

PRAXIS DER SOFTWAREENTWICKLUNG

SOLIDARISCHE RAUMNUTZUNG

Praxis der Softwareentwicklung (PSE)
Wintersemester 2024/25

Pflichtenheft

Karlsruher Institut für Technologie (KIT)
Institut für Anthropomatik und Robotik (IAR)
Forschungsgruppe Mensch-Maschine-Interaktion und Barrierefreiheit (MBI)
Prof. Dr. Kathrin Gerling
Adenauerring 10, Gebäude 50.28
76131 Karlsruhe

Betreuer: Sabrina Burtscher, Dmitry Alexandrovsky

Projektteilnehmer:

Name	E-Mail-Adresse
Antonia Ammon	uipkm@student.kit.edu
Ben Steinle	ufikl@student.kit.edu
Johannes Frohnmeier	uczkf@student.kit.edu
Alexander Klee	ukvpq@student.kit.edu
Jannik Hönlinger	uouyo@student.kit.edu

Karlsruhe, 26. März 2025

Inhaltsverzeichnis

1	Einleitung	5
2	Zielbestimmung	6
2.1	Musskriterien	6
2.2	Wunschkriterien	7
2.3	Abgrenzungskriterien	7
3	Produktübersicht	9
3.1	Betriebsbedingungen	9
3.2	Aktivitätsdiagramm-Diagramme	10
3.2.1	Admin Funktionalität	10
3.2.2	Anmeldeprozess	11
3.2.3	Termin erstellen	12
3.2.4	Termine verwalten	14
3.2.5	Termin Ansicht	15
4	Produktfunktionen	16
4.1	Unangemeldete Nutzende	16
4.2	Angemeldete Nutzende	16
4.3	Administrator*innen	17
5	Produktdaten	30
6	Nichtfunktionale Anforderungen	32
7	Benutzeroberfläche	33
7.1	Hauptseite und Buchung	33
7.2	Terminübersicht	37
7.3	Adminstration	39
8	Technische Produktumgebung	40
8.1	Entwicklungsumgebung	40
8.2	Hardware	40
8.2.1	Backend	40

8.2.2	Frontend	40
8.3	Software	41
8.3.1	Backend	41
8.3.2	Frontend	41
8.4	Schnittstellen	41
9	Testfälle und Testeszenarien	42
9.1	Testfälle	42
9.2	Testfallszenarien	43
	Glossar	45

Abbildungsverzeichnis

3.1	Aktivitätsdiagramm für die Admin-Funktionalität	10
3.2	Aktivitätsdiagramm für den Anmeldeprozess	11
3.3	Aktivitätsdiagramm für das Erstellen eines Termins	13
3.4	Aktivitätsdiagramm für das Verwalten von Terminen	14
3.5	Aktivitätsdiagramm für die Ansicht <i>Termin</i>	15
7.1	Startseite der Anwendung	33
7.2	Termin-erstellen	34
7.3	Anmeldungsseite	35
7.4	Quick Checkout	36
7.5	Reservierungsübersicht	37
7.6	Reserverierung im Kalender	38
7.7	Benutzeradministrationsoberfläche	39

1 Einleitung

In modernen und inklusiven Arbeits- und Lernumgebungen spielen Rückzugsorte eine zentrale Rolle um produktives Arbeiten und das Wohlbefinden aller gewährleisten zu können. Das Institut für Mensch-Maschine-Interaktion und Barrierefreiheit (MIB) am KIT bietet einen Ruheraum, welcher für Meetings, kurzen Pausen oder als Rückzugsort nach einer Reizüberflutung genutzt werden kann.

Derzeit wird der Status des Raumes durch ein einfaches Türschild angezeigt, welches lediglich den Status *verfügbar* oder *besetzt* anzeigt. Dieses System bietet jedoch keine Möglichkeit, die Dringlichkeit sowie die Nutzung zu kommunizieren und zu planen.

Im Rahmen dieses Projekts wird ein innovatives Buchungssystem entwickelt, das über ein klassisches *frei-* oder *gebucht-*System hinausgeht. Ziel ist es, ein System zu schaffen, das es den Nutzenden ermöglicht, ihre Bedürfnisse differenziert anzugeben. Dadurch soll eine transparente und flexible Abstimmung über die Raumnutzung ermöglicht werden, die sowohl individuellen Bedürfnissen als auch einer optimalen Raumverteilung gerecht wird.

2 Zielbestimmung

2.1 Musskriterien

- ⟨MK1⟩ Die Anwendung muss als Web-Applikation realisiert werden.
- ⟨MK2⟩ Nutzende der Anwendung müssen sich mit ihrem KIT-Konto per OIDC oder einem lokalen Gastkonto anmelden können.
- ⟨MK3⟩ Nutzende müssen sich abmelden können.
- ⟨MK4⟩ Die Anwendung muss die Ansichten Kalender, Termin, Termin-erstellen, Login, Kontenliste und Terminübersicht anbieten.
- ⟨MK5⟩ Die Ansicht *Kalender* muss einen klaren Überblick über die bereits reservierten Zeiten geben.
- ⟨MK6⟩ Die Ansicht *Kalender* muss die Öffnungszeiten des Raumes darstellen.
- ⟨MK7⟩ Die Ansicht *Kalender* muss die Termine des/r angemeldeten Nutzenden hervorgehoben darstellen.
- ⟨MK8⟩ Die Ansicht *Termin* muss die Möglichkeit bieten, genauere Informationen über einen Termin darzustellen.
- ⟨MK9⟩ Die Ansicht *Termin-Erstellen* muss die Möglichkeit bieten, einen Raum für eine bestimmte Zeitperiode zu reservieren.
- ⟨MK10⟩ Bei der Reservierung eines Raumes muss optional die Möglichkeit bestehen eine Beschreibung zu hinterlegen. Hierbei müssen die Nutzenden klar darauf hingewiesen werden, wer diese Daten einsehen kann.
- ⟨MK11⟩ Die Terminübersicht muss den Nutzenden die Möglichkeit bieten, all ihre Termine zu verwalten.
- ⟨MK12⟩ Die Priorität eines Termins ist in drei Stufen gegliedert.
- ⟨MK13⟩ Beim Erstellen eines Termins wählen die Nutzenden aus, ob sie ihren Raum mit anderen Personen teilen möchten. Dabei stehen die Optionen *Ja*, *Nein* und *Auf Anfrage* zur Auswahl.
- ⟨MK14⟩ Die Anwendung muss Terminkonflikte reibungslos mithilfe der Prioritäten lösen können. Dabei überschreiben Termine mit höherer Priorität andere.
- ⟨MK15⟩ Die Anwendung muss in der Lage sein, Nutzende per E-Mail darüber zu informieren, wenn ihr Termin durch einen Termin mit höherer Priorität überschrieben wurde.
- ⟨MK16⟩ Nutzende der Anwendung müssen in der Lage sein, eine Reservierung zu stornieren.
- ⟨MK17⟩ Es muss ein Adminkonto geben, welches per Passwort authentifiziert wird. Nur der

Server-Admin darf dieses Passwort ändern können.

⟨MK18⟩ Die Ansicht *Kontoliste* muss den Admins die Möglichkeit bieten, einzelne Konten sowie die Anmeldung per Gastkonto zu deaktivieren.

⟨MK19⟩ Admins müssen in der Ansicht *Termin* Termine löschen können.

⟨MK20⟩ Es muss ein farbcodiertes Banner geben, der den aktuellen Status des Raumes anzeigt.

⟨MK21⟩ Admins müssen in der Ansicht *Kalender* die Öffnungszeiten einstellen können

2.2 Wunschkriterien

⟨WK1⟩ Es könnte die Möglichkeit geben, mehr als einen Raum zur Buchung anzubieten. Dabei könnte eine Raumauswahl vor der Ansicht *Kalender* die verschiedenen Möglichkeiten präsentieren. Existiert nur ein Raum, wird diese Auswahl übersprungen.

⟨WK2⟩ In der Ansicht *Kalender* könnten Feiertage automatisch eingebunden werden.

⟨WK3⟩ Admins könnten die Möglichkeit haben, geplante Wartungs- und Sperrzeiten einzurichten.

⟨WK4⟩ Termine könnten nach der Buchung im iCal-Format zum Export angeboten werden.

⟨WK5⟩ Nutzende könnten in der Ansicht *Termin* die Möglichkeit haben, ihre eigenen Termine zu bearbeiten.

⟨WK6⟩ Die Ansicht *Kalender* könnte visualisieren, welche Termine bereits in der Vergangenheit liegen und wo der Übergang von der Vergangenheit zur Zukunft liegt.

⟨WK7⟩ Tooltips könnten Nutzenden erklären, wofür bestimmte Elemente der UI verwendet werden.

⟨WK9⟩ Es könnte einen physischen Panik-Button geben.

⟨WK10⟩ Die Anwendung könnte in der Lage sein, Nutzende zu informieren, falls ein gewünschter Termin frei wird.

⟨WK11⟩ Es könnte einen Quick-Checkin-Button geben, welcher eine vorausgefüllte Terminerstellung öffnet. Dieser bietet auch eine Alternative zur Interaktion mit dem Kalender.

⟨WK12⟩ Es könnte einen Quick-Checkout-Button geben, der vorzeitiges Beenden eines Termines ermöglicht.

⟨WK13⟩ Admins könnten eine Statistik-Ansicht nutzen.

2.3 Abgrenzungskriterien

⟨AK1⟩ Die Verteilung von Buchungen zwischen ähnlichen Räumen ist nicht Teil des Projekts.

⟨AK2⟩ Das Skalieren der Anwendung auf eine große Anzahl von Räumen ist nicht vorgesehen.

⟨AK3⟩ Die Buchung von Räumen für mehrere Tage ist nicht vorgesehen.

⟨AK4⟩ Die Verwaltung von Räumen, die mehrere Arbeitsplätze umfassen, ist nicht vorgesehen.

⟨AK5⟩ Die Entwicklung plattformspezifischer Anwendungen und der dafür notwendigen APIs ist nicht vorgesehen.

⟨AK6⟩ Die Reservierung ist nur für die nahe Zukunft gedacht, eine Langzeitplanung ist nicht vorgesehen.

⟨AK7⟩ Die Reservierung wiederholter Termine ist nicht vorgesehen.

⟨AK8⟩ E-Mail-Adressen von Gastkonten werden nicht verifiziert.

3 Produktübersicht

Um sich einen Überblick über die im Produkt enthaltenen Funktionen zu verschaffen, wird in diesem Kapitel eine Übersicht über die grundlegenden Funktionen in Form von Aktivitätsdiagrammen gegeben. Zudem werden die Betriebsbedingungen beschrieben.

3.1 Betriebsbedingungen

- Die Software soll in einem Docker-Container laufen.
- Die Software soll weitgehend ohne Unterbrechung laufen.
- Ständige Wartung ist nicht vorgesehen, die Software soll weitgehend autonom laufen.
- Übliche Administrationsaufgaben sollen über die Administrationsoberfläche und ohne Neustart der Software möglich sein.
- Die RAM-Nutzung soll möglichst gering sein (Ideal: unter 1GB)

3.2 Aktivitätsdiagramm-Diagramme

3.2.1 Admin Funktionalität

In Abbildung 3.1 ist das Aktivitätsdiagramm für die Admin-Funktionalität dargestellt, welches die Musskriterien $\langle MK3 \rangle$, $\langle MK4 \rangle$, $\langle MK17 \rangle$, $\langle MK18 \rangle$ und $\langle MK19 \rangle$ abdeckt, sowie die Produktfunktionen $\langle F100 \rangle$, $\langle F10 \rangle$, $\langle F30 \rangle$, $\langle F50 \rangle$, $\langle F60 \rangle$, $\langle F90 \rangle$ und $\langle F110 \rangle$.

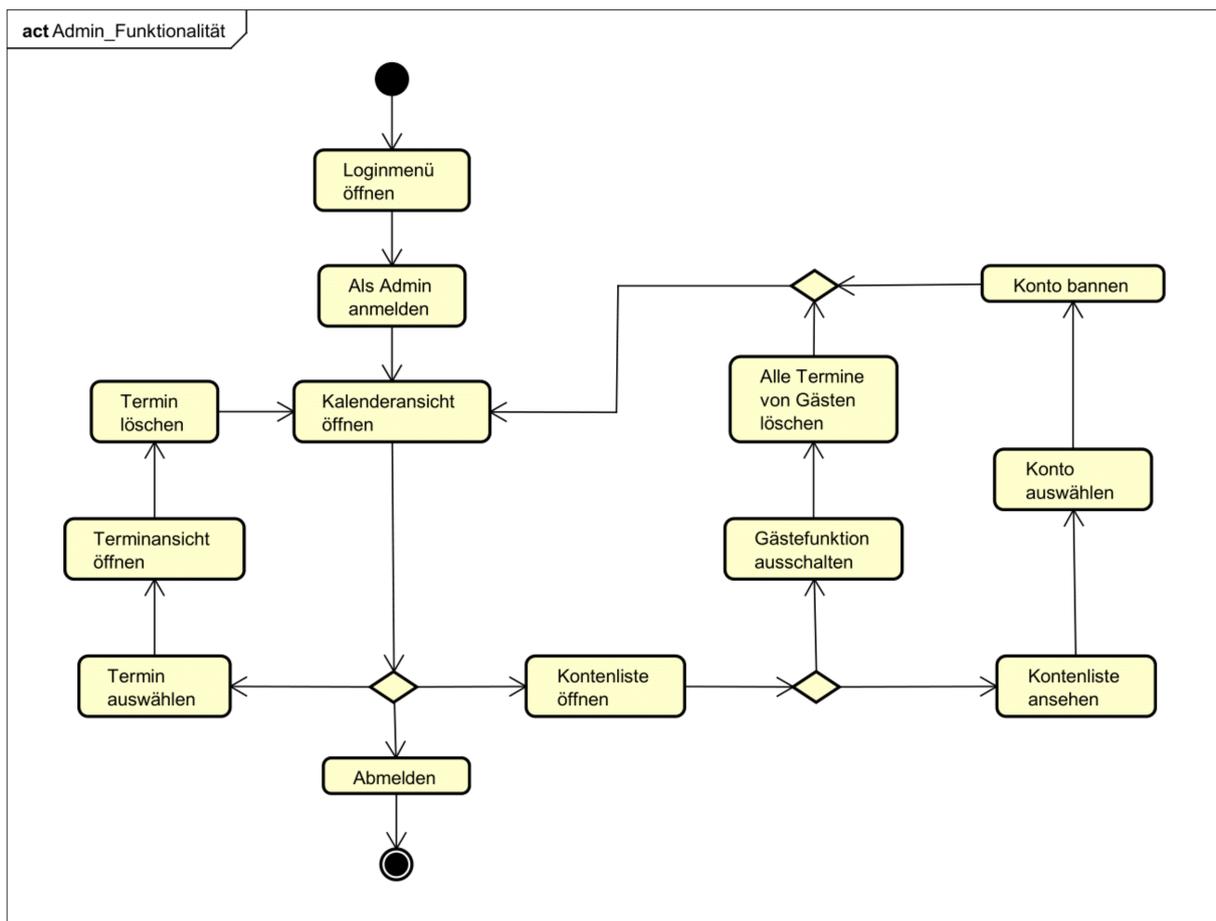


Abbildung 3.1: Aktivitätsdiagramm für die Admin-Funktionalität

3.2.2 Anmeldeprozess

In Abbildung 3.2 ist das Aktivitätsdiagramm für den Anmeldeprozess dargestellt, welches die Musskriterien $\langle MK2 \rangle$ und $\langle MK17 \rangle$ abdeckt, sowie die Produktfunktionen $\langle F20 \rangle$ und $\langle F100 \rangle$.

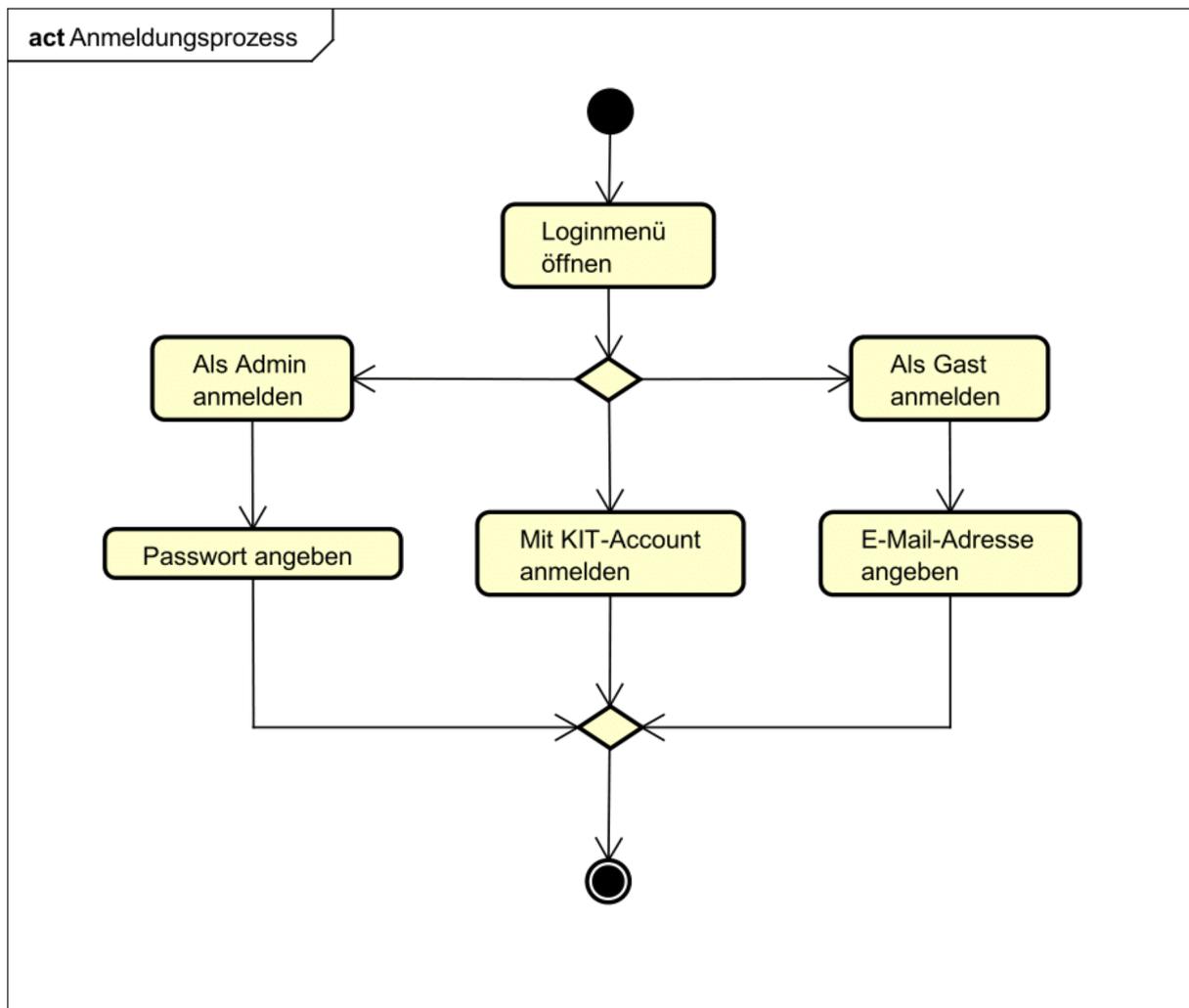


Abbildung 3.2: Aktivitätsdiagramm für den Anmeldeprozess

3.2.3 Termin erstellen

In Abbildung 3.3 ist das Aktivitätsdiagramm für das Erstellen eines Termins dargestellt, welches die Musskriterien $\langle MK2 \rangle$, $\langle MK3 \rangle$, $\langle MK5 \rangle$, $\langle MK6 \rangle$, $\langle MK9 \rangle$, $\langle MK10 \rangle$, $\langle MK12 \rangle$ und $\langle MK13 \rangle$ abdeckt, sowie die Produktfunktionen $\langle F20 \rangle$, $\langle F30 \rangle$ und $\langle F40 \rangle$.

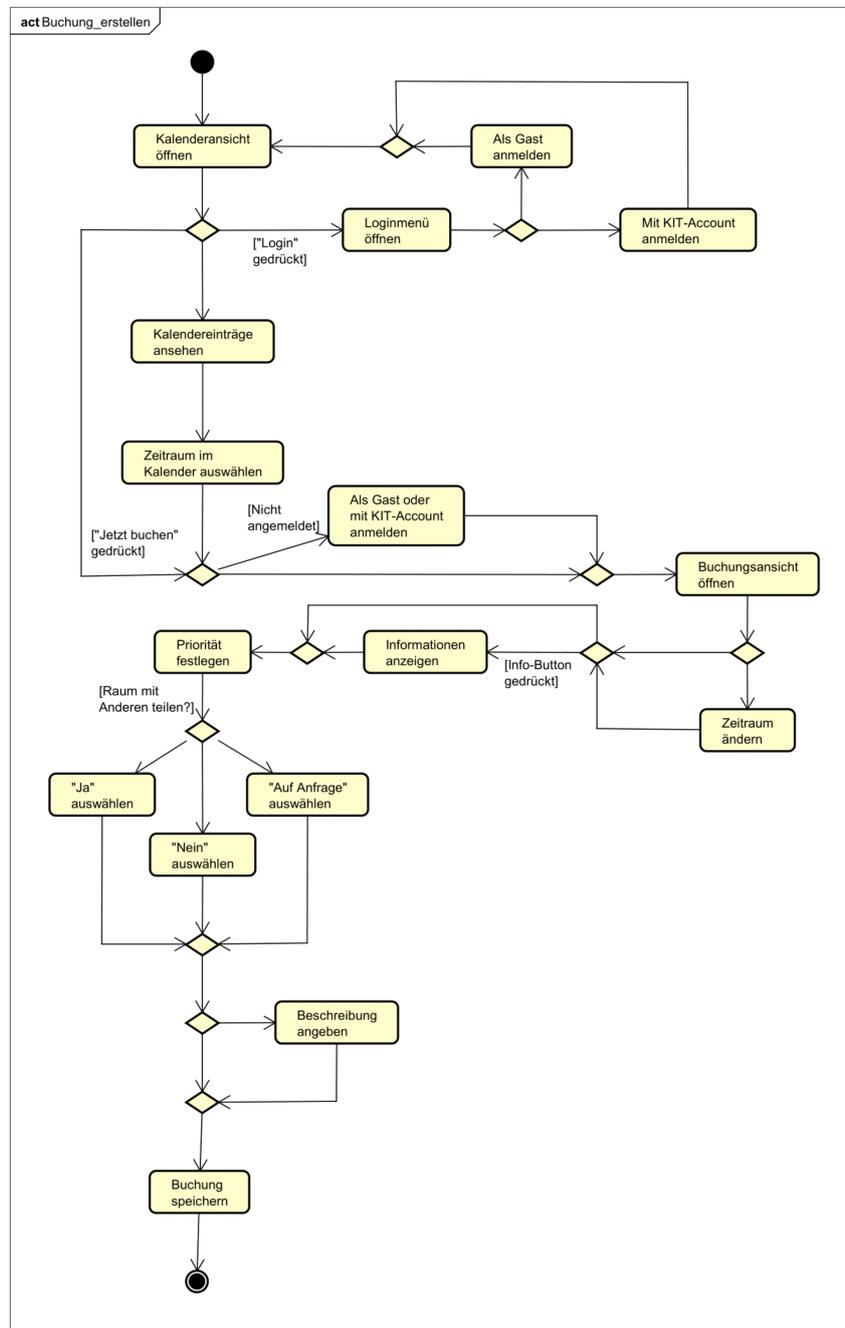


Abbildung 3.3: Aktivitätsdiagramm für das Erstellen eines Termins

3.2.4 Termine verwalten

In Abbildung 3.4 ist das Aktivitätsdiagramm für das Verwalten von Terminen dargestellt, welches die Musskriterien $\langle MK2 \rangle$, $\langle MK11 \rangle$ und $\langle MK16 \rangle$ abdeckt, sowie die Produktfunktionen $\langle F20 \rangle$ und $\langle F90 \rangle$.

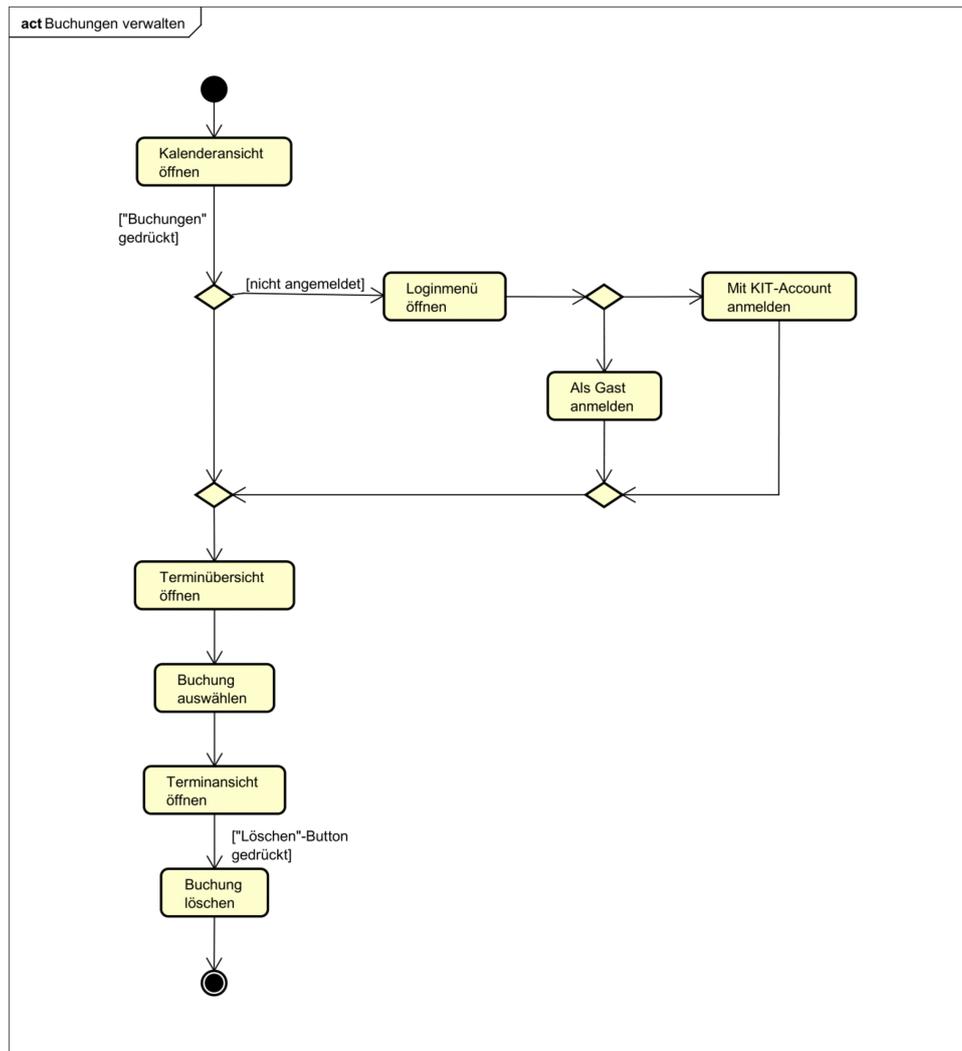


Abbildung 3.4: Aktivitätsdiagramm für das Verwalten von Terminen

3.2.5 Termin Ansicht

In Abbildung 3.5 ist das Aktivitätsdiagramm für die Ansicht *Termin* dargestellt, welches die Musskriterien $\langle MK2 \rangle$, $\langle MK5 \rangle$, $\langle MK8 \rangle$, $\langle MK16 \rangle$, $\langle MK17 \rangle$ und $\langle MK19 \rangle$ abdeckt, sowie die Produktfunktionen $\langle F20 \rangle$, $\langle F50 \rangle$, $\langle F90 \rangle$, $\langle F100 \rangle$ und $\langle F130 \rangle$.

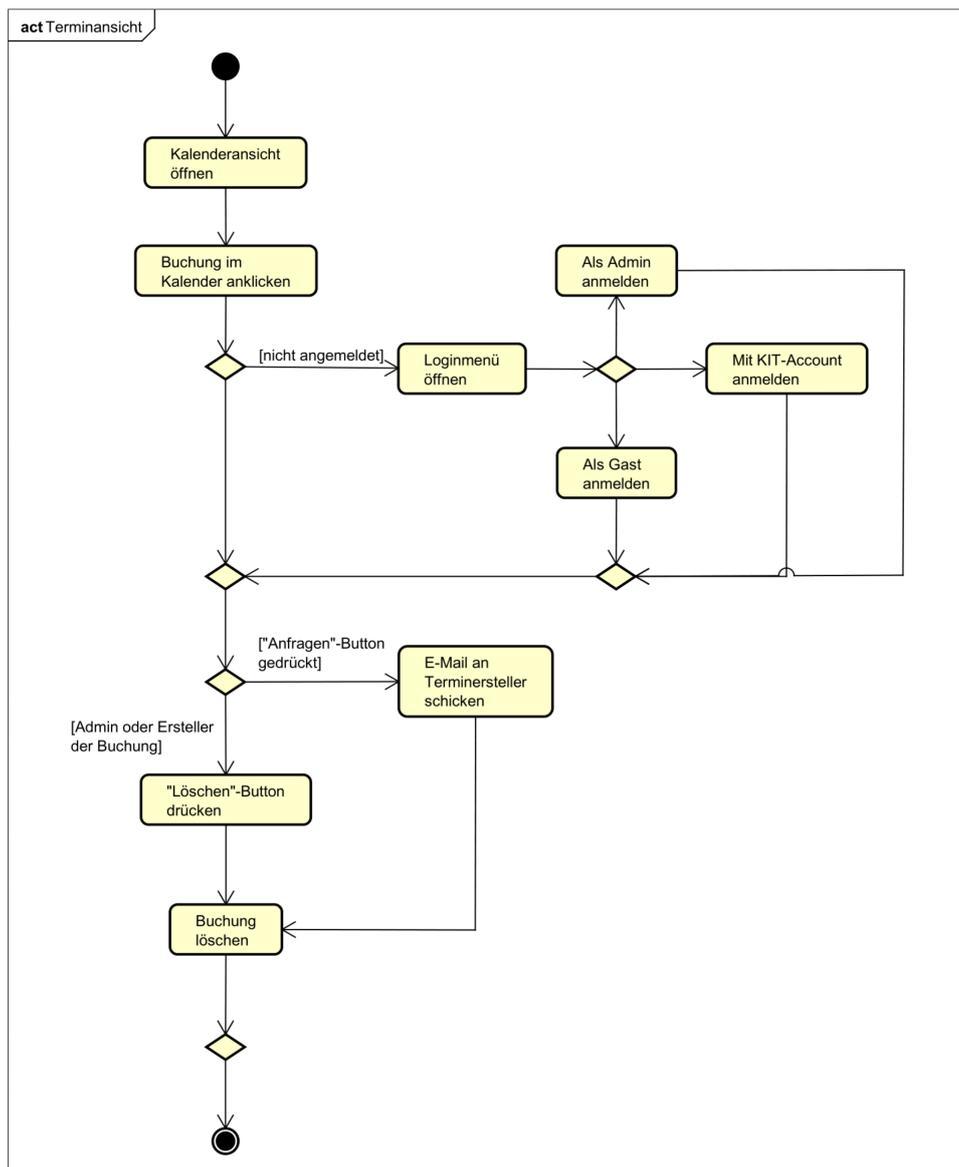


Abbildung 3.5: Aktivitätsdiagramm für die Ansicht *Termin*

4 Produktfunktionen

4.1 Unangemeldete Nutzende

Nr.	Funktion	Kriterien
$\langle F10 \rangle$	Startseite besuchen	$\langle MK1 \rangle$, $\langle MK4 \rangle$, $\langle WK1 \rangle$
$\langle F20 \rangle$	Login	$\langle MK2 \rangle$, $\langle MK4 \rangle$
$\langle F70 \rangle$	Anzeige des Raumstatus	$\langle MK18 \rangle$

Tabelle 4.1: Überblick von Funktionen für unangemeldete Nutzende.

4.2 Angemeldete Nutzende

Nr.	Funktion	Kriterien
$\langle F10 \rangle$	Startseite besuchen	$\langle MK1 \rangle$, $\langle MK4 \rangle$, $\langle WK1 \rangle$
$\langle F100 \rangle$	Login mit Adminkonto	$\langle MK17 \rangle$
$\langle F30 \rangle$	Abmelden	$\langle MK3 \rangle$, $\langle MK4 \rangle$
$\langle F40 \rangle$	Reservieren	$\langle MK9 \rangle$, $\langle MK10 \rangle$, $\langle MK12 \rangle$, $\langle MK13 \rangle$, $\langle MK14 \rangle$, $\langle WK4 \rangle$, $\langle WK11 \rangle$
$\langle F70 \rangle$	Benachrichtigung	$\langle WK10 \rangle$
$\langle F80 \rangle$	Anzeige des Raumstatus	$\langle MK20 \rangle$
$\langle F90 \rangle$	Stornierung einer Reservierung	$\langle MK16 \rangle$
$\langle F130 \rangle$	Terminkonfliktauflösung	$\langle MK14 \rangle$

Tabelle 4.2: Überblick von Funktionen für angemeldete Nutzende.

4.3 Administrator*innen

Nr.	Funktion	Kriterien
⟨F10⟩	Startseite besuchen	⟨MK1⟩, ⟨MK4⟩, ⟨WK1⟩
⟨F30⟩	Abmelden	⟨MK3⟩, ⟨MK4⟩
⟨F40⟩	Reservieren	⟨MK9⟩, ⟨MK10⟩, ⟨MK12⟩, ⟨MK13⟩, ⟨MK14⟩, ⟨WK4⟩, ⟨WK11⟩
⟨F50⟩	Löschen von Terminen durch Administratoren	⟨MK18⟩
⟨F60⟩	Deaktivierung von Konten	⟨MK19⟩
⟨F70⟩	Benachrichtigung bei freiem Raum	⟨WK10⟩
⟨F80⟩	Anzeige des Raumstatus	⟨MK20⟩
⟨F90⟩	Stornierung einer Reservierung	⟨MK16⟩
⟨F100⟩	Anmelden als Admin	⟨MK17⟩
⟨F110⟩	Deaktivieren von Gastkonto-Anmeldung	⟨MK18⟩
⟨F120⟩	Öffnungszeiten einstellen	⟨MK21⟩
⟨F130⟩	Terminkonfliktauflösung	⟨MK14⟩

Tabelle 4.3: Überblick von Funktionen für Admins.

Startseite ⟨F10⟩

Anwendungsfall: Initiales besuchen der Webseite

Anforderung: ⟨MK1⟩, ⟨MK4⟩, ⟨WK1⟩

Ziel: Den Nutzenden wird ohne Anmeldung die Ansicht *Kalender* gezeigt.

Vorbedingung: Die Nutzenden haben eine Internetverbindung und im Browser die Webseite geöffnet.

Nachbedingung Erfolg: Die Nutzenden sehen die Ansicht *Kalender* mit zukünftigen Reservierungen.

Nachbedingung Fehlschlag: Die Nutzenden sind nicht in der Lage mit der Webseite weiter zu interagieren.

Auslösendes Ereignis: Die Nutzenden verwenden ihre Browser, um die Webseite aufzurufen.

Beschreibung:

1. Die Nutzenden navigieren in ihren Browsern auf die Webseite.
2. Die Startseite lädt und zeigt den Nutzenden die Ansicht *Kalender* an.

Erweiterung: Sollte es mehr als ein Arbeitsraum geben, so sollte hier vom Nutzenden die Auswahl getroffen werden, bevor die Ansicht *Kalender* angezeigt wird.

Anmelden ⟨F20⟩

Anwendungsfall: Anmelden bzw. Registrieren

Anforderung: ⟨MK2⟩, ⟨MK4⟩

Ziel: Die Nutzenden können sich mit ihren KIT-Konten oder lokalen Gastkonten anmelden.

Vorbedingung: Die Nutzenden haben eine Internetverbindung und die Anwendung ist gestartet.

Nachbedingung Erfolg: Die Nutzenden sind angemeldet und können Ereignisse ansehen und bearbeiten.

Nachbedingung Fehlschlag: Die Nutzenden sind nicht angemeldet und können keine Ereignisse ansehen oder bearbeiten.

Auslösendes Ereignis: Die Nutzenden öffnen die Anwendung und versuchen, einem Termin zu erstellen oder mit einem bereits existierenden Termin zu interagieren.

Beschreibung:

1. Die Nutzenden wählen in einem Dialog die Anmeldemethode aus.
2. Sollten diese noch keinen Account haben, so wird statt eine Anmeldung eine Registrierung vollzogen.
3. Die Nutzenden werden, falls nötig, auf die KIT-Login-Seite weitergeleitet.
4. Falls keine lokale Gast-Anmeldung gewählt wurde, geben die Nutzenden ihre Zugangsdaten ein.
5. Die Anwendung überprüft die Zugangsdaten und meldet die Nutzenden an.
6. Bei einer Gast-Anmeldung wird zusätzlich noch eine E-Mail hinterlegt.
7. Die Nutzenden werden zur nächsten Seite weitergeleitet.

Abmelden ⟨F30⟩

Anwendungsfall: Abmelden

Anforderung: ⟨MK3⟩

Ziel: Die Nutzenden können sich von der Anwendung abmelden.

Vorbedingung: Der/die Nutzende ist angemeldet.

Nachbedingung Erfolg: Der/die Nutzende ist abgemeldet und wird auf die Startseite weitergeleitet.

Nachbedingung Fehlschlag: Der/die Nutzende ist nicht abgemeldet und erhält eine Fehlermeldung.

Auslösendes Ereignis: Die/der Nutzende wählt die Option zum Abmelden aus.

1. Der/die Nutzende wählt die Funktion zum Abmelden aus.
2. Der/die Nutzende wird abgemeldet und auf die Startseite weitergeleitet.

Reservierung $\langle F40 \rangle$

Anwendungsfall: Reservierung

Anforderung: $\langle MK9 \rangle$ und $\langle MK10 \rangle$, $\langle MK12 \rangle$, $\langle MK13 \rangle$, $\langle MK14 \rangle$, $\langle WK4 \rangle$ und $\langle WK11 \rangle$

Ziel: Die nutzende Person kann einen Raum reservieren.

Vorbedingung: Die nutzende Person ist angemeldet und in der Ansicht *Kalender*.

Nachbedingung Erfolg: Eine Reservierung wird gespeichert und die nutzende Person zur Ansicht *Termin* weitergeleitet.

Nachbedingung Fehlschlag: Die Reservierung wird nicht gespeichert und die nutzende Person erhält eine Fehlermeldung.

Auslösendes Ereignis: Die nutzende Person wählt einen Startzeitpunkt.

Beschreibung:

1. Die nutzende Person wählt einen Start- und Endzeitpunkt auf 15 Minuten genau und kann diese gegebenenfalls anpassen.
2. Die nutzende Person gibt optional eine Beschreibung an.
3. Die nutzende Person gibt an, ob sie bereit ist, den Raum mit weiteren zu teilen.
4. Die nutzende Person wählt eine Prioritätsstufe.
5. Die nutzende Person bestätigt ihre Eingaben.
6. Die Anwendung prüft, ob der Raum in dem Zeitraum verfügbar ist.
7. Ist dies der Fall, wird die Reservierung gespeichert und die nutzende Person zur Ansicht *Kalender* weitergeleitet. Wenn nötig, wird eine E-Mail an die Ersteller*innen überschriebener Reservierungen gesendet.
8. Die Anwendung überprüft, ob die angegebene Priorität andere Termine beeinflusst und löst Konflikte entsprechend. (siehe $\langle F130 \rangle$)
9. Der nutzenden Person wird dieser Konflikt mitgeteilt und sobald dieser geklärt, wird den beteiligten Parteien der Entschluss via E-Mail mitgeteilt.

Erweiterung:

1. Nach dem Erstellen des Termins könnte dieser als iCal-Export zur Verfügung gestellt werden.
2. Durch einen zusätzlichen Quick-Checkin-Button könnte dieser Prozess simpler und kürzer gemacht werden.

Löschen von Terminen durch das Adminkonto ⟨F50⟩

Anwendungsfall: Löschen von Terminen durch das Adminkonto

Anforderung: ⟨MK19⟩

Ziel: Ein/e Administrator*in kann bestehende Termine löschen.

Vorbedingung: Ein/e Administrator*in ist angemeldet und die Ansicht *Kalender* wird angezeigt.

Nachbedingung Erfolg: Der Termin wird aus dem System entfernt und der/die Administrator*in erhält eine Erfolgsmeldung.

Nachbedingung Fehlschlag: Der Termin wird nicht gelöscht und der/die Administrator*in erhält eine Fehlermeldung.

Auslösendes Ereignis: Ein/e Administrator*in wählt einen Termin aus, welcher gelöscht werden soll.

Beschreibung:

1. Ein/e Administrator*in öffnet die Ansicht *Kalender*.
2. Der/die Administrator*in wählt den Termin aus, welcher gelöscht werden soll.
3. Der/die Administrator*in wird gefragt, ob es den Termin wirklich löschen möchte (Bestätigungsabfrage).
4. Falls der/die Administrator*in bestätigt, wird der Termin gelöscht.
5. Eine Bestätigungsmeldung wird angezeigt (z.B. 'Termin erfolgreich gelöscht').

Deaktivierung von Konten ⟨F60⟩

Anwendungsfall: Deaktivierung von Konten

Anforderung: ⟨MK18⟩

Ziel: Ein Konto wird vom Adminkonto gesperrt.

Vorbedingung: Ein/e Administrator*in ist angemeldet und betrachtet die Ansicht *Kontoliste*.

Nachbedingung Erfolg: Das Konto wird deaktiviert und kann nicht mehr zum Anmelden verwendet werden.

Nachbedingung Fehlschlag: Das Konto bleibt aktiv und der/die Administrator*in erhält eine Fehlermeldung.

Auslösendes Ereignis: Der/die Administrator*in wählt ein Konto aus, welches deaktiviert werden soll.

Beschreibung:

1. Ein/e Administrator*in öffnet die Liste der Konten (Ansicht *Kontoliste*).
2. Der/die Administrator*in wählt ein Konto aus, welches deaktiviert werden soll.
3. Der/die Administrator*in bestätigt die Deaktivierung.
4. Das Konto wird deaktiviert und aus der aktiven Liste entfernt.
5. Eine Bestätigungsmeldung wird angezeigt (z.B. *Konto erfolgreich deaktiviert*).

Benachrichtigung bei freiem Raum ⟨F70⟩

Anwendungsfall: Benachrichtigung über freigewordenen Raum

Anforderung: ⟨WK10⟩

Ziel: Die Anwendung informiert Nutzende, wenn ein Raum wieder frei wird.

Vorbedingung: Die Nutzenden haben sich für den Raum interessiert und haben eine Benachrichtigung angefordert.

Nachbedingung Erfolg: Die Nutzenden werden über die Freigabe des Raums per E-Mail oder Benachrichtigung informiert.

Nachbedingung Fehlschlag: Die Nutzenden erhalten keine Benachrichtigung, und eine Fehlermeldung wird angezeigt.

Auslösendes Ereignis: Der Raum wird freigegeben, nachdem eine vorherige Reservierung aufgehoben oder geändert wurde.

Beschreibung:

1. Die Nutzenden wählen aus, ob sie für den Raum Benachrichtigungen erhalten möchten, wenn er wieder frei wird.
2. Sobald der Raum wieder verfügbar ist, prüft das System, ob die nutzende Person für den Raum eine Benachrichtigung angefordert haben.
3. Den betroffenen Nutzenden wird eine E-Mail oder Benachrichtigung zugeschickt, dass der Raum nun wieder verfügbar ist.

Anzeige des Raumstatus ⟨F80⟩

Anwendungsfall: Anzeige des Raumstatus

Anforderung: ⟨MK20⟩ ⟨WK9⟩

Ziel: Die Anwendung stellt den Raumstatus, einschließlich der aktuellen Belegung und Priorität, übersichtlich dar.

Vorbedingung: Die Nutzenden sind auf der Ansicht *Kalender* und müssen nicht angemeldet sein.

Nachbedingung Erfolg: Der Raumstatus wird auf einer öffentlichen Seite korrekt angezeigt, inklusive Belegung und Priorität (z.B. durch farbige Banner).

Nachbedingung Fehlschlag: Der Raumstatus wird nicht angezeigt oder ist unvollständig.

Auslösendes Ereignis: Die Nutzenden öffnen die Ansicht *Kalender*.

Beschreibung:

1. Die Nutzenden öffnen die Seite.
2. Die Belegung des Raumes wird mit dem entsprechenden Status angezeigt, der die aktuelle Belegung widerspiegelt.
3. Falls der Raum belegt ist, wird dies deutlich sichtbar gemacht (z.B. durch ein Banner oder ein entsprechendes Symbol).
4. Dabei wird die Priorität klar dargestellt.
5. Der Raumstatus ist für alle Nutzenden sichtbar, ohne dass eine Anmeldung erforderlich ist.

Stornierung einer Reservierung ⟨F90⟩

Anwendungsfall: Stornierung einer Reservierung

Anforderung: ⟨MK16⟩

Ziel: Die Nutzenden können eine bestehende Reservierung stornieren.

Vorbedingung: Die Nutzenden sind angemeldet und haben eine aktive Reservierung.

Nachbedingung Erfolg: Die Reservierung wird storniert und die Nutzenden erhalten eine Erfolgsmeldung.

Nachbedingung Fehlschlag: Die Reservierung wird nicht storniert und die Nutzenden erhalten eine Fehlermeldung.

Auslösendes Ereignis: Die Nutzenden wählen die Option zur Stornierung einer bestehenden Reservierung.

Beschreibung:

1. Die Nutzenden öffnen die Seite mit ihren aktuellen Reservierungen.
2. Die Nutzenden wählen die Reservierung aus, die sie stornieren möchten.
3. Die Nutzenden bestätigen die Stornierung der Reservierung (z.B. durch Klick auf *Reservierung stornieren*).
4. Eine Bestätigungsabfrage erscheint, in der die Nutzenden die Stornierung bestätigen müssen.
5. Falls die Nutzenden bestätigen, wird die Reservierung storniert.
6. Das System aktualisiert die Raumverfügbarkeit und zeigt den Nutzenden eine Bestätigungsmeldung (z.B. *Reservierung erfolgreich storniert*).
7. Die Nutzenden erhalten ggf. eine E-Mail oder eine Benachrichtigung über die Stornierung (optional).

Anmelden mit Adminkonto ⟨F100⟩

Anwendungsfall: Anmelden mit Adminkonto

Anforderung: ⟨MK17⟩

Ziel: Berechtigte Personen können sich mit Passwort authentifizieren um auf das Adminkonto zugreifen zu können.

Vorbedingung: Die nutzende Person befindet sich unangemeldet auf der Startseite.

Nachbedingung Erfolg: Der Admin ist angemeldet und auf administrative Funktionen zugreifen.

Nachbedingung Fehlschlag: Die nutzende Person bleibt unangemeldet auf der Startseite.

Auslösendes Ereignis: Die nutzende Person ist unangemeldet auf der Startseite und versucht sich als Admin anzumelden.

Beschreibung:

1. Die nutzende Person wählt die Admin-Anmeldemethode aus.
2. Die Person gibt das Admin passwort ein.
3. Die Anwendung überprüft die Zugangsdaten und lässt die anmeldung zu oder lehnt ab.
4. Der Admin ist eingeloggt und zeigt die Startseite an.

Deaktivieren von Gastkonto-Anmeldung ⟨F110⟩

Anwendungsfall: Deaktivieren von Gastkonto-Anmeldung

Anforderung: ⟨MK18⟩

Ziel: Die Anmeldung von Gastkonten wird deaktiviert.

Vorbedingung: Ein/e Administrator*in ist angemeldet und die Ansicht *Kontoliste* geöffnet.

Nachbedingung Erfolg: Gastkonten können sich nicht mehr anmelden oder registrieren.

Nachbedingung Fehlschlag: Gastkonten können sich immer noch anmelden, als auch registrieren.

Auslösendes Ereignis: Die als Admin angemeldete Person wählt die Schaltfläche *Gastkonto-Anmeldung deaktivieren*.

Beschreibung:

1. Die Person wählt die Schaltfläche *Gastkonto-Anmeldung deaktivieren* aus.
2. Die Gastkonto-Anmeldung ist nun deaktiviert.

Öffnungszeiten einstellen ⟨F120⟩

Anwendungsfall: Öffnungszeiten einstellen

Anforderung: ⟨MK21⟩

Ziel: Ein/e Administrator*in passt die Öffnungszeiten an.

Vorbedingung: Der/die Administrator*in ist angemeldet und die Ansicht *Kalender* geöffnet.

Nachbedingung Erfolg: Die Öffnungszeiten sind geändert.

Nachbedingung Fehlschlag: Die Öffnungszeiten bleiben gleich.

Auslösendes Ereignis: Die als Admin angemeldete Person wählt die Schaltfläche zum Ändern der Öffnungszeiten.

Beschreibung:

1. Die als Admin eingeloggte Person ändert die Öffnungszeiten (bzw. Schließzeiten) für einen gewählten Wochentag.

Terminkonfliktauflösung ⟨F130⟩

Anwendungsfall: Terminkonfliktauflösung

Anforderung: ⟨MK14⟩

Ziel: Die Anwendung löst Terminkonflikte automatisch.

Vorbedingung: Durch die Überlappung von einem neu gebuchten Termin entsteht ein Konflikt mit einem bereits existierenden Termin.

Nachbedingung Erfolg: Der Konflikt ist gelöst.

Nachbedingung Fehlschlag: Der Konflikt ist nicht gelöst und der neue Termin kann nicht hinzugefügt werden.

Auslösendes Ereignis: Ein neuer Termin wurde erstellt und überlappt mit einem weiteren.

Beschreibung:

1. Nachdem ein Termin erstellt worden ist, wird überprüft ob dieser überlappungen mit anderen Terminen hat.
2. Ist dies der Fall wird überprüft, ob diese Kriterien ein Überlappen der Termine erlauben.
3. Dabei wird ggf. den betroffenen Personen welche einen Termin *Auf Anfrage* gestellt haben per E-Mail die Entscheidung zum Ablehnen oder Annehmen gegeben.
4. Der Termin wird als gültig oder nicht klassifiziert und die betroffenen Personen benachrichtigt.

5 Produktdaten

Die Anwendung verwendet den Server als zentralen Speicherort für alle Daten. Die Daten werden in einer PostgreSQL-Datenbank gespeichert. Auf dem Client werden nur temporäre Daten gespeichert, die für die Funktionalität der Anwendung notwendig sind.

Clientdaten

- Anmeldecookie (falls die Anmeldung via Gastkonto ist). Diese laufen nach 30 Tagen ab.
- Zustand der Anwendung (z.B. aktuelle Seite, geöffnete Dialoge)

Serverdaten

- Daten der Nutzenden (dauerhaft gespeichert)
 - Kontoname (oder *Gast* für Gastkonto)
 - E-Mail-Adresse
 - OAuth- oder Cookie-Token
 - Blockierungsstatus (wahr oder falsch)
- Ereignisdaten (30 Tage nach dem Termin werden diese Daten anonymisiert)
 - Start- und Endzeitpunkt (Datenbank native Zeitdarstellung)
 - Beschreibung
 - Raum
 - Ersteller*in
 - Priorität
 - Kollaborationsmodus
 - E-Mail-Adresse

- Raumdaten (dauerhaft gespeichert)
 - Raumname
 - Öffnungszeiten

6 Nichtfunktionale Anforderungen

⟨NF1⟩ Die Anwendung soll schnell und einfach bedienbar sein.

⟨NF2⟩ Die Anwendung soll auf mobilen sowie Desktop-Endgeräten ohne Einschränkungen nutzbar sein.

⟨NF3⟩ Die Anwendung soll in allen gängigen Browsern lauffähig sein. Insbesondere beinhaltet dies Firefox und Chrome auf Desktop- und Android-Geräten und Safari auf iOS.

⟨NF4⟩ Die Anonymität der Nutzenden soll so weit wie möglich gewährleistet werden.

⟨NF5⟩ Die Anwendung soll in verschiedenen Sprachen dargestellt werden können, insbesondere Deutsch und Englisch.

⟨NF6⟩ Für Nutzende mit Sehschwäche soll die Anwendung durch Kompatibilität mit Screenreadern und kontrastreiche Farbgebung gut bedienbar sein.

⟨NF7⟩ Die Anwendung soll so weit wie sinnvoll möglich sich an den WCAG Richtlinien orientieren.

7 Benutzeroberfläche

7.1 Hauptseite und Buchung

Die Startseite der Anwendung ist in 7.1 dargestellt. Auf dieser haben Nutzende die Möglichkeit, dem Raum zu buchen oder auf andere Ansichten zu wechseln. Das Banner in allen Ansichten zeigt den aktuellen Raumstatus an. Insbesondere wird dadurch die Priorität des aktuellen Termins mithilfe der Farbe des Banners angezeigt.

Rechts in dieser Ansicht befindet sich ein Kalender, der die aktuelle Woche und die Termine in diesem Zeitraum anzeigt. Die Termine sind farblich nach Priorität gekennzeichnet, weitere Informationen können durch Klicken auf den Termin eingesehen werden. Neben dieser Möglichkeit, den Raum zu buchen, gibt es je nach Anmeldungsstatus einen Anmelde- oder Abmeldebutton und einen separaten Buchungsbutton, für die die Verwendung des Kalenders Probleme bereitet. Falls zur Zeit der Verwendung ein Termin des Nutzenden stattfindet, wird ein Quick-Checkout Button angezeigt, der es ermöglicht, den Raum schnell freizugeben. Siehe dazu auch 7.4.

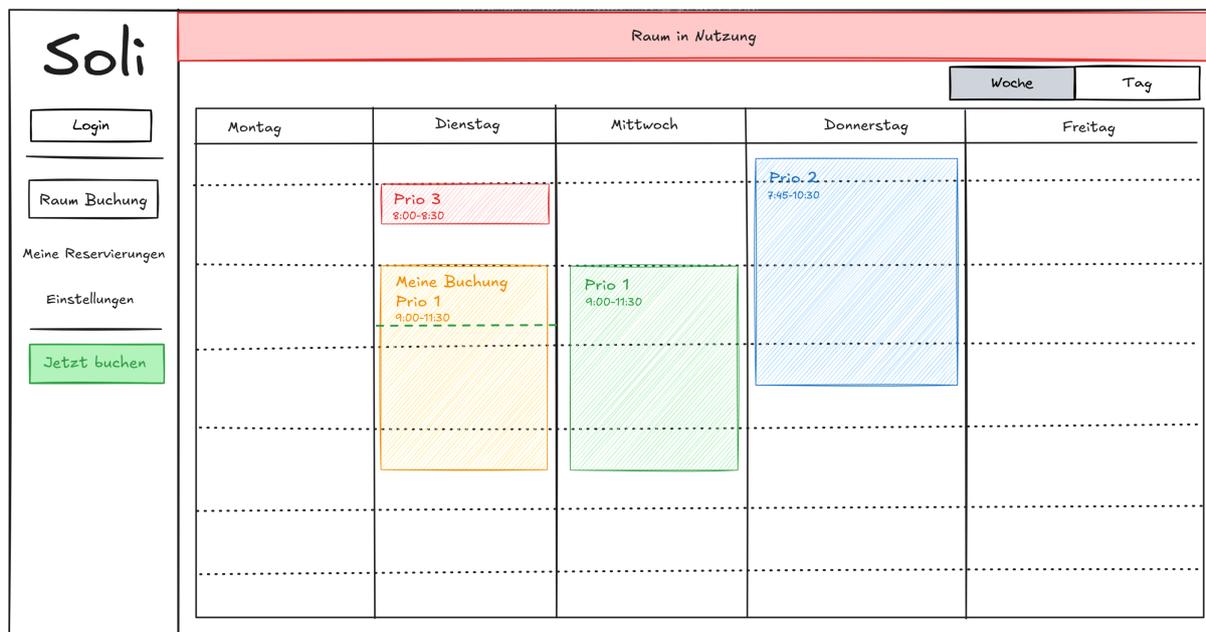


Abbildung 7.1: Startseite der Anwendung

Sollten Nutzende eine Buchung vornehmen wollen, so klicken diese in den gewünschten Zeitraum und es wird der Dialog in 7.2 dargestellt.

Der Dialog bietet Nutzenden die Möglichkeit, den genauen Start- und Endzeitpunkt des Termins festzulegen.

Außerdem können Nutzende die Priorität des Termins in Form einer Zahl zwischen 1 und 3 festlegen. Nutzende können auch angeben, ob sie bereit sind, den Raum mit anderen Nutzenden zu teilen. Für diesen Zweck werden ihnen drei Optionen bereitgestellt: *Ja*, *Nein* und *Auf Anfrage*. Siehe dazu auch 7.2.

Letztlich können Nutzende eine Beschreibung für den Termin hinterlegen, die anderen angemeldeten Nutzenden angezeigt wird.

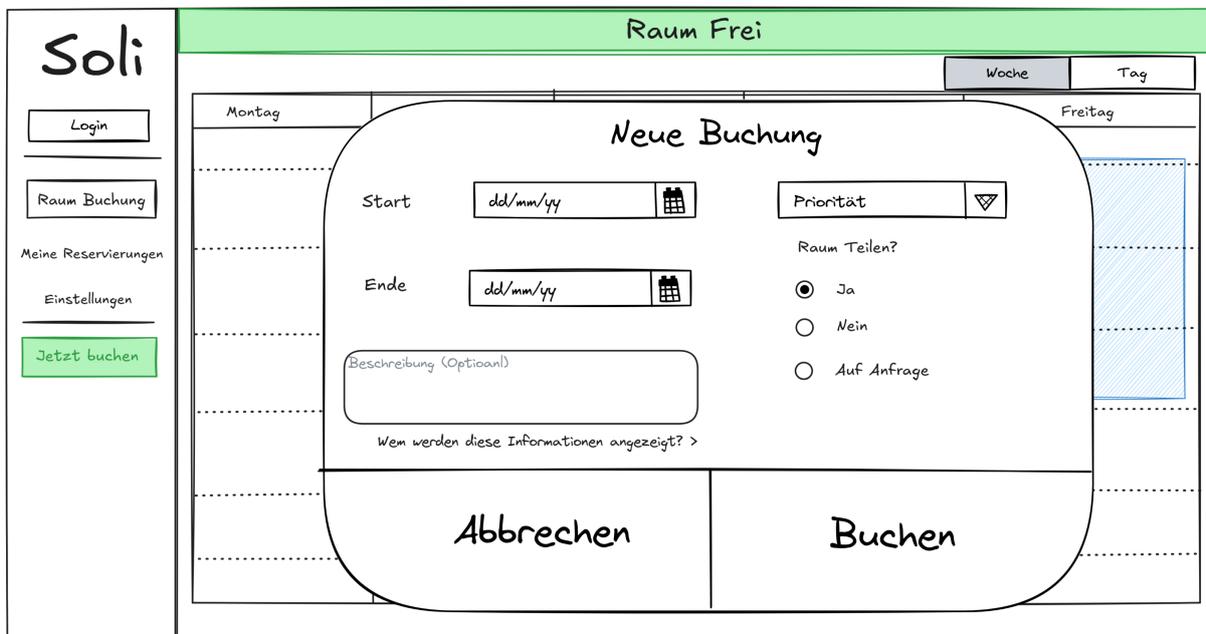


Abbildung 7.2: Termin-erstellen

Tätigen Nutzende eine Buchung, so werden diese aufgefordert, sich anzumelden. Der hierzu gehörige Dialog ist in 7.3 dargestellt. Alternativ ist dieser Dialog auch über den Anmeldungsbutton, der in Abbildung 7.1 zu sehen ist, erreichbar.

In diesem Dialog wird Nutzenden die Möglichkeit gegeben, sich mit ihrem KIT-Konto, mit einem lokalen Gastkonto oder als Admin anzumelden.

Falls Nutzende die Anmeldung per KIT-Konto wählen, werden sie auf die KIT-Login-Seite weitergeleitet. Von dort aus können sie sich mit ihren KIT-Zugangsdaten anmelden.

Falls Nutzende die Anmeldung per Gastkonto wählen, werden sie aufgefordert, eine E-Mail-Adresse anzugeben. Mit der Bestätigung dieser E-Mail-Adresse wird ein temporäres Gastkonto erstellt und die Anmeldung per Cookie gespeichert.

Falls Nutzende die Anmeldung als Admin wählen, werden sie aufgefordert das Passwort des Adminkontos einzugeben.

Nach einer erfolgreichen Anmeldung werden Nutzende auf die nächste Seite weitergeleitet.

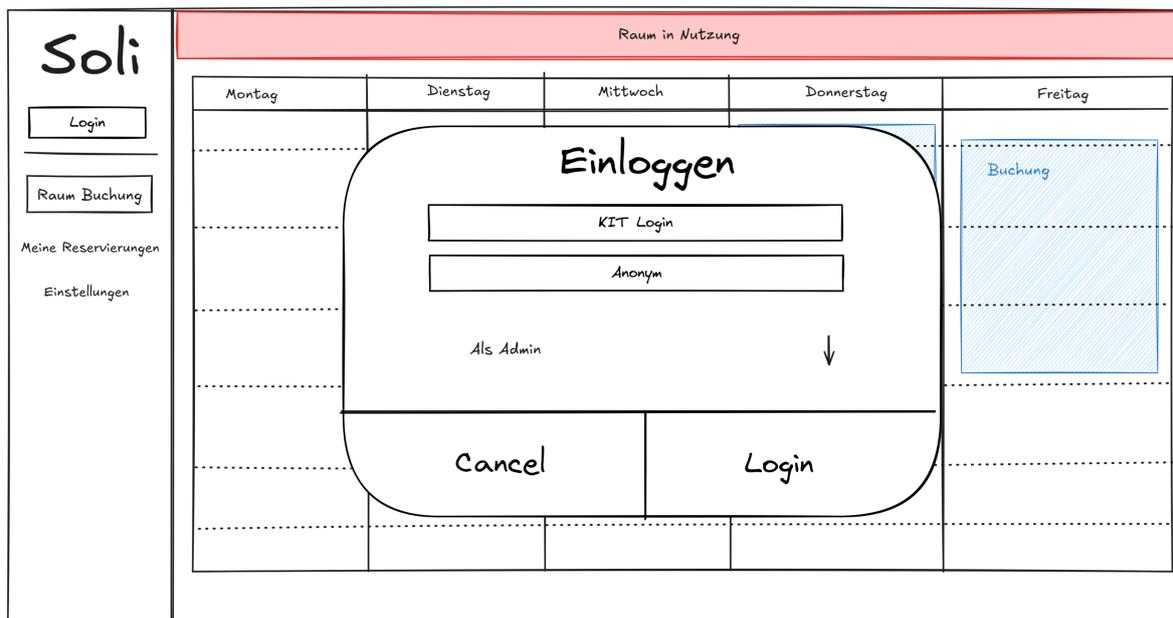


Abbildung 7.3: Anmeldungsseite

Sind Nutzende eingeloggt und belegen den Raum, so wird ihnen die in Abbildung 7.4 dargestellte Ansicht angezeigt. Hier können Nutzende den Raum wieder über den Quick-Checkout Button freigeben.

Ziel dieser Ansicht ist es, Nutzenden das frühe Freigeben des Raumes ohne unnötigen Mehraufwand zu ermöglichen.

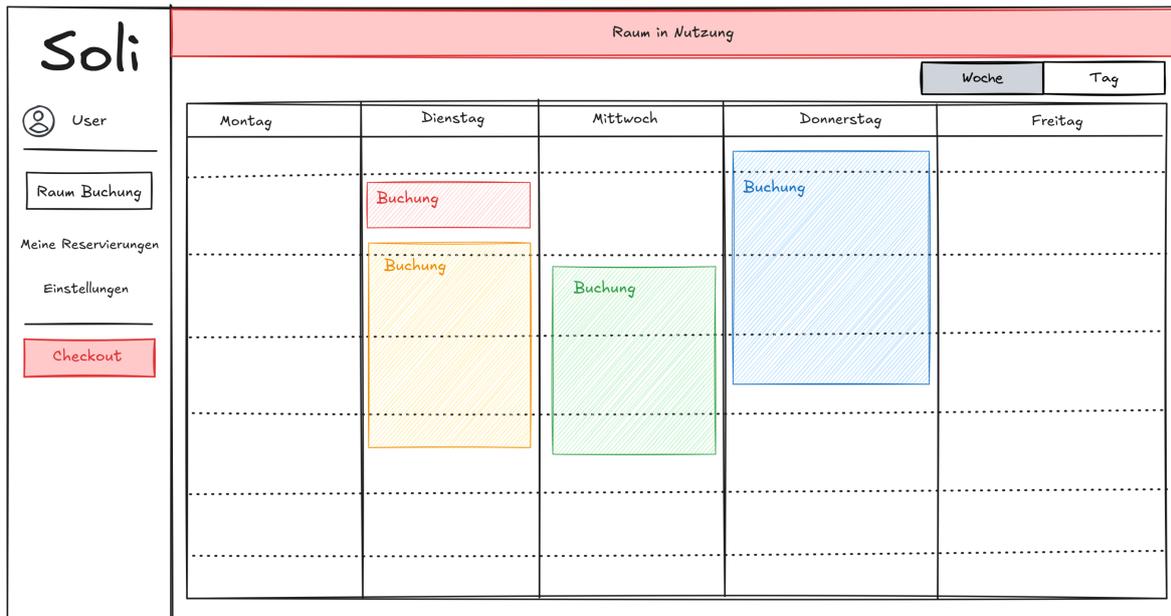


Abbildung 7.4: Quick Checkout

7.2 Terminübersicht

Nutzende, die eine Buchung vorgenommen haben, können diese in der Terminübersicht, die in Abbildung 7.5 dargestellt ist, einsehen und verwalten.

Die Termine in dieser Ansicht werden, anders als in der Ansicht *Kalender*, in Form einer sortierten Liste dargestellt. Dabei werden die Start- und Endzeitpunkte der Termine, wie auch die Priorität und die Beschreibung angezeigt.

Nutzende haben die Möglichkeit, eine Buchung zu stornieren, indem sie auf den Stornieren-Button klicken.

Außerdem können Nutzende die Ansicht *Termin* (7.6) zu einem Termin öffnen, indem sie auf den Termin klicken.

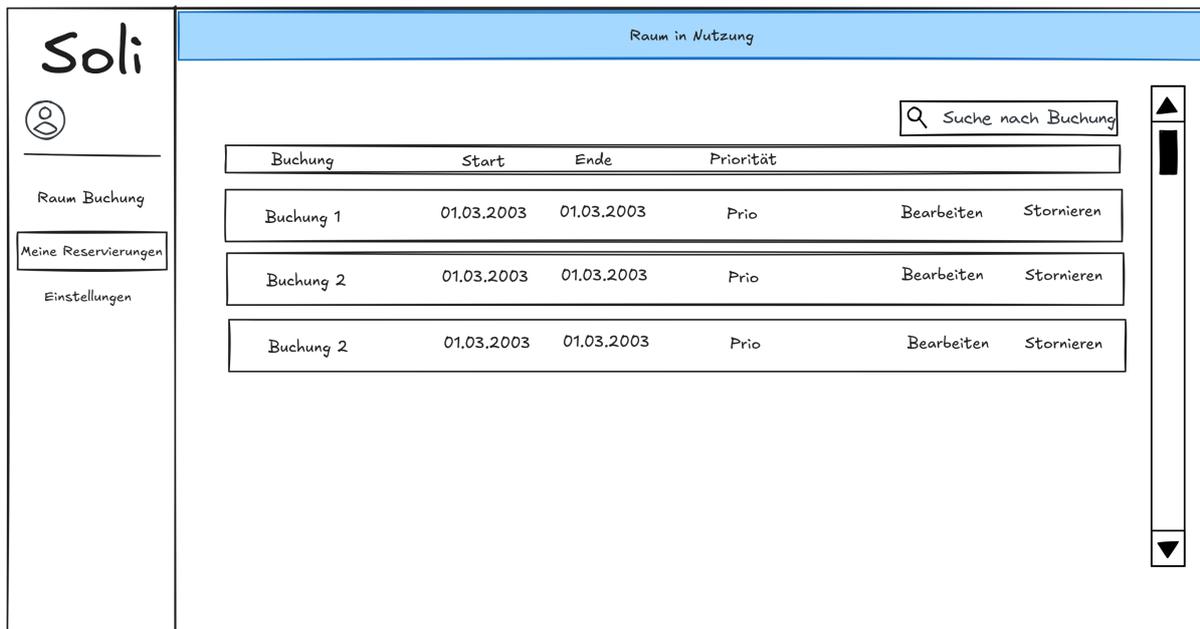


Abbildung 7.5: Reservierungsübersicht

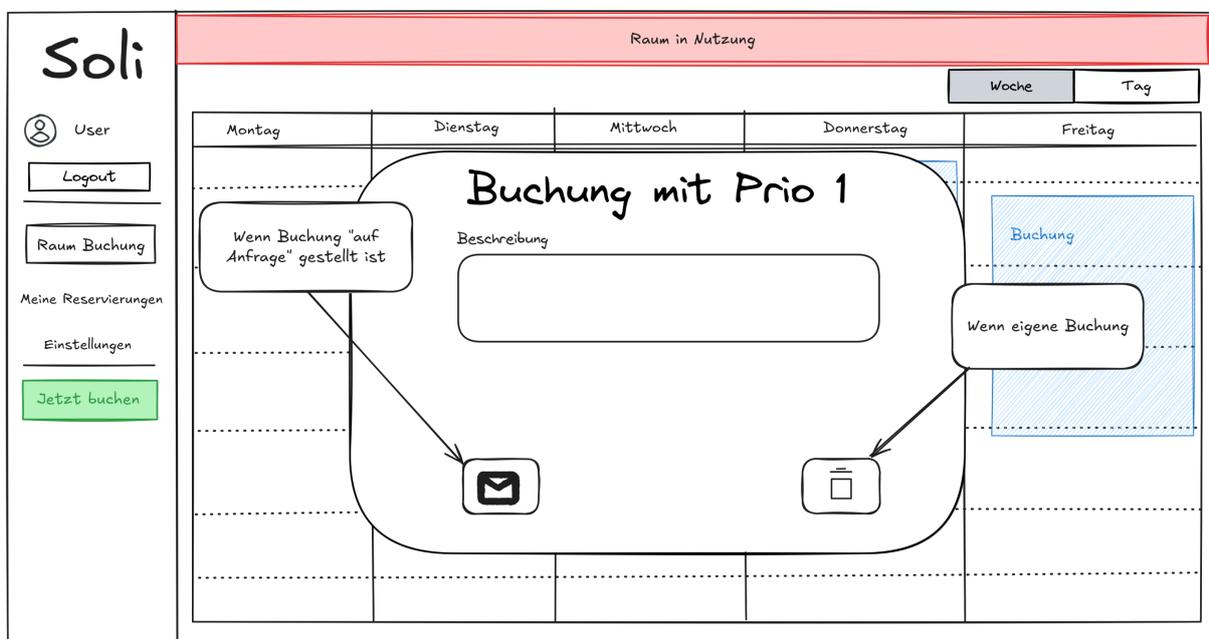


Abbildung 7.6: Reserverierung im Kalender

7.3 Administration

Ein/e Administrator*in hat die Möglichkeit, über die Benutzeradministrationsoberfläche, die in Abbildung 7.7 dargestellt ist, Nutzende einzusehen und zu verwalten.

In dieser Ansicht werden alle Nutzenden angezeigt, die sich in der Anwendung registriert haben. Ein Admin kann die Accounts von Nutzenden deaktivieren und somit ihre Anmeldung verhindern.

Um die Suche nach einem bestimmten Konten zu erleichtern, kann ein Admin nach Kontonamen filtern.

Außerdem kann ein Admin die Anmeldung per Gastkonto deaktivieren. Zweck dieser Funktion ist es, den Missbrauch von Gastkonten z.B. durch Bots in vorübergehenden Zeiten zu verhindern.

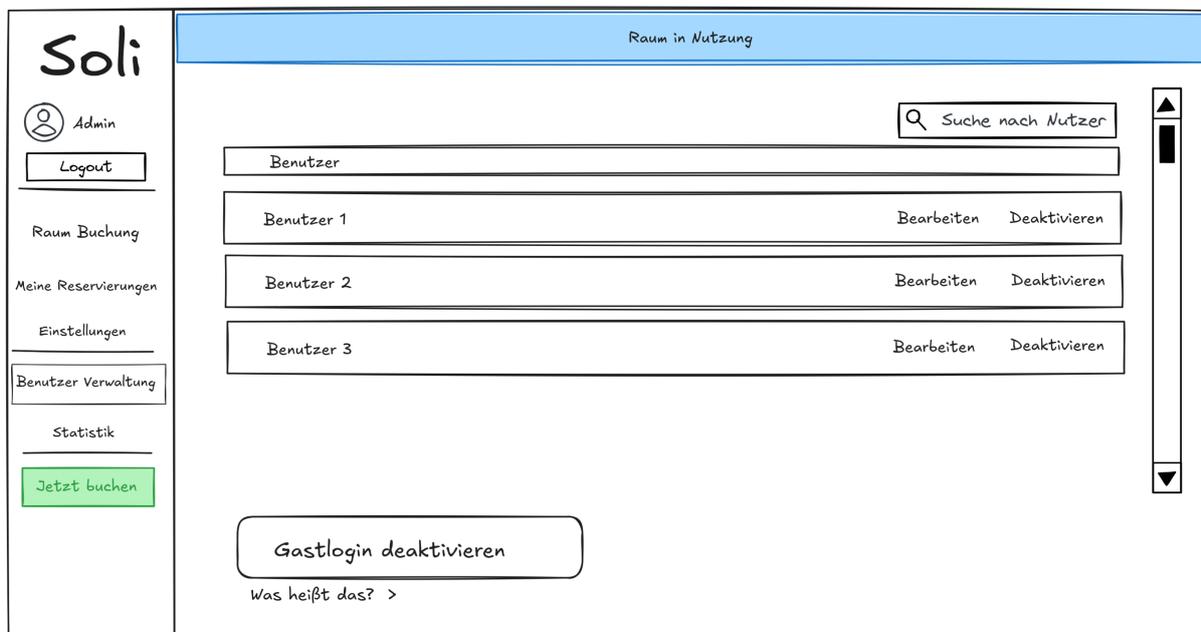


Abbildung 7.7: Benutzeradministrationsoberfläche

8 Technische Produktumgebung

8.1 Entwicklungsumgebung

- Devcontainer für reproduzierbare Entwicklungsumgebung
- IntelliJ IDEA als IDE und LaTeX-Editor
- pdfLaTeX zur Kompilierung des Dokuments
- Git zur Versionskontrolle
- Gradle zur Build-Automatisierung
- GitHub Actions für CI/CD (inkl. Tests und Dokumentation)
- GitHub Packages zur Speicherung generierter Docker-Images

8.2 Hardware

8.2.1 Backend

- AMD64-kompatibler Prozessor (emuliert in einer VM)
- mindestens 1 GB RAM

8.2.2 Frontend

- Beliebiges (mobiles) Gerät mit Internetzugang

8.3 Software

8.3.1 Backend

- Ubuntu Linux 24.04 LTS als Betriebssystem
- Docker zur Containerisierung der Anwendung
- PostgreSQL als Datenbank (in einem eigenen Container)
- Applikations-Container (gebaut in CI/CD)
- Spring Boot mit Java 21 als Backend-Framework
- JTE zur Generierung von HTML-Seiten
- Caddy als HTTPS-Reverse-Proxy

8.3.2 Frontend

- Pure HTML mit JavaScript zur Interaktivität um maximale Kompatibilität zu gewährleisten
- Vorgefertigtes CSS-Framework für konsistentes Design
- FullCalendar für den Kalender
- Moderner Browser (Chrome, Firefox, Safari, Edge) zur vollen Funktionalität

8.4 Schnittstellen

- SSR generiert (hauptsächlich) statische HTML-Seiten
- HTML-Forms zur Interaktion mit dem Backend
- Minimale REST-API zum Prüfen auf Verfügbarkeit

9 Testfälle und Testeszenarien

In diesem Kapitel definieren wir die Testfälle und Testfallszenarien.

9.1 Testfälle

Nr.	Beschreibung	Funktion
<T10>	Landeseite besuchen	<F10>
<T20>	Login	<F20>
<T30>	Abmelden	<F30>
<T40>	Reservieren	<F40>
<T50>	Löschen von Terminen durch Administratoren	<F50>
<T60>	Deaktivieren eines Kontos	<F60>
<T70>	Benachrichtigung bei freiem Raum	<F70>
<T80>	Anzeige des Raumstatus	<F80>
<T90>	Stornierung einer Reservierung	<F90>
<T100>	Login mit Adminkonto	<F100>
<T110>	Deaktivierung von Gastkonten	<F110>
<T120>	Öffnungszeiten einstellen	<F120>
<T130>	Terminkonfliktauflösung	<F130>

Tabelle 9.1: Überblick der Testfälle.

9.2 Testfallszenarien

Szenario: Besuch der Landeseite und Anmeldung/Abmeldung⟨S10⟩

Ziel: Sicherstellen, dass Nutzende die Landeseite aufrufen und sich erfolgreich anmelden bzw. abmelden können.

1. Der Nutzende besucht die Landeseite ⟨T10⟩.
2. Der Nutzende loggt sich ein ⟨T20⟩.
3. Der Nutzende meldet sich ab ⟨T30⟩.

Szenario: Reservierung eines Raums und Stornierung⟨S20⟩

Ziel: Überprüfen, ob Nutzende erfolgreich Termine reservieren können.

1. Der Nutzende besucht die Landeseite ⟨T10⟩.
2. Der Nutzende meldet sich an ⟨T20⟩.
3. Der Nutzende wählt einen verfügbaren Raum aus und reserviert diesen ⟨T40⟩.
4. Buchen ausserhalb der Öffnungszeiten sollte hierbei fehlschlagen.
5. Überprüfen, ob der Raumstatus korrekt angezeigt wird und sich ggf. geändert hat ⟨T80⟩.
6. Der Nutzende storniert die Reservierung ⟨T90⟩.
7. Andere Nutzende, welchen sich diesen Termin vorgemerkt hatten, werden benachrichtigt, dass der Raum nun Frei ist ⟨T70⟩.

Szenario: Terminverwaltung durch das Adminkonto⟨S30⟩

Ziel: Testen der administrativen Funktionalitäten zum Löschen von Terminen.

1. Ein Admin besucht die Landeseite ⟨T10⟩.
2. Der Admin meldet sich an ⟨T100⟩.
3. Der Admin löscht bestehende Termine ⟨T50⟩.
4. Der Admin meldet sich ab ⟨T30⟩.

Szenario: Kontoverwaltung durch das Adminkonto⟨S40⟩

Ziel: Testen der administrativen Funktionalitäten zum Deaktivieren von Gastkonten und einstellen der Öffnungszeiten.

1. Ein Admin besucht die Landeseite ⟨T10⟩.
2. Der Admin meldet sich an ⟨T100⟩.
3. Der Admin deaktiviert die Anmeldung von Gastkonten ⟨T110⟩.
4. Der Admin ändert die Öffnungszeiten für einen bestimmten Wochentag ⟨T120⟩.
5. Der Admin öffnet die Ansicht *Kontoliste* und deaktiviert ein Konto ⟨T60⟩.
6. Der Admin meldet sich ab ⟨T30⟩.
7. Das Anmelden mit einem Gastkonto sollte jetzt fehlschlagen ⟨T20⟩.
8. Das Anmelden mit dem deaktiviertem Konto sollte ebenfalls fehlschlagen ⟨T20⟩.
9. Die Ansicht *Kalender* sollte die neuen Öffnungszeiten darstellen.

Szenario: Terminkonflikt auflösung⟨S50⟩

Ziel: Überprüfen, ob ein Terminkonflikt richtig aufgelöst wird.

1. Konto 1 meldet sich an ⟨T20⟩.
2. Konto 1 erstellt einen Termin, und gibt die zu testende Priorität, Raumteilungsoption und Zeitperiode ein ⟨T20⟩.
3. Konto 1 meldet sich ab und Konto 2 meldet sich an.
4. Konto 2 erstellt einen Termin, welcher den anderen überlappt.
5. Es wird überprüft, ob die erwartete Konfliktauflösung stattgefunden hat und ggf. die dafür benötigten E-Mails versendet wurden ⟨T130⟩.

Glossar

AMD64 Verbreitete Prozessorarchitektur von Intel und AMD. 39

API Schnittstelle auf Quelltext-Ebene um anderen Programmen funktionen zur Verfügung zu stellen. 7, 40

Browser Software zum Navigieren von Webseiten, zum Beispiel Firefox oder Chrome. 16, 31, 40

CI/CD Baut automatisiert Software, führt Tests durch und veröffentlicht Artefakte. 39, 40

Container Isolierte Umgebung um Software unabhängig von der zugrunde liegenden Umgebung auszuführen. 8, 40

CSS Cascading Style Sheets, ist eine Sprache um das Visuelle aussehen einer Website zu definieren. 40

Devcontainer Container, welcher eine Entwicklungsumgebung bereitstellt. 39

Docker Software zur Bereitstellung von Anwendungen innerhalb von Containern. 8, 39, 40

Git Software zur Versionsverwaltung von Softwareprojekten. 39

GitHub Plattform zur Versionsverwaltung von Softwareprojekten, nutzt Git. 39

Gradle Build-Management-Tool welches auf Java basiert. 39

HTML (Hypertext Markup Language) Eine Sprache, um die Struktur und den Inhalt einer Website zu definieren. 40

iCal Dateiformat zur Speicherung von Kalenderdaten. 6

IDE (Integrated Development Environment) Software, welche alle Werkzeuge zur Softwareentwicklung in einem Programm kombiniert. 39

JavaScript Programmiersprache um das Logische Verhalten von Webseiten zu steuern. 40

OIDC (OpenID Connect) Authorisierungsframework welches vom KIT genutzt wird um dritten Webseiten Logins basierend auf KIT-Konten bereitzustellen. 5

PostgreSQL Objekt-Relationales Datenbankmanagementsystem, welches zum Speichern und Verwalten von Daten verwendet wird. 29, 40

RAM (Random Access Memory) Arbeitsspeicher. 8, 39

REST (Representational State Transfer) Architekturstil von APIs für das Internet. 40

SSR (Server-Side Rendering) Methode, um das HTML einer Website auf dem Server zu produzieren, statt im Browser. 40

UI (User Interface) Typischerweise visuelle Oberfläche, mit welcher der Nutzende interagiert. 6

VM (Virtuelle Maschine) Software zur Simulation eines Computers. 39

WCAG (Web Content Accessibility Guidelines) Standard zur Barrierefreien Gestaltung von Webseiten.. 31