

**HTBL Krems**

**Höhere Lehranstalt für Informationstechnologie**

HTL_Logo

**DIPLOMARBEIT**

**Webprojekt für Maxxwell-Computers**

Ausgeführt im Schuljahr 2018/19 von: Betreuer/Betreuerin:

Philipp Leitner 5AHIT Dipl.-Ing. Anton Hauleitner

Bernhard Hackl 5AHIT Dipl.-Ing. Anton Hauleitner

Krems, am 02.02.19

**EIDESSTATTLICHE ERKLÄRUNG**

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Krems, am 02.02.19 Verfasser / Verfasserinnen:

Philipp Leitner

Bernhard Hackl

**DIPLOMARBEIT**

**DOKUMENTATION**

|  |  |
| --- | --- |
| Namen der  Verfasser/innen | Philipp Leitner  Bernhard Hackl |
| Jahrgang / Klasse  Schuljahr | 5AHIT  2018/19 |
| Thema der Diplomarbeit | Webprojekt für Maxxwell-Computers |
| Kooperationspartner |  |

|  |  |
| --- | --- |
| Aufgabenstellung | Das Ziel dieses Projektes ist es für den Auftraggeber einen verbesserten Web-Auftritt, um mehr Interessenten zu bekommen und eine vereinfachte Bedienung der Webseite, d.h. eine gute Übersicht, laiengerechte Erstellung und Verwaltung der Projekte des Auftraggebers, sowie eine saubere Kommunikation mit Benutzern zu erreichen. |

|  |  |
| --- | --- |
| Realisierung | Das Webprojekt wurde mittels Zusammenspieles mehrerer Technologien erreicht, welche HTML, CSS, PHP, SQL und JavaScript beinhalten. Aufgeteilt wurde die Webseite in Schichten unter Einhaltung der MVC-Kriterien. Die Umsetzung erfolgte auf einem angemieteten Webserver des Auftraggebers |

|  |  |
| --- | --- |
| Ergebnisse | Das geplante Ergebnis des Web-Projektes ist eine gut strukturierte und informative Webseite mit leichter Bedienung, Verwaltung, Aufgabenplanung, sowie Benutzerverwaltung für den Auftraggeber, wodurch ein verbesserter Web-Auftritt des Unternehmens garantiert wird. |

|  |  |
| --- | --- |
| Typische Grafik, Foto etc.  (mit Erläuterung) |  |

|  |  |
| --- | --- |
| Teilnahme an Wettbewerben,  Auszeichnungen |  |

|  |  |
| --- | --- |
| Möglichkeiten der Einsichtnahme in die Arbeit |  |

|  |  |  |
| --- | --- | --- |
| Approbation  (Datum / Unterschrift) | Prüfer/in | Abteilungsvorstand / Direktor/in |

**DIPLOMA THESIS**

**Documentation**

|  |  |
| --- | --- |
| Author(s) | Philipp Leitner  Bernhard Hackl |
| Form  Academic year | 5AHIT  2018/19 |
| Topic | Webproject for Maxxwell-Computers |
| Co-operation partners |  |

|  |  |
| --- | --- |
| Assignment of tasks | The primary goal of this project is to improve the customers website in order to reach more interested people, facilitated operation of the website which includes a quick overview, easy creation and administration of the customers projects, as well as a clean communication with the users. |

|  |  |
| --- | --- |
| Realization | The web project has been made with the interaction of different technologies. These technologies include HTML, CSS, PHP, SQL and JavaScript. The website is divided into three layers following a basic MVC-approach. A server, rented by the customer served as our deployment system. |

|  |  |
| --- | --- |
| Results | The planned result of this web project was to reach a structured and informative website with a simple administration, operation and task scheduling, as well as user interaction. |

|  |  |
| --- | --- |
| Illustrative graph, photo  (incl. explanation) |  |

|  |  |
| --- | --- |
| Participation in competitions  Awards |  |

|  |  |
| --- | --- |
| Accessibility of  diploma thesis |  |

|  |  |  |
| --- | --- | --- |
| Approval  (Date / Sign) | Examiner | Head of Department / College |

Inhaltsverzeichnis

[1 Präambel 9](#_Toc536866906)

[1.1 Team 9](#_Toc536866907)

[1.2 Danksagung 9](#_Toc536866908)

[2 Pflichtenheft 9](#_Toc536866909)

[2.1 Zielbestimmungen 9](#_Toc536866910)

[2.1.1 Musskriterien 9](#_Toc536866911)

[2.1.2 Wunschkriterien 10](#_Toc536866912)

[2.2 Produkteinsatz 10](#_Toc536866913)

[2.2.1 Anwendungsbereiche 10](#_Toc536866914)

[2.2.2 Zielgruppen 10](#_Toc536866915)

[2.2.3 Betriebsbedingungen 10](#_Toc536866916)

[2.3 Produktumgebung 11](#_Toc536866917)

[2.3.1 Software 11](#_Toc536866918)

[2.3.2 Hardware 11](#_Toc536866919)

[2.3.3 Orgware 11](#_Toc536866920)

[2.4 Produktfunktionen 12](#_Toc536866921)

[2.4.1 Benutzerfunktionen 12](#_Toc536866922)

[2.4.2 Administratorfunktionen 15](#_Toc536866923)

[2.5 Produktdaten 17](#_Toc536866924)

[2.5.1 Benutzerdaten 17](#_Toc536866925)

[2.5.2 Auftragsdaten 18](#_Toc536866926)

[2.6 Produktleistungen 19](#_Toc536866927)

[2.6.1 Anmeldung 19](#_Toc536866928)

[2.6.2 Akkumulation 19](#_Toc536866929)

[2.6.3 Toleranz 19](#_Toc536866930)

[2.7 Produktübersicht 20](#_Toc536866931)

[2.8 Benutzungsoberfläche 22](#_Toc536866932)

[2.8.1 Dialogstruktur 22](#_Toc536866933)

[2.9 Qualitätsanforderungen 25](#_Toc536866934)

[2.10 Globale Testszenarien und Testfälle 26](#_Toc536866935)

[2.10.1 Benutzerfunktionstests 26](#_Toc536866936)

[*2.10.2* Administratorfunktionstests 26](#_Toc536866937)

[2.11 Entwicklungsumgebung 27](#_Toc536866938)

[2.11.1 Software 28](#_Toc536866939)

[2.11.2 Hardware 28](#_Toc536866940)

[2.12 Ergänzungen 28](#_Toc536866941)

[2.12.1 Sprachunterstützung 29](#_Toc536866942)

[2.13 Mockups 30](#_Toc536866943)

[2.13.1 Startseite 30](#_Toc536866944)

[2.13.2 Loginseite 31](#_Toc536866945)

[2.13.3 Registrierenseite 32](#_Toc536866946)

[2.13.4 Passwortvergessenseite 33](#_Toc536866947)

[2.13.5 Kontaktseite 34](#_Toc536866948)

[2.13.6 Profilseiten 35](#_Toc536866949)

[3 Technisches Pflichtenheft 45](#_Toc536866950)

[3.1 Daten 45](#_Toc536866951)

[3.1.1 Datenbank 45](#_Toc536866952)

[3.1.2 Webmaster-E-Mail 46](#_Toc536866953)

[3.2 Aufbau 47](#_Toc536866954)

[3.2.1 Model 47](#_Toc536866955)

[3.2.2 View 47](#_Toc536866956)

[3.2.3 Controller 47](#_Toc536866957)

[3.3 Tests 48](#_Toc536866958)

[3.4 Server/Client - Beschreibung 49](#_Toc536866959)

[3.4.1 Server 49](#_Toc536866960)

[3.4.2 Client 49](#_Toc536866961)

[3.5 Projektstrukturplan 51](#_Toc536866962)

[4 Theoretische Grundlagen 52](#_Toc536866963)

[5 Lösungsansätze für die Realisierung 52](#_Toc536866964)

[6 Dokumentation der Realisierung 52](#_Toc536866965)

[7 Testfälle 52](#_Toc536866966)

[8 Zusammenfassung und Ausblick 52](#_Toc536866967)

[8.1 Fazit 52](#_Toc536866968)

[8.2 Ausblick 52](#_Toc536866969)

[9 Dokumentation des Projektablaufs 52](#_Toc536866970)

[9.1 Arbeitsaufteilung 52](#_Toc536866971)

[9.2 Terminplanung 52](#_Toc536866972)

[9.3 Projekttagebuch 52](#_Toc536866973)

[10 Quellenverzeichnis 52](#_Toc536866974)

[11 Abbildungsverzeichnis 52](#_Toc536866975)

[12 Tabellenverzeichnis 52](#_Toc536866976)

[13 Anhang 52](#_Toc536866977)

# Präambel

Ein guter Web-Auftritt hat heutzutage, gerade im IT-Bereich, einen sehr großen Einfluss auf die Erfolgswahrscheinlichkeit eines Unternehmens. Wir, das Projektteam, haben uns für diese Diplomarbeit die Verbesserung des Web-Auftritts des Unternehmens „Maxxwell-Computers“ als Ziel gesetzt.

## Team

Das Projektteam besteht aus Philipp Leitner und Bernhard Hackl, Schüler der 5AHIT der Höheren Technischen Bundes- Lehranstalt, kurz HTBLA, Krems, Abteilung Informationstechnologie im Schuljahr 2018/19.

## Aufgabenteilung

Die Aufgabenteilung innerhalb unseres Teams erfolgte durch die Umsetzung des Frontend von Bernhard Hackl, dies inkludiert das Design und den Aufbau der Webseite. Das Backend wurde durch Philipp Leitner umgesetzt, welches den Umgang aller Daten, sowie die Funktionen der Webseite beinhaltet.

# Pflichtenheft

## Zielbestimmungen

Das Unternehmen Maxxwell-Computers möchte durch folgende Ziele ein Webprojekt bilden und damit dessen Online-Präsenz verbessern.

Im Folgenden bezeichnet der Benutzer einen Kunden des Auftraggebers und der Administrator den Auftraggeber des Webprojektes (Markus Beneder).

### Musskriterien

* Benutzer-Account
  + Der Benutzer kann sich selbst am System registrieren.
  + Der Benutzer muss zur Registrierung ein Anmeldeformular ausfüllen.
  + Der Benutzer kann sich am System anmelden und abmelden.
  + Der Benutzer kann sich seine persönlich angegebenen Daten anzeigen lassen in der Form eines Profils, sowie Änderungen an diesen vornehmen.
  + Der Benutzer kann seine Anmeldedaten (E-Mail und Passwort) zurücksetzen sollte er diese vergessen haben.
  + Der Benutzer kann seinen Account endgültig aus dem System löschen.
  + Der Benutzer kann einen Antrag für einen Auftrag mittels eines Formulars ausfüllen.
  + Der Benutzer kann Anträge an den Administrator senden.
  + Der Benutzer kann seine Aufträge einsehen, sowie deren derzeitigen Status abprüfen.
  + Der Benutzer kann dem Administrator Mails senden, dies ist unter der Option „Administrator kontaktieren“ in dem Benutzer-Profil möglich.
* Administrator-Account
  + Der Administrator kann sich selbst am System anmelden und abmelden.
  + Der Administrator besitzt einen einmaligen Account, welcher sicherheitsrechtlich mit stärkeren Passwort-Kriterien versehen ist als der Benutzer-Account
  + Der Administrator kann seine Aufträge in einer Übersicht in seinem Profil abrufen
  + Der Administrator kann seine Aufträge sortieren nach selbst gewählten Kriterien (Start-, Enddatum, Priorität, Status)
  + Der Administrator kann neue Aufträge erstellen, sowie bearbeiten
  + Der Administrator kann ein „Showcase“ erstellen, welches öffentlich auf der Startseite, derzeitige Aufträge für alle Webseiten-Besucher anzeigt
* Web-Auftritt
  + Der Webseiten-Besucher kann über einen gewählten URL die Webseite aufrufen
  + Der Webseiten-Besucher bekommt auf der Startseite eine allgemeine Beschreibung über das Unternehmen zu sehen
  + Der Webseiten-Besucher kann sich private Informationen über Standort/Administrator im Reiter „Kontakt“ anzeigen lassen
  + Der Webseiten-Besucher kann sich die „Showcases“ ansehen

### Wunschkriterien

* Der Benutzer hat einen direkten, in der Webseite integrierten Chat mit dem Administrator, wobei er Fragen stellen kann
* Der Administrator kann Fragen der Benutzer mittels Chats direkt beantworten

## Produkteinsatz

Im folgenden Abschnitt werden Anwendungsbereiche, Zielgruppen und Betriebsbedingungen des Webprojekts definiert.

### Anwendungsbereiche

Web-Besucher bekommen eine Übersicht über die angebotenen Dienstleistungen des Unternehmens Maxxwell-Computers, sowie Kontaktdaten. Kunden benutzen die Webseite um ihre Aufträge im Bereich IT-Dienstleistungen anzubieten und durchzuführen, dabei können sie jederzeit dessen Status abrufen und direkten Kontakt zum Auftragnehmer herstellen.

### Zielgruppen

Personengruppen, welche nach Informationen über das Unternehmen Maxxwell-Computers im Internet suchen.

Hauptsächlich Privatkunden und Unternehmer, welche nach Auftragnehmern in der IT-Dienstleistungsbranche suchen.

### Betriebsbedingungen

Dieses System soll sich bezüglich der Betriebsbedingungen nicht wesentlich von anderen Internetdiensten bzw. Internetanwendungen unterscheiden.

* Betriebsdauer: 24 Stunden/Tag (Dauerbetrieb)
* Wartungsfrei
* Die Sicherung der Datenbank muss manuell vom Administrator durchgeführt werden
* Falls nötig, ist der Administrator zur Schlichtung zwischen Benutzern verantwortlich

## Produktumgebung

Das Produkt ist weitgehend unabhängig vom Betriebssystem, sofern folgende Produktumgebung vorhanden ist.

### Software

* Client
  + www-Browser der neuesten Generation (keine textbasierten Browser)
* Server

### Hardware

* Client
  + Internetfähiger Rechner
* Server
  + Internetfähiger Server
  + Rechner, der die Ansprüche der o.g. Server Software erfüllt
  + Ausreichend Rechen- und Festplattenkapazität

### Orgware

* Gewährleistung der permanenten Internetanbindung bzw. Bezahlung des Servers (Mietkosten)

## Produktfunktionen

Im folgenden Abschnitt werden alle Funktionen des Webprojektes im Detail beschrieben.

### Benutzerfunktionen

Dieser Abschnitt beschreibt alle Funktionen der Benutzer.

#### Benutzer-Kennung

Folgend wird die Kennung des Benutzers beschrieben.

Folgendes Wortregister sollte hilfreich sein:

* Account: Benutzer kann diesen am System erstellen
* Profil: Benutzerprofil, beinhält die Kennung des Accounts & stellt Funktionen bereit
* Kennwort = Passwort

##### Registrieren

Ein Benutzer kann sich an dem System registrieren, indem er den dafür vorgesehenen Button auf der Startseite benützt bzw. wird ihm die Möglichkeit geboten, beim Login-Abschnitt einen Account zu erstellen.

Um sich einen Account zu erstellen muss der Benutzer mindestens folgende Angaben erfüllen:

* E-Mail
* Benutzername
* Passwort

Die Registrierung ist erfolgreich, wenn der Benutzername und die E-Mail-Adresse innerhalb des Systems jeweils eindeutig sind.

Die E-Mail-Adresse wird auf ihre Gültigkeit geprüft.

Mit dem erfolgreichen Abschießen des Registrierungsvorgangs ist der neue Benutzer am System angemeldet und erhält automatisch vom System eine E-Mail, welche die erfolgreiche Registrierung für den Benutzer bestätigt.

##### Anmelden

Ein Benutzer, welcher sich bereits am System registriert hat, kann sich mit einer der folgenden Kombinationen am System anmelden:

* E-Mail-Adresse
* Passwort

Oder

* Benutzername
* Passwort

Eine dieser Kombinationen tippt der Benutzer dann in das „Login-Feld“ auf der Startseite der Webseite.

##### Abmelden

Ein Benutzer kann sich, sofern er gerade am System angemeldet ist, über das „Logout-Feld“ vom System wieder abmelden.

##### Kennung anfordern

Sollte ein Benutzer seine Kennung vergessen, kann dieser auf der Startseite bei dem „Login-Feld“ eine neue anfordern, dies geschieht folgendermaßen:

Der Benutzer klickt auf das „Passwort vergessen“ Feld und wird auf eine Seite weitergeleitet, wo dieser aufgefordert wird seinen Benutzername einzugeben, sollte dieser auch in Vergessenheit geraten sein, kann alternativ auch die E-Mail-Adresse angegeben werden.

Anschließen wird dem Benutzer vom System eine E-Mail geschickt, diese enthält seine Kennung:

* Benutzername
* Passwort

##### Änderungen am Profil/Account

Möchte ein Benutzer Änderungen an seinem Account vornehmen möchten, zum Beispiel seine E-Mail-Adresse oder sein Kennwort zu ändern, ist dies in seinem Profil möglich.

Dazu muss der Benutzer bereits angemeldet sein, anschließend stehen ihm diese Funktionen auf der Seite seines Profils bereit.

Um das Kennwort zu ändern, muss zuerst das alte Passwort eingegeben und überprüft werden, damit anschließend das neue Kennwort eingegeben und übernommen werden kann.

Die E-Mail-Adresse wird ebenfalls auf der Profil-Seite geändert, indem zuerst die alte E-Mail-Adresse eingegeben und überprüft werden muss, dann kann erst die neue E-Mail-Adresse übernommen werden.

Wird eine Änderung am Profil vorgenommen, muss man sich nochmals am System mit der neuen Kennung anmelden.

Außerdem wird der Benutzer nach jeder Änderung an seinem Account per E-Mail vom System benachrichtigt.

Der Benutzername kann nicht geändert werden.

#### Persönliche Daten

Der Benutzer verfügt über persönliche Daten, welche für bestimmte Funktionen/Operationen benötigt werden. Diese setzen eine erfolgreiche Registrierung und Anmeldung vor.

##### Anzeige der persönlichen Daten

Der Benutzer kann sich seine persönlichen Daten auf der Seite seines Profils anzeigen lassen.

##### Hinzufügen der persönlichen Daten

Der Benutzer kann selbst persönliche Daten zu seinem Profil hinzufügen, dazu kann folgende Angaben zusätzlich zu E-Mail-Adresse und Benutzername machen:

* Vorname
* Nachname
* Geschlecht
* Geburtsdatum
* Firma/Unternehmen
* Fax
* Straße
* Hausnummer
* Postleitzahl
* Ort
* Land
* Homepage
* Telefonnummer
* Kurze Selbstbeschreibung

##### Ändern der persönlichen Daten

Der Benutzer kann alle in [4.1.2.2](#_Hinzufügen_der_persönlichen) hinzugefügten Daten wieder ändern.

##### Sichtbarkeit der persönlichen Daten

Die persönlichen Daten der Benutzer sind nur für den Administrator von Bedeutung und daher auch nur von diesen eingesehen werden.

Die persönlichen Daten eines Benutzers können nicht von einem anderen Benutzer eingesehen werden.

Sollte der Benutzer einen Auftrag aufgeben, werden diese Daten vorausgesetzt und daher auch bei einem Auftrag an den Administrator mitgeschickt.

#### Account löschen

Der Benutzer kann bei Bedarf seinen Account löschen.

##### Endgültiges Löschen des Accounts mit aller Daten

Wenn der Benutzer in seinem Profil eingeloggt ist, hat er die Option seinen Account nach erneuter Eingabe des Passworts zu löschen, dabei werden folgende Dinge gelöscht:

* Persönliche Daten
* Gesamte Kennung
  + E-Mail-Adresse
  + Benutzername
  + Passwort
* Alle offenen Aufträge werden gelöscht
* Alle laufenden Aufträge werden abgebrochen und ebenfalls gelöscht

#### Benachrichtigungsfunktionen

Der Benutzer kann sich weitere Informationen über dessen Aufträge holen, zum Beispiel wenn dieser noch weitere Fragen hat.

##### Benachrichtigungen per E-Mail

Der Benutzer kann den Administrator per E-Mail kontaktieren. Dies erfolgt per Klick auf den Button „E-Mail senden“ in des Benutzers Profil unter dem Abschnitt Administrator kontaktieren. Anschließend wird dieser weitergeleitet und kann nun eine E-Mail an den Administrator senden, wobei die Kontakt-E-Mail-Adresse des Administrators automatisch ausgefüllt wird.

Als Absender-E-Mail-Adresse wird, die im Profil des Benutzers hinterlegte E-Mail-Adresse verwendet.

#### Auftragsfunktion

Der Benutzer kann Aufträge erstellen und an den Administrator senden.

##### Auftrag erstellen

Der Benutzer kann indem er in seinem Profil unter dem Auftrags-Reiter neue Aufträge erstellen.

Dazu muss dieser zuerst seine Persönlichen Daten ausgefüllt haben.

Für den Auftrag müssen folgende Daten ausgefüllt werden:

* Auftragstitel
* Auftragsbeschreibung

### Administratorfunktionen

Im folgenden Abschnitt werden alle Funktionen des Administrators im Detail beschrieben.

Im Grunde hat der Administrator fast alle Funktionen der Benutzer, mit Ausnahme der Aufgabe von Aufträgen (sinnbefreit, da der Administrator Aufträge annimmt/ablehnt).

Weiters kann der Administrator seinen Account nicht einfach löschen.

#### Showcasefunktionen

Folgend werden alle Möglichkeiten des Administrators beschrieben, die Showcases auf der Startseite zu administrieren.

##### Showcase erstellen

Der Administrator kann bestimmte Aufträge auswählen und auf der Startseite in der sogenannten Showcase anzeigen lassen.

Das Showcase ist dann für alle Webseiten-Besucher verfügbar bzw. einsehbar.

##### Showcase bearbeiten

Der Administrator kann erstellte Showcases bearbeiten, dies beinhaltet die Option, wieviel von dem Auftrag angezeigt werden soll.

Unterscheiden kann der Administrator zwischen folgenden Anzeigen:

* Auftrag ohne Kontaktdaten des Auftraggebers
* Auftrag mit Kontaktdaten des Auftraggebers

##### Showcase löschen

Der Administrator kann erstellte Showcases auf der Startseite löschen, diese werden dann durch einen Platz zum Einfügen neuer ersetzt.

#### Auftragsfunktionen

Folgend werden alle Funktionen des Administrators bezüglich der Aufträge beschrieben. Dazu muss der Administrator am System angemeldet sein und sich auf seiner Profilseite befinden.

##### Aufträge ansehen

Der Administrator kann sich in seinem Profil unter dem Reiter Aufträge alle Aufträge anzeigen, diese können in einem der folgenden Zustände vorliegen

* Noch nicht behandelt
* In Bearbeitung
* Abgeschlossen

Zusätzlich werden dem Administrator alle Kontaktdaten des jeweiligen Auftrag-Erstellers angezeigt.

##### Aufträge annehmen

Befindet sich ein Auftrag im Status „Noch nicht behandelt“, kann der Administrator diesen annehmen, indem er auf diesen klickt und die Auswahl „Annehmen“ trifft.

Im Anschluss daran wird der zugehörige Benutzer über diese Änderung des Auftrages mit einer, vom System automatisch gesendeten, E-Mail benachrichtigt.

##### Aufträge ablehnen

Befindet sich ein Auftrag im Status „Noch nicht behandelt“, kann der Administrator diesen ablehnen, indem er auf diesen klickt und die Auswahl „Ablehnen“ trifft.

Im Anschluss daran wird der zugehörige Benutzer über diese Änderung des Auftrages mit einer, vom System automatisch gesendeten, E-Mail benachrichtigt.

##### Auftragszustände ändern

Der Administrator kann die Zustände der Aufträge ändern.

Sollte der Administrator einen Auftrag annehmen oder ablehnen, so wird der Zustand dieses Auftrages vom System automatisch auf „In Bearbeitung“ geändert.

Nach Fertigstellung eines Auftrages, kann der Administrator den Zustand auf „Abgeschlossen“ stellen, dies benachrichtigt den zugehörigen Benutzer automatisch vom System mit einer E-Mail.

##### Auftrags-Ersteller benachrichtigen/informieren

Der Administrator kann den Benutzer, welcher einen Auftrag erstellt hat, per E-Mail benachrichtigen, dies erfolgt durch einen Klick auf den Text „E-Mail an Benutzer senden“.

Dieser Text ist auf dem jeweiligen Auftrag zu finden.

Der Administrator wird nach Klick weitergeleitet und ihm wird die Möglichkeit geboten eine Mail an den Benutzer zu senden, dabei werden Kontakt-E-Mail des Benutzers und die E-Mail-Adresse des Administrator-Kontos vom System automatisch ausgefüllt.

## Produktdaten

Folgende Daten werden langfristig vom Webprojekt gespeichert.

### Benutzerdaten

Folgender Abschnitt beschreibt alle Informationen, welche zu einem Benutzer gespeichert werden.

#### Benutzer ohne persönliche Daten

Wenn ein Benutzer keine persönlichen Daten angibt wird folgendes gespeichert:

* Benutzer-ID (einmalig)
* Kennung
  + Benutzername (einmalig)
  + Passwort (verschlüsselt)
  + E-Mail-Adresse (einmalig)
* Sonstige Daten
  + Registrierungsdatum
  + Anmeldezeiten
  + Status (Benutzer/Administrator)

#### Benutzer mit persönlichen Daten

Wenn ein Benutzer persönliche Daten angibt wird folgendes gespeichert:

* Benutzer-ID (einmalig)
* Kennung
  + Benutzername (einmalig)
  + Passwort (verschlüsselt)
  + E-Mail-Adresse (einmalig)
* Persönliche Daten
  + Vorname
  + Nachname
  + Geschlecht
  + Geburtsdatum
  + Firma/Unternehmen
  + Kurze Selbstbeschreibung
* Kontaktdaten
  + Fax
  + Straße
  + Hausnummer
  + Postleitzahl
  + Ort
  + Land
  + Homepage
  + Telefonnummer
* Sonstige Daten
  + Registrierungsdatum
  + Anmeldezeiten
  + Status (Benutzer/Administrator)
* Auftragsdaten
  + Auftrags-ID

### Auftragsdaten

Folgende Daten werden gespeichert, sollte ein Auftrag erstellt werden.

* Auftrags-ID
* Benutzerdaten
  + Benutzer-ID
* Beschreibung
* Status (Noch nicht behandelt/In Bearbeitung/Abgeschlossen)
* Sonstige Daten
  + Erstellungsdatum

## Produktleistungen

In diesem Abschnitt werden die Anforderungen an das Produkt spezifiziert. Dabei geht es um die Leistungen, die das Webprojekt erbringen muss.

### Anmeldung

Die Zeit der Anmeldung, d.h. Überprüfung der Eingabe der Anmeldedaten (Benutzername/E-Mail und Passwort) eines Benutzers bzw. Administrators muss im Bereich von 2 bis 5 Sekunden liegen.

### Akkumulation

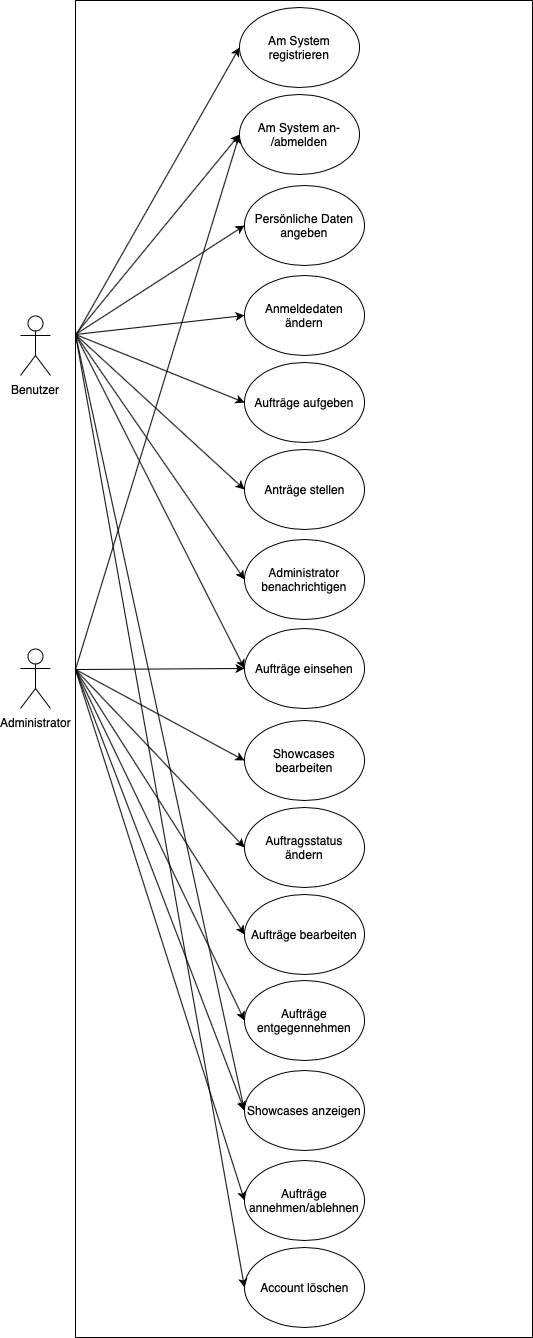
Sollte eine fehlerhafte Eingabe durch den Benutzer bzw. Administrators erfolgen, so wird dieser durch eine Fehlermeldung über die fehlerhafte Eingabe benachrichtigt.

### Toleranz

Bei fehlerhaften Eingaben soll die Korrektur der Eingabedaten möglich sein, ohne die Eingabe wiederholt vornehmen zu müssen.

## Produktübersicht

Folgende Grafik stellt die vereinbarten Funktionalitäten dar.



## Benutzungsoberfläche

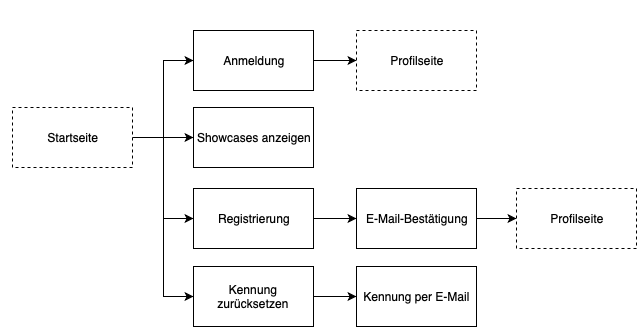
Im folgenden Kapitel werden die grundlegenden Anforderungen an die Benutzungsoberfläche beschrieben.

### Dialogstruktur

Im Folgenden wird die grobe Dialogstruktur einer fehlerfreien bzw. konfliktfreien Benutzung des Systems gezeigt.

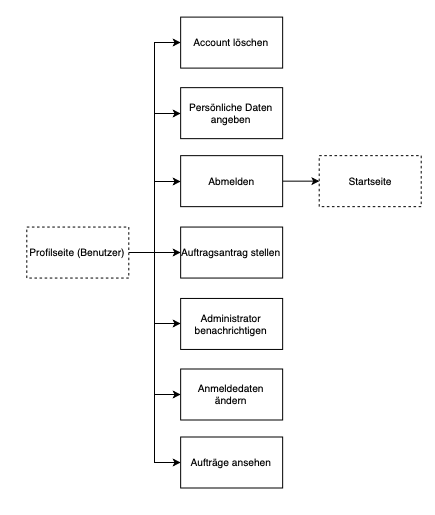
#### Startseite

Die Startseite ist die Seite, welche beim Webseitenaufruf angezeigt wird.



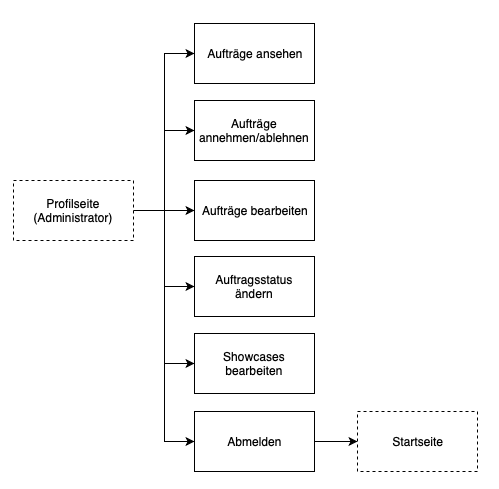
#### Profilseite (Benutzer)

Nach erfolgreicher Anmeldung kann der Benutzer seine Profilseite aufrufen und hat folgende Möglichkeiten.



#### Profilseite Administrator

Nach erfolgreicher Anmeldung kann der Administrator seine Profilseite aufrufen und hat folgende Möglichkeiten.



## Qualitätsanforderungen

Auf folgende Qualitätsmerkmale wird besonders Wert gelegt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Sehr wichtig | Wichtig | Weniger wichtig | Unwichtig |
| Robustheit |  | x |  |  |
| Zuverlässigkeit |  | x |  |  |
| Korrektheit |  | x |  |  |
| Benutzerfreundlichkeit | x |  |  |  |
| Effizienz |  | x |  |  |
| Portierbarkeit |  |  |  | x |
| Kompatibilität |  |  | x |  |

## Globale Testszenarien und Testfälle

Bestimmte Produktfunktion werden von folgenden konkreten Testfällen getestet.

### Benutzerfunktionstests

In diesem Abschnitt werden Benutzerfunktionen getestet.

#### Registrierung

Ein Benutzer registriert sich am System.

#### Anmeldung

Ein Benutzer meldet sich mit seinem erstellten Account am System an.

#### Abmeldung

Ein Benutzer meldet sich vom System ab. Er wird wieder auf die Startseite zurückgeleitet.

#### Kennung anfordern

Ein Benutzer vergisst seine Kennung und fordert beim Anmelden seine Daten per E-Mail an.

#### Änderungen am Profil

Ein Benutzer ändert sein Passwort.

#### Hinzufügen persönlicher Daten

Ein Benutzer fügt neue persönliche Daten in seinem Profil hinzu.

#### Ändern der persönlichen Daten

Ein Benutzer hat bereits persönliche Daten seinem Profil hinzugefügt und ändert diese nun.

#### Account löschen

Ein Benutzer löscht seinen Account.

### Administratorfunktionstests

In diesem Abschnitt werden Administratorfunktionen getestet.

#### Showcases erstellen

Der Administrator erstellt Showcases, welche auf der Startseite angezeigt werden.

#### Showcases bearbeiten

Der Administrator ändert Daten der Showcases.

#### Showcases löschen

Der Administrator löscht Showcases.

#### Aufträge annehmen

Der Administrator nimmt einen Auftrag an.

#### Aufträge ablehnen

Der Administrator lehnt einen Auftrag ab.

#### Auftragszustand ändern

Der Administrator ändert den Zustand eines Auftrages.

#### Auftrags-Ersteller benachrichtigen

Der Administrator benachrichtigt einen Auftrags-Ersteller über dessen Auftragsänderung.

## Entwicklungsumgebung

Es wird darauf geachtet, dass alle Entwicklertools kostenlos (Freeware) sind. Sollten Produkte lizensiert sein bzw. vom Auftraggeber bereitgestellt werden, so wird dies angeführt.

### Software

Im folgenden Abschnitt wird die verwendete Software angegeben.

#### Plattform

* PHP 7.2.11
* MySQL 5.7
* JS ECMA Script 2018

#### Tools

* Balsamiq Free Version (aktuellste Version)
* PHPStorm 2018.3 (lizensiert durch HTLKrems)
* LaTeX
* Atom (aktuellste Version)
* Git
* MsProject (aktuellste Version, lizensiert durch HTLKrems; wird nur für die Planung verwendet)

#### Browser

* Safari 12.0
* Microsoft Edge 44.17763.1.0
* Mozilla Firefox 63.0

### Hardware

Im folgenden Abschnitt wird die verwendete Hardware angegeben.

Als Hardware wird ein Server bereitgestellt, welcher bei World4you Webhosting ([www.world4you.com](http://www.world4you.com)) vom Auftraggeber angemietet ist.

Genauer, der Domainserver 2018 mit Domainregistrierung.

Die Zugangsdaten wurden bereits an die Auftragnehmer übermittelt.

## Ergänzungen

Im folgenden Kapitel werden Ergänzungen des Projektes beschrieben.

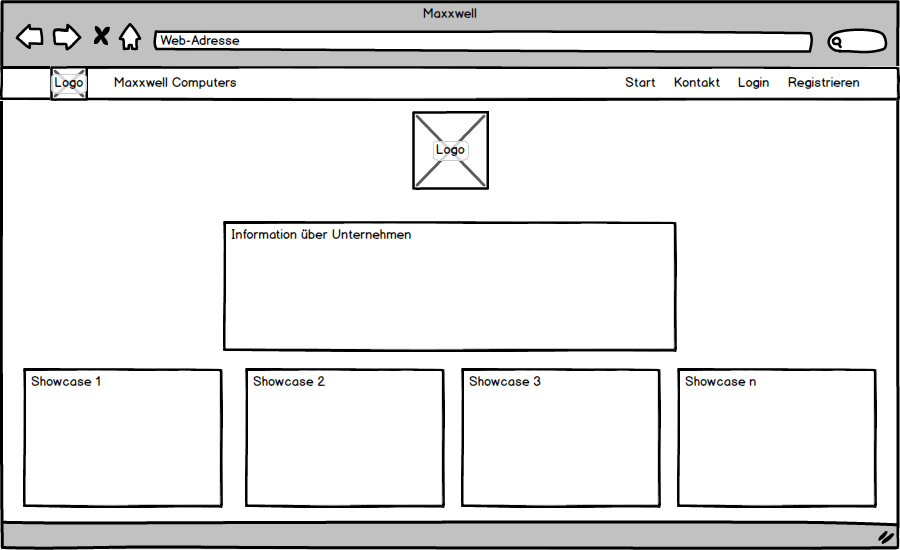
### Sprachunterstützung

Das System unterstützt in dieser Version ausschließlich deutsche Beschriftungen. Eine Art „Internet“-Umgangssprache wird ebenfalls verwendet. D. h. Begriffe wie Community oder Login werden nicht ins Deutsche übersetzt.

## Mockups

Im folgenden Abschnitt wird eine geplante Darstellung des Webprojektes beschrieben. Das Logo und die farbliche Gestaltung wird an die bereits existierende Informations-webseite (http://www.maxxwell.co.at) angelehnt.

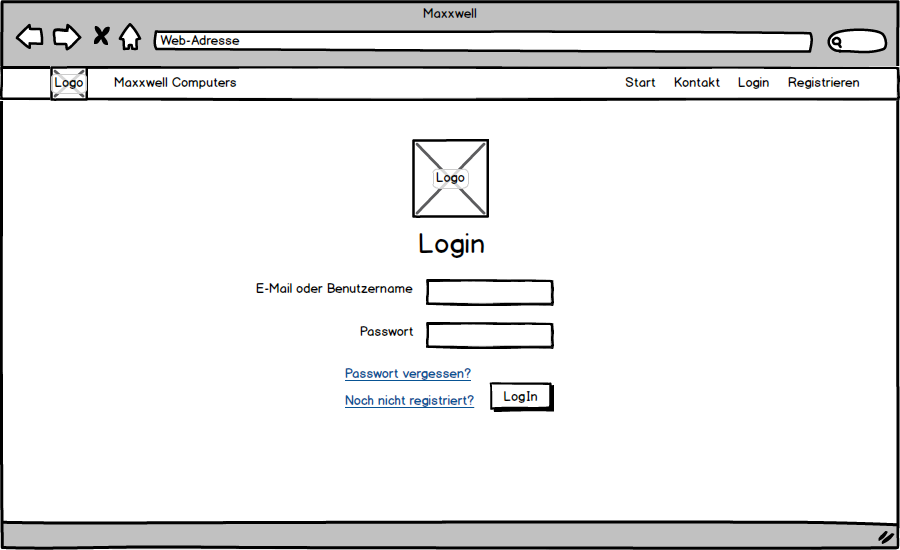
### Startseite



Durch Klick auf „Login“ wird man auf die Loginseite weitergeleitet.

Durch Klick auf „Registrieren“ wird man auf die Registrierenseite weitergeleitet.

### Loginseite

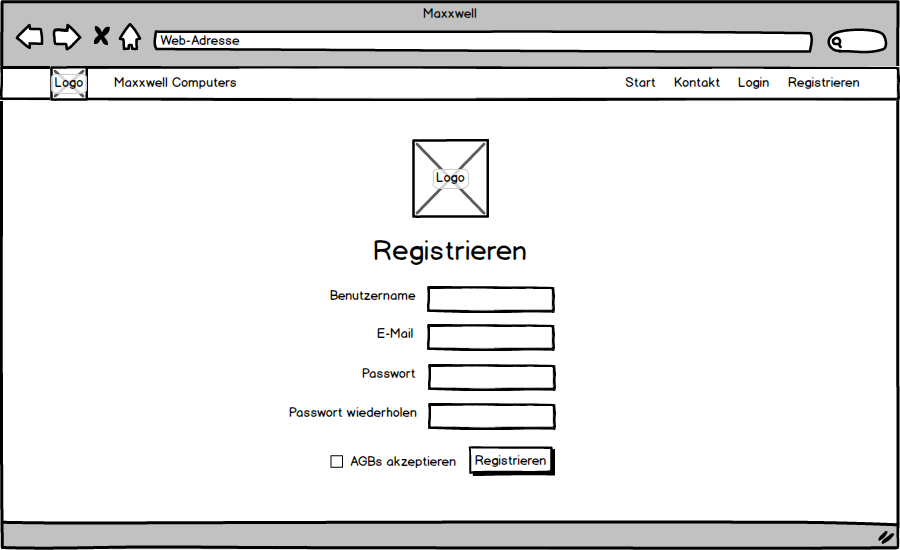


Nachdem die Daten ausgefüllt wurden, kann auf „Login“ geklickt werden (bei fehlerhafter Eingabe wird der Benutzer aufgefordert seine Eingabe zu korrigieren).

Bei Klick auf „Passwort vergessen?“, wird der Benutzer auf die Passwortvergessenseite weitergeleitet.

Bei Klick auf „Noch nicht registriert?“ wird der Benutzer auf die Registrierenseite weitergeleitet.

### Registrierenseite



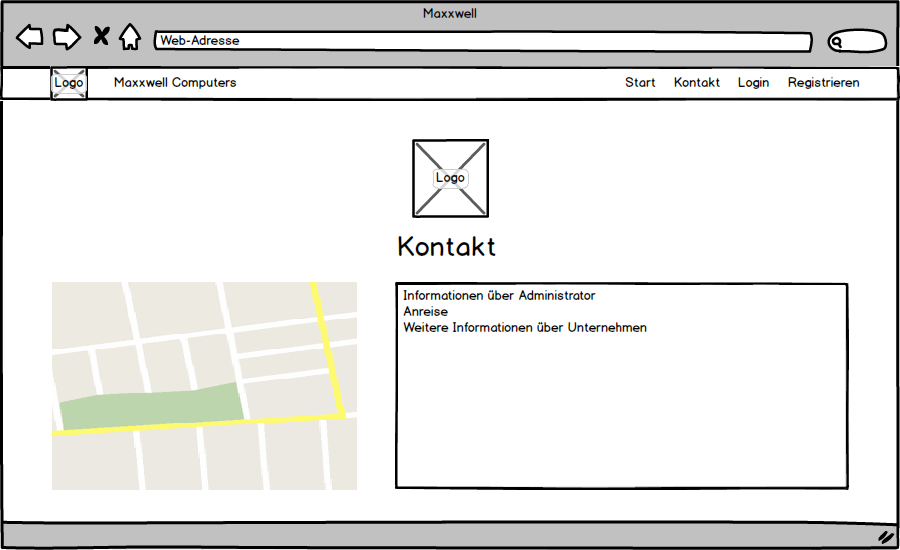
Hat der Benutzer alle Felder ausgefüllt, findet nach einem Klick auf „Registrieren“ eine Überprüfung der Eingabe statt, wobei der Benutzer aufgefordert wird etwaige Fehler auszubessern.

### Passwortvergessenseite



Existiert der eingegeben Benutzername bzw. die eingegebene E-Mail-Adresse, so wird der Benutzer nach Klick auf „Kennung per E-Mail anfordern“ darüber benachrichtigt eine E-Mail zu erhalten.

### Kontaktseite

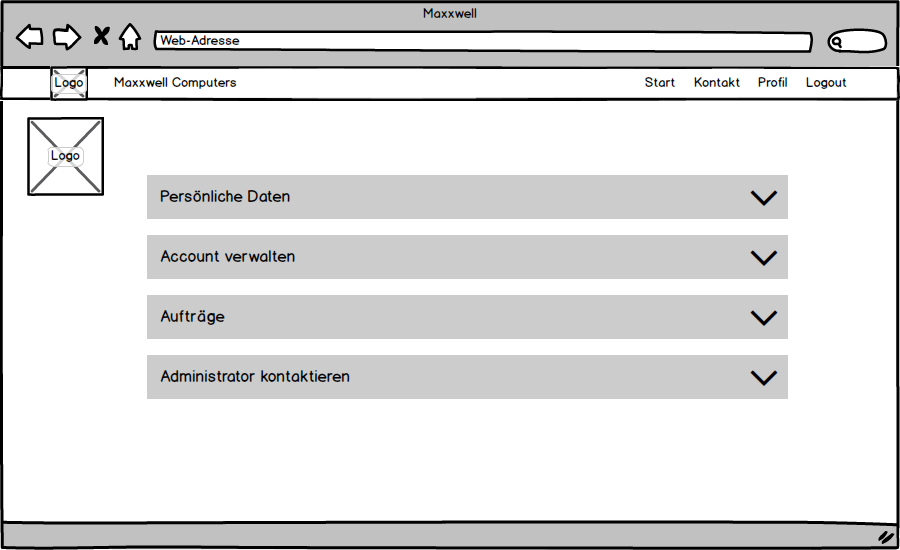


Die „Map“ stellt eine GoogleMaps-Integration dar, welche bei Klick GoogleMaps öffnet.

### Profilseiten

Im folgenden Abschnitt wird zwischen der Profilseite des Administrators und der Profilseite der Benutzer unterschieden.

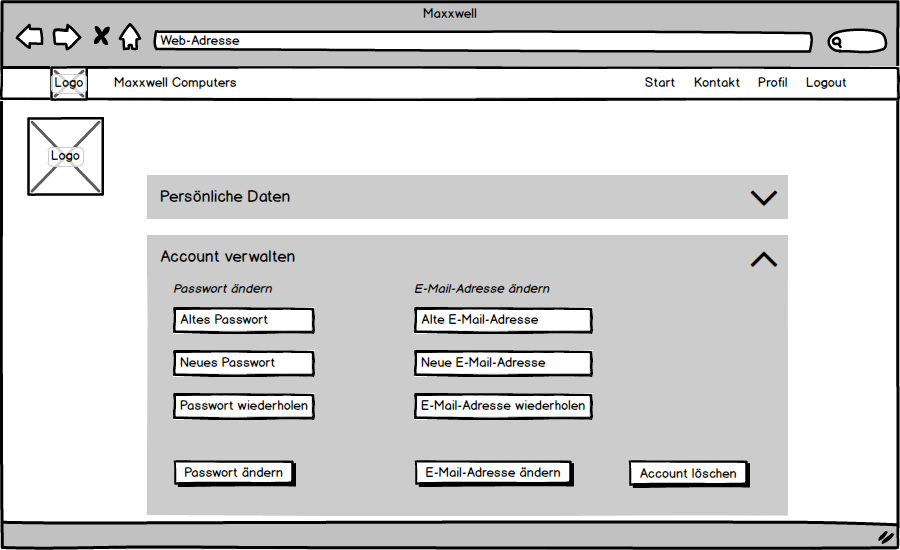
#### Benutzer-Profilseite



Die Profilseite eines Benutzers ist in vier Segmente geteilt, welche per Klick auf die Pfeile mehrere Optionen bieten.

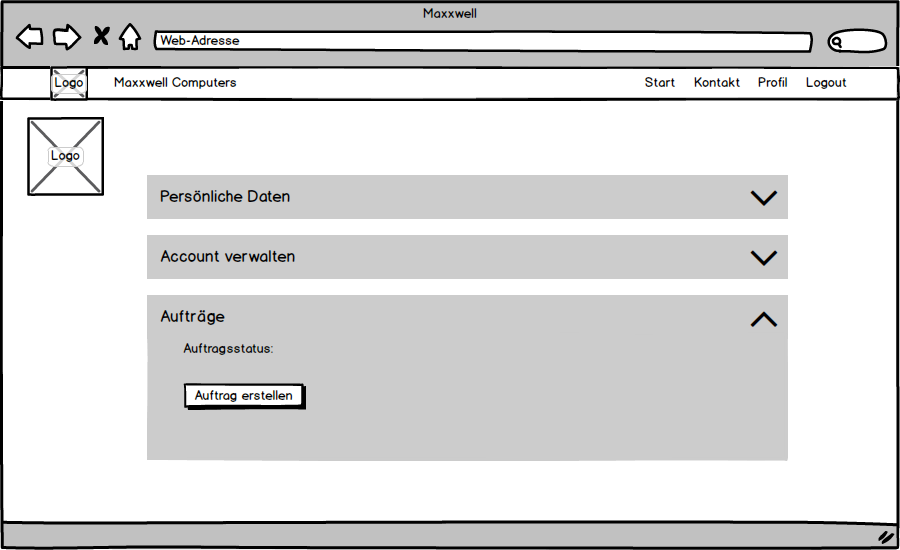


Nach einem Klick auf „Persönliche Daten“, werden alle ausfüllbaren Felder angezeigt, welche durch einen Klick auf „Speichern“ übernommen werden und später auch noch bearbeitet werden können.

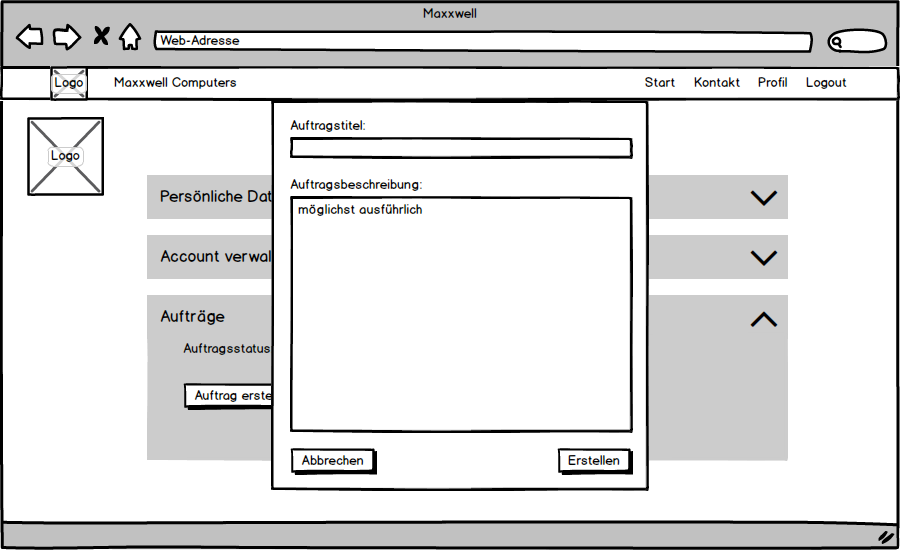


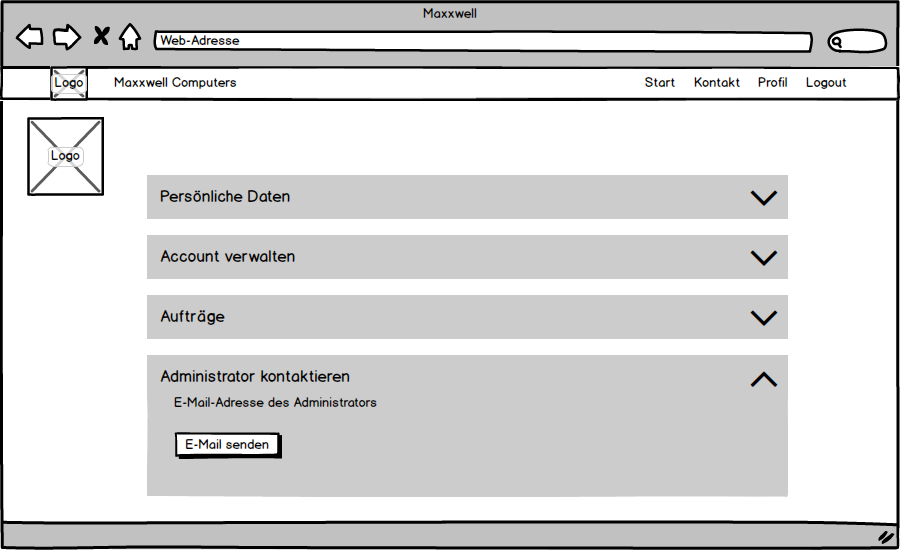
Nach einem Klick auf „Account verwalten“, werden alle ausfüllbaren Felder angezeigt, wobei der Benutzer das Passwort ändern kann und dies mit einem Klick auf „Passwort ändern“ bestätigt wird. Die E-Mail-Adresse kann auch geändert werden und durch einen Klick auf „E-Mail-Adresse ändern“ übernommen werden.

Außerdem hat der Benutzer die Möglichkeit seinen Account endgültig zu löschen. Dies muss doppelt bestätigt werden.



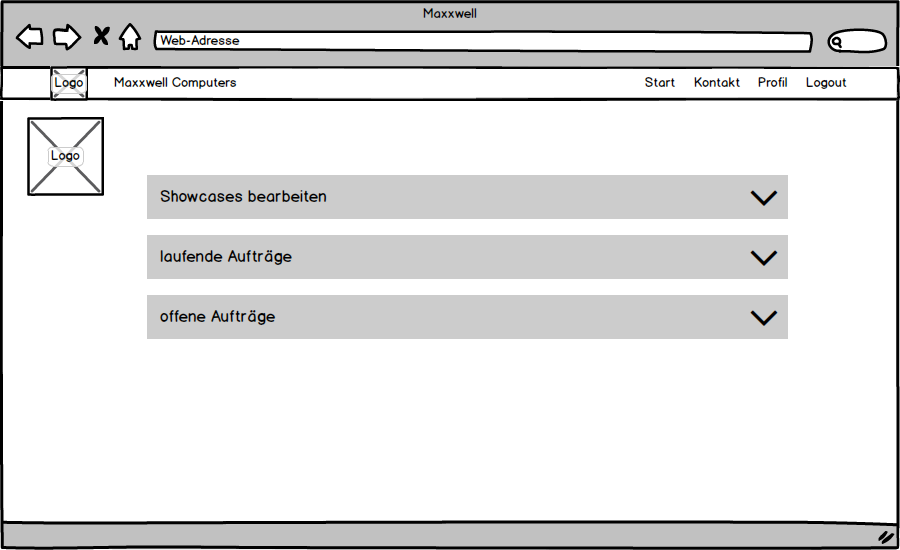
Nach einem Klick auf „Aufträge“, wird dem Benutzer der derzeitige Auftragsstatus angezeigt, sofern dieser einen Auftrag erstellt hat. Durch einen Klick auf „Auftrag erstellen“ wird folgende Eingabemaske gezeigt:



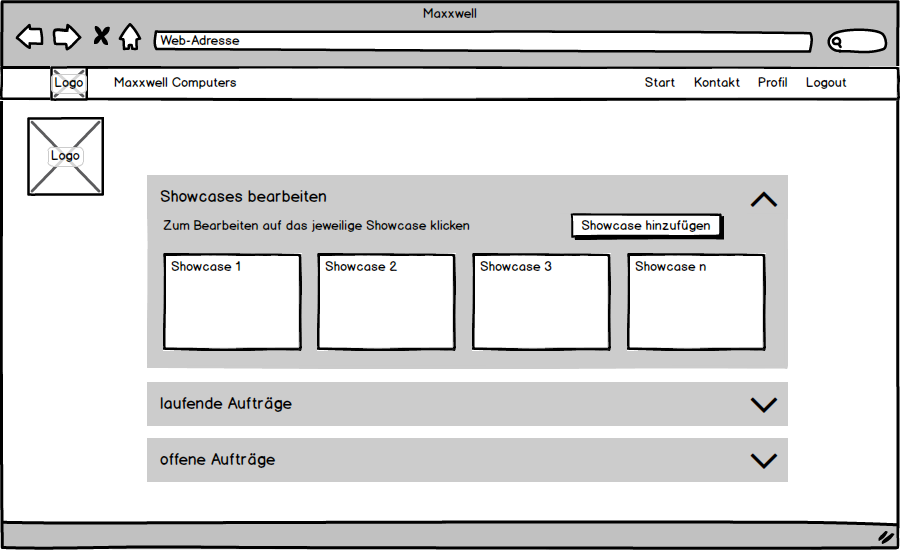
Nachdem alle Felder ausgefüllt wurden und auf „Erstellen“ geklickt wird, erscheint dieser Auftrag beim Administrator, welcher den Auftrag annehmen/ablehnen kann.

Nach einem Klick auf „Administrator kontaktieren“, kann der Benutzer eine E-Mail an den Administrator senden, per Klick auf „E-Mail senden“.

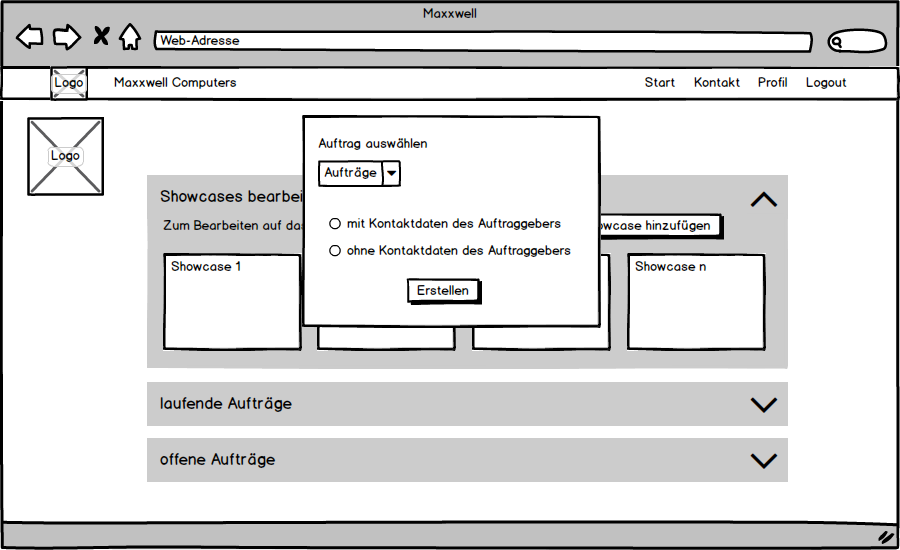
#### Administrator-Profilseite



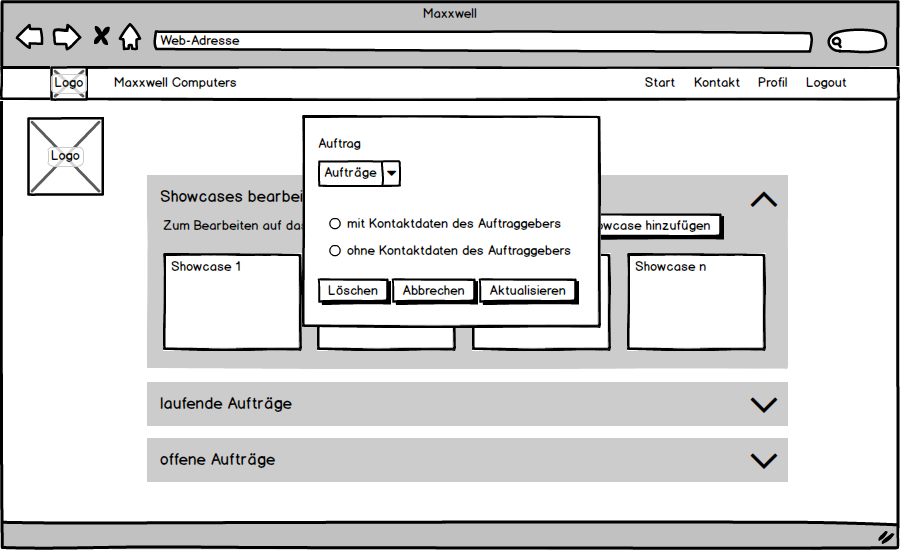
Die Profilseite des Administrators ist in drei Segmente geteilt, welche per Klick auf die Pfeile mehrere Optionen bieten.



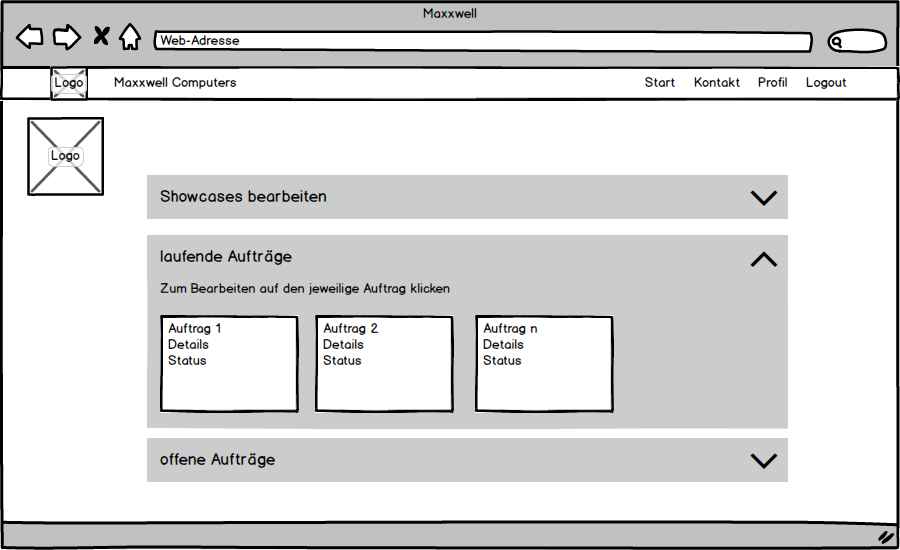
Nach einem Klick auf „Showcases bearbeiten“, kann der Administrator die Showcases, welche auf der Startseite angezeigt werden.

Per Klick auf „Showcase hinzufügen“ öffnet sich folgende Eingabemaske:

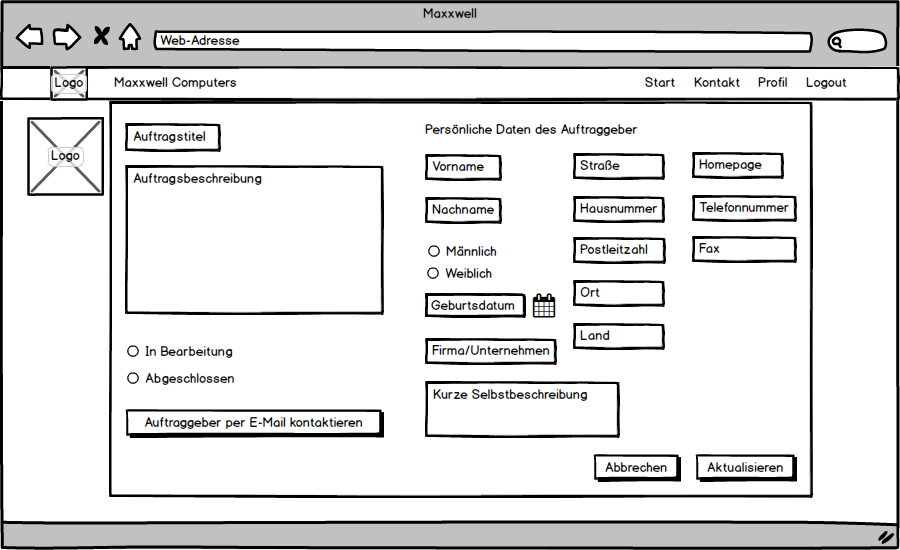
Per Klick auf einen der Showcases kann der Administrator diesen wie folgend bearbeiten:



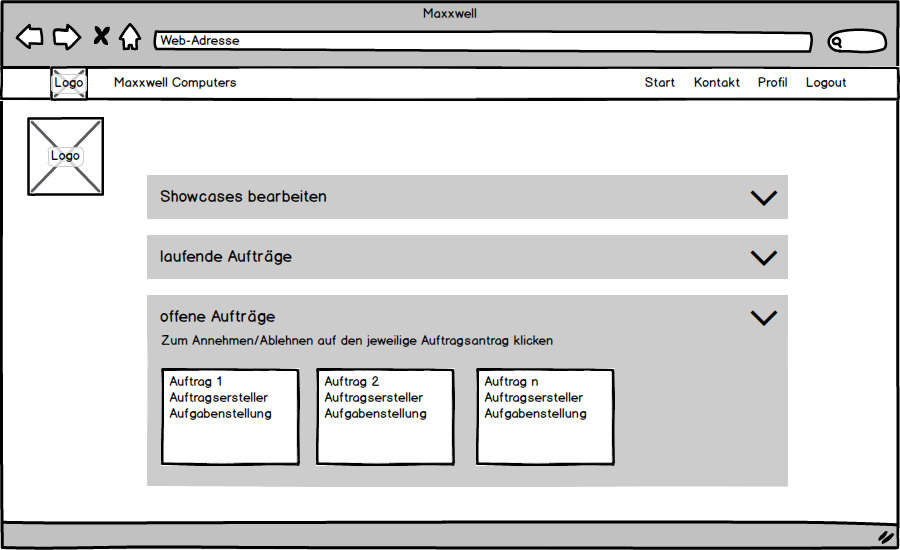
Nach einem Klick auf „laufende Aufträge“ kann der Administrator seine laufenden Aufträge ansehen und bearbeiten.



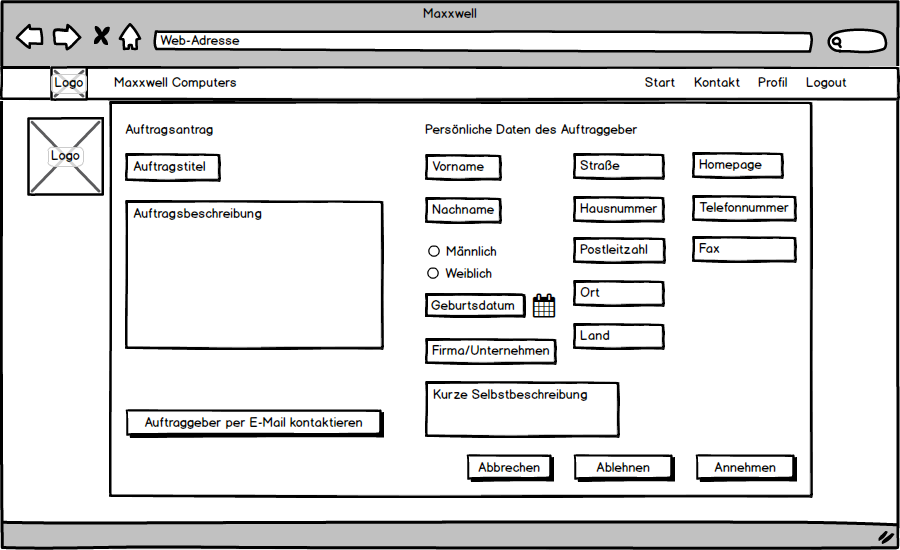
Per Klick auf einen Auftrag hat der Administrator folgende Möglichkeiten:



Nach einem Klick auf „offene Aufträge“, kann der Administrator neue Aufträge einsehen.



Per Klick auf einen dieser Aufträge, kann der Administrator einen Auftrag Annehmen/Ablehnen bzw. den Auftraggeber (Benutzer) direkt per E-Mail kontaktieren:



# Technisches Pflichtenheft

## Daten

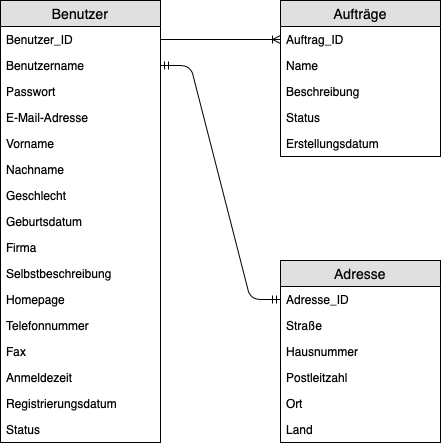
Die Daten (z.B.: Benutzer-Konten) der Webseite werden ausschließlich in der Datenbank gespeichert.

### Datenbank

Als Datenbanksystem wird uns eine MySQL-Datenbank zur Verfügung gestellt.

#### Datenbank-Aufbau

Das folgende ER-Diagramm beschreibt den Aufbau der Datenbank.



#### Datenbank-Zugriff

Der Zugriff erfolgt vorerst durch Ansteuerung des angemieteten Servers, anschließend wird die MySQL-Datenbank mit folgenden Zugangsdaten gesteuert.

* Server: mysqlsvr38.world4you.com
* Datenbank: 3810978db1
* Benutzer: sql1609210
* Passwort: m\*eyicr

### Webmaster-E-Mail

Die Zugangsdaten für neue Benutzer, sowie alle Änderungen, die per E-Mail ausgesandt werden, verwenden folgende Daten:

* Mail: Webmaster1@nerdmania.at
* Passwort: !Pwd4WebM1!

## Aufbau

Das Webprojekt ist in 3 einzelne Teile aufgebaut, nämlich in das Model (enthält Daten wie zum Beispiel die Datenbank, sowie implementiert das Model Funktionen dieser), View (alles Sichtbare des Webprojekts), Controller (die Steuereinheit, implementiert alle Funktionalitäten der Webseite).

### Model

Das Model des Webprojektes stellt die MySQL Datenbank dar. Sie stellt uns alle benötigten Daten bereit, diese inkludieren Benutzerdaten (sowie Anmeldedaten/Kennung, als auch persönliche Daten)

### View

Die Präsentationsschicht stellt unsere Daten für den Benutzer da, sie agiert durch unsere HTML und CSS – Files. Diese bauen nicht nur das Design auf, sondern reagieren auch auf Benutzereingaben und bekommen die Daten vom Controller

### Controller

Der Controller beinhaltet die Logik, basierend auf PHP, sowie JS – Files, welche die Daten aus der Datenbank holen und an die View zur Anzeige weiterschicken.

## Tests

Für bestimmte Abschnitte des Webprojekts werden Tests implementiert, dies beinhaltet eine problemlose Anwendung aller Funktionen für den Benutzer, sowie für den Administrator der gesamten Webseite.

Weiteres wird protokolliert ob bei falscher Eingabe, der Anwender, mittels Fehlermeldungen über eine inkorrekte Eingabe benachrichtigt wird.

Außerdem ist ein weiterer wichtiger Punkt der Tests, die Überprüfung der Richtigkeit der eingegebenen Daten, genauer gesagt die Überprüfung ob die Daten auch richtig von der View (Eingabe des Benutzers) in der Datenbank gespeichert werden.

## Server/Client - Beschreibung

Dieser Abschnitt handelt von der Beschreibung der Kommunikation zwischen Client und Server.

### Server

Folgend wird der verwendete Server genauer beschrieben. Der Server wird außerdem vom Auftraggeber bereitgestellt.

#### Hardware

Die Umsetzung erfolgt mittels eines Domainservers, welcher bei der Webseite world4you.com vom Auftraggeber angemietet wurde.

Die Domain lautet [www.nerdmania.at](http://www.nerdmania.at), diese wurde vom Auftraggeber eingerichtet und unter dieser Adresse soll das Webprojekt installiert werden.

* Datenbankserver
  + SQL-Server mit MySQL (neueste Version)
* Mailserver
  + POP3-, sowie IMAP-Protokoll können verwendet werden
* Webserver
  + 24/7 FTP-Login
  + Volle PHP-Unterstützung
  + MySQL-Implementierung
* Weitere Komponenten/Leistungen
  + 50GB Webspace
  + Dauerhafter Firewall-Schutz
  + 24/7 Internetanbindung

#### Sicherheit

Beschreibung der Möglichkeiten, um die Webseite vor potenziellen Angreifern zu schützen.

Es werden bereits bestimmte Zugriffe durch die Firewall von world4you-Webhosting bereits herausgefiltert.

Zusätzlich wird geschützt vor:

* SQL Injections
  + Benutzer-Accounts werden mit limitierten Rechten versehen
* Verwendung von Prepared Statements mit gebundenen Variablen. Diese werden von MySQL angeboten.
* Prüfen der Eingabe
  + Die Eingabe wird darauf überprüft, ob sie dem erwarteten Datentyp entspricht.

### Client

Der Client benötigt im Prinzip nur eine Internetverbindung, sowie Zugang zu einem funktionsfähigen Computer.

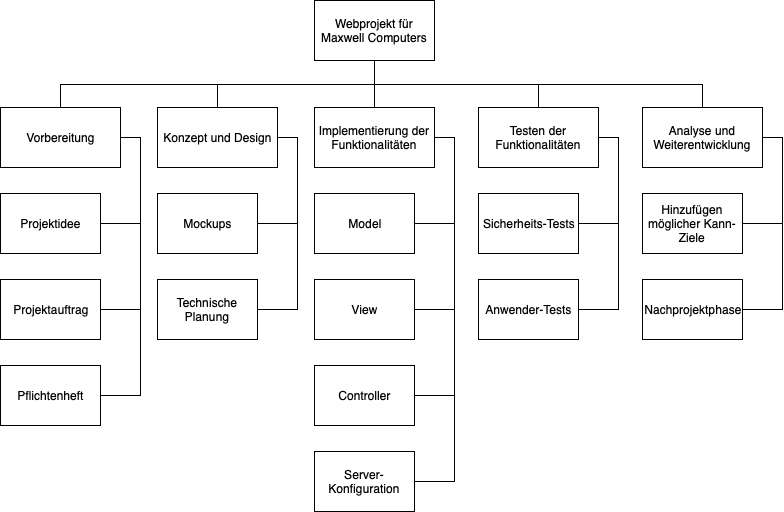
Außerdem wird ein Browser benötigt (Anforderungen, siehe Abschnitt „Pflichtenheft“), damit die URL der Webseite eingegeben werden kann.

Die Clientseitige Skriptsprachen sind HTML, CSS, sowie JavaScript.

Um weitere Funktionen der Webseite nutzen zu können, wird ein Benutzer-Account vorausgesetzt, dieser benötigt zur Erstellung eine E-Mail-Adresse und ein selbst gewähltes Passwort.

## Projektstrukturplan

Der folgenden Projektstrukturplan (PSP) dient zur groben Übersicht des Projektverlaufs. Wobei das Ziel dieses ist, eine einheitliche Strukturierung der erledigten bzw. noch zu erledigen Aufgaben zu bekommen.



# Theoretische Grundlagen

In diesem Abschnitt werden alle theoretischen Grundlagen dieser Arbeit dargestellt und erklärt, um das Wissen und die Nachvollziehbarkeit aller Schritte, die zu dieser Arbeit führten, zu bekommen. Folgend eine allgemeine Erklärung zu den verschiedenen Technologien und Konzepten.

## Eingesetzte Technologien

Im folgenden Kapitel werden die, in dieser Arbeit verwendeten Technologien allgemein erklärt und Vergleiche zu anderen ähnlichen Technologien aufgestellt.

### HTML – Hypertext Markup Language

HTML steht für Hypertext Markup Language und ist eine auf Text basierende Markup Language oder auch Auszeichnungssprache zur Strukturierung von elektronischen Dokumenten wie zum Beispiel Text mit Bildern oder auch Hyperlinks. Heutzutage basieren alle Webseiten auf HTML. Webbrowser erhalten HTML Dokumente entweder von einem Webserver oder aus dem lokalen Speicher und geben diese in Form von Webseiten wie man sie in einem Webbrowser kennt wieder. Hierbei beschreibt HTML nur die Strukturierung, aber in Kombination mit anderen Technologien wie CSS (Cascading Style Sheet) und Javascript ist es möglich die Darstellung und das Verhalten zu erweitern.

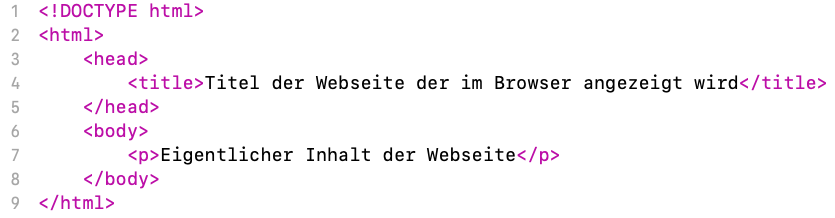
#### Struktur

Ein HTML Dokument besteht aus HTML Elementen wie Konstrukte, Bilder oder interaktive Formulare. Die meisten Elemente werden mittels eines Tag-Paar markiert, das heißt durch einen Tag am Anfang und einem am Ende. Der Anfangs-Tag beginnt mit einem “<“ und endet mit einem “>“, wobei der Endtag mit einem “</“ anfängt und einem “>“ endet. Ein vollweritges HTML Element würde dann folgendermaßen aussehen <a></a>. Aber zusätzlich ist es möglich das andere HTML Elemetne innerhalb von anderen Elementen sind, was die erweiterte Strukturierung von HTML begründet.

Weiters kann ein HTML Dokument in 3 Bereiche eingeteilt werden:

* Die Dokumentdekleration (DOCTYPE) Diese steht ganz am Anfang eines HTML Dokuments, da sie unter anderem auch die Dokumenttypdefinition definiert, heutzutage zum größten Teil HTML5
* Der HTML-Kopf (HEAD) Dieser enthält hauptsächlich Information wie technische und dokumentarische Definierung, die der Browser nicht anzeigt
* Der HTML-Körper (BODY) Dieser besteht aus den Informationen, die der Browser anzeigt

Ein standardisiertes HTML Dokument könnte also folgendermaßen aussehen:



Beispiel-Code für ein HTML-Dokument

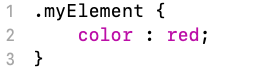
### CSS – Cascading Style Sheets

Cascading Style Sheets oder auch kurz CSS ist eine Stylesheet Sprache für elektronische Dokumente und gehört zu einer der Kernsprachen des Internets und ist ein sogenannter “living standard“. Es wird angewendet um die Darstellung von Auszeichnungssprachen wie entweder HTML- oder XML-Dokumenten zu verändern.

#### Aufbau

Eine CSS-Anweisung gibt an für welche Teile eines Dokumentes gewisse Eigenschaften oder eine Kombination von Eigenschaften gelten soll. In HTML Dokumenten ist es möglich diese Teile einzugrenzen, mit der Einführung von Klassen oder IDs. Durch den dadurch eindeutigen Namen den man im Falle von Klassen auf mehrere HTML-Elemente anwenden kann, und ID auf nur ein Element, kann man eigens bestimmten Elemente bestimmte Eigenschaften geben.

Eine CSS Anweisung besteht aus einem oder mehreren Selektoren die sich entweder auf ganze HTML Elemente, Klassen (Regel muss mit “.“ beginnen) oder IDs (Regel muss mit “#“ beginnen) bezieht. Ein Selektor bestimmt die Bedingungen, die auf ein Element zutreffen müssen um die nachfolgende Spezifizierung geltend zu machen.



Beispiel-Code für ein CSS-Dokument

Der folgende Code-Ausschnitt zeigt eine Regel, welche bei allen HTML Elementen, die mit der Klasse “myElement“ gekennzeichnet sind, die Schriftfarbe auf Rot setzt.

### JS – JavaScript

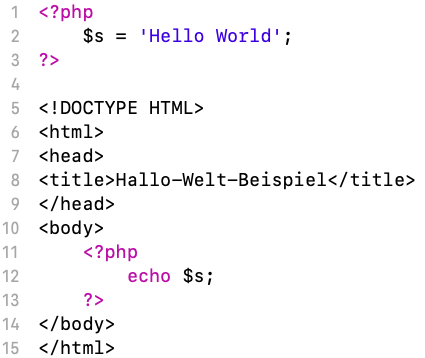
Kurz JS oder Javascript ist eine Skriptsprache, die es ermöglicht auf Webseiten dynamische Inhalte darzustellen. Sie wurde entwickelt um die Funktionalitäten von HTML und CSS noch mehr zu erweitern.

JS wird innerhalb des Browser in einem sogenannten Sandbox-Prinzip ausgeführt, das heißt, dass kein Zugriff auf Daten außerhalb des Browser möglich ist, aber auch, dass Cross-Site-Scripting verhindert wird, sprich jede Webseite wird nur mit sich selber isoliert, wäre das nicht der Fall, wäre es möglich Daten von anderen Webseiten wie zum Beispiel Login-Information auszulesen. Standardmäßig wird für jedes Skript auf einer Webseite ein einziger Thread geöffnet, daher gilt es Warteschleifen oder längere Berechnungen zu vermeiden. Um die eigene Sicherheit zu erweitern ist es außerdem möglich Javascript überhaupt zu deaktivieren. Jedoch kann es dann dazu kommen, dass bestimmte Funktionalitäten nicht mehr korrekt ausgeführt werden können.

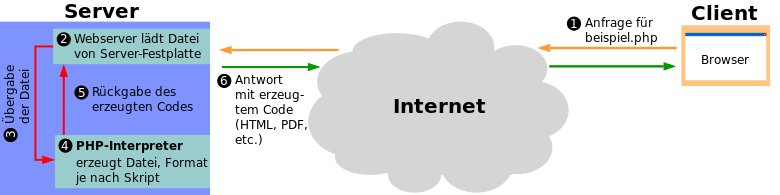
### PHP – Hypertext Preprocessor

PHP ist eine Skriptsprache, das heißt es wird meist für kleine Programme verwendet und über einen Interpreter ausgeführt, dessen Syntax ähnlich der von den Programmiersprachen C und Perl ist.

Verwendet wird PHP hauptsächlich zur Erstellung von Webanwendungen und dynamischen Webseiten, sowie Elementen in Webseiten. Einer der größten Vorteile, welcher in dieser Arbeit viel genutzt wird, ist die breite Datenbankunterstützung, dadurch wird es ermöglicht zum Beispiel SQL-Abfragen direkt im PHP zu schreiben und dann an die Datenbank zu senden, wo diese umgesetzt werden. PHP-Code kann außerdem innerhalb von HTML-Code verwendet werden.



PHP-Code Beispiel integriert in HTML.



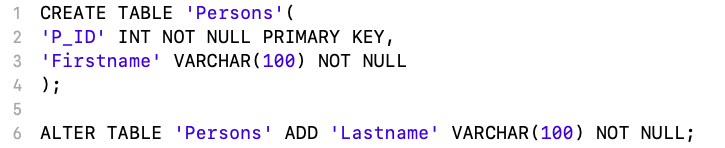
Funktionsweise von PHP

### SQL – Structured Query Language

SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanksystemen. Mit dieser Datenbanksprache werden auch Datensätze in Datenbanken modifiziert, dies inkludiert das Hinzufügen, Löschen und Bearbeiten von neuen bzw. bestehen Datensätzen. Diese Funktionen sind gegliedert und werden durch die drei folgenden Begriffe abgebidlet.

#### DDL – Data Definition Language

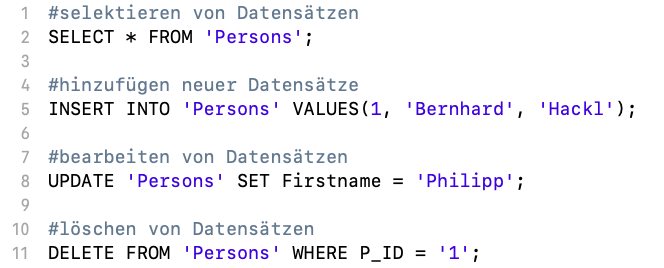
Die DDL ist zuständig für das Anlegen von Datenstrukturen und das Beschreiben von verwandten Elementen.



Code-Beispiel für DDL-Abfragen.

#### DML – Data Manipulation Language

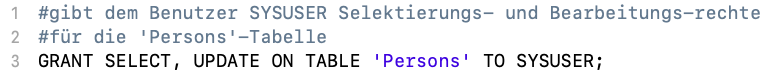
Die DML ist zuständig für alle Abfragen, die sich auf Datensätze innerhalb einer Datenbank befinden und kann diese bearbeiten, löschen, sowie auch neue Datensätze hinzufügen. Die Hauptaufgabe ist es meistens Daten auszuwählen.



Code-Beispiel für DML-Abfragen

#### DCL – Data Control Language

Dieser Teil der SQL-Sprache verwaltet Rechte von Benutzern auf Datenbanken bzw. einzelne Tabellen.

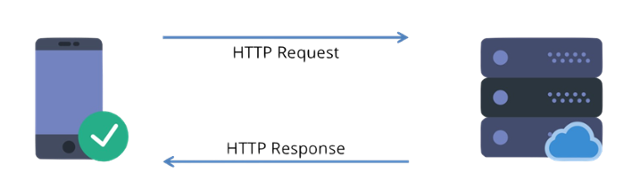


Code-Beispiel für DCL-Abfragen

## Kommunikation

Für die Bereitstellung der Verbindung zwischen mehreren Geräten müssen diese vorerst miteinander kommunizieren können. Dieses Kapitel beschreibt mehrere Möglichkeiten um eine solche Kommunikation umzusetzen.

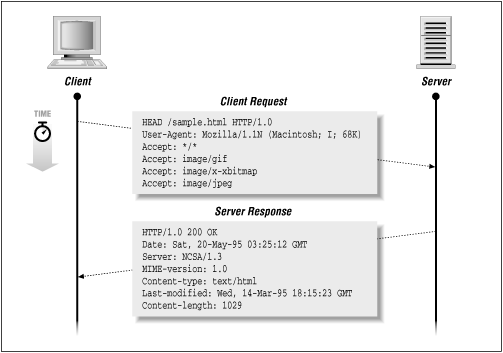
### HTTP – Hypertext Transfer Protocol

Das HTTP-Protokoll ist ein zustandsloses Protokoll, das bedeutet es kann mehrere Anfragen als voneinander unabhängige Transaktionen zu behandeln, zur Übertragung von Daten auf der Anwendungsschicht. Als Haupteinsatz zählt das Laden von Webseiten aus dem Internet in einen Webbrowser.

HTTP-Protokoll-Modell

Es kommunizieren Server und Client mit sogenannten „Nachrichten“ miteinander werden dabei, wobei es folgende zwei verschiedenen Arten gibt.

* Anfrage (Request), ist die Nachricht, die ein Client an den Server sendet
* Antwort (Response), ist die Nachricht, die ein Server den Client nach einer Anfrage zurücksendet



HTTP-Request/Response Beispiel

#### Anfragemethoden und Statuscodes

Anhand der verschiedenen Anfragemethoden weiß der Server, welche Funktionalität er erbringen soll bzw. wie er mit der Anfrage umgehen muss. Folgende Anfragemethoden werden häufig benötigt.

* **GET**

GET ist die am häufigsten verwendeten Methode, mit welcher Daten über die URI abgerufen werden können, wobei die Länge des URI begrenzt ist.

* **POST**

POST schickt unbegrenzt Daten an den Server, welche als Inhalt dieser Nachricht übertragen werden. Dadurch können neue Ressourcen auf dem Server entstehen oder bestehende modifiziert werden.

* **PUT**

Lädt eine Ressource auf den Server hoch unter Angabe dessen Ziel-URI, besteht diese bereits auf dem Server, so wird sie ersetzt.

* **HEAD**

Weist den Server an die gleichen HTTP-Header wie bei GET zu verwenden, jedoch keine Daten mittels des Nachrichtenrumpfes zu senden.

* **PATCH**

Ändert Teile einer Ressource, ohne diese vollständig zu ersetzen, wie bei PUT.

* **DELETE**

Löscht die angegeben Ressource auf dem Server.

Der Server schickt in seiner Antwort (Response) zusätzlich einen Statuscode mit, dieser beinhaltet weitere Informationen. Folgende wenige wichtige Statuscodes sollten bekannt sein.

* 100/Continue

Die laufende Anfrage wurde von dem Server noch nicht zurückgewiesen.

* 200/Ok

Die Anfrage wurde erfolgreich bearbeitet. Das Ergebnis wird in der Antwort übertragen.

* 200/Created

Die Anfrage wurde erfolgreich bearbeitet und die Ressource wurde auf dem Server erstellt.

* 300/Multiple Choices

Die angeforderte Ressource steht in verschiedenen Arten zur Verfügung.

* 400/Bad Request

Die Anfrage war nicht korrekt aufgebaut.

* 403/Forbidden

Die Anfrage kann aufgrund von fehlender Berechtigung nicht durchgeführt werden.

* 404/Not Found

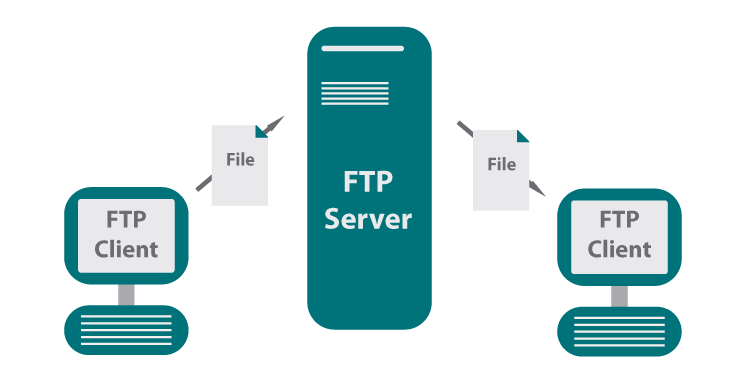
Die Anfrage zeigt auf eine Ressource, welche auf dem Server nicht existiert.

* 500/Internal Server Error

Sammelstatuscode für einen internen Serverfehler.

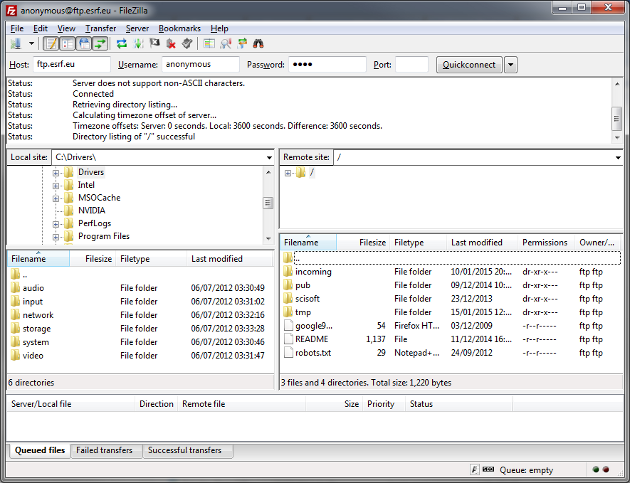
### FTP – File Transfer Protocol

FTP ist ein zustandsbehaftetes Protokoll zur Übertragung von Daten in einem IP-Netzwerk. Es wird hauptsächlich dafür benutzt Dateien auf einen Server hochzuladen (File-Upload) oder von einem Server herunterzuladen (File-Download). Standardmäßig verwendet FTP den Port 21 für die Datei-Übertragung.



FTP-Server/Client-Beispiel

Für die Übertragung wird außerdem ein FTP-Client benötigt.



FTP-Client FileZilla als Beispiel

## Benutzerschnittstelle

Die Benutzerschnittstelle, auch User Interface genannt, ist vereinfacht gesagt die Verbindung zwischen Mensch und Maschine.

Das User-Interface für Software ist häufig die graphische Oberfläche einer Anwendung, in diesem Fall das Frontend einer Webseite. Per Mausklick oder Eingabe der Tastatur greift die Webseite aus dem Frontend die Eingabe bzw. den Mausklick und schickt diese zur Verarbeitung an das Backend weiter, welches damit umgeht (zum Beispiel Daten aus der Datenbank holen) und die Daten an das Frontend zurückgibt.

### Anwenderbezogenes Design

Das sogenannte Design ist meist ausschlaggebend für den Erfolg einer Anwendung. Zum Beispiel spielt das Design bei Webseiten eine sehr große Rolle, da eine saubere und übersichtlich gestaltete Webseite eher den Benutzer dazu antreibt auf dieser zu bleiben und nicht einfach die nächste Webseite sucht.

#### Zielgruppe

Wichtig bei der Auswahl eines guten Designs ist es seine Zielgruppe zu kennen und auf die Bedürfnisse dieser einzugehen.

#### Inhalt/Content und Aufbau

Der Inhalt einer Webseite ist an die jeweilige Zielgruppe gebunden, jedoch sollte man dem Benutzer keine „Steine in den Weg legen“, also diesen nicht mit unnötigem Inhalt, welcher nicht genau zu dem Thema der Webseite gehört, aufzuhalten.

Noch dazu sollten Wert auf Reinheit und Übersichtlichkeit gelegt werden, komplexe und große Menüs mit vielen Auswahlmöglichkeiten könnten den Benutzer überfordern.

### Technologien

Verschiedene Technologien kommen zum Einsatz, wenn es um die Benutzerschnittstelle geht, dies reicht von einem einfachen Lichtschalter, bis zu Applikationen auf Smartphones.

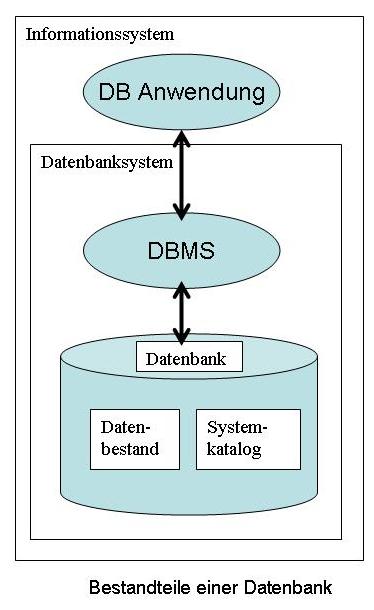
## Datenverwaltung

In diesem Abschnitt werden die verwendeten Komponenten zur Datenverwaltung näher beschrieben.

### Datenbank

Die Datenbank ist ein elektronisches System zur Verwaltung von meist besonders großen Datenmengen. Die Datenbestände können dann aus verschiedenen Teilmengen zusammengestellt und auf der Anwenderschicht dargestellt werden.

Die Datenbank selbst beinhaltet den Datenbestand, wobei das Datenbankmanagementsystem, kurz DBMS diesen verwaltet und für den Anwender unsichtbar ablegt. Zu den Funktionen des DBMS gehören neben dem Speichern, Bearbeiten und Löschen von Daten auch Vorkehrungen für Datenschutz, -sicherheit und -integrität, sowie die Verwaltung von Metadaten. Folgende Grafik dient zur Veranschaulichung des Datenbanksystems.



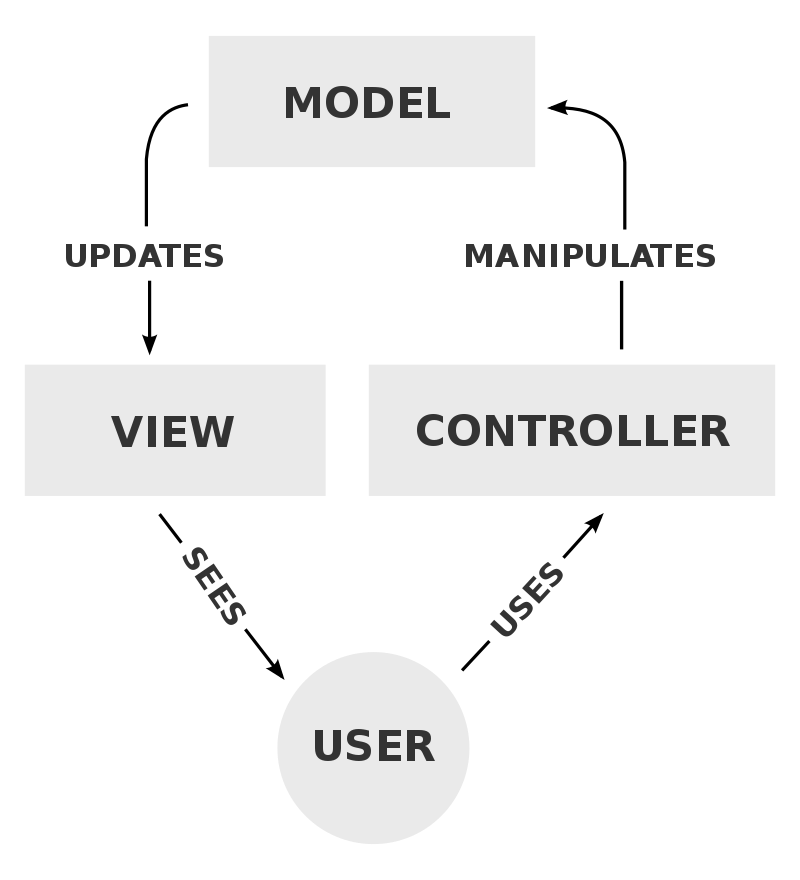
Bestandteile des Datenbanksystems

## Softwarekonzepte

In diesem Unterkapitel werden die verwendeten Software-Konzepte erklärt. Da es sich hierbei um ein Webprojekt handelt, wird das Hauptaugenmerk auf Softwarekomponenten der Webentwicklung gelegt.

### MVC – Model-View-Controller Pattern

Da es sich bei einer jeden interaktiven Anwendung um Eingabe, Auswertung und Anzeige von Daten handelt, ist das Ziel des Model-View-Controller-Ansatz, kurz MVC-Pattern, die Trennung von Eingabe, Verarbeitung und Repräsentation der Daten. Die folgende Grafik zeigt eine Übersicht des MVC-Patterns.



Aufbau des MVC-Patterns

* Das Model enthält dabei nur die Daten und dessen Verarbeitung und ist daher von der Darstellung auf dem Bildschirm des Benutzers vollkommen unabhängig.
* Die View ist für die Darstellung der aktuellen Ausgaben und Eingaben der Datenbestände verantwortlich.
* Der Controller verwaltet die Steuerung der Anwendung, dabei überwacht dieser alle Eingaben und übergibt die Eingabedaten weiter, damit bildet der Controller gemeinsam mit der View die Benutzeroberfläche.

# Lösungsansätze für die Realisierung

In diesem Kapitel werden mehrere Lösungsansätze, die sich im Vorhinein überlegt wurden, für die Realisierung des Webprojektes mit Vor- und Nachteilen beschrieben.

Für das Frontend trat im Laufe des Projektes nur die Entscheidung zwischen der Verwendung des Frameworks Bootstrap oder Materialize-CSS, diese wird im Detail in Kapitel 6 erläutert.

## Lösungsansatz 1: Verwendung von Laravel

Laravel ist ein freies Framework für die Skriptsprache PHP und beinhaltet eine große Menge an Features. Außerdem ist Laravel sehr einfach zu verstehen, robust und verfolgt den MVC-Ansatz (siehe Kapitel 4.5.1).

Die Vorteile der Verwendung dieses Frameworks beinhalten eine verbesserte Skalierung der Webapplikation, Zeitersparnisse und ein vereinfachtes Ressourcenmanagement. Einer der jedoch wichtigsten Vorteile sind die Sicherheitsfeatures bezüglich Sessions, Passwort-hashing und SQL-Prepared Statements, mehr dazu in Kapitel 6.

## Lösungsansatz 2: Plain PHP

Dieser Ansatz verfolgt der Programmierung in reinem PHP ohne Verwendung eines Frameworks. Bezüglich dieser Aufgabenstellung, wobei es sich um eine private Webseite handelt, welche viele individuelle Anforderungen besitzt, ist die Verwendung eines Frameworks nicht notwendig.

Die Vorteile beinhalten hohe Individualität, sowie ein jedes Framework seine Grenzen hat, werden diese mit diesem Lösungsansatz eliminiert. Weiters nimmt ein Framework dem Programmierer meist einen Großteil an Arbeit ab, ohne der Verwendung bekommt der Entwickler ein besseres Verständnis über sein Projekt.

## Wahl des Lösungsansatzes

Die Entscheidung für den Lösungsansatz 2 (siehe 5.2) stützte sich auf folgenden Fakten und entscheidenden Vorteilen:

* Freiheit der Wahl der Problemlösung (Authentifizierung, Validierung)
* Besseres Verständnis der fundamentalen Funktionen von PHP
* Aufgrund der Größe des Projektes ist kein Framework unbedingt notwendig
* Sicherheit trotzdem gegeben (siehe Kapitel 6)
* Viele individuelle Anforderungen (Benutzerkommunikation, Aufgabenverwaltung, spezielle Administratorfunktionen) erfordern Lösungen, welche nicht von einem Framework abgenommen werden können
* Verbesserte Validierung der individuellen Funktionen

# Dokumentation der Realisierung

In diesem Kapitel wird die Umsetzung der Diplomarbeit dokumentiert, um einen besseren Einblick in diese zu erhalten. Die folgenden Unterkapitel bestehen teilweise aus Codeausschnitten und Erklärungen für ein laienhaftes Verständnis.

Aufgrund von der eingesetzten MVC-Gliederung, wird diese durch die folgenden Kapitel weiter verwendet für ein besseres Verständnis.

## Model

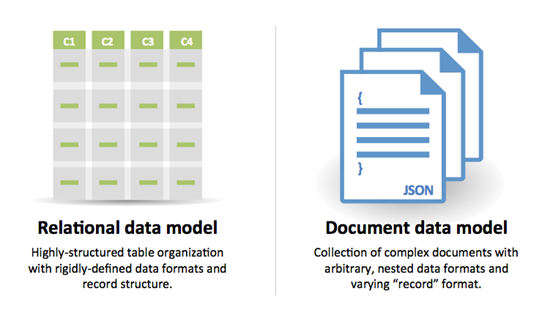
Das Model dieser Arbeit umfasst jegliche Funktionen von der Datenbank bis hin zu dieser selbst.

### Wahl des Datenbanksystems

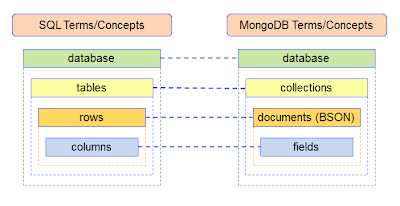
Als Technologie wurde für die Datenbank MySQL verwendet. Der Grund für diese Entscheidung wird anhand folgenden Vergleichs näher erklärt.

|  |  |
| --- | --- |
| **SQL (MySQL)** | **NoSQL (MongoDB)** |
| + relationale Datenbank | + nicht-relationale Datenbank (dokumentenbasierend) |
| + MySQL populärste Open-Source SQL-Datenbank | + Einsatz vor allem bei Anwendungen mit sehr großen Datenmengen |
| + Einsatzgebiet überwiegend Webanwendungen | + verbesserte Skalierbarkeit |
| + herausragende Stabilität | + mehr Flexibilität |
| + hohe Sicherheit | + bessere Performance |
| + einfache Administration |  |

Die folgenden Grafiken veranschaulichen deutlich die Unterschiede zwischen SQL- und NoSQL-Datenbanken.



Relationale Datenbankmodell und dokumentenbasiertes Datenbankmodell



SQL vs MongoDB (NoSQL)

Die Speicherung bei SQL-Datenbanken erfolgt, vereinfacht gesagt, in einer Tabellenform, dabei wird ein jeder voller Eintrag als eine Zeile, eine sogenannte „Row“ dargestellt, die „Columns“ bilden dabei einzelne Datensätze ab.

Bei NoSQL-Datenbanken wie MongoDB erfolgt die Speicherung in Dokumenten, dabei beinhalten sogenannte „Collections“ die Dokumente.

Ein vereinfachtes persönliches Beispiel wäre, wenn ein Arzt seine Patientenakten in mehreren Aktenschränken sucht. Bei einer SQL-Datenbank hätte er Schränke mit „Adressen“, „Versicherungen“, „Namen“, und so weiter, dies bedeutet, dass der Arzt nun anhand eines Primary Keys durch jeden Aktenschrank gehen müsste, um alle Daten für einen Patienten zu bekommen. Bei einer MongoDB beispielsweise, hätte er Schränke mit Patientennamen „A-H“, „H-M“, „M-Q“, und so weiter, wobei eine jede Akte dann ein Dokument in der Datenbank darstellt, so hat er alle Akten (Adressen, Versicherungen, Namen) zu einem Eintrag (Patient).

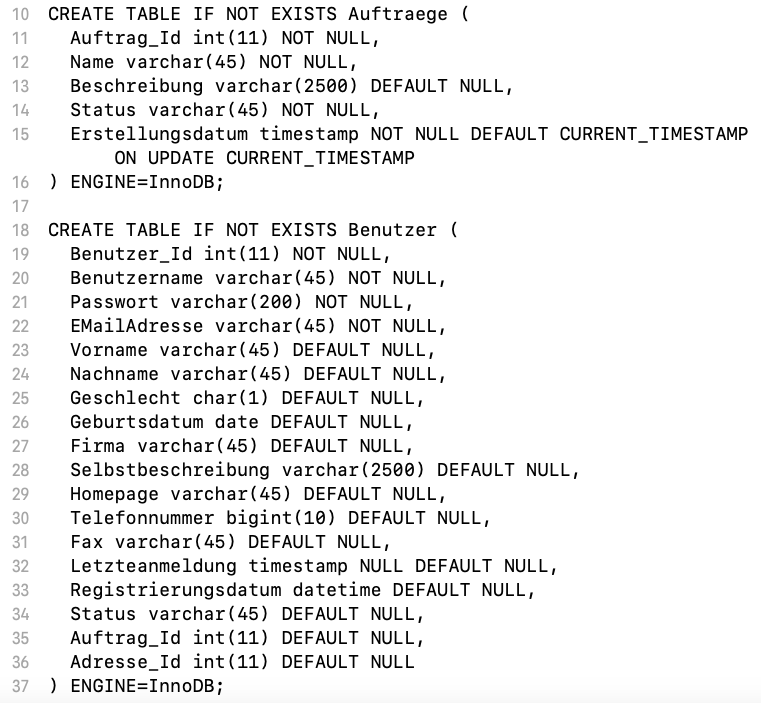
Würde man also eine Menge an verschiedenen Daten zu Personen haben, so wäre die Verwendung einer NoSQL-Datenbank eindeutig die bessere Lösung, da in dieser Arbeit nur Benutzer mit Adressen und Aufträgen existieren, also keine JOINS über viele Tabellen erfolgen, so kann genauso gut eine SQL-Datenbank verwendet werden.

Die Hauptgründe für unsere Entscheidung für eine SQL-Datenbank, waren neben den oben bereits angeführten Vorteilen, vor allem die erleichterte Administration. MySQL bietet beispielsweise das Tool „phpMyAdmin“, welches direkte Imports von SQL-Datenbanken erlaubt und diese genauso auch exportieren kann, was in der Folge der Implementierung bei dem Deployment einen großen Vorteil brachte. Das direkte Hinzufügen von Datensätzen, sowie das Absetzen und Testen von Queries konnte uns einige Fehler verhindern.

### Datenbankschema

Das Schema wurde aus dem ER-Diagramm der technischen Planung entnommen und im Laufe der Implementierung geringfügig erweitert.

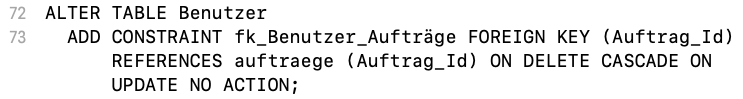
Es wurde ein SQL-Skript erstellt, welches die Datenbank selbst, sowie alle Tabellen, deren Beziehungen und Schlüssel enthält. Dieses SQL-Skript kann danach einfach importiert und zu Testzwecken verwendet werden. Die Tabellen für Benutzer und dessen Aufträge würden folgendermaßen aussehen.



SQL-Code für Benutzer und Aufträge

Da es natürlich passieren kann, dass ein Benutzer seinen Account löschen möchte, eine Funktionalität, welche in dem Benutzerprofil implementiert wurde, so muss von der Datenbank sichergestellt werden, dass auch alle Aufträge, die von diesem Benutzer ausgingen, ebenfalls gelöscht werden, da diese nur unnötige Datenreste in der Datenbank abbilden würden.

Der folgende Code eliminiert dieses Problem, indem mit dem „On Delete Cascade“ Befehl ein gelöschter Benutzer, automatisch auch alle Aufträge entfernt, die mit dem Foreign-Key, namens „Auftrag\_Id“ des Benutzers referenziert sind.



SQL-Code für On-Delete-Cascade-Eigenschaft

### Datenbankverbindung

Die Verbindung zu der Datenbank muss immer hergestellt sein, damit Änderungen, wie das Hinzufügen von Benutzern, in dieser gespeichert werden kann. Das File zur Erstellung der Verbindung wurde mit der Skriptsprache PHP erstellt, wobei die Verbindung durch eine Abstraktionsebene, namens PHP Data Objects, kurz PDO, erfolgt. Diese Abstraktionsebene ermöglicht einen einheitlichen Datenbankzugriff und würde den Umstieg auf eine andere SQL-Datenbank erleichtern.

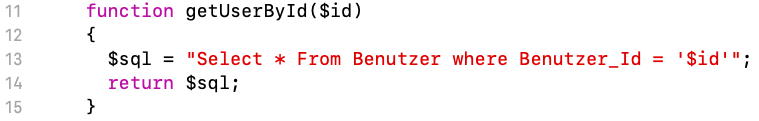
Zu Testzwecken wurde während der Implementierung, nicht direkt an der Datenbank des bereitgestellten Servers gearbeitet und stattdessen auf einer lokal eingerichteten, der folgende Code zeigt einen Teil der Verbindung zu dieser.



Code-Ausschnitt der PHP-Verbindung zur Datenbank

### Datenbankoperationen

Aufgrund von der Trennung des MVC-Konzepts, wurden alle SQL-Befehle in eine PHP-Datei geschrieben, welche alle notwendigen SQL-Queries in Methoden beinhaltet. Jegliche individuellen Werte und Daten, welche für die Datenbank benötigt werden, wurden als Parameter definiert. Der folgende Code-Ausschnitt zeigt ein Beispiel einer solchen Methode, welches alle Daten eines Benutzers aufgrund seiner einmaligen Id zurück liefert.



## View

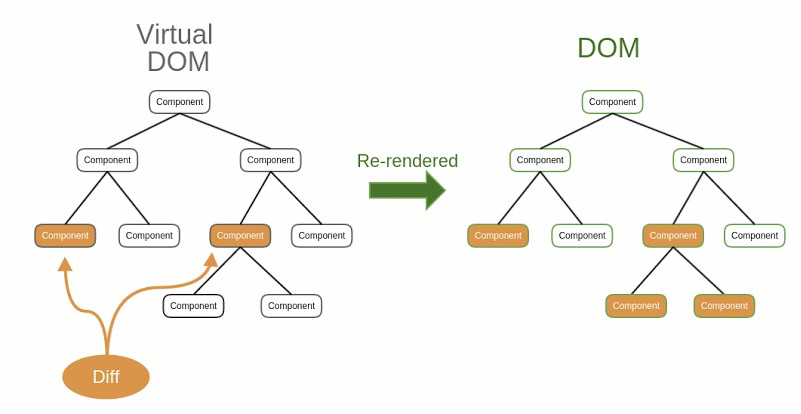
Die View ist eine Darstellung der Daten aus dem Model, und in unserem Fall die HTML-Seite in Kombination mit CSS und JavaScript.

### Wahl eines Frameworks

Da Bereits die Entscheidung getroffen wurde, eine klassische SQL Datenbank zu verwenden, und keine NoSQL-Datenbank, ist die Auswahl eingeschränkt worden auf Frontend Frameworks wie React und materializecss. Außerdem war es ein Ziel, die gesamte Webseite in dem weitverbreiteten und schlankem Material Design zu halten, und beide genannten Frameworks bieten eine Möglichkeit, dies zu implementieren. Folgendes Diagramm hat die Entscheidung erleichtert.

Wie sich herausstellte, sind sich React und Materializecss, abgesehen von der Deklarativen Sprache von React und eines virtuellem DOM (Document Object Model) , sehr ähnlich. Nun stellt sich die Frage, ob ein virtuelles DOM überhaupt sinnvoll für unseren Anwendungszweck ist. Ein virtuelles DOM ist eine Abstraktion des normalen DOM, also eigentlich eine Abstraktion einer Abstraktion. Folgendes Bild hilft zum Verständnis.

Die in einem DOM enthaltenen Komponenten sind in einer baumförmigen Struktur abgespeichert, und will man auf ein einzelnes Element zugreifen, muss oft ein langer Weg gegangen werden. Um dies zu vermeiden, erzeugt React wie bereits erwähnt immer ein virtuelles DOM.



Dieses besteht idealer Weise nur aus den wichtigsten Komponenten, die nach dem folgenden Prinzip durch einen Algorithmus im virtuellem DOM abgespeichert werden:

1. Es werden die Unterschiede zwischen DOM und virtuellem DOM ermittelt (Diff)

2. Die Veränderungen gleichzeitig im DOM und im virtuellem DOM angepasst (Re-render)

Und da nur mit dem virtuellem DOM gearbeitet wird und der Algorithmus von der Implementierung abhängt, sind alle Operationen am DOM schneller.

Doch in unserem Anwendungsfall würden wir kaum bis gar nicht von all den Features profitieren können. Also fiel die Wahl aufgrund von den gelisteten Eigenschaften und dem eben Beschriebenen auf materializecss.

### Einbindung

Das materializecss Framework besteht aus 2 Komponenten, zum einem aus dem CSS Dokument, und dem JavaSccript Dokument. Beide Dokumente können ganz leicht entweder über den CDN link, oder als lokales Dokument, welches auf der Dokumentationswebseite zur Verfügung steht, in die Webseite eingebunden werden. Außerdem wird das Google Icon Font verwendet. Also sieht die Einbindung so aus:



### Allgemeine Konfiguration

Standardmäßig halt sich materializecss in einem weiß, rosanen Stil, das passt in unserem Fall nicht zu dem vorgegebenen Stil, nämlich schwarz und pink. Dies lässt sich ändern durch erwerben eines kostenpflichtigen Theme, welches aber nicht nötig ist, wenn man es selbst ändert. Alle Standard HTML Komponenten lassen sich leicht verändern durch setzen einer Hintergrundfarbe und einer Schriftfarbe. Doch Input fields wie text oder password, verhalten sich anders und lassen sich nur durch folgende CSS Regelns korrekt ändern:



Dies bewirkt das input fields die so ausgesehen haben:



Nun so aussehen:



### Zusätzliche Komponenten

Zusätzlich galt es eine Karte auf der Kontakt Seite einzubinden. Wenn es heißt Karte, denkt jeder sofort an Google Maps. Ohne Google API Key, lässt sich so etwas nicht umsetzen. Einen API Key zu erhalten ist leicht, aber wenn man sich auf der offiziellen Google Maps Plattform umsieht, erkennt man schnell das man zwar Google Maps mit einem Google API Key gratis verwenden kann, denn es werden pro Monat 200€ in Verwendungskosten gutgeschrieben. Aber sobald man Google Maps kommerziell verwendet, gilt das alles nicht und man muss um einiges mehr zahlen. Also ist Google Maps kein Lösungsansatz.

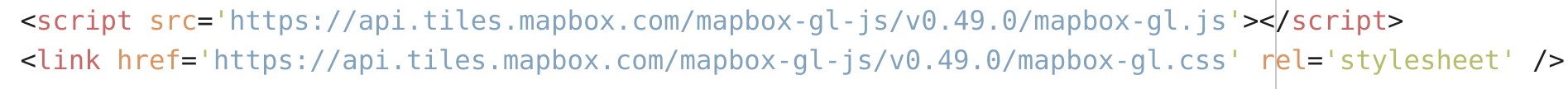
Eine schnelle Google Suche ergibt folgende free-to-use Alternativen:

- Openstreetmap  
- Waze  
- Mapbox

Alle genannten Technologien fallen innerhalb unserer Kriterien, aber Mapbox hat bei weitem die umfangreichste Konfiguration, und die bekanntesten Kunden, wie zum Beispiel CNN, IBM GitHub und MasterCard.

#### Einbindung

Mapbox verwendet die Hauseigene Mapbox GL Javascript Bibliothek, welche WebGL verwendet um deren Karte zu rendern. Diese Bibliothek kann ganz einfach eingebunden werden durch:

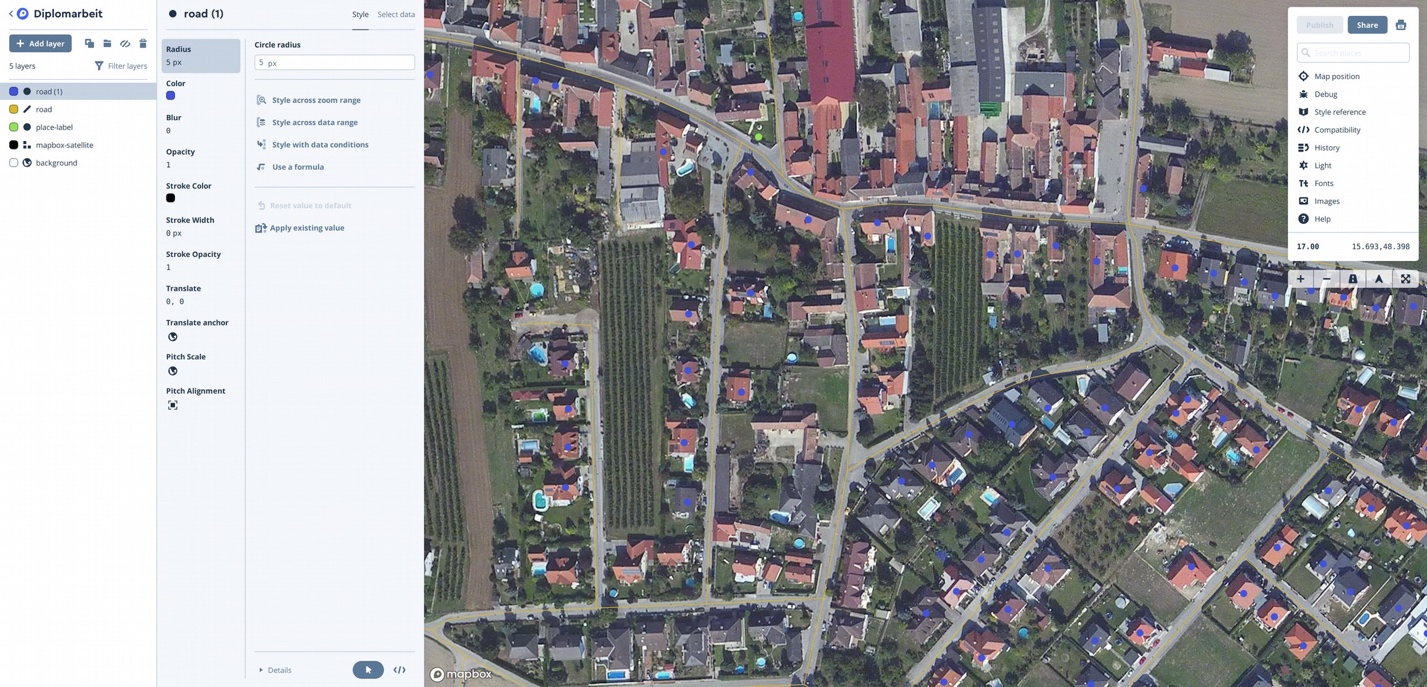


Außerdem muss eine zusätzliche Komponenten Bibliothek namens geocoder eingebunden werden. Diese ermöglicht es innerhalb der Karte ein Suchfeld einzubauen, um spezielle Orte suchen zu können.



#### Konfiguration

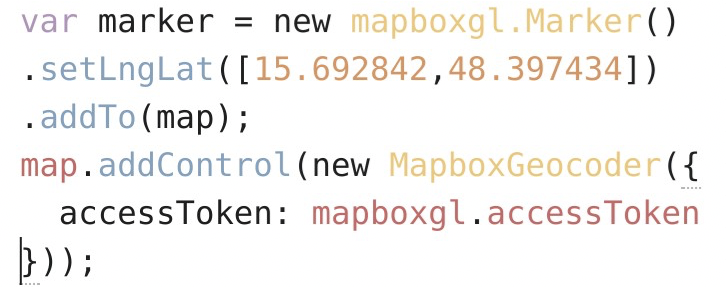
Mapbox ermöglicht es einen auf deren Webseite eine Karte zu erstellen, mit allen Komponenten, Farbschmenen, und Markern, so wie man will. In unserem Fall, war eine einfache Karte, mit geographischer Ansicht, so wie einer Satelliten Ansicht erwünscht. Auf Mapbox Studio Seite lassen sich diese Komponenten aktivieren, und deren Deckkraft einstellen. Und nach Festlegung der Map Position sieht diese so aus:



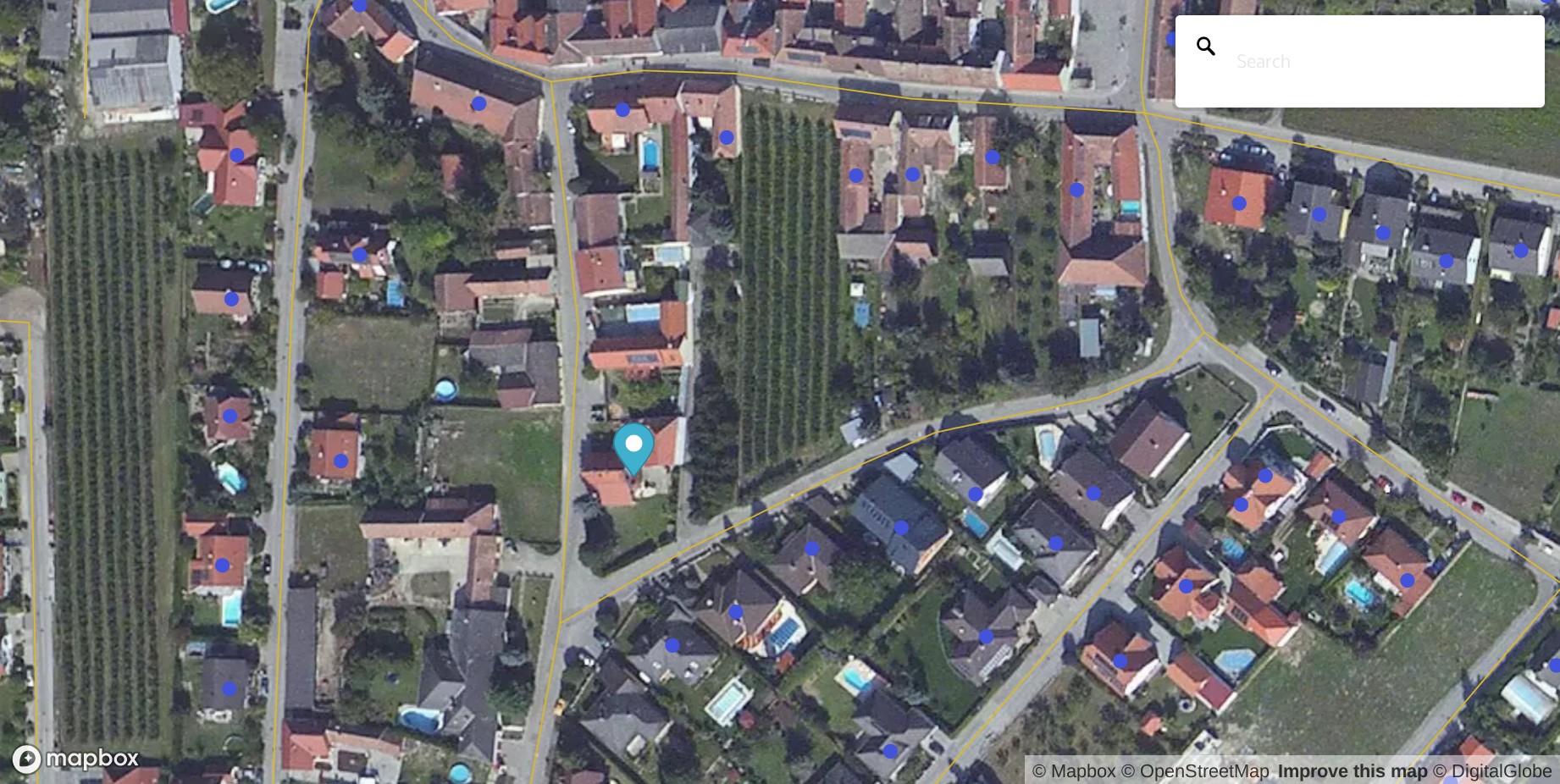
Um diese nun zu einer Webseite hinzufügen zu können stellt Mapbox den vorgefertigten Code bereit.



Jede Implementation, erhält einen zufällig generierten Zugriffscode, mit dem eindeutig identifiziert werden kann, wer diese Karte verwendet. Außerdem war ein Marker, der die Position eindeutig festlegt, sowie eine Suchleiste (die mittels geocoder umgesetzt wird) gewünscht. Also musste folgendes hinzugefügt werden:



All das führt dazu, dass die Karte dann so aussieht:



## Controller

In den folgenden Unterkapiteln wird die Umsetzung des Controllers erklärt. Der Controller ist, kurz gesagt, zuständig für die Steuerung der Anwendung und der Eingaben.

### Registrierung

Der Registrierungs-Controller verwaltet die Registrierung eines Benutzers an der Webseite.

Dabei werden folgende Daten auf ihre Richtigkeit überprüft:

* Benutzername
* E-Mail
* Passwort
* Passwort, noch einmal zur Bestätigung

#### Verschlüsselung des Passworts

Bei einer Webanwendung mit Benutzerverwaltung, muss besonders auf die Sicherheit der Speicherung des Passworts gelegt werden.

PHP stellt daher die Methode „password\_hash“ zur Verfügung, um dies zu realisieren.



Passwort-Verschlüsselung in PHP

Der zweite Parameter dieser Methode bestimmt die Art der Verschlüsselung, es wurde sich für „PASSWORD\_DEFAULT“ entschieden, welche den bcrypt-Algorithmus bzw. Blowfish-Algorithmus implementiert, dieser wird nun kurz erklärt.

##### Blowfish-Algorithmus

Der Blowfish-Algorithmus benutzt einen aufwändigen Schlüsselinitialisierungsalgorithmus. Dabei sieht ein Passwort, welches danach in der Datenbank gespeichert wird folgendermaßen aus.



Beispiel eines gehashten Passworts, das mittels des Blowfish-Algorithmus erstellt wurde

Der Term „2a“ zwischen den Dollar-Zeichen steht für den Blowfish-Algorithmus, der „08“-Term danach steht für den Exponenten zur Potenz 2, damit ergeben sich 2 hoch 8, also 256 Runden. Die Runden bedeuten, wie oft ein Passwort gehasht wurde. Der Rest des gehashten Passworts steht für den 16-stelligen „Salt“, welcher eine beliebig erstellte Zeichenkette darstellt, danach dann das verschlüsselte Passwort.

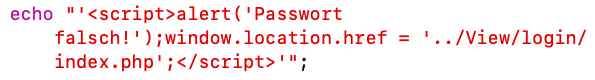
Sind nun alle eingegebenen Daten korrekt, so werden diese an das Model weitergegeben und in die Datenbank gespeichert. Weiters wird eine E-Mail an die E-Mail-Adresse des Benutzers geschickt, diese dient zur Bestätigung der Registrierung.

### Login

Der Login-Controller ist zuständig für die Verwaltung des Logins auf der Webseite, das heißt die Funktionalität, dass sich ein Benutzer an der Webseite anmelden kann.

Überprüft wird im Login auf den Status des Benutzer-Accounts, das heißt, ob dieser ein einfacher „Benutzer“ oder ein „Administrator“ ist, da sich die Profilseite und die Funktionen der beiden unterscheidet.

Sollte dem Benutzer bei der Eingabe ein Fehler unterlaufen, so wird dieser darüber mittels einer Meldung der Webseite benachrichtigt. Für diese Fehlerbehandlung wird im Controller ein JavaScript-Code ausgeführt.



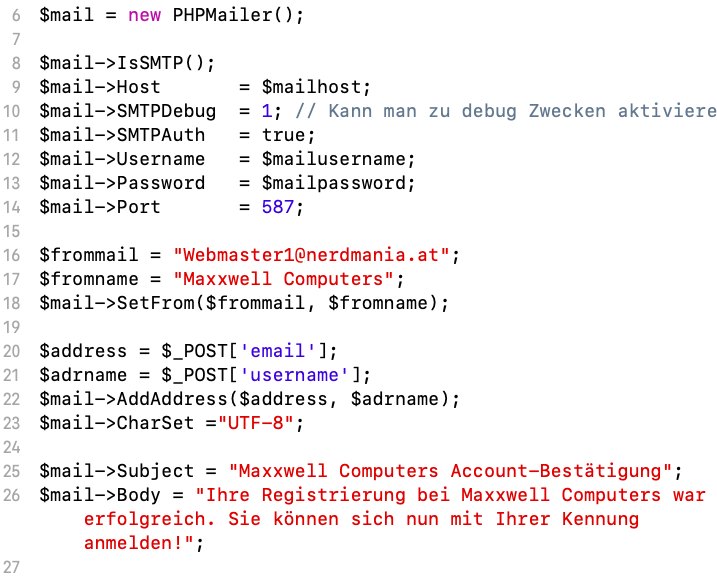
Beispiel einer Fehlerbehandlung

### Mail-Sender

Durch die Implementierung der Funktion des Benachrichtigens eines Benutzers über seine Registrierung, sowie der Funktionalität des Zurücksetzens der E-Mail-Adresse und des Passworts, muss eine automatische E-Mail versendet werden, welche nun in diesem Unterkapitel beschrieben wird.

Man entschied sich zu der Implementierung des PHPMailers, welcher genau für diese Problem zum Einsatz kommt. Der PHPMailer-Ordner wird einfach in den Projektordner abgelegt und bei der Verwendung angegeben werden.

Weiters ist für eine jede zu versendende E-Mail eine Vorlage erstellt werden, bzw. eine Konfiguration vorgenommen werden.



Beispiel einer Konfiguration für den PHPMailer

### Kennung vergessen

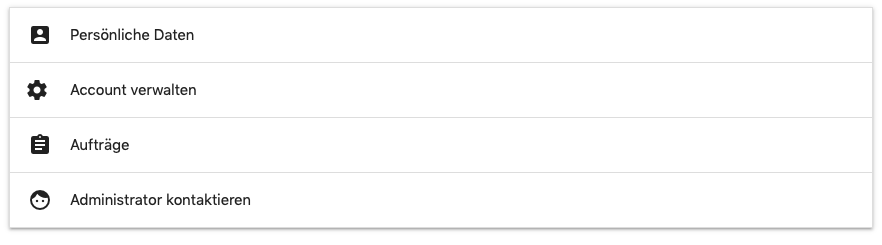
Hier kommt wieder der PHPMailer zum Einsatz, sollte das Passwort oder der Benutzername vergessen werden, so wird eine E-Mail an die E-Mail-Adresse, die in dem zum Zurücksetzen angeforderten Konto angegeben wurde. Diese E-Mail enthält ein, vom Controller, neu generiertes Passwort, welches aus Sicherheitsgründen danach geändert werden sollte und den Benutzernamen, welcher aus dem Model ausgelesen wird.

Der Controller erstellt ein neues Passwort, welches durch Zufall generiert wird und wieder mit dem Blowfish-Algorithmus verschlüsselt wird. Danach fordert der Controller das Model an, das neue, verschlüsselte Passwort in der Datenbank zu aktualisieren.

### Benutzerfunktionen

Die Benutzerfunktionen sind in folgende vier Menüs aufgeteilt.

* Persönliche Daten
* Account verwalten
* Aufträge
* Administrator kontaktieren



Benutzer-Menü

Die persönlichen Daten beinhalten eine Vielzahl an Möglichkeiten, seine persönlichen Details, wie Firma, Telefonnummer anzugeben. Sollte man diesen Daten ausgefüllt haben, klickt man auf den Button darunter, welcher den Controller auffordert die Daten an das Model weiterzugeben für die Speicherung in der Datenbank. Wird das Menü nochmals aufgeklappt, so sieht man die bereits eingegeben Daten und kann diese ändern.

Das Menü zur Accountverwaltung, beinhaltet die Möglichkeit, seine E-Mail-Adresse bzw. sein Passwort zu ändern. Führt man dies durch, so wird man im Anschluss daran zurück zum Login geworfen, wo man sich mit der neuen Kennung anmeldet, die Daten werden währenddessen vom Controller zur Sicherung an das Model weitergegeben.

Die Auftragsverwaltung wird in einem eigenen Kapitel erklärt.

Unter dem Menü „Administrator kontaktieren“ hat der Benutzer die Möglichkeit den Administrator per E-Mail zu kontaktieren. Per Klick auf den Button in diesem Menü, öffnet der Controller das Standard-E-Mail-Programm (je nach Betriebssystem) auf dem Rechner des Benutzers.

### Administratorfunktionen

Die Administratorfunktionen sind in folgende drei Menüs aufgeteilt.

* Showcases bearbeiten
* Laufende Aufträge
* Offene Aufträge



Administrator-Menü

Das Menü „Showcases bearbeiten“ ist dazu da, aus allen laufenden Aufträgen, maximal vier auszuwählen und auf der Hauptseite anzuzeigen. Der Controller zeigt dem Administrator ein Drop-Down-Menü mit allen möglichen Aufträgen zur Auswahl, diese Daten stammen von der Tabelle der Aufträge aus der Datenbank. Außerdem kann der Administrator auswählen mit, oder ohne Kontaktdaten diese anzuzeigen. Nach dieser Auswahl schickt der Controller diese Daten an das Model, welche eine Referenz des Auftrags und die Auswahl über Kontaktdaten in der Tabelle „Showcases“.

Die Menüs zu den Aufträgen werden im Kapitel genauer beschrieben.

### Aufträge

### Hauptseite

### Kontaktseite

# Testfälle

# Zusammenfassung und Ausblick

Mit dieser Arbeit wurde eine Webanwendung entwickelt, dessen Hauptaufgabe darin bestand, eine vereinfachte Administratoroberfläche zu entwickeln, die es dem Administrator ermöglicht, Aufträge von Benutzern zu verwalten und zu präsentieren. Mit dieser Webanwendung soll eine höhere Anzahl an Benutzern und damit Kunden erzielt werden.

## Fazit

Zusammenfassend kann man sagen, dass die Implementierung, bis auf ein paar wenige Abweichungen, nach der Planung vollzogen wurde. Jedoch wurde das geplante Ausmaß von 360 geplanten Arbeitsstunden (bezogen auf beide Entwickler) schlecht aufgeteilt wurde, da das Frontend weit schneller entwickelt wurde, als das Backend.

Dies ist zurückzuführen auf mangelnde Erfahrung mit der Projektplanung. Zum Beispiel mit dem Einsatz eines PHP-Frameworks, hätte bei der Entwicklung des Backend noch Zeit eingespart werden können, man entschied sich gegen den Einsatz eines solchen, aufgrund der im Kapitel angeführten Punkte. Dennoch konnte eine erfolgreiche Umsetzung, ohne schwerwiegende Probleme oder Meinungsverschiedenheiten, erfolgen.

## Ausblick

Die Entwicklung Web-Anwendung endet selten mit der ersten Version, neue Funktionen, Designs, Updates auf Frameworks, sind nur einige wenige Beispiele für Verbesserungen. Folgende Punkte könnten in der Zukunft noch hinzugefügt werden:

* Spracheneinstellungen: mehrere Sprachen stehen neben der Standardsprache Deutsch zur Auswahl, wie Englisch
* Auftragsverwaltung: Benutzer können mehrere Aufträge aufgeben und haben mehr Einstellungsmöglichkeiten

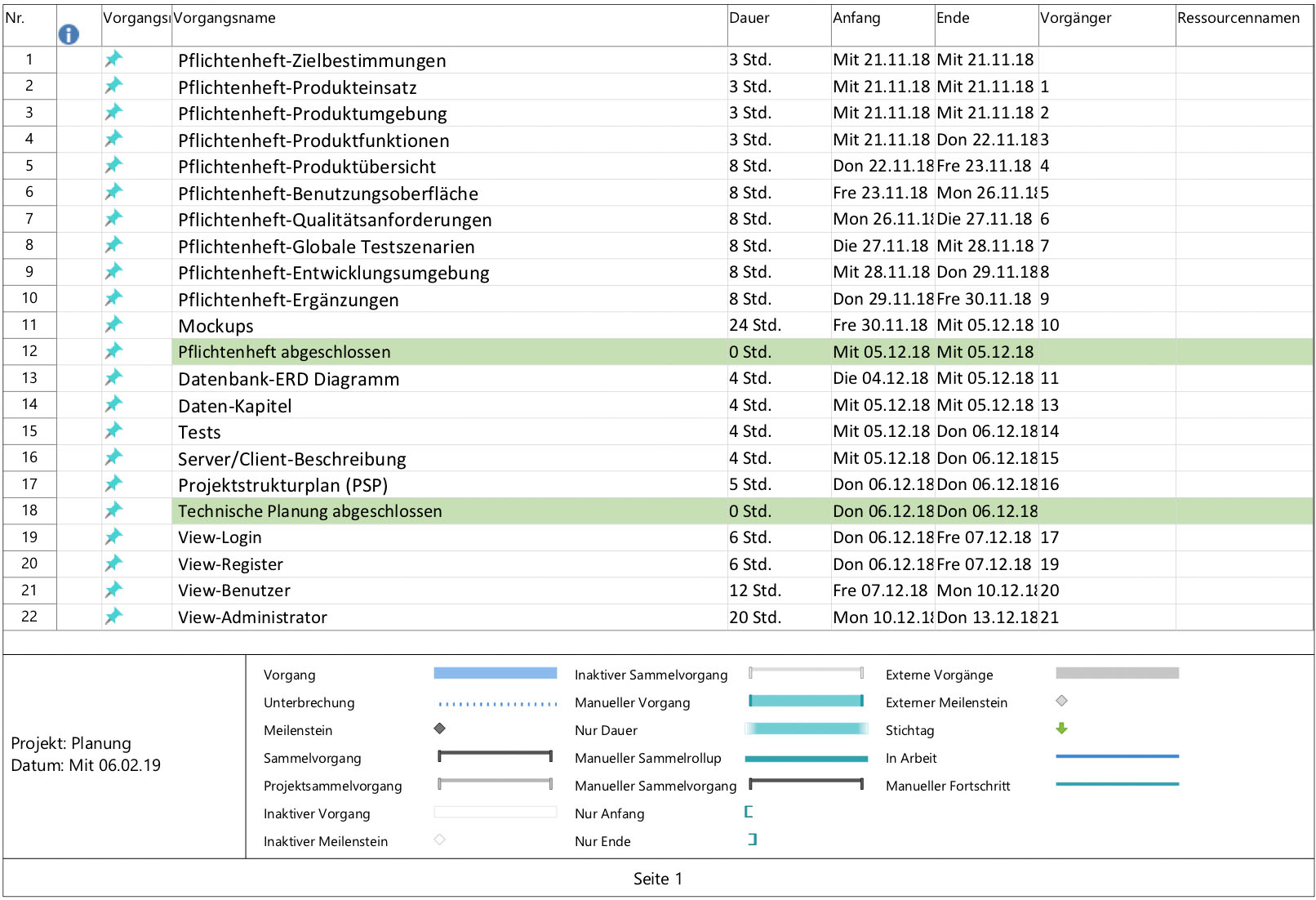
# Dokumentation des Projektablaufs

In diesem Kapitel wird die Planung dieser Arbeit behandelt.

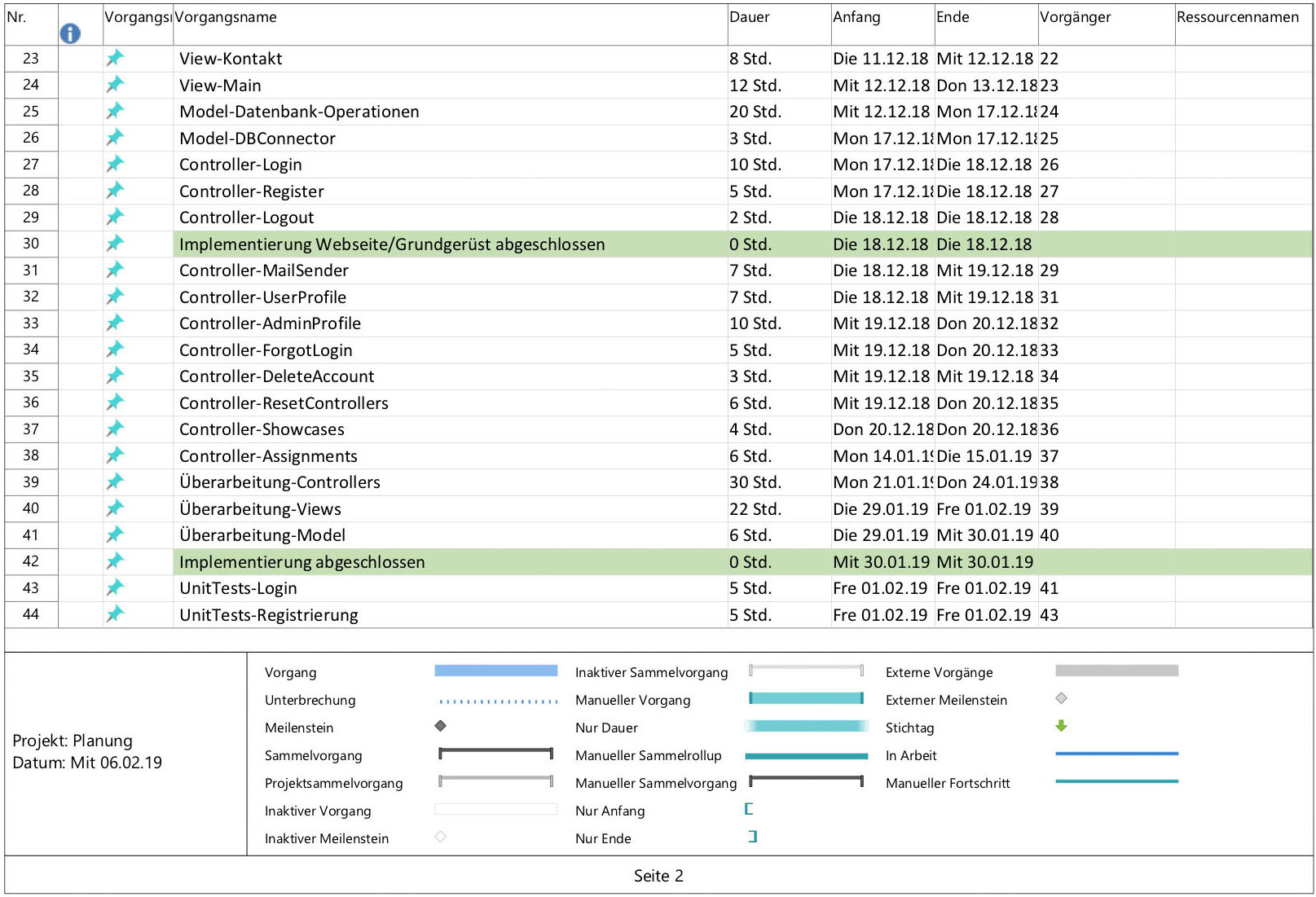
## Terminplanung

Die gesamte Planung wurde mit dem Programm Microsoft-Project erstellt.

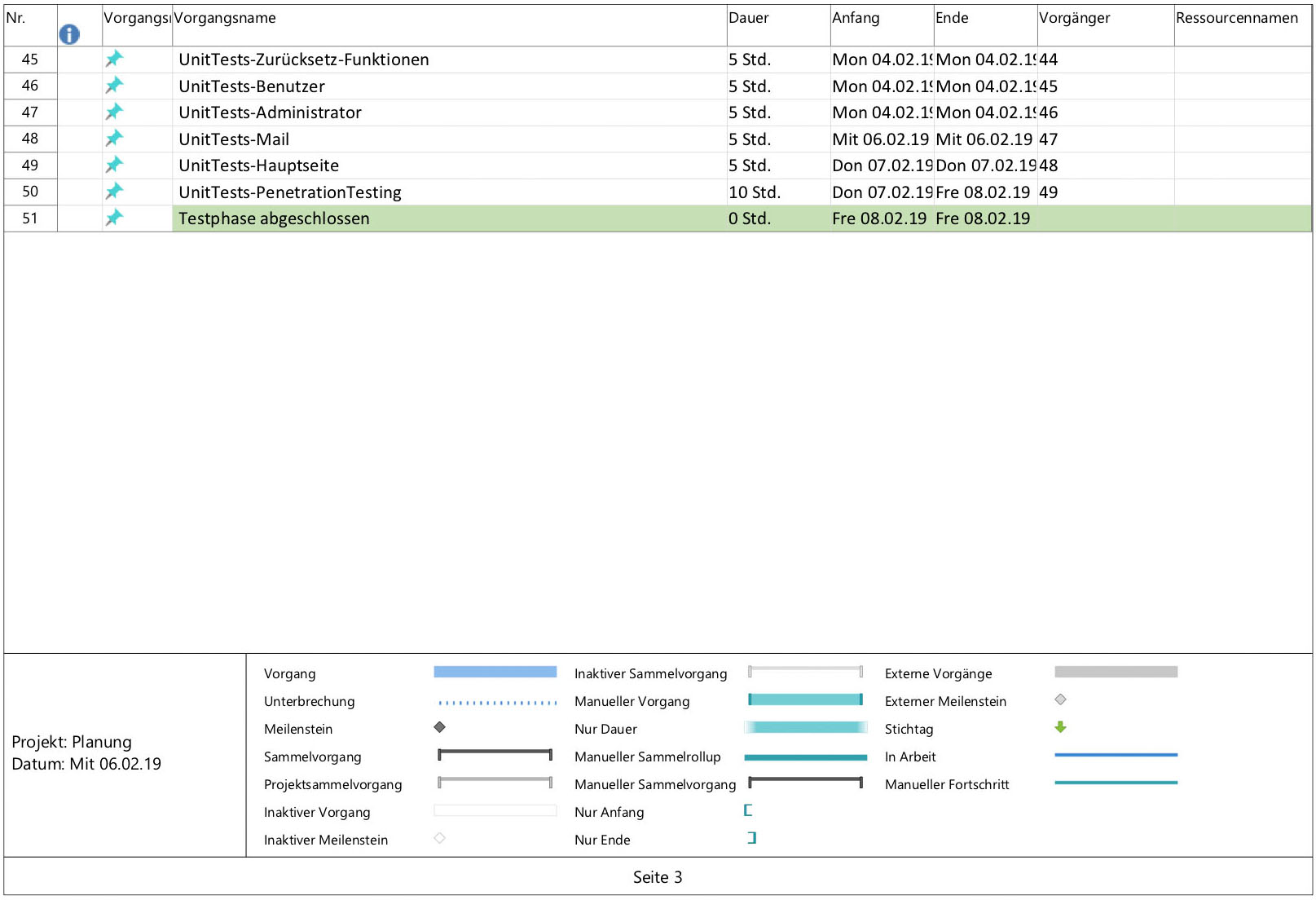
Die Arbeitspakete wurden als Tabelle dargestellt und zeitlich eingeordnet, dabei summieren sich die geplanten Stunden auf 360 Arbeitsstunden. Das GANTT-Diagramm beschreibt dann den zeitlichen Ablauf aller Arbeitspakete.



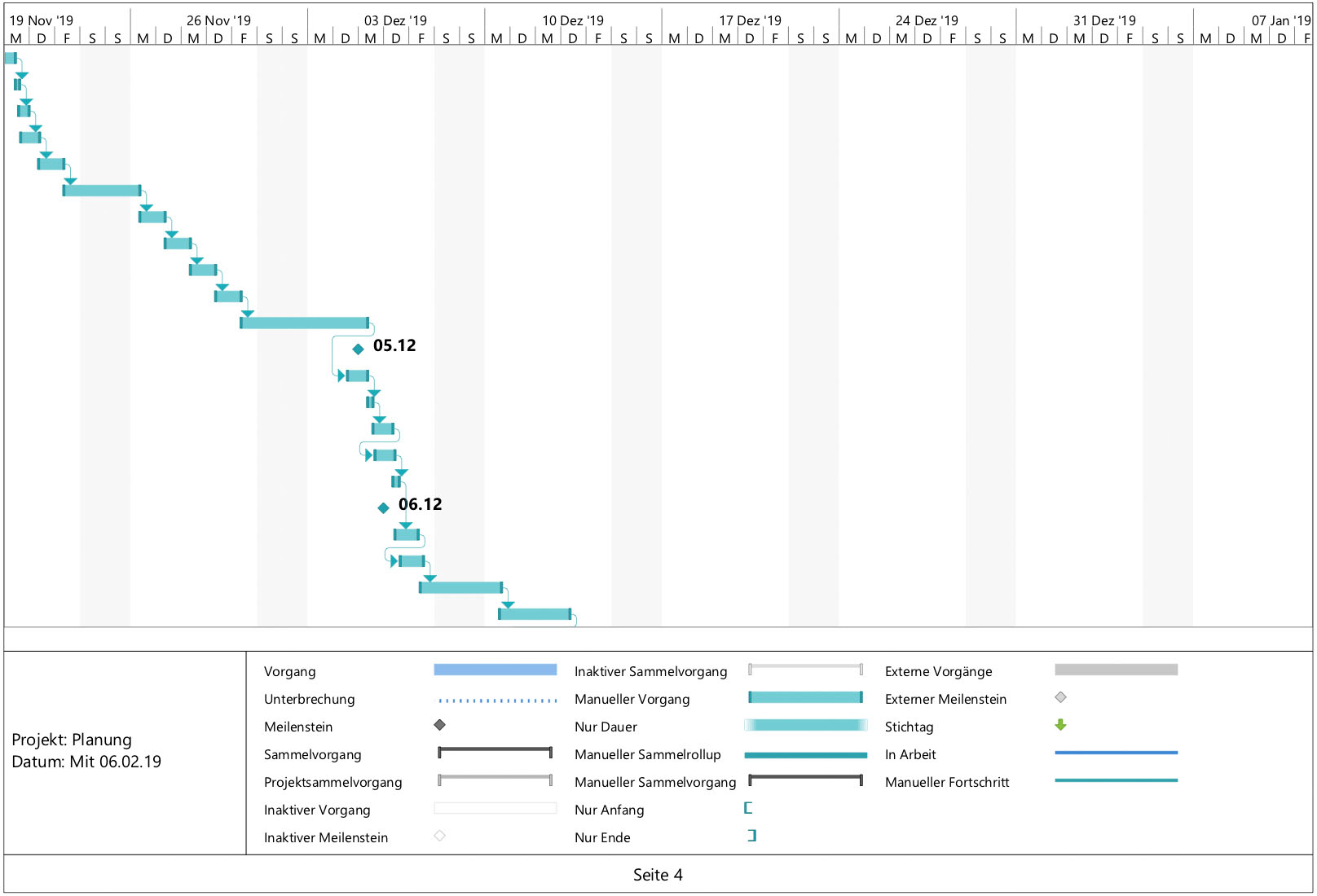
Arbeitspakete Teil1



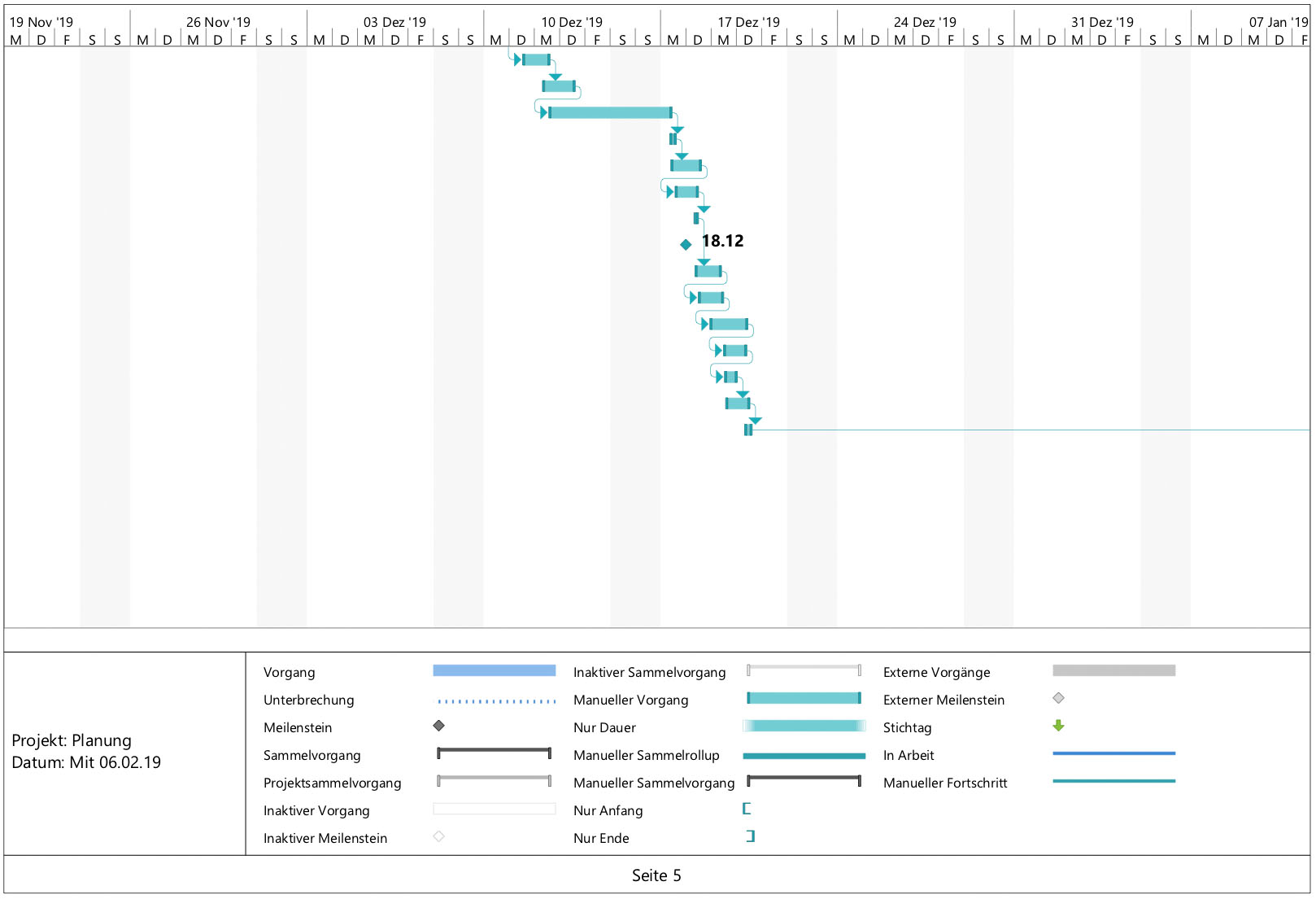
Arbeitspakete Teil2



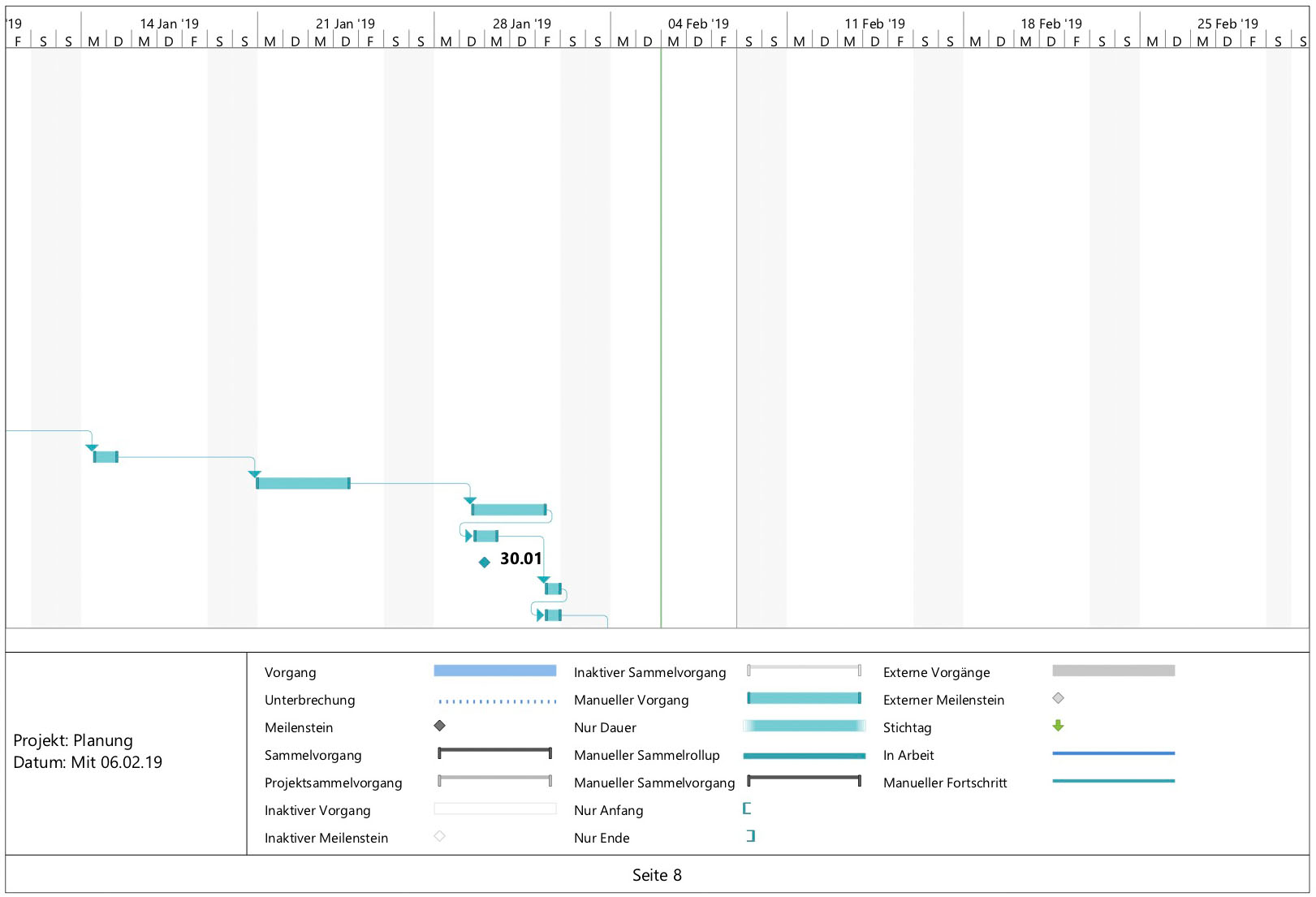
Arbeitspakete Teil3



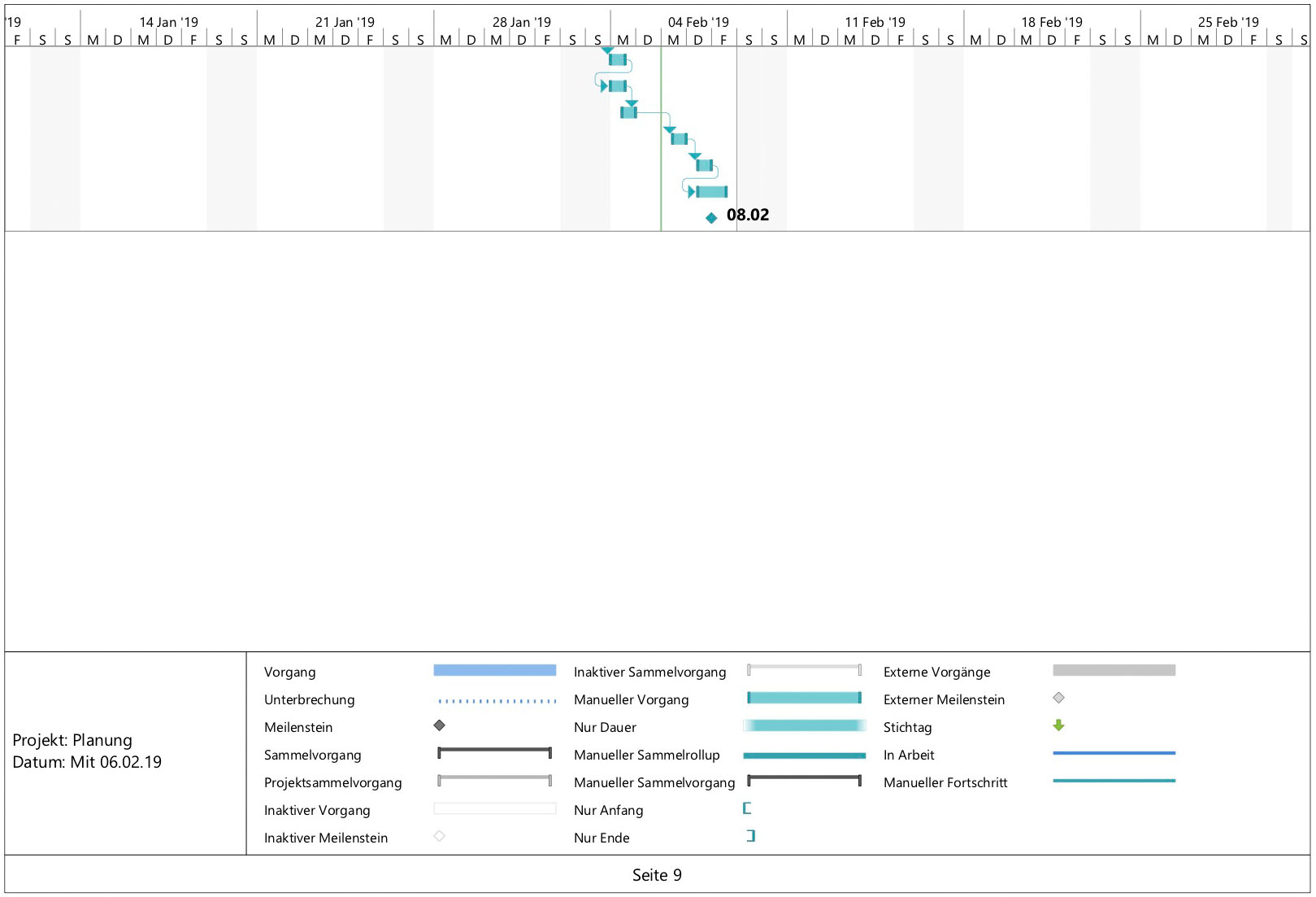
GANTT-Diagramm Teil 1



GANTT-Diagramm Teil 2



GANTT-Diagramm Teil 3



GANTT-Diagramm Teil 4

## Projekttagebuch

# Quellenverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Anhang