

IPA Marc Egli - Puzzle ITC

IPA-Daten und beteiligte Personen	
Firma, Abteilung	Puzzle ITC, /dev/ruby
Berufsschule	GIBB
Valid Experte	Lawson Mike
Hauptexperte	Müller Lorenz
Nebenexperte	Moser Michael
Verantwortliche Fachkraft	Illi Daniel
Zusätzliche verantwortliche Fachkraft	Steiner Robin
Berufsbildner	Steiner Robin
Fachrichtung	Applikationsentwicklung
Projektvorgehensmodell	SCRUM
Jahrgang der IPA-Durchführung und Kanton	IPA 2025, Bern
Abgabedatum	

Tabelle 1: IPA Daten

Teil I

Ablauf, Organisation und Umfeld

Inhaltsverzeichnis

I	Ablauf, Organisation und Umfeld	1
1	Aufgabenstellung	8
1.1	Titel der Arbeit	8
1.2	Ausgangslage	8
1.3	Detaillierte Aufgabenstellung	9
1.3.1	Mittel und Methoden	11
1.3.2	Vorkenntnisse	12
1.3.3	Vorarbeiten	12
1.3.4	Neue Lerninhalte	12
1.3.5	Arbeiten in den letzten 6 Monaten	12
2	Firmenstandards	13
2.1	Code conventions	13
2.1.1	Mehrsprachigkeit	13
2.1.2	Lizenz	13
2.2	Git commit Konvention	14
3	IPA-Schutzbedarfsanalyse	15
3.1	Datensicherheit	15
3.2	Applikationssicherheit	15
4	Organisation der IPA-Ergebnisse	16
4.1	Datensicherung	16
4.1.1	Dokumentation	16
4.1.2	Code	17
4.1.3	Wiederherstellung des Codes	17
4.2	Änderungskontrolle	18
5	Projektmethode	19
5.1	Einsatz von Scrum	19
5.1.1	Sprints	19
5.1.2	Verwaltungstool	19
5.1.3	Meetings	20
5.1.4	Abweichungen	21
5.2	Definition of Done	22
5.2.1	DoD Code	22
5.2.2	DoD Dokumentation	23

5.2.3	Akzeptanzkriterien	23
5.3	Verwendungsgrund	24
6	Projektaufbauorganisation	25
6.1	Projektrollen Scrum	25
6.2	Projektrollen IPA	26
6.3	Projektrollen Scrum in der IPA	27
7	Projektrisiken	28
7.1	Risikoanalyse	28
7.2	Risikomatrix	29
8	Zeitplan	30
8.1	Erläuterung zum Zeitplan	30
8.2	Sprints	30
9	Arbeitsjournale	32
9.1	Tag 1: 04.03.2025	32
9.2	Tag 2: 05.03.2025	35
9.3	Tag 3: 06.03.2025	38
9.4	Tag 4: 07.03.2025	41
9.5	Tag 5: 11.03.2025	45
9.6	Tag 6: 12.03.2025	49
9.7	Tag 7: 13.03.2025	52
9.8	Tag 8: 14.03.2025	55
9.9	Tag 9: 18.03	59
9.10	Tag 10: 19.03.2025	62
9.11	Tag 11: 21.03.2025	64
9.12	Tag 12: 21.03.2025	66
10	Persönliches Fazit	68
10.1	Was funktionierte weniger gut	68
10.2	Was funktionierte gut	68
10.3	Schlussreflexion	69
II	 Projektdokumentation	70
11	Kurzfassung des IPA-Berichtes	71
11.1	Ausgangssituation	71
11.2	Umsetzung	71
11.3	Ergebnis	71
12	Einführung	72
13	Analyse	74
13.1	Ist-Zustand	74

13.1.1	Personen	74
13.1.2	Abonnente	82
13.2	Soll-Zustand	85
13.2.1	Anzeigemasken	85
13.2.2	Bearbeitungsmasken	88
13.3	Fehlende Informationen	92
13.4	Änderungen am Mockup	93
13.5	Bedürfniserhebung	94
13.5.1	Zielsetzung und Planung	94
13.5.2	Methodenwahl	94
13.5.3	Fragenkatalog	96
13.5.4	Ablaufprotokoll	97
13.5.5	Auswertung	98
14	Risikoanalyse und Sicherheitsmassnahmen	105
14.1	Schnittstellen	105
14.2	Benutzer und Datenzugriffe	106
14.2.1	Datenstruktur	107
14.2.2	Beispiel Zugriff Heinz	108
14.2.3	Beispiel Zugriff Tim	109
14.2.4	Beispiel Zugriff Rudolf	110
14.2.5	Bedeutung für die Schnittstellen	110
14.2.6	Risikoanalyse	111
14.3	Risikomatrix	112
14.4	Auswertung	112
14.5	Anforderungen	113
14.5.1	Nicht funktionale Anforderungen	113
14.5.2	Funktionale Anforderungen	114
14.6	Abgrenzung	115
14.7	Persönliche Vorgehensziele	116
15	Entwurf	117
15.1	Anwendungskonzept	117
15.1.1	Anwendungsdiagramm	117
15.1.2	Anwendungsfälle	118
15.2	Systemkonzept	120
15.2.1	Betroffene Services	120
15.2.2	Lösungsvarianten	121
15.2.3	Variantenentscheid	125
15.3	Sicherheitskonzept	129
15.3.1	SQL-Injection	129
15.3.2	Cross-Site Scripting	129
15.3.3	URL Interpretation	129
15.3.4	Kommunikation HTTP/S	130
15.4	Fehlerbehandlungskonzept	130

15.4.1	Nutzereingabe	130
15.4.2	Laufzeitfehler	130
15.4.3	Exception Handling	131
15.5	Testsetup	132
15.6	Testkonzept	133
15.6.1	Testinfrastruktur	133
15.6.2	Fehlerklassen	133
15.6.3	Manuelle Tests	133
15.6.4	Automatisierte Tests	145
15.6.5	Begründung der Testwahl	155
16	Abschluss Sprint Initialisierung	156
16.1	Sprint Diagramme	157
16.1.1	Burn up	157
16.1.2	User Story Chart	157
17	Umsetzung	158
17.1	Schnittstellen	158
17.2	Hinzufügen-Button	160
17.3	Turbostreams	162
17.3.1	Partials	163
17.3.2	Name-Attribut auf dem Select-Feld	164
17.4	Globale Bedingungen	165
17.5	Einsatz von KI-Modellen	165
17.5.1	Eingabewerte	166
17.5.2	Höhere Effizienz	166
17.5.3	Höhere Effektivität	167
17.5.4	Fazit	169
17.6	Testprotokoll	171
17.6.1	Manuelle Tests	171
17.6.2	Feature Tests	174
17.6.3	Fazit	177
17.7	Instruktion	178
17.7.1	Anleitungen	178
17.7.2	Durchführung der Instruktion	184
17.7.3	Fazit	187
18	Abschluss Sprint Umsetzung	188
18.1	Sprint Diagramme	189
18.1.1	Burn up	189
18.1.2	User Story Chart	189
19	Einführung	190
19.1	Einführung des Systemes	190
19.2	Weitere Arbeiten	190

20	Abschluss Sprint Finalisierung	192
20.1	Sprint Diagramme	193
20.1.1	Burn up	193
20.1.2	User Story Chart	193
III	Anhang und Verzeichnisse	194
21	Verzeichnisse	195
21.1	Tabellenverzeichnis	195
21.2	Abbildungsverzeichnis	197
	Quellenverzeichnis	202
22	Verwendete Abkürzungen	203
23	Glossar	204
24	Anhänge	205
24.1	Git commit Message Konvention	205
24.2	Sitzungsprotokolle	206
24.2.1	Sitzung 1	206
24.2.2	Sitzung 2	208
24.3	Code	209
24.3.1	people_filter_ability.rb	209
24.3.2	people_readables.rb	210
24.3.3	people_filters_controller.rb	211
24.3.4	people_filter_helper.rb	212
24.3.5	filter_dropdown_controller.js	213
24.3.6	form_field_toggle_controller.js	214
24.3.7	form_select_controller.js	215
24.3.8	_filter.scss	215
24.3.9	_filter.html.haml	216
24.3.10	people_filters/_form.html.haml	217
24.3.11	people_filters/_form.html.haml	218
24.3.12	_qualification.html.haml	219
24.3.13	_qualification.html.haml	220
24.3.14	_qualification.html.haml	221
24.3.15	_role.html.haml	222
24.3.16	_role.html.haml	223
24.3.17	_tag.html.haml	224
24.3.18	create.turbo_stream.haml	225
24.3.19	delete.turbo_stream.haml	225
24.3.20	subscriber/_form.html.haml	225
24.3.21	routes.rb	225
24.3.22	global_conditions_spec.rb	226
24.3.23	people_filter_spec.rb	227

24.3.24 people_filter_spec.rb	228
24.3.25 people_filter_spec.rb	229
24.3.26 people_filter_spec.rb	230
24.3.27 role.rb	230
24.3.28 _main.scss	231
24.3.29 qualifications.yml	231
24.3.30 views.fr.yml	231
24.3.31 views.it.yml	232
24.3.32 views.de.yml	233
24.3.33 views.en.yml	234

1 Aufgabenstellung

1.1 Titel der Arbeit

Hitobito: Neue Generation von Personen-Filtern

1.2 Ausgangslage

Hitobito ist eine Open Source Webapplikation zum Verwalten von Mitgliedern, Events und vielem mehr. Die Ruby on Rails Applikation wurde 2012 von Puzzle ITC initiiert und wird stets weiterentwickelt.

Die Basis für die Software bildet das Webframework Ruby on Rails. Für das User Interface wird neben statischer Technologie wie HTML und CSS auch JavaScript oder Hotwire verwendet. Der komplette Source-Code steht auf Github zur Verfügung: [Hitobito](#)

Eine Kernfunktionalität von Hitobito ist das Filtern von Personenlisten und von Mailinglistenempfängern mit konfigurierbaren Filtern. Diese werden über das Webinterface konfiguriert. Das Webinterface wurde mit statischen Webtechnologien entwickelt und ist inzwischen ziemlich in die Jahre gekommen.

Eine Erneuerung dieser Komponente ist ein Wunsch vieler Kunden.

1.3 Detaillierte Aufgabenstellung

Mit dieser IPA soll ein neues UI mit Hotwire für die Persistierung von Filter-Parametern im Hitobito Generic-Wagon erstellt werden (rein Frontend).

- Die Ansichten zur Konfiguration für Filter der Personenlisten und Abonnemente werden mit dem neuen UI ersetzt.
- Die neuen Ansichten werden nach einem gegebenen Mockup umgesetzt. Dieses Mockup wurde vom Kandidaten in Zusammenarbeit mit einem UX Experten erarbeitet und muss als Grundlage für die Ausarbeitung des Interfaces verwendet werden. Des weiteren muss das Interface auf das visuelle Design der existierenden Applikation abgestimmt sein.
- Das Backend darf nicht angepasst werden, das heisst das neue Interface verwendet die bestehenden Endpunkte und schickt die Daten im selben Format wie das alte Interface. Dies muss mit automatisierten Tests sichergestellt werden.
- Formular zur Konfiguration von Personen-Listen Filter: Das bestehende Formular muss ersetzt werden durch eine neue Implementation mit den in Mittel und Methoden definierten Web Technologien. Diese neue Umsetzung muss es erlauben, dynamisch weitere Filterkriterien hinzuzufügen im Gegensatz zur alten Implementation welche mit einem statischen Formular implementiert ist.
- Formular zur Konfiguration von Abo-Empfänger Filter: Das bestehende Formular besteht aus mehreren Teilen, wovon im Rahmen der IPA nur der Teil für die Globalen Filterbedingungen angepasst werden muss. Wie bei den Personen-Listen Filter muss das Formular nun dynamisch implementiert werden. Die Formulare für die weiteren Filterbedingungen werden im Rahmen der IPA nicht angepasst.
- Code der während dieser IPA entsteht soll auf ein privates Github Repo gepushed werden. Die VFs haben dabei stets Lese-Rechte.
- Die Konventionen des Ruby Style Guide, des Rails Style Guide und für Git commit Messages müssen eingehalten werden (siehe Mittel und Methoden).

Out of Scope - wird erst nach der IPA umgesetzt:

- Filterung für Rollen, Gruppen, Events, People bei Abonnements.
- Anpassungen der Ansicht in den anderen Wagons.
- Anpassungen der bisher bestehenden Tests in Hitobito welche die zu erweiternden Ansichten betreffen.

Weitere Anforderungen zu spezifischen Bewertungskriterien:

- G1: Dokumentation fachlicher und technischer Anforderungen: Die fachlichen und technischen Anforderungen müssen dokumentiert werden.
- G10: Konforme Implementierung und Versionierung: Applikationen und Schnittstellen müssen konform implementiert und versioniert werden.
- A13: Erhebung und Dokumentation der Bedürfnisse und Umfeld: Die Bedürfnisse und das Umfeld werden adäquat erhoben und dokumentiert.
- A15: Instruktion: Es wird für den Projektowner eine Instruktion durchgeführt. Diese muss dem Projektowner die relevanten Änderungen aufzeigen.
- C11: Einsatz von KI-Modellen: Wir setzen bei Puzzle KI in Form von Kopiloten und Chatbots als Hilfsmittel ein. Die Lernenden werden im sinnvollen Einsatz von solcher KI geschult. Dies umfasst z.B. den Umgang in Bezug auf Output Validierung, Transparenz und Sicherheit. Die IPA soll möglichst repräsentativ für unseren Alltag als Entwickler sein, dementsprechen darf KI ein Teil davon sein.
- G5: Risikoanalyse und Sicherheitsmassnahme: Sicherheitsrisiken von Applikationen und Schnittstellen müssen identifiziert und adressiert werden.
- G6: Entwicklung und Anpassung des Anforderungskatalogs: Ein Anforderungskatalog für Sicherheitsmassnahmen von Applikationen und/oder Schnittstellen muss erstellt oder angepasst werden.
- User Experience und visuelles Design: Das Feature muss visuell gut gestaltet sein um die Usability und Nutzerfreundlichkeit des Features sicherzustellen.

- Versionsverwaltung mit Git (Source Code): Die Versionsverwaltung mit Git muss gemäss den Best Practices erfolgen. Es müssen sprechende und einheitliche Commit-Messages geschrieben werden und commit-spezifische Inhalte müssen passend zur Message sein und unter der Einhaltung der Firmenguidelines erfolgen.
- Bewertung von Aussagen: Aussagen in der Arbeit müssen klar zwischen persönlichen Meinungen und auf Quellen basierenden Informationen differenziert werden.

1.3.1 Mittel und Methoden

Technologie und Plattform:

- Ruby, Ruby on Rails, Active Record
- HTML, CSS, Javascript, Hotwire
- PostgreSQL
- Git

Entwicklungsumgebungen:

- IntelliJ
- Visual Studio Code
- Github
- Rake
- Rubocop

Textverarbeitung und Diagramme

- Latex
- draw.io

1.3.2 Vorkenntnisse

Marc arbeitet bereits seit einigen Monaten an Features von Hitobito. Ausserdem hat er bereits seit dem 2. Lehrjahr Erfahrung auch in anderen Ruby on Rails Projekten gesammelt.

1.3.3 Vorarbeiten

- Vorbereitung Dokumentvorlage
- Probe-IPA: Vereinheitlichung der Personenlisten- und Abonnementsfilterlogik im Backend
- Entwurf eines Mockups

1.3.4 Neue Lerninhalte

- Eigenständiges Umsetzen eines Designs nach gegebenem Mockup
- Eigenständiges Projektmanagement während der IPA

1.3.5 Arbeiten in den letzten 6 Monaten

- Umsetzung diverser Features und Bugfixes für Hitobito (Ruby on Rails)
- Probe-IPA: Vereinheitlichung der Personenlisten- und Abonnementsfilterlogik
- PostgreSQL Migration Hitobito
- Ruby on Rails Major Upgrade Hitobito

2 Firmenstandards

Standards welche von Puzzle ITC festgelegt und in dieser Arbeit verwendet werden.

2.1 Code conventions

Als Code convention werden die Ruby [Style Guides](#) verwendet. Die Überprüfung dieser Style Guidelines wird mit Rubocop (Formatter) sichergestellt. Die Konfiguration dieses Formatters ist unter [rubocop.yml](#) ersichtlich.

2.1.1 Mehrsprachigkeit

Hitobito ist eine mehrsprachige Applikation. Alle Erweiterungen oder Anpassungen müssen in Deutsch übersetzt werden. Übersetzungen werden in einer Übersetzungsdatei gespeichert oder können vom Kunden in einem Tool namens Transifex verwaltet werden.

2.1.2 Lizenz

Hitobito ist ein Open Source Projekt. In jedem File in Hitobito wird das Copyright für den jeweiligen Kunden in Kommentarform beschrieben. Diese Lizenz- und Kundeninformationen können über folgenden Befehl eingefügt werden:

```
rake license:insert
```

Die daraus entstehende Lizenz sieht wie folgt aus:

```
1  # Copyright (c) 2012 -2021 , hitobito AG . This file is part of
2  # hitobito and licensed under the Affero General Public License version 3
3  # or later . See the COPYING file at the top - level directory or at
4  # https :// github . com / hitobito / hitobito .
```

Alternativ dazu können diese Informationen mit

```
rake license:remove
```

entfernt oder mit

```
rake license:update
```

aktualisiert werden.

2.2 Git commit Konvention

Die Git commit Messages werden nach den Regeln von Puzzle ITC formuliert. Im Anhang unter Git commit Message Konvention finden sie eine Kopie der Firmenkonventionen. Diese wurden basierend auf folgendem Tutorial definiert: [Tutorial](#)

- Sprache: Englisch
- Kurze und prägnante Message, idealerweise unter 50 Zeichen
- Mit Grossbuchstaben beginnen
- Kein Punkt am Schluss
- Den *imperative mood* (Befehlsform) verwenden, also «Fix bug with X» statt «Fixed bug with X» oder «More fixes for broken stuff»
- Wenn vorhanden, Ticket referenzieren:
 - Bei Open Project Work Packages: «Add X, refs #12345»
 - Bei Gitlab/Github Issues: «Add X #12345»

3 IPA-Schutzbedarfanalyse

Welcher Schutz wird für die Daten und die Applikation im Rahmen dieser Arbeit benötigt?

3.1 Datensicherheit

Die notwendigen Daten, welche im Rahmen der IPA zu Test- und Vorführungszwecken verwendet werden, sind durch das [Faker-Gem](#) generiert. Diese Daten sind als nicht besonders schützenswert zu klassifizieren.

3.2 Applikationssicherheit

Im Rahmen der IPA wird nicht mit besonders schützenswerten Daten gearbeitet. Bei der Implementierung muss sichergestellt werden, dass Benutzer nur auf Daten zugreifen können, welche durch ihre Berechtigungen abgedeckt sind. Diese Sicherheit wird durch die Verwendung des [can-can-can](#) Gems gewährleistet. Dies ist wichtig, da sich bei späterer Implementierung der IPA in Hitobito besonders schützenswerte Daten in der produktiven Datenbank befinden.

4 Organisation der IPA-Ergebnisse

Wo werden die Ergebnisse der IPA abgelegt und wie wird im Fall eines Datenverlustes darauf eingegangen?

4.1 Datensicherung

In dieser IPA unterteilen wir die Datensicherung in:

- Dokumentation
- Code

4.1.1 Dokumentation

Dokumentation	
Tools	Git und USB
Versioniert	Ja
Interval	Mind. 2x täglich
Beschreibung	Die Dokumentation ist im ipa Repository unter dem Branch master angelegt. Sobald eine User Story der Dokumentation abgeschlossen wurde, werden die Änderungen auf den Github Server in das private Repository gepushed. Dies geschieht mind. 2x täglich. Zusätzlich wird pro Tag ein Ordner auf einem USB-Stick erstellt. Am Ende des Tages wird eine Kopie der Dokumentation in diesen Ordner geladen

Tabelle 4.1: Sicherung Dokumentation

4.1.2 Code

Code	
Tools	Git und USB
Versioniert	Ja
Interval	Mind. 2x täglich
Beschreibung	Für die Entwicklung wurden die Repositories Hitobito und Hitobito-Generic geforked. Auf diesen Repositories wird an Tagen, an welchen entwickelt wird, mind. 2x täglich committed. Zusätzlich wird zur doppelten Sicherung eine Kopie des Projektes, unter dem Ordner des jeweiligen Tages, auf den USB Stick gespeichert

Tabelle 4.2: Sicherung Code

4.1.3 Wiederherstellung des Codes

Im Falle eines Datenverlusts, können die Daten entweder über das Github Repository oder den USB-Stick wiederhergestellt werden. Bei der Wiederherstellung mit Git, wird der SSH-Key für die Klonung des Repositories benötigt. Ist dieser SSH-Key nicht verfügbar, wird die Wiederherstellung über den USB-Stick vorgenommen und das Projekt des letzten Speicherstandes kopiert. Im Falle der Wiederherstellung mit dem USB-Stick, ist mit mehr Datenverlust zu rechnen wenn dieser gegen Mittag oder Nachmittag auftritt, da die Speicherung erst am Ende des Tages erfolgt. Aus diesem Grund ist die Datenwiederherstellung mit Git zu bevorzugen.

Die Nachweise für die Datensicherung finden Sie im Anhang.

4.2 Änderungskontrolle

Version	Datum	Name	Beschreibung
Version 0.1	04.03.2025	Marc Egli	Tag 1 abgeschlossen
Version 0.2	05.03.2025	Marc Egli	Tag 2 abgeschlossen
Version 0.3	06.03.2025	Marc Egli	Tag 3 abgeschlossen
Version 0.4	07.03.2025	Marc Egli	Tag 4 abgeschlossen
Vorlage 0.5	11.03.2025	Marc Egli	Tag 5 abgeschlossen
Version 0.6	12.03.2025	Marc Egli	Tag 6 abgeschlossen
Version 0.7	13.03.2025	Marc Egli	Tag 7 abgeschlossen
Version 0.8	14.03.2025	Marc Egli	Tag 8 abgeschlossen
Version 0.9	18.03.2025	Marc Egli	Tag 9 abgeschlossen
Version 1.0	19.03.2025	Marc Egli	Tag 10 abgeschlossen
Version 1.1	20.03.2025	Marc Egli	Tag 11 abgeschlossen
Version 1.2	21.03.2025	Marc Egli	Tag 12 abgeschlossen

Tabelle 4.3: Änderungsprotokoll

5 Projektmethode

Die verwendete Projektmethode dieser IPA ist Scrum. Im folgenden Abschnitt wird der Einsatz, Abweichungen, Werkzeuge und Begründung der Wahl dieser Projektmethode beschrieben. Des Weiteren beschreibt dieser Abschnitt die Definition of Done (DoD).

5.1 Einsatz von Scrum

Wie wird Scrum im Rahmen dieser IPA eingesetzt?

5.1.1 Sprints

Die IPA wird insgesamt in drei Sprints unterteilt. Jedem Sprint wird eine Phase der Arbeit zugewiesen. Die Phase definiert das Ziel des jeweiligen Sprints. Die Aufteilung ist wie folgt:

- Sprint 1: Initialisierung
- Sprint 2: Umsetzung
- Sprint 3: Finalisierung

5.1.2 Verwaltungstool

Als Verwaltungstool wird Github Projects eingesetzt. Das Board hierzu kann unter [Github Board](#) aufgerufen werden. Das Board ist in sechs Spalten unterteilt:

- Backlog: User Stories werden grob erfasst, keine Details nötig.
- Refinement: User Stories werden genauer beschrieben und Akzeptanzkriterien werden definiert.

- Ready: User Story wurde refined und geschätzt. Sie kann jetzt bearbeitet werden.
- In-Progress: User Story wird momentan bearbeitet.
- In-Review: User Story wurde abgeschlossen, alle Akzeptanzkriterien sind erfüllt.
- Done: User Story erfüllt DoD (Definition of Done).

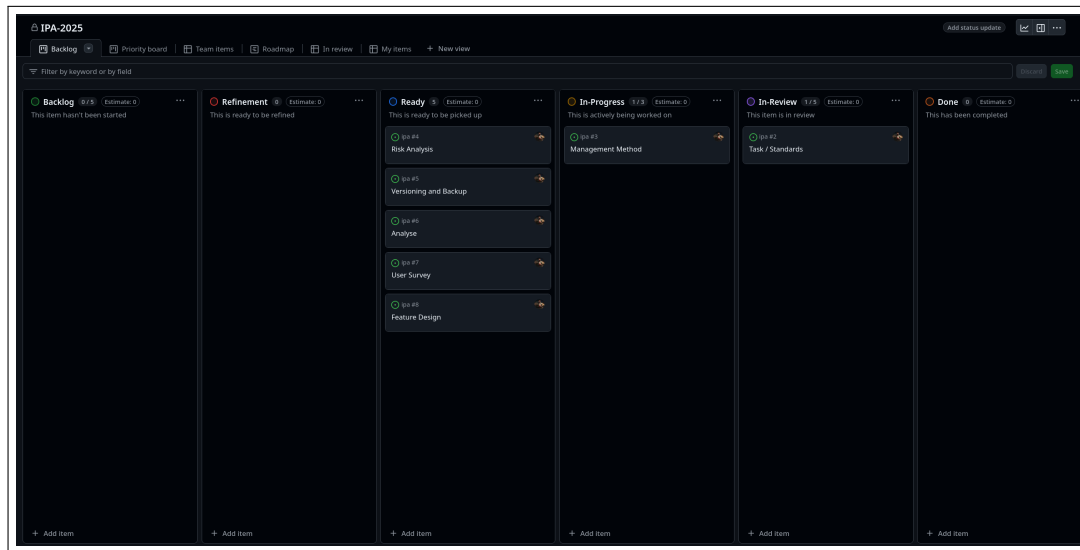


Abbildung 5.1: Github Projects Board

5.1.3 Meetings

Sprint Planning

Zu Beginn eines Sprints werden werden alle Aufgaben in Form von User Stories im Backlog erfasst. Die Stories werden anschliessend refined und danach geschätzt. Das Sprint Planning umfasst den Prozess der Erfassung von User Stories, deren Refinement und Schätzung. Konnten im letzten Sprint die geplanten User-Stories nicht alle abgeschlossen werden, umfasst das Planning zusätzlich das Neurefinement und die Neuschätzung dieser User Stories. Anwesend beim Sprint Planning ist ausschliesslich der Kandidat.

Dailies

Während eines Sprints wird jeden Tag um 09:00 Uhr ein Daily durchgeführt. Das Daily findet bei Puzzle ITC im Sitzungszimmer “Sudo” statt. Anwesend

sind dabei der Kandidat, die verantwortliche Fachkraft und die zusätzlich verantwortliche Fachkraft. Ausgenommen von dieser Regel ist der erste Tag der IPA (04.03.2025), an welchem kein Daily durchgeführt wird. Grund dafür ist, dass zu diesem Zeitpunkt noch keine Organisation und Projektvorgehensweise definiert wurde und die ersten Prozesse von Scrum erst ab dem 2. Tag eintreffen können.

Im Daily ist es dem Kandidat möglich, Fragen an seine verantwortlichen Fachkräfte zu stellen. Abläufe im Daily oder wichtige Vorkommnisse werden im Arbeitsjournal dokumentiert.

Sprintabschlüsse

Nach jedem Sprint findet ein einstündiges Meeting für den Sprintabschluss statt. Darin werden die abgeschlossenen User-Stories in der In-Review-Spalte verifiziert. Erfüllt die hinterlegte Arbeit alle Akzeptanzkriterien, wird die User Story auf Done geschoben. Sind die Akzeptanzkriterien nicht erfüllt, wird die User Story auf Refinement geschoben. Anwesend beim Sprintabschluss ist ausschliesslich der Kandidat. In Folge des Sprintabschlusses, wird das Sprint Planning durchgeführt.

5.1.4 Abweichungen

Trotz der Verwendung von Scrum, wurden Änderungen an der Definition dieser Projektvorgehensmethode vorgenommen. Grund dafür ist, dass Scrum durch die Änderungen besser auf die IPA zugeschnitten ist.

Schätzung

Scrum verzichtet auf Schätzungen in Personenstunden und verwendet deswegen eine Währung namens "Story Points". Story Points werden der [Fibonacci-Zahlenreihe](#) folgend vergeben. Der Sinn dabei ist, der Schätzung einer User Story nach Personenstunden auszuweichen.

Dieses Konzept wird in dieser IPA verworfen, um in der Lage zu sein, einen Zeitplan mit genauen Angaben in Personenstunden zu erstellen. Dies macht es dem Kandidaten möglich, besser einzuschätzen wie gut er in der Zeit liegt.

Abnahme Akzeptanzkriterien

Nach Scrum werden User Stories vom Product Owner abgenommen. Um ständige Meetings mit dem Product Owner von Hitobito und den mithergehenden Zeitverlust zu vermeiden, werden die User Stories vom Kandidaten selbst abgenommen. Der Prozess dazu ist unter [Sprintabschlüsse](#) ersichtlich.

Sprint Retro

Das Sprint Retro bietet dem Product Owner die Möglichkeit, einen Überblick über die Stimmung im Entwicklerteam zu erhalten. Sprint Retros finden monatlich im Geschäftsalltag statt. Auf das Sprint Retro wird in dieser Arbeit verzichtet. Grund ist der kleine Zeitrahmen der IPA, welcher es unnötig macht, ein solches Meeting durchzuführen.

5.2 Definition of Done

Die Definition of Done definiert, wann eine User Story abgeschlossen werden kann. Eine User Story kann erst abgeschlossen werden, wenn sie alle Kriterien der Definition of Done erfüllt. Im Rahmen der IPA werden zwei Definition of Done's verwendet. Eine für User Stories welche den Code betreffen, eine zweite für User Stories, welche die Dokumentation betreffen.

5.2.1 DoD Code

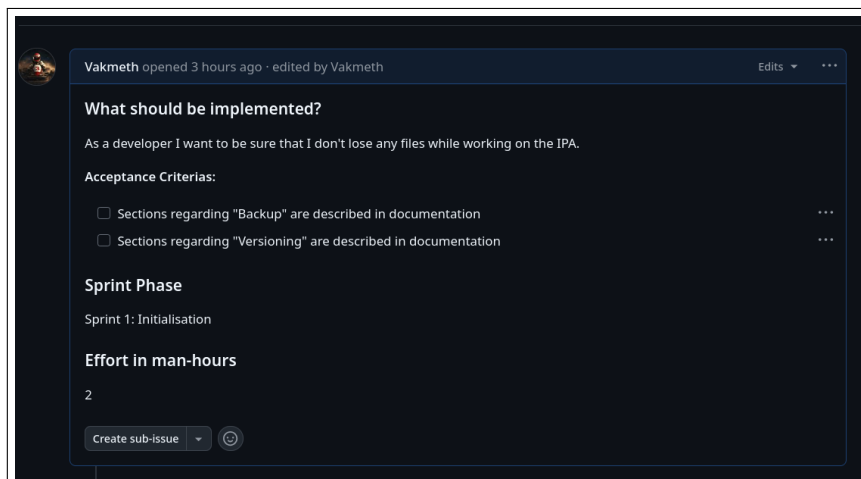
- Nur notwendige Konsolenausgaben vorhanden
- Feature relevante Tests vorhanden
- Sprechender Code implementiert
- Nicht verwendete Methoden gelöscht
- Feature manuell getestet
- Alle Akzeptanzkriterien erfüllt

5.2.2 DoD Dokumentation

- Definierte Sektion beschreiben
- Kriterien aus Kriterienkatalog erfüllt
- Kriterien gemäss Dokumentenvorlage erfüllt
- Keine Grammatik- / Rechtschreibfehler vorhanden
- Quellen angegeben

5.2.3 Akzeptanzkriterien

Die Akzeptanzkriterien einer User Story, werden im dazugehörigen Ticket verwaltet. Jede User Story wurde nach einem definierten Template erstellt, welches in Github hinterlegt wurde. Eine User Story kann folgendermassen aufgebaut sein:



The screenshot shows a GitHub issue template with the following structure:

- Header:** "Vakmeth opened 3 hours ago · edited by Vakmeth" with "Edits" and a menu icon.
- Title:** "What should be implemented?"
- Description:** "As a developer I want to be sure that I don't lose any files while working on the IPA."
- Acceptance Criteria:**
 - ☐ Sections regarding "Backup" are described in documentation
 - ☐ Sections regarding "Versioning" are described in documentation
- Sprint Phase:** "Sprint 1: Initialisation"
- Effort in man-hours:** "2"
- Footer:** "Create sub-issue" button and a GitHub logo.

Abbildung 5.2: Beispiel einer User Story

5.3 Verwendungsgrund

Die Projektvorgehensmethode wurde so gewählt, da sie für die IPA mehrere Vorteile bringt:

- **Sprint Ende:** Scrum zwingt den Entwickler dazu, am Ende des Sprints ein vorzeigbares Produkt zu haben
- **Agilität:** Wenn eine Story nicht erreicht wurde, kann sie in den nächsten Sprint gezogen werden
- **Daily:** Durch die Dailies, wird ein täglicher Austausch zwischen Fachkraft und Kandidat sichergestellt
- **Akzeptanzkriterien:** Mit den Kriterien wird das Abschliessen von halbfertigen Features verhindert
- **Board:** Durch das Github Projects Board, wird eine schnelle Übersicht über den Stand der IPA ermöglicht

6 Projektaufbauorganisation

6.1 Projektrollen Scrum

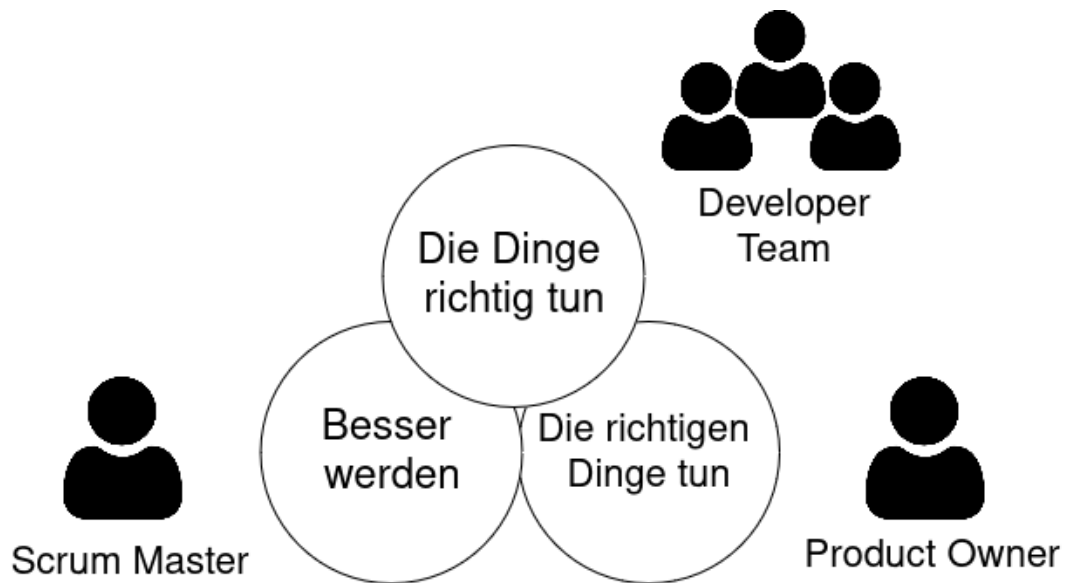


Abbildung 6.1: Rollen in Scrum, selbstgezeichnet mit Draw.io

Rollenbeschreibung	
Product Owner	Der Product Owner vertritt die Interessen des Kunden. Er priorisiert die Aufgaben im Product Backlog
Scrum Master	Der Scrum Master unterstützt die Entwickler und beseitigt Hindernisse. Er sorgt für eine kontinuierliche Verbesserung in der Arbeit
Entwicklerteam	Das Entwicklerteam arbeitet selbstorganisiert den Sprint Backlog ab. Durch Dailies wird ein laufender Informationsaustausch sichergestellt

Tabelle 6.1: Rollenbeschreibung

6.2 Projektrollen IPA

Rollenbeschreibung	
Verantwortliche Fachkraft	Unterstützt den Kandidaten von Seiten des Lehrbetriebes. Erste Anlaufstelle bei Problemen
Zusätzlich verantwortliche Fachkraft	Unterstützung für die verantwortliche Fachkraft
Experten	Validierungsexperte: Validiert die IPA-Aufgabenstellung Hauptexperte: Verantwortlich für die Bewertung der IPA Nebenexperte: Unterstützung für den Hauptexperten

Tabelle 6.2: Rollenbeschreibung

6.3 Projektrollen Scrum in der IPA

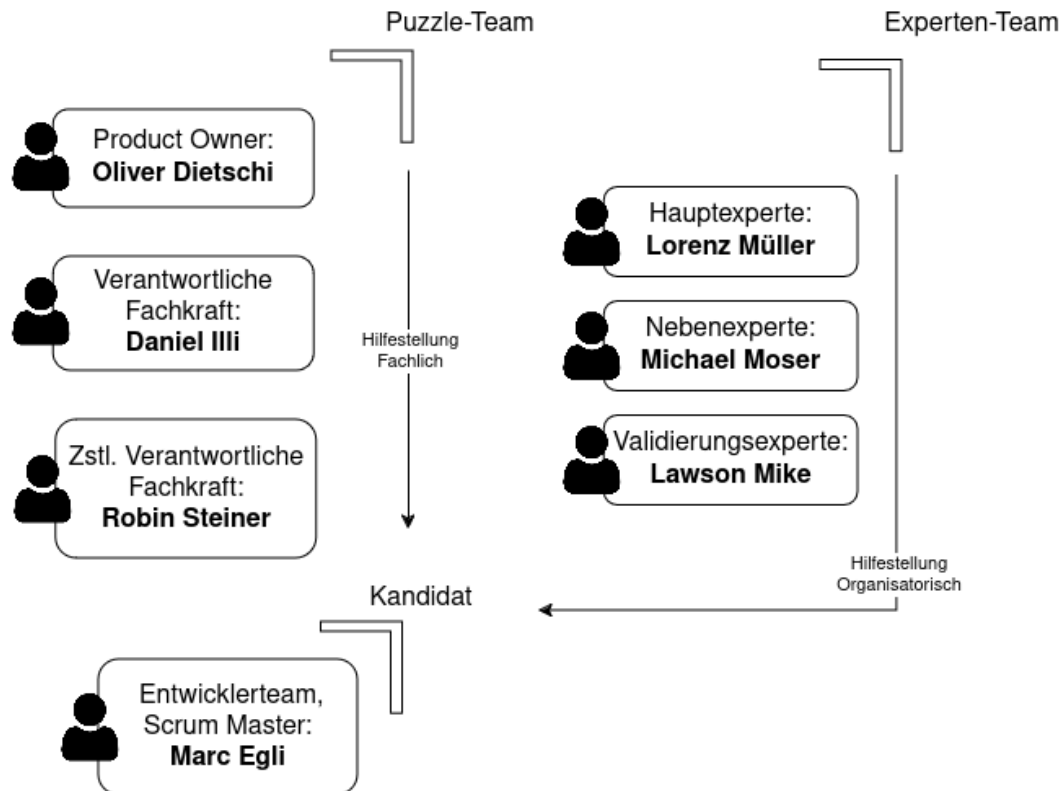


Abbildung 6.2: Rollenverteilung in der IPA, selbstgezeichnet mit Draw.io

Rollenbeschreibung IPA	
Verantwortliche Fachkraft	Daniel Illi
Zusätzlich verantwortliche Fachkraft	Robin Steiner
Validierungsexperte	Lawson Mike
Hauptexperte	Lorenz Müller
Nebenexperte	Michael Moser
Scrum Master	Marc Egli
Development Team	Marc Egli
Kandidat	Marc Egli

Tabelle 6.3: Rollenbeschreibung IPA

7 Projektrisiken

7.1 Risikoanalyse

Nr	Risikobeschreibung	Auswirkung	Vor Massnahme				Massnahmen	Nach Massnahme			
			W	S	Risiko	Handlungsweise		W	S	Risiko	Handlungsweise
1	Datenverlust	Relevante Daten der Arbeit gehen verloren.	W2	S3	Mittel	Risikominderung	Aktuellstes Backup wiederherstellen, Git oder USB-Stick	W2	S2	Niedrig	Risikoakzeptanz
2	Internet fällt aus	Interne Dienste können nicht erreicht werden, remote Meetings können nicht stattfinden	W4	S3	Hoch	Risikominderung	Internetzugang über Mobilnetzanbieter	W1	S1	Niedrig	Risikoakzeptanz
3	Änderungen haben unvorhergesehene Konsequenzen	Konzept muss angepasst werden, die Applikation muss an ungeplanten Stellen angepasst werden.	W3	S2	Mittel	Risikoakzeptanz		W3	S2	Mittel	
4	Docking Station fällt aus	Die Arbeit muss unterbrochen werden	W3	S3	Mittel	Risikominderung	Bei der internen Systementwicklerabteilung Hilfe holen	W1	S1	Niedrig	Risikoakzeptanz

Tabelle 7.1: Risikoanalyse Projektrisiken

Schadensausmass:
S1 = führt zu keinem Schaden am Projekt
S2 = führt zu geringem Schaden
S3 = hoher Schaden
S4 = führt zu schwerem Schaden am Projekt

Eintrittswahrscheinlichkeit:
W1 = unvorstellbar
W2 = unwahrscheinlich
W3 = eher vorstellbar
W4 = vorstellbar
W5 = Eintreffen hoch

7.2 Risikomatrix

W5				
W4			②	
W3		③ ③	④	
W2		①	①	
W1	② ④			
	S1	S2	S3	S4

Tabelle 7.2: Risikomatrix Projektrisiken

Legende:

- Risiko ohne Massnahme
- Risiko nach Massnahme
- Geringes Risiko
- Mittleres Risiko
- Hohes Risiko

8 Zeitplan

8.1 Erläuterung zum Zeitplan

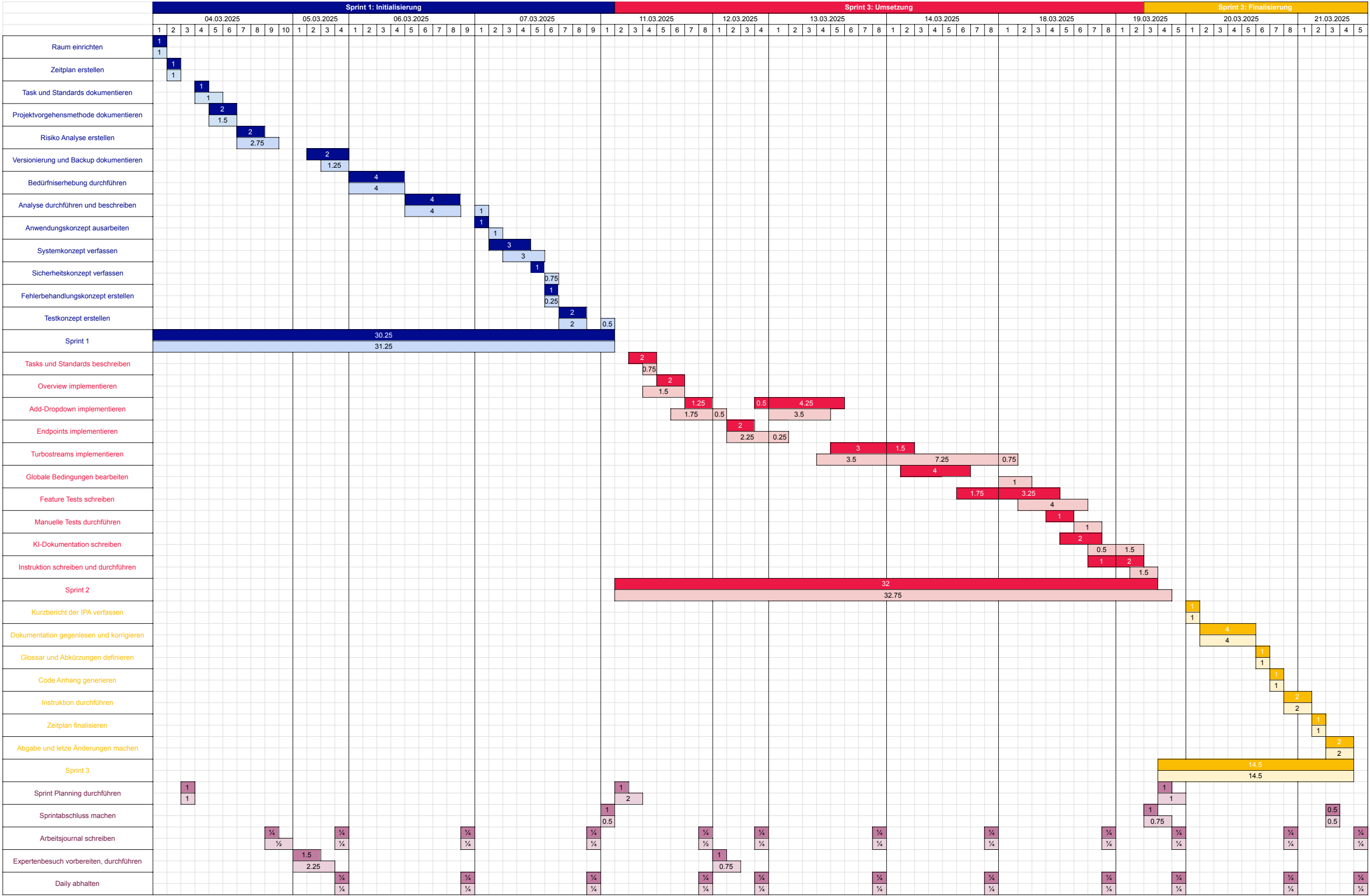
Der Zeitplan auf der folgenden Seite stellt pro Einheit, eine Stunde dar. Die Balken in den dunklen Farben repräsentieren die Soll-Stunden, die unteren in den hellen Farben die effektiv-Stunden. Arbeitsjournal und Plannings wurden aufgrund dem kleinen Aufwand aus dem Zeitplan für diese Arbeit ausgeklammert. Die drei Sprints welche in dieser IPA durchgeführt werden, sind mit unterschiedlichen Farben gekennzeichnet.

8.2 Sprints

Die IPA wurde in drei unterschiedliche Sprints aufgeteilt:

Sprint	Ziel	Datum
Initialisierung	Planung und Beschrieb der Arbeit abschliessen	15.01.2025
Umsetzung	Konzept und Umsetzung selbst abschliessen	21.01.2025
Finalisierung	Details in der Dokumentation verbessern	22.01.2025

Tabelle 8.1: Sprintziele



9 Arbeitsjournale

9.1 Tag 1: 04.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Raum einrichten, Kriterien aufhängen	Marc Egli	1	1
Zeitplan erstellen	Marc Egli	1	1
Sprint Planning durchführen	Marc Egli	1	1.5
Task / Standards beschreiben	Marc Egli	1	1
Projektvorgehensmethode beschreiben	Marc Egli	2	1.5
Risikoanalyse beschreiben	Marc Egli	2	2.75
Arbeitsjournal schreiben	Marc Egli	0.25	0.5
Total		8.25	9.25

Tabelle 9.1: Tätigkeiten Tag 1

Tagesablauf

Ich startet heute Morgen um 07.45 Uhr mit der IPA. Als Erstes begann ich damit, den Raum einzurichten, was bedeutet: Docking Station anschliessen, Wasser bereitstellen und alle Kriterien meiner IPA aufhängen. Danach habe ich alle Kriterien mit verschiedenen Farben unterteilt: Blau steht für Kriterien, welche über die gesamte IPA hinweg zählen, Rosa für Kriterien, welche in der Umsetzung zu beachten sind und Gelb für Kriterien, welche die Dokumentation betreffen. Als ich mit der Zimmereinrichtung fertig war, startete ich direkt mit dem Zeitplan. Ich passte das Template, welches ich vorbereitet habe, auf die Dauer der IPA an und machte alles bereit um die ersten User Stories einzutragen. Nachdem der Zeitplan fertig war, startete ich das Sprint Planning. Darin organisierte ich als Erstes das Daily mit meiner verantwortlichen Fachkraft und meiner zusätzlichen verantwortlichen Fachkraft. Das Daily setzte ich auf 09:00 Uhr morgens an.

Später im Planning, habe ich alle nötigen User Stories für den kommenden Sprint definiert und diese anschliessend in den Zeitplan mit der

dazugehörigen Schätzung übertragen. Auf der Uhr war nun schon 11:15 Uhr. Ich startete den ersten Teil des Beschriebes der Aufgabenstellung und der Firmenstandards und ging danach in den Mittag.

Nach dem Mittag beendete ich den Beschrieb der Aufgabenstellung und der Firmenstandards und begann mit der Projektvorgehensmethode. Hier kam ich überraschend schnell durch und konnte so nach 1.5 Stunden die Risikoanalyse beginnen, an welcher ich bis kurz vor dem Schluss des Tages, 17:30 gearbeitet habe. Beim Erstellen der Risikoanalyse bemerkte ich, dass ich noch Fragen zum Berechtigungskonzept in Hitobito hatte. Dementsprechend ging ich zu Niklas Jäggi, welcher mir dann das Konzept erklärte. Ganz am Ende schrieb ich dann noch das Arbeitsjournal.

Hilfestellungen

- Niklas Jäggi: Erklärung des Berechtigungsaufbaus in Hitobito

Reflexion

Was funktionierte gut

Der Einstieg funktionierte meiner Meinung nach sehr gut. Ich kam schnell voran und konnte die ersten paar Teile der Dokumentation beschreiben. Sogar das erste Kriterium, A11 (Projektaufbauorganisation) konnte ich schon abschliessen, was mich sehr motiviert hat.

Was funktionierte weniger gut

Obwohl ich schnell vorwärts kam, habe ich heute dennoch den geplanten Aufwand um 1/4-Stunde überschossen. Hier muss ich aufpassen, dass ich unbedingt früher anfangen das Arbeitsjournal zu schreiben. Zusätzlich hatte ich beim Sprint Planning ein Problem mit dem Erstellen eines Issue-Templates. Ich hatte mich spontan dazu entschieden, dass es sehr hilfreich wäre, ein Template zu haben, in welchem man neue Issues während der IPA erfassen kann und so nicht alles immer neu machen muss. Allerdings hatte ich noch nie ein solches Template erstellt, weswegen das Planning dann auch eine 1/2-Stunde mehr Zeit in Anspruch nahm als geplant.

Meine heutigen Erkenntnisse

Nicht allzu viel viel Zeit mit Themen verlieren, in welchen ich wenig Erfahrung habe. Besser wäre es gewesen mit dem Issue-Template zu warten und dann in einem Daily danach zu fragen. Dennoch kann ich nun das Wissen um die Erstellung dieses Templates schon als ersten Erfolg in dieser IPA verbuchen.

Nächste Schritte

Morgen werde ich eine Zusammenfassung der Risikoanalyse verfassen, um das Kriterium G5 (Risikoanalyse und Sicherheitsmassnahmen) abzuschliessen. Danach werde ich weiter am Board arbeiten, dass heisst, als nächstes die Sektionen Versionierung und Backup in der Dokumentation beschreiben. Zusätzlich findet am Morgen noch der erste Expertenbesuch statt, welcher mir perfekt dient, um meinen vorbereiteten Fragenkatalog abzuarbeiten. Hier werde ich sicher Fragen zu organisatorischen Bereichen der IPA stellen, wie dem Zeitplan, Diagrammen oder dem Code-Anhang.

9.2 Tag 2: 05.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Daily abhalten	Marc Egli, Robin Steiner, Daniel Illi	0.25	0.25
Expertenbesuch vorbereiten, durchführen	Marc Egli, Robin Steiner, Daniel Illi, Lorenz Müller	1.5	2.25
Versionierung und Backup dokumentieren	Marc Egli	2	1.25
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		4	4

Tabelle 9.2: Tätigkeiten Tag 2

Tagesablauf

Am Morgen startete ich mit der Vorbereitung des Expertenbesuches. Danach fand unmittelbar das Daily statt. Im Daily präsentierte ich den verantwortlichen Fachkräften den Stand der IPA. Danach stellte ich eine Rückfrage an Daniel Illi bezüglich des Berechtigungskonzeptes, da ich 100% sicher sein wollte, dass die Informationen, welche ich von Niklas Jäggi bezogen haben, stimmen. Die Nachfrage ergab, dass das Berechtigungskonzept stimme, jedoch ein Diagramm dies noch falsch abbildete. Ich notierte mir somit die Änderung, welche ich an diesem Diagramm noch machen muss und schloss das Daily ab. Nebst der Nachfrage zum Berechtigungskonzept, fragte ich ob es in Ordnung sei, wenn ich reale Personennamen in einem Diagramm verwende, z.B. Heinz statt Benutzer 1. Hierzu bekam ich die Bestätigung meiner verantwortlichen Fachkräfte.

Nach dem Daily fand dann der Expertenbesuch statt. Das Sitzungsprotokoll hierzu, habe ich im Anhang hinterlegt. Der Besuch lief gut und ich konnte vieles daraus mitnehmen, unter anderem, dass ich eine Person für das Gegenlesen auswählen darf, (ohne diese angeben zu müssen). Ausserdem bekam ich weitere Inputs betreffend dem Zeitplan und meiner Kriterien.

Nach dem Expertenbesuch begann ich mit der Sektion zur Versionierung und der Backup-Strategie meiner IPA. Diese konnte ich zeitig abschliessen und danach das Arbeitsjournal verfassen.

Hilfestellungen

- Daniel Illi: Nachfrage bezüglich des Berechtigungskonzeptes
- Robin Steiner und Daniel Illi: Nachfrage der Verwendung von Echtnamen in Diagrammen

Reflexion

Was funktionierte gut

Der Tag heute war vor allem dem Expertenbesuch gewidmet, welcher ich sehr positiv fand. Obwohl es noch ein paar Anpassungen zu machen gibt, so denke ich, dass durch die Hinweise meines Hauptexperten, die IPA gut herauskommen wird. Wichtig ist jetzt, dass ich das Protokoll für diesen Besuch verfasse und alle gewünschten Änderungen umsetze.

Was funktionierte weniger gut

Heute hatte ich den Eindruck, dass alles gut gelaufen ist. Obwohl es ein paar Fehleinschätzungen im Zeitplan gab, bin ich dennoch immer noch auf Kurs.

Meine heutigen Erkenntnisse

Alle Erkenntnisse, welche ich im Sitzungsprotokoll vermerkt habe. Ausserdem nehme ich noch einen Satz meines Hauptexperten mit: "Nachvollziehbarkeit ist wichtig". Für mich heisst das, alles so klar wie möglich in der Dokumentation zu beschreiben und stets einen Blick auf die Kriterien zu werfen.

Nächste Schritte

Morgen werde ich damit verbringen, die Analyse und die Bedürfniserhebung vorzubereiten. Zusätzlich werde ich die gesammelten Änderungsvorschläge meines Hauptexperten in einem Sitzungsprotokoll aufführen und im Anhang hinterlegen. Die gewünschten Änderungen werde ich dann ebenso direkt umsetzen.

9.3 Tag 3: 06.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Daily abhalten	Marc Egli, Daniel Illi, Robin Steiner	0.25	0.25
Bedürfniserhebung durchführen	Marc Egli, Thomas Ellenberg	4	4
Analyse durchführen und beschreiben	Marc Egli	4	4
Arbeitsjournals schreiben	Marc Egli	0.25	0.25
Total		8.5	8.5

Tabelle 9.3: Tätigkeiten Tag 3

Tagesablauf

Heute startete ich mit der Vorbereitung der Bedürfniserhebung. Ich habe diese als erste Aufgabe am Morgen geplant, um später für das Meeting mit Thomas Ellenberg um 13:00 Uhr, vorbereitet zu sein. Im Daily um 09:00 Uhr, präsentierte ich wie gewohnt den aktuellen Stand der IPA meinen verantwortlichen Fachkräften. Im Daily fragte mich Daniel Illi nach den besprochenen Inhalten des gestrigen Dailies, woraufhin ich ihm meine protokollierten Informationen mündlich weitergab. Danach merkte Robin an, dass ich vergessen hatte, meinen Hauptexperten nach der Verwendung von Echtnamen in Diagrammen zu fragen. Dies werde ich morgen per Mail nachholen. Zusätzlich wies mich Robin Steiner darauf hin, dass die ursprüngliche Scrum Definition vorgesehen hätte, alle drei Sprints grob zu planen und danach die Detailplanung in den Sprint Plannings zu erledigen. Da ich ein anderes Vorgehen gewählt habe, habe ich mir notiert diese Abweichung noch im Abschnitt der Projektvorgehensmethode zu dokumentieren. Abschliessend zum Daily, fragte ich Daniel Illi ob er gerade wisse, ob wir eine Anleitung für die Filterung von Personen im Hitobito haben. Er antwortete, dass ihm das nicht bekannt sei, ich dies jedoch im Benutzerhandbuch nachschlagen könne. Diese Information habe ich benötigt, um eine Wahl für die Bedürfniserhebungsmethode zu treffen und somit die Dokumentenanalyse auszuschliessen.

Nachdem ich nach dem Daily die Bedürfniserhebung vorbereitet hatte, startete ich in die Analyse. Dort begann ich damit, die Ist-Situation aufzunehmen. Gegen den Mittag wurde ich mit dem Beschrieb der Ist-Situation fertig. Nach dem Mittag startete ich direkt mit der Bedürfniserhebung mit Thomas Ellenberg als Testperson. Nachdem ich das Interview durchgeführt hatte, sammelte ich alle Resultate und definierte daraus die Bedürfnisse. Danach merkte ich, dass ich einen Fehler bezüglich den commit Messages gemacht habe. Laut Firmenstandard muss dort stets das Ticket selbst auch angegeben werden, falls eines besteht. Diverse Dokumentierungsaufgaben, welche eich am zweiten Tag erledigt habe, haben auf meinem Github Projects Board ein Ticket. Ich hatte dies allerdings nicht in der commit Message hinterlegt. Um dies zu korrigieren, habe ich in den Github Docs nachgeschlagen, um zu sehen, wie ich ältere commit Messages bearbeite. Darunter fand ich eine Anleitung, welche es mir möglich gemacht hat, die fehlenden User-Story-Referenzen in der Message zu hinterlegen. Allerdings wurden alle Commits beim Push auf den heutigen Tag gelegt. Dies entspricht nicht der Ursprungsverfassung dieser Commits. Um dieses Problem zu lösen werde ich es morgen im Daily vorbringen und allenfalls meinen Hauptexperten um Rat fragen.

Gegen 14:00 Uhr konnte ich die Bearbeitung der Bedürfnisse abschliessen und begann die Soll-Situation in der Analyse zu beschreiben. Die Soll-Situation konnte ich geggeg 16:20 Uhr abschliessen anschliessend begann ich, die Anforderungen aus der gemachten Risikoanalyse und den Bedürfnissen zu erstellen. Zum Schluss des Tages, verfasste ich wie gewohnt das Arbeitsjournal.

Hilfestellungen

- Daniel Illi: Nachfrage der Benutzerdokumentation der Personenfilterung

Reflexion

Was funktionierte gut

Ich hatte heute das Gefühl, dass ich mich viel besser konzentrieren konnte. Ich konnte viel länger gezielt arbeiten und hatte den Eindruck, dass ich schnell vorankomme. Die Bedürfniserhebung war zudem sehr interessant. Der vereinbarte Termin mit Thomas Ellenberg als Testperson, fand wie geplant statt und ich konnte alle Fragen, wie vorbereitet, stellen.

Was funktionierte weniger gut

Obwohl ich schnell vorankam, konnte ich die Analyse nicht vollständig abschliessen. Es bleiben noch die Anforderungen und die Dokumentation der Rahmenbedingungen offen. Ich rechne mit +/- einer Stunde zusätzlichen Aufwand für die Analyse.

Meine heutigen Erkenntnisse

Wichtig ist es früh meinen Fokus zu finden und unnötige Details zu vernachlässigen. Ich sollte mich während dem Arbeiten stets an die Kriterien als Leitfaden halten. Alles was ich erarbeite und nicht in den Kriterien festgehalten ist, wird auch keine Punkte geben und entspricht somit dem Aufwand für nichts.

Nächste Schritte

Der nächste Schritt wird morgen der Abschluss der Analyse. Dies sollte in den ersten zwei Stunden passieren, mehr darf ich unbedingt nicht überziehen. Danach beginne ich mit dem Entwurf welcher Systemkonzept, Testkonzept, etc. umfasst. Ziel ist es, den Entwurf morgen abschliessen zu können, um dann am kommenden Dienstag die Umsetzung zu starten.

9.4 Tag 4: 07.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Analyse abschliessen	Marc Egli	0	1
Daily abhalten	Marc Egli, Daniel Illi, Robin Steiner	0.25	0.25
Anwendungskonzept ausarbeiten	Marc Egli	1	1
Systemkonzept verfassen	Marc Egli	3	3
Sicherheitskonzept verfassen	Marc Egli	1	0.75
Fehlerbehandlungskonzept erstellen	Marc Egli	1	0.25
Testkonzept erstellen	Marc Egli	2	2
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		8.5	8.5

Tabelle 9.4: Tätigkeiten Tag 4

Tagesablauf

Heute morgen startete ich damit, die funktionalen und nicht funktionalen Anforderungen zu dokumentieren. Danach startete um 09:00 das Daily. Im Daily präsentierte ich den Stand der IPA. Danach stellte ich diverse Fragen. Die erste Frage war, ob ich meinen Hauptexperten per Mail die Frage zur Verwendung von Echtnamen in Diagrammen stellen könne. Dies bestätigte mir Robin Steiner und Daniel Illi. Danach sprach ich mein gestriges Problem der commit Messages an. Robin Steiner und Daniel Illi wiesen mich an, die Commits mit dem neuen Zeitstempel bestehen zu lassen und den Prozess des Fehlers bis hin zur Aufklärung im Daily im Arbeitsjournal zu dokumentieren. Nach der Commit-Thematik habe ich eine Frage zum Kriterium A2 gestellt. Dieses besagt, dass alle nicht gegebenen Informationen in der IPA identifiziert und dokumentiert werden müssen. Hierzu habe ich gefragt, ob dieses Kriterium implizit durch das Verfassen des Anhangs und der Dokumentation der Quellen gegeben ist. Robin Steiner verneinte dies und wies mich an eine eigene Sektion in der Dokumentation zu erstellen.

Danach haben wir das Kriterium G5 diskutiert. Im Kriterium ist definiert, dass die Sicherheitsmassnahmen mit dem Team und den Stakeholdern abgesprochen ist. Wir haben meine aufgeführten Risiken und Massnahmen

dazu diskutiert, wobei mir Robin Steiner riet die Formulierung des Risikos von Verwendung von Libraries mit Schwachstellen, neu zu formulieren. Ausserdem riet mit Daniel Illi für Risiken, welche die Berechtigungen des Benutzers betreffen, Feature-Tests als Massnahme zu definieren. Als letztes fragte ich im Daily, ob ich Anpassungen am Mockup machen dürfe. Der Fehler im Mockup ist mir heute morgen aufgefallen, als ich die funktionalen Anforderungen dokumentiert habe und das Mockup vor mir hatte. Es müsste ein Löschen-Button neben dem Bearbeiten-Button in der Filterung hinzugefügt werden, so dass der Benutzer auch Filterkriterien aus der Filterung entfernen kann. Ich präsentierte die Mockup-Änderung im Daily und bekam von meinen verantwortlichen Fachkräften die Bestätigung um die Anpassung zu machen.

Nachdem ich das Daily abgeschlossen hatte, schrieb ich den ersten Teil des Arbeitsjournals, da wir im Daily sehr viel besprochen hatten und ich alle aufgetauchten Fragen und Anmerkungen zeitnah festhalten wollte.

Nachdem ich das Daily abgeschlossen hatte, begann ich den grossen Zeitblock des Entwurfs auf kleinere User Stories umzulagern und diese in meinem Github Board aufzunehmen. Dies tat ich, da mich mein Hauptexperte darauf hinwies, kleinere Zeitblöcke einzuplanen. Direkt nachdem ich die User Stories erstellt hatte, begann ich mit dem Anwendungskonzept. Ich konnte dieses zügig abschliessen und mich dann unmittelbar dem Systemkonzept widmen. Hier stellte ich während der Verfassung meiner Lösungsvarianten ein Problem in der Aufgabenstellung meiner IPA fest: Laut Aufgabenstellung ist es mir untersagt, Anpassungen am Backend oder an den Endpoints zu machen. Genauso sei es verboten, das Datenformat der Endpoints zu ändern. Allerdings werde ich diese beiden Punkte umgehen müssen, denn:

- A: Mein Feature ist auf das Datenformat "Turbostreams" ausgelegt. Ich werde hier also zwingend Änderungen am Datenformat machen müssen
- B: Es müssen separate Endpoints für die verschiedenen Partials in der View angelegt werden, mit den bestehenden Endpoints ist das Feature nicht umsetzbar, da ich die einzelnen Filterkriterien nicht modular austauschen kann

Ich werde diese Feststellung am Montag im Daily mit meinen verantwortlichen Fachkräften besprechen und mit ihnen die nächsten Massnahmen für dieses Problem festlegen.

Nachdem ich das Systemkonzept abgeschlossen hatte, habe ich als nächstes realisiert, dass ich noch nicht erfasste Schnittstellen in der Sicherheitsanalyse nachführen muss. Durch die Analyse und den Entwurf meiner Lösung

bemerkte ich, dass ich dort noch nicht alle Schnittstellen erfasst habe. Dieses Problem habe ich auf meine Pendenzenliste geschrieben und werde es Dienstagmorgen beheben.

Zuletzt begann ich mit dem Testkonzept. Obwohl ich bis dort gut in der Zeit war, reichte es mir nicht ganz fertig, weswegen ich den Restaufwand nächsten Dienstag erledigen werde. Nebenbei schrieb ich Lorenz Müller eine Mail zur Nachfrage, ob ich Echtnamen in Diagrammen verwenden dürfe.

Hilfestellungen

- Robin Steiner und Daniel Illi: Nachfrage ob Hauptexperte per Mail kontaktiert werden darf
- Robin Steiner und Daniel Illi: Nachfrage Vorgehensweise Git-Commits
- Robin Steiner und Daniel Illi: Frage zu Kriterium A2, ob Anhang bereits als Nachweis zählt
- Robin Steiner und Daniel Illi: Diskussion der Sicherheitsrisiken und Massnahmen und somit Erfüllung von Punkt 4 in Kriterium G5.
- Robin Steiner und Daniel Illi: Nachfrage zur Formulierung des Sicherheitsrisikos zu Libraries
- Robin Steiner und Daniel Illi: Frage ob Mockup nachbearbeitet werden darf

Reflexion

Was funktionierte gut

Für mich persönlich war ein Erfolg, dass ich die Probleme in der Aufgabenstellung und meinen eigenen Fehler bezüglich der Dokumentation der Schnittstellen bei den Sicherheitsrisiken frühzeitig erkennen konnte. Dementsprechend kann ich auf diese Fehler noch vor der Umsetzung reagieren und mit einem sauberen Entwurf in diesen Sprint starten. Ausserdem hatte ich das Gefühl, dass ich heute meine Aufgaben besser organisiert hatte. Ich habe viel mehr mit Notizen für Anmerkungen von Robin und Daniel gearbeitet, was es mir ermöglichte, auch das Arbeitsjournal detailtreuer zu schreiben.

Was funktionierte weniger gut

Heute war meine Konzentration verglichen zu gestern überhaupt nicht vorhanden. Ich hatte nur zum Teil das Gefühl, dass ich richtig in den Arbeitsfluss komme. Grund dafür könnte sein, dass heute Freitag ist und ich deswegen abgelenkter war. Deswegen konnte ich auch das Tagesziel von heute nicht erreichen, was bedeutet, dass ich das Testkonzept am Montagmorgen fertigstellen muss.

Meine heutigen Erkenntnisse

Wichtig ist es Unklarheiten und mögliche Konflikte so früh wie möglich zu identifizieren, damit entsprechend auf diese eingegangen werden kann.

Nächste Schritte

Am Dienstagmorgen werde ich zuerst die Schnittstellen der Sicherheitsmassnahmen ergänzen. Danach werde ich das Testkonzept abschliessen und den Sprintabschluss machen. Nach dem Sprintabschluss folgt wie dokumentiert das Planning und die Einteilung der nächsten Aufgaben für den kommenden Sprint. Ziel ist es am Dienstag mit der Umsetzung zu starten.

9.5 Tag 5: 11.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Testkonzept abschliessen	Marc Egli	0	0.5
Sprintabschluss machen	Marc Egli	1	0.5
Daily abhalten	Marc Egli, Robin Steiner, Daniel Illi	0.25	0.5
Sprint Planning durchführen	Marc Egli	1	2
Tasks und Standards beschreiben	Marc Egli	2	0.75
Overview implementieren	Marc Egli	2	1.5
Add-Dropdown implementieren	Marc Egli	1.25	1.75
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		7.75	7.75

Tabelle 9.5: Tätigkeiten Tag 5

Tagesablauf

Den Morgen begann ich mit dem Fertigstellen des Testkonzeptes. Die Schnittstellen welche ich noch überarbeiten wollte, werde ich in einer Story unterbringen, in welcher ich die nötigen Endpoints auch gerade implementiere, so kann ich beides gleichzeitig erledigen ohne viel Zeit zu verlieren. Nach der Fertigstellung des Testkonzeptes schloss ich den Sprint ab. Nahezu alle User Stories konnte ich auf "Done" schieben. Eine Story jedoch nicht, da ich zwei Teile in der Dokumentation noch nicht beschrieben habe. Somit fiel diese User Story zurück ins "Refinement". Nach dem Sprintabschluss fand das Daily statt.

Im Daily sprach ich zum einen das Kriterium zu den Anforderungen an. Dort ist beschrieben das die Anforderungen lösungsneutral sein müssen, jedoch erfüllen meine Anforderungen dies nicht, da ich ein Mockup zur Vefügung habe und somit schon eine Lösung vorgegeben habe. Robin Steiner riet mir dieses Problem morgen mit meinem Hauptexperten zu besprechen. Danach fragte ich meine verantwortlichen Fachkräfte, ob ich Sätze in der Dokumentation mit "In diesem Abschnitt wird XY erklärt", beginnen darf. Die Antwort: Ja, wenn es sinnvoll eingesetzt wird. Die nächste meiner Fragen richtete sich an den Anforderungskatalog. Hier wird im Kriterium G6

beschrieben, dass die Sicherheitsmassnahmen hinterlegt werden müssen. Da ich bereits alle Sicherheitsrisiken und die Massnahmen dafür erfasst habe, fragte ich ob es reiche, wenn ich die Sicherheitsmassnahmen im Anforderungskatalog als Link hinterlegen könne. Robin Steiner und Daniel Illi bestätigten mir dies. Als nächstes habe ich mit meinen verantwortlichen Fachkräften Diagrammstandards beschrieben. hierbei ging es darum, ob ich Standards verwenden soll oder nicht. Die Antwort: Morgen mit dem Hauptexperten klären. Des Weiteren fragte ich, ob es möglich sei mit einem Test mehrere Anforderungen abzudecken. Daniel Illi antwortete darauf, dass dies möglich sei, wenn es sich um einen Feature Test handelt. Falls es ein Unit-Test ist, dürfe er nur eine einzelne Funktion abdecken.

Zudem fiel mir wie dokumentiert am Freitag auf, dass unsere Anforderungen nicht mit den möglichen Lösungsvarianten übereinstimmen. Ich besprach das Problem mit Daniel Illi und Robin Steiner. Daniel Illi riet mir, Anforderungen welche geändert werden müssen oder welche nicht mehr zutreffen, zu dokumentieren. Ich werde dafür einen Abschnitt "Abweichungen" in zweiten Teil der Dokumentation anlegen.

Zum Schluss besprach ich im Team den überarbeiteten Anforderungskatalog, wie es vom Kriterium G6 Punkt vier gefordert wird. Die Änderungen welche meine Stakeholder vorgeschlagen haben, nahm ich auf und werde diese im Anforderungskatalog nachführen. Änderungen bezüglich des Katalogs umfassten vor allem die Formulierung, welche laut Robin Steiner noch etwas "zu holprig" seien.

Nach dem Daily führte ich das Sprint Planning für den nächsten Sprint durch und Plant die Tasks der Umsetzung. Den Nachmittag verbrachte ich bis zum Schluss mit der Implementierung der ersten User Stories. Hierbei freute ich mich sehr, da ich endlich mit der Umsetzung starten konnte. Während des Entwickelns kamen mir Fragen zum Styling und welches Standard Template bei einem GET request auf eine Ressource zurückgegeben wird. Gemäss Kriterium A10, Punkt 4, habe ich in unserem internen Firmenchat diese Fragen an Daniel Illi gestellt, um während dem Warten auf eine Antwort spetitiv weiterarbeiten zu können. Bei der Implementation des Dropdown hatte ich zuletzt den Fehler, dass das ganze Array der Filterkriterien im Stil "[Tags, Attribute, Rollen, Qualifikationen]" angezeigt wurde. Diesen Fehler muss ich morgen noch beheben.

Zum Schluss des Tages, schrieb ich das Arbeitsjournal.

Hilfestellungen

- Robin Steiner, Daniel Illi: Alle Fragen welche gemäss Tagesablauf dokumentiert wurden

Reflexion

Was funktionierte gut

Heute hatte ich das Gefühl, dass ich sehr schnell voran kam. Ich konnte die meisten Tasks wie geplant abhandeln und bin momentan mehr oder weniger im Zeitplan. Ausserdem hatte ich viel mehr Freude an der Arbeit, da ich endlich zum ansetzen Entwickeln kam.

Was funktionierte weniger gut

Ich hatte zu Beginn Probleme in den “Entwicklermodus” zu kommen und mich schnell im Code zurecht zu finden. Dies löste sich aber schnell und ich konnte die erste User Story bezüglich der Implementation umsetzen. Zudem kam, dass ich das gesetzte Tagesziel erreicht habe und mit der Umsetzung starten konnte.

Meine heutigen Erkenntnisse

Wichtig ist es, dass ich meine Zeiten immer sauber raportiere ansonsten habe ich Probleme. Ich merkte dies im Verlaufe des heutigen Tages, als ich mich plötzlich gefragt habe “Was genau hast du jetzt die letzte Stunde gemacht?”. Als Gegenmassnahme habe ich gemerkt, dass es mir sehr hilft, wenn ich immer nach Abschluss einer User Story den Zeitblock auf Papier notiere und dann direkt in den Zeitplan einschreibe. So habe ich am Ende des Tages eine saubere Übersicht über meine Zeiten und kann diese entsprechend im Arbeitsjournal eintragen.

Nächste Schritte

Morgen werde ich den Fehler im Dropdown der Filterkriterien beheben und anschliessend mit der Umsetzung weiterverfahren. Unter anderem sollen laut

meiner Planung die Endpoints für die jeweiligen Partialis implementiert werden.

9.6 Tag 6: 12.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Add-Dropdown Fehler beheben	Marc Egli	0	0.5
Daily abhalten	Marc Egli, Robin Steiner	0.25	0.25
Expertenbesuch vorbereiten, durchführen	Marc Egli, Robin Steiner, Daniel Illi	1	0.75
Endpoints implementieren	Marc Egli	2	2.25
Dropdown implementieren	Marc Egli	0.5	0
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		4	4

Tabelle 9.6: Tätigkeiten Tag 6

Tagesablauf

Wie geplant habe ich heute Morgen zuerst den Fehler im Dropdown behoben. Der Fehler war eigentlich sehr offensichtlich: Wenn man in Ruby das “=”-Zeichen für Logik verwendet, wird der entsprechende Output auch in der View als Text angezeigt. Da ich eine For-Each-Schleife benutzt habe, um alle Filterkriterien im Dropdown als Optionen anzuzeigen und dabei das “=”-Zeichen verwendet habe, wurde am Schluss das ganze Array als Text angezeigt. Um dieses Fehlverhalten zu beheben, gilt es nur das “=” zu einem “-” zu wechseln.

Nachdem ich diesen Fehler behoben habe, führten wir das Daily durch. Anwesend waren nur Robin Steiner und ich, da ich die Terminänderung des Dailies aufgrund des Expertenbesuches zu spät mitgeteilt habe und er dadurch nicht am Daily teilnehmen konnte. Dies war jedoch nicht weiter schlimm, da ich für das heutige Daily keine Fragen notiert habe.

Im Daily präsentierte ich Robin Steiner den Stand der IPA und bestätigte, dass ich alle Punkte, welche mir beim letzten Expertenbesuch aufgezeigt wurden, behoben habe.

Nach dem Daily machte ich alles bereit für die Implementation der Endpoints. Später startete der Expertenbesuch. Die besprochenen

Angelegenheiten sind wie zuvor in einem Sitzungsprotokoll einzusehen. Nach dem Expertenbesuch führte ich meine Arbeit an den Endpoints fort. Ich realisierte, dass ich anders als geplant nur einen Endpoint, anstatt einen Endpoint pro Filterkriterium brauche. So ist die Umsetzung um so besser: Es kann der Route schlichtweg das Filterkriterium mitgegeben werden und die Action des Controllers selbst entscheidet dann über den Parameter, den ich der Route mitgebe, welches Partial gerendert werden muss. So müssen nicht vier verschiedene Endpoints mit eigenen Routen eingesetzt werden, welche am Schluss trotzdem die gleiche Aufgabe haben.

Durch die Änderung an den Anforderungen und weil ich zwischendurch ein Problem damit hatte, die nötigen Berechtigungen an die Route zu vergeben, hat die Umsetzung der Endpoints länger gedauert als gedacht.

Zum Schluss des Tages, schrieb ich wie gewohnt das Arbeitsjournal.

Hilfestellungen

- Keine

Reflexion

Was funktionierte gut

Ich war heute sehr froh, konnte ich den Fehler im Dropdown speditiv beheben. Umso besser war es, dass ich alle Endpoints in einem zusammenfassen konnte. So habe ich nun eine saubere Ausgangslage, um die Turbo Streams zu erfassen. Ebenso hatte ich das Gefühl, dass der Expertenbesuch gut verlaufen ist und mir die Tipps von Lorenz Müller weitergeholfen haben. Im Grossen und Ganzen bin ich sehr positiv eingestellt.

Was funktionierte weniger gut

Ausgenommen der Überschreitung der geschätzten Zeit für die Endpoints und das Dropdown, lief heute alles gut. Ich konnte die Fehler in meinem Konzept früh genug erkennen und habe nun mit der Zusammenfassung der Endpoints angemessen darauf reagiert.

Meine heutigen Erkenntnisse

Auch wenn ein bereits erstelltes Konzept vorhanden ist, kann man diesem nie blind vertrauen. Man sollte es stets hinterfragen und prüfen ob die Angaben darin immer noch korrekt sind. Evtl. waren noch nicht alle Informationen zum Zeitpunkt des Erstellens bekannt? Diese Aussage wird von meiner heutigen Entdeckung der Endpoints verifiziert.

Nächste Schritte

Morgen werde ich zuerst die Endpoints dokumentieren. Danach muss das Dropdown fertiggestellt und die Turbo Streams mit den Partialänderungen abgeschlossen werden. Das Ziel ist es, morgen die Filterkriterien per Dropdown auf die Übersichtskomponente hinzufügen zu können. Die Partials an sich müssen noch nicht zu 100% mit dem Mockup übereinstimmen, dafür haben ich noch die weiteren eingeplanten eineinhalb Stunden am Freitag, aber das Hinzufügen sollte funktionieren.

9.7 Tag 7: 13.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Endpoints dokumentieren	Marc Egli	0	0.25
Add-Dropdown implementieren	Marc Egli	4.25	3.5
Daily abhalten	Marc Egli	0.25	0.25
Turbo Streams implementieren	Marc Egli	3	3.5
Arbeitsjournal schreiben	0.25	0.25	
Total		7.75	7.75

Tabelle 9.7: Tätigkeiten Tag 7

Tagesablauf

Wie geplant begann ich den Tag mit der Dokumentation der Endpoints. Anschliessend fing ich an die ersten Turbostreams zu implementieren. Danach startete das Daily. Im Daily präsentierte ich den Stand der IPA. Danach stellte ich eine Frage zum Kriterium A10. Darin ist beschrieben, dass die Standpunkte der anderen Projektmitglieder erkannt wurden. Meine Frage war, ob das Arbeitsjournal als Nachweis reicht, da ich hier schon alle meine Fragen und Antworten der verantwortlichen Fachkräfte festhalte. Robin Steiner und Daniel Illi bestätigten mir dies. Nach dem Daily startet ich wieder voll mit der Implementation durch. Am Morgen und frühen Nachmittag hatte ich das Gefühl sehr schnell voranzukommen. Ich konnte das Dropdown effizient umsetzen und habe sogar noch ein paar Änderungen am Konzept gemacht, welche die Implementation sauberer machen. Gegen den späten Nachmittag hingegen, nahm meine Konzentration ab. Ich denke, dass es an einem Bug lag, den ich sehr lange bearbeitet habe. Es ging spezifisch um die Tags, welche sich nicht sauber anzeigen liessen. Um nicht noch mehr Zeit hier zu verschwenden, entschied ich mich das Problem morgen im Daily mit meinen verantwortlichen Fachkräften zu besprechen und mit den anderen Partials weiterzumachen.

Während des Implementierens hatte ich stets den Anforderungskatalog offen um sicherzustellen, dass ich alle Anforderungen abdecken kann. Wie ich bemerkte, habe ich die funktionale Anforderung 2 an mein Produkt durch die Implementation des Hinzufüge-Buttons abgedeckt. Auch die funktionale Anforderung 4 konnte ich durch die Implementation der Overview abschliessen.

Ausserdem ist mir eine Bemerkung zum Mockup bezüglich des Filterkriteriums “Felder” aufgefallen. Da ich sowieso einen modularen Aufbau des Filters geplant habe, könnte ich die Komponente lediglich mit der neuen Übersicht anordnen und am Visuellen nichts verändern, da das Filterkriterium “Felder” bereits sehr modular aufgebaut ist. Ich werde diese Erkenntnisse ebenfalls morgen im Daily mit meinen verantwortlichen Fachkräften besprechen.

Gegen Ende des Tages, schrieb ich wie gewohnt das Arbeitsjournal.

Hilfestellungen

- Daniel Illi und Robin Steiner: Nachfrage bezüglich Kriterium A10 wie dokumentiert in Tätigkeiten

Reflexion

Was funktionierte gut

Der Fokus heute Morgen war exzellent. Ich konnte viel implementieren und die ersten Anforderungen abschliessen. Zudem habe ich gemerkt, dass wenn ich endlich einen Bug behoben habe oder ich eine Hürde überwinden konnte, ich mehr Motivation entwickelte, was mir beim Fokussieren sehr half. Zudem konnte ich das gestern gesetzte Tagesziel erfüllen und die neue Benutzerschnittstelle ermöglicht das Hinzufügen von Filterkriterien über das Hinzufüge-Dropdown.

Was funktionierte weniger gut

Am Nachmittag verschwendete ich definitiv zu viel Zeit mit dem Bug beim Filterkriterium der “Tags”. Je länger ich an einem solchen Bug arbeite, desto mehr verliere ich die Konzentration und Motivation. Zusätzlich verschwende ich hier wertvolle Zeit, welche ich für andere Baustellen hätte verwenden können.

Meine heutigen Erkenntnisse

Um dem Problem des Zeitaufwandes für Bugs entgegenzukommen, definiere ich einen Zeitblock von 20 Minuten, welche ich für eine Hürde aufwenden darf. Wenn ich diese Zeithürde überschreibe, lasse ich das Problem vorerst liegen und spreche es im Daily an. Falls mich die Hürde an der Weiterarbeit hindert, gehe ich umgehend zu meiner verantwortlichen Fachkraft und hole mir , um das Problem lösen zu können.

Nächste Schritte

Morgen werde ich die letzten Anpassungen in den Partialen der Qualifikationen und Rollen machen. Ziel ist es, dass alle Partialen überarbeitet wurden und dem erstellten Mockup entsprechen.

9.8 Tag 8: 14.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Daily abhalten	Marc Egli	0.25	0.25
Turbo Streams implementieren	Marc Egli	1.5	7.25
Globale Bedingungen bearbeiten	Marc Egli	4	0
Feature Tests schreiben	Marc Egli	1.75	0
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		7.75	7.75

Tabelle 9.8: Tätigkeiten Tag 8

Tagesablauf

Den Morgen begann ich wie geplant mit der Überarbeitung des Partial für die Qualifikationen. Danach fand das Daily statt, für welches ich wieder bestimmte Fragen notiert habe.

Die erste Frage befasste sich mit dem Dropdown Problem der Tags, welches ich gestern hatte. Die Dropdowns für die Auswahl von Tags oder Rollen wurde nicht korrekt auf meiner View dargestellt und ich konnte den Fehler schlichtweg nicht finden. Daniel Illi wies mich im Daily daraufhin, dass das Dropdown evtl. zuerst per Javascript registriert werden muss. Daniel Illi zeigte mir ausserdem, in welcher Datei ich suchen muss.

Danach stellte ich eine weitere Frage zur Inklusion von Stylesheets. Ich hatte für die Filter ein eigenes Stylesheet unter dem Ordner `/components` angelegt, jedoch wurden meine Styles nicht geladen. Daniel Illi antwortete mir, dass die Stylesheets in der Datei `_main.scss` importiert werden müssen, damit auf die CSS-Klassen zugegriffen werden kann. Zum Schluss schlug ich die gestern dokumentierte Änderung des Mockups für das Filterkriterium "Felder" vor. Daniel Illi war mit der Änderung einverstanden. Somit werde ich die alte Benutzerschnittstelle für die Felder übernehmen und nur noch richtig anordnen.

Nach dem Daily, folgte ich der Anweisung von Daniel Illi und begann die Datei zu untersuchen, welche das Dropdown der Rollen und Tagfilter im JavaScript registriert. Ich erkannte hierbei, dass auf dem sogenannten Tom-Select diese JavaScript Aktion stets ausgeführt wurde. Für Turbo Aktionen wurde dabei mit einem Hook gearbeitet, welcher "turbo:load"

heisst. Das Problem: In meinem Fall werden die Tom-Selects mit einem Turbostream hinzugefügt, welcher nicht von diesem Hook abgefangen wird. Nach Recherche in der Dokumentation von Turbo und dem Abfangen von Events: <https://turbo.hotwired.dev/reference/events> wusste ich nun auch, dass es kein Event gibt, welches erst nach dem Rendern eines Turbostreams ausgelöst wird. Vor dem Rendern hätte ich auf den Hook “turbo:before-stream-render” brauchen können, allerdings bringt mir das nichts, wenn das Tom-Select noch nicht gerendert wurde. Schliesslich muss ich die JavaScript Aktion auf dem gerenderten Element ausführen.

Als Alternative habe ich mir folgendes überlegt: Theoretisch müsste es doch möglich sein einen Stimulus Controller zu implementieren, der nur die `connect()` Methode besitzt (Methode welche ausgeführt wird, wenn dieser Controller registriert wird). Es wäre dann möglich diesen Controller immer auf dem jeweiligen Tom-Select zu registrieren. Wird dann der Turbostream ausgeführt und fügt der Seite das Tom-Select hinzu, wird gleichzeitig der Stimulus Controller auf dem Tom-Select registriert. Danach wird die `connect()` Methode des Controllers ausgeführt und dementsprechend die JavaScript Aktion auf dem Element. Nach dieser Überlegung, habe ich diesen Stimulus-Controller erstellt und es hat tatsächlich wie angedacht funktioniert.

Nach dem Beheben dieses Problems, arbeite ich weiter an der Darstellung und Bearbeitung der Filterkriterien durch Turbostreams. Ich bemerkte zwischendurch, dass ich den Zeitplan massiv übertrete. Im Moment war dies aber irrelevant, da die Funktionalitäten an welchen ich arbeite so oder so implementiert werden mussten.

Ein weiteres Problem hatte ich mit der Darstellung der Rollen. Wie soll ich sie dem Benutzer anzeigen? Zuerst wollte ich alle Rollen zum zusammengesetzten String aus ihren Ebenen, Gruppen und der Rolle selbst mappen. Allerdings hätte das viel zu viel Zeit gebraucht, die Performance hätte aufgrund der vielen Rollen in Hitobito massiv von dieser Entscheidung gelitten. Deswegen zeigte ich dem Benutzer vorerst nur das Label der Rolle an, um nicht zu viel Ladezeit zu benötigen.

Gegen das Ende des Tages, überprüfte ich den heutigen Stand der Implementation nach dem Beispiel von gestern mit meinem Anforderungskatalog. Bis jetzt sind weiterhin alle Anforderungen erfüllt und ich bin auf einem guten Weg. Wenn ich am kommenden Dienstag die restlichen Funktionalitäten in den globalen Bedingungen der Abonnemente festhalte, wäre dann sogar die funktionale Anforderung 1 erfüllt und alle Grundfunktionalitäten vor dem Umbau, wären wieder vorhanden. Zudem konnte ich mit der heutigen Arbeit die funktionale Anforderung 3 erfüllen. Die Filterkriterien sind nun bearbeitbar.

Hilfestellungen

- Daniel Illi und Robin Steiner: Hinweise zur JavaScript Action und der dazugehörigen Datei
- Daniel Illi: Hinweis wie das Stylesheet in Hitobito geladen werden kann
- Daniel Illi und Robin Steiner: Besprechung der Mockup-Anpassung und Absegnung durch Daniel Illi

Reflexion

Was funktionierte gut

Der Stand des Projektes ist definitiv etwas Positives vom heutigen Tag. Die Grundfunktionalitäten sind schon fast wieder vorhanden und die Implementation ist fast deckungsgleich mit dem Mockup (ausgenommen der dokumentierten Änderungen daran). Zusätzlich konnte ich mit der Hilfe von Daniel Illi meine gestrigen Probleme beheben, was mir half, heute umso schneller vorwärts zu kommen.

Was funktionierte weniger gut

Auch wenn der Stand des Projektes gut ist, darf ich dennoch den Stand der gesamten IPA nicht vergessen. Aufgrund des viel grösseren Aufwands, bin ich mit dem Zeitplan nicht auf Kurs. Die ganzen Partials zu überarbeiten und die Turbostreams dazu zu erstellen waren definitiv sehr viel mehr Arbeit, als ich zuerst gedacht habe.

Meine heutigen Erkenntnisse

Wichtig ist es nun, dass ich versuche, so viel Zeit wie möglich zu sparen. Die globalen Bedingunge müssen noch angepasst werden und zusätzlich sollten noch Featuretests geschrieben werden. Meine Erkenntnis ist, diese Aufgaben spetitiver anzugehen, um dort die aufgebrauchte Zeit von heute wieder aufzuholen. Wenn dies nicht funktioniert, werde ich zusätzliche Überstunden einplanen. Momentan habe ich den Zeitplan wie von Lorenz Müller gewünscht auf 80 Stunden eingplant. Durch den Spielraum den mir die Anpassung von 90 Stunden auf 80 Stunden verschafft, kann ich nun 10

weitere Stunden planen (natürlich nur falls nötig). Dementsprechend froh bin ich, dass mir mein Hauptexperte zu dieser Änderung geraten hat, da ich von dieser jetzt im Notfall nur profitieren kann.

Nächste Schritte

Am Dienstag werde ich zuerst die letzten Anpassungen an Turbostreams vornehmen und diese dokumentieren. Danach werde ich direkt mit den globalen Bedingungen starten und anschliessend die Featuretests und die manuellen Tests für mein Produkt durchführen.

9.9 Tag 9: 18.03

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Turbostreams implementieren	Marc Egli	0	0.75
Daily abhalten	Marc Egli	0.25	0.25
Globale Bedingungen bearbeiten	Marc Egli	0	1
Feature Tests schreiben	Marc Egli	3.25	4
Manuelle Tests durchführen	Marc Egli	1	1
KI-Dokumentation schreiben	Marc Egli	2	0.5
Instruktion schreiben und durchführen	Marc Egli	1	0
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		7.75	7.75

Tabelle 9.9: Tätigkeiten Tag 9

Tagesablauf

Heute begann ich mit den letzten Anpassungen an den Turbostreams. Diese waren recht schnell erledigt und ich konnte alle restlichen funktionalen und nicht funktionalen Anforderungen meines Anforderungskatalogs abdecken. Dann starteten wir wie immer mit dem Daily um 09:00 Uhr. Im Daily sprach ich zu Beginn eine Anpassung am Mockup an, welche mir im Schlaf über das Wochenende zugekommen ist. Es wäre für den Benutzer doch viel sinnvoller, wenn er die Eingaben direkt im Formular machen könnte, ohne zuerst auf ein Bearbeiten-Symbol zu klicken. So müsste er nur auf die Benutzerschnittstelle der Filterung navigieren und anschliessend die neuen Werte eingeben. Daniel Illi meinte zu der Idee, dass er sie als Stakeholder so akzeptiert. Des Weiteren habe ich mit Daniel Illi einen Termin für das Durchführen der Instruktion vereinbart.

Nach dem Daily begann ich mit dem Bearbeiten der globalen Bedingungen. Ich bekam diese schnell durch und startete direkt mit den Featuretests. Während dem Schreiben der Tests, hatte ich stets mein Testkonzept zur Hand. Bei dem Testfall Nr. 17 merkte ich, dass dieser nicht nötig ist, da bereits der Testfall Nr. 12 den Zugriff des Benutzers auf die Personenfilterung abdeckt. Falls der Zugriff nicht gegeben ist, kann der Benutzer auch den Personenfilter in der Benutzerschnittstelle nicht einsehen. Dementsprechend habe ich den Testfall verworfen und dies im Testprotokoll so dokumentiert.

Später kam bei mir dann noch eine Frage bezüglich dem Stubben von Umgebungsvariablen auf. Ein Fehleralert wird im Hitobito nur angezeigt, wenn die Umgebung auf Produktion gestellt ist. Da mir nicht klar war, wie ich diese Umgebungsvariable stubben kann, schrieb ich Daniel Illi auf dem Puzzle Chat und bat ihn um Hilfe. Da er im Moment noch ein Meeting hatte, schrieb er mir, dass er später persönlich vorbei komme. Mit Daniel Illi konnte ich das Problem anschliessend lösen und die restlichen Tests implementieren.

Zuletzt führte ich die manuellen Tests durch und begann mit der Dokumentation der Verwendung von KI in dieser IPA.

Hilfestellungen

- Daniel Illi und Robin Steiner: Mockup Änderung und Terminvereinbarung Instruktion
- Daniel Illi: Stubbing von Umgebungsvariablen

Reflexion

Was funktionierte gut

Trotz des Verzuges im Zeitplan, konnte ich die Implementation meiner IPA abschliessen. Der Code steht und funktioniert auch, was durch meine Tests bewiesen wird. Dies erleichtert mich sehr, da ich von diesem Teil der IPA am meisten Respekt hatte. Zudem bin ich im Zeitplan wieder auf Kurs und denke, dass ich das Ende wie geplant einhalten kann.

Was funktionierte weniger gut

Ein Teil von mir hatte gehofft, mit den Featuretests schnell fertig zu werden, um noch mehr Zeit zu gewinnen. Allerdings, hatte ich während des Schreibens der Tests immer wieder Konzentrationsprobleme. Das Testen ist für mich schlichtweg nicht der schöne Teil der Entwicklerarbeit. Umso erleichteter bin ich nun, da alle Testfälle nach Testkonzept umgesetzt werden konnten.

Meine heutigen Erkenntnisse

Wie ich realisiert habe, wurde der Zeitblock für die User-Story “Turbostreams implementieren” im Zeitplan massiv zu gross. Im Nachhinein, wäre es besser gewesen, ich hätte diese User-Story in weitere, kleinere User Stories unterteilt. Zum Beispiel hätte ich eine User-Story für die Turbostreams und dann für jede Überarbeitung der Benutzerschnittstelle eines Filterkriteriums, eine User Story machen können. Meine Erkenntnis dadurch ist, nicht probieren allzu viele Aufgaben in einem Ticket zusammenzufassen. Besser ist es, kleine Blöcke zu machen. Das macht die Abnahme der Akzeptanzkriterien im Sprint Review auch einfacher.

Nächste Schritte

Morgen werde ich zuerst die Dokumentation zur Verwendung von KI in meiner IPA fertigstellen. Danach bereite ich die Instruktion vor. Wenn beides erledigt ist, mache ich den Sprintabschluss für die Umsetzungsphase. Ich werde danach das Sprintplanning für den letzten Sprint durchführen und alle Tickets für die Detailarbeiten an der Dokumentation verfassen. Das Ziel für morgen ist es, den Umsetzungssprint abzuschliessen und alles bereit zu haben für den letzten Sprint der Finalisierung.

9.10 Tag 10: 19.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Daily abhalten	Marc Egli	0.25	0.25
KI-Dokumentation schreiben	Marc Egli	0	1.5
Instruktion schreiben und durchführen	Marc Egli	2	1.5
Sprintabschluss machen	Marc Egli	1	0.75
Sprint Planning durchführen	Marc Egli	1	1
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		4.5	5.25

Tabelle 9.10: Tätigkeiten Tag 10

Tagesablauf

Heute startete ich den morgen mit dem Abschluss der KI-Dokumentation. Danach bereitete ich die Instruktion für Donnerstag vor. Im Daily heute hatte ich keine Fragen, alles geht nun gegen ein Ende zu und es geht darum noch die letzten Feinschliffe an der Arbeit zu machen. Nach dem Daily und dem Vorbereiten der Instruktion habe ich den Sprintabschluss gemacht. Hier habe ich gemerkt, dass ich sehr knapp in der Zeit für den heutigen Tag bin. Um gleichwohl, das gestrige Ziel zu erreichen, habe ich mich dazu entschieden, das Ziel von heute zu erfüllen und dafür etwas von meiner Restzeit zu verwenden. Somit kann ich morgen dann optimal in den Finalisationssprint starten. Nach dem Planning habe ich abschliessend das Arbeitsjournal geschrieben.

Hilfestellungen

- Keine

Reflexion

Was funktionierte gut

Obwohl ich mehr Zeit dafür benötigte konnte ich das gestern gesetzte Ziel erreichen. Der Umsetzungssprint ist abgeschlossen und alle User Stories sind

bereite um im letzten Sprint bearbeitet zu werden.

Was funktionierte weniger gut

Heute habe ich keinen Punkt den ich hier aufführen könnte. Auch wenn ich mehr Zeit benötigte, sehe ich dies nicht als direkten Rückschlag im Projekt, da ich dafür die nötige Zeit eingeplant habe.

Meine heutigen Erkenntnisse

Nicht gegen Ende schlapp machen. Am Freitagmorgen werde ich die Arbeit abgeben, darum jetzt noch Endspurt und Zähne zusammenbeissen um die Arbeit fertigzubringen.

Nächste Schritte

Morgen startet der Finalisationssprint. Wichtig ist, dass ich hier speditif durchgehe und Fehler in der Dokumentation korrigieren kann. Ausserdem werde ich die Instruktion mit Daniel Illi durchführen, was es mir ermöglicht den letzten Teil der Dokumentation abzuschliessen und das Kriterium A15 zu erfüllen.

9.11 Tag 11: 21.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Kurzbericht der IPA verfassen	Marc Egli	1	1
Dokumentation gegenlesen und korrigieren	Marc Egli	4	4
Glossar und Abkürzungen definieren	Marc Egli	1	1
Code Anhang generieren	Marc Egli	1	1
Instrukton finalisieren	Marc Egli	1	1
Arbeitsjournal schreiben	Marc Egli	0.25	0.25
Total		8.25	8.25

Tabelle 9.11: Tätigkeiten Tag 11

Tagesablauf

Heute ging es an die Finalisierung der Arbeit. Ich verfasste als erstes den Kurzbericht der IPA und begann danach die Dokumentation gegenzulesen. Danach erstellte ich das Glossar mit allen Abkürzungen und Definitionen. Im Daily hatte ich keine besonderen Fragen an meine verantwortlichen Fachkräfte, ich präsentierte lediglich den Stand der Arbeit.

Gegen den Mittag führten Daniel Illi und ich dann die Instruktion durch. Dies sehr gut und ich konnte mit der Instruktion nachweisen, dass die Benutzerschnittstelle intuitiv erstellt wurde. Falls der Benutzer nicht weiter weiss, hat sich die erstellte Instruktion als praktisch und um Erfolg führend erwiesen. Am Ende des Tags wertete ich die Ergebnisse der Instruktion aus. Danach schrieb ich das Arbeitsjournal.

Hilfestellungen

- Keine

Reflexion

Was funktionierte gut

Heute kam ich schnell voran und konnte viele Fehler in der Dokumentation beheben. Zusätzlich sind nun praktisch alle Kapitel darin beschrieben, nur noch der letzte Sprintabschluss muss gemacht werden, danach ist alles fertig. Ich war während des Tages immer wieder ein bisschen nervös, da die Abgabe nun schon vor der Tür steht. Ich denke aber, dass ich nun nur noch einen Schlussprint hinlegen muss und dann alles gut kommt.

Was funktionierte weniger gut

Alles funktionierte heute gut, es gab keine unvorhergesehenen Herausforderungen oder Problem.

Meine heutigen Erkenntnisse

Ich muss unbedingt viel Zeit in das Gegenlesen stecken und absichern, dass alles perfekt beschrieben ist. Es ist mir sehr wichtig das diese Arbeit gut herauskommt, dementsprechend muss ich versuchen die Dokumentation fehlerlos zu gestalten.

Nächste Schritte

Morgen werde ich den Zeitplan fertig finalisieren, den Sprintabschluss machen und eine frühe Abgabe wagen. Dies aus dem Grund, das wenn etwas schief geht, gleichwohl schon eine Version meiner Dokumentation im PKORG abgelegt ist. Den Rest der Zeit werde ich mit Korrekturen an der Dokumentation verbringen, bis am Mittag die Durchführung der IPA endet.

9.12 Tag 12: 21.03.2025

Tätigkeiten	Beteiligte Personen	Aufwand geplant (std)	Aufwand effektiv (std)
Daily abhalten	Marc Egli	0.25	0.25
Instruktion anpassen	Marc Egli 1	1	
Zeitplan finalisieren	Marc Egli	1	1
Abgabe und letzte Änderungen machen	Marc Egli	2	2
Arbeitsjournal schreiben			
Total		4.25	4.25

Tabelle 9.12: Tätigkeiten Tag 12

Tagesablauf

Heute war ein stressiger Tag. Die ganze Abgabe und Korrektur machte mich nervöser als gedacht. Ich begann den Tag damit, letzte Anpassungen an der Instruktion zu machen. Danach finalisierte ich den Zeitplan und machte mich an die letzten Änderungen. Stand jetzt, sind noch zwei Stunden bis zur Abgabe vorhanden. Ich werde in diesen Stunden so viel korrigieren wie möglich und das ganz Dokument so gut es geht auf Fehler prüfen. Im Daily heute fragte ich meine verantwortlichen Fachkräfte, ob ich den Code im Anhang formatieren müsste. Sie meinten an sich nicht, wenn es jedoch nicht zu viel Aufwand macht dann schon. Ich werde den Code in den nächsten Stunden so gut formatieren wie es die Zeit zulässt.

Nachdem ich dieses Journal fertiggeschrieben habe, werde ich die erste Abgabe auf das PKORG laden. So verhindere ich, dass bei einem Notfall, eine verspätete Abgabe entsteht. Diesen Abzug gilt es unbedingt zu verhindern.

Hilfestellungen

- Daniel Illi und Robin Steiner: Nachfrage zur Formatierung von Code im Anhang

Reflexion

Was funktionierte gut

Ich konnte die Arbeit finalisieren und bin insgesamt sehr zufrieden damit. Ich habe alle nötigen Informationen im Bericht dokumentiert und freue mich nun auch ein bisschen die Arbeit endlich abgeben zu können.

Was funktionierte weniger gut

Nebst dem Stress und dem Druck von heute, lief alles tiptop. Allerdings habe ich das Gefühl, das mich genau dieser Stress besser abliefern lässt, ich durch erhöhte Motivation und schnelleres Arbeiten bemerken konnte.

Meine heutigen Erkenntnisse

Meine Erkenntnis von heute: Die IPA war eine sehr aufregende, stressige aber auch lehrreiche Erfahrung. Ich konnte meine Arbeit nun abschliessen und werde den Freitagnachmittag geniessen.

Nächste Schritte

Die Abgabe werde ich voraussichtlich in zwei oder eineinhalb Stunden signieren. Danach, gilt es in den kommenden Wochen die Präsentation und das Fachgespräch vorzubereiten.

10 Persönliches Fazit

Eine Analyse nach Beendung der Arbeit an der IPA.

10.1 Was funktionierte weniger gut

Die Dokumentationsaufträge nahmen insgesamt mehr Zeit in Anspruch als gedacht. Vielfach war das auf Probleme mit meiner Konzentration zurückzuführen. Beim Dokumentieren muss viel hinterfragt und recherchiert werden, so kam ich nicht wie beim Programmieren in den “Tunnel” wo ich mich nur noch aufs Machen konzentrieren konnte. Obwohl ich dadurch Zeit beim Dokumentieren verlor, war ich so dennoch sicher das meine Informationen korrekt sind und ich auf einem guten Weg bin.

10.2 Was funktionierte gut

Während der Arbeit hatte ich stets ein gutes Zeitmanagement. Ich wusste, wann ich Überstunden machen musste oder wann ich die Zeit wieder aufholen kann. So konnte ich beruhigt in den letzten Tag starten, da ich nur noch die Dokumentation erneut gegenlesen musste.

Weiter zur Umsetzung. Für mich war es das klare Highlight in der Arbeit. Die Überarbeitung der Benutzerschnittstellen, war sehr spannend, vor allem da ich mein Wissen um Hotwire für ein Feature einsetzen konnte, dass von Benutzern verwendet wird. Vor der Umsetzungs, war ich allerdings sehr nervös. Ich dachte, dass ich die Umetzung evtl. nicht wie geplant durchführen könnte und dass es sicher viel zu viel Aufwand für die Überarbeitung der Benutzerschnittstelle ist. Während des Entwickelns kam ich aber dann sehr schnell in diesen “Tunnel” und konnte mich nur noch aufs Programmieren konzentrieren. Schon nach dem ersten Tag erkannte ich, dass ich es schaffen kann und war deswegen zusätzlich motiviert.

Nach der Umsetzung kann ich sagen, dass ich sehr froh bin Scrum verwendet zu haben. Diese Arbeits- und Projektmethodik ist sehr nahe an meinem sonstigen Arbeitsalltag und ich musste dadurch nicht eine neue

Projektmethode kennenlernen. Ausserdem ermöglichte Scrum es mir, nach jedem Sprint den Stand meines Projektes zu prüfen und so eine Einschätzung meiner momentanen Lage zu treffen. Da ich insgesamt zwei User Stories während der IPA beim Sprintabschluss zurück ins Refinement schieben musste, war ich um die Agilität welche Scrum mir gab sehr erleichtert.

10.3 Schlussreflexion

Für mich lässt sich die IPA in drei Wörtern beschreiben: Spannend, lehrreich und stressig. Ich freue mich jetzt schon auf die Abgabe, da ich dann bestimmt erstmals wieder zurücklehen und mich für eine kurze Zeit erholen kann. Allerdings bin ich aber auch sehr positiv überrascht, wie gut die Arbeit verlief. Zudem kann ich mit vollstem Gewissen behaupten, mein Können und die Fähigkeiten welche ich mir in den letzten vier Jahren angeeignet habe unter Beweis gestellt zu haben. Selbstverständlich bin ich aber dennoch sehr auf die Bewertung dieser Arbeit und hoffe das ich das Feedback zu dieser in mein weiteres Berufsleben weitertragen kann.

Mit dem Ergebnis bin ich sehr zufrieden. Ich bin der Meinung, alles was geht rausgeholt zu haben und kann die Arbeit mit einem guten Gewissen abgeben.

Teil II

Projektdokumentation

Hitobito: Neue Generation von Personen-Filtern

Autor: Marc Egli

11 Kurzfassung des IPA-Berichtes

11.1 Ausgangssituation

Hitobito ist eine Open-Source-Software zur Verwaltung von Mitgliedern und Communities. Sie ermöglicht es, komplexe Gruppenstrukturen individuell nach den Bedürfnissen des Nutzers abzubilden. Mit Hitobito lassen sich Veranstaltungen, Kurse, Rollen sowie Mailinglisten effizient organisieren. Technisch basiert Hitobito auf einer Wagon-Core-Architektur. Das bedeutet, dass es ein zentrales Core-Projekt gibt, das durch sogenannte Wagons erweitert werden kann. Diese Wagons dienen dazu, Funktionen zu ergänzen, anzupassen oder spezifisch für einzelne Kunden zu gestalten. Auch die jeweilige Gruppenstruktur wird im kundenspezifischen Wagon definiert. Hitobito bietet für die Konfiguration der Filterung von Personen und Abonnenten eine Benutzerschnittstelle an. Diese Benutzerschnittstelle soll im Rahmen dieser Arbeit verbessert werden.

11.2 Umsetzung

Auf Grundlage dieser Anforderung wurde ein Konzept entwickelt, welches die Überarbeitung dieser Benutzerschnittstelle behandelt. Der darauf basierende Code wurde durch verschiedene automatisierte Testverfahren überprüft und abgesichert. Zusätzlich werden alle funktionalen Anforderungen sowohl während als auch nach der Entwicklung manuell getestet. Alle Codeänderungen erfolgen ausschließlich im Core-Projekt. Für die Entwicklung kommt dabei der Generic Wagon zum Einsatz.

11.3 Ergebnis

Das Konzept konnte grösstenteils wie geplant durchgeführt werden. Alle Änderungen daran wurden im Arbeitsjournal sowie in der Projektdokumentation beschrieben.

12 Einführung

Puzzle ITC ist ein schweizer Anbieter für Softwarelösungen. Die Firma hat ihren Hauptsitz in Bern, besitzt aber weitere Standorte in Zürich, Luzern und Deutschland (Thüringen). Puzzle bietet als Unternehmen die ganze Palette an IT-Services an, von Digital Transformation bis hin zu Data Analytics. Nebst den vielen Angeboten tritt Puzzle dabei immer seine Grundwerte nach aussen, welche im Puzzlehouse abgebildet werden.



Abbildung 12.1: Puzzle House

Hitobito ist eines der Angebote von Puzzle. Es ist ein Community-Management Tool und als Open-Source Projekt auf Github zu finden. Das Tool wird von zahlreichen Verbänden, Parteien und Organisationen verwendet und befindet sich darum in einer kontinuierlichen Weiterentwicklung. Mit dem Wagons-Gem ermöglicht es Hitobito zudem spezielle Kundenanpassungen in einem eigenen "Wagon" zu vollziehen, ohne die Software anderer Kunden mit-anzupassen.

Ich selbst arbeite jetzt seit einem halben Jahr im Hitobito und nahm darin vor allem Upgrades und Migrationen vor. So durfte der Kandidat bspw. das Upgrade von RoR (Ruby on Rails) von 6.1 auf 7.1 vornehmen oder die Migration von MySQL auf PostgreSQL vollziehen.

Da Hitobito von zahlreichen Kunden verwendet wird, ist die Applikation über die Jahre gewachsen. Viele Features wurden implementiert, um sie schnell dem Kunden zur Verfügung zu stellen. Mit einem immer wachsenden Anforderungskatalog, ergaben sich dadurch komplexe Arbeitsabläufe, welche im Tool etabliert wurden. Einer dieser komplexen Abläufe, ist die Filterung nach Personen oder Abonnementen.

Mit dieser IPA eine neue Benutzerschnittstelle entworfen werden eine besser User Experience für den Benutzer zu schaffen.

In einer Zeit, in welcher Unternehmen mehr den je Wert auf ein sauberes Design und der User Experience von Webseiten und Applikationen geben, das auch in einer älteren Applikation zu etablieren. Gerade bei einem Community-Management Tool wie Hitobito, welches tagtäglich von Personen bedient wird, welche nicht das technische Know-How dahinter besitzen, ist es wichtig, Arbeitsabläufe so einfach wie möglich zu entwerfen, um maximale Effizienz für diese Personen zu garantieren. Durch eine Vereinfachung der Hitobito-Filter, machen wir damit einen ersten Schritt in die richtige Richtung.

13 Analyse

In der Analyse der IPA wird der Rahmen geschaffen, in welchem der Kandidat später während des Implementierens arbeitet. Sie befasst sich mit der Aufnahme von Ist- und Zielzustand des Produktes. Ausserdem werden darin funktionale und nicht funktionale Anforderungen erfasst. Es wird definiert, wo sich die IPA abgrenzt.

13.1 Ist-Zustand

Im folgenden Abschnitt wird erklärt, wie die Filterung von Personen und Abonnenten im Hitobito zum jetzigen Stand abläuft.

13.1.1 Personen

Die Personenfilterung bietet zwei Funktionen: Die Filterung selbst und die Speicherung eines Personenfilters. Um auf die Personenfilter zugreifen zu können, muss als erstes auf eine Gruppenübersicht navigiert werden.

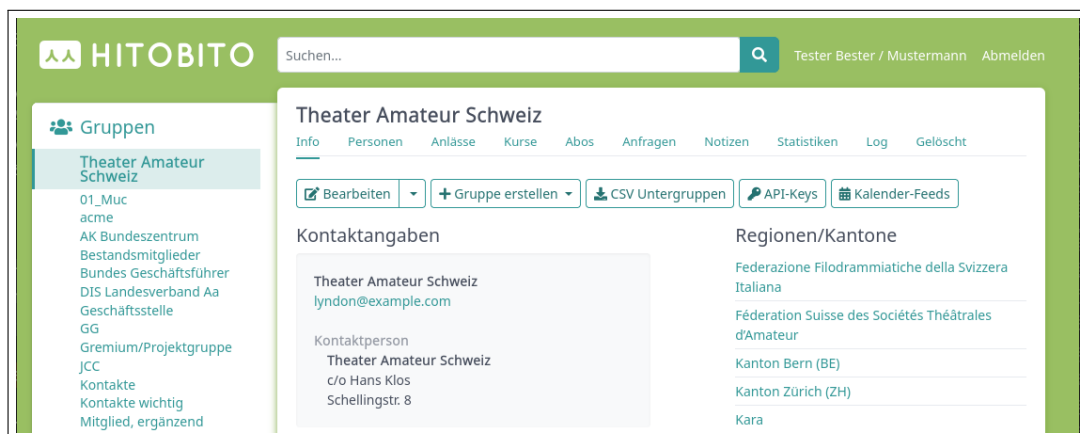


Abbildung 13.1: Gruppenübersicht Hitobito

Danach werden unter dem Reiter “Personen” alle Personen der ausgewählten Gruppe angezeigt.

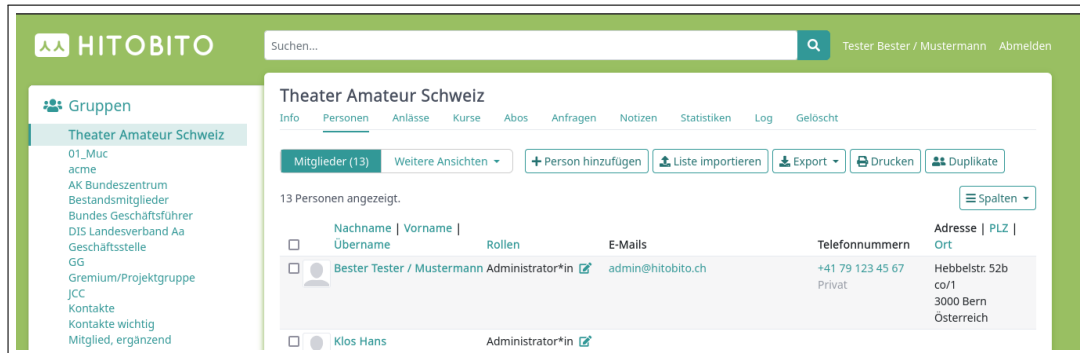


Abbildung 13.2: Personenübersicht Hitobito

Unter dem Dropdown “Weitere Ansichten” kann der Benutzer entweder einen bestehenden Personenfilter auf die angezeigte Personenliste anwenden oder einen neuen Personenfilter erstellen. Der neue Personenfilter kann unter der Option “Neuer Filter...” erstellt werden.



Abbildung 13.3: Filterauswahl Hitobito

Nach dem Klick auf diese Option, wird der Benutzer auf eine Filterübersicht weitergeleitet. Die Filterübersicht besteht aus Buttons für die Speicher und Suche, drei Radio-Buttons und fünf Dropdowns.

Die drei Radio-Buttons am Anfang, definieren auf welcher Ebene gesucht wird. Jedes Dropdown bietet dem Benutzer die Möglichkeit, Filterkriterien zu definieren. Die Art des Filterkriteriums ist durch den Dropdownnamen gegeben.

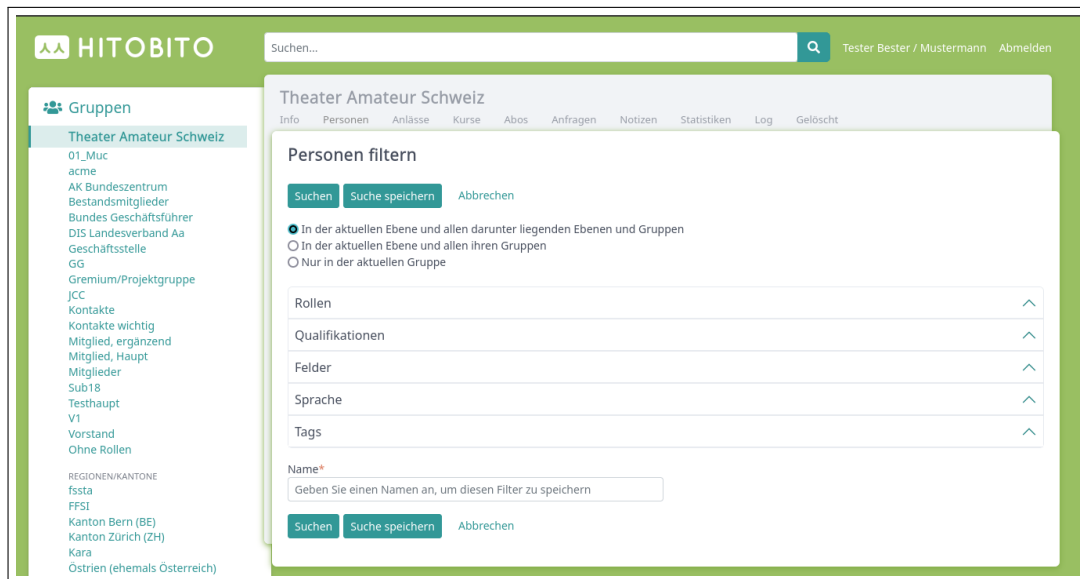


Abbildung 13.4: Filterübersicht Hitobito

Im Dropdown “Rollen” werden alle Rollen pro Ebene aufgelistet. Der Benutzer kann per Klick auf eine Checkbox die Rollen auswählen, welche Personen durch die Filterung aufweisen müssen. Möchte der Benutzer alle Rollen einer Ebene auswählen, kann er dies mit einem Klick auf den Namen einer Ebene tun. In diesem Beispiel wäre das ein Klick auf “Hauptebene”. Nach dem Klick auf den Ebenennamen werden alle Checkboxes abgehakt. Somit wurden alle Rollen der angeklickten Eben ausgewählt.

Des weiteren kann für die Rollen ein Gültigkeits-Zeitraum definiert werden. Die Gültigkeit wird vom Benutzer via Radio-Buttons ausgewählt. Zuletzt kann im “Rollen” Dropdown entschieden werden, ob archivierte Rollen bei der Filterung berücksichtigt werden. Das UI bietet dem Benutzer dafür eine weitere selbststehende Checkbox.

Abbildung 13.5: Filterkriterium Rollen

Abbildung 13.6: Filterkriterium Rollen - Zeitraum und Archivierung

Im Dropdown “Qualifikationen” werden alle auswählbaren Qualifikationen angezeigt. Der Benutzer wählt alle gewünschten Qualifikationen per Checkbox aus. Per Radio-Button entscheidet der Benutzer danach, ob die Person alle oder mindestens eine der angeklickten Rollen aufweisen muss.

Abschliessen für die Qualifikationen kann die Gültigkeit per Radio-Buttons und ein Stichdatum per Input-Feld definiert werden.

Abbildung 13.7: Filterkriterium Qualifikationen

Beim Filterkriterium “Felder” handelt es sich um zusätzliche Personenattribute, welche in die Filterung miteinbezogen werden.

Abbildung 13.8: Filterkriterium Felder

Unter die Kategorie der “Felder” fallen folgende Personenattribute:

Attribut	Datentyp
Alter	Zahl
Firmenname	Text
Geburtstag	Zeit
Geschlecht	Text
Haupt-E-Mail	Text
Hausnummer	Zahl
Land	Text
Nachname	Text
Ort	Text
PLZ	Zahl
Postfach	Text
Strasse	Textwert
Vornamen	Textwert
Zusätzliche Adresszeile	Textwert
Übername	Textwert

Tabelle 13.1: Felder-Attribute

Die Attribute können über ein Dropdown ausgewählt werden. Wurde das Attribut ausgewählt kann zusätzlich die Genauigkeit des Filterkriteriums definiert werden.

Für Textwerte bieten sich folgende Genauigkeiten an:

- ist genau
- enthält
- enthält nicht

Für Zahlenwerte bieten sich folgende Genauigkeiten an:

- ist genau
- ist grösser als
- ist kleiner als

So kann der Benutzer bestimmen wie genau sein eingegebener Wert dem ausgewählten Attribut entsprechen soll.

Das Filterkriterium “Sprachen” bietet die Möglichkeit aus den Optionen Deutsch, Englisch oder Französisch auszuwählen. Es können auch mehrere Sprachen ausgewählt werden. Die Auswahl erfolgt über Checkboxes.

The screenshot shows the 'Personen filtern' (Filter people) interface. On the left is a sidebar with a tree structure under 'Theater Amateur Schweiz'. The main panel has a title 'Personen filtern' and three buttons: 'Suchen' (Search), 'Suche speichern' (Save search), and 'Abbrechen' (Cancel). Below the buttons are three radio buttons for the search scope: 'In der aktuellen Ebene und allen darunter liegenden Ebenen und Gruppen' (selected), 'In der aktuellen Ebene und allen ihren Gruppen', and 'Nur in der aktuellen Gruppe'. There are four filter criteria listed with expand/collapse arrows: 'Rollen', 'Qualifikationen', 'Felder', and 'Sprache'. The 'Sprache' criterion is expanded, showing three checkboxes: 'Deutsch', 'English', and 'Français', all of which are currently unchecked.

Abbildung 13.9: Filterkriterium Sprache

Das Filterkriterium “Tags” macht es möglich, Personen nach gegebenen Tags zu suchen. Über zwei Inputfelder kann definiert werden, welche Tags eine Person haben oder nicht haben muss. Die Input-Felder sind Dropdowns welche dem Benutzer die Eingabe vervollständigen, wenn dieser nach einem Tag sucht.

The screenshot shows the 'Personen filtern' (Filter people) interface with the 'Tags' criterion expanded. The sidebar on the left is the same as in the previous image. The main panel shows the same search scope radio buttons. The 'Tags' criterion is expanded, showing two input fields. The first field is labeled 'Personen einbeziehen, die mindestens einen der untenstehenden Tags enthalten' (Include people who contain at least one of the following tags) and contains a dropdown menu with 'Alt' selected and a text input field with 'CEWE-Demo' entered. The second field is labeled 'Personen ausschliessen, die mindestens einen der untenstehenden Tags enthalten' (Exclude people who contain at least one of the following tags) and contains a dropdown menu with 'Fußball' selected and a text input field with 'Gewünschte Tag-Namen eingeben' (Enter desired tag names) entered.

Abbildung 13.10: Filterkriterium Tags

Wurden alle Filterkriterien vom Benutzer definiert, kann dieser die Konfiguration des Personenfilters speichern. Dazu muss ein Name des Filters definiert werden. Will der Benutzer den Filter nicht speichern, kann der auf den Button “Suchen” klicken. Anschliessend werden auf der Personenübersicht nur Personen angezeigt, welche die Filterkriterien erfüllen. Wird der Filter nicht gespeichert, muss dieser erneut von Hand zusammengestellt werden.

The screenshot shows the 'Personen filtern' (Filter People) interface. On the left is a sidebar with a tree view of the organizational structure, including categories like 'Theater Amateur Schweiz', '01_Muc', 'acme', 'AK Bundeszentrum', 'Bestandsmitglieder', 'Bundes Geschäftsführer', 'DIS Landesverband Aa', 'Geschäftsstelle', 'GG', 'Gremium/Projektgruppe', 'JCC', 'Kontakte', 'Kontakte wichtig', 'Mitglied, ergänzend', 'Mitglied, Haupt', 'Mitglieder', 'Sub18', 'Testhaupt', 'V1', 'Vorstand', 'Ohne Rollen', 'REGIONEN/KANTONE', 'fssta', 'FFSI', 'Kanton Bern (BE)', 'Kanton Zürich (ZH)', 'Kara', and 'Östrien (ehemals Österreich)'. The main panel is titled 'Personen filtern' and contains three buttons: 'Suchen', 'Suche speichern', and 'Abbrechen'. Below these are three radio buttons: 'In der aktuellen Ebene und allen darunter liegenden Ebenen und Gruppen' (selected), 'In der aktuellen Ebene und allen ihren Gruppen', and 'Nur in der aktuellen Gruppe'. There are five dropdown menus for 'Rollen', 'Qualifikationen', 'Felder', 'Sprache', and 'Tags'. At the bottom, there is a 'Name*' field with the placeholder text 'NewFilter' and another set of 'Suchen', 'Suche speichern', and 'Abbrechen' buttons.

Abbildung 13.11: Filterkriterium Tags

13.1.2 Abonnente

Die Abonnentenfilterung kann nicht im Hitobito wie die Personenfilterung angezeigt werden. Stattdessen wird die Filterung bei einem Export der Abonnenten als CSV angewendet. So werden im resultierenden CSV nur Personen angezeigt, welche den Filterkriterien entsprechen.

Die Abonnentenfilterung kann erneut über die Gruppenübersicht ausgemacht werden.



Abbildung 13.12: Gruppenübersicht

Danach muss auf den Reiter “Abos” navigiert werden.

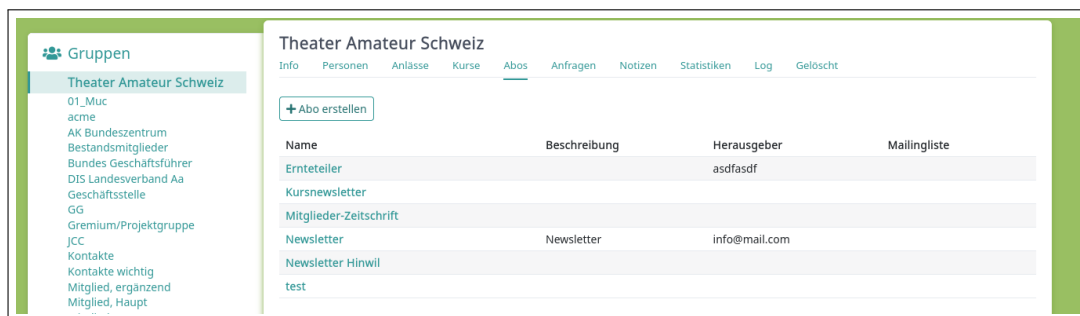


Abbildung 13.13: Abonnenten Übersicht

Auf dieser Ansicht kann der Benutzer per Klick ein Abonnement auswählen.

Daraufhin wird der Benutzer auf die Infoseite des Abonnements weitergeleitet. Um die Abonnentenfilterung einzusehen, muss nun auf den Reiter “Abonnenten” navigiert, werden.



Abbildung 13.14: Übersicht einzelnes Abonnementes

In dieser Ansicht ist es dem Benutzer möglich, diverse Filterkriterien für die Abonnenten zu definieren. Im Rahmen dieser IPA werden ausschliesslich die globalen Bedingungen überarbeitet, weswegen die anderen Filterkriterien zu vernachlässigen sind. Die globalen Bedingungen können über das Bearbeitungssymbol angepasst werden.

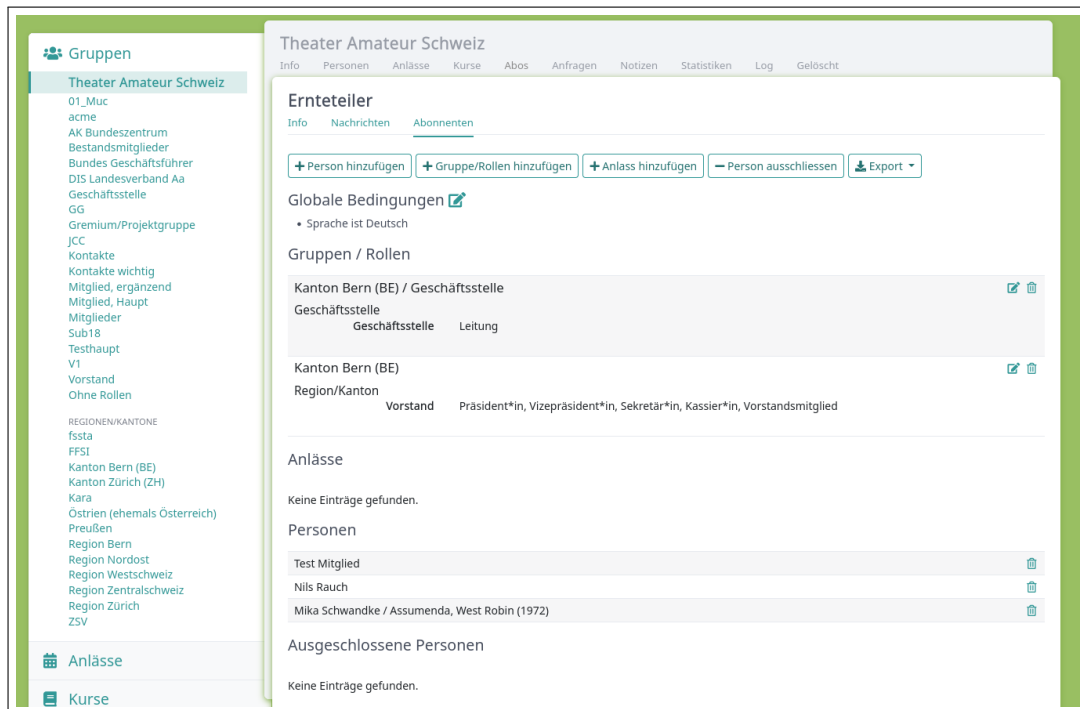


Abbildung 13.15: Abonnentenübersicht

In den globalen Bedingungen bieten sich zwei Filterkriterien an: Felder und Sprache. Die Funktionsweise ist dabei die gleiche, wie die der Personenfilter (weiter oben beschrieben).

Der Benutzer wählt die Filterkriterien aus und speichert diese anschliessend. Die gespeicherten Bedingungen werden in der Textbox unter dem Titel “Globale Bedingungen” auf der Abonnentenübersicht angezeigt.

The screenshot shows the 'Theater Amateur Schweiz' web application. On the left is a sidebar with a 'Gruppen' (Groups) menu. The main content area is titled 'Ernteteiler' and has tabs for 'Info', 'Nachrichten', and 'Abonnenten'. The 'Abonnenten' tab is active, showing a section titled 'Globale Bedingungen'. At the top of this section are two buttons: 'Speichern' (Save) and 'Abbrechen' (Cancel). Below this is a 'Felder' (Fields) section with a list of filter criteria: 'Alter' (Age), 'Firmenname' (Company Name), and an empty dropdown. Each criterion has a corresponding 'ist genau' (is exactly) dropdown and a text input field. To the right of each input field is a trash icon. Below the 'Felder' section is a 'Sprache' (Language) section with three radio buttons: 'Deutsch' (selected), 'English', and 'Français'. The sidebar on the left lists various groups under 'Theater Amateur Schweiz', including '01_Muc', 'acme', 'AK Bundeszentrum', 'Bestandsmitglieder', 'Bundes Geschäftsführer', 'DIS Landesverband Aa', 'Geschäftsstelle', 'GG', 'Gremium/Projektgruppe', 'JCC', 'Kontakte', 'Kontakte wichtig', 'Mitglied, ergänzend', 'Mitglied, Haupt', 'Mitglieder', 'Sub18', 'Testhaupt', 'V1', 'Vorstand', and 'Ohne Rollen'. At the bottom of the sidebar, it says 'REGIONEN/KANTONE', 'fssta', and 'EEET'.

Abbildung 13.16: Globale Filterungsbedingungen für Abonnenten

13.2 Soll-Zustand

In der Vorarbeit für diese IPA wurde vom Kandidaten ein Mockup erstellt. Das Produkt der IPA soll nach diesem Mockup umgesetzt werden. Das angefertigte Mockup repräsentiert den Soll-Zustand und wird im folgenden Abschnitt erklärt.

13.2.1 Anzeigemasken

Zuerst werden die Anzeigemasken des Mockups vorgestellt. Dabei handelt es sich um Anzeigen, welche der Benutzer ausschliesslich lesen und nicht bearbeiten kann.

Filterübersicht

Sobald der Benutzer auf “Neuer Filter...” in der Personenübersicht einer Gruppe klickt, soll er auf folgende Ansicht weitergeleitet werden:

Filter-Kriterien - Show Maske - Keine Filter definiert

☐ In der aktuellen Ebene und allen darunter liegenden Ebenen und Gruppen
☐ In der aktuellen Ebene und allen ihren Gruppen
☐ Nur in der aktuellen Gruppe

Um einen Filter auszuwählen, bitte klicke auf den hinzufügen Button

Name:

Attribut
Gruppen
Qualifikationen
Tags

Abbildung 13.17: Mockup Filterübersicht

Die Übersicht enthält die Radio-Buttons der alten Benutzerschnittstelle. Die

massgebliche Änderung zum Soll-Zustand, betrifft den Hinzufügen-Button. Dieser agiert als Dropdown, sobald der Benutzer auf ihn klickt. Das Dropdown besitzt vier Filterungskriterien, nicht mehr die fünf, wie sie momentan im Hitobito anzutreffen sind. Sind noch keine Filterkriterien erfasst, wird dies dem Benutzer mit einem Text signalisiert. Die Speicherkomponente des Filters bleibt bestehen.

Filterübersicht mit Tags und Felder

Hat der Benutzer seine Filterkriterien definiert, werden diese in Boxen angezeigt. Die definierten Filterkriterien verschwinden aus dem Dropdown unter dem Hinzufügen-Button. Stattdessen können die definierten Filterkriterien über den Stift in der oberen, rechten Ecke einer Box, bearbeitete werden.

Filter-Kriterien - Show Maske - Tags und Felder definiert

☐ In der aktuellen Ebene und allen darunter liegenden Ebenen und Gruppen
☐ In der aktuellen Ebene und allen ihren Gruppen
☐ Nur in der aktuellen Gruppe

Tags

Einschliessen UM-ID-179513 UM-ID-189613

Ausschliessen UM-ID-179516

Felder

Attributname Alter

Wert: <= 42

Attributname

Wert: genau 42

Name:

Gruppen Qualifikationen

Abbildung 13.18: Mockup mit Tags und Felder definiert

In dieser Ansicht hat der Benutzer das Filterkriterium “Tags” und “Felder” definiert. Diese Optionen verschwinden somit aus dem Dropdown. Wurden zu

jedem Filterkriterium Bedingungen erfasst, verschwindet der gesamte Hinzufügen-Button.

Filterübersicht mit Qualifikationen und Rollen

Im folgenden Bild hat der Benutzer Tags, Rollen und Qualifikationen als Filterkriterien erfasst.

Filter-Kriterien - Show Maske - Alle Filter definiert

☐ In der aktuellen Ebene und allen darunter liegenden Ebenen und Gruppen
☐ In der aktuellen Ebene und allen ihren Gruppen
☐ Nur in der aktuellen Gruppe

Tags

Einschliessen
UM-ID-179513
UM-ID-189613

Ausschliessen
UM-ID-179516

Rollen

Gruppen:
SAC Geschäftsleitung

Rollen:
Mitarbeiter*in (schreibend)
Andere

Gültigkeit:
Aktiv
Erstellt
Gelöscht
Inaktiv

Stichdatum:
12.08.2025
bis
25.09.2025

Qualifikation

Qualifikationen:
Bikeleiter*in
Bergführer

Kriterium:
☐ Person hat alle
☐ Person hat mind. eine Qualifikation

Gültigkeit:
Aktiv
Gültige

Stichdatum:
12.08.2025
bis
25.09.2025

Abbildung 13.19: Mockup mit Qualifikationen und Rollen definiert

13.2.2 Bearbeitungsmasken

Nachfolgend werden die Anzeigen vorgestellt, welche der Benutzer bearbeiten kann. Diese sind für den Benutzer zugänglich, indem er ein Filterkriterium entweder per Hinzufügen-Button hinzufügt oder per Editier-Button, Stift in der oberen rechten Ecke der Box, bearbeitet.

Felder

Die Felder können im neuen Mockup durch ein Dropdown ausgewählt werden. Danach kann wie zuvor die Genauigkeit bestimmt werden. Will der Benutzer ein weiteres Attribut zu seinen Suchbedingungen hinzufügen, kann er dies über den Hinzufügen-Button in der rechten unteren Ecke. Auf dieser Benutzerschnittstelle ist es möglich, die Sprache als Attribut auszuwählen. Für die Sprache kann standardmässig keine Genauigkeit definiert werden.

The mockup shows a rectangular box with a thin border. Inside, there are three rows of input fields. The first row is labeled 'Attribut:' and contains a dropdown menu with 'Alter' selected. The second row is labeled 'Wert:' and contains a dropdown menu with '>=' selected and a text input field with '42'. The third row contains two buttons: 'Speichern' and 'Abbrechen'. In the bottom right corner of the box, there is a circular button with a plus sign (+).

Abbildung 13.20: Bearbeitungsmaske Filterkriterium Felder

Rollen

Die Rolle kann über die Gruppe oder den Rollennamen definiert werden. Wird unter dem Eingabefeld “Gruppen” eine Gruppe hinterlegt, werden alle dazugehörigen Rollen dieser Gruppe in die Filterung miteinbezogen. Will der User nur einzelne Rollen in die Filterung miteinbeziehen, kann er dies über das Rollen-Eingabefeld. Die Gültigkeit der Rolle wird kann über ein suchbares Dropdown ausgewählt werden. Das Element des Stichdatums bleibt bestehen. Die Checkbox für die archivierten Rollen wird im Mockup vernachlässigt und alle Rollen werden standardmässig, sofern diese die Gültigkeitsbedingungen erfüllen, miteinbezogen.

The mockup shows a form titled 'Gruppen:' with four input fields and two buttons. The first field is 'Gruppen:' with buttons 'Bergsteiger' and 'Maurer'. The second field is 'Rolle:' with buttons 'Coach', 'Chef', and 'Mitglied'. The third field is 'Gültigkeit:' with buttons 'Aktiv', 'Erstellt', 'Gelöscht', and 'Inaktiv'. The fourth field is 'Stichdatum:' with a calendar icon, a text input, the word 'bis', another calendar icon, and the date '25.09.2025'. At the bottom are two buttons: 'Speichern' and 'Abbrechen'.

Abbildung 13.21: Bearbeitungsmaske Filterkriterium Gruppe

Qualifikationen

Die Qualifikationen können innerhalb der Bearbeitungsmaske ebenfalls mit einem suchbaren Dropdown ausgewählt werden. Die Radio-Buttons und das Stichdatum bestehen wie bisher. Die Gültigkeit kann wie bei den Rollen über ein suchbares Stichdatum ausgewählt werden.

Abbildung 13.22: Bearbeitungsmaske Filterkriterium Qualifikationen

Tags

Die Definition der Tags bleibt bestehen, wie in der Soll-Situation beschrieben

Abbildung 13.23: Bearbeitungsmaske Filterkriterium Tags

13.3 Fehlende Informationen

Alle bekannten Informationen bei Beginn der IPA, wurden in der Ist-Situation und der Soll-Situation beschrieben. Als unbekannte Informationen gelten: Die Bedürfnisse des Benutzers, Sicherheitsrisiken, sowie die daraus resultierenden Produktanforderungen. Die genannten Bereiche werden in den kommenden Abschnitten aufgenommen und analysiert. Alle unbekannten Informationen, welche einzelne Definitionen oder Abläufe betreffen, wurden über das Internet oder KI recherchiert und anschliessend im Anhang unter dem Datum der Verwendung hinterlegt.

13.4 Änderungen am Mockup

Während des Implementierens wurden Aspekte entdeckt, welche am Mockup geändert werden mussten. Diese werden im folgenden Abschnitt dokumentiert.

- **Bearbeitung der Filterkriterien:** Um Filterkriterien zu bearbeiten muss nicht auf ein Bearbeitungssymbol geklickt werden. Die Bearbeitung kann direkt nach dem Hinzufügen eines Filterkriteriums vorgenommen werden
- **Gruppendropdown im Rollenfilterkriterium:** Es wird kein Dropdown zur Filterung der Rollen nach Gruppen zur Verfügung gestellt. Die Rollen sind alle über ein einzelnes Dropdown auswählbar.
- **Entfernung eines Filterkriteriums:** Als zusätzliche Funktionalität wird ein Mülleimer Symbol auf jedem Filterkriterium eingefügt, um dieses von der Filterkonfiguration zu entfernen.
- **Filterkriterium wird beibehalten:** Die Benutzerschnittstelle für das Bearbeiten des Filterkriteriums “Felder” wird nicht modifiziert und lediglich in der neuen Benutzerschnittstelle angeordnet.

13.5 Bedürfniserhebung

Um die Bedürfnisse der Kunden vor der Entwicklung zu identifizieren, wird eine Bedürfniserhebung durchgeführt. Angewendete Modelle, Befragungstechniken und Erhebungen, werden im folgenden Abschnitt dokumentiert. Es wird zuerst die Zielsetzung und Planung definiert, danach die Methode der Erhebung ausgewählt, die Erhebung durchgeführt und zuletzt alle Daten und identifizierten Bedürfnisse analysiert.

13.5.1 Zielsetzung und Planung

Mit dieser Bedürfniserhebung sollen Anforderungen an das Produkt auf Kundenseite ausgemacht werden. Aus zeitlichen Gründen werden die Anforderung einer Person an das Produkt analysiert. Bei der Person handelt es sich um Thomas Ellenberg, dem Projektleiter von Hitobito. Für die Bedürfniserhebung wurden vier Stunden geplant. Zwei Stunden werden für die Vorbereitung verwendet, eine Stunde für die Durchführung und eine Stunde für die Auswertung der Erhebung.

Beteiligt an der Bedürfniserhebung sind Marc Egli und Thomas Ellenberg. Marc Egli ist für die Durchführung der Erhebung zuständig. Da die Bedürfniserhebung im Rahmen der IPA durchgeführt wird, wird mit keinem Budget geplant.

13.5.2 Methodenwahl

Es stehen vier Methoden zur Bedürfniserhebung zur Verfügung:

- Umfragen
- Interviews und Fokusgruppen
- Beobachtungen
- Dokumentenanalyse

In dieser Arbeit wurde als Methode das Interview gewählt. Die Begründung resultiert aus folgendem Ausschlussverfahren:

Methode	Gedankengänge	Verwendung
Umfragen	Hierfür wird eine grosse Benutzergruppe benötigt, um ein aussagekräftiges Resultat daraus zu ziehen. Für die Organisation einer solchen Benutzergruppe besteht keine Zeit.	Nein
Interviews und Fokusgruppen	Durch ein Interview können mehr Informationen gewonnen werden als einer Umfrage. Bspw. Beobachtungen der Gefühle des Benutzers oder Gedankengänge können aufgenommen werden. Um dennoch den Zeitrahmen der IPA nicht zu verletzen, müsste das Interview mit nur einer Person durchgeführt werden.	Ja
Beobachtungen	Um Bedürfnisse mit einer Beobachtung durchzuführen, muss zuerst ein Testskript geschrieben werden. In einem Testskript muss jeder Schritt und die daraus folgenden Aufgabe klar definiert sein. Dies geht mit einem hohen Zeitaufwand für die Verfassung des Testskripts einher.	Nein
Dokumenten-analyse	Die Dokumentenanalyse kann mit dem Benutzerhandbuch von Hitobito durchgeführt werden. Die daraus entstehenden Bedürfnisse würden aber vom Analyst selbst kommen, nicht direkt vom Benutzer.	Nein

Tabelle 13.2: Methodenwahl

13.5.3 Fragenkatalog

Die Fragen im Interview basieren auf dieser [Anleitung](#). Während des Interviews, werden die nachkommenden Fragen gestellt:

Offene Einleitungsfragen

- Frage 1: Was bist du für eine Person? Beschreibe dich kurz
- Frage 2: Was ist dir wichtig im Leben?
- Frage 3: Welchen Karriereweg hast du hinter dir?
- Frage 4: Wie kamst du das erste Mal in Kontakt mit Hitobito?

Fragen zu Hitobito allgemein

- Frage 5: Was fasziniert dich an Hitobito?
- Frage 6: Welche Teile der Applikation stören dich selbst?

Fragen zur Hitobito Filterung

- Frage 7: Welche Erfahrungen hast du mit der Filterung von Personen und Abonnenten in Hitobito gemacht?
- Frage 8: Was stört dich an dieser Filterung?
- Frage 9: Was würdest du an dieser Filterung ändern?
- Frage 10: Welche zusätzlichen Funktionen wünschst du dir für diese Filterung?
- Frage 11: Wie würdest du alle Mängel und zusätzlichen Features auf priorisieren?
- Frage 12: Welche Dringlichkeit hat eine Überarbeitung der Filterung für dich?
- Frage 13: Welche Kosten wärest du maximal bereit zu zahlen?

13.5.4 Ablaufprotokoll

Tätigkeit	Antwort der Testperson	Mimik und Gestik
Begrüssung der Testperson	Danke vielmals für die Einladung.	Testperson scheint positiv eingestimmt zu sein, ein bisschen nervös.
Erklärung des Ablaufs	Perfekt, ja das stimmt so für mich.	Testperson richtet den Blick aufmerksam auf den Befrager.
Frage 1	Ich bin teilweise Product Owner von Hitobito. Eigentlich ist Oliver Dietschi der Product Owner, aber ich höre von vielen Kunden, was ihre Bedürfnisse sind.	Testperson Lächelt leicht.
Frage 2	Im Moment ist mir Nachhaltigkeit unheimlich wichtig. In allen Bereichen, Beziehungen und Freundschaften. Auch was ich kaufe und wie ich lebe.	Testperson wird ernster. Stimme senkt sich.
Frage 3	Ich habe ursprünglich das KV beim Inter Discount gemacht. Im Detailhandel. Danach habe ich bei der Swisscom gearbeitet und später gemerkt, dass das wohl doch nichts für mich ist. Danach habe ich ein Wirtschaftsstudium an der BFH gemacht. Dort habe ich gemerkt, dass mir Projektmanagement sehr zusagt. Anschliessend habe ich im Bereich Ausbildung und in der Stahlindustrie als Projektmanager gearbeitet. Schlussendlich bin ich in der IT gelandet.	Testperson ist ernster zu Beginn der Antwort, lächelt beim Übergang zu der IT.

Tabelle 13.3: Ablaufprotokoll Teil 1

13.5.5 Auswertung

Im kommenden Abschnitt werden die wichtigsten Bedürfnisse aufgelistet. Die Bedürfnisse werden nach Dringlichkeit und Relevanz priorisiert. Die Datengrundlage dafür, bietet das hinterlegte Ablaufprotokoll.

Die Dringlichkeit wird mit den Stufen D1-D3 definiert. Die Stufen sind folgendermassen zu beurteilen:

Die Reihenfolge der Bedürfnisse in der Tabelle entspricht ihrer Priorisierung

Fazit

Betroffen von den Anpassungen anhand der Bedürfnissen sind die Benutzerschnittstellen zur Personenfilterung und zur Anpassung der globalen Bedingung in den Abonnements. Somit muss das Filterungssystem wie es bis jetzt in der Applikation besteht, überarbeitet und durch das hinterlegte Mockup ersetzt werden. Da die Bedürfniserhebung erst während der IPA stattgefunden hat und das Mockup als Vorarbeit erledigt wurde, werden für die weitere Arbeit ausschliesslich das Bedürfnis “Modularer Aufbau” und “Filterkriterium Sprache” in die Anforderungen aufgenommen.

Tätigkeit	Antwort der Testperson	Mimik und Gestik
Frage 4	Ich war selber ein sehr engagierter Pfadfinder. Ich wusste noch gar nicht, wie Hitobito funktioniert. Ein Kollege von mir, Tobi Hinderling, hatte dann von einer Firma namens Puzzle ITC gehört, welche Hitobito betreibt. Er sagte sie sei super cool organisiert und er wolle unbedingt dort arbeiten. Das machte er dann auch und hatte das Projektmanagement bei Hitobito inne. Tobias und ich, wir waren immer wieder am "Aare böttle" zusammen und da hat er mir gesagt, ich soll doch auch zu Puzzle kommen, denn ihn würde mehr das UX als das Projektmanagement interessieren. Wir könnten dann intern die Rollen wechseln und genau so ist es geschehen.	Testperson Lacht bei der Anekdote zu Tobias Hinderling.
Frage 5	Die Kunden faszinieren mich an Hitobito. Wir haben eine sehr breite Kundenpalette. Wir streben auch eine langfristige Kunden an und nehmen nicht jeden. Wir haben sehr viele Non-Profit Gruppen und generell auch sehr angenehme Kunden.	Denkt kurz nach, ist ernster bei der Antwort.
Frage 6	Das war sehr lange her. Keine Ahnung. Zu lange her.	Schüttelt Kopf.
Frage 7	Ich habe die Filterung auch schon verwendet aber meist im Umfang von Tests und Abnahme der Akzeptanzkriterien von Tickets.	Testperson nickt als Bestätigung.

Tabelle 13.4: Ablaufprotokoll Teil 2

Tätigkeit	Antwort der Testperson	Mimik und Gestik
Frage 8	Was mich am meisten stört ist das UI. Alles ist ein bisschen verzettelt. Du weist nicht genau, wo du suchen musst. Niemand weiss genau, was man suchen kann. Die Filter können unheimlich viel, aber du musst ein bisschen suchen.	Gestikulierte in Richtung Laptop, Stimme wird lauter.
Frage 9	Ich würde versuchen, alles zu vereinheitlichen. Evtl. die Felder und Sprache zusammennehmen. Auch die Rollen haben eine coole Funktion bei der Auswahl der Hauptebene. Aber diese Funktion kennt einfach niemand. Das müsste irgendwo genauer erklärt werden. Funktional ist enorm viel drin. Aber sie hegen einen gewissen Pain.	Testperson wird ruhiger, überlegt, Stimme wird leiser.
Frage 10	Gute Frage. Nein ich glaube nicht, für mich hat die Filterung alles was sie bringt. Aber fairerweise brauche ich die Filterung ein bisschen zu wenig.	Testperson überlegt, Kopf seitlich angelegt, Stimme wird leiser.

Tabelle 13.5: Ablaufprotokoll Teil 3

Tätigkeit	Antwort der Testperson	Mimik und Gestik
Frage 11	<p>Als erstes das Sprachen Dropdown aus der Landschaft entfernen und unter Felder anordnen. Der nervt mich tatsächlich sehr. Sowieso haben wir für jedes Attribut ein Dropdown in den Feldern, aber für die Sprachen hängt es hier einfach so herum?</p> <p>Ansonsten finde ich alles andere ok. Geschlecht ist etwas anderes was mich stört. Als Mann musst du dort ein "m" eingeben und als Frau ein "w". Auch "w" wenn du die Sprache auf französisch eingestellt hast. Und weisst du was? Wenn du kein Geschlecht hast, musst du das Feld leer lassen, was einfach niemand weiss.</p>	<p>Testperson regt sich auf, Gestikuliert in Richtung Hitobito Filterung auf Laptop.</p>
Frage 12	<p>Da kommt bei mir der Ökonom durch. Alles ist nice to have. Wir haben in Hitobito im Moment noch andere Bausteine, welche ich noch dringender priorisiere. Die Filterung ist unschön und nicht toll, aber sie funktioniert. Das ist das Wichtigste. Da gibt es andere Sachen, Code welcher älter ist oder aktiv zu Bugs führt. Da sehe ich momentan mehr Probleme.</p>	<p>Stimme der Testperson wird ruhiger und überlegter.</p>

Tabelle 13.6: Ablaufprotokoll Teil 4

Tätigkeit	Antwort der Testperson	Mimik und Gestik
Frage 13	Das ist eine Scheissfrage. Jeder Kunde sagt, dass er ein festes Budget habe und will nicht mehr ausgeben. Jeder Kunde würde vermutlich nichts zahlen, da aus seiner Sicht das Feature funktionieren müsste. Da es ein OpenSource Projekt ist, ist die Finanzierung eh noch schwieriger geregelt. Wir sagen, wir sind OpenSource und entwickeln das Produkt dann einfach im Auftrag von Kunden weiter.	Testperson lacht zu Beginn seiner Antwort. Lachen mündet in Lächeln und verschwindet gegen Ende der Antwort. Person gestikuliert mittelmässig.
Verabschiedung	Vielen Dank dir. Wenn du Fragen oder noch weiteres wissen musst, kannst du gerne nochmals zu mir kommen.	Testperson lächelt und verabschiedet sich mit einem kurzem Wink.

Tabelle 13.7: Ablaufprotokoll Teil 5

Stufe	Beschreibung
D1	Höchste Dringlichkeit. Bedürfnis muss in den nächsten Wochen umgesetzt werden. Nicht Erfüllung führt zu hohem Nutzerverlust.
D2	Mittlere Dringlichkeit. Bedürfnis muss in den nächsten Monaten umgesetzt werden. Nicht Erfüllung führt zu überschaubarem Nutzerverlust.
D3	Niedrige Dringlichkeit. Bedürfnis muss im nächsten Jahr umgesetzt werden. Nicht Erfüllung führt zu keinem Nutzerverlust.

Tabelle 13.8: Dringlichkeitsstufen

Bedürfnis	Störfaktor	Dringlichkeit	Begründung
Modularer Aufbau	Das Filterungs UI ist unübersichtlich und der Benutzer kann nicht verstehen, welche Optionen zur Filterung zur Verfügung stehen. Die Filterung soll einem modularen Aufbau folgen, welcher den User durch die Filterung führt.	D1	Durch den unübersichtlichen Aufbau der Filterung und die Reaktion der Testperson, wird diesem Bedürfnis die höchste Priorität zugeschrieben.
Filterkriterium "Sprache"	Das Filterkriterium "Sprache" soll mit dem Filterkriterium "Felder" vereinheitlicht werden	D2	Testperson hat das Bedürfnis zur Vereinheitlichung der Filterkriterien während des Interviews mehrmals wiederholt, weswegen dieses Bedürfnis die zweithöchste Priorität einnimmt.
Option "Geschlecht"	Die Option "Geschlecht" soll via Dropdown anzupassen sein. Die Optionen sollen sich anhand der ausgewählten Sprache in Hitobito anpassen.	D3	Während des Interviews wurde dieser Punkt nur nebenbei erwähnt, weswegen dieses Bedürfnis als letztes aufgeführt wird.

Tabelle 13.9: Bedürfnisse der befragten Person

14 Risikoanalyse und Sicherheitsmassnahmen

14.1 Schnittstellen

Action	Controller	Funktion
new	PeopleFiltersController	Diese Schnittstelle liefert die Benutzerschnittstelle zur Filterkonfiguration zurück.
edit	PeopleFiltersController	Diese Schnittstelle liefert die Benutzerschnittstelle zur Filterkonfiguration zurück. Die Benutzerschnittstellen sind mit den Werten aus dem definierten Filter befüllt. Der Filter wird über die Angabe einer Filter-ID definiert.
delete	PeopleFiltersController	Diese Schnittstelle ermöglicht das Löschen von Filterkriterien auf der Benutzerschnittstelle. Als Parameter wird der Name des Filterkriteriums mitgegeben.

Tabelle 14.1: Schnittstellen

14.2 Benutzer und Datenzugriffe

Benutzer im Hitobito besitzen immer eine Rolle. Die Rolle des Benutzers bestimmt seine Berechtigungen. Die Berechtigungen welche ein User haben kann sind:

Name	Berechtigung
Group_Full	Hat Schreib- und Leserechte auf seiner Gruppe
Group_Read	Hat Leserechte auf seiner Gruppe
Layer_Full	Hat Schreib- und Leserechte auf seiner Gruppe und den Gruppen, welche der Ebene dieser Gruppe unterliegen.
Layer_Read	Hat Leserechte auf seiner Gruppe und den Gruppen welche der Ebene dieser Gruppe unterliegen.
Layer_And_Below_Full	Hat Schreib- und Leserechte auf seiner Gruppe, allen Gruppen der Ebene dieser Gruppe und allen unterliegenden Ebenen.
Layer_And_Below_Read	Hat Leserechte auf seiner Gruppe, allen Gruppen der Ebene dieser Gruppe und allen unterliegenden Ebenen.

Tabelle 14.2: Berechtigungen

Um die Berechtigungen besser verständlich zu machen, dienen folgende Diagramme:

14.2.1 Datenstruktur

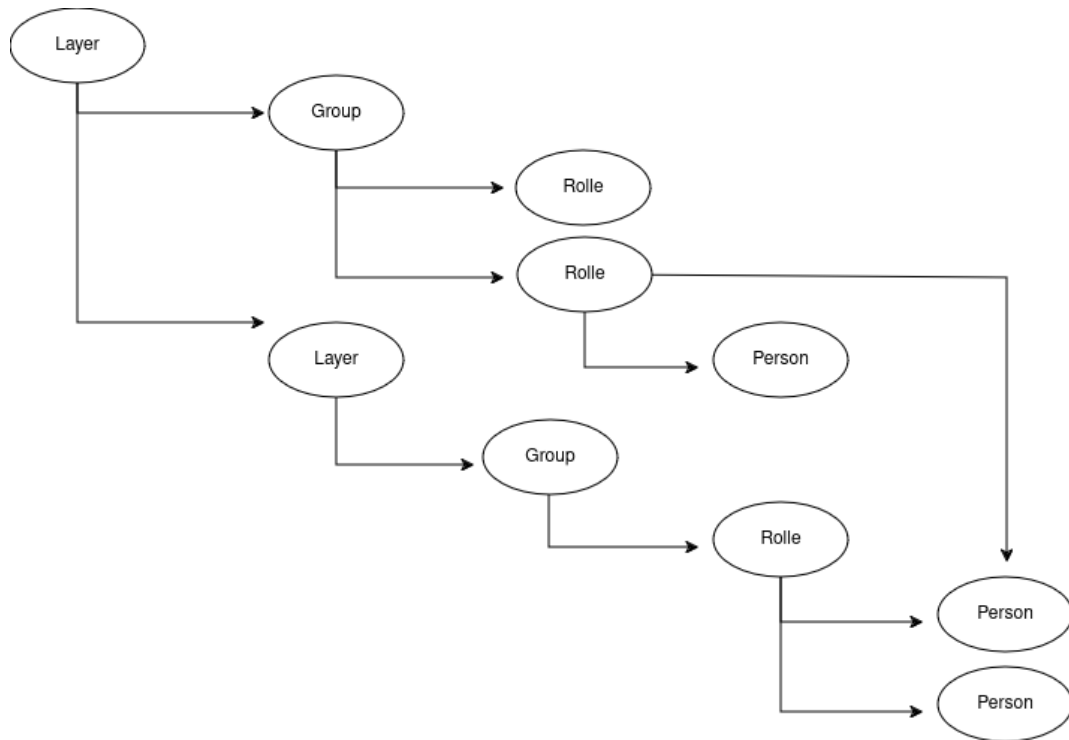


Abbildung 14.1: Gruppen und Ebenen, selbstgezeichnet mit Draw.io

Die Berechtigungen verwalten den Zugriff auf Layer und Gruppen. Ein Layer kann mehrere Gruppen haben, eine Gruppe besitzt mehrere Rollen und eine Rolle kann wiederum mehrere Personen besitzen. Personen können mehrere Rollen und somit eine Vielzahl von Berechtigungen besitzen.

14.2.2 Beispiel Zugriff Heinz

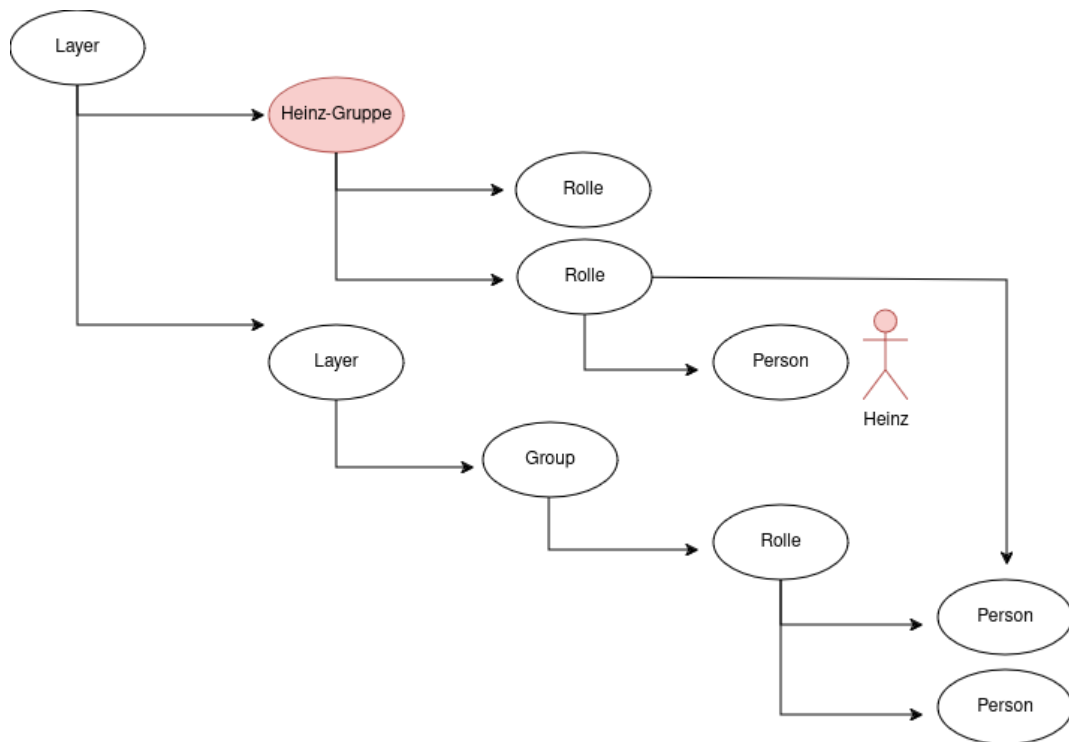


Abbildung 14.2: Beispiel Berechtigungen von Heinz, selbstgezeichnet mit Draw.io

Dieses Diagramm erklärt das Beispiel der Berechtigung "Group_Full". Wir haben einen User namens Heinz in unserem System. Heinz besitzt eine Rolle, welche mit der Heinz-Gruppe verknüpft ist. Die Rolle besitzt die Berechtigung "Group_Full".

Dank dieser Verknüpfung besitzt Heinz Schreib- und Leserechte auf die Heinz-Gruppe.

14.2.3 Beispiel Zugriff Tim

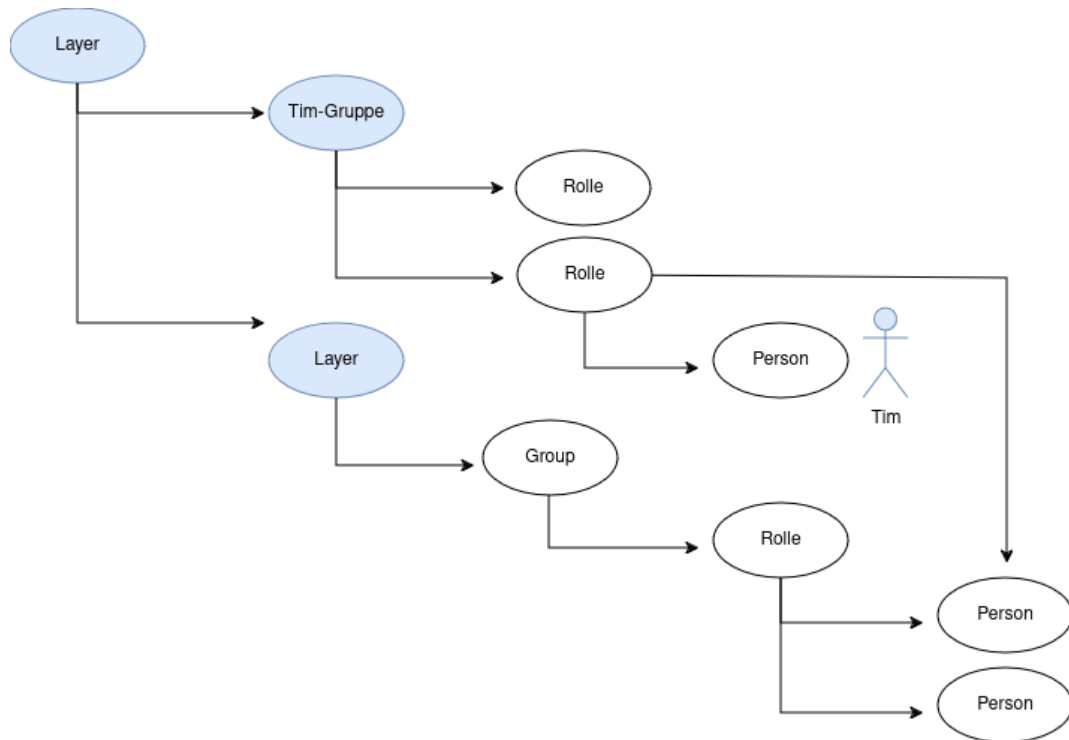


Abbildung 14.3: Beispiel Berechtigungen von Tim, selbstgezeichnet mit Draw.io

Dieses Diagramm erklärt das Beispiel der Berechtigung "Layer_Full". Wir haben einen User names Tim in unserem System. Tim besitzt eine Rolle, welche mit der Tim-Gruppe verknüpft ist. Die Rolle besitzt die Berechtigung "Layer_Full".

Durch diese Verknüpfung hat Tim Schreib- und Leserechte auf alle Gruppen, welche dem Layer seiner Gruppe unterliegen.

14.2.4 Beispiel Zugriff Rudolf

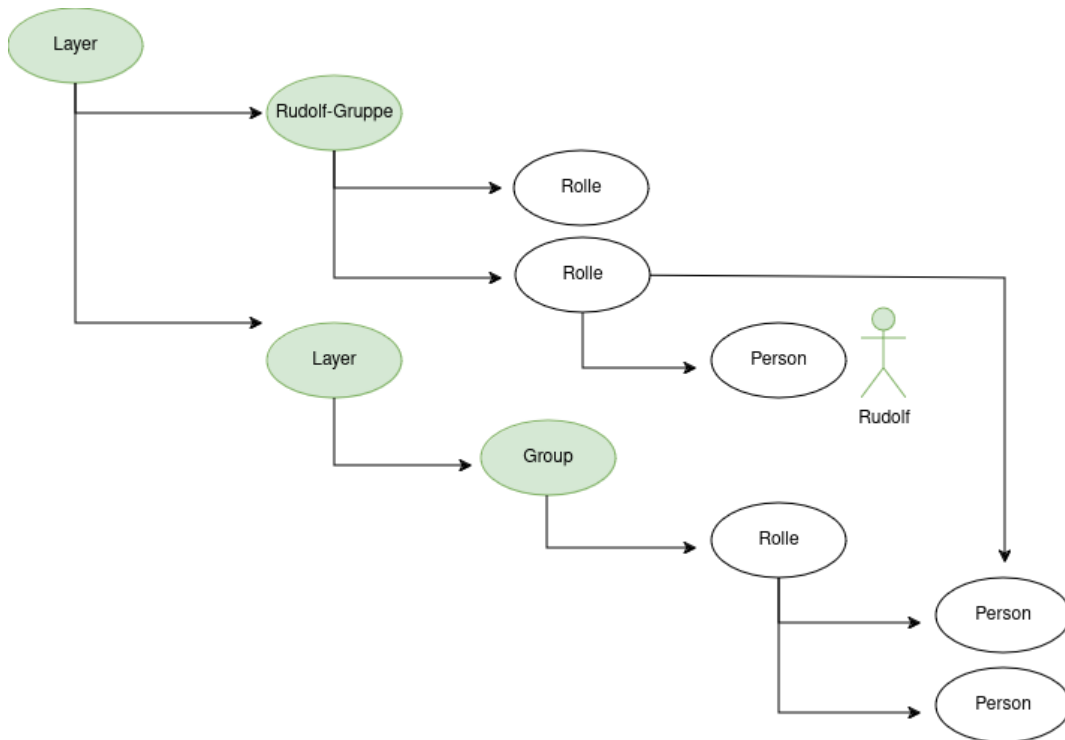


Abbildung 14.4: Beispiel Berechtigungen von Tim, selbstgezeichnet mit Draw.io

Dieses Diagramm erklärt das Beispiel der Berechtigung "Layer_Full_And_Below". Wir haben einen Benutzer namens Rudolf in unserem System. Rudolf besitzt eine Rolle, welche mit der Rudolf-Gruppe verknüpft ist. Die Rolle besitzt die Berechtigung "Layer_Full_And_Below".

Durch diese Verknüpfung hat Rudolf Schreib- und Leserechte auf alle Elemente, Layer und Gruppen, welche dem Layer der Rudolf-Gruppe unterliegen.

14.2.5 Bedeutung für die Schnittstellen

Durch die erklärten Berechtigungen, welche von den Rollen der Benutzern gegeben sind, werden die Rückgabewerte der Schnittstellen gefiltert. Da im Rahmen dieser IPA eine Frontendanpassung gemacht wird, müssen bei der Berechtigungslogik keine Anpassungen gemacht werden. Die Berechtigungslogik wird wie beschrieben verwendet.

14.2.6 Risikoanalyse

Nr	Risikobeschreibung	Auswirkung	Vor Massnahme				Massnahmen	Nach Massnahme			
			W	S	Risiko	Handlungsweise		W	S	Risiko	Handlungsweise
1	Daten ausserhalb der Berechtigung eines Benutzers werden angezeigt	Benutzer kann verbotene Informationen einsehen	W2	S2	Niedrig	Risikominderung	Daten werden vor dem Anzeigen im Filter anhand der Berechtigungen des Benutzers gefiltert	W1	S1	Niedrig	Risikoakzeptanz
2	Benutzer kann einen Filter auf einer Ebene speichern, auf welcher er keinen Zugriff hat	Verwirrte Benutzer durch den neuen Filter	W2	S2	Niedrig	Risikominderung	Sicherstellen das der Benutzer nur Filter seiner Berechtigung entsprechend speichern kann.	W1	S1	Niedrig	Risikoakzeptanz
3	SQL-Injection in ein Filter Eingabefeld (XSS)	Datenbank kann ausgelesen oder verändert werden	W4	S4	Hoch	Risikominderung	Alle Eingaben des Benutzers escapen	W2	S1	Niedrig	Risikoakzeptanz
4	Bash-Injection in ein Filter Eingabefeld (XSS)	Schädliche Befehle werden serverseitig ausgeführt	W3	S4	Hoch	Risikominderung	Alle Eingaben des Benutzers escapen	W2	S1	Niedrig	Risikoakzeptanz
5	Falsche Verwendung einer Library	Schwachstelle der Library kann von Angreifern ausgenutzt werden	W2	S3	Mttel	Risikominderung	Dokumentation der Libraries gut durchgehen, Libraries verwenden welche in der Rails Landschaft etabliert sind	W2	S2	Niedrig	Risikoakzeptanz

Tabelle 14.3: Risikoanalyse Sicherheitsrisiken

Schadensausmass:
S1 = führt zu keinem Schaden am Projekt
S2 = führt zu geringem Schaden
S3 = hoher Schaden
S4 = führt zu schwerem Schaden am Projekt

Eintrittswahrscheinlichkeit:
W1 = unvorstellbar
W2 = unwahrscheinlich
W3 = eher vorstellbar
W4 = vorstellbar
W5 = Eintreffen hoch

14.3 Risikomatrix

W5				
W4				③
W3				④
W2	③ ⑤ ④			⑤
W1	① ②			
	S1	S2	S3	S4

Tabelle 14.4: Risikomatrix Sicherheitsrisiken

Legende:

- Risiko ohne Massnahme
- Risiko nach Massnahme
- Geringes Risiko
- Mittleres Risiko
- Hohes Risiko

14.4 Auswertung

Die aufgeführten Risiken sowie die entsprechenden Massnahmen wurden mit den Stakeholdern besprochen und von ihnen abgesegnet. Durch die Bestätigung der Stakeholder, werden die Massnahmen zur Risikominderung in der Anforderungskatalog überführt.

14.5 Anforderungen

Alle Anforderungen an das Produkt werden im folgenden Abschnitt beschrieben. Die Priorisierung ist durch die Reihenfolge der Anforderungen gegeben. Somit erhält die erste Anforderung die höchste Priorität. Nicht funktionale und funktionale Anforderungen werden als gleichwertig bewertet. Somit erhält die erste nicht funktionale Anforderung die gleiche Priorität wie die erste funktionale Anforderung.

14.5.1 Nicht funktionale Anforderungen

Nicht funktionale Anforderung 1

Beschreibung	Die Eingabefelder der Filter Benutzerschnittstelle unterbindet XSS Angriffe, gemäss Sicherheitsrisiko 3 und Sicherheitsrisiko 4
Messbarkeit	Benutzer können keine Bash, SQL oder sonstige Injections in das Eingabefeld einfügen.

Tabelle 14.5: Nicht funktionale Anforderung 1

Nicht funktionale Anforderung 2

Beschreibung	Es werden im Rahmen der Implementation etablierte Libraries. Basiert auf Sicherheitsrisiko 5
Messbarkeit	Die verwendeten Libraries sind bekannt und besitzen eine Vielzahl von Downloads

Tabelle 14.6: Nicht funktionale Anforderung 2

Nicht funktionale Anforderung 3

Beschreibung	Filterung soll ausschliesslich verifizierte Daten anzeigen, basiert auf Sicherheitsrisiko 1
Messbarkeit	Benutzer können keine Daten einsehen, auf welche sie keine Berechtigungen haben.

Tabelle 14.7: Nicht funktionale Anforderung 3

Nicht funktionale Anforderung 4

Beschreibung	Speicherung von Filtern sind nur auf dem Benutzer zugänglichen Ebenen möglich. Basiert auf Sicherheitsrisiko 2
Messbarkeit	Filter welche vom Benutzer gespeichert werden, sind ausschliesslich auf seiner Ebene zugänglich.

Tabelle 14.8: Nicht funktionale Anforderung 4

Nicht funktionale Anforderung 5

Beschreibung	Die gewählte Implementation der Überarbeit der Benutzerschnittstelle berücksichtigt, dass zukünftig weitere Filterkriterien dazukommen können.
Messbarkeit	Es wird eine generische Methode zur Implementation gewählt, so dass zukünftige Filterkriterien nach dem vorgegebenen Muster implementiert werden können.

Tabelle 14.9: Nicht funktionale Anforderung 5

Nicht funktionale Anforderung 6

Beschreibung	Die Benutzerschnittstelle ist visuell ansprechend implementiert und folgt einem modularem Aufbau. Gemäss Bedürfnis aus Bedürfniserhebung
Messbarkeit	Es ist dem Benutzer möglich, durch die hinterlegte Instruktion, das Produkt zweckmässig zu verwenden.

Tabelle 14.10: Nicht funktionale Anforderung 6

14.5.2 Funktionale Anforderungen

Funktionale Anforderung 1

Beschreibung	Die Filterung bietet mit der neuen Implementation die gleiche Grundfunktionalität wie bisher.
Messbarkeit	Alle Funktionen, welche in der Ist-Situation dokumentiert wurden, sind nach der Implementation immer noch vorhanden.
Testart	Manuell und automatisiert

Tabelle 14.11: Funktionale Anforderung 1

Funktionale Anforderung 2

Beschreibung	Neue Filterkriterien können mit dem Hinzufügen-Button hinzugefügt werden.
Messbarkeit	Die beschriebene Funktion kann in der Benutzerschnittstelle ausgeführt werden.
Testart	Manuell und automatisiert

Tabelle 14.12: Funktionale Anforderung 2

Funktionale Anforderung 3

Beschreibung	Die Bedingungen der Filterkriterien können über den Bearbeiten-Button editiert werden.
Messbarkeit	Die beschriebene Funktion kann in der Benutzerschnittstelle ausgeführt werden.
Testart	Manuell und automatisiert

Tabelle 14.13: Funktionale Anforderung 3

Funktionale Anforderung 4

Beschreibung	Neue Filterkriterien werden in Boxen anstatt von Dropdowns angezeigt.
Messbarkeit	Die beschriebene Funktion ist in der Benutzerschnittstelle ersichtlich.
Testart	Manuell und automatisiert

Tabelle 14.14: Funktionale Anforderung 4

Funktionale Anforderung 5

Beschreibung	Das Filterkriterium "Sprache" wurde im Filterkriterium "Felder" untergebracht. Gemäss Bedürfnis 2 aus Bedürfniserhebung
Messbarkeit	Es besteht kein eigenes Filterkriterium "Sprache" mehr im Dropdown.
Testart	Manuell und automatisiert

Tabelle 14.15: Funktionale Anforderung 5

14.6 Abgrenzung

Das Ziel dieser Arbeit ist es, eine komplette Überarbeitung der Benutzerschnittstelle von Personen- und Abonnementsfilter im Hitobito einzubauen. Sämtliches Testing wird nur für die neue Benutzerschnittstelle gemacht, kaputte oder alte Tests werden deaktiviert und kommentiert, sofern diese nichts mit der Implementation des Produktes zu tun haben. Diese Arbeit beschränkt sich auf die Überarbeitung der Benutzerschnittstelle des Personenfilters und der Benutzerschnittstelle der globalen Bedingungen der

Abonnenten. Andere Benutzerschnittstellen, welche mit der Filterung verbunden sind, werden erst nach der IPA überarbeitet.

Während dieser IPA wird die Funktionalität nur mit dem Hitobito und Hitobito-Generic Wagon garantiert. Alle anderen Wagons werden erst nach der IPA überprüft und ergänzt.

14.7 Persönliche Vorgehensziele

Zeitrahmen	
Beschreibung	Der erstellte Zeitplan und Sprints werden erfolgreich eingehalten.
Messbarkeit	Die IPA wird komplett und in der definierten Frist abgeben.

Tabelle 14.16: Zeitrahmen

Hotwire Kenntnisse	
Beschreibung	Das Wissen und die Fachkompetenz um Hotwire wird vertieft.
Messbarkeit	Das Wissen kann in zukünftigen Aufträgen angewendet werden.

Tabelle 14.17: Hotwire Kenntnisse

Selbstständigkeit	
Beschreibung	Die Arbeit soll selbständig geführt werden, allfällige Fragen sind immer mit einem konkreten Lösungsvorschlag verbunden.
Messbarkeit	Es wird wenig und sinnvolle Hilfe von den verantwortlichen Fachkräften benötigt.

Tabelle 14.18: Selbstständigkeit

15 Entwurf

15.1 Anwendungskonzept

Im folgenden Abschnitt werden die Anwendungsfälle des Benutzers dokumentiert.

15.1.1 Anwendungsdiagramm

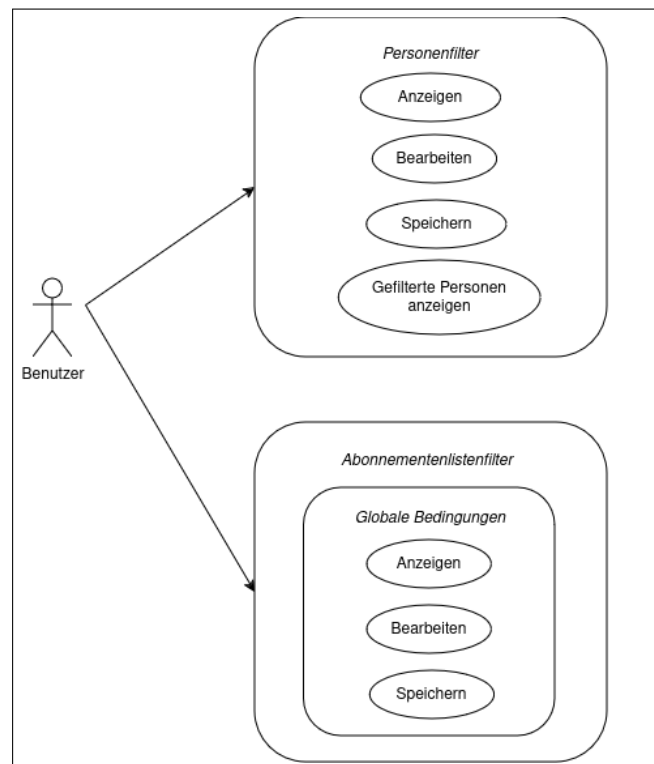


Abbildung 15.1: Anwendungsdiagramm

15.1.2 Anwendungsfälle

Aus dem Anwendungsdiagramm werden die 4 Use-Cases entnommen und hier im Detail beschrieben. Vier Use-Cases, da die Anzeige, die Bearbeitung und die Speicherung sowohl bei den Personenfiltern wie auch bei den Abonnementenfiltern über die gleiche Benutzerschnittstelle läuft.

Filter speichern	
Kurzbeschreibung	Der Benutzer kann die ausgewählten Filterkriterien speichern.
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer besitzt die nötigen Rechte um einen Filter zu erstellen • Mind. ein Filterkriterium wurde ausgewählt
Ablauf	<ol style="list-style-type: none"> 1. Benutzer benennt den Filter, optional und nur bei der Filterung von Personen 2. Benutzer klickt auf speichern
Resultat	Der Filter wurde in der Datenbank persistiert und ein Success-Alert wird ausgegeben.

Tabelle 15.1: Anwendungsfall: Filter speichern

Filterkriterien anzeigen	
Kurzbeschreibung	Der Benutzer kann die Filterkriterien der gespeicherten Filter einsehen.
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer hat einen Filter gespeichert
Ablauf	<ol style="list-style-type: none"> 1. Navigiert zum Filter
Resultat	Die Filterkriterien werden dem Benutzer angezeigt.

Tabelle 15.2: Anwendungsfall: Filterkriterien anzeigen

Filterkriterien bearbeiten	
Kurzbeschreibung	Der Benutzer kann die Filterkriterien bearbeiten.
Vorbedingungen	<ul style="list-style-type: none">• Der Benutzer hat einen Filter gespeichert
Ablauf	<ol style="list-style-type: none">1. Der Benutzer klickt auf den Bearbeiten-Button
Resultat	Die Filterkriterien werden dem Benutzer angezeigt.

Tabelle 15.3: Anwendungsfall: Filterkriterien bearbeiten

Personen filtern	
Kurzbeschreibung	Der Benutzer kann die definierten Personenfilter auf eine Liste von Personen anwenden.
Vorbedingungen	<ul style="list-style-type: none">• Benutzer besitzt Rechte um auf eine Personenliste zuzugreifen• Benutzer hat einen Personenfilter für diese Liste gespeichert
Ablauf	<ol style="list-style-type: none">1. Benutzer Navigiert zum Personenfilter2. Benutzer klickt auf den Filternamen
Resultat	Die Personen in der Personenliste werden gefiltert und dem Benutzer angezeigt.

Tabelle 15.4: Anwendungsfall: Personen filtern

15.2 Systemkonzept

Bei dieser Arbeit wird mit einem bestehenden System gearbeitet, dieses muss entsprechend angepasst werden. Um die nötigen Anpassungen besser sichtbar zu machen, werden im folgenden Abschnitt die Betroffenen Services identifiziert. Anschliessend werden mögliche Lösungsvarianten konzeptioniert. Aus den Lösungsvarianten wird per Variantenentscheid eine Lösungsvariante ausgearbeitet.



Abbildung 15.2: Services

15.2.1 Betroffene Services

Hitobito wird in zwei Services unterteilt, der Rails Applikation und der Postgres Datenbank.

Rails Applikation / Webserver

Die Rails Applikation verwaltet die Geschäftslogik von Hitobito. Die Erweiterungen dieser Arbeit werden alle in diesem Service vorgenommen. Je nach Kunde, werden hier Codeteile aus den anderen Wagons verwendet. In dieser Arbeit wird ausschliesslich der Core und der Generic Wagon angepasst.

Postgres Datenbank

Die Datenbank von Hitobito läuft auf PostgreSQL. Sämtliche Abfragen auf die PostgreSQL Datenbank werden via SQL-Queries gemacht. Als ORM (Object Relational Mapping) wird Active Record verwendet.

15.2.2 Lösungsvarianten

Lösungsvariante 1

Die Idee des nachfolgenden Konzeptes ist die Aufteilung des Mockups in Elemente, welche später durch Turbo angesteuert werden können. Hierbei sollen nur Turbostreams verwendet werden. Diese ermöglichen es, Elemente im HAML mit der ID eines Divs, zu diesem Div hinzuzufügen.

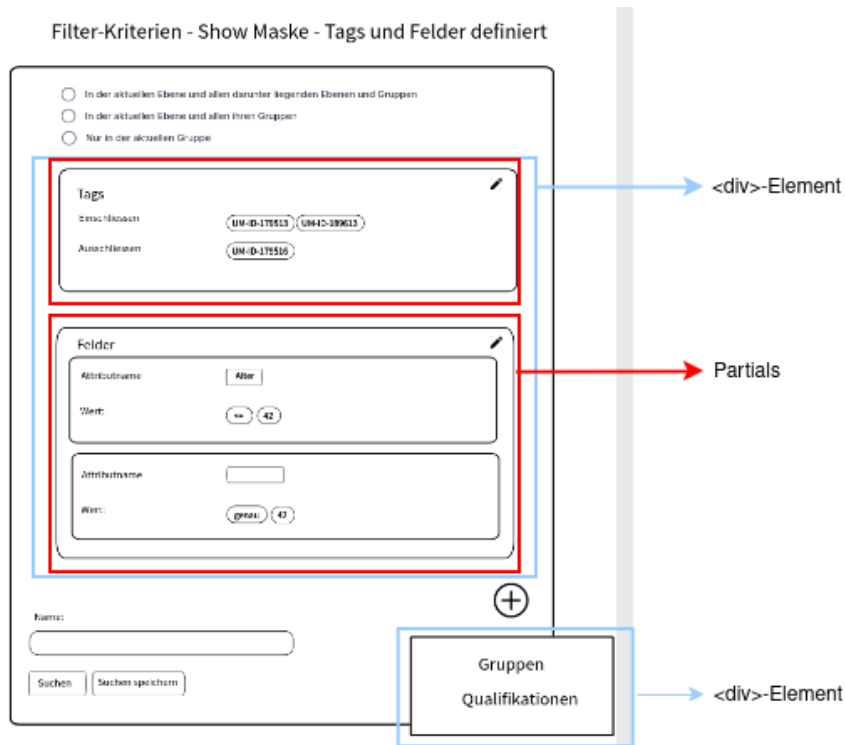


Abbildung 15.3: Lösungsvariante 1: Turbo-Konzept

So werden keine Turboframes benötigt und es müssen lediglich nur noch die Endpoints angepasst werden.

Das daraus resultierende Klassendiagramm sieht so aus:

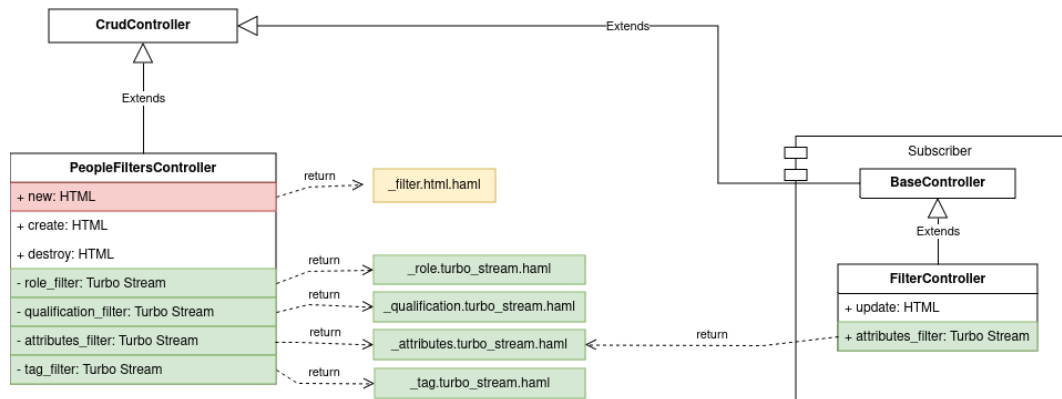


Abbildung 15.4: Lösungsvariante 1: Klassendiagramm

Rot markierte Felder stehen dabei für Funktionen, welche entfernt werden. Grün steht für Funktionen, welche hinzugefügt werden. Gelb steht für Funktionen oder Dateien, welche bearbeitet werden müssen.

Im `PeopleFilterController` wie im `FilterController` werden zusätzliche Endpoints angelegt, welche die jeweiligen Partials als Turbostream zurückgeben. Mit den Turbo Streams werden die Partials einem `textttdiv` angehängt oder von diesem entfernt. Da die Filterkriterien mittels der Turbostreams angezeigt werden, wird der Endpoint “new” im `PeopleFilterController` nicht mehr benötigt. Die View `_filter.html.haml` muss so geändert werden, dass sie alle Boxen, welche durch die Filterkriterien auf der Benutzerschnittstelle darstellen, in ein `Div` verpacken. Dieses `Div` kann später in den Turbostreams referenziert werden.

Lösungsvariante 2

In der zweiten Lösungsvariante wird mit Turboframes statt der Turbostreams gearbeitet. Für jedes Partial besteht zu Beginn ein Turboframe. Wird im Dropdown auf eines der Filterkriterien geklickt, wird der Inhalt des Turboframes mit dem Formular für das jeweilige Filterkriterium befüllt.

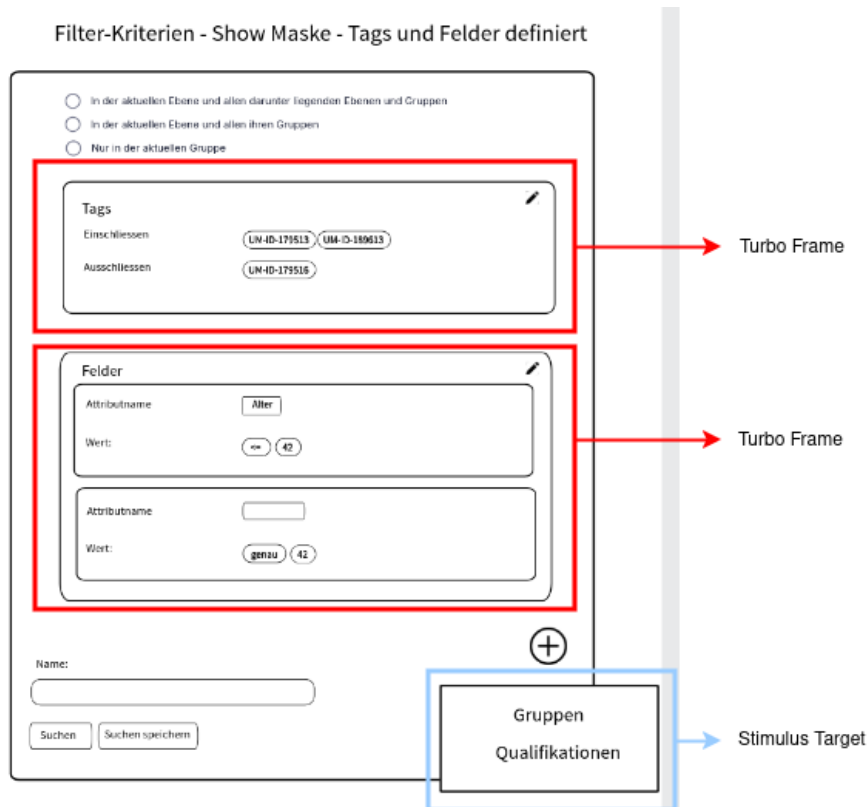


Abbildung 15.5: Lösungsvariante 2: Turbo-Konzept

Die Option im Dropdown wird durch einen Stimulus Controller entfernt. So muss kein zusätzliches Turboframe für das Dropdown angelegt werden.

Eine Umsetzung dieses Turbo-Konzeptes führt zu folgendem Klassendiagramm.

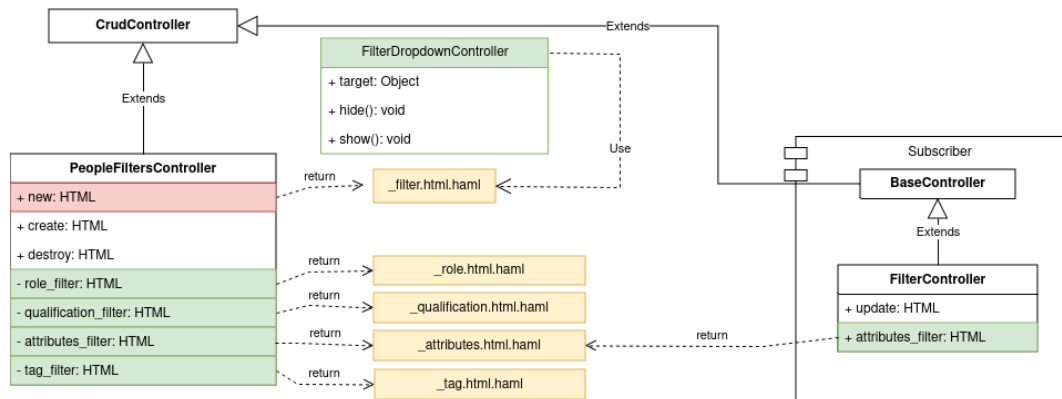


Abbildung 15.6: Lösungsvariante 2: Klassendiagramm

Massgebend ist der Unterschied, dass hier keine Turbostreams benötigt werden. Es wird nach wie vor für jedes Filterkriterium ein Endpoint erstellt. Dieser liefert allerdings die bereits bestehenden Partial zurück. An den Partial selbst muss ein Turboframe eingebaut werden. So wird jedes Formular eines Filterkriteriums von einem Turboframe umschlossen. Als zusätzliche Klasse entsteht in dieser Variante der **FilterDropdownController**. Die Aufgabe dieses Controllers ist es, die Filterkriterien bei der Auswahl aus dem Dropdown zu entfernen. Wurden alle Filterkriterien ausgewählt, blendet der Controller das Dropdown komplett aus.

15.2.3 Variantenentscheid

Um eine geeignete Entscheidung für eine der beschriebenen Lösungsvarianten zu treffen, wird eine Bewertungsmatrix verwendet. Die Bewertungsmatrix bewertet die Lösungsvarianten nach definierten Kriterien. Die Kriterien können mit Punkten von 1 bis 10 bewertet werden, wobei 1 Punkt für das Nicht-Erfüllen eines Kriteriums und 10 für das Erfüllen des Kriteriums steht.

Folgende Kriterien wurden definiert:

Kriterium	Beschreibung	1 Punkt	10 Punkte
Zeitaufwand	Wie viel Zeit wird für die Implementation benötigt?	Grosser Zeitaufwand, IPA ist mit diesem Konzept nicht umsetzbar.	Kleiner Zeitaufwand, IPA ist problemlos umsetzbar.
Einführung	Ist das Konzept einfach in die produktive Umgebung einzuführen? Müssen Migrationen vorgenommen werden?	Einführung in produktive Umgebung ist unmöglich.	Einführung in produktive Umgebung ist problemlos möglich.
Anforderungen	Kann mit dieser Variante jede Anforderung erfüllt werden?	Mehrere Anforderung können nicht erfüllt werden.	Alle Anforderunge können erfüllt werden.
Performance	Ist das Konzept performant? Spart das Konzept Zeit / Requests?	Konzept ist nicht performant, Filterungen sind durch Implementation deutlich langsamer.	Es wird viel Zeit durch die Implementation des Konzepts gespart.

Tabelle 15.5: Variantenentscheid Kriterien

Bewertungen

Im folgenden Abschnitt werden den definierten Kriterien Gewichtungen hinzugefügt, da nicht jedes Kriterium gleich wichtig für diese IPA ist.

Gewichtungen in %

- **Zeitaufwand 40%:** Um eine funktionelle Lösung am Ende der IPA aufweisen zu können, wird dem Zeitaufwand eine hohe Gewichtung zugeordnet.
- **Einführung 20%:** Es ist wichtig eine Lösung zu implementieren, welche schnell ihren Weg in die produktive Umgebung findet, da die Einführung von der IPA ausgenommen ist, wird diesem Kriterium eine geringere Gewichtung zugeordnet.
- **Anforderungen 30%:** Damit die definierten Anforderungen an das Produkt erfüllt werden können und die Erfüllung zur Endnote der IPA beiträgt, wird diesem Kriterium eine hohe Gewichtung zugeordnet.
- **Performanz 10%:** Ist eine Applikation zu langsam und benötigt mehrere Minuten bis sie Resultate geladen hat, kann das den Benutzer schnell vor den Kopf stossen und dazu führen, dass dieser die Applikation in Zukunft nicht mehr verwendet.

Lösungsvariante 1

Kriterium	Bewertung	Beschreibung
Zeitaufwand	5	Mittlerer Zeitaufwand. Durch die vielen Neuimplementationen der Turbostreams geht viel Zeit verloren.
Einführung	8	Einführung möglich, es werden keine grossen Schwierigkeiten auftreten.
Anforderungen	10	Alle Anforderung können potentiell durch diese Lösungsvariante erfüllt werden.
Performance	8	Performance wird optimiert.

Tabelle 15.6: Bewertung Lösungsvariante 1

Lösungsvariante 2

Kriterium	Bewertung	Beschreibung
Zeitaufwand	6	Durch weniger Neuimplementationen kann viel Zeit gespart werden.
Einführung	6	Einführung möglich, könnte aufgrund des Einsetzens von Turboframes in den bereits bestehenden Partials Probleme ergeben.
Anforderungen	10	Alle Anforderung können potentiell durch diese Lösungsvariante erfüllt werden.
Performance	7	Performanz wird optimiert, jedoch weniger als es mit den Turbostreams möglich ist.

Tabelle 15.7: Bewertung Lösungsvariante 2

		Lösungsvariante 1: Turbostreams		Lösungsvariante 2: Turboframes	
Kriterium	Gewichtung	Ungewichtet	Gewichtet	Ungewichtet	Gewichtet
Zeitaufwand	40%	5	2	6	2.4
Einführung	20%	8	1.6	6	1.2
Anforderungen	30%	10	3	10	3
Performanz	10%	8	0.8	7	0.7
Total	100%	31	7.4	29	7.3

Tabelle 15.8: Nutzwertanalyse

Fazit

Durch die bessere Möglichkeit zur Einführung und der erhöhten Performanz hat die Lösungsvariante 1 die Lösungsvariante 2 in der Endbewertung knapp übertroffen. Für die Implementation wird fortlaufend die Lösungsvariante 2 verwendet.

15.3 Sicherheitskonzept

Um die Sicherheit im Umbau der Filter sicherzustellen, wird ein Sicherheitskonzept benötigt. Das Ziel dieses Konzeptes ist es, mögliche Angriffe aufzuführen und die Blockade dieser Angriffe zu dokumentieren.

15.3.1 SQL-Injection

Die einzige Benutzereingabe, welche in dieser IPA auftritt, ist die Texteingabe im Filterkriterium "Felder". Da die Eingabe des Benutzers nicht direkt in die Postgres Datenbank gespeichert wird, ist diese Eingabe nicht für eine SQL-Injection gefährdet. Selbst wenn die SQL-Abfrage direkt gemacht würde, verhindert das ORM ActiveRecord mit seinen Standardmethoden, dass schädliche Eingaben abgespeichert werden. Dies geschieht unter anderem durch das Escapen der Strings.

15.3.2 Cross-Site Scripting

Da in dieser Erweiterung Benutzereingaben an das Rails-Backend gesendet werden, muss der Cross-Site Scripting Angriff ebenfalls berücksichtigt werden. Rails selbst bietet dafür einen eingebauten Abwehrmechanismus. Ein solcher XSS-Angriff kann folgendermassen aussehen:

```
1 <h2>Welcome <script>alert("This is a XSS attack!")</script></h2>
```

Standardmässig escaped Rails diese Eingaben und ändert die Spezialbuchstaben. So wird aus der Eingabe oben:

```
1 <h2>Welcome &lt;script&gt;alert\  
2 (&quot;This is a XSS attack!&quot;)&lt;/script&gt;</h2>
```

15.3.3 URL Interpretation

Bei der URL-Interpretation fabriziert der Angreifer eine URL, um damit auf die persönlichen Daten eines Benutzers zuzugreifen. Dabei kann der Angreifer

versuchen, die URL zu erraten. Dieser Angriff wird in Hitobito mit dem Gem can-can-can verhindert. Mit diesem Gem wird sichergestellt, dass der Absender der Anfrage die nötigen Berechtigungen für das Einsehen der Informationen hat. Die Prüfung der Berechtigungen sieht wie folgt aus:

```
1 class Ability include CanCan::Ability
2   def define_root_abilities
3     can :manage, :all
4     # root cannot change her email, because this is what makes her root.
5     cannot :update_email, Person do |p|
6       p.root?
7     end
8   end
end
```

15.3.4 Kommunikation HTTP/S

Die Umgebungen auf der Integration und Produktion kommunizieren via HTTPS. Somit ist die verschlüsselte Kommunikation beim Transfer von produktiven Daten gesichert.

15.4 Fehlerbehandlungskonzept

Bei der Entwicklung und während der Laufzeit können stets Fehler oder nicht vorgesehene Probleme entstehen. Im folgenden Abschnitt wird dokumentiert, wie mit diesen Fällen umgegangen wird.

15.4.1 Nutzereingabe

Bei der Nutzereingabe des Users werden keine möglichen Exceptions erwartet. Der Benutzer kann im Filter, in der Suche nach einem bestimmten Text alles eingeben, ohne Einschränkungen. Mögliche Angriffe werden gemäs den definierten [Sicherheitsmassnahmen](#) behandelt.

15.4.2 Laufzeitfehler

Tritt in der Applikation ein Laufzeit Fehler auf, wird dies sowohl in den Log,s wie in der Sentry Umgebung von Hitobito aufgezeigt. Im Sentry werden zusätzlich die aufgetretenen Execptions gesammelt, um den Entwicklern eine

Übersicht über allfällige Bugs zu geben. Gesammelte Exceptions können einem Entwickler zugewiesen oder wenn sie gefixed wurden, vom Sentry entfernt werden. Für diese IPA ist keine Modifizierung an der Sentry Umgebung nötig.

15.4.3 Exception Handling

In Ruby können Exception mit `rescue` abgefangen werden. Das folgende Beispiel macht das Exceptionhandling anhand eines 403 Fehlers bei Zugriff auf einen nicht erlaubten Endpoint sichtbar.

```
1 rescue_from CanCan::AccessDenied do |exception|
2   respond_to do |format|
3     format.json do
4       render json: {status: 403, error:
5         I18n.t("devise.failure.not_permitted_to_view_page")},
6         status: 403
7     end
8     format.all do
9       raise exception unless Rails.env.production?
10      redirect_to root_path, alert:
11        I18n.t("devise.failure.not_permitted_to_view_page")
12    end
13  end
14 end
```

Sobald die `AccessDenied` Exception geworfen wird, wird diese vom `ApplicationController` in Hitobito abgefangen. Im Rahmen dieser IPA kann diese Exception vorkommen, wenn ein Benutzer versucht, die Filterung von Personen einzusehen, auf welche er keine Berechtigungen hat. Im Rahmen dieser IPA wird auf das bestehende Exceptionhandling zurückgegriffen. Es wird kein zusätzliches Exceptionhandling benötigt.

15.5 Testsetup

Um im Hitobito Tests ausführen zu können wird folgendes Setup benötigt:

- Bash-Konsole
- Docker oder Docker Desktop bei Windows als OS.
- Geklontes Github Repository von <https://github.com/hitobito/development.git>.
- Core Wagon wurde in das Verzeichnis `/app` unter dem geklonten Development Repository eingefügt. Core Wagon kann unter <https://github.com/hitobito/hitobito.git> geklont werden.
- Generic Wagon wurde in das Verzeichnis `/app` unter dem geklonten Development Repository eingefügt. Generic Wagon kann unter https://github.com/hitobito/hitobito_generic.git geklont werden.

Besteht das beschriebene Setup, muss zur Ausführung der Tests, in das Development Repository navigiert werden. Dort müssen folgende Befehle der Reihe nach, wie hier beschrieben, ausgeführt werden:

- `bin/dev-env.sh` - Startet Hitobito Konsole
- `hit test prep` - Bereitet Tests vor, kompiled Assets
- `hit test` - Führt Migrationen durch, bereitet Testumgebung vor
- `rspec <path_to_tests>` Variable in Klammern muss durch den Pfad zu den Tests ersetzt werden. Wird ein Ordner unter dem Pfad angegeben, werden alle Tests unter diesem Ordner ausgeführt.

15.6 Testkonzept

15.6.1 Testinfrastruktur

Es gibt zwei Arten von Tests, welche in dieser Arbeit relevant sind:

- **Feature Tests:** Testet eine Funktion über die Benutzerschnittstelle
- **Manuelles Testen:** Testet das gesamte Feature

Hitobito verwendet für das Ausführen lokaler Tests RSpec 3.13.0. Für die manuellen Tests wird Firefox 80.0 (64-bit) verwendet. Als Gerät wird ein Laptop mit Pop!_OS 22.04 LTS verwendet.

Die bestehenden Tests, welche die Funktionalität der gesamten Applikation sicherstellen, werden für das Testing vernachlässigt. Es werden keine Anpassungen an diesen Tests vorgenommen. Die bestehenden Tests werden aus zeitlichen Gründen bei Fehlschlägen auskommentiert und ignoriert. Im Rahmen dieser Arbeit, sind ausschliesslich die selbstverfassten Tests relevant.

15.6.2 Fehlerklassen

Bezeichnung	Fehlerklasse	Beschreibung
FK0	Fehlerfrei	Keine Fehler
FK1	Nicht erfolgsgefährdend	Kleine Fehler, beeinträchtigen Funktion nur bedingt.
FK2	Erfolgsgefährdend	Fehler welche die Funktion beeinträchtigen.

Tabelle 15.9: Fehlerklassen

15.6.3 Manuelle Tests

Die manuellen Tests werden lokal, mit den Testdaten von Hitobito durchgeführt. Die Testdaten können in der Hitobito-Konsole mit `hit rails wagon seed` eingespielt werden. Bei Fragen zur Aktivierung der Hitobito Konsole, den [Beschrieb zum Testsetup](#) einsehen.

Mit dem Generic Wagon bietet sich ein Benutzer-Account für das Login an:

Bezeichnung	Username	Passwort	Berechtigungen
Admin	admin@hitobito.ch	demo	Administrator mit vollem Zugriff

Tabelle 15.10: Accounts für manuelle Tests

Bei den manuellen Tests muss stets einer der oben beschriebenen Accounts verwendet werden. In den Testszenarien wird der Account mit “Als {Account} anmelden” beschrieben. { Account } steht hierbei als Platzhalter für den Account des jeweiligen Benutzer. Das Login basiert auf folgendem Ablauf:

- Navigation auf `localhost:3000/users/sign_in`
- Anmeldeinformationen des Accounts eingeben
- Auf Anmelde-Button klicken

Testfall Nr. 1	
Testname	Personenfilter anzeigen
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none">• Nicht funktionale Anforderung 6• Funktionale Anforderung 1• Funktionale Anforderung 4
Voraussetzungen	Ein eigener Personenfilter wurde erfasst und gespeichert
Testszenario	<ul style="list-style-type: none">• Als Admin anmelden• Mittels Reiter “Personen” auf Personenliste navigieren• Maus Im Filter Dropdown über dem Namen des Filters positionieren• Erscheinende Option “Bearbeiten” auswählen
Erwartetes Resultat	<ul style="list-style-type: none">• Alle Filterkriterien und deren Bedingungen werden angezeigt

Tabelle 15.11: Testfall 1

Testfall Nr. 2	
Testname	Filterkriterium "Tags" bearbeiten
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none"> • Nicht funktionale Anforderung 6 • Funktionale Anforderung 1 • Funktionale Anforderung 3
Voraussetzungen	Ein eigener Personenfilter wurde erfasst und gespeichert
Testszenario	<ul style="list-style-type: none"> • Als Admin anmelden • Mittels Reiter "Personen" auf Personenliste navigieren • Maus Im Filter Dropdown über dem Namen des Filters positionieren • Erscheinende Option "Bearbeiten" auswählen • Auf Box des Filterkriteriums "Tags" navigieren • Bearbeiten-Button in der oberen rechten Ecke klicken • "Puzzle ITC" als "Tag" bei dem Eingabefeld "Einschliessen" hinzufügen • Speichern-Button innerhalb der Box des Filterkriteriums klicken
Erwartetes Resultat	<ul style="list-style-type: none"> • Box des Filterkriteriums "Tags" wird mit dem Tag "Puzzle ITC" ergänzt.

Tabelle 15.12: Testfall 2

Testfall Nr. 3	
Testname	Filterkriterium "Rollen" bearbeiten
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none"> • Nicht funktionale Anforderung 6 • Funktionale Anforderung 1 • Funktionale Anforderung 3
Voraussetzungen	Ein eigener Personenfilter wurde erfasst und gespeichert
Testszenario	<ul style="list-style-type: none"> • Als Admin anmelden • Mittels Reiter "Personen" auf Personenliste navigieren • Maus Im Filter Dropdown über dem Namen des Filters positionieren • Erscheinende Option "Bearbeiten" auswählen • Auf Box des Filterkriteriums "Rollen" navigieren • Bearbeiten-Button in der oberen rechten Ecke klicken • "Hauptebene" bei dem Eingabefeld "Gruppen" hinzufügen • "Region/Kanton - Präsident*in" bei dem Eingabefeld "Rollen" hinzufügen • "Aktive" beim Eingabefeld "Gültigkeit" erfassen • "18.05.2006" beim ersten Eingabefeld neben dem Stichdatum hinzufügen

Tabelle 15.13: Testfall 3

Testszenario	<ul style="list-style-type: none">• “20.05.2006” beim zweiten Eingabefeld neben dem Stichdatum hinzufügen• Speichern-Button innerhalb der Box des Filterkriteriums klicken
Erwartetes Resultat	<ul style="list-style-type: none">• Box des Filterkriteriums “Rollen” wird mit den eingegebenen Informationen ergänzt.

Tabelle 15.14: Testfall 3

Testfall Nr. 4	
Testname	Filterkriterium “Qualifikationen” bearbeiten
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none"> • Nicht funktionale Anforderung 6 • Funktionale Anforderung 1 • Funktionale Anforderung 3
Voraussetzungen	Ein eigener Personenfilter wurde erfasst und gespeichert
Testszenario	<ul style="list-style-type: none"> • Als Admin anmelden • Mittels Reiter “Personen” auf Personenliste navigieren • Maus Im Filter Dropdown über dem Namen des Filters positionieren • Erscheinende Option “Bearbeiten” auswählen • Auf Box des Filterkriteriums “Qualifikationen” navigieren • Bearbeiten-Button in der oberen rechten Ecke klicken • “Leitung” bei dem Eingabefeld “Qualifikationen” hinzufügen • “Person hat alle” bei der Radio-Buttons Gruppe “Kriterium” answählen • “Gültige” beim Eingabefeld “Gültigkeit” auswählen • Das Eingabefeld “Stichdatum” leer lassen

Tabelle 15.15: Testfall 4

Testfall Nr. 4	
Testszenario	<ul style="list-style-type: none">• Speichern-Button innerhalb der Box des Filterkriteriums klicken
Erwartetes Resultat	<ul style="list-style-type: none">• Box des Filterkriteriums "Qualifikationen" wird mit den eingegebenen Informatinen ergänzt.

Tabelle 15.16: Testfall 4

Testfall Nr. 5	
Testname	Filterkriterium "Felder" bearbeiten
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none"> • Nicht funktionale Anforderung 6 • Funktionale Anforderung 1 • Funktionale Anforderung 3
Voraussetzungen	Ein eigener Personenfilter wurde erfasst und gespeichert
Testszenario	<ul style="list-style-type: none"> • Als Admin anmelden • Mittels Reiter "Personen" auf Personenliste navigieren • Maus Im Filter Dropdown über dem Namen des Filters positionieren • Erscheinende Option "Bearbeiten" auswählen • Auf Box des Filterkriteriums "Felder" navigieren • Bearbeiten-Button in der oberen rechten Ecke klicken • "Alter" bei dem Eingabefeld "Attribut" hinzufügen • "42" bei dem Eingabefeld "Wert" eingeben • ">=" beim Eingabefeld "Genaugikeit" auswählen • Auf Hinzufügen-Button klicken • "Firmennamen" bei dem zweiten Eingabefeld "Attribut" hinzufügen

Tabelle 15.17: Testfall 5

Testfall Nr. 5	
Testszenario	<ul style="list-style-type: none">• “Puzzle ITC” bei dem zweiten Eingabefeld “Wert” eintragen• Speichern-Button innerhalb der Box des Filterkriteriums klicken
Erwartetes Resultat	<ul style="list-style-type: none">• Box des Filterkriteriums “Felder” wird mit den eingegebenen Informationen ergänzt.

Tabelle 15.18: Testfall 5

Testfall Nr. 6	
Testname	Sprache einschränken
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none"> • Funktionale Anforderung 5
Voraussetzungen	Ein eigener Personenfilter wurde erfasst und gespeichert
Testszenario	<ul style="list-style-type: none"> • Als Admin anmelden • Mittels Reiter “Personen” auf Personenliste navigieren • Maus Im Filter Dropdown über dem Namen des Filters positionieren • Erscheinende Option “Bearbeiten” auswählen • Auf Box des Filterkriteriums “Felder” navigieren • Bearbeiten-Button in der oberen rechten Ecke klicken • “Sprache” bei dem Eingabefeld “Attribut” hinzufügen • “Deutsch” bei dem Eingabefeld “Wert” eingeben • Speichern-Button innerhalb der Box des Filterkriteriums klicken
Erwartetes Resultat	Box des Filterkriteriums wird mit Bedingung zur Sprache ergänzt

Tabelle 15.19: Testfall 6

Testfall Nr. 7	
Testname	Filterkriterium "Tags" hinzufügen
Testmethode	Manuell
Anforderung	<ul style="list-style-type: none"> • Funktionale Anforderung 2
Voraussetzungen	Keine
Testszenario	<ul style="list-style-type: none"> • Als Admin anmelden • Mittels Reiter "Personen" auf Personenliste navigieren • Im Filter Dropdown Option "Neuer Filter..." auswählen • Hinzufügen-Button in der unteren rechten Ecke klicken • "Tags" aus dem Dropdown auswählen
Erwartetes Resultat	<ul style="list-style-type: none"> • Box des Filterkriteriums "Tags" erscheint • Option "Tags" ist aus dem Dropdown zum Hinzufügen von Filterkriterien verschwunden

Tabelle 15.20: Testfall 7

15.6.4 Automatisierte Tests

Alle Tests werden mit Testdaten ausgeführt, diese werden entweder erst per Fixtures gesetzt oder im Test dynamisch generiert. In den Testfällen werden teils auf Testdaten verwiesen, welche hier dokumentiert sind:

Bezeichnung	Bemerkung
top_leader	Admin Rechte, hat Zugriff auf Personen- und Abonnementenfilterung
bottom_member	Kein Zugriff auf Personen- und Abonnementenfilterung

Tabelle 15.21: Testdaten

Feature Test

Die Feature Tests werden mittels `$ rspec spec/features` ausgeführt

Testfall Nr. 8	
Testname	Erfolgreiche Filterung nach Tags
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Funktionale Anforderung 2
Voraussetzungen	<code>sign_in top_leader</code>
Testszenario	<ul style="list-style-type: none">• Personenlisten nach Tags filtern
Parameter	<ul style="list-style-type: none">• Tagname• Boolean Wert: Tag einschliessen oder ausschliessen
Erwartetes Resultat	<ul style="list-style-type: none">• Alle Personen in der resultierenden Personenliste weisen das gefilterte Tag auf

Tabelle 15.22: Testfall 8

Testfall Nr. 9	
Testname	Erfolgreiche Filterung nach Rollen
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Funktionale Anforderung 1
Voraussetzungen	sign_in top_leader
Testszenario	<ul style="list-style-type: none">• Personenlisten nach Rollen filtern
Parameter	<ul style="list-style-type: none">• Gruppenname• Rollennamen• Gültigkeit• Stichdatum
Erwartetes Resultat	<ul style="list-style-type: none">• Alle Personen in der resultierenden Personenliste weisen die gefilterte Rolle auf

Tabelle 15.23: Testfall 9

Testfall Nr. 10	
Testname	Erfolgreiche Filterung nach Qualifikationen
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none"> • Funktionale Anforderung 1
Voraussetzungen	sign_in top_leader
Testszenario	<ul style="list-style-type: none"> • Personenlisten nach Rollen filtern
Parameter	<ul style="list-style-type: none"> • Name der zu filternden Qualifikationen • Kriterium (Person hat alle oder mind. eine der Qualifikationen) • Gültigkeit • Stichdatum
Erwartetes Resultat	<ul style="list-style-type: none"> • Alle Personen in der resultierenden Personenliste weisen die gefilterte Qualifikationen auf

Tabelle 15.24: Testfall 10

Testfall Nr. 11	
Testname	Erfolgreiche Filterung nach Feldern
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none"> • Funktionale Anforderung 1
Voraussetzungen	sign_in top_leader
Testszenario	<ul style="list-style-type: none"> • Personenlisten nach Feldern filtern
Parameter	<ul style="list-style-type: none"> • Name der zu filternden Felder • Wert der Felder • Genauigkeit
Erwartetes Resultat	<ul style="list-style-type: none"> • Alle Personen in der resultierenden Personenliste weisen die gefilterten Felder auf

Tabelle 15.25: Testfall 11

Testfall Nr. 12	
Testname	Mitglied hat keinen Zugriff auf die Personenfilterung
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Nicht funktionale Anforderung 3
Voraussetzungen	sign_in bottom_member
Testszenario	<ul style="list-style-type: none">• Auf Personenliste navigieren
Parameter	Keine
Erwartetes Resultat	<ul style="list-style-type: none">• Personenliste wird dem Benutzer nicht angezeigt

Tabelle 15.26: Testfall 12

Testfall Nr. 13	
Testname	XSS-Angriff
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Nicht funktionale Anforderung 1
Voraussetzungen	sign_in top_leader
Testszenario	<ul style="list-style-type: none">• Personenlisten nach Feldern filtern• XSS-Eingabe tätigen
Parameter	<ul style="list-style-type: none">• Name der zu filternden Felder• Javascript Befehl (XSS-Eingabe)
Erwartetes Resultat	<ul style="list-style-type: none">• XSS-Eingabe wird als String interpretiert

Tabelle 15.27: Testfall 13

Testfall Nr. 14	
Testname	Globale Bedingungen der Abonnemente können angepasst werden.
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none"> • Funktionale Anforderung 1
Voraussetzungen	sign_in top_leader
Testszenario	<ul style="list-style-type: none"> • Globale Bedingungen in Abonnementen erfassen
Parameter	<ul style="list-style-type: none"> • Name der zu filternden Felder • Wert der Felder • Genauigkeit
Erwartetes Resultat	<ul style="list-style-type: none"> • Export der Abonnenten als CSV beinhaltet alle Personen, welche die globalen Bedingungen erfüllen.

Tabelle 15.28: Testfall 14

Testfall Nr. 15	
Testname	Mitglied hat keinen Zugriff auf die Filterung der Abonnemente
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Nicht funktionale Anforderung 3
Voraussetzungen	sign_in bottom_member
Testszenario	<ul style="list-style-type: none">• Auf globale Bedingungen der Abonnemente navigieren
Parameter	Keine
Erwartetes Resultat	<ul style="list-style-type: none">• Globale Bedingungen können vom Benutzer nicht bearbeitet werden.

Tabelle 15.29: Testfall 15

Testfall Nr. 16	
Testname	Personenfilter kann abgespeichert werden
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Funktionale Anforderung 1
Voraussetzungen	sign_in top_leader
Testszenario	<ul style="list-style-type: none">• Neuen Personenfilter erfassen• Personenfilter abspeichern
Parameter	<ul style="list-style-type: none">• Name des Personenfilters• Parameter für das jeweilige Filterkriterium
Erwartetes Resultat	<ul style="list-style-type: none">• Personenfilter wurde abgespeichert und kann wiederverwendet werden

Tabelle 15.30: Testfall 16

Testfall Nr. 17	
Testname	Mitglied kann Personenfilter nicht abspeichern
Testmethode	Rspec Feature Test
Anforderung	<ul style="list-style-type: none">• Nicht funktionale Anforderung 4
Voraussetzungen	sign_in bottom_member
Testszenario	<ul style="list-style-type: none">• Auf Personenliste navigieren
Parameter	Keine
Erwartetes Resultat	<ul style="list-style-type: none">• Personenlisten werden nicht angezeigt und Benutzer kann somit keinen neuen Personenfilter erfassen.

Tabelle 15.31: Testfall 17

15.6.5 Begründung der Testwahl

Die Idee hinter dem verfassten Testkonzept, ist leicht auszumachende visuelle Änderungen manuell zu testen. Überprüfungen, welche die Richtigkeit der Filterung sicherstellen, werden mit Featuretests abgehandelt, da die Richtigkeit bei einem grossen Datensatz durch manuelles Testen nicht garantiert werden kann.

Des Weiteren wird vermerkt, dass folgende Anforderungen nicht durch Tests abgedeckt wurden:

- Nicht funktionale Anforderung 2
- Nicht funktionale Anforderung 5

Da diese Anforderungen nicht durch Features oder manuelle Tests abgedeckt werden können, werden diese Anforderungen durch regelmässige Eigenüberprüfung des Entwicklers sichergestellt. Abweichungen werden im Arbeitsjournal dokumentiert.

16 Abschluss Sprint Initialisierung

An dieser Stelle endet der erste Sprint welcher sich mit der Initialisierung und der Konzeption der Arbeit befasste. Der Sprint dauerte vom 04.03.2025 bis am 11.03.2025.

Title	Status
<div> <div>Refinement</div> <div>1</div> <div>Estimate: 0</div> <div>This is ready to be refined</div> </div>	
<div> <div>1</div> <div>Task / Standards #2</div> </div>	Refinement
+ Add item	
<div> <div>Done</div> <div>10</div> <div>Estimate: 0</div> <div>This has been completed</div> </div>	
<div> <div>2</div> <div>Testconcept #11</div> </div>	Done
<div> <div>3</div> <div>System Concept #10</div> </div>	Done
<div> <div>4</div> <div>Exception Handling Concept #12</div> </div>	Done
<div> <div>5</div> <div>Security Concept #13</div> </div>	Done
<div> <div>6</div> <div>Analysis #6</div> </div>	Done
<div> <div>7</div> <div>Concept of Use #9</div> </div>	Done
<div> <div>8</div> <div>Needs assessment #7</div> </div>	Done
<div> <div>9</div> <div>Versioning and Backup #5</div> </div>	Done
<div> <div>10</div> <div>Risk Analysis #4</div> </div>	Done
<div> <div>11</div> <div>Management Method #3</div> </div>	Done

Abbildung 16.1: Backlog nach Sprint Initialisierung

Im Sprint konnten ausgenommen einer User Story alle Stories umgesetzt werden. Die User Story “Tasks / Standards” konnte nicht vollständig umgesetzt werden. Grund dafür ist, dass die Abschnitte “Schutzbedarfsanalyse”, “Einführung” und “Risikoanalyse für Projektrisiken” in der Dokumentation noch nicht beschrieben wurden. Gemäss dem Vorgehen wird die User Story zurück in die Spalte “Refinement” gesetzt und im kommenden Sprint Planning im neuen Sprint eingeteilt.

16.1 Sprint Diagramme

16.1.1 Burn up



Abbildung 16.2: Burn up nach Sprint 1

Anhand des Burn ups sieht man, dass die meisten User Stories bis zum Sprintabschluss abgeschlossen werden konnten. Die Anzahl der User Stories hat sich am 07.03.2025 erhöht, da die User Story “Entwurf” aufgeteilt wurde. Dies geschah auf Anweisung des Hauptexperten Lorenz Müller.

16.1.2 User Story Chart

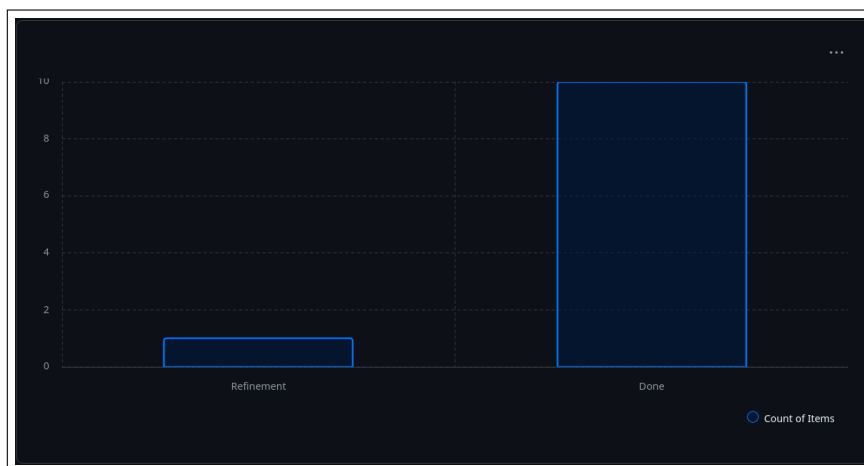


Abbildung 16.3: Charts nach Sprint 1

Als Fazit werden im obenstehenden Bild die Verteilung der Status aller User Stories im Sprint angezeigt.

17 Umsetzung

17.1 Schnittstellen

Im Entwurf wurde definiert, dass pro Filterkriterium eine Schnittstelle (Endpoint) definiert wird. Während des Implementierens wurde diese Definition geändert. Grund dafür war die Erkenntnis, dass alle Filterkriterien über die gleichen Schnittstelle geladen werden können. Die Aufgabe der Schnittstelle ist es, ein Partial zu rendern. Welches Partial gerendert werden muss, wird über einen mitgegebenen Parameter entschieden, welcher den Namen des Filterkriteriums trägt.

Die dafür implementierte Controller-Action wurde so definiert:

```
1  def filter_criterion
2    compose_role_lists
3    possible_tags
4    @filter_criterion = params[:filter_criterion]
5    if @filter_criteria.include?(@filter_criterion.to_sym)
6      respond_to do |format|
7        format.turbo_stream { render 'create', status: :ok }
8      end
9    end
10  end
```

Das Filterkriterium wird über den Parameter “filter_criterion” identifiziert. Die Action wird bei jedem Request auf die Route `http://localhost:3000/de/groups/1/people_filters/:filter_criterion` ausgeführt. Die Route wurde entsprechend in der Datei `routes.rb` hinterlegt

```
1  # Turbo Routes
2  get 'people_filters/:filter_criterion' => 'people_filters#filter_criterion'
```

Ein Beispiel für einen Request auf diese Schnittstelle ist folgende Abfrage:

`https://localhost:3000/people_filters/tag`

Der Request kann folgendermassen aufgelöst werden:

Turbo Request	
Host	https://localhost:3000
Ressource	/people_filters
Wert des Parameters filter_criterion	/tag
Funktion	Rendert das Tag-Partial

Tabelle 17.1: Turbo Request

Nach dem Muster des oben beschriebenen Requests können sämtliche Partialas weiterer Kriterien gerendert werden. Um das Partial eines anderen Filterkriteriums zu rendern, muss lediglich der Wert des Parameters “filter_criterion” auf den Namen des anderen Filterkriteriums angepasst werden. Durch diese Implementation, müssen nicht wie ursprünglich geplant vier verschiedene Schnittstellen implementiert werden sondern nur eine Schnittstelle, welche die Unterscheidung mit einem Parameter regelt.

17.2 Hinzufügen-Button

Während des Entwickelns, traten auch beim Hinzufügen-Button Änderungen am Konzept auf. Das Hauptproblem war, dass der Button, nachdem alle Filterkriterien hinzugefügt wurden, nicht mit den Turbostreams entfernt werden konnte. Als Alternative wurde der Stimulus Controller verwendet, der in der Variante 2 des Variantenentscheides vorgeschlagen wurde.

Der Stimulus Controller wurde nach folgendem Schema aufgebaut:

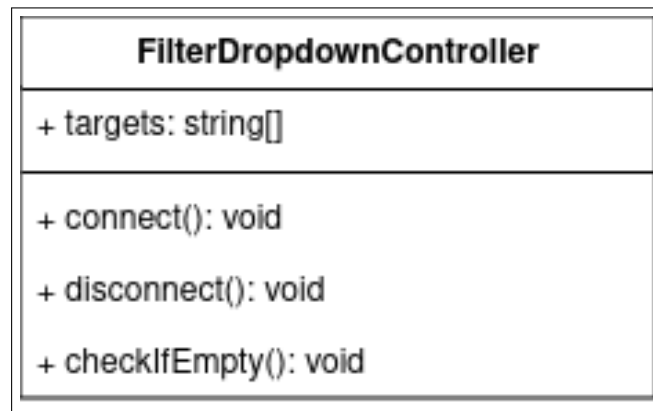


Abbildung 17.1: FilterDropdownController

Der Controller überwacht Änderungen am Hinzufüge-Dropdown. Wenn eine Option im Dropdown durch einen Turbostream entfernt wird, dann bekommt der Stimulus Controller diese Änderung mit. Der Controller prüft anschliessend ob das Dropdown andere Optionen besitzt. Wenn nicht, wird der Hinzufüge-Button entfernt.

Der Controller kann die Änderungen nur erkennen, wenn dieser korrekt im Template referenziert wird:

```
1 %div.dropdown.ml-3{"data-controller": "filter-dropdown"}
2   %button.btn.dropdown-toggle{ type: 'button', "data-bs-toggle": "dropdown",
3     "aria-expanded": "false"}
4     %i.fa.fa-plus
5     %ul.dropdown-menu{"data-filter-dropdown-target": "menu"}
6       - @filter_criteria.each do |filter_criterion|
7         = link_to t("people_filters.#{filter_criterion}.title"),
          "/groups/#{@group.id}/people_filters/#{filter_criterion}",
          class: "dropdown-item", id: "dropdown_option_#{filter_criterion}", data:
          {turbo_stream: true}
```

Auf Zeile 1 wird mit dem “data-controller” der Stimulus Controller registriert. Sobald diese Referenz einem Controller zugeordnet werden kann, wird die `connect()`-Methode dieses Controllers ausgeführt.

```
1 connect() {
2   this.observer = new MutationObserver(this.checkIfEmpty.bind(this));
3   if (this.hasMenuTarget) {
4     this.observer.observe(this.menuTarget, { childList: true });
5   }
6 }
```

In diesem Fall wird ein `MutationsObserver` registriert, welcher prüft ob das registrierte Element leer ist. Damit dieser `MutationsObserver` auf die Dropdownliste der Filterkriterien registriert werden kann, muss die Dropdownliste mit allen Filterkriterien als “Target” erfasst werden.

Dieses Target kann im Controller folgendermassen definiert werden:

```
1 static targets = ["menu"];
```

Wurde das Target sowohl im Stimulus Controller wie im Template mit dem gleichen Namen definiert, in diesem Fall “menu”, kann das Target innerhalb des Stimulus Controller mit `this.menuTarget` angesprochen werden. Im Falle der Implementation, wird der MutationsObserver auf diesem Target registriert. Wann immer eine Änderung an diesem Target durchgeführt wird, wird nach der Registrierung die Methode ‘`checkIfEmpty()`’ ausgeführt:

```
1  checkIfEmpty() {  
2    if (this.menuTarget.children.length === 0) {  
3      this.element.remove();  
4    }  
5  }
```

Diese Methode entfernt den Hinzufüge-Button und das Dropdown mit allen Filterkriterien, sobald keine Optionen mehr darin verfügbar sind.

17.3 Turbostreams

Mittels der Turbostreams wurden die einzelnen Ansichten der Filterkriterien aktualisiert. Wie im Abschnitt “Schnittstellen” erklärt, waren im Entwurf ursprünglich für jedes Filterkriterium ein Turbostream angedacht. Durch die Verwendung nur einer Schnittstelle, wurde entsprechend nur ein generischer Turbostream für das Hinzufügen eines Filterkriteriums auf der Übersicht implementiert:

```
1  = turbo_stream.prepend "overview" do  
2    %div{id: "#{@filter_criterion}-configuration"}  
3    = render(layout: 'layouts/filter', locals: { type: @filter_criterion.to_sym  
4      }) do  
5        = render @filter_criterion  
6  = turbo_stream.remove "dropdown-option-#{@filter_criterion}"  
7  = turbo_stream.remove "filter-advice"
```

Der Turbostream identifiziert das Filterkriterium mit der “`@filter_criterion`” Variable. Anschliessend fügt der das Partial des identifizierten Filterkriteriums zur Übersicht hinzu und entfernt das Filterkriterium aus dem Hinzufüge-Dropdown. Zuletzt wird der Hinweis zur Benutzung des Hinzufüge-Dropdowns, bei noch nicht definierten Filterkriterien, entfernt.

Nebst dem Turbostream, welcher ein Filterkriterium zur Übersicht hinzufügt, wird ein Turbostream für die gegenteilige Aktion benötigt:

```
1 = turbo_stream.remove "#{@filter_criterion}-configuration"
2
3 = turbo_stream.append "filter-criteria-dropdown" do
4 = link_to t("people_filters.#{@filter_criterion}.title"),
5           "/groups/#{@group.id}/people_filters/#{@filter_criterion}",
           class: "dropdown-item", id: "dropdown-option-#{@filter_criterion}",
           data: {turbo_stream: true}
```

Dieser Turbostream entfernt das Filterkriterium aus der Übersicht und fügt es dem Hinzufüge-Dropdown hinzu. Die Aktion wird durch ein Mülleimer-Symbol auf der Box des Filterkriteriums ausgelöst:

Abbildung 17.2: Filterkriterium “Tag” mit Mülleimer-Symbol

Somit sind zwei Turbostreams für alle vier Filterkriterien verantwortlich. Welche der beiden Turbostreams ausgeführt werden muss, wird im Controller definiert. Die Definition basiert auf der HTTP-Methode des Requests.

17.3.1 Partial

Nebst der Implementation der Turbostreams wurden die Benutzerschnittstellen der Filterkriterien überarbeitet. Die Anpassungen wurden dem Mockup entsprechend durchgeführt. Alle Änderungen am Mockup wurden gemäss der Absprachen im Daily im Arbeitsjournal dokumentiert.

17.3.2 Name-Attribut auf dem Select-Feld

Um ein Dropdown in Ruby darzustellen, wird ein “select”-Element verwendet. Dieses Element wurde mehrfach für die Benutzerschnittstellen der Filterkriterien verwendet. Für das Filterkriterium “Felder” wird dieses Element über Rubycode generiert und nicht als HAML in einem Template hinterlegt. Während der Implementation wurde bekannt, dass dieses Element nicht mit der neuen Umsetzung funktioniert. Der Grund dafür lag beim Name-Attribut:

```
1 select(:filters, "attributes[#{time}][constraint]",
2     filters,
3     {selected: constraint},
4     html_options.merge(class:
5         'attribute_constraint_dropdown
6         ms-3 form-select form-select-sm'))
```

Nach auf der Zeile 1 wird mit “attributes[#{time}][constraint]” der “Name” dieses Elements definiert. Jedoch wird durch das Symbol “:filters” angegeben, dass es sich bei diesem Dropdown um ein Filter-Dropdown handelt.

Rails generiert das “Name” Attribut aus diesen zwei Angaben. Die Ausgabe des “Name” Attributs kommt nach dieser Generation so daher:

```
1 filters[attributes[time][constraint]] = value
```

Der Controller erwartet aber das Format:

```
1 filters[attributes][time][constraint] = value
```

Man notiere, dass in dem zweiten Beispiel die Werte “attributes”, “time” und “constraint” nicht im “filters” Array gespeichert sind. Um das richtige Format zu übermitteln, wurde das “Name” Attribut manuell in den HTML Optionen überschrieben:

```
1 html_options.merge(name: "filters[attributes][#{time}][constraint]")
```

Durch die Sicherstellung des korrekten Formates, kann das Filterkriterium “Felder” erneut vom Controller interpretiert und gespeichert werden.

17.4 Globale Bedingungen

Die globalen Bedingungen der Abonnenten und der Personenfilter, verwenden die gleichen Benutzerschnittstellen für das Filterkriterium “Felder”. Durch die Anpassung der Benutzerschnittstelle des Filterkriteriums, wurden die globalen Bedingungen indirekt auch angepasst.

Abbildung 17.3: Globale Bedingungen

Anders als beim Personenfilter, ist es nicht möglich, das Filterkriterium zu entfernen. Dies wurde so implementiert, um die alte Funktionalität nicht zu verändern sondern nur die Benutzerschnittstelle selbst anzupassen. Das Mülleimer-Symbol wie es bei den Filterkriterien des Personenfilters abgebildet ist, wurde durch eine Variable aus der Ansicht der globalen Bedingungen ausgeblendet:

```
1 = render(layout: 'layouts/filter', locals: { entry: @mailing_list, type:
  :attributes, delete: false}) do
```

Die Variable “delete” gibt an, ob das Symbol zur Löschung auf der Benutzerschnittstelle verfügbar sein soll. Beim Wert “false”, wird das Symbol ausgeblendet. Im Personenfilter ist die Variable stets auf “true” gesetzt.

17.5 Einsatz von KI-Modellen

Während des Implementierens wird das KI-Modell ChatGPT mit der gratis Version GPT-4o verwendet. Das Modell soll dazu verwendet werden geschriebenen Code zu vereinfachen, Hinweise zu aufkommenden Problemen zu geben und Syntax Fehler zu beheben. Die Nutzung des Modelles und die Prüfung der Antworten werden im weiteren Abschnitt beschrieben.

17.5.1 Eingabewerte

Bei der Verwendung von ChatGPT werden folgende Regeln für die Eingabewerte definiert:

- Alle Eingabewerte sind in Englisch zu schreiben.
- Es wird stets die Option zur Belegung der Quellen definiert, um die resultierenden Informationen anschliessend zu prüfen.
- Es werden keine sicherheitsrelevanten Informationen wie Passwörter oder SSH-Keys an das Modell weitergegeben.
- Bei der Verwendung des Modells für die Entscheidungsfindung, müssen die nötigen Informationen über das Umsystem angegeben werden.
- Bei der Verwendung des Modells für die Syntaxvalidierung, muss die Programmiersprache des Eingabewertes definiert werden.

17.5.2 Höhere Effizienz

Die höhere Effizienz durch die Verwendung des KI-Modells wird durch die Analyse von Syntaxfehlern gewährleistet. Als Beispiel und Beweis dafür ist folgendes Beispiel aus der Umsetzungsphase dieser IPA zu betrachten:

```
1 first(".btn.btn-primary", text: "Suchen").aclick
2 within(".table.table-striped.table-hover") do
3   # Head table row and table row of found user
4   expect(all("tr").size).to eq(2)
5   expect(page).to have_selector("#person_#{alice.id}")
6 end
```

Es wird die Zeile 1 betrachtet. Darin wird ein Button ausgemacht und anschliessend geklickt. Durch einen Tippfehler wurde anstatt die Methode “click” die Methode “aclick” angegeben, der Buchstabe “a” ist der Fehler. Dieser Fehler wurde in der IDE des Kandidaten nicht angezeigt weswegen der Fehler nicht ausgemacht werden konnte. Durch den Einsatz von ChatGPT wurde der Tippfehler erkannt und konnte beseitigt werden. Das KI-Modell hat in diesem Fall verhindert, Ressourcen für das Finden dieses Problems aufzuwenden und gewährt dem Kandidaten dadurch eine höhere Effizienz in der Arbeitsweise.

Mit den definierten Regeln für die Eingabewerte des Modells wurde die Syntaxvalidierung mit folgendem Eingabwert vorgenommen:

Setup:

Programming language: Ruby Framework: Ruby on Rails (RoR) Testing Framework: RSpec Test Type: Feature Test

```
1 I would like you validate the syntax of following code based on the above setup:
2
3 first(".btn.btn-primary", text: "Suchen").aclick
4 within(".table.table-striped.table-hover") do
5     # Head table row and table row of found user
6     expect(all("tr").size).to eq(2)
7     expect(page).to have_selector("#person_#{alice.id}")
8 end
```

Die entsprechende Ausgabe von ChatGPT war folgende:

Based on your setup-Ruby programming language, Ruby on Rails framework, RSpec for testing, and a feature test-the provided code has a minor syntax issue. The method `aclick` should be replaced with `click` to simulate a click action. Here's the corrected code:

```
1 first(".btn.btn-primary", text: "Suchen").click
2 within(".table.table-striped.table-hover") do
3     # Head table row and table row of found user
4     expect(all("tr").size).to eq(2)
5     expect(page).to have_selector("#person_#{alice.id}")
6 end
```

17.5.3 Höhere Effektivität

Während der Umsetzungsphase wurde ChatGPT die Meinung von ChatGPT zu den einzelnen Lösungsvarianten abgefragt. Das Ziel dahinter war, eine "externe" Einschätzung der Lösungen für das IPA-Produkt zu erhalten und potentielle Fehler in den Lösungsvarianten auszumachen.

Die Informationen wurden dem Modell mit folgendem Eingabewert übergeben:

Setup

Programming language: Ruby

Framework: Ruby on Rails (RoR), Hotwire

Application: Server Side Rendered

I would like you to to compare two of the described concepts below based on the above setup:

Problem: The filtering inside a community management system should be reworked. Instead of reloading the whole page when adding filters, the filter configuration should work with hotwire to only reload parts of the user interface. Two concepts were created to solve this problem:

Solution A: Every filter criterion has one turbo-stream. The partials of the filter-criterion are added to the overview via the turbo-stream append method. The user is able to trigger the turbostream with a dropdown which holds all filter criteria. Once the user clicks on one dropdown option, the turbostream is activated and removes the option from the dropdown. The user may also delete a filter criterion on the overview by click on a delete-symbol. Then, the reverse process is triggered.

Solution B: Each filter criterion in the ui is surrounded by a turbo-frame. The user is able to add a filter criterion by clicking on a option in a dropdown which holds all filter criteria. A stimulus controller removes the option then of the dropdown and replaces the turbo frame with the configuration of the chosen filter criterion.

Die Ausgabe von ChatGPT war folgende:

Recommendation

For the described filtering functionality, Solution B using Turbo Frames with Stimulus controllers is recommended. This approach allows each filter criterion to be independently managed and updated within its frame, providing a modular and maintainable structure. The use of Stimulus controllers enhances interactivity by handling client-side behaviors, such as removing options from the dropdown and updating frame content, without necessitating complex server-side logic.

This method aligns with best practices for progressively enhancing user interfaces, starting with server-side rendered responses and adding interactivity as needed.

Prüfung

Durch die Aktivierung der Option zur Quellenhinterlegung von ChatGPT wurde parallel zu dieser Antwort folgendes Video hinterlegt: [Turbostreams vs Turboframes](#). Während der Prüfung der Angaben, wurde durch das hinterlegte Video die Schwachstellen von Turboframes identifiziert:

- Turboframes können nur ein Frame aktualisieren
- Können nur Aktualisierungsaktionen auslösen, kein Hinzufügen oder Löschen
- Keine Backend-Aktualisierungen können ausgelöst werden.

Obwohl ChatGPT die Lösung A preferierte, wurde aus den Angaben aus dem Video klar, dass die Lösung B die bessere Lösungsvariante sein wird. Grund dafür ist, dass mehrere Aktionen zugleich ausgeführt werden müssen, dazu ist das Turboframe laut dem referenzierten Video von ChatGPT nicht fähig. In diesem Fall hat ChatGPT die falsche Entscheidung getroffen, doch durch die Prüfung der Antwort konnte ein richtiger Schluss aus den Informationen des KI-Modells gezogen werden.

17.5.4 Fazit

Bei der Verwendung des KI-Modells ist es wichtig dieses nur für die definierten Fälle und mit den definierten Eingabewerten zu verwenden. Weiter

wird im oberen Beispiel sichtbar, dass die Angaben von ChatGPT zu prüfen sind, um die richtigen Schlüsse für das eigene Projekt zu ziehen. Die Prüfung kann durch eine zusätzliche Recherche oder die Aktivierung der Hinterlegung von Quellen in ChatGPT durchgeführt werden. Die Hinterlegten Quellen sollten dennoch nochmals vom Benutzer geprüft werden, um sicherzustellen, dass die Informationen mit den Projektanforderungen übereinstimmen.

17.6 Testprotokoll

17.6.1 Manuelle Tests

Resultat Testfall Nr. 1	
Testname	Personenfilter anzeigen
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 6• Funktionale Anforderung 1• Funktionale Anforderung 4
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.2: Resultat Testfall 1

Resultat Testfall Nr. 2	
Testname	Filterkriterium "Tags" bearbeiten
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 6• Funktionale Anforderung 1• Funktionale Anforderung 3
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.3: Resultat Testfall 2

Resultat Testfall Nr. 3

Testname	Filterkriterium "Rollen" bearbeiten
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 6• Funktionale Anforderung 1• Funktionale Anforderung 3
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.4: Resultat Testfall 3

Resultat Testfall Nr. 4

Testname	Filterkriterium "Qualifikationen" bearbeiten
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 6• Funktionale Anforderung 1• Funktionale Anforderung 3
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.5: Resultat Testfall 4

Resultat Testfall Nr. 5

Testname	Filterkriterium "Felder" bearbeiten
Anforderungen	<ul style="list-style-type: none"> • Nicht funktionale Anforderung 6 • Funktionale Anforderung 1 • Funktionale Anforderung 3
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.6: Resultat Testfall 5

Resultat Testfall Nr. 6

Testname	Sprache einschränken
Anforderungen	<ul style="list-style-type: none"> • Funktionale Anforderung 5
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.7: Resultat Testfall 6

Resultat Testfall Nr. 7

Testname	Filterkriterium "Tags" hinzufügen
Anforderungen	<ul style="list-style-type: none"> • Funktionale Anforderung 5
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.8: Resultat Testfall 7

17.6.2 Feature Tests

Resultat Testfall Nr. 8	
Testname	Erfolgreiche Filterung nach Tags
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 2
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.9: Resultat Testfall 8

Resultat Testfall Nr. 9	
Testname	Erfolgreiche Filterung nach Rollen
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.10: Resultat Testfall 9

Resultat Testfall Nr. 10	
Testname	Erfolgreiche Filterung nach Qualifikationen
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.11: Resultat Testfall 10

Resultat Testfall Nr. 11	
Testname	Erfolgreiche Filterung nach Feldern
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.12: Resultat Testfall 11

Resultat Testfall Nr. 12	
Testname	Mitglied hat keinen Zugriff auf die Personenfilterung
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 3
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.13: Resultat Testfall 12

Resultat Testfall Nr. 13	
Testname	XSS-Angriff
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.14: Resultat Testfall 13

Resultat Testfall Nr. 14	
Testname	Globale Bedingungen der Abonnemente können angepasst werden.
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.15: Resultat Testfall 14

Resultat Testfall Nr. 15	
Testname	Mitglied hat keinen Zugriff auf die Filterung der Abonnemente.
Anforderungen	<ul style="list-style-type: none">• Nicht funktionale Anforderung 3
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.16: Resultat Testfall 15

Resultat Testfall Nr. 16	
Testname	Personenfilter kann abgespeichert werden.
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	18.03.2025
Testergebnis	Bestanden
Beschreibung	-
Fehlerklasse	FK0

Tabelle 17.17: Resultat Testfall 16

Resultat Testfall Nr. 17	
Testname	Mitglied kann Personenfilter nicht abspeichern.
Anforderungen	<ul style="list-style-type: none">• Funktionale Anforderung 1
Testperson	Marc Egli
Ausführungsdatum	-
Testergebnis	Nicht durchgeführt
Beschreibung	Während des Implementierens wurde bekannt wurde erkannt, dass dieser Testfall bereits durch den Testfall Nr. 12 abgedeckt wird. Hat der Benutzer keinen Zugriff auf die Personenfilterung, ist es ihm auch nicht möglich, einen Personenfilter abzuspeichern. Aus diesem Grund wird dieser Testfall verworfen.
Fehlerklasse	-

Tabelle 17.18: Resultat Testfall 17

17.6.3 Fazit

Beinahe alle Tests konnten problemfrei durchgeführt werden. Alle Anforderungen ausgenommen der nicht funktionalen Anforderungen zwei und fünf (wie dokumentiert), werden durch die Tests abgedeckt. Da die Qualität und Funktion der Arbeit mit den Tests sichergestellt werden konnte, werden keine weiteren Anpassungen vorgenommen.

17.7 Instruktion

Die Instruktion dient dem Benutzer zum Verständnis der Benutzerschnittstelle. Es ist eine Anleitung, welche den Benutzer bei der Arbeit mit der Benutzerschnittstelle unterstützen soll. Im weiteren Abschnitt wird die Instruktion für dieses IPA-Feature bereitgestellt und durch eine Durchführung verifiziert.

17.7.1 Anleitungen

Personenfilterung

In Hitobito ist es möglich alle Personen einer Ebene zu filtern. Navigieren Sie dazu zuerst auf die Gruppe auf welcher Sie die Personen filtern möchten. Dies können Sie über die Gruppenliste auf der linken Seite der Anwendung.

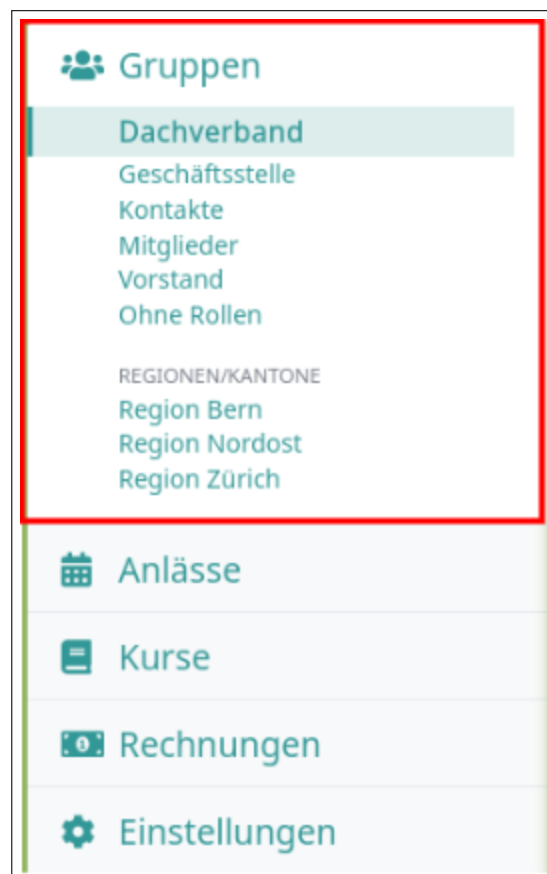


Abbildung 17.4: Gruppenliste

Haben Sie die Gruppe ausgewählt, können Sie über den Reiter “Personen” auf der Informationsseite alle Personen der Gruppe anzeigen.

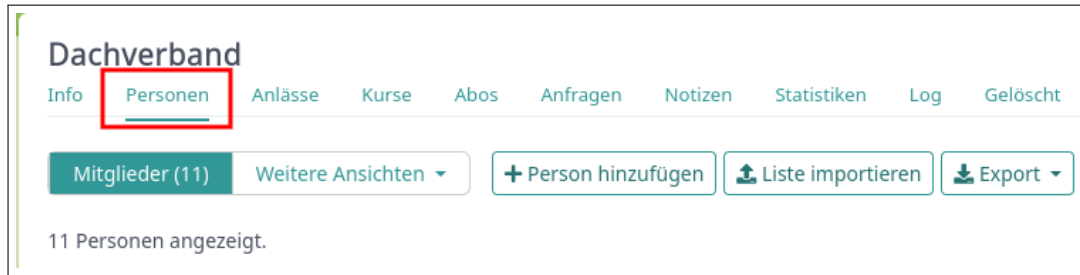


Abbildung 17.5: Personenreiter

Über das Dropdown “Weitere Ansichten” ist es Ihnen möglich einen Neuen Filter für die Personenliste zu erfassen. Klicken Sie dazu auf die Option “Neuer Filter...”.

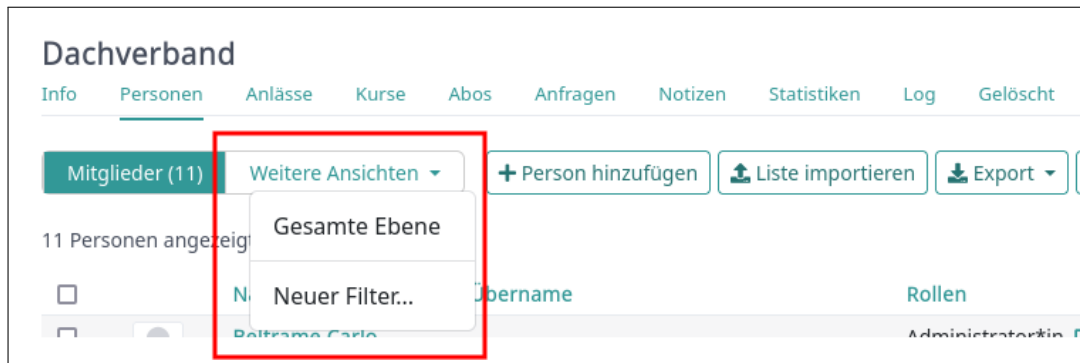


Abbildung 17.6: Neuer Filter erstellen

Auf der neuen Ansicht auf welcher Sie sich danach befinden, können Sie ihre Filter konfigurieren. Um ein neues Filterkriterium hinzuzufügen, navigieren Sie mit der Maus auf das “+” Zeichen. Alle auswählbaren Optionen werden ihnen als Dropdown unter dem “+” angezeigt.

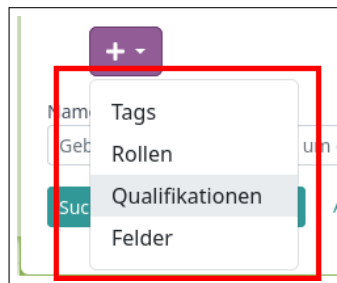


Abbildung 17.7: Filterkriterium hinzufügen

Das Filterkriterium wird nach dem Klick auf die Option im Dropdown als graue Box angezeigt. Innerhalb dieser Box finden sich diverse Eingabefelder zur Definition dieses Filterkriteriums. Das Filterkriterium kann auch wieder entfernt werden. Klicken Sie dazu auf das Mülleimer-Symbol in der rechten oberen Ecke der grauen Box.

Abbildung 17.8: Filterkriterium bearbeiten oder entfernen

Den Prozess des Hinzufügens und Löschs von Filterkriterien können Sie solange wiederholen, bis Sie die gewählten Einstellungen des Filters konfiguriert haben. Sind Sie mit der Konfiguration zufrieden, können sie den Personenfilter speichern oder nur danach suchen. Wenn Sie den Filter speichern möchten, muss diesem zuerst ein Name über das Eingabefeld zugewiesen werden.

Abbildung 17.9: Personenfilter speichern

Wenn Sie den Filter speichern, ist dieser unter dem Dropdown “Weitere Ansichten” in der Personenansicht der Gruppe verfügbar und kann mit einem Klick auf den Namen des Filters wiederverwendet werden. Wenn nur nach den Filterkriterien gesucht wurde, kann der Filter nicht wiederverwendet werden und der Filter wird einmalig auf die Personenliste angewendet.



Abbildung 17.10: Wiederverwendung eines gespeicherten Filters

Globale Bedingungen der Abonnenten

In Hitobito ist es möglich globale Bedingungen für Abonnenten zu definieren. Personen welche diese Bedingungen erfüllen, werden den Abonnements zugeteilt. Um auf die globalen Bedingungen eines Abonnements zu navigieren, muss zuerst eine Gruppe aus der Gruppenliste ganz links der Anwendung ausgewählt werden.

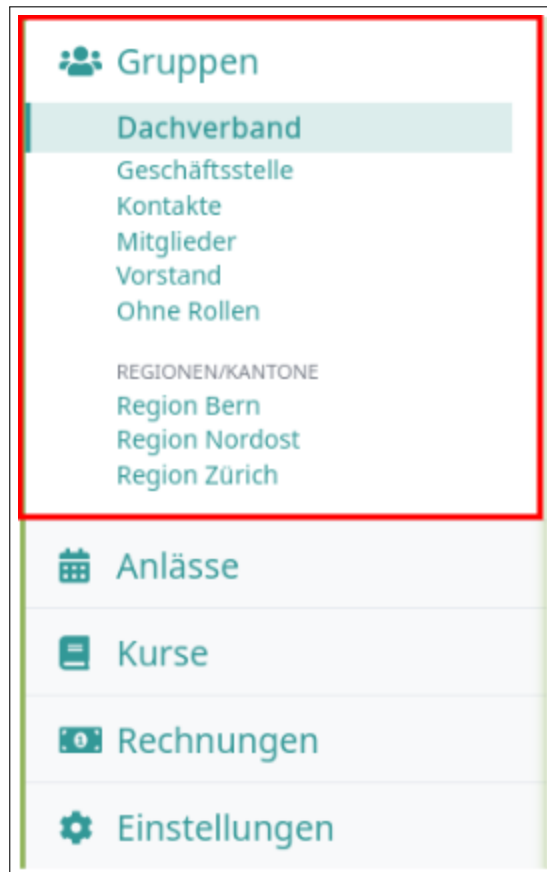


Abbildung 17.11: Gruppenliste

Klicken Sie anschliessend auf den Reiter “Abonnemente” um alle Abonnemente der Gruppe anzuzeigen. In der Liste der Abonnemente klicken Sie anschliessend auf das Abonnement, in welchem Sie die globalen Bedingungen anpassen möchten.



Abbildung 17.12: Abonnentenübersicht der Gruppe

In der Detailansicht des einzelnen Abonnements, klicken Sie auf den Reiter “Abonnente”. Anschliessend können Sie über das Bearbeiten-Symbol die globalen Bedingungen des Abonnements bearbeiten.

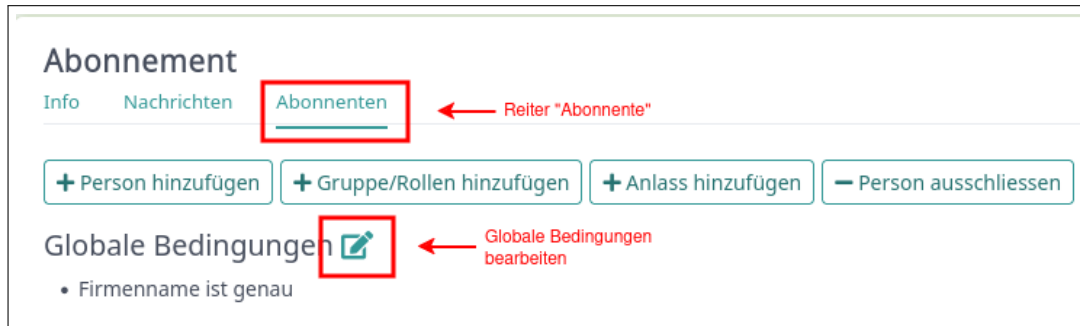


Abbildung 17.13: Detailansicht eines Abonnements

In der Konfiguration der globalen Bedingungen können anschliessend alle Personenattribute mit eine Dropdown hinzugefügt werden. Sind alle Konfigurationen zu ihrer Zufriedenheit, können sie die globalen Bedingungen über den “Speichern” Button speicher.

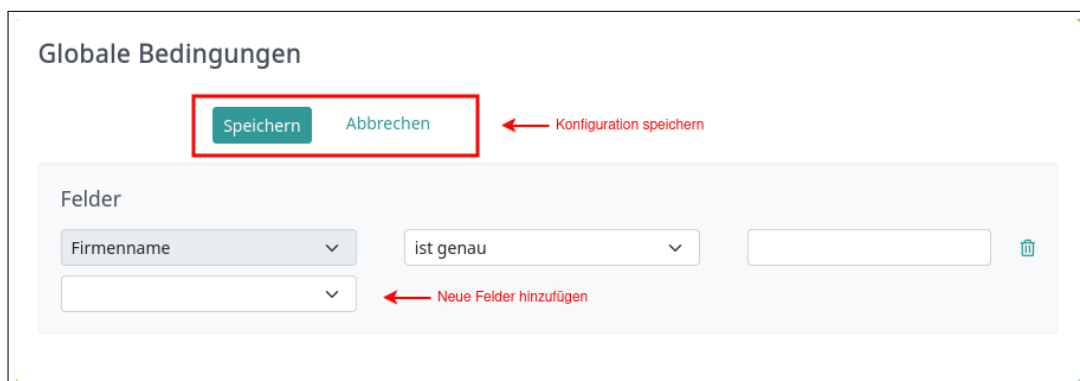


Abbildung 17.14: Globale Bedingungen bearbeiten

Bei erfolgreicher Speicherung werden die Bedingungen in der Detailansicht des Abonnements als Text dargestellt.

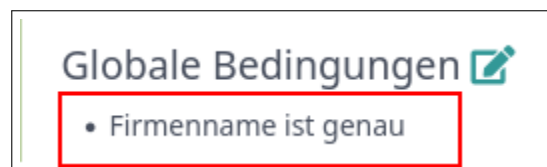


Abbildung 17.15: Globale Bedingungen gespeichert

17.7.2 Durchführung der Instruktion

Personenfilterung

Durchführungsprotokoll Instruktion Personenfilterung 1.1	
Datum	20.03.2025
Name des Instruktors	Marc Egli
Name der Testperson	Daniel Illi
Funktion der Testperson	Entwickler, Software Architekt in Hibobito
Testgerät	Lenovo ThinkPad
Betriebssystem	Pop!_OS
Aufgabe	<ul style="list-style-type: none"> • Auf der Gruppe “Kontakte” und der Ebene “Dachverband” soll ein neuer Personenfilter angelegt werden. • Der Personenfilter soll nach Personen mit den Tags “Nevermind” und “Someone” filtern. • Der Personenfilter soll nach den Rollen “Group::TopLayerOffice::Manager” und “Group::Local::Leader” filtern. • Das Stichdatum der Rollen ist vom 04.03.2025 bis zum 13.03.2025 • Die Gültigkeit der Rollen soll “Aktiv” sein. • Der Personenfilter soll unter dem Namen “Danis Filter” abgespeichert werden.
Hilfsmittel	Der Testperson wird eine Ausgedruckte Version der Instruktion bereitgestellt.
Beobachtungen und Anmerkungen	<ul style="list-style-type: none"> • Testperson findet sich durch die Angaben in der Instruktion zurecht. • Testperson benötigt einen Moment für das Finden des Dropdowns, findet dies danach.

Tabelle 17.19: Durchführungsprotokoll Instruktion Personenfilterung 1.1

Durchführungsprotokoll Instruktion Personenfilterung 1.2

Beobachtungen und Anmerkungen	<ul style="list-style-type: none">• Testperson hat Probleme mit der Auswahl der Gruppe, mehrere Gruppen haben den gleichen Namen im Dropdown.• Testperson kann den Filter korrekt abspeichern.
-------------------------------	---

Tabelle 17.20: Durchführungsprotokoll Instruktion Personenfilterung 1.2

Globale Bedingungen

Durchführungsprotokoll Instruktion globale Bedingungen	
Datum	20.03.2025
Name des Instruktors	Marc Egli
Name der Testperson	Daniel Illi
Funktion der Testperson	Entwickler, Software Architekt in Hibobito
Testgerät	Lenovo ThinkPad
Betriebssystem	Pop!_OS
Aufgabe	<ul style="list-style-type: none"> • Auf der Gruppe “Dachverband” sollen die globalen Bedingungen angepasst werden. • Die globalen Bedingungen sollen nach Abonnenten mit dem Firmennamen “Puzzle ITC” mit der Genauigkeit “ist genau” filtern. • Der globalen Bedingungen sollen nach Abonnenten mit dem Alter grösser als 56 filtern. • Die globalen Bedingungen sollen nach Abonnenten mit dem Ort “Riedbach” filtern. • Die überarbeiteten globalen Bedingungen sollen anschliessend gespeichert werden.
Hilfsmittel	Der Testperson wird eine Ausgedruckte Version der Instruktion bereitgestellt.
Beobachtungen und Anmerkungen	<ul style="list-style-type: none"> • Testperson findet zuerst das richtige Abonnement nicht. Dieses wurde nicht in der Aufgabenstellung definiert. • Testperson kann die globalen Bedingungen schnell bearbeiten. • Testperson konnte die globalen Bedingungen speichern.

Tabelle 17.21: Durchführungsprotokoll Instruktion globale Bedingungen

17.7.3 Fazit

Durch die Durchführung der Instruktion konnte verifiziert werden, dass die erstellte Anleitung dem Benutzer hilft die Grundfunktionalitäten des implementierten Features zu benutzen. Es wurde durch die Durchführung verifiziert das die Instruktion den Benutzer bei der Anwendung der Personenfilterung wie der Bearbeitung der globalen Bedingungen zum Erfolg führt.

Während der Durchführung wurden zwei Probleme identifiziert: Die Aufgabenstellung zur Bearbeitung der globalen Bedingungen eines Abonnementes war zu ungenau. Hier hätte der Name des zu bearbeitenden Abonnementes hinterlegt werden müssen. Des weiteren wurde erkannt, das die Angaben der Rollen innerhalb des Filterungsdropdowns der Personenfilter zu ungenau ist. Um dies zu beheben könnte zusätzlich der Name der darüberliegenden Ebene angezeigt werden.

18 Abschluss Sprint Umsetzung

An dieser Stelle endet der erste Sprint welcher sich mit der Umsetzung der Arbeit befasste. Der Sprint dauerte vom 11.03.2025 bis am 19.03.2025.

<div> ▼ Refinement 1 Estimate: 0 This is ready to be refined ... </div>		
1	<div> 🕒 Instruction #22 </div>	<div> Refinement ▼ </div>
<div> + Add item </div>		
<div> ▼ Done 19 Estimate: 0 This has been completed ... </div>		
2	<div> ✅ Implement Endpoints #14 </div>	<div> Done ▼ </div>
3	<div> ✅ Task / Standards #2 </div>	<div> Done ▼ </div>
4	<div> ✅ Create Turbo Streams #15 </div>	<div> Done ▼ </div>
5	<div> ✅ Implement Add-Dropdown #18 </div>	<div> Done ▼ </div>
6	<div> ✅ Global Conditions #17 </div>	<div> Done ▼ </div>
7	<div> ✅ Feature Tests #19 </div>	<div> Done ▼ </div>
8	<div> ✅ Manual Tests #20 </div>	<div> Done ▼ </div>
9	<div> ✅ AI-Dokumentation #21 </div>	<div> Done ▼ </div>

Abbildung 18.1: Backlog nach Sprint Umsetzung

Im Sprint konnten ausgenommen einer User Story alle Stories umgesetzt werden. Die User Story “Instruktion” konnte nicht vollständig umgesetzt werden. Grund dafür ist das die Instruktion mit Daniel Illi erst am Donnerstag (20.03.2025) durchgeführt wird. Gemäss dem Vorgehen wird die User Story zurück in die Spalte “Refinement” gesetzt und im kommende Sprint Planning im neuen Sprint eingeteilt.

18.1 Sprint Diagramme

18.1.1 Burn up



Abbildung 18.2: Burn up nach Sprint 2

Anhand des Burn ups sieht man, dass die meisten User Stories bis zum Sprintabschluss abgeschlossen werden konnten. Durch den heutigen Sprintabschluss wurden die Akzeptanzkriterien der einzelnen User Stories abgenommen und bei Erfüllung in die Spalte “Done” geschoben.

18.1.2 User Story Chart

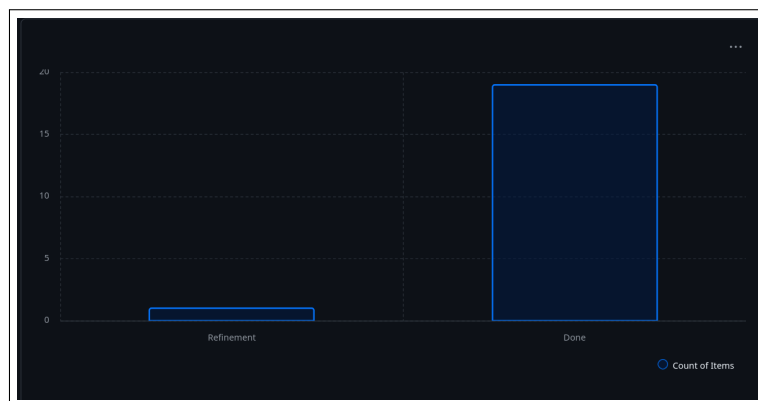


Abbildung 18.3: Charts nach Sprint 2

Als Fazit werden im obenstehenden Bild die Verteilung der Status aller User Stories nach diesem Sprint angezeigt. Insgesamt wurden im Rahmen der IPA bis zu diesem Punkt 19 User Stories abgeschlossen und eine User Story befindet sich im Refinement.

19 Einführung

Im folgenden Abschnitt wird behandelt, wie das IPA Feature in die bestehende Hitobito Applikation eingeführt werden wird und welche weitere Arbeiten vor diesem Schritt erledigt werden müssen.

19.1 Einführung des Systemes

Die Arbeiten in dieser IPA werden wie ein normales Feature behandelt. Das bedeutet, der Code wird sich erst mindestens einem Code Review unterziehen müssen und wenn nötig verbessert werden. Der Betrieb muss letztlich nicht unbedingt angepasst werden und ist in der Dokumentation auf GitHub beschrieben. Jedoch müssen vor dem Merge in den produktiven Branch des Cores noch einige Arbeiten erledigt werden, welche innerhalb dieser IPA nicht beachtet wurden.

19.2 Weitere Arbeiten

Beim Filterkriterium “Tags” ist es momentan möglich sowohl bei dem Eingabefeld zum Ausschliessen wie auch beim Eingabefeld zum Einschliessen der Tags das gleiche Tag einzugeben. Dies ist ein logischer Fehler, der jedoch noch auf die alte Grundfunktionalität zurückzuführen ist. Bevor das IPA Feature gemerged wird sollte für die Behebung dieses Problems etwas Zeit investiert werden.

Während der Umsetzung kam von Daniel Illi der Vorschlag, man könnte für die einzelnen Filterkriterien eine Validierung für deren Werte einfügen. In diesem Fall, müsste es, wie zu Beginn im Mockup angedacht, möglich sein ein einzelnes Filterkriterium speichern zu können. Bei Fehlschlägen der Validierung würde dann ein entsprechender Fehler auf der Benutzerschnittstelle angezeigt werden. Diese Anpassung könnte vor einem Review oder als separates Feature zur Filterung implementiert werden.

Ein wichtiger Aspekt ist zuletzt das Durchtesten mit allen Wagons von Hitobito. Bei vielen Anpassungen mit Core, sollte die Funktionalität in den

Wagons mit manuellen sowie automatisierten Tests sichergestellt werden. Am Besten wird dies vor dem Review durch einen anderen Entwickler durchgeführt, um die Aufkommenden Fehler direkt selber beheben zu können und so Zeit im Review zu sparen.

20 Abschluss Sprint Finalisierung

An dieser Stelle endet der erste Sprint welcher sich mit der Finalisierung der Arbeit befasste. Der Sprint dauerte vom 19.03.2025 bis am 21.03.2025.

Title	Status
In-Progress 1 <small>Estimate: 0</small> This is actively being worked on	
1 Delivery and last changes #28	In-Progress
+ Add item	
Done 25 <small>Estimate: 0</small> This has been completed	
2 Glossar and Word Definition #25	Done
3 Code annex #26	Done
4 Reread and correct #24	Done
5 Short Report IPA #23	Done
6 Timeplan #27	Done
7 Instruction #22	Done

Abbildung 20.1: Backlog nach Sprint Finalisierung

In diesem Sprint konnten alle User Stories umgesetzt werden. Die letzte User Story “Delivery and last changes” wird bis zum Ende und der Abgabe der Arbeit in der Spalte “In-Progress” bleiben. Wurde die Arbeit abgegeben und signiert, kann diese User Story abgeschlossen werden.

20.1 Sprint Diagramme

20.1.1 Burn up

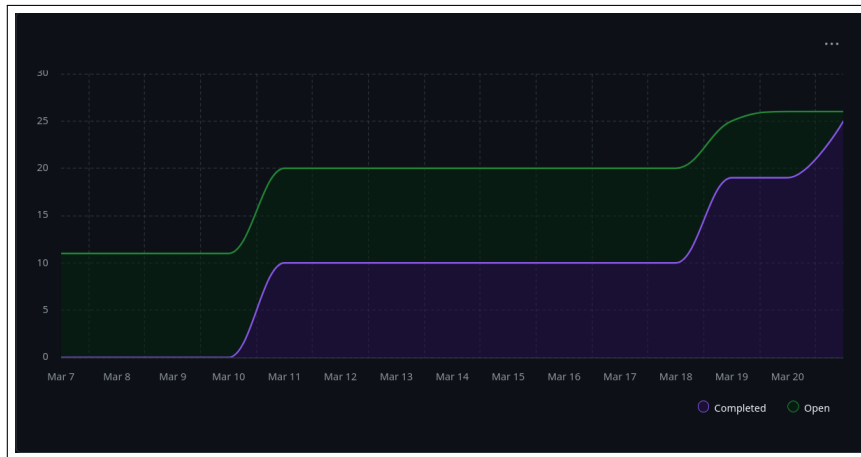


Abbildung 20.2: Burn up nach Sprint 3

Anhand des Burn ups sieht man, dass die meisten User Stories bis zum Sprintabschluss abgeschlossen werden konnten. Mit dem heutigen Tag, wurden alle User Stories (ausgenommen der erwähnten) abgeschlossen.

20.1.2 User Story Chart

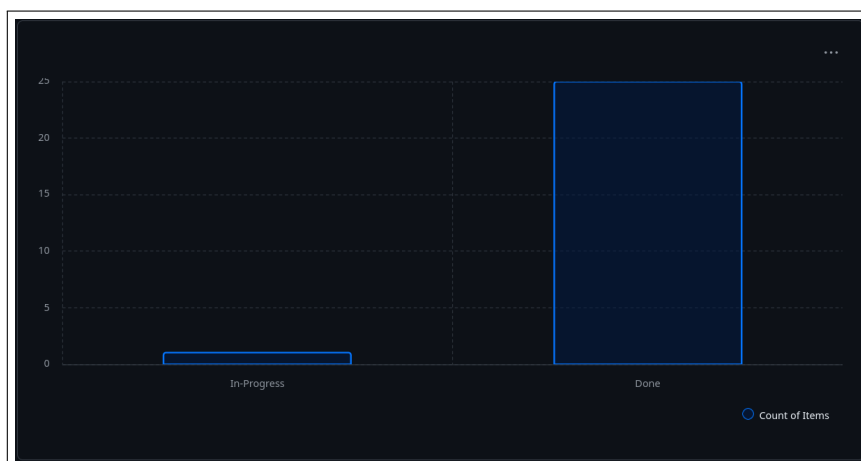


Abbildung 20.3: Charts nach Sprint 3

Insgesamt konnten in dieser IPA 25 User Stories abgeschlossen werden.

Teil III

Anhänge und Verzeichnisse

Hitobito: Neue Generation von Personen-Filtern
Autor: Marc Egli

21 Verzeichnisse

21.1 Tabellenverzeichnis

1	IPA Daten	1
4.1	Sicherung Dokumentation	16
4.2	Sicherung Code	17
4.3	Änderungsprotokoll	18
6.1	Rollenbeschreibung	25
6.2	Rollenbeschreibung	26
6.3	Rollenbeschreibung IPA	27
7.1	Risikoanalyse Projektrisiken	28
7.2	Risikomatrix Projektrisiken	29
8.1	Sprintziele	30
9.1	Tätigkeiten Tag 1	32
9.2	Tätigkeiten Tag 2	35
9.3	Tätigkeiten Tag 3	38
9.4	Tätigkeiten Tag 4	41
9.5	Tätigkeiten Tag 5	45
9.6	Tätigkeiten Tag 6	49
9.7	Tätigkeiten Tag 7	52
9.8	Tätigkeiten Tag 8	55
9.9	Tätigkeiten Tag 9	59
9.10	Tätigkeiten Tag 10	62
9.11	Tätigkeiten Tag 11	64
9.12	Tätigkeiten Tag 12	66
13.1	Felder-Attribute	79
13.2	Methodenwahl	95
13.3	Ablaufprotokoll Teil 1	97
13.4	Ablaufprotokoll Teil 2	100
13.5	Ablaufprotokoll Teil 3	101
13.6	Ablaufprotokoll Teil 4	102
13.7	Ablaufprotokoll Teil 5	103
13.8	Dringlichkeitsstufen	103
13.9	Bedürfnisse der befragten Person	104
14.1	Schnittstellen	105
14.2	Berechtigungen	106

14.3	Risikoanalyse Sicherheitsrisiken	111
14.4	Risikomatrix Sicherheitsrisiken	112
14.5	Nicht funktionale Anforderung 1	113
14.6	Nicht funktionale Anforderung 2	113
14.7	Nicht funktionale Anforderung 3	113
14.8	Nicht funktionale Anforderung 4	114
14.9	Nicht funktionale Anforderung 5	114
14.10	Nicht funktionale Anforderung 6	114
14.11	Funktionale Anforderung 1	114
14.12	Funktionale Anforderung 2	115
14.13	Funktionale Anforderung 3	115
14.14	Funktionale Anforderung 4	115
14.15	Funktionale Anforderung 5	115
14.16	Zeitraahmen	116
14.17	Hotwire Kenntnisse	116
14.18	Selbstständigkeit	116
15.1	Anwendungsfall: Filter speichern	118
15.2	Anwendungsfall: Filterkriterien anzeigen	118
15.3	Anwendungsfall: Filterkriterien bearbeiten	119
15.4	Anwendungsfall: Personen filtern	119
15.5	Variantenentscheid Kriterien	125
15.6	Bewertung Lösungsvariante 1	126
15.7	Bewertung Lösungsvariante 2	127
15.8	Nutzwertanalyse	128
15.9	Fehlerklassen	133
15.10	Accounts für manuelle Tests	134
15.11	Testfall 1	135
15.12	Testfall 2	136
15.13	Testfall 3	137
15.14	Testfall 3	138
15.15	Testfall 4	139
15.16	Testfall 4	140
15.17	Testfall 5	141
15.18	Testfall 5	142
15.19	Testfall 6	143
15.20	Testfall 7	144
15.21	Testdaten	145
15.22	Testfall 8	146
15.23	Testfall 9	147
15.24	Testfall 10	148
15.25	Testfall 11	149
15.26	Testfall 12	150
15.27	Testfall 13	151
15.28	Testfall 14	152

15.29	Testfall 15	153
15.30	Testfall 16	154
15.31	Testfall 17	155
17.1	Turbo Request	159
17.2	Resultat Testfall 1	171
17.3	Resultat Testfall 2	171
17.4	Resultat Testfall 3	172
17.5	Resultat Testfall 4	172
17.6	Resultat Testfall 5	173
17.7	Resultat Testfall 6	173
17.8	Resultat Testfall 7	173
17.9	Resultat Testfall 8	174
17.10	Resultat Testfall 9	174
17.11	Resultat Testfall 10	174
17.12	Resultat Testfall 11	175
17.13	Resultat Testfall 12	175
17.14	Resultat Testfall 13	175
17.15	Resultat Testfall 14	176
17.16	Resultat Testfall 15	176
17.17	Resultat Testfall 16	176
17.18	Resultat Testfall 17	177
17.19	Durchführungsprotokoll Instruktion Personenfilterung 1.1	184
17.20	Durchführungsprotokoll Instruktion Personenfilterung 1.2	185
17.21	Durchführungsprotokoll Instruktion globale Bedingungen	186
22.1	Verwendete Abkürzungen	203
23.1	Glossar	204
24.1	Protokoll Sitzung 1.1	206
24.2	Protokoll Sitzung 1.2	207
24.3	Protokoll Sitzung 2.1	208
24.4	Protokoll Sitzung 2.2	209

21.2 Abbildungsverzeichnis

5.1	Github Projects Board	20
5.2	Beispiel einer User Story	23
6.1	Rollen in Scrum, selbstgezeichnet mit Draw.io	25
6.2	Rollenverteilung in der IPA, selbstgezeichnet mit Draw.io	27
8.1	Zeitplan IPA	31

12.1	Puzzle House	72
13.1	Gruppenübersicht Hitobito	74
13.2	Personenübersicht Hitobito	75
13.3	Filterauswahl Hitobito	75
13.4	Filterübersicht Hitobito	76
13.5	Filterkriterium Rollen	77
13.6	Filterkriterium Rollen - Zeitraum und Archivierung	77
13.7	Filterkriterium Qualifikationen	78
13.8	Filterkriterium Felder	78
13.9	Filterkriterium Sprache	80
13.10	Filterkriterium Tags	80
13.11	Filterkriterium Tags	81
13.12	Gruppenübersicht	82
13.13	Abonnenten Übersicht	82
13.14	Übersicht einzelnes Abonnementes	83
13.15	Abonnentenübersicht	83
13.16	Globale Filterungsbedingungen für Abonnenten	84
13.17	Mockup Filterübersicht	85
13.18	Mockup mit Tags und Felder definiert	86
13.19	Mockup mit Qualifikationen und Rollen definiert	88
13.20	Bearbeitungsmaske Filterkriterium Felder	89
13.21	Bearbeitungsmaske Filterkriterium Gruppe	90
13.22	Bearbeitungsmaske Filterkriterium Qualifikationen	91
13.23	Bearbeitungsmaske Filterkriterium Tags	91
14.1	Gruppen und Ebenen, selbstgezeichnet mit Draw.io	107
14.2	Beispiel Berechtigungen von Heinz, selbstgezeichnet mit Draw.io	108
14.3	Beispiel Berechtigungen von Tim, selbstgezeichnet mit Draw.io	109
14.4	Beispiel Berechtigungen von Tim, selbstgezeichnet mit Draw.io	110
15.1	Anwendungsdiagramm	117
15.2	Services	120
15.3	Lösungsvariante 1: Turbo-Konzept	121
15.4	Lösungsvariante 1: Klassendiagramm	122
15.5	Lösungsvariante 2: Turbo-Konzept	123
15.6	Lösungsvariante 2: Klassendiagramm	124
16.1	Backlog nach Sprint Initialisierung	156
16.2	Burn up nach Sprint 1	157
16.3	Charts nach Sprint 1	157
17.1	FilterDropdownController	160
17.2	Filterkriterium "Tag" mit Mülleimer-Symbol	163
17.3	Globale Bedinungen	165
17.4	Gruppenliste	178

17.5	Personenreiter	179
17.6	Neuer Filter erstellen	179
17.7	Filterkriterium hinzufügen	179
17.8	Filterkriterium bearbeiten oder entfernen	180
17.9	Personenfilter speichern	180
17.10	Wiederverwendung eines gespeicherten Filters	181
17.11	Gruppenliste	182
17.12	Abonnementenübersicht der Gruppe	182
17.13	Detailansicht eines Abonnements	183
17.14	Globale Bedingungen bearbeiten	183
17.15	Globale Bedingungen gespeichert	183
18.1	Backlog nach Sprint Umsetzung	188
18.2	Burn up nach Sprint 2	189
18.3	Charts nach Sprint 2	189
20.1	Backlog nach Sprint Finalisierung	192
20.2	Burn up nach Sprint 3	193
20.3	Charts nach Sprint 3	193
24.1	Git commit Message Konvention	205

Quellenverzeichnis

- [Verbindung zwischen repositories verstehen] <https://docs.github.com/en/repositories/viewing-activity-and-data-for-your-repository/understanding-connections-between-repositories>, (04.03.2025)
- [Issue Templates konfigurieren] <https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/configuring-issue-templates-for-your-repository>, (04.03.2025)
- [Translating] <https://dict.leo.org/german-english>, (04.03.2025)
- [Icon made by Freeplk from <http://www.flaticon.com/>] https://www.flaticon.com/free-icon/user_1077114?term=person&page=1&position=1&origin=search&related_id=1077114, (04.03.2025)
- [Product Owner Definition] <https://agilescrumgroup.de/product-owner-aufgaben/>, (04.03.2025)
- [Scrum Master Definition] <https://agilescrumgroup.de/scrum-master-aufgaben/>, (04.03.2025)
- [Entwickler Definition] <https://scrumguide.de/entwickler/>, (04.03.2025)
- [Bedürfniserhebung - Aufbau und Ablauf] https://easy--feedback-de.translate.goog/umfrage-beispiele/bedarfsanalyse-fragebogen-vorlage/bedarfsanalyse-aufbau-ablauf-schritte/?_x_tr_sl=de&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=sc, (06.03.2025)
- [Bedürfniserhebung - Interviews] <https://kreativ.mfg.de/digitale-kultur/kompass-digitale-kultur/prozess/nutzerinnen-gruppe/bedarfsanalyse-interviews/>, (06.03.2025)
- [Ist-Situation Hitobito Bilder] <https://demo.hitobito.com>, (06.03.2025)
- [Hochkommas in Latex] <https://thinkscience.co.jp/en/downloads/ThinkSCIENCE-LaTeX-habits-to-avoid.pdf>, (06.03.2025)
- [Git Commit Messages bearbeiten] <https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/changing-a-commit-message>, (06.03.2025)

- [Codeinput in Latex] https://www.overleaf.com/learn/latex/Code_listing, (07.03.2025)
- [XSS Attacks in Rails] <https://www.stackhawk.com/blog/rails-xss-examples-and-prevention/>, (07.03.2025)
- [Spezialzeichen in Latex] https://en.wikibooks.org/wiki/LaTeX/Special_Characters, (07.03.2025)
- [Action in view auslesen] <https://stackoverflow.com/questions/8053312/rails-how-to-determine-controller-action-in-view>, (11.03.2025)
- [User Story eines anderen Repositories referenzieren] <https://stackoverflow.com/questions/60268714/github-how-to-reference-an-issue-in-a-commit-from-a-different-repository>, (11.03.2025)
- [Before Action in Controller] <https://stackoverflow.com/questions/45615591/how-does-only-at-before-action-work-in-rails>, (11.03.2025)
- [Mapping in Rails] <https://womanonrails.com/one-line-map-ruby>, (11.03.2025)
- [HTML to HAML Converter] <https://awsm-tools.com/html-to-haml>, (11.03.2025)
- [For-Loops in Rails Views] <https://stackoverflow.com/questions/13166847/rails-for-loop-view>, (11.03.2025)
- [API Endpoints in Rails] <https://dev.to/vladhilko/introduction-to-rails-api-how-to-create-your-first-endpoint-in-less-than-a-minute>, (12.03.2025)
- [Variable in views] <https://stackoverflow.com/questions/15171850/how-to-create-and-use-variable-in-views-template-in-rails>, (13.03.2025)
- [Rendering of Turbostreams] <https://discuss.hotwired.dev/t/turbostream-cant-append-new-records-to-my-index-view/3569>, (13.03.2025)
- [Delete Request für Turbostreams] <https://stackoverflow.com/questions/75656122/how-to-have-a-delete-link-respond-to-turbo-stream-and-html-in-rails-7>, (13.03.2025)
- [Template für JavaScript in Latex] <https://tex.stackexchange.com/questions/89574/language-option-supported-in-listings>, (13.03.2025)

[Tom-Select] <https://tom-select.js.org/examples/options/>, (14.03.2025)

[Übersetzung von Texten für die View] <https://dict.leo.org/german-english/datum>, (14.03.2025)

[Turbostreams vs Turboframes] <https://youtu.be/vnDWSGtz0Cc>,
(19.03.2025)

22 Verwendete Abkürzungen

Abkürzung	Bedeutung
CSV	Comma seperated values
DoD	Definition of Done
HAML	HTML Abstraction Markup Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IPA	Individuelle Praktische Arbeit
KI	Künstliche Intelligenz
ORM	Object-Relational Mapping
UI	User Interface
UML	Unified Modeling Language
USB	Universal Serial Bus
UX	User Experience
VF	Verantwortliche Fachkraft
XSS	Cross-site scripting

Tabelle 22.1: Verwendete Abkürzungen

23 Glossar

Bezeichnung	Bedeutung
Active Record	ORM in Rails
Backend	Serverseitiger Teil einer Applikation, welcher für den Benutzer nicht sichtbar ist
Core	Wagon welcher die Grundfunktionalität von Hitobito beinhaltet
Div	Container für Element in der Benutzerschnittstelle
Formatter	Software welche die Dateien automatisch formatiert
Frontend	Teil der Applikation mit welcher der Benutzer direkt interagiert
Gem	Library als Rubycode welcher bestimmte Funktionalität zur Verfügung stellt
Hitobito	Community Management Tool
Hotwire	Frontend Framework für Ruby on Rails
Mockup	Entwurf einer Benutzerschnittstelle, dient der Orientierung während der Umsetzung
Partials	Benutzerschnittstelle welche wiederverwendet wird
Product Backlog	Ort an welchem die User Stories verwaltet werden
Scrum	Agile Projektvorgehensmethode
SSH-Key	Zugang zum SSH-Protokoll, vergleichbar zu einem Benutzernamen mit Passwort
Stakeholder	Synonym für "Kunde" im Scrum Prozess
Transifex	Software in welcher Übersetzter Wörter übersetzten und diese dann als .yml Datei zur Verfügung gestellt werden
Turboframe	Container innerhalb einer Benutzerschnittstelle welcher sich auswechseln lässt
Turbostreams	Technologie um einen Teil der Benutzerschnittstelle zu aktualisieren
Wagon	Zusätzliche Funktionalität für jeweiligen Kunden. Kann an Core angehängt werden

Tabelle 23.1: Glossar

24 Anhänge

24.1 Git commit Message Konvention

Konvention Commit Message

Falls keine besonderen Vorgaben durch den Kunden vorhanden, empfehlen wir – angelehnt an den Artikel [How to Write a Git Commit Message](#) – folgende Konvention zu verwenden:

- Sprache: Englisch
- Kurze und prägnante Message, idealerweise unter 50 Zeichen ([Details](#))
- Mit Grossbuchstaben beginnen ([Details](#))
- Kein Punkt am Schluss ([Details](#))
- Den *imperative mood* (Befehlsform) verwenden, also «Fix bug with X» statt «Fixed bug with X» oder «More fixes for broken stuff» ([Details](#))
- Wenn vorhanden das Ticket referenzieren:
 - Bei Open Project Work Packages: «Add X, refs #12345»
 - Bei Gitlab/Github Issues: «Add X #12345»

Dies entspricht grundsätzlich auch dem Stil wie ihn viele Open Source Projekte wie z.B. der [Linux Kernel](#), [Spring Boot](#), [Rails](#) oder auch [Git](#) selber anwenden.

Für grössere Projekte, bei welchen auch das Changelog automatisiert generiert wird, kann die [Conventional Commits](#) Spezifikation sinnvoll sein.

Abbildung 24.1: Git commit Message Konvention

24.2 Sitzungsprotokolle

24.2.1 Sitzung 1

Datum	13.03.2025
Anwesende Personen	<ul style="list-style-type: none">• Lorenz Müller• Robin Steiner• Daniel Illi• Marc Egli
Besprechungspunkte	
Vorstellung	Kurze Vorstellungsrunde
Rollenverteilung	Die Rollenverteilung während der IPA wurde nochmals kurz erläutert.
Arbeitsplatz und Material	Marc Egli bestätigte, dass alle nötigen Materialien vorhanden sind, er an keinem anderen Projekt arbeitet und er sich an seinem Arbeitsplatz konzentrieren kann.
Detailbeschrieb besprechen	Der Detailbeschrieb wurde angeschaut, Unklarheiten wurden geklärt.
Bewertungskriterien Hinweise	Lorenz Müller erwähnte, dass der IPA-Kurzbericht nun im Teil 2 der IPA-Dokumentation hinterlegt werden muss.
Bewertungskriterium KI-Einsatz	Lorenz Müller wies darauf hin, dass das Kriterium C1: Einsatz von KI-Modellen falsch verstanden wurde. Es wurde festgelegt das das Kriterium nach den festgelegten Punkten der verantwortlichen Fachkräfte bewertet wird.
Verwendung künstliche Intelligenz	Lorenz Müller erklärte das die Verwendung von KI erlaubt ist, diese jedoch gekennzeichnet werden muss.
Individuelle Bewertungskriterien	Die Individuellen Bewertungskriterien wurden nochmals angeschaut und besprochen.

Tabelle 24.1: Protokoll Sitzung 1.1

Besprechungs Punkte	
Zeitplan besprechen	Lorenz Müller wies daraufhin, dass der Aufbau der IPA mit drei Sprints nach Scrum nicht sinnvoll wäre. Um weiteren Zeitaufwand zu verhindern, wurde festgelegt, dass die Projektmethode und der Zeitplan wie vorbereitet weitergeführt werden, ohne dies mit einer Abwertung zu ahnden. Des Weiteren wurde von Lorenz Müller angemerkt, dass stets Tätigkeiten im Zeitplan angegeben werden müssen. Wochenenden könnten ausserdem aus dem Zeitplan ausgeklammert werden. Die Planung soll ausserdem so umgeschrieben werden, dass 80 Stunden Aufwand für die IPA geschätzt wird. Tickets welche 8 Stunden einnehmen, sollen in weitere kleinere Tickets aufgeteilt werden.
Arbeitsjournal besprechen	Lorenz Müller erklärte, dass wichtige Erkenntnisse in der Reflexion vorkommen müssen, Anmerkungen zu der Zeit seien zu vernachlässigen, da diese in den geschätzten Stunden ersichtlich sind.
Risikoanalyse besprechen	Die Risikoanalyse wurde angeschaut. Lorenz Müller wies daraufhin, dass eine weiter Risikoanalyse für Projektrisiken angefertigt werden müsse. Diese Risikoanalyse soll im ersten Teil der IPA hinterlegt sein.
Festlegung des zweiten Besuchstages	Ein zweiter Besuchstag wurde am 12.03.2025 via Google Meet festgelegt, der Zugang für das Remote-Meeting wurde erteilt.
Hinweise zu Tester und Rechtschreibprüfer	Lorenz Müller wies daraufhin, dass eine Person die IPA-Dokumentation gegenlesen und auf Rechtschreibfehler überprüfen darf. Diese Person muss nicht erwähnt werden. Des Weiteren darf ein Tester ausgewählt werden, welcher das Produkt prüft. Dieser muss angegeben werden, falls verwendet.

Tabelle 24.2: Protokoll Sitzung 1.2

24.2.2 Sitzung 2

Datum	13.03.2025
Anwesende Personen	<ul style="list-style-type: none"> • Lorenz Müller • Robin Steiner • Daniel Illi • Marc Egli
Besprechungs Punkte	
Stand der IPA	Marc Egli präsentiert den Stand der IPA
Zeitplan	Marc Egli präsentiert den Zeitplan
Erläuterung Zeitverlust	Marc Egli erklärt, worauf der Zeitverlust im ersten Sprint zurückzuführen ist.
Stand der Dokumentation überprüfen	<p>Marc Egli präsentiert die Dokumentation und stellt Fragen dazu, Lorenz Müller gibt Feedback und beantwortet Marc Egli's Fragen:</p> <ul style="list-style-type: none"> • Frage von Marc Egli: Muss das Datenschutzkonzept hinterlegt werden? • Antwort von Lorenz Müller: Nur falls es für diese Arbeit relevant ist. Hier gilt es Schwerpunkte zu setzen. • Anmerkung von Lorenz Müller zu den Bildern: Bilder müssen auch bei Vergrösserung des Dokumentes noch zu lesen sein, darauf achten. • Frage von Marc Egli: Es wurde bereits ein Mockup für die zu implementierende Benutzerschnittstelle entworfen. Wie können die Anforderungen in diesem Fall lösungsneutral formuliert werden? • Antwort von Lorenz Müller: Wenn das Mockup als Vorarbeit deklariert wurde, kann darauf Bezug genommen werden. Bei der Lösungsneutralität geht es darum das entworfene Konzept zu hinterfragen und nicht einfach zu akzeptieren.

Tabelle 24.3: Protokoll Sitzung 2.1

<p>Stand der Dokumentation überprüfen</p>	<ul style="list-style-type: none"> Anmerkung von Lorenz Müller: Zu jeder Überschrift muss ein entsprechender Einleitungstext vorhanden sein, darauf achten. <p>Lorenz Müller erklärt den Ablauf der Präsentation und des Fachgesprächs</p> <ul style="list-style-type: none"> Die Präsentation muss zwischen 15-20 Minuten dauern. Die Demo sollte zwischen 5-10 Minuten dauern. Keine Zeit verlieren, da die Zeit auf Kosten des Fachgesprächs verloren geht. Im Fachgespräch werden Fragen zu sechs Fragenkomplexen kommen. Pro Fragenkomplex werden vier Unterfragen gestellt.
<p>Präsentation und Fachgespräch</p>	
<p>Terminvereinbarung</p>	<p>Der Termin für Fachgespräch und Präsentation wird besprochen. Lorenz Müller merkt an, dass er eine Umfrage für den Termin aufschalten werde.</p>
<p>Abschluss</p>	<p>Teilnehmer verabschieden sich.</p>

Tabelle 24.4: Protokoll Sitzung 2.2

24.3 Code

24.3.1 people_filter_ability.rb

```

1 class PeopleFilterAbility < AbilityDsl::Base
2   on(::PeopleFilter) do
3     permission(:contact_data).may(:new).all
4     permission(:group_read).may(:new).in_same_group
5 +   permission(:group_read).may(:filter_criterion).in_same_group
6     permission(:group_and_below_read).may(:new).in_same_group_or_below
7 +
8     permission(:group_and_below_read).may(:filter_criterion).in_same_group_or_below
9     permission(:layer_read).may(:new).in_same_layer
10 +   permission(:layer_read).may(:filter_criterion).in_same_layer
11     permission(:layer_full).may(:create, :destroy, :edit, :update).in_same_layer
12     permission(:layer_and_below_read).may(:new).in_same_layer_or_below
13 +
14     permission(:layer_and_below_read).may(:filter_criterion).in_same_layer_or_below
15     permission(:layer_and_below_full).may(:create, :destroy, :edit, :update).in_same_layer_or_below
16   end
17 end

```

24.3.2 people_readables.rb

```
1 class PersonReadables < GroupBasedReadables
2   attr_reader :group
3
4   - def initialize(user, group = nil)
5   + def initialize(user, group = nil, roles_join = nil)
6     super(user)
7
8   +   @roles_join = roles_join || {roles: :group}
9     @group = group
10
11     if @group.nil?
12     end
13
14     def accessible_people
15       if user.root?
16         Person.only_public_data.then do |scope|
17           - group ? scope.joins(roles: :group).distinct : scope
18           + group ? scope.joins(@roles_join).distinct : scope
19         end
20       else
21         scope = Person.only_public_data.where(accessible_conditions.to_a).distinct
22         if has_group_based_conditions?
23           # Only add these joins when really necessary, because they are extremely
24           # expensive to
25           # compute when there are a lot of people and roles
26           - scope = scope.joins(roles: :group).where(groups: {deleted_at: nil})
27           + scope = scope.joins(@roles_join).where(groups: {deleted_at: nil})
28         end
29       end
30     end
31   end
32 end
```

24.3.3 people_filters_controller.rb

```
1 class PeopleFiltersController < CrudController
2   skip_authorize_resource only: [:create]
3
4   + before_action :set_filter_criteria, except: [:destroy]
5   +
6   + def filter_criterion
7   +   compose_role_lists
8   +   possible_tags
9   +   @filter_criterion = params[:filter_criterion]
10  +   if @filter_criteria.include?(@filter_criterion.to_sym)
11  +     respond_to do |format|
12  +       if request.method == "GET"
13  +         format.turbo_stream { render "create", status: :ok }
14  +       end
15  +       if request.method == "POST"
16  +         format.turbo_stream { render "delete" }
17  +       end
18  +     end
19  +   end
20  + end
21  +
22  + def set_filter_criteria
23  +   @filter_criteria = [:tag, :role, :qualification, :attributes]
24  + end
25
26  def compose_role_lists
27    @role_types = Role::TypeList.new(group.class)
28    @qualification_kinds = QualificationKind.list.without_deleted
29    + .map { |qualification|
30    +   [qualification.label, qualification.id, qualification.id]
31    + }
32    + @roles = Role.all.map { |role| [role.type, role.id, role.id] }
33    + @kinds = Person::Filter::Role::KINDS.each_with_index
34    + .map { |kind, index|
35    +   [t("people_filters.form.filters_role_kind.#{kind}"),
36    +   t("people_filters.form.filters_role_kind.#{kind}"),
37    +   index + 1]
38    + }
39    + @validities = [
40    +   [t("people_filters.qualification.validity_label.active"), "active", 1],
41    +   [t("people_filters.qualification.validity_label.reactivateable"),
42    +   "reactivateable", 2],
43    +   [t("people_filters.qualification.validity_label.not_active_but_reactivateable"),
44    +   "not_active_but_reactivateable", 3],
45    +   [t("people_filters.qualification.validity_label.not_active"),
46    +   "not_active", 4],
47    +   [t("people_filters.qualification.validity_label.all"), "all", 5],
48    +   [t("people_filters.qualification.validity_label.none"), "none", 6],
49    +   [t("people_filters.qualification.validity_label.only_expired"),
50    +   "only_expired", 7]
51    + ]
52  end
53 end
```

24.3.4 people_filter_helper.rb

```
1 module PeopleFilterHelper
2   content << content_tag(:div, class: "flex-none") do
3     - select_tag("filters[attributes][#{time}][key]",
4     - options_from_collection_for_select(people_filter_attributes_for_select,
5     :last, :first, key),
6     + select(:filters, "attributes[#{time}][key]",
7     + people_filter_attributes_for_select,
8     + {selected: key},
9     + html_options.merge(disabled: true,
10     class: 'attribute_key_dropdown form-select
11     form-select-sm'))
12   end
13   content << content_tag(:div, class: "flex-none") do
14     - select_tag("filters[attributes][#{time}][constraint]",
15     - options_from_collection_for_select(filters, :last, :first, constraint),
16     + select(:filters, "attributes[#{time}][constraint]",
17     + filters,
18     + {selected: constraint},
19     + html_options.merge(class:
20     - 'attribute_constraint_dropdown
21     - ms-3 form-select form-select-sm'))
22     + 'attribute_constraint_dropdown
23     + ms-3 form-select form-select-sm', name:
24     "filters[attributes][#{time}][constraint]"))
25   end
26 end
```

24.3.5 filter_dropdown_controller.js

```
1  +// Copyright (c) 2023, Schweizer Alpen-Club. This file is part of
2  +// hitobito_sac_cas and licensed under the Affero General Public License version 3
3  +// or later. See the COPYING file at the top-level directory or at
4  +// https://github.com/hitobito/hitobito
5  +
6  +import { Controller } from "@hotwired/stimulus";
7  +
8  +export default class extends Controller {
9  +  static targets = ["menu"];
10 +
11 +  connect() {
12 +    this.observer = new MutationObserver(this.checkIfEmpty.bind(this));
13 +    if (this.hasMenuTarget) {
14 +      this.observer.observe(this.menuTarget, { childList: true });
15 +    }
16 +  }
17 +
18 +  disconnect() {
19 +    if (this.observer) {
20 +      this.observer.disconnect();
21 +    }
22 +  }
23 +
24 +  checkIfEmpty() {
25 +    if (this.menuTarget.children.length === 0) {
26 +      this.element.style.visibility = "hidden";
27 +    } else {
28 +      this.element.style.visibility = "visible";
29 +    }
30 +  }
31 +}
```

24.3.6 form_field_toggle_controller.js

```
1 export default class extends Controller {
2   static targets = ["toggle"];
3
4   toggle(event) {
5     -   if(event.target.tagName === "SELECT") {
6     -     const selected =
7       event.target.options[event.target.options.selectedIndex];
8 +     const selected =
9       event.target.options[event.target.options.selectedIndex];
10
11     -   if (selected.dataset.visibility === "true") {
12     -     this.toggleTarget.classList.remove("hidden");
13     -   } else {
14     -     this.toggleTarget.classList.add("hidden");
15     -   }
16     -   } else {
17 +     if (selected.dataset.visibility === "true") {
18       this.toggleTarget.classList.remove("hidden");
19 +     } else {
20 +       this.toggleTarget.classList.add("hidden");
21     }
22   }
23
24   untoggle() {
25     -   this.toggleTarget.classList.add("hidden");
26     -   }
27   }
28 }
```

24.3.7 form_select_controller.js

```
1  +import {Controller} from "@hotwired/stimulus";
2  +
3  +
4  +export default class extends Controller {
5  +
6  +  connect() {
7  +    + this.elementAdded(this.element)
8  +  }
9  +
10 +  elementAdded(element) {
11 +    + var app;
12 +
13 +    app = window.App || (window.App = {});
14 +    app.tomSelects = {};
15 +
16 +    app.tomSelects[element.id] = new TomSelect(`#${element.id}`, {
17 +      plugins: element.multiple ? ["remove_button"] : [],
18 +      create: false,
19 +      onItemAdd() {
20 +        + this.setTextboxValue("");
21 +        + if(!this.dropdown.classList.contains("single") ) {
22 +          + this.refreshOptions();
23 +        }
24 +      },
25 +      render: {
26 +        no_results() {
27 +          // Render localized "no results" message
28 +          const message = this.input.dataset.chosenNoResults || "No results
29 +          found";
30 +          + return `<div class="no-results">${message}</div>`;
31 +        },
32 +      },
33 +    });
34 +  }
```

24.3.8 _filter.scss

```
1  +.w-10 {
2  +  width: 10%;
3  +}
4  +
5  +.w-90 {
6  +  width: 90%;
7  +}
```


24.3.9 _filter.html.haml

```
1  +- caption ||= t("people_filters.#{type}.title")
2  +%div.well
3  +   %div.d-flex.flex-row
4  +     %h3.w-50.mb-3= caption
5  +     - if delete
6  +       %div.w-50.d-flex.justify-content-end
7  +       = link_to "/groups/#{@group.id}/people_filters/turbo/#{type.to_s}",
8  +       class: "button btn", data: {turbo_method: :delete} do
9  +         %i.fa.fa-trash
10 +     = yield
11 -- id = type.to_s.pluralize
12 -- caption ||= t("people_filters.#{type}.title")
13 -- open_class = "show" if entry.filter_chain[type.to_sym].present?
14 -
15 -.accordion-item
16 -   .accordion-header
17 -     %a.accordion-button.p-2.bg-white.text-black{ href: "##{id}", data: {
18 -       bs_toggle: :collapse } }
19 -     = caption
20 -   .accordion-collapse.collapse{ id: id, class: open_class }
21 -     .accordion-body
22 -     = yield
```

24.3.10 people_filters/_form.html.haml

```
1 += standard_form(path_args(entry), noindent: true, stacked: true) do |form|
2 +
3 + = render 'search_or_save_buttons', f: form
4 +
5 + = render_extensions :form, locals: { f: form }
6 + = form.error_messages
7 +
8 + = render 'range', f: form
9 +
10 + .mt-3.mb-3.ml-3#overview
11 +   - if action_name == 'edit'
12 +     %div{id: "role-configuration"}
13 +       - if entry.filter_chain[:role]
14 +         = render(layout: 'layouts/filter', locals: { type: :role }) do
15 +           = render 'role', f: form
16 +
17 +       - if @qualification_kinds.present?
18 +         - if entry.filter_chain[:qualification]
19 +           %div{id: "qualification-configuration"}
20 +             = render(layout: 'layouts/filter', locals: { type: :qualification })
21 +         do
22 +           = render 'qualification', f: form
23 +
24 +       - if entry.filter_chain[:attributes]
25 +         %div{id: "attributes-configuration"}
26 +           = render(layout: 'layouts/filter', locals: { type: :attributes }) do
27 +             = render 'attributes', f: form
28 +       - if entry.filter_chain[:tag]
29 +         %div{id: "tag-configuration"}
30 +           = render(layout: 'layouts/filter', locals: { type: :tag }) do
31 +             .label-columns
32 +             = render 'tag'
33 +
34 +       - if action_name == 'new'
35 +         %h3#filter-advice t("people_filters.filter_advice")
36 +         %div.dropdown.ml-3{"data-controller": "filter-dropdown"}
37 +           %button.btn.dropdown-toggle{ type: 'button', "data-bs-toggle": "dropdown",
38 +             "aria-expanded": "false"}
39 +             %i.fa.fa-plus
40 +             %ul.dropdown-menu#filter-criteria-dropdown{"data-filter-dropdown-target":
41 +               "menu"}
42 +               - @filter_criteria.each do |filter_criterion|
43 +                 - if entry.filter_chain[filter_criterion.to_sym].nil?
44 +                   = link_to t("people_filters.#{filter_criterion}.title"),
45 +                     "/groups/#{@group.id}/people_filters/#{filter_criterion}",
46 +                     class: "dropdown-item", id: "dropdown-option-#{filter_criterion}",
47 +                     data: {turbo_stream: true}
48 +
49 +       - if can?(:create, entry)
50 +         %label.required=PeopleFilter.human_attribute_name(:name)
51 +         = f.input_field :name, placeholder: t('.save_filter_placeholder'), class:
52 +           'mb-3 ', required: false
53 +         = form.input_field :name, placeholder: t('.save_filter_placeholder'), class:
54 +           'mb-3 ', required: false
55 +
56 +       = render 'search_or_save_buttons', f: f
57 +       = render 'search_or_save_buttons', f: form
```

24.3.11 people_filters/_form.html.haml

```
1 += standard_form(path_args(entry), noindent: true, stacked: true) do |form|
2 +
3 + = render 'search_or_save_buttons', f: form
4 +
5 + = render_extensions :form, locals: { f: form }
6 + = form.error_messages
7 +
8 + = render 'range', f: form
9 +
10 + .mt-3.mb-3.ml-3#overview
11 + - if action_name == 'edit'
12 +   %div{id: "role-configuration"}
13 +   - if entry.filter_chain[:role]
14 +     = render(layout: 'layouts/filter', locals: { type: :role }) do
15 +       = render 'role', f: form
16 +
17 +   - if @qualification_kinds.present?
18 +   - if entry.filter_chain[:qualification]
19 +     %div{id: "qualification-configuration"}
20 +     = render(layout: 'layouts/filter', locals: { type: :qualification
21 +   }) do
22 +     = render 'qualification', f: form
23 +
24 +   - if entry.filter_chain[:attributes]
25 +     %div{id: "attributes-configuration"}
26 +     = render(layout: 'layouts/filter', locals: { type: :attributes }) do
27 +       = render 'attributes', f: form
28 +   - if entry.filter_chain[:tag]
29 +     %div{id: "tag-configuration"}
30 +     = render(layout: 'layouts/filter', locals: { type: :tag }) do
31 +       .label-columns
32 +       = render 'tag'
33 +
34 +   - if action_name == 'new'
35 +     %h3#filter-advice Um eine Filterkriterium auszuwählen, bitte klicke
36 +     auf den Hinzufuege-Button
37 +     %div.dropdown.ml-3{"data-controller": "filter-dropdown"}
38 +     %button.btn.dropdown-toggle{ type: 'button', "data-bs-toggle":
39 +     "dropdown", "aria-expanded": "false"}
40 +     %i.fa.fa-plus
41 +
42 +     %ul.dropdown-menu#filter-criteria-dropdown{"data-filter-dropdown-target":
43 +     "menu"}
44 +     - @filter_criteria.each do |filter_criterion|
45 +       - if entry.filter_chain[filter_criterion.to_sym].nil?
46 +         = link_to t("people_filters.#{filter_criterion}.title"),
47 +         "/groups/#{@group.id}/people_filters/#{filter_criterion}",
48 +         class: "dropdown-item", id:
49 +         "dropdown-option-#{filter_criterion}", data: {turbo_stream: true}
```

24.3.12 _qualification.html.haml

```

1 -- filter = entry.filter_chain[:qualification]
2 -- filter_args = filter ? filter.args : {}
3 +- filter_qualifications = entry.filter_chain[:qualification]
4 +- filter_args = filter_qualifications ? filter_qualifications.args : {}
5 +- present_selected_qualifications = filter_qualifications.present? ?
6   filter_qualifications.to_hash[:qualification_kind_ids] : []
7 +- qualification_options =
8   options_from_collection_for_select(@qualification_kinds, :second, :first,
9   present_selected_qualifications)
10
11 -.label-columns
12 - = field_set_tag(t('.prompt_qualification_selection')) do
13 -   - unless can?(:index_full_people, @group)
14 -     .alert.alert-warning= t('.not_enough_permissions')
15 +- present_selected_validity = filter_qualifications.present? ?
16   filter_qualifications.to_hash[:validity] : []
17 +- validity_options = options_from_collection_for_select(@validities, :second,
18   :first, present_selected_validity)
19
20 -.controls
21 - @qualification_kinds.each do |kind|
22 -   dom_id = "qualification_kind_id_#{kind.id}"
23 -   = label_tag(dom_id, class: 'checkbox inline') do
24 -     = check_box_tag("filters[qualification][qualification_kind_ids][]",
25 -       kind.id,
26 -       filter &&
27 -       filter.args[:qualification_kind_ids].include?(kind.id),
28 -       id: dom_id)
29 -   = kind.to_s
30
31 +.label-columns.mr-5
32 + - unless can?(:index_full_people, @group)
33 +   .alert.alert-warning= t('.not_enough_permissions')
34
35 -.controls
36 - &nbsp;
37 +.d-flex.flex-row.align-items-center.mb-2
38 + .w-100
39 + %label= t("people_filters.qualification.title")
40 + .w-90.shown{"data-controller": "form-select"}
41 + = select_tag("filters[qualification][qualification_kind_ids][]",
42 +   qualification_options,
43 +   class: 'form-control tom-select w-100',
44 +   id: 'qualification-select',
45 +   "data-controller": "form-select",
46 +   multiple: true,
47 +   data: { chosen_no_results: t('global.chosen_no_results') },
48 +   placeholder:
49 +   t("people_filters.qualification.prompt_qualification_placeholder") } )
50 + .d-flex.flex-row.align-items-center.mb-2
51 + .w-100
52 + %label= t("people_filters.qualification.criterion")
53 + .w-90.shown.d-flex.flex-column
54 + .d-flex.flex-row
55 +   = label_tag('filters_qualification_match_one', class: 'radio inline') do
56 +     = radio_button_tag('filters[qualification][match]', 'one', true)
57 +     = t('people_filters.simple_radio.match.one')
58 +   .d-flex.flex-row
59 +     = label_tag('filters_qualification_match_all', class: 'radio inline') do
60 +       = radio_button_tag('filters[qualification][match]', 'all',
61 +         filter_args[:match] == 'all', %w(not_active none
62 +         only_expired).include?(filter_args[:validity]) ? { disabled: true } : {})
63 +       = t('people_filters.simple_radio.match.all')

```

24.3.13 _qualification.html.haml

```

1 - .controls
2 -   = label_tag('filters_qualification_match_one', class: 'radio inline') do
3 -     = radio_button_tag('filters[qualification][match]', 'one', true)
4 -     = t('people_filters.simple_radio.match.one')
5 -   = label_tag('filters_qualification_match_all', class: 'radio inline') do
6 -     = radio_button_tag('filters[qualification][match]', 'all',
7 -       filter_args[:match] == 'all', %w(not_active none
8 -       only_expired).include?(filter_args[:validity]) ? { disabled: true } : {})
9 -     = t('people_filters.simple_radio.match.all')
10 -
11 - = field_set_tag(t('.prompt_validity')) do
12 -   %div.row
13 -     = render 'simple_radio',
14 -       attr: "filters[qualification][validity]",
15 -       value: 'active',
16 -       checked: true # first item is checked per default
17 -     = render 'simple_radio',
18 -       attr: "filters[qualification][validity]",
19 -       value: 'reactivateable',
20 -       checked: filter_args[:validity] == 'reactivateable'
21 -     = render 'simple_radio',
22 -       attr: "filters[qualification][validity]",
23 -       value: 'not_active_but_reactivateable',
24 -       checked: filter_args[:validity] == 'not_active_but_reactivateable'
25 -     = render 'simple_radio',
26 -       attr: "filters[qualification][validity]",
27 -       value: 'not_active',
28 -       checked: filter_args[:validity] == 'not_active'
29 -     = render 'simple_radio',
30 -       attr: "filters[qualification][validity]",
31 -       value: 'all',
32 -       checked: filter_args[:validity] == 'all'
33 -     = render 'simple_radio',
34 -       attr: "filters[qualification][validity]",
35 -       value: 'none',
36 -       checked: filter_args[:validity] == 'none'
37 -     = render 'simple_radio',
38 -       attr: "filters[qualification][validity]",
39 -       value: 'only_expired',
40 -       checked: filter_args[:validity] == 'only_expired'
41 -
42 - == field_set_tag(t('.prompt_date'), id: 'reference-date', class:
43 -   'form-horizontal', style: "display: #{{w(all
44 -   none).include?(filter_args[:validity]) ? 'none' : 'block'}}") do
45 -   .control-group.row
46 -     .col-2
47 -       .input-group
48 -       + .d-flex.flex-row.align-items-center.mb-2
49 -       + .w-100
50 -       %label= t("people_filters.qualification.validity")
51 -       + .w-90.shown.d-flex.flex-column{"data-controller": "form-select"}
52 -       = select_tag('filters[qualification][validity]',
53 -         validity_options,
54 -         class: 'form-control tom-select w-100',
55 -         "data-controller": "form-select",
56 -         id: 'qualification-validity-select',
57 -         multiple: false,
58 -         data: { chosen_no_results: t('global.chosen_no_results') },
59 -         placeholder:
60 -         t("people_filters.qualification.prompt_qualification_validity_placeholder") } )

```

24.3.14 _qualification.html.haml

```
1 + .d-flex.flex-row.align-items-center.mb-2
2 +   .w-10
3 +   %label= t('.prompt_date')
4 +   .w-90.shown.d-flex.flex-row{"data-controller": "form-select"}
5 +   .input-group.w-75.mr-3
6 +     %span.input-group-text= icon(:'calendar-alt')
7 +     = text_field_tag("filters[qualification][reference_date]",
8 +       filter_args[:reference_date], class: 'date form-control form-control-sm')
9 -   .col-10.col-form-label
10 -     = t('.reference_date_help_inline')
11 -
12 -= field_set_tag(t('.prompt_year'), id: 'year-scope', class: 'form-horizontal',
13 -   style: "display: #{filter_args[:validity] == 'all' ? 'block' : 'none'};") do
14 -   .control-group.row
15 -     = label_tag(:start_at_year_from, t('.start_at_year_label'), class:
16 -       'control-label col-form-label col-2')
17 -     .col-2
18 -     = number_field_tag("filters[qualification][start_at_year_from]",
19 -       filter_args[:start_at_year_from], class: 'form-control form-control-sm',
20 -       placeholder: 'YYYY')
21 -     .col-1.col-form-label
22 -     = t('.start_at_year_infix')
23 -     .col-2
24 -     = number_field_tag("filters[qualification][start_at_year_until]",
25 -       filter_args[:start_at_year_until], class: 'form-control form-control-sm',
26 -       placeholder: 'YYYY')
27 -   .control-group.row
28 -     = label_tag(:finish_at_year_from, t('.finish_at_year_label'), class:
29 -       'control-label col-form-label col-2')
30 -     .col-2
31 -     = number_field_tag("filters[qualification][finish_at_year_from]",
32 -       filter_args[:finish_at_year_from], class: 'form-control form-control-sm',
33 -       placeholder: 'YYYY')
34 -     .col-1.col-form-label
35 -     = t('.finish_at_year_infix')
36 -     .col-2
37 -     = number_field_tag("filters[qualification][finish_at_year_until]",
38 -       filter_args[:finish_at_year_until], class: 'form-control form-control-sm',
39 -       placeholder: 'YYYY')
40 +   .w-25.col-form-label
41 +     = t('.reference_date_help_inline')
```

24.3.15 _role.html.haml

```

1 -- filter = entry.filter_chain[:role]
2 +- filter_roles = entry.filter_chain[:role]
3 +- present_selected_roles = filter_roles.present? ?
  filter_roles.to_hash[:role_type_ids] : []
4 +- role_options = options_from_collection_for_select(@roles, :second, :first,
  present_selected_roles)
5
6 -.label-columns
7 - = field_set_tag(t('.prompt_role_selection')) do
8 -   - @role_types.each do |layer, groups|
9 -     .layer{ class: [group.klass.label,
10      @group.layer_group.class.label].include?(layer) && 'same-layer' }
11 -     %h4.filter-toggle= layer
12 -     - groups.each do |group, role_types|
13 -       .group.control-group{ class: group == @group.klass.label &&
14         'same-group' }
15 -       %h5.filter-toggle= group
16 -       .controls
17 -       - role_types.each do |role_type|
18 -         - id = "filters_role_role_type_ids_#{role_type.id}"
19 -         = label_tag(nil, id, class: 'checkbox inline') do
20 -           = check_box_tag("filters[role][role_type_ids][]",
21             role_type.id,
22             filter &&
23             filter.to_hash[:role_types].include?(role_type.to_s),
24             id: id)
25 -           = role_type.label
26 +- present_selected_kinds = filter_roles.present? ? filter_roles.to_hash[:kind] :
  []
27 +- kind_options = options_from_collection_for_select(@kinds, :second, :first,
  present_selected_kinds)
28
29 - = field_set_tag(t('.prompt_role_duration_selection')) do
30 -   .row.control-group.mt-3
31 -   = render 'simple_radio',
32 -     attr: "filters[role][kind]",
33 -     value: 'active_today',
34 -     checked: !filter ||
35 -       !Person::Filter::Role::KINDS.include?(filter.args[:kind])
36 -   .row.control-group.mt-2
37 -   .d-flex.col-xl-6
38 -   %label.mt-1.nowrap.w-50=t('.prompt_role_duration_from')
39 -   .input-group.w-50
40 -   %span.input-group-text= icon(:'calendar-alt')
41 -   = text_field_tag("filters[role][start_at]", filter &&
42     filter.args[:start_at], class: 'date col-2 form-control form-control-sm')
43 -   %label.mx-2.mt-1.w-50=t('.prompt_role_duration_until')
44 -   .input-group.w-50
45 -   %span.input-group-text= icon(:'calendar-alt')
46 -   = text_field_tag("filters[role][finish_at]", filter &&
47     filter.args[:finish_at], class: 'date col-2 form-control form-control-sm')
48 -   .row.control-group.mt-2
49 -   - Person::Filter::Role::KINDS.each do |key|
50 -     = render 'simple_radio',
51 -       attr: "filters[role][kind]",
52 -       value: key,
53 -       checked: filter && filter.args[:kind] == key
54 -   .row.control-group.mt-3
55 -   = label_tag(nil, 'filters_role_include_archived', class: 'checkbox inline
56   mt-2') do
57 -   = check_box_tag('filters[role][include_archived]', true, filter &&
58     true?(filter.args[:include_archived]))
59 -   = t('.include_archived')

```

24.3.16 _role.html.haml

```
1 +.label-columns.mr-5
2 + .d-flex.flex-row.align-items-center.mb-2
3 +   .w-10
4 +   %label= t("people_filters.role.title")
5 +   .w-90
6 +   = select_tag('filters[role][role_type_ids][]',
7 +               role_options,
8 +               class: 'form-control tom-select w-100',
9 +               "data-controller": "form-select",
10 +              id: 'role-select',
11 +              multiple: true,
12 +              data: { chosen_no_results: t('global.chosen_no_results'),
13 +                    placeholder: t('.prompt_role_placeholder') } )
13 + .d-flex.flex-row.align-items-center.control-group.mb-2
14 +   .w-10
15 +   %label= t("people_filters.role.duration_title")
16 +   .w-90.d-flex.flex-row.align-items-center
17 +     .input-group.w-50
18 +       %span.input-group-text= icon(:'calendar-alt')
19 +       = text_field_tag("filters[role][start_at]", filter_roles &&
20 +         filter_roles.args[:start_at], class: 'date form-control')
21 +       %label.mx-2.mt-1=t('.prompt_role_duration_until')
22 +     .input-group.w-50
23 +       %span.input-group-text= icon(:'calendar-alt')
24 +       = text_field_tag("filters[role][finish_at]", filter_roles &&
25 +         filter_roles.args[:finish_at], class: 'date form-control')
24 + .d-flex.flex-row.align-items-center
25 +   .w-10
26 +   %label= t("people_filters.role.role_kind_title")
27 +   .w-90{"data-controller": "form-select"}
28 +   = select_tag('filters[role][kind][]',
29 +               kind_options,
30 +               class: 'form-control tom-select w-100',
31 +               "data-controller": "form-select",
32 +               id: 'role-kind-select',
33 +               multiple: true,
34 +               data: { chosen_no_results: t('global.chosen_no_results'),
35 +                     placeholder: t('.prompt_role_kind_placeholder') } )
```


24.3.17 _tag.html.haml

```
1 -- filter = entry.filter_chain[:tag]
2 -- selected = filter ? filter.to_hash[:names] : []
3 -- options = options_from_collection_for_select(@possible_tags, :second, :first,
4   selected)
5 +- present_filter_tags = entry.filter_chain[:tag]
6 +- present_selected = present_filter_tags ? present_filter_tags.to_hash[:names] :
7   []
8 +- present_options = options_from_collection_for_select(@possible_tags, :second,
9   :first, present_selected)
10
11 -%p= t('.prompt_tag_selection')
12 +- absent_filter = entry.filter_chain[:tag_absence]
13 +- absent_selected = absent_filter ? absent_filter.to_hash[:names] : []
14 +- absent_options = options_from_collection_for_select(@possible_tags, :second,
15   :first, absent_selected)
16
17 -. shown
18 - = select_tag('filters[tag][names][]',
19   options,
20   class: 'form-control tom-select',
21   multiple: true,
22   data: { chosen_no_results: t('global.chosen_no_results'),
23     placeholder: t('.prompt_tag_placeholder') } )
24
25 +.d-flex.flex-column
26 + .d-flex.flex-row.align-items-center.mb-2
27 + .w-100
28 +   %label= t('.prompt_tag_selection')
29 + .w-90.shown
30 +   = select_tag('filters[tag][names][]',
31     present_options,
32     class: 'form-control tom-select w-100',
33     "data-controller": "form-select",
34     id: 'present-tag-select',
35     multiple: true,
36     data: { chosen_no_results: t('global.chosen_no_results'),
37       placeholder: t('.prompt_tag_placeholder') } )
38
39 +.d-flex.flex-row.align-items-center
40 + .w-100
41 +   %label= t('.prompt_tag_absence_selection')
42 + .w-90.shown
43 +   = select_tag('filters[tag_absence][names][]',
44     absent_options,
45     class: 'form-control tom-select w-100',
46     id: 'absent-tag-select',
47     "data-controller": "form-select",
48     multiple: true,
49     data: { chosen_no_results: t('global.chosen_no_results'),
50       placeholder: t('.prompt_tag_placeholder') } )
```

24.3.18 create.turbo_stream.haml

```
1 += turbo_stream.prepend "overview" do
2 +   %div{id: "#{@filter_criterion}-configuration"}
3 +     = render(layout: 'layouts/filter', locals: { type: @filter_criterion.to_sym,
4         delete: true}) do
5 +       = render @filter_criterion
6 += turbo_stream.remove "dropdown-option-#{@filter_criterion}"
7 += turbo_stream.remove "filter-advice"
```

24.3.19 delete.turbo_stream.haml

```
1 += turbo_stream.remove "#{@filter_criterion}-configuration"
2 +
3 += turbo_stream.append "filter-criteria-dropdown" do
4 +   = link_to t("people_filters.#{@filter_criterion}.title"),
5       "/groups/#{@group.id}/people_filters/#{@filter_criterion}",
6       class: "dropdown-item", id: "dropdown-option-#{@filter_criterion}",
7       data: {turbo_stream: true}
```

24.3.20 subscriber/_form.html.haml

```
1 - .accordion
2 -   = render(layout: 'people_filters/filter', locals: { entry: @mailing_list,
3       type: :attributes }) do
4 -     = render 'people_filters/attributes', f: f, entry: @mailing_list
5 -
6 -   - FeatureGate.if('person_language') do
7 -     = render(layout: 'people_filters/filter', locals: { type: :language }) do
8 -       = render 'people_filters/language', f: f, entry: @mailing_list
9 -
10 -   = render_extensions('filter_form', locals: { entry: entry })
11 + = render(layout: 'layouts/filter', locals: { entry: @mailing_list, type:
12     :attributes, delete: false}) do
13 +   = render 'people_filters/attributes', f: f, entry: @mailing_list
14 + = render_extensions('filter_form', locals: { entry: entry })
```

24.3.21 routes.rb

```
1 +   # Turbo Routes
2 +   get 'people_filters/:filter_criterion' => 'people_filters#filter_criterion'
3 +   delete 'people_filters/turbo/:filter_criterion' =>
4     'people_filters#filter_criterion'
```

24.3.22 global_conditions_spec.rb

```
1  + # frozen_string_literal: true
2  +
3  + # Copyright (c) 2024, Schweizer Alpen-Club. This file is part of
4  + # hitobito_sac_cas and licensed under the Affero General Public License version 3
5  + # or later. See the COPYING file at the top-level directory or at
6  + # https://github.com/hitobito/hitobito_sac_cas.
7  +
8  + require "spec_helper"
9  +
10 + describe "Global Conditions", js: true do
11 +   let(:lang) { :de }
12 +   let(:top_leader) { people(:top_leader) }
13 +   let(:alice) { people(:bottom_member) }
14 +
15 +   context "modify" do
16 +     let(:path) {
17 +       edit_group_mailing_list_filter_path(group_id: MailingList.first.group_id,
18 +       mailing_list_id: MailingList.first.id)
19 +     }
20 +
21 +     before {
22 +       sign_in(top_leader)
23 +       visit path
24 +     }
25 +
26 +     it "can add global conditions" do
27 +       select "Firmenname", from: "attribute_filter"
28 +
29 +       first(".attribute_constraint_dropdown").find("option[value='not_match']").select_option
30 +       first("input.form-control[type='text']").set("Puzzle ITC")
31 +
32 +       select "Alter", from: "attribute_filter"
33 +
34 +       all(".attribute_constraint_dropdown")[1].find("option[value='equal']").select_option
35 +       all("input.form-control[type='text']")[1].set(56)
36 +
37 +       first(".btn.btn-primary", text: "Speichern").click
38 +
39 +       within("div#main") do
40 +         expect(page).to have_text("Firmenname enthaelt nicht Puzzle ITC")
41 +         expect(page).to have_text("Alter ist genau 56")
42 +       end
43 +
44 +       context "member access" do
45 +         before {
46 +           Rails.env.stub(production?: true)
47 +           sign_in(alice)
48 +         }
49 +
50 +         it "can't access global conditions" do
51 +           visit path
52 +           expect(page).to have_selector("div.alert-danger", text: "Sie sind nicht
53 +           berechtigt, diese Seite anzuzeigen")
54 +         end
55 +       end
56 +     end
57 +   end
58 + end
```

24.3.23 people_filter_spec.rb

```
1   + # frozen_string_literal: true
2   +
3   + # Copyright (c) 2024, Schweizer Alpen-Club. This file is part of
4   + # hitobito_sac_cas and licensed under the Affero General Public License version 3
5   + # or later. See the COPYING file at the top-level directory or at
6   + # https://github.com/hitobito/hitobito_sac_cas.
7   +
8   + require "spec_helper"
9   +
10  + describe "PeopleFilter", js: true do
11  +   let(:lang) { :de }
12  +   let(:top_leader) { people(:top_leader) }
13  +   let(:alice) { people(:bottom_member) }
14  +
15  +   context "filtering" do
16  +     let(:path) { new_group_people_filter_path(group_id: Group.first.id) }
17  +
18  +     before {
19  +       sign_in(top_leader)
20  +       visit path
21  +       find(".btn.dropdown-toggle").click
22  +     }
23  +
24  +     context "tags" do
25  +       before {
26  +         alice.tag_list.add("lorem")
27  +         alice.tag_list.add("ipsum")
28  +         alice.save!
29  +         find("#dropdown-option-tag").click
30  +       }
31  +
32  +       it "can filter by present tags" do
33  +         expect(page).to have_selector("div#tag-configuration")
34  +         find("#present-tag-select-ts-control").set("lorem")
35  +         find("#present-tag-select-opt-2").click
36  +
37  +         first(".btn.btn-primary", text: "Suchen").click
38  +         within(".table.table-striped.table-hover") do
39  +           # Head table row and table row of found user
40  +           expect(all("tr").size).to eq(2)
41  +           expect(page).to have_selector("#person_#{alice.id}")
42  +         end
43  +       end
44  +
45  +       it "can filter by absent tags" do
46  +         visit path
47  +         find(".btn.dropdown-toggle").click
48  +         find("#dropdown-option-tag").click
49  +         expect(page).to have_selector("div#tag-configuration")
50  +         find("#absent-tag-select-ts-control").set("ipsum")
51  +         find("#absent-tag-select-opt-1").click
52  +
53  +         first(".btn.btn-primary", text: "Suchen").click
54  +         expect(page).to have_no_selector(".table.table-striped.table-hover")
55  +         expect(page).to have_no_selector("#person_#{alice.id}")
56  +       end
57  +     end
58  +   end
59  + end
```

24.3.24 people_filter_spec.rb

```
1 + context "roles" do
2 +   before {
3 +     find("#dropdown-option-role").click
4 +   }
5 +
6 +   it "can filter by present roles" do
7 +     expect(page).to have_selector("div#role-configuration")
8 +     # Select role
9 +     find("#role-select-ts-control").set(alice.roles.first.label)
10 +    find("#role-select-opt-2").click
11 +
12 +    # Select date
13 +    find("#filters_role_start_at").set("2025-03-18")
14 +    find("#filters_role_finish_at").set("2025-03-30")
15 +
16 +    # Select role kind
17 +    find("#role-kind-select-ts-control").set("Aktiv")
18 +    find("#role-kind-select-opt-1").click
19 +
20 +    first(".btn.btn-primary", text: "Suchen").click
21 +    within(".table.table-striped.table-hover") do
22 +      # Head table row and table row of found user
23 +      expect(all("tr").size).to eq(2)
24 +      expect(page).to have_selector("#person_#{alice.id}")
25 +    end
26 +  end
27 + end
28 +
29 + context "qualification" do
30 +   before {
31 +     find("#dropdown-option-qualification").click
32 +   }
33 +
34 +   it "can filter by qualifications" do
35 +     expect(page).to have_selector("div#qualification-configuration")
36 +
37 +     # Select qualification
38 +     find("#qualification-select-ts-control")
39 +       .set(alice.qualifications.first.qualification_kind.label)
40 +     find("#qualification-select-opt-4").click
41 +
42 +     # Select reference date
43 +     find("#filters_qualification_reference_date").set("2025-03-05")
44 +
45 +     first(".btn.btn-primary", text: "Suchen").click
46 +     within(".table.table-striped.table-hover") do
47 +       # Head table row and table row of found user
48 +       expect(all("tr").size).to eq(2)
49 +       expect(page).to have_selector("#person_#{alice.id}")
50 +     end
51 +   end
52 + end
```

24.3.25 people_filter_spec.rb

```
1  +   context "attributes" do
2  +     before {
3  +       find("#dropdown-option-attributes").click
4  +     }
5  +
6  +     it "can filter by attributes" do
7  +       expect(page).to have_selector("div#attributes-configuration")
8  +
9  +       select "PLZ", from: "attribute_filter"
10  +
11  +       first(".attribute_constraint_dropdown").find("option[value='equal']").select_option
12  +       first("input.form-control[type='text']").set(alice.zip_code)
13  +
14  +       select "Vorname", from: "attribute_filter"
15  +
16  +       all(".attribute_constraint_dropdown")[1].find("option[value='equal']").select_option
17  +       all("input.form-control[type='text']")[1].set(alice.first_name)
18  +
19  +       first(".btn.btn-primary", text: "Suchen").click
20  +       within(".table.table-striped.table-hover") do
21  +         # Head table row and table row of found user
22  +         expect(all("tr").size).to eq(2)
23  +         expect(page).to have_selector("#person_#{alice.id}")
24  +       end
25  +     end
26  +
27  +     it "is xss-attack immune" do
28  +       expect(page).to have_selector("div#attributes-configuration")
29  +
30  +       select "Ort", from: "attribute_filter"
31  +
32  +       first(".attribute_constraint_dropdown").find("option[value='match']").select_option
33  +       first("input.form-control[type='text']").set("<script>alert('Hacked!');</script>")
34  +
35  +       first(".btn.btn-primary", text: "Suchen").click
36  +       expect {
37  +         page.driver.browser.switch_to.alert
```

24.3.26 people_filter_spec.rb

```
1 + context "saving" do
2 +   let(:path) { new_group_people_filter_path(group_id: Group.first.id) }
3 +
4 +   before {
5 +     sign_in(top_leader)
6 +     visit path
7 +     find(".btn.dropdown-toggle").click
8 +   }
9 +
10 + it "can save filter" do
11 +   find("#dropdown-option-role").click
12 +
13 +   find("#role-select-ts-control").click
14 +   find("#role-select-opt-1").click
15 +
16 +   filter_name = "Filtername"
17 +   fill_in "people_filter_name", with: filter_name
18 +   first(".btn.btn-primary", text: "Suche speichern").click
19 +
20 +   expect(page).to have_selector("div.alert-success", text: "Filter
    #{filter_name} wurde erfolgreich erstellt.")
21 + end
22 + end
23 + end
24 +
25 + context "member access" do
26 +   let(:path) { new_group_people_filter_path(group_id: Group.find_by(name:
    "Top").id) }
27 +
28 +   before {
29 +     Rails.env.stub(production?: true)
30 +     sign_in(alice)
31 +   }
32 +
33 + it "can't access people filtering" do
34 +   visit path
35 +   expect(page).to have_selector("div.alert-danger", text: "Sie sind nicht
    berechtigt, diese Seite anzuzeigen")
36 + end
37 + end
38 + end
```

24.3.27 role.rb

```
1 def to_hash
2 +   merge_duration_args(role_types: args[:role_types], role_type_ids:
    args[:role_type_ids], kind: args[:kind])
3 end
```

24.3.28 _main.scss

```
1 def to_hash
2 +   merge_duration_args(role_types: args[:role_types], role_type_ids:
3   args[:role_type_ids], kind: args[:kind])
4 end
```

24.3.29 qualifications.yml

```
1 + bottom_member:
2 +   person: bottom_member
3 +   origin: "Verwalter"
4 +   qualification_kind: sl
5 +   start_at: 2025-03-04
6 +   finish_at: 2025-03-30
```

24.3.30 views.fr.yml

```
1 people_filters:
2 +   filters_role_kind:
3 +     active_today: Actuellement actif
4 +     active: Actif
5 +     created: Cree
6 +     deleted: Supprime
7 +     inactive: Inactif
8
9 tag:
10 +   prompt_tag_selection: Inclure
11 +   prompt_tag_absence_selection: Exclure
12
13 role:
14 +   role_kind_title: Validite
15 +   prompt_role_placeholder: Saisir le nom du role souhaite
16 +   prompt_role_kind_placeholder: Saisir la validite souhaitee
17 +   duration_title: Date de reference
18
19 qualification:
20 +   criterion: Critere
21 +   validity: Validite
22 +   validity_label:
23 +     active: Actif
24 +     reactivateable: Reactivable
25 +     not_active_but_reactivateable: Inactif mais reactivable
26 +     not_active: Non actif
27 +     all: Tous
28 +     none: Personne
29 +     only_expired: Uniquement les produits expires
30 +   prompt_qualification_placeholder: Saisir le nom de qualification souhaite
31 +   prompt_qualification_validity_placeholder: Saisir la validite souhaitee
```


24.3.31 views.it.yml

```
1 people_filters:
2 +   filters_role_kind:
3 +     active: Attualmente attivo
4 +     created: Creato
5 +     deleted: Annullato
6 +     inactive: Inattivo
7
8 tag:
9 +   prompt_tag_selection: Includere
10 +   prompt_tag_absence_selection: Escludere
11
12 role:
13 +   role_kind_title: Validita
14 +   prompt_role_placeholder: Inserire il nome del ruolo desiderato
15 +   prompt_role_kind_placeholder: Inserire la validita desiderata
16 +   duration_title: Data chiave
17
18 qualification:
19 +   criterion: Criterio
20 +   validity: Validita
21 +   validity_label:
22 +     active: Attivo
23 +     reactivateable: Riattivabile
24 +     not_active_but_reactivateable: Non e attivo ma puo essere riattivato
25 +     not_active: Non attivo
26 +     all: Tutti
27 +     none: Nessuno
28 +     only_expired: Solo scaduto
29 +   prompt_qualification_placeholder: Inserire il nome della qualifica desiderata
30 +   prompt_qualification_validity_placeholder: Inserire la validita desiderata
```

24.3.32 views.de.yml

```
1 people_filters:
2 +   filters_role_kind:
3 +     active: Aktiv
4 +     created: Erstellt
5 +     deleted: Geloescht
6 +     inactive: Inaktiv
7
8 tag:
9 +   prompt_tag_selection: Einschliessen
10 +   prompt_tag_absence_selection: Ausschliessen
11
12 role:
13 +   role_kind_title: Gueltigkeit
14 +   prompt_role_placeholder: Gewuenschter Rollenname eingeben
15 +   prompt_role_kind_placeholder: Gewuenschte Gueltigkeit eingeben
16 +   duration_title: Stichdatum
17
18 qualification:
19 +   criterion: Kriterium
20 +   validity: Gueltigkeit
21 +   validity_label:
22 +     active: Aktiv
23 +     reactivateable: Reactivable
24 +     not_active_but_reactivateable: Nicht aktiv aber reaktivierbar
25 +     not_active: Nicht aktiv
26 +     all: Alle
27 +     none: Keine
28 +     only_expired: Nur abgelaufene
29 +   prompt_qualification_placeholder: Gewuenschter Qualifikationsname
30 +   prompt_qualification_validity_placeholder: Gewuenschte Gueltigkeit
```

24.3.33 views.en.yml

```
1 people_filters:
2 +   filters_role_kind:
3 +     active: Active
4 +     created: Created
5 +     deleted: Deleted
6 +     inactive: Inactive
7
8 tag:
9 +   prompt_tag_selection: Include
10 +   prompt_tag_absence_selection: Exclude
11
12 role:
13 +   role_kind_title: Validity
14 +   prompt_role_placeholder: Enter the desired role name
15 +   prompt_role_kind_placeholder: Enter the desired validity
16 +   duration_title: Key date
17
18 qualification:
19 +   criterion: Criterion
20 +   validity: Validity
21 +   validity_label:
22 +     active: Active
23 +     reactivateable: Reactivateable
24 +     not_active_but_reactivateable: Not active but reactivateable
25 +     not_active: Not active
26 +     all: All
27 +     none: None
28 +     only_expired: Only expired
29 +   prompt_qualification_placeholder: Enter desired qualification name
30 +   prompt_qualification_validity_placeholder: Enter the desired validity
```