

Ämtliplantool
ICT-Campus

Monatsplanung

PUNCS
Beeliverys
Plappls

	August	September	Oktober	November	Dezember	Januar
Medienraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Pausenraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Umgebung	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Matinée	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL

	Februar	März	April	Mai	Juni	Juli
Medienraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Pausenraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Umgebung	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Matinée	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL

Heute
 Einteilungen

Admin-Bereich
 Ämtlikatalog
 Teams & Lernende
Monatsplanung
 Arbeitstage

Eingeloggt als Coach

Ämtliplantool ICT-Campus

Fullstack-Webapplikation zur Verwaltung der Ämtlieinteilung am ICT-Campus - Basisausbildung Informatik der Schweizerischen Post AG

Modul Selbständige, freiwillige Vertiefungsarbeit

Eingereicht von Jonas Vetsch

Eingereicht bei Michael Moser

Datum (vorläufig) 22. April 2026

DIES IST EIN ENTWURF, da die Arbeit noch nicht abgeschlossen ist. Das Backend ist bereits entwickelt (siehe GitHub Repo). Das Frontend ist noch ausstehend. Das Projekt wird voraussichtlich bis Juli 2026 abgeschlossen.

Inhaltsverzeichnis

1	Abstract.....	3
2	Ausgangslage	3
	2.1 Situation Heute (2025).....	3
	2.2 Zielgruppe.....	3
	2.3 Ziele nach SMART	3
	2.4 Projekt-Ausschlüsse (Out of Scope).....	3
	2.5 Rollen	4
3	Anforderungen	4
	3.1 Lastenheft.....	4
	3.2 Anforderungen.....	5
4	Use Case Diagramm.....	7
5	Mockups.....	8
	5.1 Login.....	8
	5.2 Coach-Login	8
	5.3 Teamverantwortlich-Login	13
	5.4 Lernenden-Login	15
6	Datenbank.....	16
	6.1 ERD	16
7	Benutzer, Rollen und Rechte	17
	7.1 Logins	17
	7.2 Rollen	17
	7.3 Rechte (Permissions)	18
8	API Contracts (Endpoint-Design, Verträge).....	21
	8.1 Stammdaten (Jonas)	21
	8.2 Monatszuteilungen (Jonas)	30
	8.3 Tageseinteilungen (Linus)	33
	8.4 Login.....	33
9	Backend	34
	9.1 Technologien	34
	9.2 Backend.....	34
	9.3 Dokumentation der wichtigsten Mechanismen	34
	9.4 BA-Jahrgangwechsel / Reset	34
10	Frontend.....	34
	10.1 Technologien	34
	10.2 Error-Handling-Strategie im Frontend	35
11	Testfälle	38
12	Integration	38

13	Auswertung	38
13.1	Auswertung der Anforderungen.....	38
13.2	Zielprüfung.....	38
13.3	Reflexion.....	38
13.4	Fazit.....	38

Links zur Zusammenarbeit

Kanban-Board

<https://trello.com/in-vite/b/694b18e1624e2b9c36e6040a/ATTI2235612e953f0e2a271f0e0d0282a5ba94C8E313/baemtli>

ERD-Design

https://lucid.app/lucidchart/90c40483-88e3-4f12-bc30-1abdca9f97fb/edit?invitationId=inv_366c6437-1009-4ca8-97a1-68a8f2112678&page=0_0#

Mockup-Design

<https://www.figma.com/design/oy5CYokC0t9I9RVOmQw7IT/BAemtli?node-id=0-1&t=vum-zgghKMWuOZvko-1>

GitHub Repo

https://github.com/BA-2025-2026/JAVA_BAemtli

1 Abstract

2 Ausgangslage

2.1 Situation Heute (2025)

2.2 Zielgruppe

2.3 Ziele nach SMART

1. **Stammdatenverwaltung durch Coaches:** Bis zum Projektabschluss können Coaches der Basisausbildung (BA) sowohl die Lernenden (Vor- und Nachname, Teamzugehörigkeit) als auch die Teams (Teamname) zentral verwalten (CRUD).
2. **Ämtli-Verwaltung durch Coaches:** Bis zum Projektabschluss können Coaches die Ämtlikategorien (z. B. Medienraum, Pausenraum) verwalten (CRUD) und diese über einen Status (aktiv/inaktiv) ins Tagesgeschäft einbinden/vom Tagesgeschäft entfernen, ohne Daten dauerhaft löschen zu müssen.
3. **Monatsplanung durch Coaches:** Bis zum Projektabschluss können Coaches im Ämtliplantool die Teams den jeweiligen Ämtlikategorien pro Monat zuordnen (CRUD).
4. **Tagesplanung durch Lernende:** Bis zum Projektabschluss kann pro Team eine verantwortliche Person die Teammitglieder eigenständig den täglichen Ämtleinheiten des aktuellen Monats zuweisen (CRUD). Andere Teammitglieder sowie Coaches haben für diese Zuweisungen ausschliesslich Leserechte.
5. **Einsicht für Coaches:** Bis zum Projektabschluss verfügen Coaches über eine tagesaktuelle Live-Einsicht in **alle** Ämtlizuteilungen, sowohl auf Tages- als auch auf Monatsebene (Read).
6. **Einsicht für Lernende:** Bis zum Projektabschluss haben alle Lernenden eine tagesaktuelle Live-Einsicht in die Ämtlizuteilungen **ihres eigenen Teams** (Tages- und Monatsansicht, Read).
7. **Plattformoptimierung:** Das Ämtliplantool ist bis zum Projektabschluss für die Nutzung auf Desktopgeräten in Browserumgebungen optimiert.

2.4 Projekt-Ausschlüsse (Out of Scope)

Keine Automatisierung: Keine automatisierte Zuteilung von Personen zu Ämtlis; die Planung erfolgt rein manuell durch die Verantwortlichen.

Kein Benachrichtigungssystem: Benachrichtigungsfunktionen oder Erinnerungs-Mails (z. B. Reminder-Service) sind nicht Bestandteil des Grundumfangs.

Keine Schreibrechte für Coaches bei Tagesplanung: Coaches besitzen ausschliesslich Leserechte auf die täglichen Personenzuteilungen; die Datenpflege obliegt den Teams.

Fokus auf Einjährigkeit: Keine Unterstützung für Mehrjährigkeit oder Archivierung vergangener Jahrgänge; das System wird nach Abschluss eines BA-Jahres (z.B. Ende Juli) bereinigt.

Kein Mobile-Design: Die Benutzeroberfläche wird nicht für mobile Endgeräte (Smartphones) optimiert; der Fokus liegt rein auf der Browser-Nutzung am Desktop.

Keine manuelle Historien-Bearbeitung: Vergangene Monate können nur eingesehen, aber nicht mehr nachträglich verändert werden (Read-only Historie).

Kein automatischer Datenimport: Neue Lernende und Teams werden durch Coaches bzw. Verantwortliche manuell im System angelegt (kein automatisierter Import aus externen Listen).

2.5 Rollen

2.5.1 Coach (Administrator)

Die Coaches fungieren als Systemadministrator. Sie verwalten die Stammdaten (Lernende, Teams, Ämtlikategorien) und legen den groben Rahmen für die Planung fest. Sie entscheiden, welches Team in welchem Monat für welches Ämtli (z. B. Medienraum) zuständig ist. Sie haben lesenden Zugriff auf alle Detailplanungen, greifen aber nicht aktiv in die tägliche Personenzuteilung der Teams ein.

2.5.2 Teamverantwortliche*r (Team-Planer)

Diese Rolle ist die operative Schnittstelle innerhalb eines jeden Teams. Pro Team hat genau eine Person diese erweiterten Rechte. Der/die Teamverantwortliche ist dafür zuständig, die Mitglieder des eigenen Teams den konkreten Tagen im Monat zuzuweisen. Diese Rolle besitzt Schreibrechte (CRUD) für die Tagesplanung des eigenen Teams, kann aber keine globalen Einstellungen oder andere Teams verändern.

2.5.3 Lernende (Team-Mitglieder)

Dies ist die Standardrolle für alle anderen Teammitglieder. Sie dient primär der Information und Transparenz. Lernende können jederzeit einsehen, welches Teammitglied an welchem Tag für welches Ämtli eingeteilt ist (Live-Einsicht). Ihr Zugriff ist auf die Leseberechtigung (Read-only) für das **eigene** Team beschränkt.

3 Anforderungen

3.1 Lastenheft

Vgl. Lastenheft von Michael Moser.

3.2 Anforderungen

3.2.1 Funktionale Anforderungen

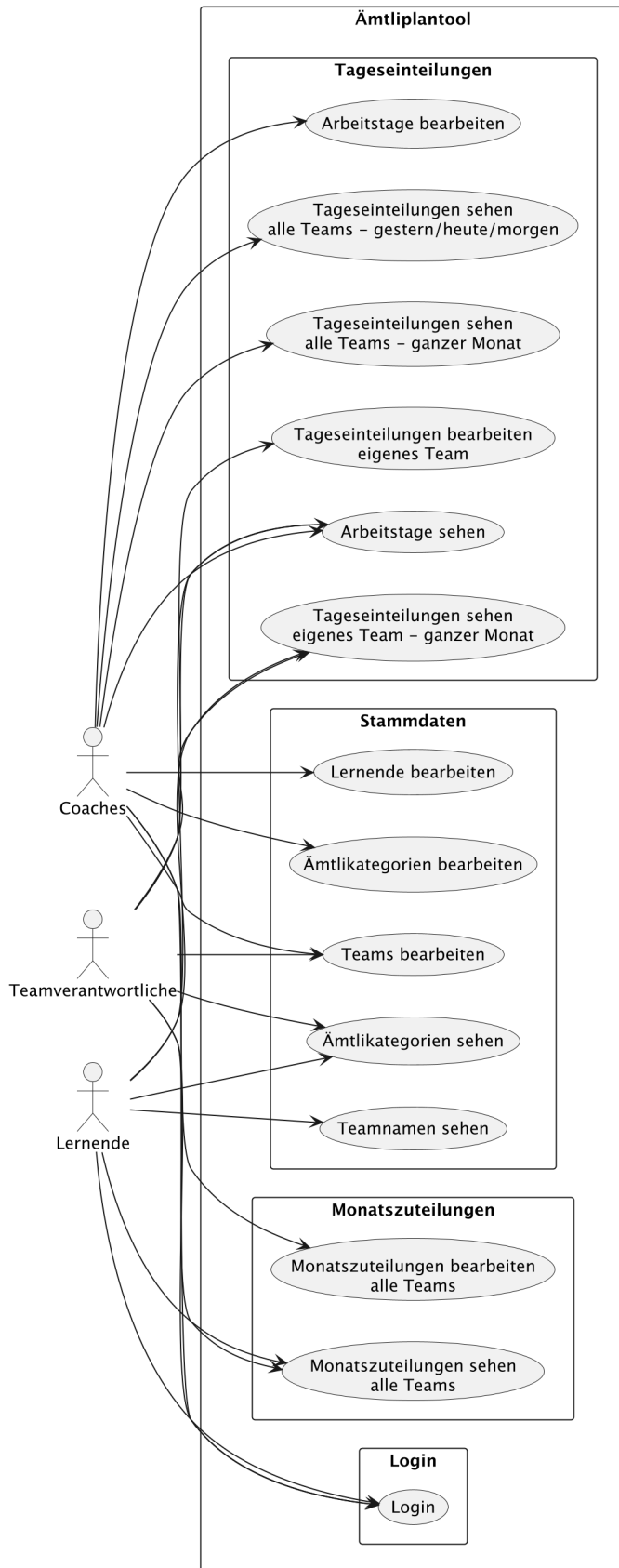
ID	Anforderung	Priorität
F-01	Authentifizierung: Das System ermöglicht Logins für Coaches (nur ein einziges Login, global), Teamverantwortliche (ein Login pro Team) und Lernende (ein Login pro Team).	Muss
F-02	Benutzerverwaltung: Coaches können Lernende (Vorname, Nachname, Teamzugehörigkeit) und Teams anlegen, bearbeiten und löschen.	Muss
F-03	Ämtlikatalog: Coaches können Ämtli-Kategorien (z. B. "Medienraum") inkl. Beschreibungstext erstellen (CRUD) und auf aktiv/inaktiv setzen.	Muss
F-04	Monatliche Team-Zuweisung (Monatsplanung): Coaches können Teams für jeden spezifischen Monat einer oder mehreren Ämtli-Kategorien zuordnen.	Muss
F-05	Einteilungen (Tagesplanung): Der/die Teamverantwortliche kann Teammitglieder einzelnen Tagen und Ämtlis innerhalb des aktuellen Monats zuweisen. Wenn ein Team in einem Monat mehrere Ämtli innehat, sind in der Tagesplanung für jeden Tag auch mehrere Personen einteilbar. Pro Ämtli und Tag eine Person (Bsp.: Am 20.01. macht Maria Musterfrau den Pausenraum, Max Blumenberger die Umgebung).	Muss
F-06	Lesezugriffe Einteilungen: Coaches und reguläre Lernende haben auf die Einteilungen nur Lesezugriff.	Muss
F-07	Übersicht (Coach): Das System bietet eine tagesaktuelle Gesamtansicht über alle laufenden Ämtlis und Verantwortlichkeiten.	Muss
F-08	Übersicht (Lernende): Lernende sehen die aktuelle Tages- und Monatsplanung ausschliesslich für ihr eigenes Team.	Muss
F-09	Historie: Vergangene Monate müssen im System erhalten bleiben und für alle Rollen lesbar (Read-only) sein.	Muss
F-10	Datenpersistenz: Alle Eingaben werden dauerhaft in einer Datenbank gespeichert und bleiben bei Systemneustart erhalten.	Muss
F-11	Export/Druck: Das System bietet eine zum Drucken oder Exportieren optimierte Ansicht der Monatspläne.	Kann
F-12	Kalenderansicht: Die Planung und Einsicht erfolgt über ein grafisches Kalender- bzw. Monatsraster.	Soll

F-13	Jahresübersicht: Eine zusammenfassende Ansicht aller Zuteilungen über das gesamte Ausbildungsjahr hinweg.	Kann
F-14	Arbeitstagsauswahl für Coaches: Das System bietet den Coaches eine Funktion, um festzulegen, welche Tage des aktuellen BA-Jahres BA-Arbeitstage sind und welche nicht (WORKDAY). Standardmässig könnten beim Anlegen eines neuen BA-Jahres dort immer bereits Mi-Fr angewählt sein. Die Coaches müssten dann nur noch die Änderungen für Ferien, Feiertage und ganz-wöchige BA-Wochen machen.	Muss
F-15	Fairnessübersicht für Teamverantwortliche: Beim Teamverantwortlichen-Login (ggf. auch bei den Coaches) gibt es eine Übersicht, die für jede*n Lernende*n (nur des eigenen Teams) die Summe der bereits gemachten und der in der Zukunft liegenden aber bereits zugeteilten Ämtleinheiten anzeigt. Damit kann die teamverantwortliche Person ein Auge auf die faire Verteilung der Arbeitslast haben und bei der Einteilung des jeweils nächsten Monats dies berücksichtigen.	Kann

3.2.2 Nichtfunktionale Anforderungen

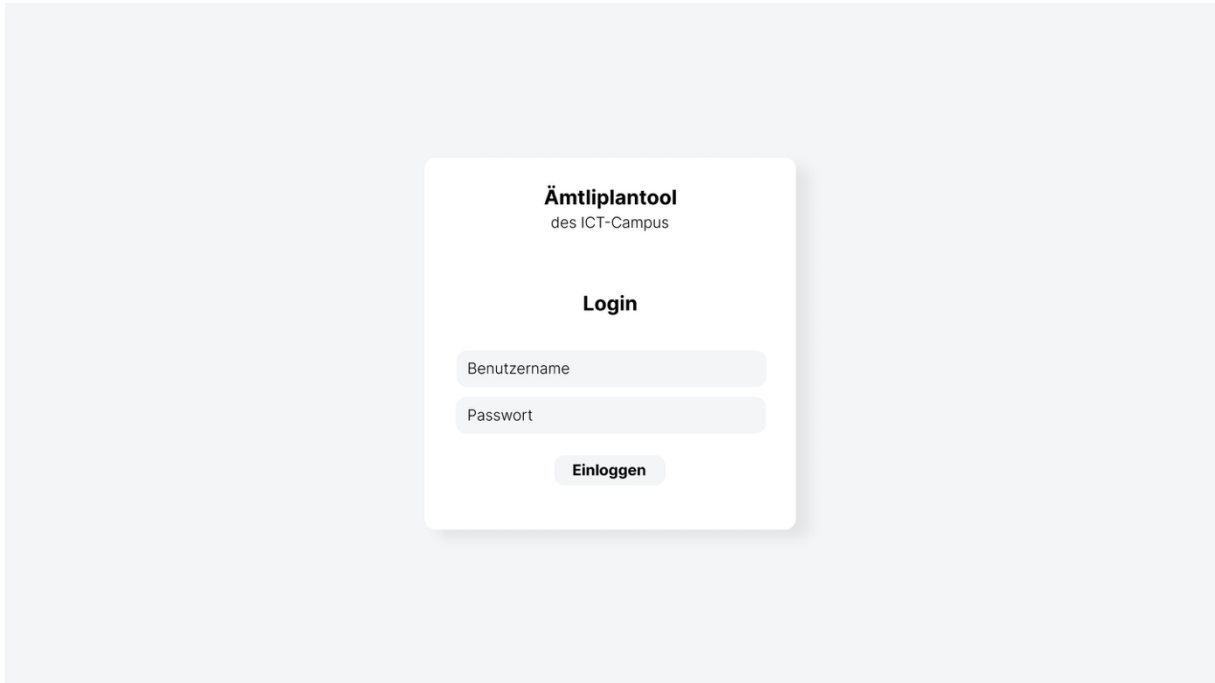
ID	Anforderung	Priorität
NF-01	Desktop-Optimierung: Die Benutzeroberfläche ist primär für Browser auf Desktop-Geräten optimiert.	Muss
NF-02	Benutzerfreundlichkeit: Die Zuweisung von Personen soll mit möglichst wenigen Klicks intuitiv durchführbar sein.	Soll
NF-03	Sprachkonvention: Codekommentare, Datenbanktabellen, Attribute und technische Dokumentation sind in englischer Sprache verfasst.	Soll
NF-04	Dokumentation: Das Projekt ist mit Techstack und deren Versionsnummern und soweit möglich mit allen wichtigen Informationen dokumentiert, sodass das Ämtliplantool von Lernenden zukünftiger Jahrgänge oder von Coaches gewartet werden kann.	Soll
NF-05	Datensicherheit: Lernende können keine Planungsdaten anderer Teams manipulieren und haben nur die in der Benutzermatrix festgelegten Leserechte.	Muss
NF-06	Wartbarkeit: Der Code ist durchgehend kommentiert, um eine Übergabe an den nächsten Jahrgang zu ermöglichen.	Muss
NF-07	Single Sign-On (SSO): Integration bestehender Authentifizierungsprozesse (z. B. Campboard/AD).	Kann

4 Use Case Diagramm

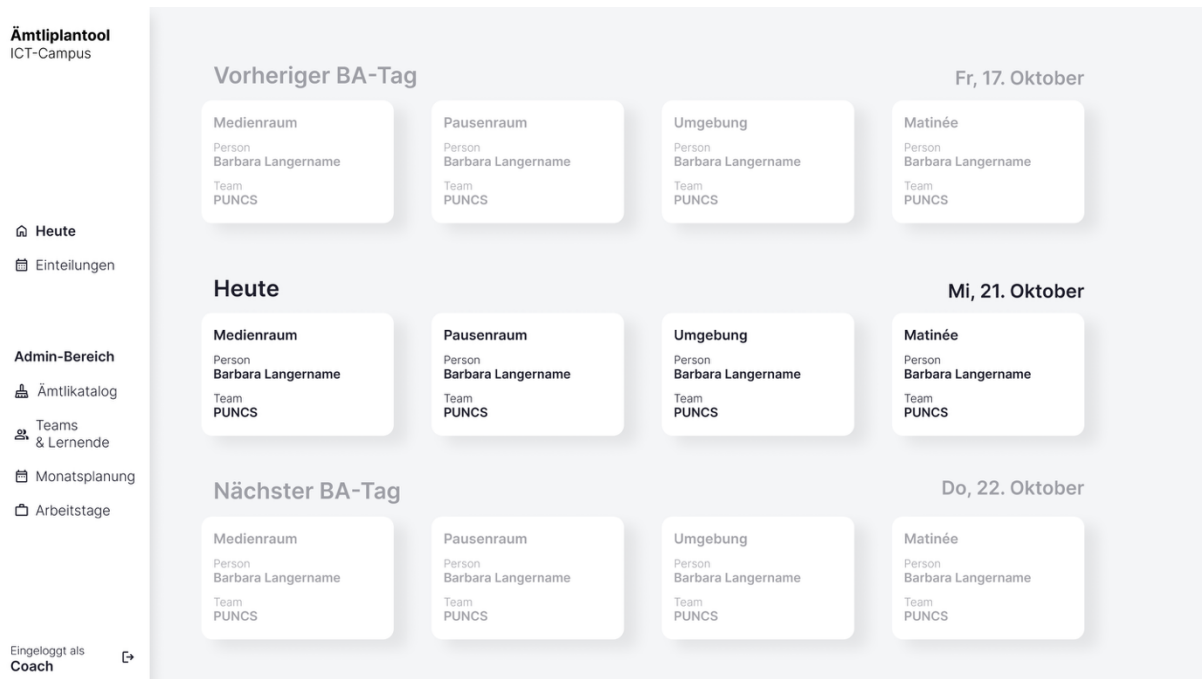


5 Mockups

5.1 Login



5.2 Coach-Login



Ämtliplantool
ICT-Campus

< Oktober 2025 >

🏠 Heute

📅 Einteilungen

Admin-Bereich

📖 Ämtlikatalog

👥 Teams
& Lernende

📅 Monatsplanung

📅 Arbeitstage

Eingeloggt als
Coach

Team	Medienraum PUNCS	Pausenraum Beeliverys	Umgebung Beeliverys	Matinée Plapplis
Mi 1.10.	Hansi Hase	Hansi Hase	Hansi Hase	Hansi Hase
Do 2.10.	Moni Thor	Moni Thor	Moni Thor	Moni Thor
Fr 3.10.	Mike Shiva	Mike Shiva	Mike Shiva	Mike Shiva
Mo 6.10.	Hansi Hase	Hansi Hase	Hansi Hase	Hansi Hase
Di 7.10.	Moni Thor	Moni Thor	Moni Thor	kein Matinée
Mi 8.10.	Mike Shiva	Mike Shiva	Mike Shiva	Mike Shiva
Do 9.10.	Sonnen Strahl	Sonnen Strahl	Sonnen Strahl	Sonnen Strahl
Fr 10.10.	Tux Pinguin	Tux Pinguin	Tux Pinguin	kein Matinée
Mi 15.10.	Maria Mantel	Maria Mantel	Maria Mantel	Maria Mantel
Do 16.10.	Brown Bruce	Brown Bruce	Brown Bruce	Brown Bruce
Fr 17.10.	Reto Folke	Reto Folke	Reto Folke	Reto Folke
Mi 22.10.	Maria Mantel	Maria Mantel	Maria Mantel	Maria Mantel
Do 23.10.	Brown Bruce	Brown Bruce	Brown Bruce	Brown Bruce
Fr 24.10.	Reto Folke	Reto Folke	Reto Folke	Reto Folke
Mi 29.10.	Maria Mantel	Maria Mantel	Maria Mantel	Maria Mantel
Do 30.10.	Brown Bruce	Brown Bruce	Brown Bruce	Brown Bruce
Fr 31.10.	Reto Folke	Reto Folke	Reto Folke	Reto Folke

Ämtliplantool
ICT-Campus

Ämtlikategorien

+ Neue Ämtlikategorie

Noch keine Ämtlikategorie vorhanden

🏠 Heute

📅 Einteilungen

Admin-Bereich

📖 Ämtlikatalog

👥 Teams
& Lernende

📅 Monatsplanung

📅 Arbeitstage

Eingeloggt als
Coach

Ämtliplantool
ICT-Campus

🏠 Heute

📅 Einteilungen

Admin-Bereich

📖 Ämtlikatalog

👥 Teams & Lernende

📅 Monatsplanung

📅 Arbeitstage

Eingeloggt als
Coach

Ämtlikategorien

+ Neue Ämtlikategorie

Medienraum

Beschreibung

Am Morgen bei Arbeitsbeginn bis 8:15 Uhr: Fenster öffnen / kippen, ggf. Storen hoch oder runter, Stühle richten.

Vor Mittagspause: Fenster öffnen / Lüften, Stühle / Tische reinigen

Vor Arbeitsende: Kurz lüften, Stühle geraderücken / versetzte Sitzordnung, Ordnung schaffen, Stühle / Tische reinigen.

Freitag: Tische und Stühle zusammenstellen, Boden möglichst frei machen.

Fenster schliessen.

Pausenraum

Beschreibung

Nach der „z'Nüni“-Pause: Tische abputzen, Geschirr in Geschirrspüler räumen, Ordnung schaffen. Bänke & Tische reinigen

Nach der Mittagspause bis 13:00 Uhr: Tische abputzen, Geschirrspüler ausräumen, Geschirr in Geschirrspüler räumen, Ordnung schaffen. Mikrowellen reinigen!

Vor Arbeitsende bis 16:00 Uhr: Tische abputzen, Geschirrspüler, Geschirr in Geschirrspüler räumen, Ordnung schaffen Bänke & Tische reinigen und unter Tische schieben und Fenster schliessen.

Umgebung

Beschreibung

Am Morgen bei Arbeitsbeginn: Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Räume Lüften.

Vor Arbeitsende: Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Fözele* Fenster kontrollieren und schliessen. Wenn nötig: Gang & Treppen staubsaugen

Matinée

Beschreibung

Am Morgen vor der 1. Präsentation ca. 08:15 Uhr: Zusammenfassung der Präsentationen, Aufträge und Lösungen sowie aufgetauchter Fragen des Vortages in Absprache mit dem Modul-Lead-Coach.

Teilnahme freiwillig. PIBS: Durchführung muss organisiert werden.

Ämtliplantool
ICT-Campus

🏠 Heute

📅 Einteilungen

Admin-Bereich

📖 Ämtlikatalog

👥 Teams & Lernende

📅 Monatsplanung

📅 Arbeitstage

Eingeloggt als
Coach

Freitag: Tische und Stühle zusammenstellen, Boden möglichst frei machen.

Fenster schliessen.

Nach der „z'Nüni“-Pause: Tische abputzen, Geschirr in Geschirrspüler räumen, Ordnung schaffen Bänke & Tische reinigen und unter Tische schieben und Fenster schliessen.

Umgebung

Beschreibung

Am Morgen bei Arbeitsbeginn: Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Räume Lüften.

Vor Arbeitsende: Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Fözele* Fenster kontrollieren und schliessen. Wenn nötig: Gang & Treppen staubsaugen

Matinée

Beschreibung

Am Morgen vor der 1. Präsentation ca. 08:15 Uhr: Zusammenfassung der Präsentationen, Aufträge und Lösungen sowie aufgetauchter Fragen des Vortages in Absprache mit dem Modul-Lead-Coach.

Teilnahme freiwillig. PIBS: Durchführung muss organisiert werden.

Labor aufräumen

Beschreibung

Dies ist ein Beispiel für ein inaktives Ämtli.

Coaches können ein Ämtli auf inaktiv setzen, wenn es z.B. während eines BA-Jahrganges nicht gebraucht wird.

So müssen sie es nicht löschen und im nächsten Jahrgang neu erfassen, sondern es bleibt einfach hier gespeichert und kann für einen nächsten BA-Jahrgang reaktiviert werden.

Ämtliplantool
ICT-Campus

- 🏠 Heute
- 📅 Einteilungen
- Admin-Bereich**
- 📖 Ämtlikatalog
- Teams & Lernende**
- 📅 Monatsplanung
- 📅 Arbeitstage

Eingeloggt als
Coach

Teams & Lernende

Noch kein Team vorhanden

+ Neues Team anlegen

Ämtliplantool
ICT-Campus

- 🏠 Heute
- 📅 Einteilungen
- Admin-Bereich**
- 📖 Ämtlikatalog
- Teams & Lernende**
- 📅 Monatsplanung
- 📅 Arbeitstage

Eingeloggt als
Coach

Teams & Lernende

Plapplis ✎ 🗑

- | | | | |
|------------------------------------|--------------------------------|---------------------------------|------------------------------|
| Maria Nachnametta ✎ 🗑 | Lampen Schirm ✎ 🗑 | Glücks Strähne ✎ 🗑 | Otto Walkes ✎ 🗑 |
| Rhabarber Barbara ✎ 🗑 | Max Muster ✎ 🗑 | Mario Küblis ✎ 🗑 | Peter Pan ✎ 🗑 |
| Jakob Bleuer ✎ 🗑 | Lernende*r hinzufügen + | | |

PUNCS ✎ 🗑

- | | | | |
|-----------------------------|--------------------------------|--------------------------------|-----------------------------|
| Hansi Hase ✎ 🗑 | Anna Moser ✎ 🗑 | Sonnen Strahl ✎ 🗑 | Noah Arche ✎ 🗑 |
| Bruno Mars ✎ 🗑 | Lernende*r hinzufügen + | | |

+ Neues Team anlegen

Ämtliplantool
ICT-Campus

- 🏠 Heute
- 📅 Einteilungen

- Admin-Bereich
- 📖 Ämtlikatalog
- 👥 Teams & Lernende
- 📅 Monatsplanung
- 📅 Arbeitstage

Eingeloggt als **Coach**

Monatsplanung

PUNCS
Beeliverys
Plapplis

	August	September	Oktober	November	Dezember	Januar
Medienraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Pausenraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Umgebung	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Matinée	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL

	Februar	März	April	Mai	Juni	Juli
Medienraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Pausenraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Umgebung	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Matinée	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL

Ämtliplantool
ICT-Campus

- 🏠 Heute
- 📅 Einteilungen

- Admin-Bereich
- 📖 Ämtlikatalog
- 👥 Teams & Lernende
- 📅 Monatsplanung
- 📅 Arbeitstage

Eingeloggt als **Coach**

Arbeitstage

BA 2025-2026

August							September							Oktober							November											
kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	
31					01	02	03	38	01	02	03	04	05	06	07	40			01	02	03	04	05	44							01	02
32	04	05	06	07	08	09	10	37	08	09	10	11	12	13	14	41	06	07	08	09	10	11	12	45	03	04	05	06	07	08	09	
33	11	12	13	14	15	16	17	38	15	16	17	18	19	20	21	42	13	14	15	16	17	18	19	46	10	11	12	13	14	15	16	
34	18	19	20	21	22	23	24	39	22	23	24	25	26	27	28	43	20	21	22	23	24	25	26	47	17	18	19	20	21	22	23	
35	25	26	27	28	29	30	31	40	29	30						44	27	28	29	30	31			48	24	25	26	27	28	29	30	

Dezember							Januar							Februar							März										
kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So
49	01	02	03	04	05	06	07	1			01	02	03	04	5							01	9							01	
50	08	09	10	11	12	13	14	2	05	06	07	08	09	10	11	6	02	03	04	05	06	07	08	10	02	03	04	05	06	07	08
51	15	16	17	18	19	20	21	3	12	13	14	15	16	17	18	7	09	10	11	12	13	14	15	11	09	10	11	12	13	14	15
52	22	23	24	25	26	27	28	4	19	20	21	22	23	24	25	8	16	17	18	19	20	21	22	12	16	17	18	19	20	21	22
1	29	30	31					5	26	27	28	29	30	31	9	23	24	25	26	27	28	13	23	24	25	26	27	28	29		

April							Mai							Juni							Juli										
kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So	kw	Mo	Di	Mi	Do	Fr	Sa	So
14			01	02	03	04	05	18				01	02	03	23	01	02	03	04	05	06	07	27			01	02	03	04	05	
15	06	07	08	09	10	11	12	19	04	05	06	07	08	09	10	24	08	09	10	11	12	13	14	28	06	07	08	09	10	11	12
16	13	14	15	16	17	18	19	20	11	12	13	14	15	16	17	25	15	16	17	18	19	20	21	29	13	14	15	16	17	18	19
17	20	21	22	23	24	25	26	21	18	19	20	21	22	23	24	26	22	23	24	25	26	27	28	30	20	21	22	23	24	25	26
18	27	28	29	30				22	25	26	27	28	29	30	31	27	29	30						31	27	28	29	30	31		

5.3 Teamverantwortlich-Login

Ämtliplantool
ICT-Campus

- Einteilungen
- Jahresübersicht
- Ämtlikategorien

Eingeloggt als
Teamchef*in
Beeliverys

< Oktober 2025 >

Team	Pausenraum Beeliverys	Umgebung Beeliverys
Mi 1.10.	Hansi Hase	Hansi Hase
Do 2.10.	Moni Thor	Moni Thor
Fr 3.10.	Mike Shiva	Mike Shiva
Mo 6.10.	Hansi Hase	Hansi Hase
Di 7.10.	Moni Thor	Moni Thor
Mi 8.10.	Mike Shiva	Mike Shiva
Do 9.10.	Sonnen Strahl	Sonnen Strahl
Fr 10.10.	Tux Pinguin	Tux Pinguin
Mi 15.10.	Maria Mantel	Noch nicht eingeteilt.
Do 16.10.	Brown Bruce	Noch nicht eingeteilt.
Fr 17.10.	Reto Folke	Noch nicht eingeteilt.
Mi 22.10.	Noch nicht eingeteilt.	Noch nicht eingeteilt.
Do 23.10.	Brown Bruce	Noch nicht eingeteilt.
Fr 24.10.	Reto Folke	Noch nicht eingeteilt.
Mi 29.10.	Noch nicht eingeteilt.	Noch nicht eingeteilt.
Do 30.10.	Brown Bruce	Noch nicht eingeteilt.
Fr 31.10.	Noch nicht eingeteilt.	Noch nicht eingeteilt.

Ämtliplantool
ICT-Campus

- Einteilungen
- Jahresübersicht
- Ämtlikategorien

Eingeloggt als
Teamchef*in
Beeliverys

Ämtlikategorien

Medienraum

Beschreibung

Am Morgen bei Arbeitsbeginn bis 8:15 Uhr: Fenster öffnen / kippen, ggf. Storen hoch oder runter, Stühle richten.

Vor Mittagspause: Fenster öffnen / Lüften, Stühle / Tische reinigen

Vor Arbeitsende: Kurz lüften, Stühle geraderücken / versetzte Sitzordnung, Ordnung schaffen, Stühle / Tische reinigen.

Freitag: Tische und Stühle zusammenstellen, Boden möglichst frei machen.

Fenster schliessen.

Pausenraum

Beschreibung

Nach der „z“Nüni“-Pause: Tische abputzen, Geschirr in Geschirrspüler räumen, Ordnung schaffen. Bänke & Tische reinigen

Nach der Mittagspause bis 13:00 Uhr:
Tische abputzen, Geschirrspüler ausräumen, Geschirr in Geschirrspüler räumen, Ordnung schaffen. Mikrowellen reinigen!

Vor Arbeitsende bis 16:00 Uhr:
Tische abputzen, Geschirrspüler, Geschirr in Geschirrspüler räumen, Ordnung schaffen Bänke & Tische reinigen und unter Tische schieben und Fenster schliessen.

Umgebung

Beschreibung

Am Morgen bei Arbeitsbeginn:
Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Räume Lüften.

Vor Arbeitsende:
Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Fözele Fenster kontrollieren und schliessen. Wenn nötig: Gang & Treppen staubsaugen

Matinée

Beschreibung

Am Morgen vor der 1. Präsentation ca. 08:15 Uhr:
Zusammenfassung der Präsentationen, Aufträge und Lösungen sowie aufgetauchter Fragen des Vortages in Absprache mit dem Modul-Lead-Coach.

Teilnahme freiwillig. PIBS: Durchführung muss organisiert werden.

Ämtliplantool
ICT-Campus

Jahresübersicht

-  Einteilungen
-  Jahresübersicht
-  Ämtlikategorien

	August	September	Oktober	November	Dezember	Januar
Medienraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Pausenraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Umgebung	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Matinée	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL

	Februar	März	April	Mai	Juni	Juli
Medienraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Pausenraum	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Umgebung	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL
Matinée	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL	PU BE PL

Eingeloggt als
Teamchef*in
Beeliverys 

5.4 Lernenden-Login

Ämtliplantool
ICT-Campus

- Einteilungen
- Jahresübersicht
- Ämtlikategorien

Eingeloggt als
Lernende*r
Beeliverys

< Oktober 2025

Team	Pausenraum Beeliverys	Umgebung Beeliverys
Mi 1.10.	Hansi Hase	Hansi Hase
Do 2.10.	Moni Thor	Moni Thor
Fr 3.10.	Mike Shiva	Mike Shiva
Mo 6.10.	Hansi Hase	Hansi Hase
Di 7.10.	Moni Thor	Moni Thor
Mi 8.10.	Mike Shiva	Mike Shiva
Do 9.10.	Sonnen Strahl	Sonnen Strahl
Fr 10.10.	Tux Pinguin	Tux Pinguin
Mi 15.10.	Maria Mantel	Noch nicht eingeteilt.
Do 16.10.	Brown Bruce	Noch nicht eingeteilt.
Fr 17.10.	Reto Folke	Noch nicht eingeteilt.
Mi 22.10.	Noch nicht eingeteilt.	Noch nicht eingeteilt.
Do 23.10.	Brown Bruce	Noch nicht eingeteilt.
Fr 24.10.	Reto Folke	Noch nicht eingeteilt.
Mi 29.10.	Noch nicht eingeteilt.	Noch nicht eingeteilt.
Do 30.10.	Brown Bruce	Noch nicht eingeteilt.
Fr 31.10.	Noch nicht eingeteilt.	Noch nicht eingeteilt.

Ämtliplantool
ICT-Campus

- Einteilungen
- Jahresübersicht
- Ämtlikategorien

Eingeloggt als
Lernende*r
Beeliverys

Ämtlikategorien

Medienraum

Beschreibung

Am Morgen bei Arbeitsbeginn bis 8:15 Uhr: Fenster öffnen / kippen, ggf. Storen hoch oder runter, Stühle richten.

Vor Mittagspause: Fenster öffnen / Lüften, Stühle / Tische reinigen

Vor Arbeitsende: Kurz lüften, Stühle geraderücken / versetzte Sitzordnung, Ordnung schaffen, Stühle / Tische reinigen.

Freitag: Tische und Stühle zusammenstellen, Boden möglichst frei machen.

Fenster schliessen.

Pausenraum

Beschreibung

Nach der „z“Nüni“-Pause: Tische abputzen, Geschirr in Geschirrspüler räumen, Ordnung schaffen. Bänke & Tische reinigen

Nach der Mittagspause bis 13:00 Uhr: Tische abputzen, Geschirrspüler ausräumen, Geschirr in Geschirrspüler räumen, Ordnung schaffen. Mikrowellen reinigen!

Vor Arbeitsende bis 16:00 Uhr: Tische abputzen, Geschirrspüler, Geschirr in Geschirrspüler räumen, Ordnung schaffen Bänke & Tische reinigen und unter Tische schieben und Fenster schliessen.

Umgebung

Beschreibung

Am Morgen bei Arbeitsbeginn: Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Räume Lüften.

Vor Arbeitsende: Eingangsbereich und bei Lounge ggf. Müll entsorgen und Ordnung schaffen, Fözele Fenster kontrollieren und schliessen. Wenn nötig: Gang & Treppen staubsaugen

Matinée

Beschreibung

Am Morgen vor der 1. Präsentation ca. 08:15 Uhr: Zusammenfassung der Präsentationen, Aufträge und Lösungen sowie aufgetauchter Fragen des Vortages in Absprache mit dem Modul-Lead-Coach.

Teilnahme freiwillig. PIBS: Durchführung muss organisiert werden.

Ämtliplantool
ICT-Campus

- ☰ Einteilungen
- ☰ Jahresübersicht
- ☰ Ämtlikategorien

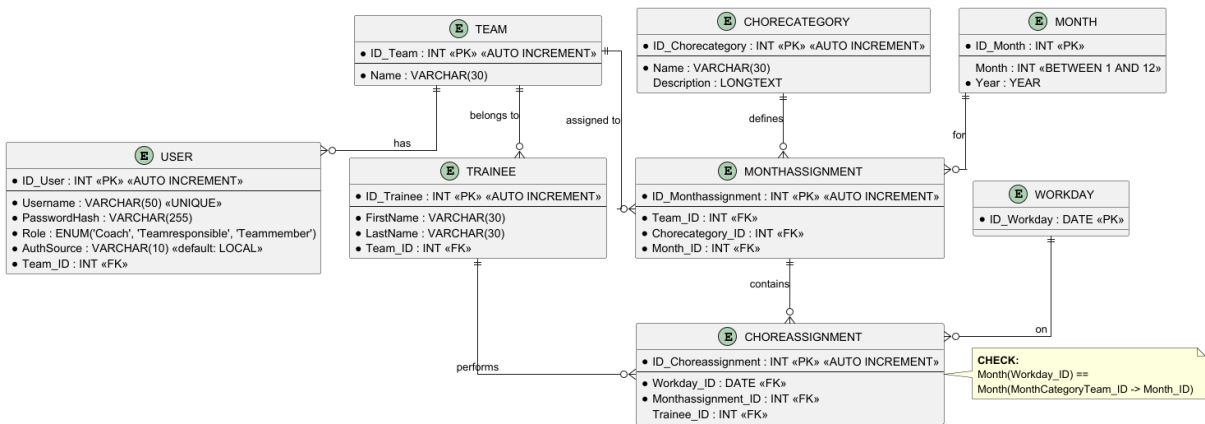
Eingeloggt als
Lernende*r
Beeliverys



6 Datenbank

Für dieses Projekt ist eine MySQL Datenbank vorgesehen.

6.1 ERD



7 Benutzer, Rollen und Rechte

7.1 Logins

Das hier vorgelegte Projekt zielt auf ein MVP (minimal viable product) hin. Entsprechend muss die Komplexität der Benutzer (Zugänge bzw. Logins für Personen) so weit wie möglich reduziert werden. Die folgenden Logins sind notwendig.

Notwendige Logins

Login-Nummer	Wer?	Beschreibung
A	Coaches	Alle Coaches nutzen diesen Account gemeinsam .
B1 bis Bn	Teamverantwortliche*r des Team n	Je eine Person (der*die Ämtliverantwortliche) pro Team hat einen solchen Account für sich allein .
C1 bis Cn	Ein Login für alle Lernenden eines Teams n	Alle Lernenden eines Teams nutzen diesen Account gemeinsam . Ein solcher Account pro Team.

7.2 Rollen

Wir Projektmitglieder sind im ersten Jahr der Informatikausbildung. Entsprechend wenig Kenntnisse haben wir von Backend- und Frontendentwicklung. Wir erachten es als wichtig, den Horizont zu sehen, ehe wir uns in den Code stürzen, damit wir nicht in eine Sackgasse fahren.

Das Ämtliplantool braucht Rollen und Authentifizierung. Ein komplexes Thema. Nachfolgend versuchen wir, einen Rahmen aufzuspannen, in dem wir das Tool entwickeln können, ohne bereits Auth-Expert*innen zu sein.

Dazu ist es wichtig, zuerst klare Rollen und *Rechte* zu definieren. Damit können wir bei Beginn der Backendentwicklung bereits mit den verschiedenen *Rechten* arbeiten, unabhängig davon, wie sich Benutzer später authentifizieren. Wir wissen erst, *was* gemacht werden darf und was nicht. Nicht aber, ob der Nutzer authentifiziert ist. Die Authentifizierung können wir zu einem späteren Zeitpunkt implementieren, sofern das Backend eine klare Rollenstruktur befolgt (Business-Logic).

Die Authentifizierung ist dann die eine Identitätsprüfung. Dadurch wird festgestellt, ob Nutzer A wirklich Person A ist. Sobald die Identität eines Nutzers festgestellt ist, erfolgt die weitere Verarbeitung ausschliesslich über dessen *Rechte*. Und eben diese *Rechte* können wir heute bereits bei der Entwicklung berücksichtigen, obwohl wir das Thema Authentifizierung aktuell noch nicht beherrschen. Später könnten wir für die Authentifizierung das Active Directory unseres Campus verwenden.

Grundsätzlich muss zwischen Rollen und Rechten (= Berechtigungen) unterschieden werden. Gemäss Benutzermatrix sind die Rollen im Ämtliplantool «Coach», «Teamverantwortlich», «Lernende*r». Mit «CRUD» beziehen wir uns auf Create, Read, Update, Delete – Erstellen, Lesen, Ändern, Löschen. Nachfolgend beschreiben wir zuerst die Rollen.

7.2.1 Rollen

7.2.1.1 Rolle «Coach»

Dabei nutzen alle Coaches im MVP dasselbe Login, sie teilen sich also einen Zugang und nutzen dasselbe Passwort. Dies, um die Komplexität tief zu halten im Rahmen des MVP. Das Coachlogin darf den Ämtlikatalog verwalten (CRUD), die Team-Monats-Zuteilung verwalten (CRUD), Teams verwalten (CRUD) und Lernende verwalten (CRUD). Das Coachlogin darf keine Tageszuteilungen (Am Tag X macht Lernende Y das Ämtli Z) vornehmen. Auf die Tageszuteilungen hat das Coachlogin nur read. Dies aber für alle Tageszuteilungen (alle Teams).

7.2.1.2 Rolle «Teamverantwortlich»

Jedes Team hat eine teamverantwortliche Person «Teamverantwortlich», in der aktuellen BA wären das bei drei Teams also drei Teamverantwortliche, die je ihr eigenes, persönliches Login brauchen. Die Teamverantwortlichen haben Read auf den Ämtlikatalog, die Team-Monats-Zuteilung (alle Teams) und auf die Teamnamen. Sie sehen, welche Lernende in ihrem eigenen Team sind, haben jedoch keine Rechte auf die Teamzugehörigkeiten anderer Teams. Teamverantwortliche haben CRUD auf die Tageszuteilungen des eigenen Teams und keine Rechte auf Tageszuteilungen der anderen Teams.

7.2.1.3 Rolle «Lernende*r»

Die Lernenden nutzen im MVP ein gemeinsames Login, allerdings wieder pro Team. Bei drei Teams brauchen wir also nochmal drei Logins für Lernende, eines pro Team. Das Lernendenlogin eines Teams hat Read auf den Ämtlikatalog, die Team-Monats-Zuteilung, die Teamnamen, die Mitglieder des eigenen Teams (nur des eigenen Teams) und auf die Tageszuteilungen des eigenen Teams. Keine Rechte auf Daten anderer Teams, bis auf den Teamnamen. Also eigentlich dieselben Rechte wie das Teamverantwortlichlogin, einfach ohne CUD.

7.3 Rechte (Permissions)

Aus diesen Überlegungen, die die Anforderungen abbilden, haben wir die Rechte entwickelt und jedem Recht einen Namen vergeben. Die folgende Benutzermatrix fasst dies nochmal zusammen.

7.3.1 Struktur der Berechtigungen

Jede Berechtigung (Authority) im System ist nach dem Schema `ressource:aktion:bereich` aufgebaut:

1. **Ressource (Was?):** Das Objekt des Zugriffs (z. B. `trainee`, `chore-category`, `day-assignment`).
2. **Aktion (Wie?):** Die Art des Zugriffs. Wir unterscheiden strikt zwischen `read` (Lesen) und `write` (Erstellen, Ändern, Löschen).
3. **Bereich/Scope (Wo?):** * `:all`: Erlaubt den Zugriff auf alle Datensätze systemweit (unabhängig vom Team).
 - `:team`: Beschränkt den Zugriff auf Datensätze, die dem **eigenen Team** des Benutzers zugeordnet sind.

7.3.2 Beispielhafte Anwendung

Dieses System erlaubt es uns, komplexe Anforderungen einfach abzubilden:

- **Datenschutz:** Ein Lernender erhält das Recht `trainee:read:team`. Er kann somit alle Teamkollegen sehen, hat aber keinen Zugriff auf die Daten von Lernenden in anderen Teams.
- **Administrative Kontrolle:** Ein Coach erhält `trainee:read:all` und `trainee:write:all`, um die Stammdaten aller Teilnehmer über alle Teams hinweg zu pflegen.
- **Operative Planung:** Teamverantwortliche besitzen `day-assignment:write:team`. Sie können die Ämtli für ihr eigenes Team einteilen, aber die Einteilungen anderer Teams nicht manipulieren.

7.3.3 Technische Vorteile

- **Transparenz:** Im Programmcode (Controller) ist durch Annotationen wie `@PreAuthorize("hasAuthority('trainee:read:team')")` sofort ersichtlich, welche Sicherheitsregeln gelten.
- **Wartbarkeit:** Neue Rollen können durch einfaches Zuweisen bestehender "Scopes" erstellt werden, ohne den Java-Code anpassen zu müssen.
- **Eindeutigkeit:** Die Benennung `:team` statt `:own` wurde bewusst gewählt, da im Ämtliplantool die Zusammenarbeit innerhalb der Gruppe (Team) und nicht das Individuum im Zentrum steht.

Berechtigungen		Rollen			PERMISSION NAME
		A (Coaches)	Bn (Verantwortliche*r)	Cn (Lernende)	
Stammdaten Lernende: Vor- und Nachname, Teamzugehörigkeit					
	CUD	X			<code>trainee:write:all</code>
	Read all	X			<code>trainee:read:all</code>
	Read, aber nur im eigenen Team		X	X	<code>trainee:read:team</code>
Stammdaten Teams: Teamname					
	CUD	X			<code>team:write:all</code>
	Read all	X	X	X	<code>team:read:all</code>
Ämtlikategorie (Name, aktiv/inaktiv, ggf. Beschreibung)					
	CUD	X			<code>chore-category:write:all</code>

Read all	X	X	X	chore-category:read:all
Monatsplanung (Team-Ämtli-Monat)				
CUD	X			month-assignment:write:all
Read all	X	X	X	month-assignment:read:all
Tagesplanung (Lernende*r-Ämtli-Tag)				
CUD (nur eigenes Team)		X		day-assignment:write:team
Read (nur eigenes Team)		X	X	day-assignment:read:team
Read (alle Teams)	X			day-assignment:read:all

8 API Contracts (Endpoint-Design, Verträge)

Nachfolgend definieren wir die Datenschnittstellen. Für jeden Use Case des Use Case Diagramms ist eine oder mehrere Schnittstellen (API-Endpoints) nötig.

8.1 Stammdaten (Jonas)

8.1.1 Ämtlikategorien

Endpoint	GET /chorecategories
Beschreibung	Alle bestehenden Ämtli lesen
Use Case	Ämtlikategorien sehen
Permission	chorecategory:read:all
Scope	global
Request Body	-
Response	Liste von Ämtliobjekten, jeweils: idChorecategory int not null name String not null description String nullable
Status Codes to be expected	Erfolg: 200 OK (Wenn keine Ämtli vorhanden sind, wird eine leere Liste [] geschickt.)

Endpoint	GET /chorecategories/{id}
Beschreibung	Die Infos eines bestehenden Ämtlis im Ämtlikatalog lesen
Use Case	Ämtlikategorien sehen
Permission	chorecategory:read:all
Scope	global
Request Body	-
Response	idChorecategory int not null name String not null description String nullable
Status Codes to be expected	Erfolg: 200 OK Fehler: 400 Bad Request Fehler: 404 Not Found (es besteht kein Ämtli mit der angegebenen ID)

Endpoint	POST /chorecategories
Beschreibung	Ein neues Ämtli im Ämtlikatalog anlegen
Use Case	Ämtlikategorien bearbeiten
Permission	chorecategory:write:all
Scope	global
Request	name String not null, max. 30 Zeichen description String nullable
Response	idChorecategory int name String not null description String nullable
Status Codes to be expected	Erfolg: 201 Created Fehler: 400 Bad Request Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 409 Conflict (es besteht bereits ein Ämtli mit diesem Namen)

Endpoint	PATCH /chorecategories/{id}
Beschreibung	Die Infos eines bestehenden Ämtlis im Ämtlikatalog bearbeiten
Use Case	Ämtlikategorien bearbeiten
Permission	chorecategory:write:all
Scope	global
Request Body	Keine id, da diese bereits in der Pfadvariable ist name String not null, max. 30 Zeichen description String nullable (Alle Felder optional, mindestens 1 Feld muss aber geschickt werden)
Response	idChorecategory int not null name String not null description String nullable
Status Codes to be expected	Erfolg: 200 OK Fehler: 400 Bad Request (z.B. leerer Name oder ungültiges JSON empfangen) Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht kein Ämtli mit der angegebenen ID) Fehler: 409 Conflict (es besteht bereits ein Ämtli mit diesem Namen)

Endpoint	DELETE /chorecategories/{id}
Beschreibung	Ein bestehendes Ämtli vom Ämtlikatalog komplett und dauerhaft löschen
Use Case	Ämtlikategorien bearbeiten
Permission	chorecategory:write:all
Scope	global
Request Body	-
Response	-
Status Codes to be expected	Erfolg: 204 No Content Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 409 Conflict (z.B. ein Fremdschlüssel zeigt noch auf dieses Ämtli) Fehler: 404 Not Found (es besteht kein Ämtli mit der angegebenen ID)

8.1.2 Teams

Endpoint	GET /teams
Beschreibung	Alle bestehenden Teams lesen
Use Case	Teamnamen sehen
Permission	team:read:all
Scope	global
Request Body	-
Response	Liste von Teamobjekten, jeweils: idTeam int not null name String not null
Status Codes to be expected	Erfolg: 200 OK (Wenn keine Teams vorhanden sind, wird eine leere Liste [] geschickt.)

Endpoint	GET /teams/{id}
Beschreibung	Ein spezifisches Team lesen
Use Case	Teamnamen sehen
Permission	team:read:all
Scope	global
Request Body	-
Response	idTeam int not null name String not null
Status Codes to be expected	Erfolg: 200 OK Fehler: 404 Not Found (es besteht kein Ämtli mit der angegebenen ID)

Endpoint	POST /teams
Beschreibung	Ein neues Team anlegen
Use Case	Teams bearbeiten
Permission	team:write:all
Scope	global
Request	name String not null, max. 30 Zeichen
Response	idTeam int name String not null
Status Codes to be expected	Erfolg: 201 Created Fehler: 400 Bad Request Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 409 Conflict (es besteht bereits ein Team mit diesem Namen)

Endpoint	PATCH /teams/{id}
Beschreibung	Die Infos eines bestehenden Teams bearbeiten
Use Case	Teams bearbeiten
Permission	team:write:all
Scope	global
Request Body	Keine id, da diese bereits in der Pfadvariable ist name String not null, max. 30 Zeichen
Response	idTeam int not null name String not null
Status Codes to be expected	Erfolg: 200 OK Fehler: 400 Bad Request (z.B. leerer Name oder ungültiges JSON empfangen) Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht kein Team mit der angegebenen ID) Fehler: 409 Conflict (es besteht bereits ein Team mit diesem Namen)

Endpoint	DELETE /teams/{id}
Beschreibung	Ein bestehendes Team komplett und dauerhaft löschen
Use Case	Teams bearbeiten
Permission	team:write:all
Scope	global
Request Body	-
Response	-
Status Codes to be expected	Erfolg: 204 No Content Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht kein Team mit der angegebenen ID) Fehler: 409 Conflict (z.B. ein Fremdschlüssel zeigt noch auf dieses Team)

8.1.3 Lernende

Endpoint	GET /trainees/{id}
Beschreibung	Ein*e spezifische*r Lernende*r lesen (es wird nach der ID gesucht)
Use Case	Lernende sehen
Permission	trainee:read:all or trainee:read:team
Scope	trainee:read:all: global trainee:read:team: own team only
Request Body	-
Response	<p>caller with permission: trainee:read:all May request a trainee of any team</p> <p>caller with permission: trainee:read:team May only request a trainee in the same team as the caller</p> <p>Structure of Trainee-Data idTrainee int not null firstName String not null lastName String not null teamId int not null</p>
Status Codes to be expected	<p>Erfolg: 200 OK</p> <p>Fehler: 400 Bad Request (id <= 0)</p> <p>Fehler: 401 Unauthorized falls der Aufrufer nicht eingeloggt ist</p> <p>Fehler: 403 Forbidden, wenn ein User zwar eingeloggt ist, aber keines der genannten Rechte besitzt</p> <p>Fehler: 404 Not Found, wenn es keinen Lernenden mit dieser ID gibt</p>

Endpoint	GET /trainees?teamId={id}
Beschreibung	Lernende eines spezifischen Teams lesen (es wird nach Team-ID gesucht)
Use Case	Lernende sehen
Permission	trainee:read:all or trainee:read:team
Scope	trainee:read:all: global trainee:read:team: own team only
Request Body	-
Response	<p>Returns a list of all trainees in the requested team</p> <p>caller with permission: trainee:read:all May request trainees of any team</p> <p>caller with permission: trainee:read:team May only request a trainee in the same team as the caller</p> <p>Structure of Trainee-Data idTrainee int not null firstName String not null lastName String not null teamId int not null</p>
Status Codes to be expected	<p>Erfolg: 200 OK</p> <p>Fehler: 400 Bad Request (id <= 0)</p> <p>Fehler: 401 Unauthorized falls der Aufrufer nicht eingeloggt ist</p> <p>Fehler: 403 Forbidden, wenn ein User zwar eingeloggt ist, aber keines der genannten Rechte besitzt</p> <p>Fehler: 404 Not Found, wenn es kein Team mit dieser ID gibt</p>

Endpoint	GET /trainees
Beschreibung	Alle Lernenden lesen
Use Case	Lernende sehen
Permission	trainee:read:all or trainee:read:team
Scope	trainee:read:all: global trainee:read:team: own team only
Request Body	-

Response	<p>caller with permission: trainee:read:all List of all trainees (from all teams)</p> <p>caller with permission: trainee:read:team Response contains all trainees of the callers team only</p> <p>Structure of Trainee-Data idTrainee int not null firstName String not null lastName String not null teamId int not null</p>
Status Codes to be expected	<p>Erfolg: 200 OK (List may be empty if no trainees) Fehler: 401 Unauthorized falls der Aufrufer nicht eingeloggt ist Fehler: 403 Forbidden, wenn der User zwar eingeloggt ist, aber keines der genannten Rechte besitzt</p>

Endpoint	POST /trainees
Beschreibung	Eine*n neue*n Lernende*n anlegen
Use Case	Lernende bearbeiten
Permission	trainee:write:all
Scope	global
Request	<p>firstName String not null, max. 30 Zeichen lastName String not null, max. 30 Zeichen teamId int not null</p>
Response	<p>idTrainee int not null firstName String not null lastName String not null teamId int not null</p>
Status Codes to be expected	<p>Erfolg: 201 Created Fehler: 400 Bad Request Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (teamId gibt es in der Datenbank nicht) Fehler: 409 Conflict (es besteht bereits ein Lernender mit identischem Vor- und Nachnamen)</p>

Endpoint	PATCH /trainees/{id}
Beschreibung	Die Infos eines*r bestehenden Lernende*n bearbeiten
Use Case	Lernende bearbeiten
Permission	trainee:write:all
Scope	global
Request Body	Keine id, da diese bereits in der Pfadvariable ist firstName String not null, max. 30 Zeichen lastName String not null, max. 30 Zeichen teamId int not null (alle Felder optional, wobei mindestens eines mitgeschickt werden muss)
Response	idTrainee int not null firstName String not null lastName String not null teamId int not null
Status Codes to be expected	Erfolg: 200 OK Fehler: 400 Bad Request (z.B. leerer Name oder ungültiges JSON empfangen) Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht kein Lernender mit der angegebenen ID oder es besteht kein Team mit der angegebenen ID) Fehler: 409 Conflict (es besteht bereits ein Lernender mit dem selben Vor- und Nachnamen)

Endpoint	DELETE /trainees/{id}
Beschreibung	Ein*e bestehende*r Lernende*r komplett und dauerhaft löschen
Use Case	Lernende bearbeiten
Permission	trainee:write:all
Scope	global
Request Body	-
Response	-
Status Codes to be expected	Erfolg: 204 No Content Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht kein Lernender mit der angegebenen ID)

ON DELETE Regel: Bei Lernenden schlage ich vor, dass wenn ein Lernender gelöscht wird, automatisch alle Tageszuteilungen dieses Lernenden auch gelöscht werden. Das ist zwar gefährlich, doch ist es auch sehr mühsam, wenn jemand z.B. im Mai kündigt oder im Februar

aus der verlängerten Probezeit ausscheidet: Bei ON DELETE NO ACTION müsste man jede Tageseinteilung löschen gehen.

ON DELETE CASCADE scheint mir hier die bessere Wahl: Wir müssen einfach sicherstellen, dass es im UI ein aufwendiges Löschen ist beim Dialog (der Dialog sollte z.B. erfordern, dass man Vor- und Nachname des Lernenden eintippt, ehe man löschen darf, vgl. GitHub wenn man ein Repo löschen will).

8.2 Monatszuteilungen (Jonas)

Endpoint	GET /monthassignments/{id}
Beschreibung	Eine spezifische Monatszuteilung lesen
Use Case	Monatszuteilungen sehen alle Teams
Permission	month-assignment:read:all
Scope	global
Request Body	-
Response	idMonthassignment int not null teamId int not null categoryId int not null monthId int not null
Status Codes to be expected	Erfolg: 200 OK Fehler: 400 Bad Request (id <= 0) Fehler: 401 Unauthorized falls der Aufrufer nicht eingeloggt ist Fehler: 404 Not Found, wenn es keine Monatszuteilung mit dieser ID gibt

Endpoint	GET /monthassignments
Beschreibung	Alle Monatszuteilungen lesen
Use Case	Monatszuteilungen sehen alle Teams
Permission	month-assignment:read:all
Scope	global
Request Body	-
Response	Liste aller Monatszuteilungen, wobei jede Monatszuteilung so aufgebaut ist: idMonthassignment int not null teamId int not null categoryId int not null monthId int not null

Status Codes to be expected	Erfolg: 200 OK (List may be empty if no assignments were made yet) Fehler: 401 Unauthorized falls der Aufrufer nicht eingeloggt ist Fehler: 403 Forbidden, wenn der User zwar eingeloggt ist, aber das genannte Recht nicht besitzt
------------------------------------	---

Endpoint	POST /monthassignments
Beschreibung	Eine neue Monatszuteilung erstellen z.B. PUNCS machen Medienraum im Januar
Use Case	Monatszuteilungen bearbeiten
Permission	month-assignment:write:all
Scope	global
Request	teamId Int not null categoryId Int not null monthId int not null
Response	idMonthassignment int not null teamId Int not null categoryId Int not null monthId int not null
Status Codes to be expected	Erfolg: 201 Created Fehler: 400 Bad Request Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (z.B. teamId gibt es in der Datenbank nicht) Fehler: 409 Conflict (es besteht bereits eine identische Monatszuteilung)

Endpoint	PATCH /monthassignments/{id}
Beschreibung	Änderungen an einer Monatszuteilung vornehmen
Use Case	Monatszuteilungen bearbeiten alle Teams
Permission	month-assignment:write:all
Scope	global
Request Body	Keine id, da diese bereits in der Pfadvariable ist teamId Int not null categoryId Int not null monthId int not null (alle Felder optional, wobei mindestens eines mitgeschickt werden muss)
Response	idMonthassignment int not null teamId Int not null categoryId Int not null monthId int not null
Status Codes to be expected	Erfolg: 200 OK Fehler: 400 Bad Request (z.B. leere ID oder ungültiges JSON empfangen) Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht keine Monatszuteilung mit der angegebenen ID oder es besteht kein Team/Ämtlikategorie/Monat mit der angegebenen ID) Fehler: 409 Conflict (es besteht bereits eine exakt identische Monatszuteilung)

Endpoint	DELETE /monthassignments/{id}
Beschreibung	Eine bestehende Monatszuteilung komplett und dauerhaft löschen
Use Case	Monatszuteilungen bearbeiten alle Teams
Permission	month-assignment:write:all
Scope	global
Request Body	-
Response	-
Status Codes to be expected	Erfolg: 204 No Content Fehler: 403 Forbidden (der Aufrufer hat die nötigen Rechte nicht) Fehler: 404 Not Found (es besteht keine Monatszuteilung mit der angegebenen ID)

8.3 Tageseinteilungen

8.4 Login

9 Backend

9.1 Technologien

Tools, Versionen

9.2 Backend

9.2.1 Projekt-Setup mit start.spring.io

The screenshot shows the Spring Initializr web interface. On the left, under 'Project', 'Gradle - Groovy' is selected. Under 'Language', 'Java' is selected. Under 'Spring Boot', '3.5.10 (SNAPSHOT)' is selected. The 'Project Metadata' section includes: Group: net.ictcampus, Artifact: baemtl, Name: baemtl, Description: Chores management tool for the ICT Campus of Swiss Post AG, Package name: net.ictcampus.baemtl, Packaging: Jar, Configuration: Properties, and Java version: 21. On the right, the 'Dependencies' section lists: Spring Web (WEB), Spring Data JPA (SQL), MySQL Driver (SQL), Lombok (DEVELOPER TOOLS), and Validation (IO). An 'ADD DEPENDENCIES...' button is visible at the top right of the dependencies section.

9.3 Dokumentation der wichtigsten Mechanismen

Damit die Menschen der Nachwelt auch noch verstehen was und wie unser Code ist. Wir sollten es ihnen möglichst leicht machen, den Code zu warten, damit das Tool hoffentlich möglichst lange in der BA genutzt werden wird! Yayyyy.

9.4 BA-Jahrgangwechsel / Reset

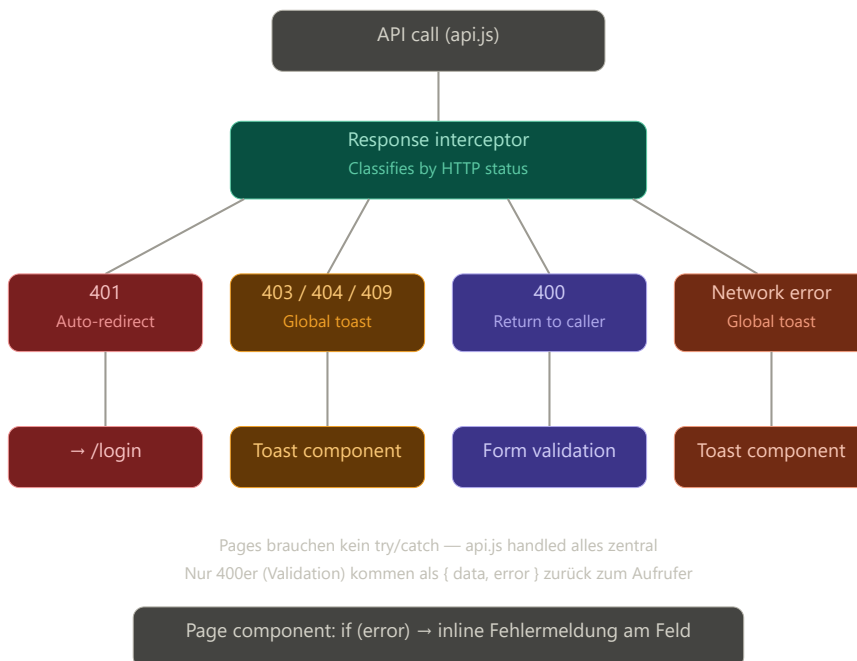
Das System ist auf Einjährigkeit ausgelegt. Ein manueller Reset der transaktionalen Daten (Zuweisungen, Teams, Lernende) erfolgt jährlich vor dem Start der neuen Basisausbildung im August. Stammdaten der Ämtlikategorien bleiben zur Wiederverwendung erhalten.

10 Frontend

10.1 Technologien

Tools, Versionen

10.2 Error-Handling-Strategie im Frontend



Das Ämtliplantool ist eine CRUD-intensive Applikation mit über 20 REST-Endpoints, verteilt auf mindestens fünf Entitäten (Teams, Lernende, Ämtlikategorien, Monatszuteilungen, Tageseinteilungen). Jeder dieser Endpoints kann auf unterschiedliche Weise fehlschlagen. Ohne eine durchdachte Error-Handling-Strategie müsste in jeder Page-Komponente bei jedem API-Aufruf ein eigener try/catch-Block geschrieben werden, was den Code unübersichtlich und schwer wartbar macht.

Unsere Lösung orientiert sich an einem in der React- bzw. Next.js-Welt etablierten Interceptor-Pattern: Fehler werden zentral im API-Service klassifiziert und je nach HTTP-Statuscode automatisch an die richtige Stelle geroutet. Die einzelnen Page-Komponenten bleiben dadurch nahezu frei von Error-Handling-Logik.

10.2.1 Fehlerklassifizierung nach HTTP-Statuscode

Unser Spring-Boot-Backend gibt bei Fehlern strukturierte JSON-Responses mit einem message-Feld zurück. Im Frontend nutzt der zentrale API-Service (api.js) den HTTP-Statuscode, um zu entscheiden, wie ein Fehler behandelt wird. Dabei unterscheiden wir vier Kategorien.

10.2.1.1 401 Unauthorized – Automatische Weiterleitung

Wenn das Backend einen 401-Statuscode zurückgibt, bedeutet das, dass der Benutzer nicht (mehr) eingeloggt ist – etwa weil die Session abgelaufen ist. In diesem Fall leitet der API-Service den Benutzer automatisch zur Login-Seite weiter (`window.location.href = "/login"`). Es wird kein Toast angezeigt und kein Fehler an die aufrufende Komponente zurückgegeben, da die Weiterleitung die einzig sinnvolle Reaktion ist. Dieses Verhalten ist besonders relevant, weil das Ämtliplantool mit drei verschiedenen Login-Typen arbeitet (Coach, Teamverantwortlich, Lernende).

10.2.1.2 403, 404 und 409 – Globaler Toast

Fehler mit den Statuscodes 403 (Forbidden), 404 (Not Found) und 409 (Conflict) sind informativ: Der Benutzer kann in der Regel nichts direkt dagegen unternehmen. Ein 403 bedeutet, dass die Berechtigung fehlt – beispielsweise, wenn ein Lernender versucht, einen Endpoint aufzurufen, der nur Coaches vorbehalten ist. Ein 404 tritt auf, wenn eine Ressource nicht mehr existiert, etwa weil ein anderer Coach das Team bereits gelöscht hat. Ein 409 zeigt einen Konflikt an, zum Beispiel wenn ein Team mit dem gleichen Namen bereits existiert oder eine identische Monatszuteilung schon vorhanden ist.

Für all diese Fälle zeigt der API-Service automatisch einen globalen Toast an – eine rote Benachrichtigungskarte, die von oben in die Seite gleitet, die Fehlermeldung des Backends anzeigt und nach fünf Sekunden wieder verschwindet. Die aufrufende Komponente erhält zwar das Error-Objekt zurück, muss es aber nicht selbst verarbeiten.

10.2.1.3 400 Bad Request – Rückgabe an den Aufrufer

Validierungsfehler (400) sind die einzige Kategorie, bei der die aufrufende Page-Komponente den Fehler selbst verarbeiten sollte. Beispiele sind ein leerer Teamname, ein zu langer Nachname oder eine ungültige ID. Diese Fehler können dem Benutzer am besten direkt am betroffenen Formularfeld angezeigt werden – etwa als roter Hinweistext unter dem Input-Feld. Der API-Service zeigt in diesem Fall keinen Toast an, sondern gibt das Error-Objekt im Rückgabewert zurück, damit die Komponente es inline darstellen kann.

10.2.1.4 Netzwerkfehler – Globaler Toast

Wenn das Backend nicht erreichbar ist (z. B. weil der Server nicht läuft oder die Netzwerkverbindung unterbrochen ist), fängt der API-Service die Exception im catch-Block ab und zeigt einen globalen Toast mit der Meldung «Server nicht erreichbar» an.

10.2.2 Architektur der Implementierung

Die Implementierung besteht aus drei Bausteinen, die zusammenwirken:

10.2.2.1 Zentraler API-Service (api.js)

Die Funktion `fetchWith` bildet den Kern des Error-Handlings. Sie nimmt jede API-Antwort entgegen, prüft den Statuscode und routet den Fehler entsprechend. Bei einem 204 (No Content) – typisch für DELETE-Operationen – gibt sie `{ data: null, error: null }` zurück, da kein Response-Body vorhanden ist. Bei einem 401 leitet sie sofort zum Login weiter. Bei 403, 404 und 409 ruft sie die globale `showError`-Funktion auf, die den Toast auslöst, und gibt zusätzlich `{ data: null, error: json }` zurück. Bei einem 400 gibt sie nur das Error-Objekt zurück, ohne einen Toast auszulösen. Bei einer erfolgreichen Antwort gibt sie `{ data: json, error: null }` zurück.

Dieses `{ data, error }`-Pattern ist in der React-Welt weit verbreitet (unter anderem bekannt durch Supabase und SWR) und ermöglicht einen einheitlichen, vorhersagbaren Umgang mit API-Antworten.

10.2.2.2 Error-Context und Toast-Komponente

Der `ErrorProvider` ist ein React Context, der über das gesamte Layout gespannt wird. Er verwaltet den aktuellen Fehlerzustand und rendert bei Bedarf die Toast-Komponente. Diese erscheint als fixierte Karte am oberen Bildschirmrand, gleitet per CSS-Animation in die Seite

hinein und verschwindet nach fünf Sekunden automatisch wieder. Der Benutzer kann den Toast auch manuell schliessen.

Damit der API-Service (der ausserhalb des React-Komponentenbaums lebt) auf den Toast zugreifen kann, wird eine Singleton-Referenz (`getGlobalShowError`) exportiert. Sobald der `ErrorProvider` gemountet ist, setzt er diese Referenz. Der API-Service ruft sie per `getGlobalShowError()?.(message)` auf – das optionale Chaining stellt sicher, dass kein Fehler auftritt, falls der Provider noch nicht initialisiert ist.

10.2.2.3 Einbindung im Layout

Der `ErrorProvider` wird in der zentralen `layout.js` um den gesamten Seiteninhalt (Sidebar und Main) gewickelt. Dadurch ist der Toast auf jeder Seite verfügbar, ohne dass einzelne Seiten etwas konfigurieren müssen.

10.2.3 Resultat für die Page-Komponenten

Durch diese Architektur bleiben die Page-Komponenten praktisch frei von Error-Handling-Code. Ein typischer Löschvorgang sieht folgendermassen aus: Die Komponente ruft `api.delete` auf, prüft in einer einzigen Zeile ob ein Fehler vorliegt (`if (error) return`), und aktualisiert andernfalls den lokalen State. Falls ein Fehler auftritt, hat der API-Service den Toast bereits automatisch ausgelöst – die Komponente muss nichts weiter tun.

Nur bei Formularen mit Validierungsfehlern (400) lohnt es sich, das `error`-Objekt auszuwerten und die Fehlermeldung direkt am betroffenen Eingabefeld anzuzeigen. Dies betrifft beispielsweise das Erstellen und Bearbeiten von Teams, Lernenden und Ämtlikategorien.

Dieses Vorgehen ist Industry Best Practice in der React- und Next.js-Entwicklung. Es entspricht dem Interceptor-Pattern, wie es etwa Axios als Built-in-Feature anbietet, und dem `{ data, error }`-Pattern, das in der modernen React-Welt als Standard gilt. Die Vorteile sind eine klare Trennung von Verantwortlichkeiten, minimaler Boilerplate in den Page-Komponenten und eine konsistente Benutzererfahrung bei Fehlern über die gesamte Applikation hinweg.

11 Testfälle

12 Integration

13 Auswertung

13.1 Auswertung der Anforderungen

13.2 Zielprüfung

13.3 Reflexion

13.4 Fazit