

Reflexionsdokument - das Team

Tatsächlicher Zeitplan

Parent	Key	Issue Type	Summary	Sprint	Story point estimate	Story point actual	Dev
None	SA1-67	Task	Datenbank Exemplarisch Befüllen	SA1 Sprint 4	1	0,5	0,5
	SA1-66	Task	Django Admin Aufräumen	SA1 Sprint 4	2	0,5	1,5
	SA1-64	Task	Farben anpassen	SA1 Sprint 4	1	2	-1
	SA1-63	Task	Landing Page	SA1 Sprint 4	2	1	1
	SA1-62	Task	UpdateView Bewerber Persönliche Daten	SA1 Sprint 3	1	0,25	0,75
	SA1-61	Task	Passwort change of User Backend + Frontend	SA1 Sprint 4	2	1	1
	SA1-60	Task	Bewerbung darf nicht gelöscht werden nachdem sie angenommen wurde	SA1 Sprint 3	1	1	0
	SA1-59	Task	Frontend Finalisierung Bewerber	SA1 Sprint 3	2	3	-1
	SA1-58	Task	Frontend Finalisierung Zulassungsausschuss	SA1 Sprint 3	2	4	-2
	SA1-51	Task	Protected Routes die Verhindern das nicht eingeloggte oder falsche Nutzergruppen zugriff haben	SA1 Sprint 2	1	0,25	0,75
						None Ergebnis	1,5
SA1-10	SA1-37	Subtask	DeleteView erstellen	SA1 Sprint 2		0,125	
	SA1-29	Subtask	UpdateView erstellen	SA1 Sprint 2		0,125	
	SA1-28	Subtask	CreateView erstellen	SA1 Sprint 2		0,125	
	SA1-27	Subtask	Model Erstellen	SA1 Sprint 2		0,125	
	SA1-10	Story	Ich als Bewerber möchte Daten in eine Maske eingeben um bewerbungsrelevante Informationen bereitzustellen/upzudaten.	SA1 Sprint 2	2	0,5	1,5
						SA1-10 Ergebnis	1,5
SA1-11	SA1-55	Subtask	Detaillansicht Arbeitserfahrungen	SA1 Sprint 2		0,25	
	SA1-54	Subtask	Detaillansicht Schulabschlüsse	SA1 Sprint 2		0,25	
	SA1-53	Subtask	Detaillansicht Universitätsabschlüsse	SA1 Sprint 2		0,25	
	SA1-52	Subtask	Übersicht Bewerbungsinformationen	SA1 Sprint 2		0,25	
	SA1-39	Subtask	Detaillansicht Bewerbung	SA1 Sprint 2		1	
		Story	Ich als Bewerber möchte eine Übersicht über meine Bewerbung haben, um meine Daten zu sehen und bearbeiten zu können.	SA1 Sprint 2	2	2	0
						SA1-11 Ergebnis	0
SA1-12	SA1-56	Subtask	Filterung der Tabellarischen Ansicht	SA1 Sprint 2		0,125	
	SA1-47	Subtask	Zulassungstatus in Listenansicht Bewerbung	SA1 Sprint 2		0,125	
	SA1-42	Subtask	Zulassungstatus in Detaillansicht Bewerbung	SA1 Sprint 2		0,125	
	SA1-41	Subtask	Detaillansicht Bewerbung mit Button um Bewerber zuzulassen	SA1 Sprint 2		1,125	
	SA1-40	Subtask	Tabellarische Ansicht Bewerbungen	SA1 Sprint 2		1,5	
	SA1-12	Story	Ich als Zulassungsausschuss möchte eine Übersicht über alle Bewerbungen haben	SA1 Sprint 2	5	3	2
						SA1-12 Ergebnis	2
SA1-13	SA1-13	Story	Ich als Bewerber möchte ein Feedback zu meiner Bewerbung angezeigt bekommen	SA1 Sprint 3	2	1	1
						SA1-13 Ergebnis	1
SA1-16	SA1-35	Task	Vorlage auswählen & anpassen	SA1 Sprint 1	2	1	1
	SA1-34	Task	Datenfelder definieren	SA1 Sprint 1	1	1	0
	SA1-19	Task	Git Build aufsetzen / Qualitätsicherung	SA1 Sprint 1	1	1	0
	SA1-18	Task	Git Workflow abklären	SA1 Sprint 1	1	0,5	0,5
	SA1-17	Task	Repository Initialisierung	SA1 Sprint 1	1	0,5	0,5
		Epic	Vorbereitung/Initialisierung	SA1 Sprint 2			
						SA1-16 Ergebnis	2
SA1-3	SA1-46	Task	Modul Test	SA1 Sprint 4	2	1,5	0,5
	SA1-45	Task	Stress Test	SA1 Sprint 4	2	1	1
	SA1-36	Task	Heroku Deployment	SA1 Sprint 3	2	2,5	-0,5
		Epic	Orchestrirung Bug Fixing	SA1 Sprint 4			
						SA1-3 Ergebnis	1
SA1-4	SA1-50	Subtask	Login View UX-Design	SA1 Sprint 1		0,25	
	SA1-21	Subtask	Login View erstellen	SA1 Sprint 1		0,125	
	SA1-4	Story	Ich als Bewerber möchte Daten in eine Maske eingeben um mich zu verifizieren.	SA1 Sprint 1	1	0,375	0,625
						SA1-4 Ergebnis	0,625
SA1-43	SA1-65	Subtask	Zulassungsmöglichkeit blockieren wenn 200 Zulassungen erreicht sind	SA1 Sprint 3		0,125	
	SA1-44	Subtask	Anzeige über Anzahl zugelassener Bewerber & wenn bei 200 keine Zulassungen mehr möglich	SA1 Sprint 3		0,125	
	SA1-43	Story	Ich als Zulassungsausschuss möchte sehen wie viele Bewerber ich noch zulassen kann	SA1 Sprint 3	1	0,25	0,75
						SA1-43 Ergebnis	0,75
SA1-5	SA1-49	Subtask	Bewerber Registrierung UX Design	SA1 Sprint 1		0,125	
	SA1-24	Subtask	Bewerber Modell erstellen	SA1 Sprint 1		0,125	
	SA1-23	Subtask	User Modell von Django überschreiben: ist_bewerber Komponente hinzufügen	SA1 Sprint 1		0,125	
	SA1-5	Story	Ich als Bewerber möchte Daten in eine Maske eingeben um mich zu registrieren.	SA1 Sprint 1	1	0,375	0,625
						SA1-5 Ergebnis	0,625
SA1-6	SA1-22	Subtask	Login View erstellen	SA1 Sprint 1		0,125	
	SA1-6	Story	Ich als Zulassungsausschuss möchte Daten in eine Maske eingeben um mich zu verifizieren.	SA1 Sprint 1	1	0,125	0,875
						SA1-6 Ergebnis	0,875
SA1-8	SA1-57	Subtask	Empfehlungsschreiben Fronted	SA1 Sprint 3	1	1	
	SA1-38	Subtask	Empfehlungsschreiben Model	SA1 Sprint 3		0,125	
	SA1-33	Subtask	CreateView: Name des Empfehlenden und Empfehlungsschreiben in Textfeld	SA1 Sprint 3		0,125	
	SA1-8	Story	Ich als Dritter möchte Text in eine Maske eingeben um ein Empfehlungsschreiben für einen Bewerber zu hinterlegen.	SA1 Sprint 3	4	1,25	2,75
						SA1-8 Ergebnis	2,75
SA1-9	SA1-32	Subtask	UpdateView: Nur Zulassung ist auswählbar	SA1 Sprint 2		0,125	
	SA1-31	Subtask	Detaillview: Einzelner Bewerber	SA1 Sprint 2		0,125	
	SA1-30	Subtask	ListView: Bewerber Anzeigen	SA1 Sprint 2		0,125	
	SA1-9	Story	Ich als Zulassungsausschuss möchte Bewerber auswählen können um diese zuzulassen.	SA1 Sprint 2	3	0,375	2,625
						SA1-9 Ergebnis	2,625
						Gesamtergebnis	14,625

Abbildung 1 Zeitplan

Zeitplanabweichungen

Zum Großteil wurde der geplante Zeitplan eingehalten. Die Sprints wurden wie geplant durchgeführt und zum festgelegten Datum fertiggestellt. Lediglich bei einzelnen Aufgaben wurde zu viel eingeplant, was auch in Abbildung 1 zu erkennen ist. Abweichungen auf User Story- und Taskebene, sowie die genauen zeitlichen Unterschiede, werden im tatsächlichen Zeitplan dargestellt. Die geplanten Bi-Daily Meetings wurden bis zum 23.11.2020 wie geplant durchgeführt.

Es gab lediglich zwei Aufgaben, die im anfangs geplanten Zeitplan nicht berücksichtigt wurden und somit zusätzliche Zeit in Anspruch genommen haben.

Eine geplante Einführung in den Framework Django, die an der Kick-Off Veranstaltung am 14.11.2020 angesetzt wurde, wurde im ersten Zeitplan nicht miteingeplant. Da sich dieser Crashkurs als sehr zeitintensiv herausstellte, wurden mehr Arbeitsstunden im ersten Sprint benötigt, als geplant, um nicht in Verzug zu kommen.

Ebenso wurde das Deployment zunächst nicht als zu erledigende Aufgabe definiert. Diese kam ebenfalls erst im weiteren Verlauf des Projekts hinzu und wurde am 18.11.2020 durchgeführt.

Bis auf diese Abweichungen wurde der erste Zeitplan wie geplant umgesetzt und alle Sprints wurden bis zum definierten Fälligkeitstermin erledigt. Alle Deadlines wurden eingehalten, da der Aufwand der Arbeitspakete richtig geplant wurde. Die von uns definierten Zeitangaben zu den jeweiligen Storypoints haben sich als zu groß herausgestellt und sollten in folgenden Projekten kleiner gehalten werden.

Prozess

Unser geplanter Prozess hat sich als effektiv herausgestellt und hat uns zum gewünschten Ergebnis geführt. Dies lässt sich an einigen positiven Beispielen erläutern, dennoch haben sich auch Verbesserungspotentiale herausgestellt.

Die geplanten Meetings waren für den Prozess sehr nützlich, es konnten immer alle Anliegen während dieser Besprechungen geklärt werden und es waren sehr wenig weitere Kommunikation über andere Kanäle (WhatsApp, E-Mail usw.) nötig. Als zusätzlicher Kommunikationskanal, welcher zunächst nicht berücksichtigt wurde, kamen die Issues in GitHub zum Einsatz. Commit Messages und Branches wurden wie geplant einheitlich benannt und dadurch konnte alles von jedem Teammitglied nachvollzogen werden. Eine zusätzliche Einheit im Prozess, welche zunächst nicht von uns eingeplant war, war der Einsatz von Breakout-Sessions in Zoom. Hier konnten kleine Teams für jeweilige Aufgaben, während unserer Meetings gegründet werden. Zoom wurde ebenfalls benutzt, um alle 2 Tage zu mergen und um bei merge Konflikten, diese mithilfe von pair programming zu lösen. Bei diesem Zusammenführen der einzelnen Komponenten hat sich unsere geplante kontinuierliche Integration bewährt. Mithilfe von Jira wurde der aktuelle Prozess und auch der Prozessstatus konstant für alle Mitglieder ersichtlich und es konnte eine konstante Aufgabenerledigung gewährleistet werden. Unsere branch commits wurden ebenfalls mit unseren Issues und Stories in Jira verknüpft, um möglichst viel Transparenz zu schaffen. Die PEP8-Checks ermöglichten die von uns gewünschte Gleichmäßigkeit im Prozess.

Im Gegensatz zu diesen positiven Erfahrungen in unserem Prozess, gab es auch Punkte, die für weitere Projekte verbessert werden können. Scrum konnte von uns nicht wie gewünscht eingesetzt werden, da die Projektzeit relativ kurz war und dadurch wichtige Elemente wie die Retrospektive sehr grob durchgeführt wurden. Unsere geplanten Reviews der Pull-Requests von zwei Personen wurde ebenfalls nicht so akribisch wie geplant durchgeführt. Das lag daran, dass der vorgestellte Code von der jeweiligen Person dem Team vorgestellt wurde und der Pull-Request anschließend nur von einer Person reviewed wurde. Aufgrund der Teamgröße und der unterschiedlichen Wissensstände, war eine gleichmäßige Einbindung in den Prozess und beispielsweise den Programmierteil schwierig. Diese Problem kann in weiteren Projekten gelöst werden, indem die Aufgaben in noch kleinere Teilaufgaben unterteilt werden und somit jedem Mitglied viele kleine Aufgaben zugewiesen werden können. Daraus folgt ein weiterer Schwachpunkt in unserem Prozess, die zu grobe und ungranulare Planung. Das UI-Design wurde zum Beispiel als eine Task geplant, was sich im Prozessverlauf aber als zu grob herausstellte. Dies könnte durch eine Trennung von Styling und Layout gelöst werden, um mehrere Teilaufgaben zu schaffen und dadurch mehrere Personen einzubinden. Das parallele Arbeiten hat in unserem Prozess nicht so gut funktioniert, wie es geplant wurde, die Abhängigkeiten einzelner Aufgaben waren nicht so granular. Ein Beispiel war die Trennung von Zulassungsausschuss- und Bewerbungsbewertungsaufgaben, da Ansichten und Modell auf andere Ansichten und Modelle aufgebaut haben, konnte teilweise nicht parallel gearbeitet werden. Die Prozessplanung war in manchen Punkten zu grob, da es schwer war bei diesem Projekt die perfekte Balance zu finden.

Erfahrungen im Team

Die Zusammenarbeit in unserem Team war sehr positiv, dies lässt sich an einigen Beispielen aufzeigen.

Die Organisation der Meetings hat durchgehend funktioniert, es wurden stets Termine gefunden, an denen es jedem Teammitglied möglich war, teilzunehmen. Durch die ständige Kommunikation, sowie die Meetings alle 2 Tage, war ein konstanter Austausch möglich. Ebenso war die Hilfsbereitschaft und die Hilfe bei gegenseitigen Fragen stets gegeben, wie zum Beispiel bei der Einführung in Django, um alle Teammitglieder auf den gleichen Stand zu bringen. Abstimmungen im Team, wie beispielsweise die Vergabe der Rollen, konnten ebenfalls reibungslos durchgeführt werden.

Die von uns eingesetzte Arbeitsweise während den Meetings, hat sich ebenfalls bewährt. Nach der gemeinsamen Besprechung der aktuellen Stände, wurde von den Teammitgliedern an den jeweiligen Aufgaben weitergearbeitet, um diese Neuerungen in einem anschließenden Gespräch wiederum im Team zu teilen.

Der Rollenverteilung im Team, ist jeder gerecht geworden und hat sich um seine Zuständigkeiten gekümmert. Dadurch, dass wie sehr breit aufgestellt waren und die Aufgaben verteilt wurden, gab es keine Konflikte hierbei. Ein regelmäßiges konstruktives Feedback des Teams an einzelne Mitglieder und die erledigten Aufgaben, hat die Zusammenarbeit weiter gestärkt.

Dennoch gab es auch Herausforderungen und Verbesserungspotentiale für zukünftige Projekte. Da das Team nach den jeweiligen Aufgaben geteilt wurde, konnte nur bis zu einem gewissen Grad parallel und unabhängig voneinander gearbeitet werden. Grund hierfür war der geringe Umfang und die geringe Komplexität. Dieses parallele Arbeiten hat dazu geführt, dass teilweise nicht klar war, welche Aufgaben bereits erledigt wurden und zusätzliche Kommunikation entstanden ist. Da nun die Arbeitsweise im Team und der einzelnen Teammitglieder bekannt ist, könnte dieses Problem in zukünftigen Projekten vorgebeugt werden. Ebenso mindern festgelegte Standards diesen Umstand, zum Beispiel sollte eine Aufgabe nach Erledigung direkt von *in progress* zu *in review* gezogen werden, um Verzögerungen zu verhindern.

Die schriftliche Kommunikation, sowie die Dokumentation einzelner Schritte war ebenfalls nicht so ausführlich, wie geplant. Ein Beispiel hierfür waren die Pull-Requests, es gab ein Template zum Ausfüllen, um mehr Transparenz zu schaffen und um Probleme vorzubeugen. Diese Template wurde meist aus Zeitgründen nicht oder nur unvollständig genutzt.

Eine weitere Herausforderung war die vollständige Bearbeitung im Team auf digitalen Weg. Wie sich aber im Laufe des Projekts herausstellte, war diese digitale Zusammenarbeit sehr positiv, da zum Beispiel das Teilen eines Bildschirms eine sehr nützliche Funktion für die Teamarbeit ist. Eine weitere Herausforderung dieser digitalen Bearbeitung war das Team Building und das Kennenlernen, ohne persönlichen Kontakt. Durch die vielen gemeinsamen Meetings und Besprechungen, hat unser Team diese Hürde schnell überwunden.

09.11.2020 First Meeting

Date

09.11.2020

Duration: 19:00 - 22:00

Participants

- @ Lukas Humpe
- @ Mostapha Ahaduch
- @ Alexander Spindeler
- @ Max Hammer
- @ Enrico Piscopello

Topics

- Kennenlernen
- Framework
- Meeting Planung
- Rollenzuweisung
- Planung Vorgehensweise

Decisions



Für Django als Framework entschieden



Bi-Daily vom 16. und Pair-Programming Tage am Samstag 21.11 Final Version ist geplant für den 23.11



Umsetzung der Anforderungen des ersten Zeitplans:

1. Aufgaben & Meilensteine
 - > Userstories aus Anforderungen der Aufgabe ausarbeiten
2. Ergebnisse für alle Meilensteine
 - > Bewerten der Ergebnisse in der Retrospektive des Meilensteins
3. Geschätzter Aufwand für alle Aufgaben
 - > Abschätzung des Arbeitsaufwands in XS, S, M, L, XL
4. Abhängigkeiten zwischen Aufgaben
 - > Kommentare in Userstories
5. Aufgabenzuweisung
 - > Rollen identifizieren und vor jedem Sprint zuweisen



Koordinator: LHu
Schreiber: EPi
Überwacher: ASp
Prüfer: MHa
Integrator: MAh



Zulassungsausschussaccounts per Hand anlegen



Im Account Bewerbungen einsehen + Möglichkeit neue Bewerbung zu erstellen (vgl. KLIPS)



Erst auf Funktionalität konzentrieren, danach verschönern



Nächster Schritt Einführung Django (14.11. 10 Uhr) + relevante Daten definieren

14.11.2020 Kick-Off Prototype Project

Date

14.11.2020

Duration: 9:00 - 16:00

Participants

- @ Max Hammer
- @ Alexander Spindeler
- @ Lukas Humpe
- @ Mostapha Ahaduch
- @ Wenhao Peng
- @ Enrico Piscopello

Topics

- Git Workflow abklären
- Repository Initialisierung
- Grundlegendes Verständnis von Django bei jedem Teilnehmer schaffen
 - Pair-Programming: Grundstruktur & User Modell überschreiben
- Aufteilung der einzelnen Aufgabenbereiche
- Inhalte für Planungsdokumente

Tasks

- Planungsdokumente: @ Enrico Piscopello
- UX Template anpassen: @ Max Hammer @ Alexander Spindeler
- Registrierung & Backend: @ Wenhao Peng @ Lukas Humpe
- Git Build: @ Mostapha Ahaduch

Sprint Backlog

Schlüssel	Zusammenfassung	S	Bearbeiter	Story Point Estimate	Unteraufgaben
SA1-35	Vorlage auswählen & anpassen	✓	Max Hammer	2.0	
SA1-34	Datenfelder definieren	✓	Unassigned	1.0	
SA1-19	Git Build aufsetzen / Qualitätsicherung	✓	Mostapha Ahaduch	1.0	
SA1-18	Git Workflow abklären	✓	Unassigned	1.0	
SA1-17	Repository Initialisierung	✓	Mostapha Ahaduch	1.0	
SA1-6	Ich als Zulassungsausschuss möchte Daten in eine Maske eingeben um mich zu verifizieren.	📌	Wenhao Peng	1.0	SA1-22
SA1-5	Ich als Bewerber möchte Daten in eine Maske eingeben um mich zu registrieren.	📌	Lukas Humpe	1.0	SA1-24 , SA1-49 , SA1-23

SA1-4

Ich als Bewerber möchte Daten in eine Maske eingeben um mich zu verifizieren.



Lukas
Humpe

1.0

SA1-21 , SA1-50

8 Vorgänge

Decisions



Punkte für Prozessplan erarbeitet



Risiken und mögliche Lösungen erarbeitet



Code: Zulassungsausschuss & Empfehlungsschreiben Team Mostapha (@ Mostapha Ahaduch + @ Alexander Spindeler + @ Wenhao Peng)



Code: Bewerber Team Lukas (@ Lukas Humpe + @ Max Hammer)



Zeitaufwand der Userstories wird auf einer Skala 1 - 4 bewertet

16.11.2020 Bi-Daily/ Finish Sprint 1

Date

16.11.2020

Duration: 19:00 - 21:00

Participants

- @ Max Hammer
- @ Lukas Humpe
- @ Enrico Piscopello
- @ Alexander Spindeler
- @ Mostapha Ahaduch
- @ Wenhao Peng

Topics

- Alle Datenfelder für die Modelle Bewerber, Bewerbung, Universitätsabschluss, Schulabschluss und Arbeitserfahrung definieren
- Sprint Retrospective
- Planungsdokument Finale Version
- Zusammenführen aller Code-Komponenten
- mögliche Probleme während des Sprints identifizieren und lösen
- Absprache der einzelnen Aufgaben im Team
 - mögliche neue Aufgaben im Sprint definieren und zuteilen

Tasks

- Bewerber Model&Views: @ Lukas Humpe @ Wenhao Peng
- UI Design & UX Design Zulassungsausschussansicht: @ Alexander Spindeler
- UX Design Bewerberansicht: @ Max Hammer
- Sitzungsprotokoll & Finalisierung Planungsdokument: @ Enrico Piscopello
- Git Build & Modultests: @ Mostapha Ahaduch

Sprint Backlog

Schlüssel	Zusammenfassung	S	Bearbeiter	Story Point Estimate	Unteraufgaben
SA1-51	Protected Routes die Verhindern das nicht eingeloggte oder falsche Nutzergruppen zugriff haben		Lukas Humpe		
SA1-12	Ich als Zulassungsausschuss möchte eine Übersicht über alle Bewerbungen haben		Alexander Spindeler	5.0	SA1-47 , SA1-42 , SA1-56 , SA1-41 , SA1-40
SA1-11	Ich als Bewerber möchte eine Übersicht über meine Bewerbung haben, um meine Daten zu sehen und bearbeiten zu können.		Max Hammer	2.0	SA1-54 , SA1-53 , SA1-55 , SA1-52 , SA1-39
SA1-10	Ich als Bewerber möchte Daten in eine Maske eingeben um bewerbungsrelevante Informationen bereitzustellen/upzudaten.		Lukas Humpe	2.0	SA1-27 , SA1-37 , SA1-29 , SA1-28
SA1-9	Ich als Zulassungsausschuss möchte Bewerber auswählen können um diese zuzulassen.		Wenhao Peng	3.0	SA1-32 , SA1-31 , SA1-30

Decisions



Softwareprojekt soll auf Heroku deployed werden



Zulassungsausschussmitglieder können sich nur mit fest zugewiesenen Account anmelden und können sich nicht registrieren.



Bewerbungen Modular

Schulabschlüsse und Co. werden erst am ende einer Bewerbung hinzugefügt. Hierbei können mehrere Schulabschlüsse und Co. unabhängig von einer Bewerbung erstellt werden

18.11.2020 Bi-Daily/Finish Sprint 2

Date

18.11.2020

Duration: 19:00 - 21:00

Participants

- @ Max Hammer
- @ Lukas Humpe
- @ Wenhao Peng
- @ Alexander Spindeler
- @ Mostapha Ahaduch
- @ Enrico Piscopello

Topics

- Zusammenführen der Code-Komponenten, welche im Sprint 2 entwickelt wurden
- Sprint Retrospective
- mögliche Anpassungen für den nächsten Sprint besprechen

Tasks

- Empfehlungsschreiben Maske: @ Lukas Humpe
- Frontend Finalisierung: @ Alexander Spindeler
- UX Design Empfehlungsschreiben Maske: @ Max Hammer
- Vorbereitung Deployment: @ Mostapha Ahaduch
- Sitzungsprotokoll: @ Enrico Piscopello

Sprint Backlog

Schlüssel	Zusammenfassung	S	Bearbeiter	Story Point Estimate
SA1-62	UpdateView Bewerber Persönliche Daten	✓	Unassigned	1.0
SA1-60	Bewerbung darf nicht gelöscht werden nachdem sie angenommen wurde	✓	Max Hammer	1.0
SA1-59	Frontend Finalisierung Bewerber	✓	Max Hammer	2.0
SA1-58	Frontend Finalisierung Zulassungsausschuss	✓	Alexander Spindeler	2.0
SA1-43	Ich als Zulassungsausschuss möchte sehen wie viele Bewerber ich noch zulassen kann	📌	Alexander Spindeler	1.0
SA1-36	Heroku Deployment	✓	Mostapha Ahaduch	2.0
SA1-13	Ich als Bewerber möchte ein Feedback zu meiner Bewerbung angezeigt bekommen	📌	Max Hammer	2.0
SA1-8	Ich als Dritter möchte Text in eine Maske eingeben um ein Empfehlungsschreiben für einen Bewerber zu hinterlegen.	📌	Lukas Humpe	4.0

8 Vorgänge

Decisions



"Passwort Zurücksetzen" Feature für Bewerber und Zulassungsausschussmitglieder entwickeln in nächstem Sprint

21.11.2020 Bi-Daily/ Finish Sprint 3

Date

21.11.2020

Duration: 9:00 - 16:00

Participants

- @ Max Hammer
- @ Lukas Humpe
- @ Mostapha Ahaduch
- @ Wenhao Peng
- @ Alexander Spindeler
- @ Enrico Piscopello

Topics

- Zusammenführen der Code-Komponenten und gemeinsames Testing

Tasks

- ReadMe Dokument: @ Wenhao Peng
- Benutzerhandbuch: @ Enrico Piscopello
- Passwort Vergessen Feature, Finalisierung Heroku Deployment & Modultests: @ Mostapha Ahaduch
- Landing-Page: @ Max Hammer
- Bug-Fixing & Neue Feature an UI Design angepasst: @ Alexander Spindeler
- Repository Clean-Up: @ Lukas Humpe
- Testdaten in PROD Datenbank: @all

Sprint Backlog

Schlüssel	Zusammenfassung	S	Bearbeiter	Story Point Estimate	Unteraufgaben
SA1-67	Datenbank Exemplarisch Befüllen	✓	Unassigned	1.0	
SA1-66	Django Admin Aufräumen	✓	Lukas Humpe	2.0	
SA1-64	Farben anpassen	✓	Alexander Spindeler	1.0	
SA1-63	Landing Page	✓	Alexander Spindeler	2.0	
SA1-61	Passwort change of User Frontend	✓	Max Hammer	2.0	
SA1-46	Modul Test	✓	Mostapha Ahaduch	2.0	
SA1-45	Stress Test	✓	Mostapha Ahaduch	2.0	

7 Vorgänge

Decisions



Entwicklung von Modul Tests



Landing Page erstellen



Stress Tests nicht zielführend, da Heroku Deployment nur 20 gleichzeitige User erlaubt

23.11.2020 Bi-Daily/Abgabe Website

Date

23.11.2020

Duration: 19:00 - 21:00

Participants

- @ Max Hammer
- @ Lukas Humpe
- @ Mostapha Ahaduch
- @ Wenhao Peng
- @ Alexander Spindeler
- @ Enrico Piscopello

Topics

- Finale Abgabe der Website
 - Release auf GitHub
 - Final Deployment of Heroku

Tasks

- Vorbereitung der Themenpunkte des Reflexiondokuments: @all