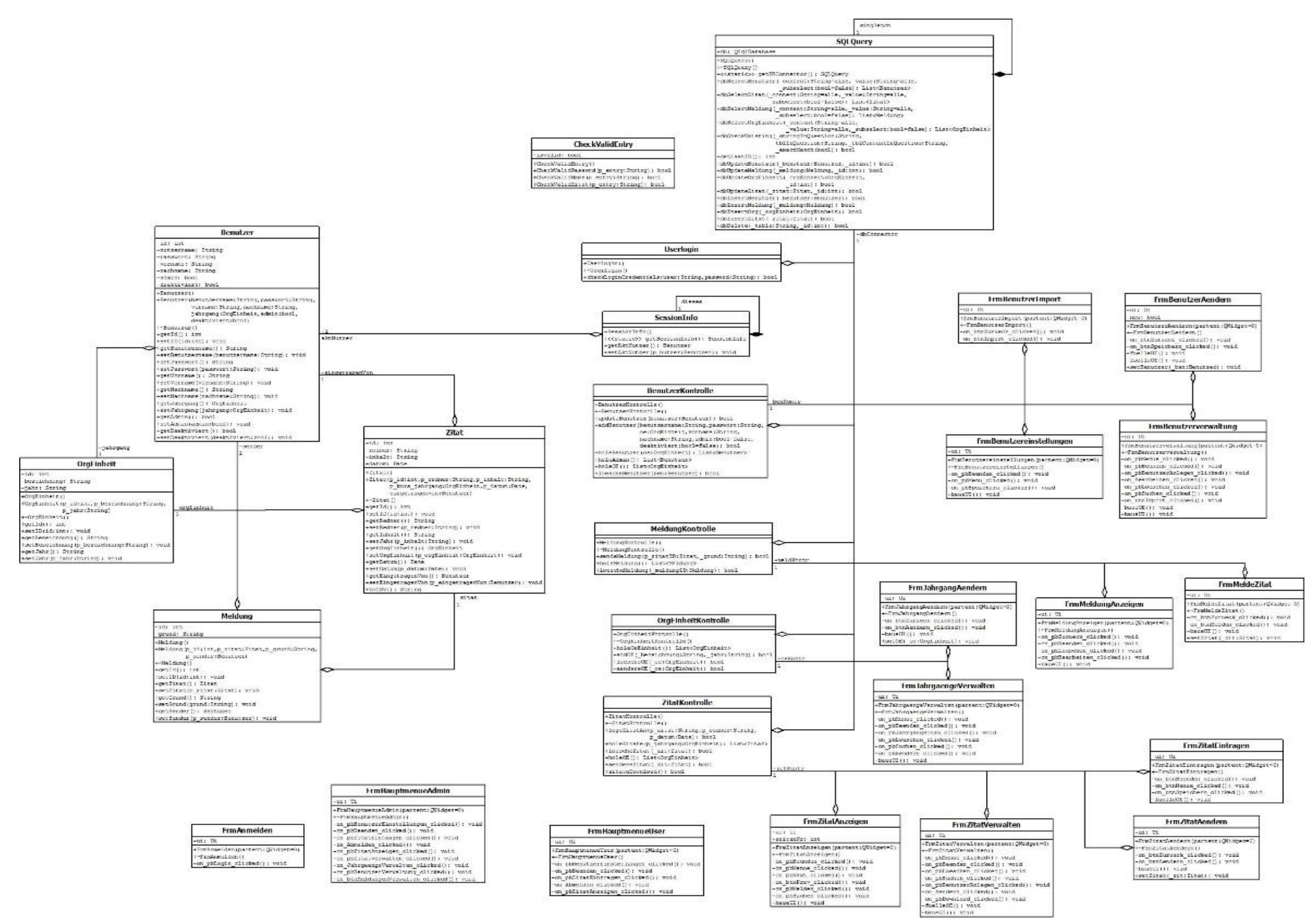
- QT-Creator 5.11 IDE
- SQLite Mozilla Firefox Plugin

Personal

- Klassenlehrer FIAE12

A.2 Klassendiagramm (Konzept)

Untenstehend ist das ausführliche Klassendiagramm.


SQLQuery
CheckValidEntry
Userlogin
SessionInfo
Benutzer
BenutzerKontrolle
FrmBenutzerImport
FrmBenutzerAendern
FrmBenutzerverwaltung
FrmBenutzereinstellungen
OrgEinheit
Zitat
MeldungKontrolle
FrmJahrgangAendern
FrmMeldungAnzeigen
FrmMeldeZitat
Meldung
OrgEinheitKontrolle
FrmJahrgaengeVerwalten
ZitatKontrolle
FrmZitatEintragen
FrmHauptmenueAdmin
FrmZitatAnzeigen
FrmZitatVerwalten
FrmZitatAendern
FrmAnmelden
FrmHauptmenueUser

A.3 Tabellen zur Zeitplanung

Detaillierte Zeitplanung

Detaillierte Zeitplanung	
Vorgang	**Eingeplante Zeit**
Analyse	
→Anforderungsaufnahme	6 h
→Durchführbarkeitsprüfung	6 h
→Rücksprache mit Auftraggeber	4 h
Planung/Design	
→Klassendiagramme/MockUps	11 h
→Datenmodelle	6 h
→Pflichtenheft	3 h
Durchführung	
→Benutzerverwaltung	8 h
→Datenbankverbindung	8 h
→Zitatfilterung	8 h
→Kursverwaltung	8 h
→Zitatanzeige/Verwaltung	20 h
→Zitaterfassung/Export	10 h
→Nutzerlogin	6 h
→Oberflächengestaltung	4 h
Dokumentation	
→Projektdokumentation	10 h
→Installationsbeschreibung	2 h
Gesamtzeit	**120 h**

Table 3: Detaillierte Zeiplanung

A.4 Lastenheft (Auszug)

Im nachfolgenden Lastenheftauszug sind die Anforderungen an die fertige Applikation definiert.

Die Applikation soll folgende Funktionalitäten bieten:

- User sollen nach Jahrgängen unterteilt werden
- Zitate sollen auf einer Oberfläche eintragbar und auslesbar sein
- Es soll einen Export der Zitate in eine Datei geben
- Das Projekt soll nach dem Model-View-Controller Modell konzipiert sein
- Zitate sollen löschbar und änderbar sein
- Userdaten sollen löschbar und änderbar sein
- Zitate sollen bewertet werden können

A.5 Pflichtenheft (Auszug)

Im nachfolgenden Pflichtenheftauszug wird die Umsetzung der Anforderungen aus dem Lastenheft beschrieben.

Umsetzung der Anforderungen:

- User werden nach Organisationseinheiten sortiert
- Zitate sollen auf einer QT-Form eintragbar und auslesbar sein
- Zitate werden im CSV-Format exportiert
- Jedes Modul soll klar in Model, View oder Controller eingeteilt werden können
- Zitate und Userdaten sind über eine QT-Oberfläche änderbar
- Zitate sollen pro User nur einmal bewertet werden können

A.6 Erklärung MVC

Das Model-View-Control Modell beschreibt eine Architektur, in der jedes Modul einer der drei Gruppen Model, View und Control eindeutig zuordenbar ist. Die View beinhaltet alle Oberflächen, also z. B. HTML Seiten die keine eigene Logik haben und nur eine Ansicht bzw. Formatierung beschreiben. Der Controller verarbeitet Zugriffe auf die Oberfläche, z. B. Knopfdrücke oder die Einfügung von zusätzlichen Absätzen in die bestehende Struktur der Oberfläche sowie das befüllen der HTML Oberfläche mit Daten. Dafür benötigte Daten werden vom Model bereitgestellt, welches beispielsweise auf Datenbanken zugreift und die eingelesenen Daten verarbeitet und weiterleitet. Interaktionen mit der Oberfläche durch den User werden ebenfalls vom Controller ausgewertet und an das Backend weitergeleitet.

Angefügt ist eine beispielhafte Skizze, die das Zusammenspiel der drei Komponenten bildlich grob erläutern soll.

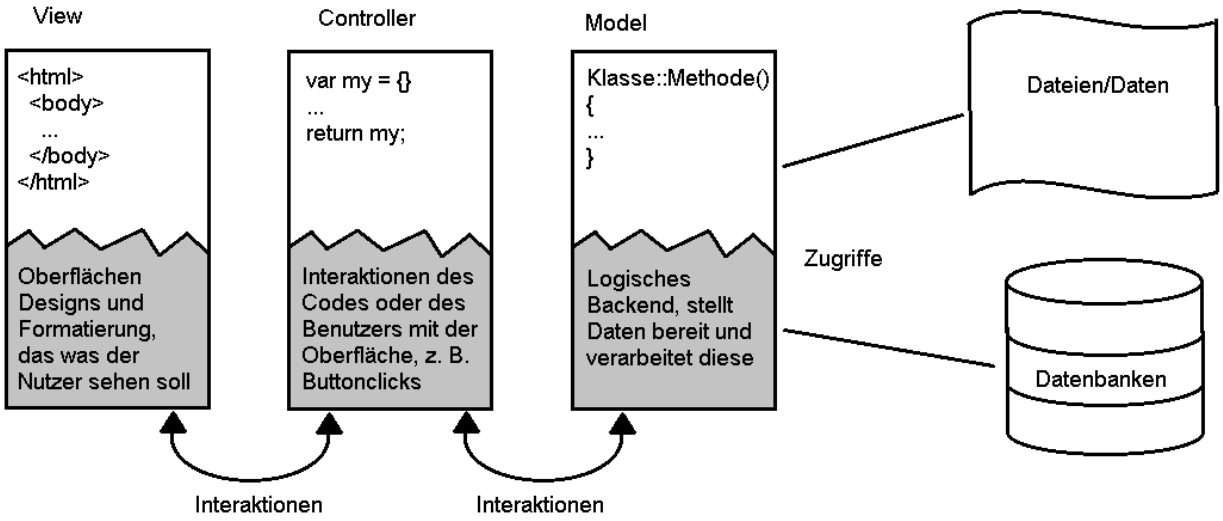


Figure 2:MVC Beispielhafte Darstellung

A.7 Installationsbeschreibung

Kopieren sie den Ordner „ZitatErfassung“ an den gewünschten Punkt. Öffnen sie den Qt Creator und gehen sie zu Datei->Projekt öffnen, alternativ Strg+O. Dann navigieren sie zu dem vorhin gewählten Verzeichnis. Wählen sie das Projekt ZitatErfassung.pro aus. Nachdem Qt Creator das Projekt geladen hat, verschieben sie die beiliegende DBZitat.sqlite in den von Qt Creator erstellten Ordner, in dem auch die ausführbare Datei liegt. Nun können sie das Programm im Qt Creator über den grünen Pfeil oder Strg+R starten.

A.8 Benutzerhandbuch

Anbei ist das Benutzerhandbuch.