

Projektbericht-

I3_Antragsverwaltungstool

1. Planung [A.G.]

1.1 Kurze Erläuterung des zu lösenden Problems

Der StuRa wollte ein Online Antragsverwaltungstool (auf der StuRa Website) zum vereinfachen des Antragsstellens, der Bearbeitung und des anschließenden Sichern/Speichern der Anträge. (Was im Moment zum Teil noch Analog passiert das heißt dadurch ohne gemeinsame Formatierung und an keiner Stelle automatisiert.)

Unser Tool soll auf funktionaler Ebene erstmal nur Grundfunktionen wie das einreichen ,bearbeiten sowie anschließend das abspeichern übernehmen, nützliche Zusätze dazu wäre das Anträge zusammengefasst und in eine Art Liste gespeichert werden können (Tagesordnung) sowie direkt gedruckt werden können. Zudem soll unsere Implementation als ein Grundgerüst für mögliche weiterentwicklung dienen, eine denkbare weiterentwicklung wäre zum Beispiel eine Version für mobile Geräte.

Auf technischer Ebene wurden wir kaum eingeschränkt (siehe Systemwide Req.). Als Vorlage wurde uns die Website <https://antragsgruen.de/> präsentiert an der wir uns zumindest optisch orientieren konnten.

1.2 Zustand – Beginn Semester 2

Zu Semesterbeginn haben wir die größtenteils abgeschlossene Anforderungsanalyse aus dem letzten Semester übernommen bzw. damit weitergearbeitet. Zudem hatten wir uns letztes Semester schon einige Gedanken über die Tools gemacht ,welche wir nun zur Implementierung nutzen wollten. Wir waren somit fast mit der Elaboration-Phase fertig und konnten mit der Construction-Phase beginnen.

Leider mussten wir feststellen das unsere alte Ansprechpartnerin „Liselotte Ruthe“ vom StuRa sich nicht mehr für uns verantwortlich fühlte bzw. aus dem StuRa ausgetreten ist. Zudem war "Axel Schiller" zwar der Antragsteller jedoch nicht unser Ansprechpartner, daher konnten wir zu Beginn dieses Semester erstmal keine Rücksprache mit unserem eigentlichen Kunden halten, später erfüllte diese Position dann "Johann Boxberger".

Hinzu kommt das Corona bedingt ein Lockdown vorherrscht welcher die Kommunikation in den virtuellen Raum lenkt, dies hat uns nicht groß behindert war jedoch im Bereich der Motivation und Kommunikation durchaus merkbar.

Hier noch Essence Navigator Bild einfügen IT-Plan 3-4

1.3 Projektorganisation

1.3.1 Teamaufbau und Aufgabenbereiche

Unser Team hat sich von 8 auf 5 Personen (Jacob Flaskamp [J.F.], Ruben-David Kraus[R.K.], Richard Müller[R.M.], Duy Tien Nguyen[T.N], Andreas Gröber[A.G]) verkleinert, welche gelegentlich von unserem Coach Felix Müller unterstützt wurde. Daraus folgt das wir auch unsere Rollen und Aufgabenbereiche angepasst haben bzw. mussten. Die Kürzel in den eckigen Klammern spiegeln in diesem Dokument die Namen der Autoren wieder.

Ruben und Richard haben größtenteils selbständig die Implementierung und das entwerfen der Architektur sowie die Dokumentierung darüber übernommen. Jacob und Tien haben das Testen (den Teil den Ruben und Richard nicht getestet haben) und dessen Dokumentierung übernommen. Andreas hat die Projektplanung sowie den leitenden Teil bei der Analyse-Dokumentation übernommen mit Ruben und Jacob als unterstützende Personen.

Zudem ist Ruben Mitglied im StuRa und somit ein schneller Kommunikationsweg zu unserem Kunden, bei Fragen oder Rücksprache an unseren Kunden bzw. der gesamten StuRa Leitung war das überaus hilfreich.

1.3.2 Kommunikation und Meetings

Wir haben uns am Anfang des Semesters darauf geeinigt uns einmal in der Woche, coronabedingt in Form eines Discord-Gruppenanruf, zu treffen. Dies ist auch jede Woche Am Montag zwischen 17:00 und 19:00 passiert die Meetings haben je nach Inhalt etwa 30 Minuten bis 90 Minuten in Anspruch genommen. Zudem haben wir uns Freitag von 8:00 bis 10:00 bei Bedarf getroffen, falls wir Zwischenergebnisse teilen, Meinungen von den anderen Teammitgliedern einholen oder zusammen etwas erarbeiten wollten.

Außerdem haben wir Whatsapp benutzt um diese Treffen zu koordinieren bzw. bei Bedarf zu verschieben , falls jemand mal nicht konnte.

Da wir zu beginn des Semesters keinen wirklichen Kunden als Ansprechpartner hatten, kamen bis mitte Mai auch keine Kundenmeetings zu stand, dort haben wir dann Johann Boxberger als neuen Ansprechpartner kennen gelernt und darauf geeinigt das wir uns aller 2 Wochen Montags 16:00 zum kurzen Informationsaustausch treffen, zudem konnten wir uns bei Bedarf auf Discord Fragen stellen bzw. diese beantworten. Diese Treffen fanden immer in der Mitte einer Iteration statt, falls zu betreffender Iteration noch fragen bestanden, zudem wurde der Kunde bei diesen Meetings über unsere Fortschritte informiert und sofern möglich, genannte Fortschritte oder neue Funktionalitäten gezeigt, um die Meinung des Kunden bzw. Anwenders einzuholen und bei Bedarf noch Änderungen vorzunehmen.

Wichtige Informationen aus den Meetings (Team sowie Kunden) wurden in einem Protokoll festgehalten, um bei Bedarf besprochenes bzw. Ergebnis dessen nochmal zu rekapitulieren.

Da wir ihre Empfehlung für sinnvoll erachten und abgesehen von SE-1 noch nie jemand von uns innerhalb eines OpenUP bzw. eines Projekts dieser Größe gearbeitet hat, haben wir die Iterationslänge auf 2 Wochen festgelegt. dadurch haben wir ein Größeres Meeting jeden 2. Montag in welcher wir die Iteration reviewen und Ziele sowie das vorgehen in der nächsten besprechen.

An den anderen Montagen besprechen wir kurz unsere zwischen Ergebnisse und klären gegebenenfalls Fragen oder geben Hilfestellung falls nötig.

1.3.3 Model/Idee der Planung

Wie von ihnen vorgegeben haben wir unser Projekt als OpenUP umgesetzt bzw. dies best möglichst versucht. Wir haben dabei zum Großteil Iterativ gearbeitet zum anderen Teil aber auch in einer Art V-Modell um eine Art Überblick zu behalten und uns nicht an Kleinigkeiten aufzuhalten.

Grob gesehen wussten wir etwa in welchen Wochen des Semesters wir den Entwurf abschließen, implementieren und testen müssen um den Umfang aller Use-Cases und deren Funktionalitäten bereit zu stellen. (Auch wenn die Hälfte der Use-Cases am Ende noch nicht implementiert worden sind.)

Innerhalb der einzelnen Iterationen haben wir jedoch nochmal alles überarbeitet sowie verbessert, bei Bedarf auf Kundenwünsche reagiert und unser Ergebnis eher an unsere aufgewandte Zeit angepasst als anders herum.

Unsere Hauptanforderungen wurden erfüllt und sind in unserer Software integriert, zudem haben wir beim arbeiten darauf geachtet das andere Entwickler, möglicherweise der Jahrgang nach uns, unser System ohne große Schwierigkeiten weiterentwickeln können.

Beim planen haben Ich [A.G.] vorerst nicht den Aufwand einzelner Tasks abgeschätzt dies wäre jedoch wahrscheinlich sinnvoll gewesen, Ich habe zum Ende des Projekts hin erst damit angefangen und es hat beim „gerechten“ verteilen der Tasks und beim Planen über einen größeren Zeitraum durchaus Sinn gemacht. Zudem wurden sich Gedanken zu Risiken und deren Lösung in Form einer Risklist gemacht zum Teil konnten dadurch Probleme vermieden werden oder schnell gelöst werden. Einnige Sachen haben wir jedoch in der Weise wie sie passiert sind, nicht kommen gesehen.

1.3.4 Way of Working

1.3.4.1 Eingesetzte Tools

Wir haben unser Projekt in einem Github-Repository angelegt und haben dort die Möglichkeit genutzt mit Github-Issues zu arbeiten.

Zur Kommunikation wurden Discord sowie Whatsapp (Mobil) genutzt.

Wir haben uns zudem während des ganzen Projektes an den Meilensteinen des Essence-Navigators sowie der OpenUp-Vorlage orientiert.

Um den Quellcode zu dokumentieren wurde Pydoctor genutzt.

Zur Erstellung der Dokumentation wurde hauptsächlich VS-Code verwendet unter Anleitung des Praktikums.

Wie erwähnt wurde eine Risklist erstellt.

Eine Übersicht über die eingesetzten Tools bei der Implementierung folgt dann in Punkt 2.2.3.4 in

diesem Dokument.

1.3.4.2 Eingesetzte Strategien

Höchste Priorität bei der Start-Phase war es Momentum (eine Eigendynamik oder Workflow) zu erzeugen, dies wurde über wöchentliche Meetings bzw. durch regelmäßiges arbeiten am Projekt sowie der Spezialisierung in die verschiedenen Rollen, um Multitasking zu vermeiden, sowie über Konfliktvermeidung auch erreicht.

Das Erzeugen des Momentums erfordert am Anfang möglicherweise etwas mehr Zeit, aber es ermöglicht oder vereinfacht das kontinuierliche sowie konzentrierte arbeiten am Projekt und sorgt schlussendlich für Fortschritt an den Hauptfunktionalitäten. Leider wurde dies an einigen Stellen durch unseren Kunden etwas gestört z.B. als wir etwa einen Monat keine Reaktion auf die Nachfrage nach unserem zugesagten Server bekamen und nicht wirklich wussten wie es weiter geht.

Konflikte wurden größtenteils am Anfang des Projekts beseitigt bzw. direkt am Zeitpunkt des Auftretens über Kompromisse oder einer klaren Entscheidung über das weitere Verfahren. Es kam nie zu wirklich großen Diskussionen oder Auseinandersetzungen, grundsätzlich hat jeder sein bestmögliches gegeben um das Projekt erfolgreich abzuschließen.

Ziele wurden nach dem SMART-Format ([https://de.wikipedia.org/wiki/SMART_\(Projektmanagement\)](https://de.wikipedia.org/wiki/SMART_(Projektmanagement))) formuliert, um die Sinnhaftigkeit sowie eine gewisse Effektivität dieser zu gewährleisten. Ich persönlich empfinde diese Art der Zielsetzung als sinnvoll, habe allerdings keinen wirklichen Vergleichswert um sagen zu können das es effektiv ist bzw. unserem Projekt weitergeholfen hat.

1.3.4.3 Grundregeln

Grundsätzlich wurde versucht ein Arbeitsumfeld zu schaffen indem man gerne und mit möglichst wenig Störungen, auf das Ziel eines möglichst gut (nach Kundenvorstellung) funktionierenden Produktes hinarbeitet sowie einer dementsprechend guten Bewertung des Projektes.

Außerdem sollte jedes Teammitglied etwa den gleichen zeitlichen Aufwand bzw. Leistung in das Projekt investieren um Fairness zu gewährleisten, pro Woche sollte dieser Umfang etwa in 2 bis 4 (Teammeetings zählen nicht zu dieser Zeit) Zeitstunden die sinnvoll genutzt werden, liegen. Dies ist aber nicht immer gelungen. Der Hauptgrund dafür war wahrscheinlich die eher „kurzsichtige“ Planung, als Beispiel hatten wir in einigen Iterationen keine Analyse bzw. wirkliche Testaufgaben außer die Dokumentation zu überarbeiten während unser Implementierungsteam eigentlich immer beschäftigt war.

2. Durchführung [A.G,R.K,R.M,J.F]

2.1 Überblick über Phasen und Iterationen [A.G]

Wie in Teil 1.3.3 erwähnt hatten wir eine Grobplanung die durch Störfaktoren (in 3.3.1 behandelt) etwas nach hinten verschoben wurde. Die Iterationen 1 bis 3 wurden in SE-1 abgehandelt und es finden sich keine Ausführung dieser in diesem Dokument.

Ab dem 03.07. werden keine Änderungen am Quellcode unseres Projektes mehr vorgenommen bis dahin werden noch Kleinigkeiten verändert die keine großen Ausführungen benötigen.

Iteration	geplante Phase (Sicht Anfang SE-2)	tatsächliche Phase	Datum	Aufgaben	Besonderheiten/Ereignisse
1	Iception	Iception	SE-I	Anforderungsanalyse	-
2	Iception	Iception	SE-I	Anforderungsanalyse	-
3	Elaboration	Elaboration	SE-I	Anforderungsanalyse, erstes testen der Tools(Implementierung)	Server angefragt
4	Elaboration /Construction	Elaboration	5.04 - 19.04	Requirements genau abklären, Vision nochmal abgleichen, Dokumentation SE-1 überarbeiten, Prototypen bauen(Implementierung)	Keinen Ansprechpartner beim Stura mehr
5	Construction	Construction /Elaboration	19.04 - 03.05.	Testcases, Erste Funktionalitäten im Prototypen implementieren, Dokumentation überarbeiten (Domainmodell + Use-Cases sowie Use Case Modell)	Prototyp ist nun eher eine Demo-Version
6	Construction	Construction /Elaboration /Inception	03.05 - 17.05	Teilweise neue Anforderungsanalyse, Demovorstellen → Meinungen einholen, Überarbeitung Vision + Use-Cases, Weiter an Funktionalitäten arbeiten/austesten was geht(implementierung)	13.Mai Johann Boxberger als neuer Ansprechpartner und neuen Kunden kennengelernt
7	Construction /Transition	Construction	17.05- 31.05	Datenbank sowie verknüpfung erstellen, Dokumentation überarbeiten, Möglichkeit des Logins bereitstellen/über Lösung nachdenken	17.05 StuRa-Server zugesagt, Demo-Version erfüllt Grundfunktionen

Iteration	geplante Phase (Sicht Anfang SE-2)	tatsächliche Phase	Datum	Aufgaben	Besonderheiten/Ereignisse
8	Transition	Construction	31.05-14.06	einarbeiten in neues Webinterface, aufsetzen eines Apache-Webservers, erste tests starten	Wechsel aufs Django webinterface
9	Transition	Transition/Construction	14.06-28.06	Projekt Implementation auf Server kopieren und dort testen, letzte Funktionalitäten hinzufügen und diese testen, Bericht/Dokumentationen Zuständigkeitsbereiche klären und auf grobe Struktur einnigen	21.06 StuRa stellt Server bereit
10	Bericht schreiben	Bericht schreiben/Transition	28.06-04.07	letzte Änderungen vornehmen, Bugsfixen, Übergabe vorbereiten, Bericht schreiben	01.07 Projekt Vorstellung SE-2

wurde erfüllt, noch nicht vollständig erfüllt, nicht erfüllt, extra,

2.2 Gliederung der Durchführung [A.G]

Wir werden die nun folgende Dokumentation der Durchführung anhand der 4 Projekt-Phasen Inception, Elaboration, Construction, and Transition gliedern. Die Tabelle in Punkt 2.1 soll bei der Orientierung helfen, da wir zum Teil innerhalb der Iteration springen (siehe Iteration 6) und dieser Teil des Projektes aus der Sicht von 4 Personen zusammengetragen wird.

2.2.1 Inception-Phase [A.G,R.K,R.M,J.F]

2.2.1.1 Zustand

Diese Phase betrifft die Iterationen 1 bis 3 sowie 6, da unser Auftragsteller wechselte mussten wir auch unsere Anforderungsanalyse an ihn anpassen müssen. Dies ist leider erst ab dem 13.Mai passiert.

Dazu kommt das unsere Dokumente aus SE-1 noch sehr fehlerhaft waren, diese konnten wir zum Teil aber auch ohne die Hilfe unseres Kunden anpassen. Grob gesehen haben wir bestimmt nochmal etwa 1/3 der gesamten Anforderungsanalyse machen müssen darunter fallen Teile der Dokumentation, Use-Cases und Vision.

2.2.1.2 Ziele

Ziel	Datum	Erläuterung	Erfüllt?
Vollständige Anforderungsanalyse	SE-1	Wurde aufgrund des Kundenwechsels und fehlern erneuert	Nein
erste Anforderungsanalyse	05.04 bzw. 17.05	Vision und Use-Cases an Johann Boxbergers wünsche angepasst	Ja
Theoretischen Plan haben was zu Implementieren ist	17.05	Wissen wie man Kundenwünsche in der Theorie umsetzt	Ja

2.2.1.3 Probleme

Unser Hauptproblem bestand darin das wir nach dem Semesterwechsel keinen Ansprechpartner mehr beim StuRa hatten und zudem uns keiner weiteres Sagen konnte oder wollte, bis zu dem Zeitpunkt als Johann die Rolle des Kunden und Nutzer einnahm. Ein weiteres Problem bestand in unserer fehlerhaften Analyse aus SE-1 die wir aber relativ schnell mitunter auch mit Hilfe von unserem Coach Felix schnell verbessert haben sobald dies möglich war.

2.2.1.4 Aktivitäten

Hauptaktivitäten

Unsere Hauptaktivität bestand darin das wir die Webanwendung so gut es geht vollumfänglich Implimentieren zu können. Da wir nicht auf ein bestehendes System aufsetzen beziehungsweise weiterentwickeln gab es einige Herausforderungen denen Wir uns als Gruppe stellen mussten.

- Auswahl der Anwendungsart = Webanwendung
- Auswahl der dafür verfügbaren Techniken = HTML, etc.
- Beschaffung der Hardware vom StuRa = Hat geklappt aber sehr spät im Projekt
- Ausschließliche Verwendung von Open Source Lizenzen

Anforderungserhebung und -analyse

Der StuRa vorderte von uns ein Tool mit welchen User unbestimmter Art, meistens aber Professoren und HTW Studeten, verschiedene Arten von Anträgen beim StuRa der HTW-Dresden einreichen können. Das sollte sehr einfach funktionieren so das die Bedinung leicht ist und gut verständlich ist. Desweiteren soll das zu entwickelnde Tool auch die möglichkeit haben die verschieden eingereichten Anträge wieder in einer Listenform anzuzeigen. Zudem soll eine Bearbeitungs möglichkeit für StuRa Mitglieder geschaffen werden. Also ein Tool für das Einrechn und Bearbeiten von StuRa Anträgen war gefordert.

Nach dem esten Kundengespräch im neuen Semester überprüften wir nocheinmal unseren

Wissenstand und fragten nach Änderungen.

Problemstellung	Analyse
Einfache Bedienbarkeit	Verwendung von HTML Formularen
Verschiedene Antragsarten	Die verschiedenen Antragsarten die der StuRa implementiert haben möchte erkennen und von Antragsformularen für die Händische Ausfüllung in einfache HTML Formulare umwandeln
Ausgabe der eingereichten Anträge in Listenform	Eigene HTML seite die als Übersicht der Einzelnen Anträge dienen soll
Bearbeitungs möglichkeit für StuRa Mitglieder	Verwendung eines Login systems der Verschiedene Funktionen sperrt oder zugänglich macht
Bearbeitung der eingereichten Anträge	StuRa eigenes Formular für die Bearbeitung der jeweiligen Anträge zuzüglich der Elemente für die Beschlussfassung des Sturas
Open Source Lizenzen verwenden	Nur auf Open Source Frameworks, etc. zurückgreifen

2.2.1.5 Review

Insgesamt wurde unsere neue Vision und die vorgeschlagenen Use-Cases (die Implementierung aller 8 war damals noch angedacht aber eher nicht realisierbar) angenommen.

Grundsätzlich gab es wenig Input was Requirements und Anforderungen angeht vom Kunden selbst, wir haben das meiste als Idee vorgestellt die wurden dann angenommen oder abgelehnt, mitunter hatte Johann selber noch Ansprüche an das Layout oder Design gestellt die wir beachtet haben. Johann war von unserer überarbeiteten Vision an das Projekt überzeugt und wir konnten die eigentlich Arbeit eigentlich schnell wieder aufnehmen.

Zudem haben wir Felix nochmal über unsere Analyse (Domainmodell,Vision,ProjectPlan) drüberschauen lassen und haben seinen Input umgesetzt dies greift zum Teil schon über in die Elaboration Phase.

2.2.2 Elaboration-Phase [A.G,R.K,R.M,J.F]

2.2.2.1 Zustand

Wir waren mitten in der Elaboration Phase beim Übergang von SE1 zu SE2, unsere Anforderungsanalyse war fertig aber fehlerhaft, es wurde sich darauf geeinigt welche Sprachen bzw. Tools man für die Implementierung verwenden möchte. Diese Phase betrifft die Iterationen 3 bis 6

2.2.2.2 Ziele

Ziel	Datum	Erläuterung	Erfüllt?
Plan für weiteres vorgehen	17.05	ein neue Grobplanung um die Use-Cases noch zu Implementieren	erstellt aber nicht realisierbar → Ergebnis eher dem Aufwand angepasst als anders herum
Neue Anforderungsanalyse	17.05	Vision,Use-Cases,Projectplan und Domainmodell erneuert	Ja
Test-Cases erstellen	19.04 bzw 17.05	-	Ja
Architektur festgelegt	03.05 bzw. 17.05	Auf Systemstruktur einigen, Programmiersprache wählen	Teilweise → geplante Architektur funktionierte nur Teilweise und musste zum Teil neu konstruiert werden
Aufgabenbereiche zuteilen	03.05	Im Bereich test sowie Implementierung einzelne Tasks zuweisen und grobe Problemstellungen zuweisen	-

2.2.2.3 Probleme

Fehlerhafte Anforderungsanalyse (wurde in der Inception-Phase abgehandelt)

Wir hatten nicht wirklich eine Ahnung wo wir Anfangen sollten, wir hatten eine Menge vor und keinen richtigen Anhaltspunkt wo wir starten sollten bzw. mit welchen Tools oder welchen Programmiersprachen. Somit musste unsere Implementierungsgruppe mithilfe der Testgruppe ausprobieren mit welchen Hilfsmitteln,Sprachen,Tools wir arbeiten wollen bzw. mit welchen es sinnvoll ist unser Projekt um zu setzen dies braucht eine Menge Zeit und war nicht immer Risikofrei.

Wir haben keinen Analysten aus SE1 mehr und mussten uns Wissen über den StuRa neu aneignen um weiter in die Dömäne zu schauen um das Problem richtig an zu gehen bzw. um das richtige Problem zu lösen.

2.2.2.4 Aktivitäten

Hauptaktivitäten

Anforderungserhebung und -analyse

Nach der groben Richtung und der Erörterung der Domäne, sowie des Fachlichen Problems der Aufgabenstellung, führten wir zahlreiche Meetings mit den Auftraggebern durch. Hier haben wir den Fokus auf alle möglichen Uses-Cases der Anwendung gelegt. Aufbauend auf diesem Wissen,

haben wir die ersten Use-Case-Diagramme erarbeitet und begonnen das Use-Case-Modell anzufertigen. Dies hat uns anfänglich eher viele Probleme bereitet. Leider war das einarbeiten in die Prozesse des StuRa sehr anstrengend und zeitintensiv. Da es sich beim StuRa um eine sehr komplexe Struktur bzw. Beziehung aus Personen Positionen und Förmlichkeiten handelt. Hier gibt es viele Punkte die beachtet werden müssen welche für uns vollumfänglich kaum zu verstehen waren. Hier hatten wir viele Nachfragen bei jedem Meeting mit den Themenstellern bis wir die ganze Domäne vollumfänglich verstanden hatten.

Entwurf

Im Entwurf haben wir begonnen zu Planen in welche Richtung wir technisch gehen wollen. Hier haben wir den Fokus auf zu nutzende Technologien gelegt. Wir haben uns informiert welche Komponenten für unser System als Webanwendung nötig sind. In dieser Phase hat sich herauskristalisiert das wir einen Webserver benötigen. Die Lösung zur Haltung der Daten erschien erst als txt Dateien sinnvoll, wir behielten uns jedoch eine Datenbanklösung im Hinterkopf.

Wir befassten uns außerdem welche einzelnen Komponenten des Systems notwendig sein sollten. Wir arbeiteten aus der Analyse heraus, das wir Funktionalitäten zum Einfügen und zum Anzeigen von Anträgen bereitstellen müssen. Mit diesem Wissen planten wir ebenfalls die Kommunkation mit dem Webserver und den bis jetzt vorhandenen Datenstrukturen.

Außerdem haben wir abgeklärt welche Programmiersprachen bzw. welche Sprachen generell für einen Webservice benötigt werden. Hier haben wir uns auf die im Frontend dominanten Sprachen: HTML / CSS / JavaScript geeinigt. Hier hatte Richard bereits ein kleines Konzept im Kopf, welches Design und welche Stuktur die Webseite haben sollte. Da wir im Backend eine möglichst gut verständliche Sprache nutzen sollten, haben wir uns auch aufgrund von Vorkentnisse auf Python geeinigt. Jedoch schien es alles deutlich einfacher als es am Ende war. Hier plante wir die erste Kommunikation mit der Datenbank, welche über einfache Scripte abfolgen sollte. Der Plan sah vor das wir in SQL Abfragen formulieren wollten, welche wir dann in die Anwednung übernehmen wollten. Darauf aufbauend befasste sich Ruben , da er Vorwissen in Python hatte, mit dem Backend Part und den damit verbundenen Möglichkeiten der Umsetzung. Da wir es hier möglichst einfach halten wollten, wollten wir zunächst auf ein Framework verzichten und planten zuerst nur mit wenigen Bibliotheken und einfachen Skripten.

Zusätzlich planten wir erste Versionen des Webserver und beschäftigten uns mit der groben Infrastruktur im StuRa. Hier haben wir uns angeeignet, wie ein Webserver einzurichten ist und Erfahren wie dieser im StuRa bereitgestellt wird.

Implementierung

Hier gab Richard einen Einstieg in HTML als mit größte Komponente des Frontends. Da er schon Erfahrungen mit Frontend-Sprachen hatte, konnte er uns in diesem Fall viel Wissen vermitteln und uns alle anfallenden Fragen bereits erklären. Danach haben wir uns alle mit den geplanten Technologien auseinandergesetzt, aber gemerkt, dass ohne konkretes Ziel ein Anwenden der Sprachen kaum sinnvoll war. Es schien uns am sinnvollsten Richard (mit den Erfahrungen im Frontend) für selbigen Part einzusetzen. Wir hatten hier vorerst eine unterstützende Funktion. Dabei wurde bereits eine erste Demo Version von Richard erstellt. Hier haben wir uns ein gewisses Konzept für das Frontend überlegt, um einerseits eine gewisse Einheitlichkeit zu bekommen und um das Frontend ebenfalls nach einem Plan entwickeln zu können. Da das Feedback zum

Größtenteil positiv war, wurde die Demo Version weiterverfolgt.

Im Backend fehlte uns leider das nötige Wissen um die einsetzbaren Technologien möglichst gut abzuschätzen. Es wurden mit Python erste Scripte, die CRUD Operationen ausführen sollen, zu planen / zu erstellen. Die Scripte bauten auf den HTML Dateien der Demo auf und sollten den input der HTML Forms entgegennehmen und in die Datenbank schreiben.

Test

Es wurden bereits erste Testfälle für die Demo geplant. Leider begrenzte sich das ganze nur auf einzelne Dummy clicks ohne Funktion da wir weder einen Webserver, noch eine Datenbank online zur Verfügung hatten. Hier haben wir versucht das ganze lokal über Container zu betreiben.

Dokumentation

Einzelne Dokumente wurden überarbeitet darunter Glossary, Domainmodell, Use-Casemodell, Lessons learned (eingefügt). Zudem wurde die Planung etwas abgeändert

2.2.2.5 Review

Richards Vorwissen und Rubens begeisterungsfähigkeit für das Projekt waren in der Phase unnerlässlich, gegen Ende dieser Phase (etwa hälfte des Semesters) fiel eigentlich schon die Entscheidung ob wir die Grundanforderungen Implementieren können oder nicht. Bekräftigt wurde dies als unser Kunde sich unsere Demo-Version angeschaut hat und dies in etwa seine Vorstellung eines Online Antragsverwaltungstools war.

In der Überleitung von Elobaration zu Konstruktion-Phase gab es auf einmal Fortschritt zum ersten mal seit Start des Semesters konnten wir richtig an unserem Projekt in zusammenarbeit mit unserem Kunden arbeiten, das einzige was an der Stelle noch fehlte war unser StuRa Server.

Und Obwohl die Situation erst sehr angespannt und auch nicht gerade gut aussah möchte Ich [A.G] an dieser Stelle sagen, dass wir alle darauf vertraut haben das wir es noch schaffen können, das Projekt zumindest im Ansatz zu vollenden, jeder von uns hat auch wenn nicht immer motiviert sein Ding bzw. seine Aufgabe durchgezogen, keiner hat sich auch nur im Ansatz beschwert und Jeder ist relativ entspannt geblieben und hat geschaut das er bzw. wir das Beste aus einer eher schlechten Situation machen.

2.2.3 Construction-Phase [A.G,R.K,R.M,J.F]

2.2.3.1 Zustand

Diese Phase hat Bestandteile in den Iterationen 5-8. Wie schon erwähnt habe wir hier große Fortschritte in Hinblick auf die Implementierung erzielen können.

2.2.3.2 Ziele

Ziel	Datum	Erläuterung	erfüllt?
Grundanforderungen implementiert	14.06	Das Ausfüllen von Anträgen sowie bearbeiten soll auf dem lokal host möglich sein	Ja
Use-Cases implementiert bzw. funktionsfähig	21.06 bzw. 28.06	Nur 4 von 8 wurden implementiert und stehen derzeit dem Nutzer zur Verfügung	Teilweise
Features zur Benutzerfreundlichkeit bzw. Usability implementiert	14 bzw. 21.06	Hilfetexte auf der Landingpage sowie innerhalb der Anträge eingefügt	Ja
Unser Projekt auf Stura Umgebung Installiert	28.06	Unsere lokal gehostete Architektur nun auf den Stura Webserver verlagern	Ja

2.2.3.3 Probleme

Wir standen unter Imensem Zeitdruck und mussten uns in der Anfangsphase vielleicht auf zu viele Dinge auf ein mal konzentrieren. Zudem war meine [A.G] Planung mitunter zu locker bzw. kurzsichtig und wir hatten sehr viel zu tun.

Zudem kommt das wie schon erwähnt uns unser Server knapp einen bis anderthalb Monate zu Spät übergeben wurde.

2.2.3.4 Aktivitäten

Verwendete Tools zur Implementierung

- Jet Brains
 - DataGrip
 - PHP Storm
 - PyCharm
- MySQL
 - DataGrip
- VM-ware Player
- Internetbrowser
 - Google Chrome (V. 91.0.4472.114)
 - Safari (V. 14.0.3)
 - Microsoft Edge (V. 91.0.864.59)
 - Mozilla Firefox (V. 89.0.2)

Implementierung

Als Entwicklungstool für die Webanwendung im Frontend wurde anfangs Jet Brains PHP Storm mit der Studenten Lizenz verwendet da dieses Tool einige sehr nützliche Features mitbrachte wie die eien Projektextplorer eine Implementierung von Git was die Dateiverwaltung sehr einfach gestaltete sowie einen Tool eigenen Browser der die HTML Seiten korrekt darstellen kann dies machte es möglich live Änderungen am HTML, CSS code vorzunehmen und dierekt ein Ergebniss zu sehen ohne erst eine Browsereite neu laden zu müssen. Das ist eine Eigenschaft des Programm die Ich (Richard Müller) sehr schätze.

Die erste Version der Webanwendung beschrieb nur die Grafische Oberfläche für den Endbenutzer ohne ein Backend. Diese Version wurde mit dem Kunden regelmäßig abgestimmt und nach seinen Wünschen und Vorstellungen erweitert und angepasst.

Bei der Implementierung der Backend Lösung stießen wir auf etliche Probleme. Die erste geplante Version des Tools sah vor das eingereichte Formulare als Textdateien auf dem Server abgespeichert werden. Mit dieser Lösung wollten wir auf eine Datenbank verzichten, da in unseren Augen eine Datenbank zu viele Funktionen hat die Wir nicht benötigen. Da allerdigs der Aufwand eine Textdatei mit den Inputs eines HTML Formulars zu erstellen, abzuspeichern und dann wieder in ein HTML Formular einzulesen viel zu umständlich war entschlossen wir uns kurzfristig auf eine MySQL Datenbank umzuschwitchen da so die Datenspeicherung über eine Tabellenverwaltung deutlich vereinfacht wurde und eine bessere übersicht der Daten ermöglichte.

Das Einrichten der Datenbank geschah über eine lokale MySQL Server Installation die erst in einer Virtuellen-Umgebung und dann später auf dem "echten" Server des StuRas installiert wurde.

Von einem Problem in das nächste. Die Datenbank lief für sich und die HTML-Seite lief für sich, aber es gab keine Möglichkeit diese Sinnvoll zu verbinden. Der este Plan sah vor das nach unserer Entscheidung für Python unter zuhelfenahme von PHP-Scripten als "Auslöser" die Input Daten aus den Formularen mittels "insert" Anweisungen in die Datenbank geschrieben werden sollten. Dies war nicht möglich da der verwendete Webserver Apache PHP scripte nicht ausführt. Auch nach mehrfacher Konfiguration des Webserver weigerte er sich immernoch die Scripte ordnungsgemäß oder überhaupt auszuführen.

Da wir uns am Anfang des Semesters gegen ein Web Framework entschieden haben, da keines der Teammitglieder Erfahrungen mit dem Arbeiten mit Web Frameworks hat, oder einen HTML Bauskasten o.Ä. mussten wir einsehen das wir mit unserer Lösung nicht weiterkahmen. Daraufhin Entschlossen Wir das die Webanwendung auf das Django Framework umgebaut werden muss. Dies bereitete einige Probleme da sich das Implementierungs-Team erst auf das Framework einarbeiten musste und dann die Frontend Lösung umschreiben musste, dies hat einiges an Zeit beansprucht. Die Syntax für das incuden externer CSS und Java Script Files war eine andere als im herkömmlichen HTML, da das Framework eine eigene File- und Arbeitsstruktur mitbrachte.

Der Umbau erfolgte zuerst nur im Backend, so dass es erstmal kleine Tests im neuen Framework gab. Der Vorteil von Django war hier ganz klar, dass eine Kommunikation mit der Datenbank nicht selbst geschehen muss. Hier wird alles direkt vom Framework verwaltet und unser Problem mit der Datenbank wurde gelöst.

Django brachte aber noch mehr Vorteile und Möglichkeiten mit die wir so gut es möglich war versuchten für unsere Zwecke zu nutzen. Wir verwarfen die Idee mit PHP srripten und stellten

unsere Kommunikation auf pures Python um da Django ein in Python geschriebenes Framework ist. Damit konnten wir die Formulare einfach anhand der input/textarea/selects namen/values auslesen und in die Datenbank schreiben.

Parallel dazu wurde an einer Login lösung gearbeitet. Die erste herangehenweise war eine htaccess Datei als Login- und Sicherheitslösung zu verwenden da der geplante Hauptwebserver Apache war. Allerdings war dies nicht möglich da die htaccess Datei nicht ohne weiteres mit Django kompartibel war und somit ein Datenaustausch von Login-Informationen erschwert wurde. Lösung des ganzen war die Erstellung einer eigenen Loginpage der Frontanwendung.

Django bot eine eigene Admin Seite an die mit diversen Funktionen ausgestattet war, wie das Erstellen und Bearbeiten von Benutzern oder das Auslesen und Manipulieren der Datenbank Tabellen. Dies macht die vorherige Entwicklung und Einbettung einer eigenen Adminseite überflüssig. Zudem ist die Django interne Seite mit einigen Sicherheits features ausgestattet die bei der eingenen Lösung nur schwer bis garnicht Implementiert werden könnte. Darunter fällt beispielsweise die Unterscheidung zwischen normalen Benutzer und Administrator da nur ein Admin die Admin-Seite aufrufen können soll.

Da unser StuRa Server trotz mehrfacher Nachfrage beim StuRa auf sich warten lies mussten wir kreativ werden für die weitere Implemtierung. Dazu schafften wir eine lokale installation unserer einzelnen Lösungen. Konkret wurde eine Linux VM mit Ubuntu aufgesetzt mit der ein MySQL Datenbankserver betrieben wurde. Jet Brains PyCharm brachte die Django Framework Serverinstans mit, mit welcher es möglich war unter localhost ein Server zu testen. Mit dieser spartanischen Lösung wurde weiterentwickelt und debuggt. Das Problem war nur das diese Installation auf einem privaten Rechner eines Gruppenmitglieds installiert war und so nur eine Person den Code der anderen ausführen konnte. Dies stelle kein großes Problem dar da unser Programm für die Meetings (Discord) die Möglichkeit anbot live eine Bildschirmübertragung zu starten damit jeder verfolgen konnte was ging und was nicht.

Mit diesen zum Teil immer wiederkehrenden Problemen wurde nun weiter implementiert, debuggt und die Fortschritte mit dem Kunden besprochen bis ca. Mitte Juni der StuRa uns einen Server bereitstellen konnte. Damit konnte die Finale Implementierungsphase sowie die finale Installation beginnen. Diese Phase begann auch schon mit dem erstem Problem, der Server wollte das Djangoprojekt nicht ausführen da ein wsgi error ausgab. Dieses Problem wurde mit einem Guide behoben. Damit wurde Django und die dazu passende Python Instanz in eine Virtuelle-Umgebung kopiert, installiert und Konfiguriert.

Die grundlegene Kommunikation sieht vor das der Apache Webserver die hereinkommenden Anfragen der User an nimmt und an den Djangoserver weiterleitet und umgekehrt, dazu kommt das der Webserver die Antwort von Django dem User darstellt. Die Schwierigkeit dabei lag daran dem Server einen weiteren Virtuellen-Host zu konfigurieren sowie die Firewall zu bearbeiten.

Das installieren des MySQL Servers machte keine Probleme und verlief reibungslos genau so wie das tauschen der Django Projekte Fehlerfrei lief.

Im letzten Schritt der Implemetierungsphase wurden Bugs behoben und der Code aufgeräumt, nicht genutzte Scripte und Funktionen entfernt oder überarbeitet.

Es wurde für das Starten des Servers sowie das migrieren neuer Daten ein Shellscript geschrieben, welches in den Autostart des Linux Servers gebracht werden sollte. Dies war allerdings nicht

Erfolgreich. Das Script welches Problemlos den Server startet und die Daten migriert wurde nicht vom Server automatisch im Startup gestartet. Es wurde versucht über eine rc.lokal Datei zu starten, nicht erfolgreich. Das Erstellen und aktivieren eines eigenen Services, nicht erfolgreich. Das verwenden einer Crontabelle, nicht erfolgreich. Das Kopieren der Datei in das init.d Verzeichnis mit anschließender Konfiguration, nicht erfolgreich. Da dies keine Anforderung war und nur der einfachen Wartbarkeit dienen sollte wurde die Idee verworfen und der Prozess in eine Screen-Session geschickt. Diese muss nach jedem reboot des Servers neu manuell gestartet werden. Der Server hat keinen automatischen reboot konfiguriert.

Dokumentation und Test

Wir haben versucht dem Implementierungsteam zu helfen wo es möglich war. Dies erwies sich oft als schwierig da unser Implementierungsteam zwar großen Fortschritt machte aber dafür auch in eine Art "Bubble" gerieten in die man fachlich schwer einsteigen konnte. Dazu haben wir die einzelnen Dokumentationen die noch zu überarbeiten waren überarbeitet und haben angefangen (wo es schon möglich war) an den einzelnen Abgabe-Dokumentationen zu arbeiten. Es wurden zudem erste Tests an der lokal-Host Variante durchgeführt, später dann am StuRa-Server

2.2.3.5 Review

Auf der einen Seite kam es innerhalb dieser Phase zu Rückschlägen die man hätte vermeiden können, zum einen hätte wir die Tools zum Start besser testen müssen und spezifisch darauf prüfen was sie am Ende entwerfen sollen bzw. wie die einzelnen Systembausteine am Ende miteinander kommunizieren sollen. Man hätte möglicherweise mehrere Prototypen entwerfen sollen, als den ersten direkt in die Demo-Version zu entwickeln. Auf der anderen wurde durch die Lösung mit dem Django Webtool eine großartige Basis geschaffen um beide Hauptfunktionen zu implementieren, dies war ein sehr glücklicher Umstand den wir genutzt haben, um noch eine Zusatz-Funktionalität, die Admin-Seite, bereit zu stellen.

2.2.4 Transition-Phase [A.G,R.K,R.M,J.F]

2.2.4.1 Zustand

Diese Phase behandelt die Iteration 9 und 10(Die 10 Iteration war eigentlich dafür angedacht ausschließlich den Bericht und andere Dokumente zu verfassen). Unser System lief ab dem 28.06 auf den StuRa Servern und wir konnten jetzt besser den je testen, zudem bot sich die Gelegenheit einer Art Live-Test im StuRa durchzuführen und unser System wenigstens noch eine Woche zur Verfügung zu stellen um Bugs zu fixen und letzte User-Experiences einzuholen und zu reagieren d.h noch kleine Änderungen vorzunehmen wo es machbar war um das Benutzer Erfahrung zu verbessern.

2.2.4.2 Ziele

Ziel	Datum	Erläuterung	erfüllt?
User-Test durchführen und Produkt daraufhin nochmal anpassen	28.06	Im StuRa getroffen und wenigstens Johann das ganze System aus Sicht des Hauptbenutzers mal testen lassen	theoretisch ja, praktisch hatten wir zu wenig Zeit um das ganze System dem gesamten StuRa vorzustellen und diese User-Experiences noch zu verarbeiten
Use-Cases die implementiert sind testen	bis zum 02.07	bekannte Bugs fixen	Bekannte Bugs wurden nicht alle gefixt

2.2.4.3 Probleme

Wir haben etwa anderthalb Wochen vor Projektabgabe mit dieser Phase begonnen und hatten nur wenig Zeit, da wir nebenbei noch einen Bericht schreiben mussten fielen die neuen Funktionen bzw. Verbesserungen etwas kurz aus.

2.2.4.4 Aktivitäten

Test und bugfixing

Das Testen wurde hintenangestellt, da die Implementierung große Schwierigkeiten hatte und somit lange nichts zum Testen vorhanden war. Zum Teil wurde das Testen der implementierten Funktionen bereits während der Implementierung übernommen, da bei einer Webanwendung mit Datenbankbindung beim Coden getestet werden muss, ob alles wie geplant funktioniert. Dies nahm natürlich den Testern Arbeit ab, diese aber auch eine längere Zeit lang in der Entwicklungsphase "überflüssig" macht. Da die lokale Instanz sich nicht oder nur mit erheblichem Aufwand kopiert ließ, blieb eine Testmöglichkeit für die Tester lange aus.

Django funktionierte auf den Computern der restlichen Gruppenmitglieder nicht richtig oder ließ sich garnicht erst starten. Trotz mehrfacher Installations- und Konfigurationsversuchen weigertete sich Django auf anderen Windows Installationen zu funktionieren. Desweiteren hatte ein Tester das Problem, dass seine Hardware die Virtualisierung im BIOS nicht unterstützte, was das Ausführen des Datenbankservers in der VM unmöglich machte. Damit war das Test Team auf den Server des Sturas angewiesen, welcher erst zum Ende des Projektes hin bereitgestellt wurde.

Nachdem uns der Server des Sturas zur Verfügung stand wurde direkt mit dem Testen der Software begonnen. Wir fokussierten uns darauf sicherzustellen, dass alle vollständig implementierten Use-Cases ohne Bugs oder sonstige Probleme später für den Anwender angenehm zu nutzen sind.

In der Entwicklung lag der Fokus auf der Implementierung der vom StuRa gewünschten Funktionen, weshalb diese für uns auch Priorität beim Testen hatten. Das Einreichen von Anträgen, sowie das Anzeigen und Bearbeiten von bereits eingereichten Anträgen musste unbedingt bis zur Auslieferung einwandfrei funktionieren, weshalb der Fokus sehr auf diesen drei use-cases lag.

Die Software wurde während der Entwicklung bereits sehr gut debuggt, weshalb sich beim Testen kaum größere Fehlfunktionen gefunden wurden, welche dem Team Implementierung nicht bereits

bekannt waren.

Aufgrund der knappen Zeit wurden die gefundenen Bugs und Probleme direkt in Anschluss mit dem Team Implementierung[Ruben und Richard] besprochen. Kleine "Schönheitsfehler" wie etwa Rechtschreibfehler oder unschöne Formatierung wurden direkt live geklärt und behoben.

Größere Fehler, wie die fehlende Übermittlung der Stelle des Antrags beim Abfragen der Daten, konnten nicht sofort, aber bis zum nächsten Tag behoben werden. Dies ermöglichte es in der knappen verbliebenen Zeit direkt weitere Test an der überarbeiteten Software durchzuführen und diese Bugfixes ausführlich zu prüfen und bei Bedarf erneut zu bugfixen.

Es konnten allerdings nicht alle gefundenen Probleme bis zur Auslieferung behoben werden. So ist es aktuell zwar möglich beim Bearbeiten eines Antrags die absolute Mehrheit auf "ja" zu setzen, allerdings wird diese Datenänderung dann nicht wie alle anderen Änderungen an die Datenbank übermittelt.

Trotzdem haben die Tests dafür gesorgt, dass die meisten Fehler und Probleme rechtzeitig entdeckt und behoben werden konnten. Dies hat es ermöglicht die Software rechtzeitig in einen auslieferungsbereiten Zustand zu bringen, sodass der StuRa bedenkenlos mit dem System arbeiten kann, wenn die Auslieferung abgeschlossen ist. Da Ruben selber im StuRa aktiv ist, kann die Software sogar nach der Auslieferung von einem Experten weiter betreut und gewartet werden. Selbst die Weiterentwicklung des Tools durch ein anderes Team kann so einfach unterstützt werden.

Das System hat in der Theorie (und auch in der Praxis) einige Schwachstellen, welche ein böswilliger Akteur/Angreifer ausnutzen könnte, um es anzugreifen und unbenutzbar zu machen. Diese Probleme sind uns bereits während der Planung des Projektes aufgefallen. Dazu gehören die Möglichkeit einer DOS Attacke auf den Server und das Einreichen unzähliger unechter Anträge, welche ebenfalls den Server überfordern könnten und die Datenbank mit unnötigen Daten zumüllen. Auch wenn der Server durch diese Anträge nicht überlastet wird, so wird trotzdem die Arbeit des StuRa behindert. Ebenfalls könnte auch ein SQL-Injection Angriff möglich sein, durch welchen sich (auch ohne Adminrechte) eventuell Daten aus der Datenbank löschen lassen könnten.

Da die Sicherung des Systems gegen solche Angriffe aber keine Anforderung seitens des Stura war, haben wir uns auch nicht wirklich mit der Lösung dieser Probleme beschäftigt. Es wären einige weitere Skripte und log-Dateien notwendig gewesen, um für einen soliden Schutz zu sorgen. Da diese nicht implementiert wurden haben wir im Test auch garnicht erst auf die besagten Schwachstellen getestet. Aus den gleichen Gründen haben wir auch bewusst den Test zur Reaktion der Software auf unerwartete Sonderzeicheneingaben nicht durchgeführt.

Übergabe und Dokumentation

Die geplante Übergabe findet am 7.7.2021 statt. Das Übergabeprotokoll wurde bereits erstellt und wird mit dem Kunden abgeglichen und anschließend unterzeichnet.

Zudem schreiben wir in der Halb-Iteration 10 diesen Bericht und aktualisieren die für die Bewertung ausschlaggebenden Dokumente. Dies nimmt einen erheblichen Teil unserer Zeit ein (Zusammengerechnet etwa 30-50 Stunden Arbeit)

2.2.4.5 Review

Die Zeit in dieser Iteration war sehr knapp, dennoch haben wir eigentlich jeden implementierten Use-Case ein Stück verbessert und aus Sicht des Nutzers sinnvolle Ergänzungen getätigt. Wir haben unserer Meinung nach noch das Beste aus den 2 Wochen vor Abgabe rausgeholt was im Rahmen dieses Projektes möglich war. Wir hätten uns dennoch einen Live-Test mit dem ganzen StuRa gewünscht nicht nur mit unserem Ansprechpartner

3. Ergebnisse [R.K,A.G]

3.1 Was waren unsere Ziele?

Am Anfang des Projektes haben wir unsere Ziele sehr hoch gesteckt und wollten, alle Use-Cases des StuRa umsetzen. Allerdings haben wir nur die Use-Cases für die herkömmliche Verwendung des Tools umsetzen können. Also Anträge stellen, Anträge bearbeiten und #die Möglichkeit bieten eigene Anträge und deren Status (noch in Arbeit, vertagt, beschlossen oder abgelehnte) einsehen zu können.

Außerdem haben wir versucht noch zusätzliche Anforderungen, in Form von Add-Ons aufzunehmen, ein Beispiel hierfür wäre die Idee der Automatischen generierung einer Tagesordnung oder einer verbesserten Druckvorlage für Anträge, praktisch haben wir uns schlussendlich erstmal auf die Hauptanwendung (Unsere Use-Cases) konzentriert.

```

@startuml "Use-Case-Model"
left to right direction
actor Antragssteller <<Person>>
actor Präsidium <<Person>>

rectangle Antragsverwaltungstool {
    (Wahlantrag stellen) <<UC01>> as wahl
    (Antrag auf Ressourcen stellen) <<UC02>> as ressourcen
    (Offene Anträge Ansehen) <<UC03>> as ansehen
    (Änderungsantrag vom Antragsteller) <<UC04>> as änderung
    (Offene Anträge ansehen) <<UC05>> as ansehen_p
    (Antrag bearbeiten) <<UC06>> as bearbeiten
    (Antrag ablehnen) <<UC07>> as ablehnen
    (Antrag vertagen) <<UC08>> as vertagen
}
left to right direction
Antragssteller --> wahl
Antragssteller --> ressourcen
Antragssteller --> ansehen
Antragssteller --> änderung

Präsidium --> ansehen_p
Präsidium --> bearbeiten
Präsidium --> ablehnen
Präsidium --> vertagen

@enduml

```

3.2 Was haben wir davon Geschafft?

Unser Tool bietet die Möglichkeit Anträge aller gewünschten Formen durch externe und interne Stellen einzureichen. Hier werden die Anträge mit ihren Pflicht Daten vom System verarbeitet und entsprechend nach ihren zuvor modellierten Attributen gespeichert. Diese Speicherung erfolgt permanent. Wir haben ebenfalls geschafft eine Antragsnummer basierend auf Jahr(der Legislaturperiode) und Nummer (der nächsten Plenumsitzung) zu generieren und diese als Antragsnummer abzubilden. Dabei haben wir nach Vorgabe des StuRa eine dynamische Antragsnummer für jeden Antrag generiert, dies wurde zudem ausgiebig getestet und funktioniert nach unserem Stand ohne Fehler.

Eine weitere implementierte Funktionalität des Tools, stellt die Bearbeitung der Anträge dar. Hier haben wir die Möglichkeit geschaffen, einzelne Anträge gezielt auszuwählen und dann durch einen autorisierten Nutzer bearbeiten zu lassen. Hier bekommt der StuRa ebenfalls die Möglichkeit die Beschlüsse des Antrags direkt einzutragen und falsche Angaben zu korrigieren.

Zusätzlich lässt sich vom Tool auch eine Liste anzeigen, diese Liste gibt entweder alle Anträge aus oder filtert die Anträge nach Stelle. Durch die Farbe des Hyperlinks kann man erkennen welche Anträge schon eingesehen wurden und welche noch nicht eingesehen wurden . Darauf aufbauend

lassen sich mit den Hyperlinks der Antragsnummer, die gestellten Anträge ebenfalls anzeigen bzw. führen zur Bearbeitung. Der nächste Schritt wäre hier den Status des Antrags anzeigen zu lassen dies ist jedoch nicht implementiert (Dies behandelt Use-Case 7 und 8).

Außerdem bietet das Tool die Möglichkeit eines Logins. Hier war unser Ziel für die Nutzer des "Präsidiums" Benutzer-Logins einzurichten, damit ausschließlich diese Zugang zu den Anträgen im Tool bekommen. Wir haben es geschafft das nur verifizierte Nutzer Anträge ansehen und bearbeiten dürfen. Auch durch Angabe von URLs zu Anträgen ist ohne Login keine Anzeige möglich.

Zusätzlich zu diesem Login gibt es eine Administrationsseite zum tieferen Verwalten der Anträge. Hier können die Datenstrukturen, über die normale Eingabemaske hinaus, ergänzt / geändert / entfernt werden.

Des weiteren haben wir es geschafft, das die ganze Seite einem gewissen Konzept folgt. Wir hatten das Ziel die Benutzung der Seite möglichst einfach und so modern wie möglich zu halten. Dabei stand stets der Nutzer im Vordergrund. Es sollte erreicht werden, dass obwohl bei den Anträgen sehr komplexe Gegebenheiten gelten, eine einfache Nutzung des Tools ohne recherche notwendig ist. #Dies haben wir durch erklärende Texte und Popups zumindest bei unserem kleinen User-Test erreicht.

Zusammengefasst haben wir die Use-Cases 01,02,05 sowie 06 implementiert und getestet. Diese werden innerhalb unseres Tools in der nächsten Woche 5-11.Juli offiziell übergeben, inoffiziell stehen diese auf dem StuRa-Server schon zur Verfügung.

3.3 Was haben wir nicht erreicht?

3.3.1 Störfaktoren

Leider konnten wir einige Funktionalitäten innerhalb des Projektes nicht umsetzen. Hier ist als großer Faktor der Zeitdruck zu sehen. Dadurch das wir relativ spät im Projekt auf ein Framework umgestiegen sind, mussten wir uns zuerst darin einarbeiten um die Funktionsweise zu verstehen. Da uns dies relativ viel Zeit kostete, waren wir am Anfang von SE II im Verzug. Da in diese Maßnahmen viel Energie geflossen ist, fehlte diese uns leider an anderen Stellen im Projekt bzw. bei der Bearbeitung genannter Funktionalitäten. Jedoch lässt sich aber sagen, dass wir hier im Hintergrund vielleicht mehr erreicht haben als wir eigentlich dachten.

Durch das Framework haben wir einen guten Startpunkt für die weitere Entwicklung geschaffen und weitere Funktionalitäten können ohne große Probleme hinzugefügt werden, das ist zwar das nicht mehr Teil unseres Projektes dennoch eine Art zusätzliche Anforderung welche wir in Kooperation mit unserem Kunden beschlossen haben, um die Weiterentwicklung zu ermöglichen und zu fördern.

Zudem sind die Funktionalitäten die wir umgesetzt haben, auch benutzerfreundlich umgesetzt und wurden von unserem Kunden positiv entgegen genommen bzw. gerne verwendet.

Ein anderer Störenderfaktor war unsere Kundenumgebung, zusammengefasst haben wir am Anfang des Semesters zum Teil mehrere Wochen auf Anfragen keine Rückmeldung bekommen, dies war störend jedoch nicht Projektgefährdend. Dazu kommt, dass es einen Wechsel von

Auftragsteller bzw. Kunden gab welche unterschiedlichen Prioritäten bzw. Ansichten zum Tool hatten, dies wäre sogar förderlich gewesen wenn genannter Wechsel nicht in der Mitte des Semesters passiert wäre.

Das allerdings größte Hindernis im Bezug auf unseren Kunden war, das wir Aufgrund von Misskommunikation und Wechsel im StuRa, erstmal keinen Server hatten und unsere Website selber vorerst auf einem lokal gehostetem Webserver implementieren und testen mussten, dies war möglich aber umständlich.

Wir hätten wahrscheinlich in der Hinsicht noch mehr auf unsren Kunden zugehen müssen haben, aber nach unserer Meinung eigentlich alles getan um die Kommunikation von unserer Seite aus aufrecht zu halten (E-Mail sowie Discord Nachrichten von mind. 2 Personen).

3.3.2 Was wurde nicht erreicht

Was haben wir also nicht geschafft ? Im Hinblick auf unsere Use-Cases haben wir leider einige Dinge nicht erreicht darunter zählen die Use-Cases 03,04,07 und 08. Die größte Funktionalität die nicht erreicht wurde, ist die Widerspiegelung des Status des Antrags. Da diese sehr komplex und vielseitig ist, haben wir uns dafür entschieden diese vorerst nicht zu implementieren. Leider haben wir bisher keine Möglichkeit gesehen diese im aktuellen Stand des Projektes bei der Antragsbearbeitung unterzubringen. In den Modellen existiert jedoch ein Attribut welches diese abbildet. Erschwerend kam hinzu das durch den Status des Antrags gewisse Abläufe abgebildet bzw. ausgedrückt werden sollten. Dadurch das wir hier nicht die Zeit hatten die Abläufe komplett zu verstehen und diese zu modellieren blieb dieses Feature leider außen vor.

Ebenfalls nicht unterstützt wird das Feature des Einsehens der offenen Anträge durch externe Nutzer. Hier war das Ziel für externe Nutzer einzusehen welche Anträge gerade in Bearbeitung sind und welche bereits abgeschlossen. Dadurch das wir die oben genannte Status anzeige der Anträge ebenfalls nicht abbilden konnten mussten wir dieses Feature ebenfalls weglassen. Erschwerend kam hinzu, das hier bei uns Bedenken im Thema Datenschutz auftraten. Da hier personenebezogene Daten im Antrag abgelegt werden (Name, Email, Beantwortung sensibler Fragen) wäre nur ein Ansehen eines einzelnen, selbst gestellten Antrages, sinnvoll. Uns ist in diesem Fall keine gute Lösung in der Projektzeiteingefallen, quasi kurzzeitige Links zum ansehen von Anträgen zu generieren.

Ein Feature das noch verbesserungswürdig ist, ist das ansehen von Anträgen im "Intern" Bereich. Hier kann man die Anträge nur ansehen indem man in den Bearbeitungsmodus des Tools wechselt. Man zeigt einen Antrag also an, indem man keine Änderungen im Antrag durchführt. Hier gibt es sicher eine bessere Lösung die Anträge als nicht bearbeitbar anzuzeigen.

Auch das Ablehnen und Vertagen von Anträgen mussten wir weglassen. Besonders beim Vertagen des Antrags spielen wieder die Status der Anträge eine Rolle, welcher nicht implementiert sind. Beim Vertagen wird der Antrag auf eine nächste Sitzung verschoben, was genau die abzubildenden Abläufe im StuRa aufzeigt. Hier hätten wir 1. das Feature des Antraggssatus gebraucht und 2. hätte es durch das Ablehnen einer Antragsnummer Änderung bedarft bzw. hätte man hier auch die Tagesordnung der jeweiligen Sitzung beachten müssen, welche als Add-On ebenfalls nicht umgesetzt wurde.

3.4 Reflexionen der Teammitglieder

Jacob Flaskamp [J.F.], Ruben-David Kraus[R.K.], Richard Müller[R.M.], Duy Tien Nguyen[T.N], Andreas Gröber[A.G]

3.4.1 Jacob Flaskamp [J.F.]

Was habe ich gelernt?

Ich habe gelernt, was so alles hinter der Entwicklung von komplexen Software Systemen steht und wie viele verschiedene Aufgabenbereiche es bei solch einem Projekt eigentlich gibt. Ohne ein gut organisiertes Team mit einem klaren Plan würde man nie in absehbarer Zeit zu einem guten Ergebnis kommen können. Selbst ein anfänglich leicht erscheinendes Projekt wird später mit hoher Wahrscheinlichkeit zu einer Herausforderung werden. Es können jederzeit unvorhergesehene Probleme oder Hürden auftreten, weshalb man im Idealfall auch auf diese vorbereitet sein sollte (siehe Risklist). Ein Beispiel dafür ist die starke Verzögerung bei der Bereitstellung des Servers für unser Tool. Auf dieses Problem waren wir leider nicht so gut vorbereitet.

Vor allem ist auch ein gut geschriebener Projektplan mit konkreten Anforderungen notwendig. Unser Team hat z.B. den Fehler gemacht die "Anforderung" Tagesordnung nicht gut genug mit dem Themensteller (StuRa) abzusprechen, weshalb wir unnötig Zeit in Planung und Entwurf investiert haben, obwohl diese Funktion später wieder verworfen werden musste.

Worauf bin ich stolz?

Auf mein Team und das Ergebnis, das wir erreicht haben. Trotz einiger unvorhergesehener Hindernisse haben wir meiner Meinung nach viel erreicht und eine solide Basis geschaffen, auf welcher nächstes Semester auch noch ein weiteres Team aufbauen kann. Auch wenn wir nicht alle ursprünglich geplanten use-cases realisieren konnten, so ist doch ein sehr nützliches Tool entstanden, dass dem StuRa in Zukunft viel Arbeit ersparen/erleichtern wird.

Was hat gut funktioniert?

Die regelmäßigen Treffen zu Besprechungen fanden immer statt. Fast nie hat ein Teammitglied gefehlt (und falls doch hat derjenige bereits vorher Bescheid gegeben). Dies hat dafür gesorgt, dass neue Aufgaben und Probleme schnell und rechtzeitig geklärt werden konnten. Außerdem wussten so auch immer alle Teammitglieder, welchen tasks sie aktuell nachgehen mussten. Bei zusätzlichem Bedarf für Absprachen konnten immer schnell und flexibel weitere Besprechungstermine vereinbart werden. Alle Teammitglieder haben sich super verstanden und es kam nie zu Konflikten untereinander. Jeder hat seine Aufgaben wichtig genommen und sich nicht auf der Arbeit der Anderen ausgeruht, wodurch wir auch immer gut in unserem Zeitplan bleiben konnten.

Was würde ich beim nächsten Projekt anders machen?

Ich hätte gern auch noch die anderen Rollen besser kennengelernt. Vor allem bei der Implementierung würde ich beim nächsten Mal gern mitwirken und tiefere Einblicke gewinnen. Außerdem würde ich mir noch mehr Mühe geben rechtzeitig für eine Testumgebung zu sorgen, da wir leider durch die starke Verzögerung bei der Serverbereitstellung viel Zeit zum Testen verloren haben.

Selten ist es vorgekommen, dass eine Aufgabe kurzzeitig unter den Tisch gefallen ist, da jeder dachte, dass ein Anderer bearbeitet. Im genauen Gegensatz dazu ist es mir auch passiert, dass ich und ein weiteres Teammitglied beide die Anwendungsdokumentation geschrieben haben. Daraufhin musste ich meine Arbeit verwerfen, was ich sehr ärgerlich fand. Um solche Missverständnisse zu vermeiden würde ich beim nächsten Projekt gern noch genauer mit GitHub issues oder etwas ähnlichem arbeiten.

3.4.2 Ruben-David Kraus [R.K.]

Was habe ich gelernt ?

Im Projekt habe ich zahlreiche Erfahrungen gesammelt. Einerseits dass die Kommunikation im Team bei solch einem Vorgehen die wichtigste Komponente von allen ist. Kommunikation bildet die Grundlage für ein zielgerichtetes Arbeiten. Wird nicht ausreichend kommuniziert, wird aneinander vorbeigearbeitet. Oft werden Aufgaben nicht klar verteilt, deswegen sollte alles aktiv besprochen und abgeklärt werden. Auch in meiner Rolle, der Implementierung ist es essenziell miteinander zu arbeiten und nicht jeder für sich. Das hat sich im Laufe des Projekts deutlich herausgestellt. Wird in gemeinsamen 'Sessions' programmiert, ist der Output deutlich besser, da man sich direkt absprechen kann.

Auch der Umfang den solch ein Projekt schnell erreichen kann, auch wenn man denkt "ach das geht schon", ist doch sehr beachtlich. Ich denke unserem gesamten Team ist klar geworden wie wichtig es ist Dinge bis in relativ hohe Detailgrade zu planen, damit Informationen nicht verloren gehen oder man ein gutes Grundgerüst hat, mit der "eigentlichen" Arbeit des Implementierens zu beginnen. Zusätzlich hat man gemerkt, dass das eigentlich Entwickeln der Software einen gar nicht so großen Teil des Prozesses ausmacht als man eigentlich dachte. Ich finde es gut dass hier natürlich alle Prozesse der Softwareentwicklung ansprechend beleuchtet wurden.

Auch die Wichtigkeit von Schaubildern und Diagrammen ist mir deutlicher geworden. Es ist deutlich einfacher, beispielsweise eine Domäne zu verstehen bzw. gemeinsam über ein Thema zu diskutieren wenn jeder, unabhängig vom Fachwissen, das Thema in kurzer Zeit überblicken und 'verstehen' kann.

Ich habe auch mitgenommen, dass die Entwicklung von Demos oder ein "minimum viable product" Ansatz eine gute Möglichkeit ist früh Feedback einzuholen, damit nicht in Richtung Entwickelt wird, die am Ende falsch sind.

Außerdem habe ich gelernt, dass es wichtig ist nicht alles selbst machen zu wollen. Wenn man versucht alle Schnittstellen für komplexe Anwendungen zu programmieren, kommt man schnell an seine Grenzen und man merkt dass man mit seinem aktuellen Wissen nicht weiterkommt. Daher habe ich auch gelernt, wie wichtig Frameworks sein können und dass sie einem zahlreiche sowie gute Ansätze bieten auf lästige Arbeit zu verzichten.

Darauf aufbauend habe ich ebenfalls viele Erkenntnisse, bezüglich Model-View-Controller Architekturen gewonnen. Ich habe an vielen Stellen gemerkt wie sinnvoll es ist Funktionalitäten einzeln abzugrenzen. Hier hatte ich oft einen 'AHA!'-Moment indem ich die Sinnhaftigkeit dieses Architekturmusters verstanden habe. Hier konnte ich mir oft langen und komplex zu lesenden Code ersparen.

Auch wie wichtig REST-API in der aktuellen Zeit ist. Es war spannend zu sehen wie einzelne Requests und Responses über den Server liefen und die Kommunikation mit dem Client zu beobachten. Damit einhergehend habe ich ebenfalls gelernt einen Webserver einzurichten und eine Anwendung in eine Produktivumgebung einzuführen und zu betreiben.

Was hat gut funktioniert ?

In unserem Projekt hat vieles gut funktioniert. Einerseits war die Kommunikation mit anderen Teammitgliedern stets freundlich und auf Augenhöhe. Es kam nie zu Streitigkeiten oder Unsachlichkeit in den Diskussionen.

Ebenfalls war die Arbeit in der Implementierung sehr angenehm. Wir haben oft direkt miteinander geredet und "Pair-Programming" betrieben, was das Arbeiten deutlich angenehmer gemacht hat, als Solo for sich hinzucoden. Wir haben stets lösungsorientiert gearbeitet und dem Anderen nie Vorwürfe gemacht wenn mal etwas nicht funktionierte. Ich hatte das Gefühl das wir immer an einem Strang gezogen haben und gemeinsam daran orientiert waren das Problem zu lösen.

Was würde ich beim nächsten Projekt anders machen ?

Leider gab es auch ein paar Punkte die mir zu denken gegeben haben. Einerseits ist der StuRa als Auftraggeber sehr gut, alle sind stets nett und bemüht und man merkt das Sie auch an einer guten Umsetzung interessiert sind. Da der StuRa aber leider auf freiwilliger Arbeit basiert und es zur Zeit unseres Projektes einen größeren Mitglieder Abgang gab, fehlt es an dieser Stelle an nötigen Personen die uns betreuten. Am Ende war nur noch ein Student für uns zuständig, der auch selbst inmitten der Prüfungsphase für unser Projekt nur wenig Zeit aufwenden konnte. Da dieser Fakt natürlich schwer abzuschätzen ist, würde ich eventuell vorher beim StuRa anfragen ob genug Kapazitäten für die Betreuung vorhanden sind.

Die Art des Themas hat mir im Nachhinein ebenfalls zu denken gegeben. Ich hatte das Gefühl, dass das komplette von Grund auf Entwickeln, von Software eine große Herausforderung darstellt. Wenn es hier eher wenig, z.B. berufliche Vorkenntnisse gibt, ist es sehr schwer sich in alle Themengebiete einzuarbeiten und einem fehlt die nötige Informationsdichte um für Entscheidungen genug Möglichkeiten zu haben. Hier ist es mir teilweise schwer gefallen den Überblick zu behalten, was nun wo gut angewendet werden kann. Wir haben aus Unwissenheit lange in eine Richtung gearbeitet, die uns am Ende keinen Erfolg brachte. Daher würde ich mich im Vorhinein mehr in Richtung Umsetzung (Frameworks, Programmiersprachen, State-of-the-Art) informieren um genug planen zu können.

Außerdem würde ich beim nächsten Projekt auch gerne einmal SCRUM anwenden bzw. noch mehr mit meinen Teammitgliedern kommunizieren und gemeinsam zu arbeiten. Ich denke es fällt oft schwer im Druck des Semesters genug Motivation zu finden, viel Arbeit ins Projekt zu investieren. Ich habe gemerkt das es viel mehr Spaß macht simultan zusammen zu arbeiten und das möchte ich gerne intensivieren.

3.4.3 Richard Müller [R.M.]

Im Sommersemester 2021 starteten wir in die zweite Runde des Moduls „Software Engineering“ bei Herrn Prof. Anke. Auch wir wurden von der Verkleinerung der Gruppe nicht verschont, da dieses Semester die Mitglieder des Studiengangs „Wirtschaftsingenieurwesen“ wie geplant ausschieden.

Somit wurde unsere Gruppenstärke von acht auf fünf reduziert was dazu führte das einige Rollen doppelt besetzt wurden.

Das erste begutachten unseres Beleges des letzten Semesters machte uns klar das hier ganz klar Luft nach oben ist. Nach der Konsultation mit Herrn Prof. Anke bezüglich unseres Beleges machte sich das Team daran die Tipps umzusetzen und in zwei kleineren Grüppchen vorerst weiter zu machen. Die eine Gruppe kümmerte sich um die 3 Wichtigsten Use Cases und überarbeitete diese während das andere Team damit begann den Beleg des vorherigen Semesters zu überarbeiten. Als Basis dienten die Vorschläge und Tipps von Herrn Prof. Anke die in das Abgabedokument eingepflegt wurden. Da ich die Rollen Architekt und Entwickler zusammen mit Herrn Kraus innehatte, fokussierten wir uns auf die Use Cases und wie man diese am besten Implementieren könnte. Wir einigten uns also darauf eine Webanwendung mit Datenbank zu bauen, da allerdings nur Ich über die nötigen Kenntnisse der Webentwicklung (HTML, CSS, Java Script) verfügte habe Ich für meine Gruppenmitglieder einen kleinen Workshop „How to Website“ abgehalten in denen Ich die Basics erklärt habe. Für die Finale Implementierung haben Wir uns entschlossen das Ich die Webanwendung bauen werde und Herr Kraus sich um die Datenbank und die Bereitstellung des Servers kümmert. Das hat auch gut funktioniert dennoch hat der Kundenkontakt in diesem Semester nicht so wirklich geklappt. Da wir den StuRa als Kunden hatten und dieser gerade ein paar Personalprobleme hatte, hatten wir das Nachsehen, da sich erstmal keiner wirklich zuständig gefühlt hat und wir quasi „herumgereicht“ wurden. Dies führte dazu das wir unseren Server erst Mitte Juni 2021 bekommen haben was sich massiv auf die Testzeiträume auswirkte da nun die Feinabstimmung der einzelnen Softwareprojekte folgte. Wir wollten ohne Framework arbeiten allerdings erlaubte das die Webanwendung nicht da die Connection mit der Datenbank nicht klappte. Da die Zeit schon recht knapp war und wir keine Zeit für langes recherchieren hatten, entschieden wir uns die Webanwendung auf das Django Framework umzubauen diese Herausforderung meisterten wir jedoch sehr schnell dadurch konnten wir nun endlich mit testen anfangen und bugfixing betreiben was unserem Zeitplan sehr zugute kam.

Rückwirkend betrachtet gab es ein paar Stellen die nicht wie erwünscht gelaufen sind und leider Zeit in Anspruch genommen hat die nicht eingeplant war, genauso wie es durch das Fehlen der Hardware trotz mehrfacher Nachfrage zu unnötigen Verzögerungen kam was unseren Zeitplan immer wieder durcheinander gebracht hat.

Alles in allem war es eine Herausragende Erfahrung dieses Softwareprojekt zu betreuen. Es hat sehr großen Spaß gemacht und die Gruppe hat trotz des mangelnden Kundenkontakts am Anfang des Semesters nie aufgegeben und das Beste aus der Situation gemacht! Dies hat dazu geführt das nun am Ende dieser wilden Berg- und Talfahrt der Softwareentwicklung, nahezu alle Use-Cases implementiert und getestet werden konnten.

3.4.4 Duy Tien Nguyen[T.N]

- Was habe ich gelernt?

Erstens habe ich gelernt, wie man mit anderen Teammitgliedern schnell kommunizieren kann. Wir haben eine Whatsapp-Gruppe und eine Discord-Gruppe erstellt, mit denen wir Nachrichten miteinander schicken und Meetings organisiert haben. Zweitens habe ich gelernt, wie man im Team die Aufgaben teilt. Jeder kann etwas gut. Man muss wissen, was er gut kann und was ihm gefällt, um diese Aufgabe gut zu erledigen. Außerdem habe ich gelernt, dass ich manchmal um Hilfe von den anderen bitten kann, dass sich Mitglieder gegenseitig unterstützen sollen.

- Worauf bin ich stolz?

Ich bin darauf stolz, dass ich die Aufgaben meiner Rolle erledigen konnte. Am Anfang war es schwer, weil ich die Aufgaben nicht so gut kannte. Aber mithilfe der anderen und der Dokumentationen konnte ich sie erledigen.

- Was hat gut funktioniert?

Die Zusammenarbeit zwischen den Teammitgliedern hat gut funktioniert. Obwohl wir uns nicht im Präsenz treffen konnten, haben wir uns regelmäßig getroffen und über die Probleme diskutiert. Die Aufteilung der Aufgaben hat auch gut funktioniert, weil jeder die Aufgaben wählen konnte, die ihn interessiert.

- Was würde ich beim nächsten Projekt anders machen?

Ich würde mich auf die Technologie, die bei der Implementierung genutzt wird, vorbereiten. In diesem Projekt haben wir HTML, CSS und JavaScript genutzt. Ich habe diese Sprachen nicht so gut gekannt, deswegen konnte ich bei der Implementierung nicht viel helfen.

3.4.5 Andreas Gröber[A.G]

Was habe ich gelernt?

Worauf bin ich stolz?

Was hat gut funktioniert?

Was würde ich beim nächsten Projekt anders machen?