

# Hausaufgabe 1 - Abgabe 1

## Planungsdokument

**Teamname:** 8GILE

**Gruppe:** 8

### **Vorwort:**

Die Grundstruktur des folgenden Projektplans wurde als gemeinsame Gruppenleistung in drei Meetings erstellt. Zunächst haben wir im Team das Konzept besprochen und Anforderungen formuliert, um einen gemeinsamen Überblick über das Projekt zu erhalten. Daraus wurden Epics und User Stories erstellt und ein Domainmodell abgeleitet. Die Daten für die Aufgaben wurden zwischen den Meetings erarbeitet und gemeinsam zusammengetragen. Für die endgültige Ausformulierung wurden folgende Aufgaben an die Teammitglieder verteilt:

### **Alexander Bertram:**

Erstellung & Textproduktion des Zeitplans, Formatvorschlag des Zeitplans, Strukturierung der Aufgaben

### **Behrad Rajabiefumani:**

Risikobewertung, Qualitätssicherung, Vorschlag des Kanban Boards

### **Jonas Klepke:**

Risikobewertung & Textproduktion, Use case diagram, Protokolle der Meetings

### **Patrick Lindner:**

Prozessplan & Textproduktion, UI Prototyp & Erstellung, Vorschlag der Frameworks und Architektur, Anforderungen

### **Sebastian Zaun:**

Zeitplan und Erstellung einer Excel-Datei, Textproduktion der Anforderungsanalyse, Formatvorschlag des Zeitplans

### **Wei Dai:**

Prozessplan, Terminplanung, Koordination der Meetings, Erstellung der Abgabedatei

# Anforderungen

## Use cases

- Der Bewerber loggt sich ein und kann sich für den Studiengang Informatik als Bewerber eintragen.
- Der Bewerber loggt sich ein und sieht den Status seiner Bewerbung.
- Der Bewerber loggt sich ein und kann seine Bewerbung zurückziehen.
  
- Ein Mitglied des Zulassungsausschusses (MZA) loggt sich ein und sieht eine Liste aller Bewerber zum Studiengang Informatik.
- MZA loggt sich ein, klickt auf einen Listeneintrag und kann alle Details einer Bewerbung sehen und hier die Bewerbung annehmen, ablehnen oder für eine engere Auswahl markieren. Dies tritt erst offiziell in Kraft, wenn der ganze Bewerbungsprozess abgeschlossen ist. Der Bewerber erhält erst eine entsprechende Meldung, wenn der gesamte Bewerbungsprozess für den Studiengang abgeschlossen ist.

## Datenhaltung

Zu einem Bewerber werden die folgenden personenbezogenen Daten erhoben und verarbeitet:

- Vorname und Nachname\*\*
- Geschlecht\*\*
- Adresse\*\*
- E-Mail\*\*
- Telefonnummer\*
- EU-Student
- Wartesemester
- Abiturzeugnis und Durchschnittsnote
- Krankenversicherungsbescheinigung\*\*
- Weitere Zeugnisse\*
- Weitere Dokumente (weitere Belege, Empfehlungsschreiben)\*

\* Optional

\*\* Nur für die interne Verarbeitung und die Generierung von Anschreiben vorgesehen, kann nicht durch MZA eingesehen werden, um eine Beeinflussung zu vermeiden.

Zu einem Studiengang werden folgende Daten verfügbar sein:

- Name des Studiengangs
- Regelstudienzeit in Semestern
- Startdatum
- ECTS
- Anzahl Studienplätze
- Bereits belegte Studienplätze
- Numerus Clausus

Zu einer Bewerbung werden die Folgenden Daten verfügbar sein:

- Bewerbungsdatum
- Zuordnung zum Bewerber
- Zuordnung zum Studiengang
- Status (sichtbar für den Benutzer, ändert sich erst, wenn der gesamte Bewerbungsprozess abgeschlossen ist)
- Status intern

### **Optionale Funktionen**

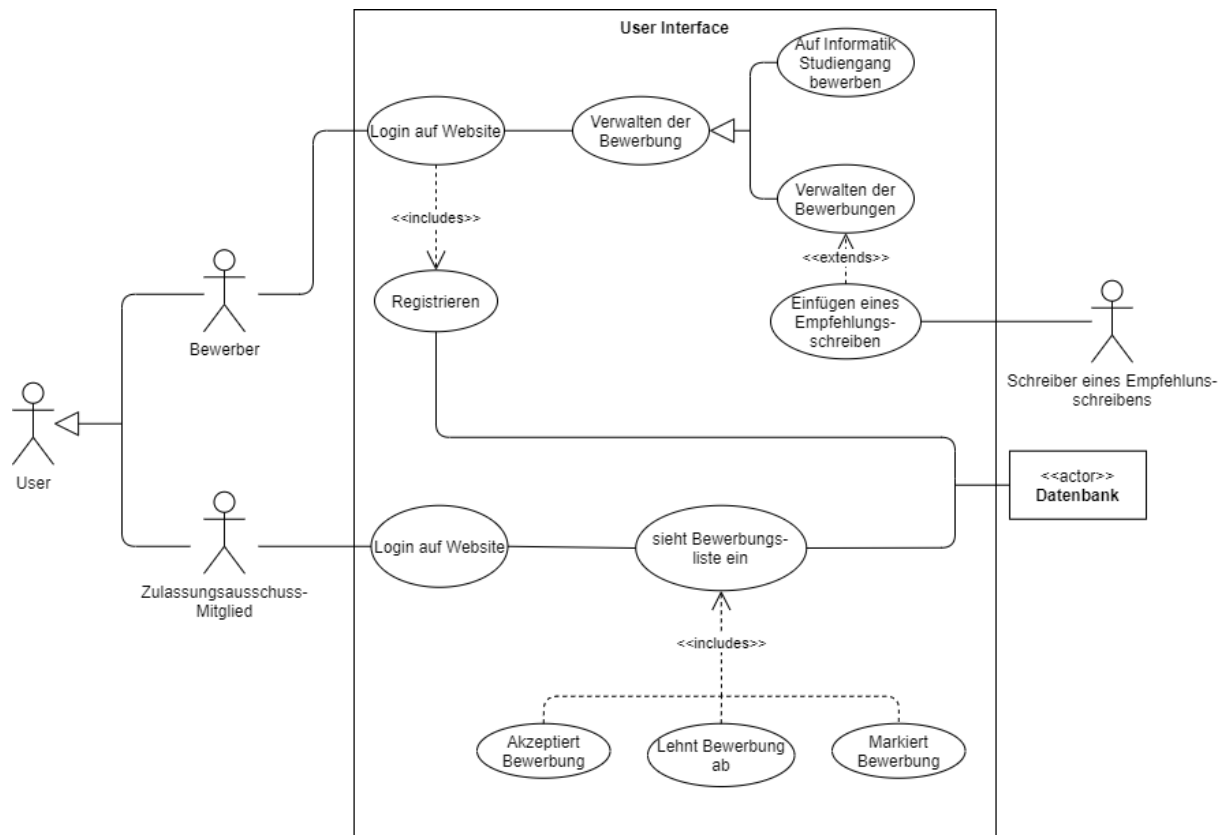
Im folgenden Punkt werden Funktionen beschrieben die Aufgrund mangelnder Zeit nicht mehr im Prototypen aufgenommen werden können, aber für das fertige Produkt eine entscheidende Rolle spielen:

- Der Bewerber registriert sich in der Applikation durch das Angeben aller Daten, die unter Datenhaltung zum Bewerber angegeben sind, inklusive eines Passwortes für seinen Bewerberaccount.
- Der Bewerber loggt sich ein und fügt einen Schreiber für ein Empfehlungsschreiben seinem Bewerberaccount hinzu.

Das MZA loggt sich ein und beendet den Bewerbungsverfahren für den Studiengang Informatik. Damit werden Bestätigungs- und Ablehnungs-E-Mails versendet und der Status wird in der Applikation für den Bewerber sichtbar.

Ein Schreiber für das Empfehlungsschreiben erhält eine E-Mail des Systems, welche besagt, dass er von einem Bewerber eingeladen wurde ein Empfehlungsschreiben hochzuladen inklusive eines persönlichen Links, auf dem er Empfehlungs-Dokumente hochladen kann.

## Visualisierung der Anforderungen



## **UI-Prototype**

Im folgenden Abschnitt wird der UI-Prototyp vorgestellt. Es handelt sich hierbei lediglich um eine grobe Orientierungshilfe die einen Überblick über die UI verschaffen soll und nicht um einen festen Design- oder Styleguide. Das fertige Endprodukt kann von diesem Prototyp abweichen.

## Bewerbung

...

zurücktreten

Studiengang: Informatik

Bewerber: 543

Belegte Plätze: 123/200

Status: Ausstehend

Beworben am: 29.02.2021

Frist: 30.05.2021

[Informationen zum Studiengang](#)

## Bewerbung

...

annehmen

zurücktreten

Studiengang: Informatik

Bewerber: 543

Belegte Plätze: 123/200

Status: Akzeptiert

Beworben am: 29.02.2021

Frist: 30.05.2021

[Informationen zum Studiengang](#)




[← Zurück](#)



## BSc Informatik

Begin	01.10.2021	Regelstudienzeit	6 Semester	ECTS	180
Studienplätze	200	Bewerber	123/200		
Numerus Clausus	/	Abschluss	BSc Informatik		

[Allgemeine Infos zum Studiengang](#)  
[Modulhandbuch](#)


Verbindlich bewerben 

Bewerbungen Informatik   456  
Plätze Belegt/Vorgemerkt/Frei   100/350/200

- ...
- Bewerbungsprozess abschließen
  - Markierte annehmen
  - Markierte ablehnen

	Studiengang	Beworben am	Nationalität	Empfehlungsschreiben	Praktika	
✓	Informatik	18.02.2020	deutsch	2	/	...
✗	Informatik	18.02.2020	deutsch	/	1	...
?	Informatik	18.02.2020	deutsch	1	/	...
✓	Informatik	18.02.2020	chinesisch	1	3	...
	Informatik	18.02.2020	deutsch	/	2	...



 Zurück

Bewerber: 13bd0fc9-1a4d-4ace-881e-7ac69b31f2c8

...

Studiengang

Informatik

Nationalität

deutsch

Wartesemester

2

Beworben am

23.04.2020

Abschlüsse

Schulausbildung	Name der Schule	Abitur	2.5	pdf
Berufsausbildung	Name des Berufscolleg	Handelskaufmann	2.0	pdf
Studium	Universität zu Köln	B.Sc Gesundheitsökonomie	4.0	pdf

Dokumente

Empfehlungsschreiben	Apple Inc.	pdf
Empfehlungsschreiben	Microsoft Corperation	pdf
Praktikum	Apple Inc.	pdf
Praktikum	Microsoft Corperation	pdf
Lebenslauf		pdf
Zertifikat Auslandsaufenthalt		pdf
FSJ		pdf

# BIPS - Zeitplanung

8GILE

Supervisor: Patrick

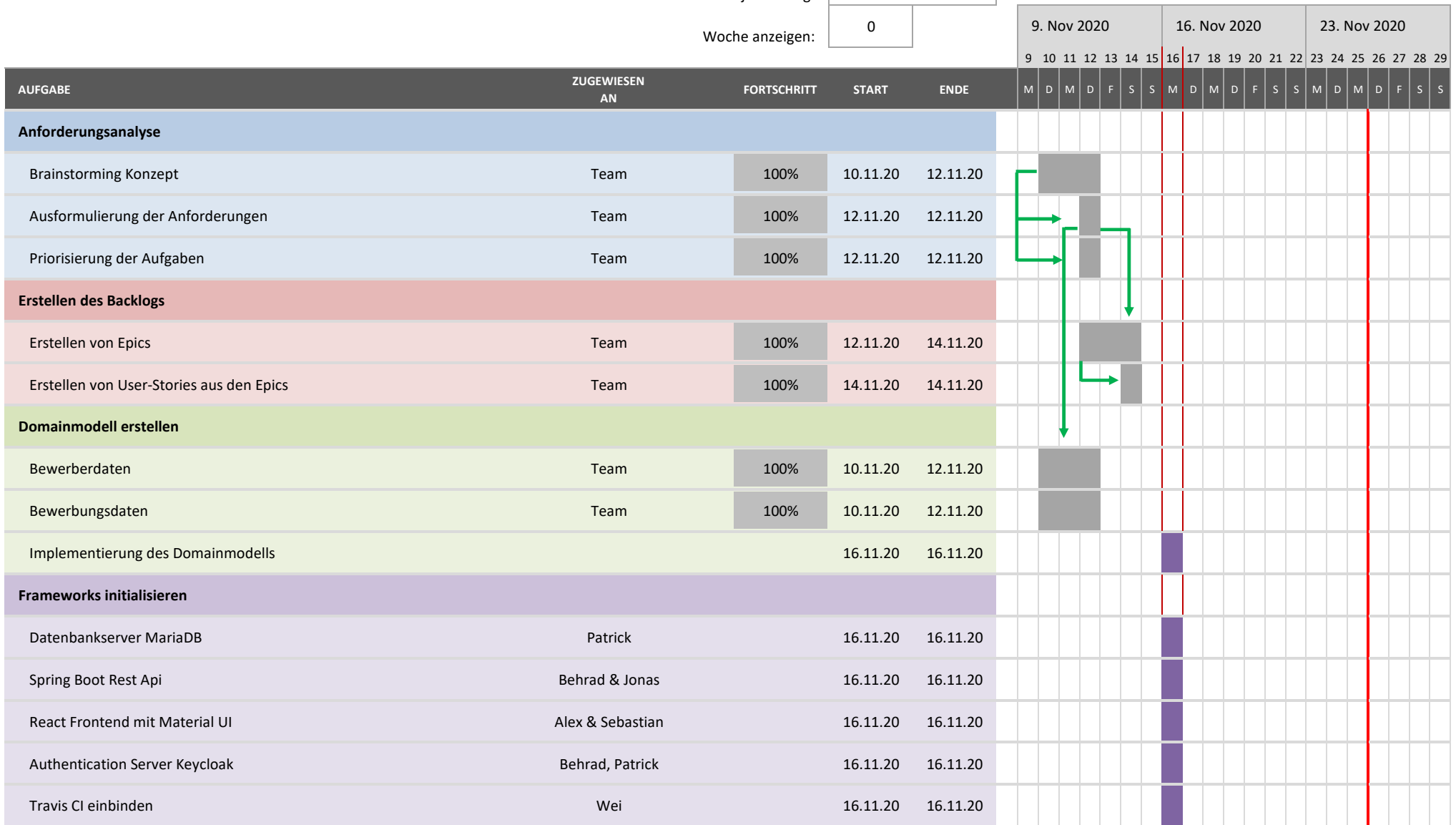
EINFACHES GANTT-DIAGRAMM von Vertex42.com  
<https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html>

Projektanfang:

Mo, 11.16.2020

Woche anzeigen:

0



Woche anzeigen:

0

						9. Nov 2020							16. Nov 2020							23. Nov 2020						
						9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
						M	D	M	D	F	S	S	M	D	M	D	F	S	S	M	D	M	D	F	S	S
AUFGABE	ZUGEWIESEN AN	FORTSCHRITT	START	ENDE																						
<b>Anlegen einer Bewerbung</b>																										
Backend: Bewerbung anlegen	Behrad & Jonas		17.11.20	19.11.20																						
Frontend: Eingabemaske Bewerber erstellen	Alex & Sebastian		17.11.20	19.11.20																						
Backend und Frontend verbinden	Patrick & Wei		20.11.20	21.11.20																						
Doku	Team		17.11.20	21.11.20																						
<b>Bewerbungsanzeige Zulassungsausschuss</b>																										
Backend: Bewerbungsliste anzeigen/sortieren	Behrad & Jonas		17.11.20	19.11.20																						
Backend: Bewerbung in Liste annehmen/ablehnen/vormerken	Behrad & Jonas		17.11.20	19.11.20																						
Frontend: Eingabemaske erstellen	Alex & Sebastian		17.11.20	19.11.20																						
Backend mit Frontend verbinden	Patrick & Wei		20.11.20	21.11.20																						
Doku	Team		17.11.20	21.11.20																						
<b>Bewerber: Anzeige der Bewerbungen</b>																										
Backend: Eigene Bewerbung ausgeben	Behrad & Jonas		17.11.20	19.11.20																						
Backend: Angebot annehmen/ablehnen	Behrad & Jonas		17.11.20	19.11.20																						
Backend: Bewerbung zurückziehen	Behrad & Jonas		17.11.20	19.11.20																						
Frontend: Eingabemaske erstellen	Alex & Sebastian		17.11.20	19.11.20																						
Backend und Frontend verbinden	Patrick & Wei		20.11.20	21.11.20																						
Doku	Team		17.11.20	21.11.20																						

Woche anzeigen:

0

Woche anzeigen: <div>0</div>						9. Nov 2020							16. Nov 2020							23. Nov 2020										
						9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29				
AUFGABE	ZUGEWIESEN AN					FORTSCHRITT		START	ENDE	M	D	M	D	F	S	S	M	D	M	D	F	S	S	M	D	M	D	F	S	S
Deteilansicht Zulassungsausschuss																														
Backend: Bewerbungsdetails anzeigen	Behrad & Jonas							17.11.20	19.11.20																					
Frontend: Eingabemaske erstellen	Alex & Sebastian							17.11.20	19.11.20																					
Backend und Frontend verbinden	Patrick & Wei							20.11.20	21.11.20																					
Doku	Team							17.11.20	21.11.20																					
Replanning																														
Evaluation bisheriger Arbeit	Team							19.11.20	19.11.20																					
Replanning	Team							20.11.20	20.11.20																					
Testing und Bug-Fixing																														
Finale Tests Bewerbersicht	Team							22.11.20	24.11.20																					
Finale Tests Sicht Zulassungsausschuss	Team							22.11.20	24.11.20																					
Finales Fixing beider Sichten	Team							23.11.20	25.11.20																					
Abgabe Prototyp																														

Woche anzeigen:

0

						9. Nov 2020							16. Nov 2020							23. Nov 2020						
						9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
						M	D	M	D	F	S	S	M	D	M	D	F	S	S	M	D	M	D	F	S	S
AUFGABE	ZUGEWIESEN AN	FORTSCHRITT	START	ENDE																						
<b>Registrieren und Einloggen ermöglichen</b>																										
Registrieren und Einloggen ermöglichen			26.11.20	30.11.20																						
<b>Empfehlungsschreiben anfordern</b>																										
Empfehlungsschreiben anfordern			26.11.20	30.11.20																						
<b>Empfehlungsschreiben Bewerbern zuordnen</b>																										
Empfehlungsschreiben Bewerbern zuordnen			26.11.20	30.11.20																						
<b>Sonstige Anlagen hinzufügen (Arbeitszeugnisse,...)</b>																										
Sonstige Anlagen hinzufügen			26.11.20	30.11.20																						
Neue Zeilen ÜBER dieser einfügen																										

#### Legende



Abhängigkeit



Optionale Funktionserweiterung

# Begründung Zeitplanung

Für die Zeitplanung wurde eine Excel-Datei erstellt und mit den Meilensteinen und den dazugehörigen Aufgaben befüllt. Diese Datei wurde in einem Google Drive Ordner veröffentlicht, sodass jedes Teammitglied darauf zugreifen kann. Dort können der Fortschritt einzelner Aufgaben, sowie der gesamten Meilensteine nachvollzogen werden und gegebenenfalls kurzfristig Änderungen am Plan vorgenommen werden.

## Identifizierte Meilensteine:

- **Anforderungsanalyse:**  
Zunächst soll ein Konzept für das gesamte Projekt erarbeitet werden. Da gerade die Anforderungen des Kunden nicht genau formuliert wurden, müssen wir uns in der Gruppe darauf einigen, welche Anforderungen essentiell für ein lauffähiges Produkt sind. Aufgrund des engen Zeitplans haben wir uns darauf geeinigt, dass jedes Teammitglied sich in Eigenregie Gedanken über die Anforderungen macht, um diese in einem Meeting zu identifizieren und in den darauffolgenden Tagen mit der Arbeit am Projekt beginnen zu können.
- **Backlog erstellen:**  
Identifizierte Anforderungen werden in das Backlog übertragen. Die Aufteilung in kleinschrittige Aufgaben ist aufgrund einer ausführlichen Anforderungsanalyse leicht zu realisieren.
- **Domainmodell erstellen:**  
Ausgehend von der Anforderungsanalyse soll eine Datenstruktur für die Bewerber und Bewerbung konstruiert werden. Zuerst soll in einem Meeting geplant werden, welche Daten essentiell für die Struktur sind. Anschließend wird dieses Modell technisch implementiert, sodass es als Input für die Datenbank erfasst werden kann. Da es sich um ein einfaches Datenobjekt handelt, sollte die Erstellung nicht mehr als einen Tag in Anspruch nehmen.
- **Frameworks initialisieren:**  
Innerhalb eines Tages sollen die Frameworks initialisiert werden. Durch berufliche Erfahrungen eines Teammitgliedes in den meisten Umgebungen sollte dies einfach vonstatten gehen. Außerdem dabei werden die für die einzelnen Komponenten zuständigen Teammitglieder kurz in ihr Framework eingeführt.

Anstatt die Meilensteine in Komponenten wie Frontend, Backend und Datenbank aufzuteilen, hat sich das Team entschieden die folgenden Meilensteine so zu definieren, dass nach Abschluss direkt eine grundlegende Funktion der Software funktionsfähig ist. Damit rückt das Ziel bis zur Prototypabgabe die grundlegende Funktionalität der Software zu gewährleisten näher.

- **Anlegen einer Bewerbung:**  
Es sollen im Frontend und im Backend die Möglichkeit implementiert werden eine Bewerbung im System anzulegen. Dabei beschäftigen sich die jeweiligen "Experten" mit den Funktionen ihres Fachgebietes. Nachdem die Front- und Backendfunktionen separat entwickelt worden sind, sollen sie miteinander verbunden werden. Während der kompletten Entwicklungsphase sollen außerdem die Funktionen der Software für

ein Benutzerhandbuch dokumentiert werden. Da sich, in diesem und den nächsten beiden Meilensteinen, die jeweiligen Aufgaben stark ähneln, verteilt sich der jeweilige Arbeitsaufwand auf alle drei Meilensteine. Deswegen wird für diese Implementation eine ganze Woche im Zeitplan reserviert.

- **Bewerbungsanzeige Zulassungsausschuss:**  
Nach diesem Meilenstein soll es dem einem Mitglied des Zulassungsausschusses möglich sein alle Bewerbungen in einer Liste zu sehen und ggf. diese nach gewünschten Kriterien zu sortieren. Nach dem gleichen Muster wie im Punkt "Anlegen einer Bewerbung" werden Front- und Backend separat entwickelt, anschließend durch Integratoren zusammengeführt und während des gesamten Prozesses die neuen Funktionen dokumentiert. Die Zeiteinschätzung gleicht der im vorherigen Meilenstein.
- **Bewerber, Anzeigen einer Bewerbung:**  
Nach Abschluss dieses Meilensteines soll es dem Bewerber möglich sein seine angelegten Bewerbungen einzusehen und diese nach Zulassung ggf. anzunehmen oder abzulehnen. Die Integration der Komponente und der Zeitaufwand gleicht den vorherigen Meilensteinen.
- **Detailansicht Zulassungsausschuss:**  
Einem Mitglied des Zulassungsausschusses soll es danach möglich sein eine detaillierte Ansicht einer einzelnen Bewerbung anzeigen zu lassen und diese ggf. zu akzeptieren oder abzulehnen. Die Integration der Komponente und der Zeitaufwand gleichen den vorherigen Meilensteinen.
- **Replanning:**  
Es soll je nach Fortschritt in der Entwicklung entschieden werden, ob man bis zur Prototypabgabe noch versucht weitere Funktionen zu implementieren oder sich ggf. auf die essentiellen Funktionen konzentriert werden soll. Innerhalb von ungefähr zwei Tagen inmitten der Entwicklungsarbeit soll im Team geklärt werden wie in der Entwicklung weiter vorgegangen werden soll.
- **Testing und Bug-Fixing:**  
Nach Beendigung dieses Meilensteines soll ein funktionsfähiger Prototyp der Software vorhanden sein. Das ganze Team beschäftigt sich in erster Linie mit der Suche von Fehlern in der Software, welche danach gemeinsam behoben werden sollen. Für diese Aufgabe wird die gesamte Zeit nach der Implementierung der grundlegenden Funktionen in Anspruch genommen, um ein möglichst gutes Ergebnis bis zur Abgabe gewährleisten zu können.

Die weiteren im Zeitplan aufgeführten Meilensteine behandeln weitere noch nicht genau geplante Funktionen, die bei genügend Zeit noch zur Software hinzugefügt werden sollen. Ob dies getan wird soll nach dem Meilenstein Replanning geklärt sein.

# Risikobewertung

Wir haben uns dazu entschieden die Risiken, die bei einem solchen Projekt entstehen, zu kategorisieren. Zu diesen Kategorien gehören unserer Meinung nach zum einen die technischen Herausforderungen und zum anderen die organisatorischen Aspekte, die den Ablauf behindern könnten. Für jedes Risiko werden im Folgenden Lösungsansätze beschrieben, deren Umsetzung eine gute Strategie zur Reduzierung der Risiken sein kann.

Zu den identifizierten technischen Hauptrisiken gehören mangelnde Kenntnisse über die verwendeten Frameworks, Programmierpraktiken und deren Umsetzung. Gute Programmierkenntnisse wären für eine reibungslose Umsetzung des Projekts wünschenswert, jedoch nicht immer gleich von Anfang an vorhanden. Zum einen gibt es Vorerfahrungen aus beruflicher Sicht, zum anderen eben auch Erfahrungen aus dem Studium mit keinen externen Kenntnissen. Als Lösungsansatz zur Reduzierung dieses Problems sind wir als Gruppe der Meinung, dass es am sinnvollsten wäre komplexe und technisch schwierige Aufgaben an Personen abzugeben, die bereits gute Vorkenntnisse besitzen. Die Strategie dabei ist es, dass sich die Gruppenmitglieder in den zugeteilten Aufgabenbereichen spezialisieren, sodass jeder ausreichend kompetent in seinem Bereich ist, schnell Funktionen implementieren kann und diese auch weniger Fehler aufweisen. Gerade im Hinblick auf die kurze Zeit eines „Sprints“ ist es daher sinnvoll systemkritische Aufgaben an Personen abzugeben, die sich dazu in der Lage sehen das Grundgerüst des Systems zu bauen.

Die Zeit, wie bereits erwähnt, ist ebenfalls ein Risiko, welches wir in die Gruppe der organisatorischen Risiken einordnen. Eineinhalb Wochen Coding für eine funktionsfähige Software und deren Dokumentation ist knapp bemessen und im vollen Umfang nicht umsetzbar. Wir orientieren uns daher an der agilen Entwicklungsmethode Scrum und versuchen Kernprozesse wie zum Beispiel das Daily Retro zu übernehmen, indem wir uns alle zwei Tage über den Stand der Dinge austauschen. Durch gründliche Planung und Priorisierung der wichtigsten Anforderungen, können auch unter Zeitmangel die wichtigsten Funktionen implementiert werden, um einen funktionierenden Prototypen bereitzustellen. Je nachdem wie die Umsetzung funktioniert und wie viel Zeit am Ende übrig bleibt wird in der Gruppe darüber diskutiert, welche weiteren Features möglicherweise übernommen und eingefügt werden. Zudem kann das System in Komponenten aufgeteilt werden, sodass man die Implementierung der verschiedenen Komponenten den Gruppenmitglieder separat zuteilen und somit parallel an verschiedenen Funktionen gearbeitet werden kann.

Ein weiteres Risiko im Hinblick auf die kurze Zeit kann darin bestehen, dass weniger automatisierte Tests geschrieben werden, da diese nicht direkt zum Prototypen beitragen. Die fehlenden Tests können negative Auswirkungen auf die Qualität des Endproduktes haben, weil eventuell Bugs entstehen. Wir haben uns dazu entschieden eine Testphase am Ende der Entwicklung einzuführen, sobald ein lauffähiger Prototyp existiert. So wollen wir zumindest teilweise grobe Fehler in der Software vermeiden.

Des Weiteren haben wir in unseren Meetings die ungenauen Anforderungen des Kunden als ein weiteres Hauptrisiko identifiziert. Es wird gesagt, dass das neue System „weniger Schrott“ sein soll als das Alte. Dies entspricht keiner aussagekräftigen Anforderung für die vom Kunden gewünschten Funktionalitäten. Es ist unklar wie diese Prozesse ablaufen,



sodass man viel spekulieren muss und vermutlich Funktionen dadurch überflüssig sind oder fehlen. Deswegen haben wir in unseren Meetings über mögliche Anforderungen des Systems gesprochen und ausgeschrieben. Die Erstellung von Epics, User Stories und der Zusammenführung im Backlog hilft dabei, das System beziehungsweise das Verfahren besser zu verstehen und die systemkritischen Anforderungen entsprechend umzusetzen.

# Prozessplan

## Entwicklungsprozess

Bei der Entwicklung des Produktes wird ein zu Scrum ähnlicher Entwicklungsprozess verfolgt. Ähnlich zu Scrum, da die Rolle des Scrum-Masters und des Product-Owners durch Teammitglieder belegt werden, da das Team zu klein und unerfahren ist um ein Teammitglied für den Product-Owner oder den Scrum-Master zu entbehren. Das Backlog wird in Zusammenarbeit des gesamten Teams erstellt und priorisiert. Die Sprints werden aufgrund mangelnder Zeit maximal für zwei Tage angesetzt und mindestens alle zwei Tage wird ein Daily abgehalten. Die Meetings werden über Discord gehalten und (außer Daily) vom Schriftführer protokolliert.

## Quellcode und Integration

Der Produzierte Code wird auf dem GitHub Server gespeichert und bei jedem push in jeden Branch von Travis-CI gebaut und getestet. Bei der Zusammenarbeit mehrerer Entwickler wird das Git-Flow-Modell verfolgt, um sicherzustellen, dass es immer einen funktionierenden Branch gibt (master), der nur durch einen Merge request aus dem Develop gemerged werden soll. Dies geschieht nach jedem Sprint. Alle Dokumentationen des Quellcodes und Variablen- und Methodennamen werden in Englisch verfasst.

Es sind insgesamt folgende Module geplant die in einer Docker umgebung integrierbar sind:

- Relational Database
- Backend
- Frontend
- Identification Manager

## Planungsdokumente

Alle Projektplanungs-Dokumente werden über Google Docs erstellt und dort in einem geteilten Ordner gespeichert. Einen "Getting started" Guide des Projekts wird im Git Repository unter README.md verfügbar sein. Alle Planungsdokumente werden in deutscher Sprache verfasst.

## Design der UI

Um einen groben Überblick über alle Funktionen zu verschaffen und die Anforderungsanalyse zu vereinfachen, wurde ein UI Prototyp mit dem kostenlosen online Editor <https://vectr.com/> erstellt. Dieser Prototyp dient auch als ungefähre Designvorgabe bei der Implementierung der UI.