

PRAXIS DER SOFTWAREENTWICKLUNG

SOLIDARISCHE RAUMNUTZUNG

Praxis der Softwareentwicklung (PSE)
Wintersemester 2024/25

Implementierungsbericht

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	Kriterien	6
2.1	Musskriterien	6
2.2	Musskriterien Implementierung	7
2.3	Wunschkriterien	8
2.4	Wunschkriterien Implementierung	9
3	Datenmodell	10
3.1	ER-Diagramm	10
4	Tests und Coverage	13
4.1	Coverage	14
5	Aktivität	15
6	View	17
6.1	Hauptseite und Buchung	17
6.2	Terminübersicht	25
6.3	Adminstration	28
	Glossar	32

Abbildungsverzeichnis

3.1	Legende: ER-Diagramm	10
3.2	ER-Diagramm	11
3.3	Neues ER-Diagramm	12
5.1	Verteilung der Aufgaben	15
5.2	Commits pro Stunde und Wochentag	16
6.1	Startseite der Anwendung	17
6.2	Implementierung: Startseite der Anwendung	18
6.3	Implementierung: Startseite der Anwendung im Lightmode	18
6.4	Termin-erstellen	19
6.5	Implementierung: Termin-erstellen	20
6.6	Implementierung: Termin-erstellen	20
6.7	Anmeldungsseite	21
6.8	Implementierung: Anmeldungsseite	22
6.9	Quick Checkout	23
6.10	Implementierung: Quick Checkout	24
6.11	Reservierungsübersicht	25
6.12	Implementierung: Reservierungsübersicht	26
6.13	Reserverierung im Kalender	27
6.14	Implementierung: Reserverierung im Kalender	27
6.15	Benutzeradministrationsoberfläche	28
6.16	Implementierung: Benutzeradministrationsoberfläche	29
6.17	Implementierung: Konfiguration	30
6.18	Implementierung: Visualisierung der Öffnungszeiten	30
6.19	Implementierung: Statistikansicht	31

Notes

1 Einleitung

Das Ziel dieses Projekts ist es, eine Webanwendung *Soli* zur solidarischen Raumnutzung zu entwickeln. Die Website soll das Buchen von Räumen ermöglichen, wobei der Fokus darauf liegt, Buchungen mit differenzierten Prioritäten zu ermöglichen.

Gegenstand dieses Implementationsberichts ist die Beschreibung der Implementierung der Anwendung. Dabei liegt der Fokus auf den Unterschieden zwischen dem System nach der Implementierung und der Systembeschreibung des Entwurfs- und Pflichtenhefts.

2 Kriterien

In diesem Kapitel wird auf die Kriterien eingegangen, die im Pflichtenheft spezifiziert wurden. Für jedes Kriterium wird angegeben, ob es im Rahmen der Implementierung umgesetzt wurde, beziehungsweise weshalb nicht. Wurde ein Kriterium abweichend vom Pflichten- und Entwurfsheft implementiert, werden hier diese Anpassungen erläutert und aufgeführt, weshalb die Änderungen notwendig waren.

2.1 Musskriterien

Folgende Musskriterien wurden im Pflichtenheft aufgeführt:

⟨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 Musskriterien Implementierung

Die Funktionalität aller Musskriterien wurde in der Implementierung umgesetzt. Es gibt allerdings kleine Anpassungen, die im Folgenden beschrieben werden.

⟨MK1⟩ Umgesetzt.

⟨MK2⟩ Umgesetzt.

⟨MK3⟩ Umgesetzt.

⟨MK4⟩ Umgesetzt.

⟨MK5⟩ Umgesetzt.

⟨MK6⟩ Umgesetzt.

⟨MK7⟩ Umgesetzt. Mit einem Badge auf den Terminen werden die eigenen Termine hervorgehoben.

⟨MK8⟩ Umgesetzt. In der Ansicht *Termin* werden Start- und Endzeit, Priorität, Kollaborativität, sowie die Beschreibung angezeigt.

⟨MK9⟩ Umgesetzt.

⟨MK10⟩ Umgesetzt. Ein Paragraph informiert über die Sichtbarkeit der Buchungsbeschreibung.

⟨MK11⟩ Umgesetzt. Die Ansicht *Terminübersicht* zeigt alle Termine des angemeldeten Nutzenden in einer Tabelle an und gibt die Möglichkeit, Termine zu löschen.

⟨MK12⟩ Umgesetzt. Die Prioritäten sind Hoch, Mittel und Niedrig.

⟨MK13⟩ Umgesetzt.

⟨MK14⟩ Umgesetzt. Bei der Terminkonfliktauflösung wird zusätzlich noch nach einer Bestätigung zum Überschreiben anderer Termine gefragt.

⟨MK15⟩ Umgesetzt.

⟨MK16⟩ Umgesetzt. Stornieren eines Termins ist über die Ansicht *Termin* mit dem Button *Löschen* möglich.

⟨MK17⟩ Umgesetzt. Das Admin Password wird per Docker-Compose festgelegt und kann demnach nur durch den Server-Admin geändert werden.

⟨MK18⟩ Umgesetzt.

⟨MK19⟩ Umgesetzt.

⟨MK20⟩ Umgesetzt.

⟨MK21⟩ Umgesetzt. Admins können Öffnungszeiten einstellen, für diese Möglichkeit gibt es eine neue Ansicht.

2.3 Wunschkriterien

Folgende Wunschkriterien wurden im Pflichtenheft aufgeführt:

⟨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.

2.4 Wunschkriterien Implementierung

Einige dieser Wunschkriterien wurden ebenfalls umgesetzt. Im Folgenden wird beschrieben, welche Wunschkriterien umgesetzt wurden und welche nicht.

⟨WK1⟩ Umgesetzt. Die zur Verfügung stehenden Räume können über die Ansicht *Räume* konfiguriert werden. Den Nutzenden wird falls nötig eine Raumauswahl vor der Ansicht *Kalender* präsentiert. Existiert nur ein Raum, wird diese Auswahl übersprungen.

⟨WK2⟩ Umgesetzt. Feiertage werden nur visuell angezeigt und blockieren keine Buchung. Eine URL zu einem iCal Ferienkalender kann in der Docker-Compose Konfiguration hinterlegt werden. Aktuell wird der von Thunderbird (Mozilla) für Deutsche Feiertage verwendet.

⟨WK3⟩ Nicht umgesetzt.

⟨WK4⟩ Umgesetzt. Termine werden in der Ansicht *Termin* im iCal-Format zum Export angeboten. Diese können von Nutzenden reibungslos in persönliche Kalender importiert werden.

⟨WK5⟩ Umgesetzt. Nutzende können in der Ansicht *Termin* die Beschreibung ihrer Termine bearbeiten. Um die Oberfläche einfach zu halten, wurde die Bearbeitung anderer Felder nicht umgesetzt.

⟨WK6⟩ Umgesetzt. Die Ansicht *Kalender* visualisiert, welche Termine bereits in der Vergangenheit liegen und wo der Übergang von der Vergangenheit zur Zukunft liegt. Dafür wird die aktuelle Zeit mit einem roten Strich markiert.

⟨WK7⟩ Umgesetzt. Tooltips erklären, wofür die wichtigsten Elemente der UI verwendet werden. Im Verlauf der QA-Phase sollen diese basierend auf Feedback weiter verbessert werden.

⟨WK9⟩ Nicht umgesetzt.

⟨WK10⟩ Nicht umgesetzt. Die Anwendung informiert Nutzende nicht, wenn ein gewünschter Termin frei wird, da keine überzeugende Kombination aus Use-Case und guter Nutzererfahrung gefunden wurde.

⟨WK11⟩ Umgesetzt. Der Quick-Checkin-Button mit der Benennung "Jetzt Buchen" füllt im Buchungsdialog den Aktuellen Zeitslot bei Nutzung des Buttons bereits aus.

⟨WK12⟩ Umgesetzt. Mit dem Quick-Checkout-Button können Nutzende den aktuellen Termin vorzeitig beenden.

⟨WK13⟩ Umgesetzt. Admins können eine globale Statistik-Ansicht nutzen um die Buchungen pro Wochentag, Tag des Monats und Monat anzuzeigen.

3 Datenmodell

3.1 ER-Diagramm

Einige der Funktionen, welche im Verlauf der Entwicklungsphase umgesetzt wurden, erforderten, beispielsweise etwa zur Speicherung von Öffnungszeiten, Anpassungen am Datenmodell. Entsprechend möchten wir hier ein aktualisiertes ER-Diagramm präsentieren, welches die Änderungen und Erweiterungen des Datenmodells zeigt.

In Abbildung 3.2 finden Sie das alte ER-Diagramm aus dem Entwurfsheft und in Abbildung 3.3 das neue ER-Diagramm.

Zum besseren Verständnis ist in Abbildung 3.1 eine Legende für die Symbole, welche in den Diagrammen verwendet werden.

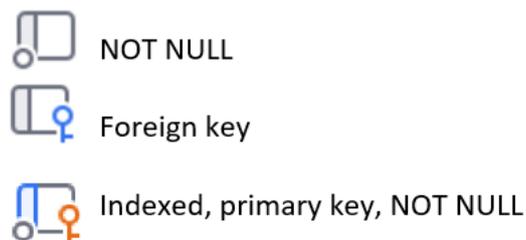


Abbildung 3.1: Legende: ER-Diagramm

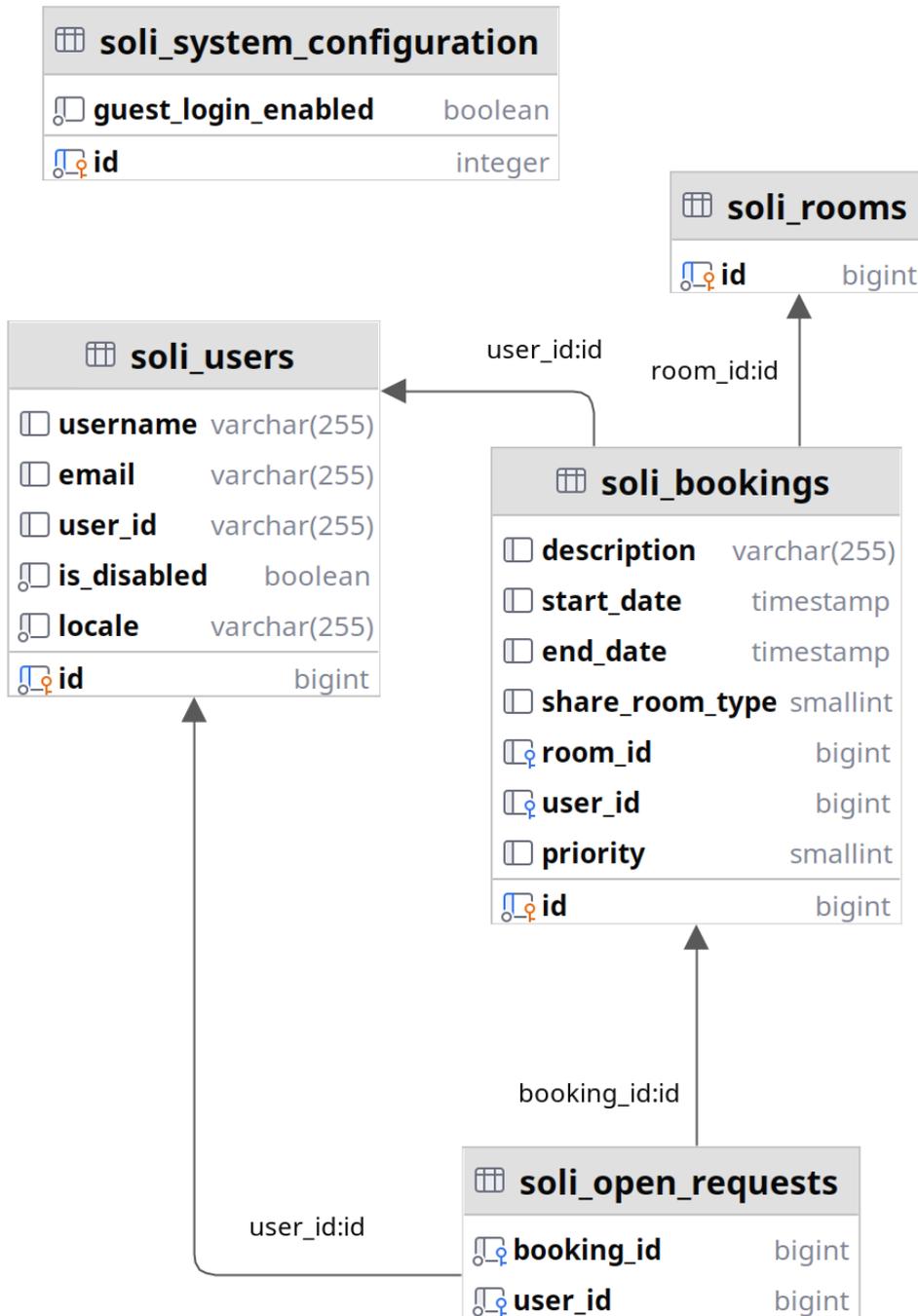


Abbildung 3.2: ER-Diagramm

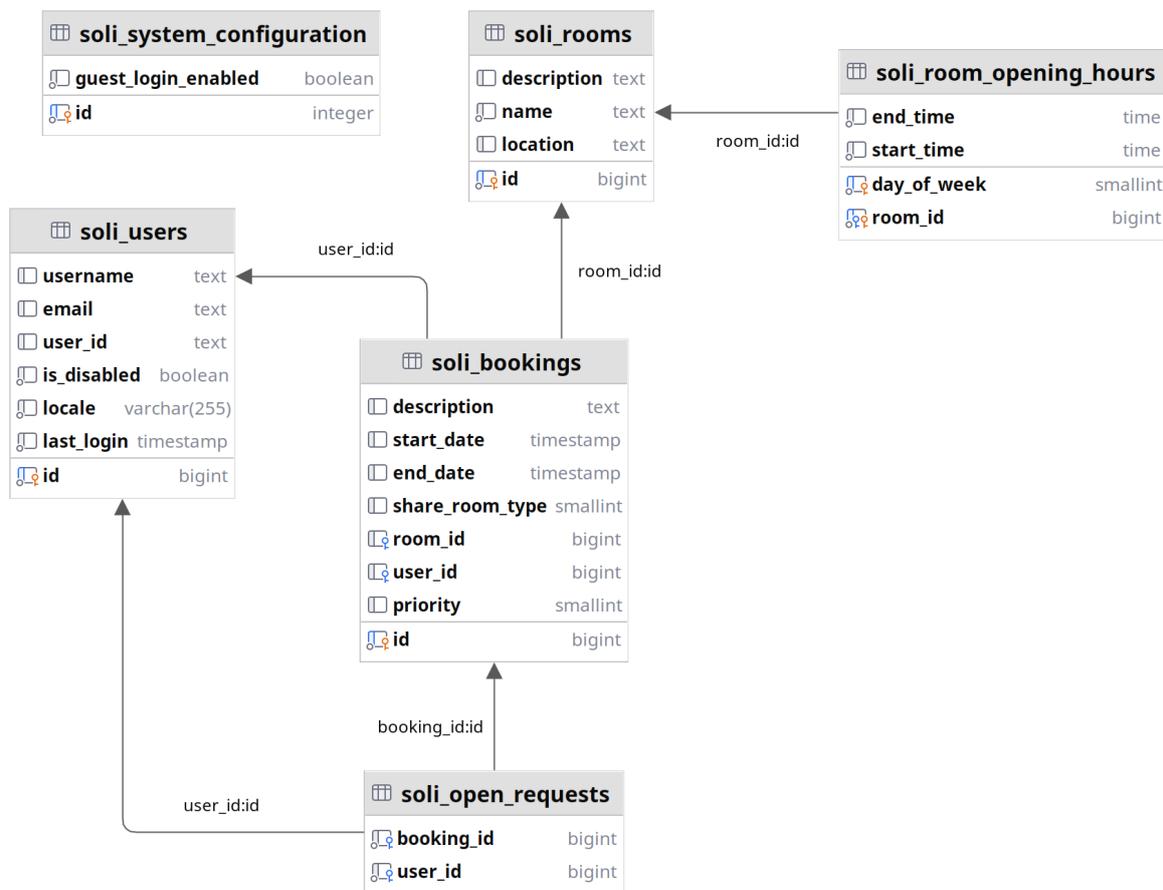


Abbildung 3.3: Neues ER-Diagramm

4 Tests und Coverage

Für kontinuierliches Testen und zur Sicherstellung der Qualität des Codes wurden sowohl Unit- als auch Integrationstests geschrieben. Diese werden nach jedem Push auf das Repository durch GitHub CI automatisiert ausgeführt.

Umgesetzt sind die Tests mit JUnit als Testframework, Mockito für Mocking und werden von Gradle gestartet. Damit sichergestellt werden kann, dass die Tests in einem Realitätsnahen Umfeld laufen, wird eine normale Postgres-Datenbank verwendet, welche in einem von Gradle gestarteten Docker-Container läuft.

Um die Abdeckung unseres Quellcodes durch die Tests sicherzustellen, setzen wir JaCoCo ein. JaCoCo wird ebenfalls von Gradle gestartet und generiert nach jedem Testdurchlauf einen Report, der in GitHub CI angezeigt wird. Bei einer zu niedrigen Testabdeckung wird zudem ein fehlgeschlagener Check in GitHub CI generiert.

Ausgenommen von den Tests ist lediglich der Inhalt des Pakets *config*, da dieser lediglich der Konfiguration von Spring dient und nicht gut durch Unit- oder Integrationstests geprüft werden kann, sowie *view* da dies schwer automatisiert visuell zu überprüfen.

4.1 Coverage

Zum Ende der Implementationsphase haben wir in mehr als 150 Tests eine Coverage von 67% erreicht. Diese gliedert sich wie folgt in die verschiedenen Pakete auf:

Paket	Line Coverage
<i>Controller</i>	69%
<i>Domain</i>	87%
<i>DTO</i>	79%
<i>Filter</i>	100%
<i>Repository</i>	100%
<i>Service</i>	85%
<i>Gesamt</i>	67%

Tabelle 4.1: Coverage der verschiedenen Pakete

5 Aktivität

In Abbildung 5.1 ist die Verteilung der Aufgaben für die Implementierungsphase unter unseren Teammitgliedern zu sehen. Wie zu erkennen ist, wurde mit einzelnen Ausnahmen jedem Teammitglied für jede Woche eine Aufgabe zugeteilt. Da wir vermuteten, dass die Umsetzung der Öffnungszeitenfunktion überdurchschnittlich kompliziert sein könnte, entschieden wir uns, dieser Funktion zwei Wochen zuzuweisen.

Die im Diagramm dargestellte Zeitaufteilung wurde nahezu vollständig eingehalten, die Zeitschätzungen bei der Aufteilung der Funktionen scheint dem tatsächlichen Aufwand also sehr nahe gewesen zu sein.

Verschiedene kleinere Aufgaben (etwa Anpassungen nach Feedback) wurden erst im Laufe der Phase festgestellt und je nach Kapazität der Teammitglieder verteilt, da es sich bei diesen jedoch um eine große Zahl sehr kleiner Änderungen handelt, werden diese hier nicht weiter aufgeführt.

Des Weiteren wurden einige kleinere Darstellungsfehler oder Randfälle bei der Implementierung festgestellt und vorgemerkt, damit sie mithilfe von weiterem Feedback in der Qualitätssicherungsphase behoben werden können.

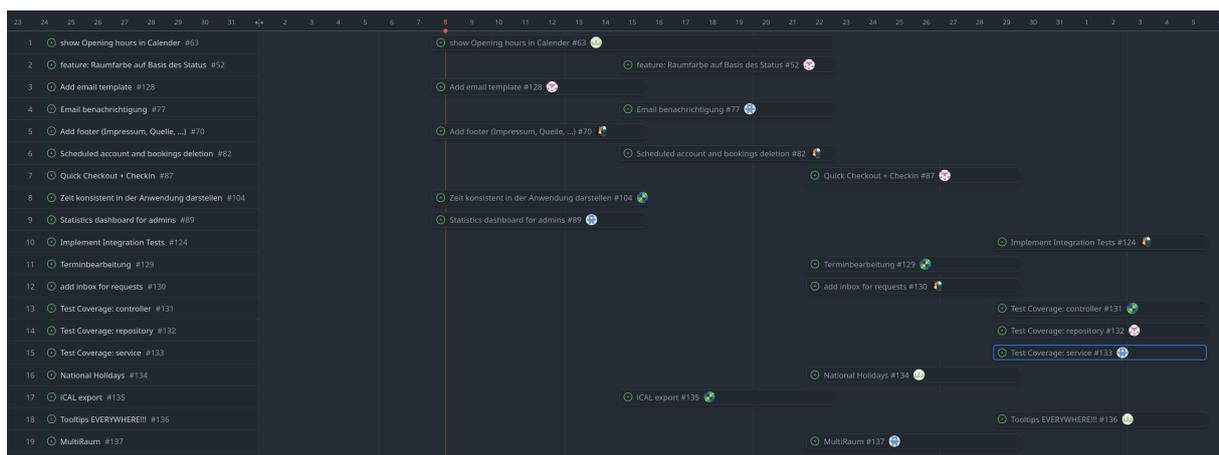


Abbildung 5.1: Verteilung der Aufgaben

In Abbildung 5.2 ist die Verteilung der Commits pro Stunde und Wochentag dargestellt. In dieser ist zu erkennen, dass die Entwicklung der Anwendung größtenteils in den üblichen Arbeitszeiten von 9 bis 17 Uhr verlief. Während an Wochenenden und Feiertagen einzelne Commits

vorkommen, sind diese in der Regel auf kleinere Änderungen beschränkt. Für die Planung des Projekts ist dies ein klarer Erfolg, da die Aufgaben offenbar gut über die Zeit verteilt werden konnten.

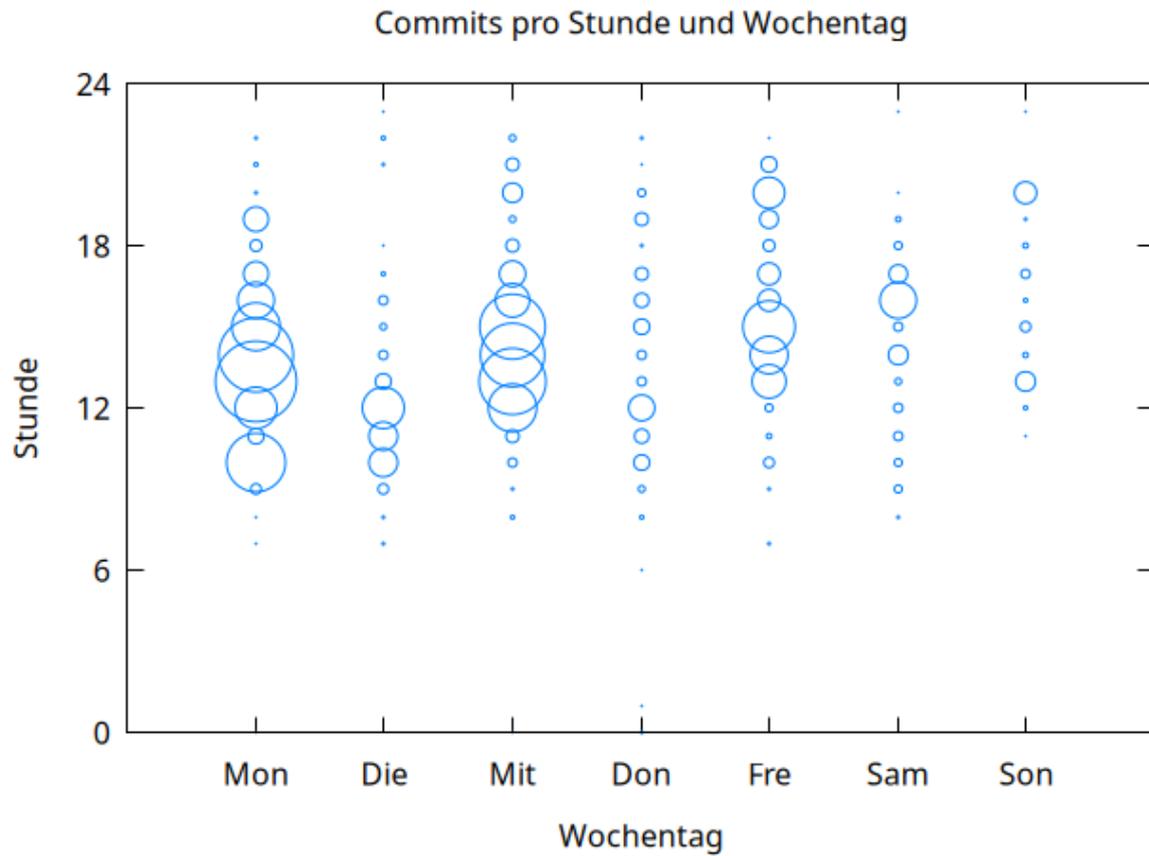


Abbildung 5.2: Commits pro Stunde und Wochentag

6 View

6.1 Hauptseite und Buchung

Das Mockup der Startseite der Anwendung ist in Abbildung 6.1 dargestellt.

Die entsprechende implementierte Startseite wird gezeigt in Abbildung 6.3, dabei ist zu beachten, dass wir neben einen *Light Mode* auch einen *Dark Mode* implementiert haben, welcher beispielhaft für die *Kalender* Ansicht in Abbildung 6.2 zu sehen ist. Im Folgenden werden wir zur besseren Erkennlichkeit im Dokument immer die Bilder der Implementation im *Light Mode* verwenden.

Der Wechsel zwischen den beiden Modi ist über die Browser-Einstellungen möglich.

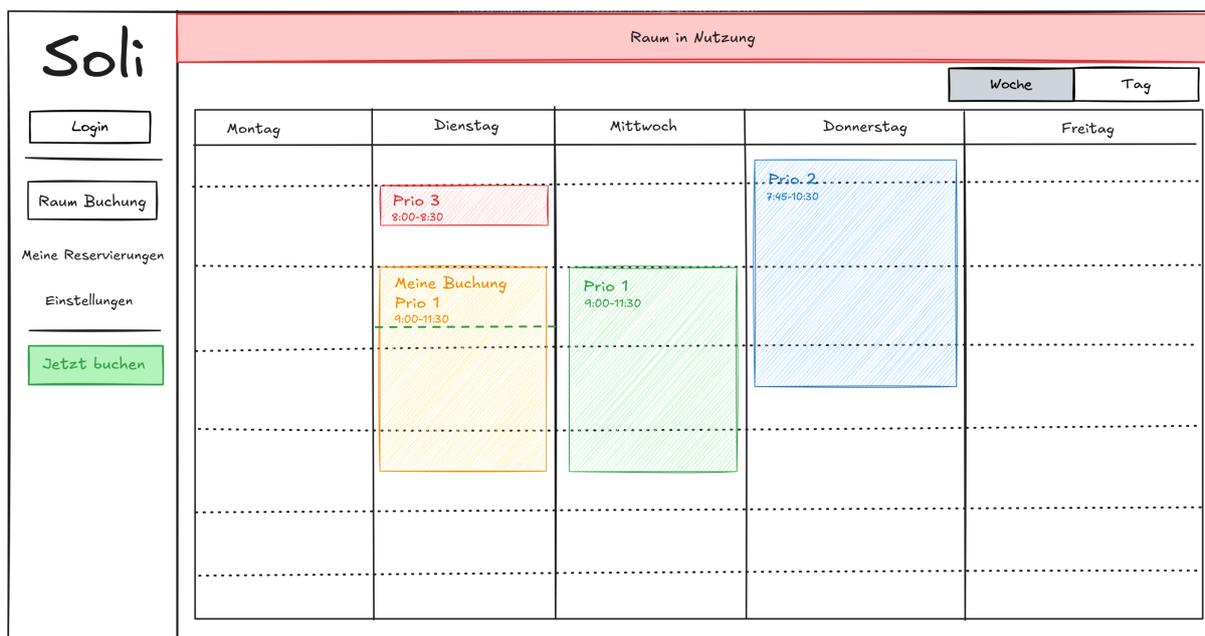


Abbildung 6.1: Startseite der Anwendung



Abbildung 6.2: Implementierung: Startseite der Anwendung



Abbildung 6.3: Implementierung: Startseite der Anwendung im Lightmode

Sollten Nutzende eine Buchung vornehmen wollen, so klicken diese in den gewünschten Zeitraum und es wird der Dialog in Abbildung 6.4 dargestellt.

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

Außerdem können Nutzende die Priorität des Termins als *Niedrig*, *Mittel* oder *Hoch* angeben. Nutzende können auch bestimmen, ob sie bereit sind, den Raum mit anderen Nutzenden zu teilen. Für diesen Zweck werden Ihnen drei Optionen bereitgestellt: *Ja*, *Nein* und *Auf Anfrage*.

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

Die Implementierung des Dialogs ist in Abbildung 6.5 und Abbildung 6.6 zu sehen.

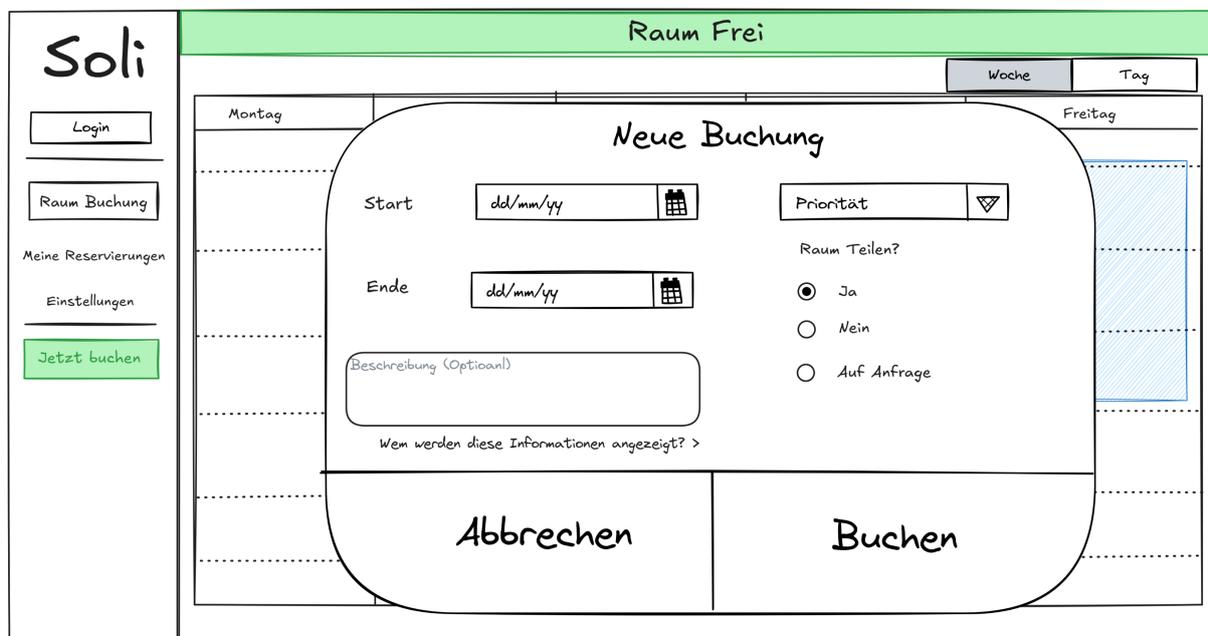


Abbildung 6.4: Termin-erstellen

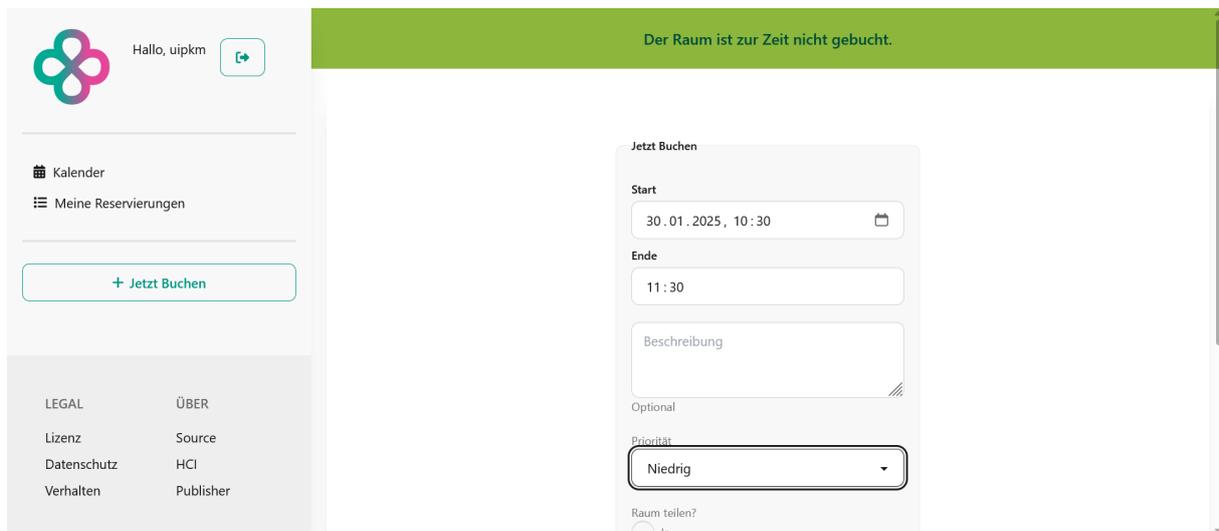


Abbildung 6.5: Implementierung: Termin-erstellen

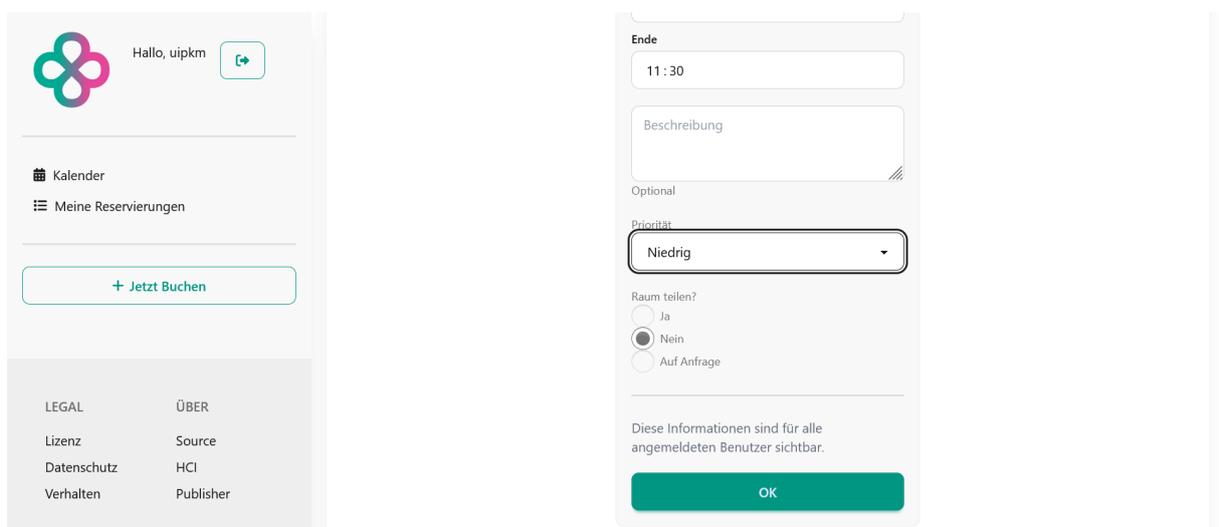


Abbildung 6.6: Implementierung: Termin-erstellen

In 6.7 ist die Anmeldungsansicht dargestellt und in 6.8 ist die tatsächliche Implementierung zu sehen.

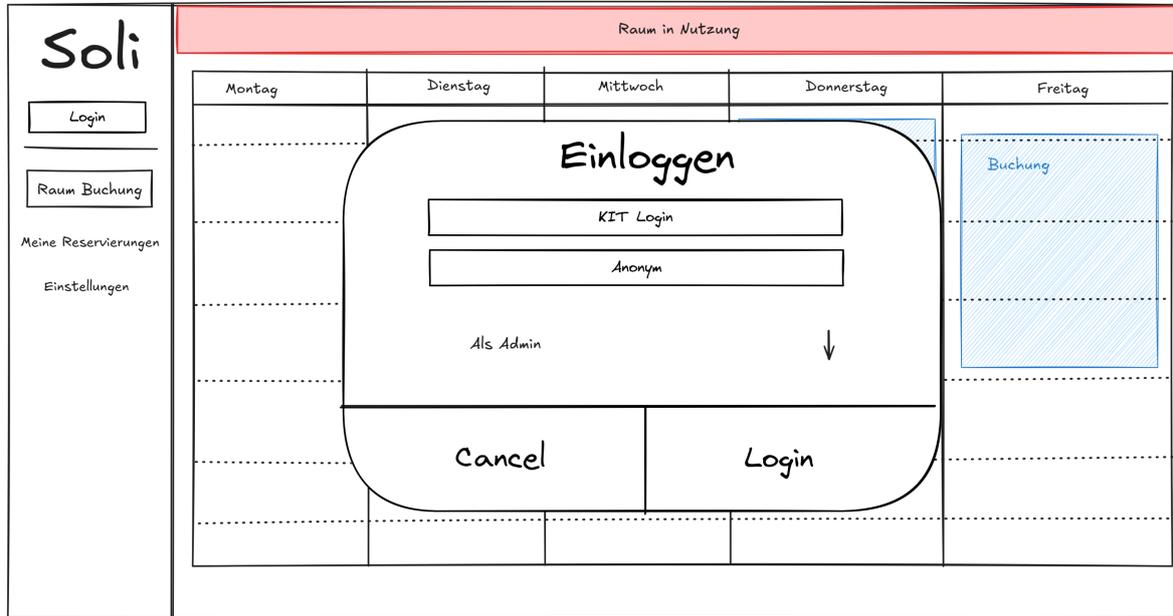


Abbildung 6.7: Anmeldungsseite

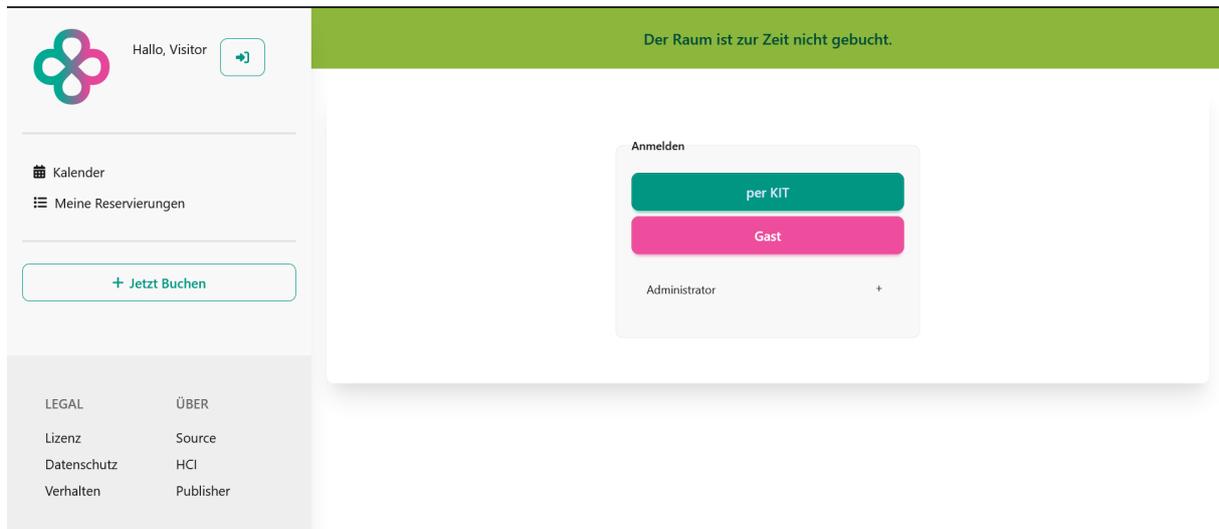


Abbildung 6.8: Implementierung: Anmeldungsseite

Sind Nutzende eingeloggt und belegen den Raum, so wird ihnen die in Abbildung 6.9 dargestellte Ansicht angezeigt. Hier können Nutzende den Raum wieder über den Quick-Checkout-Button freigeben.

Die Implementierung davon ist in Abbildung 6.10 zu sehen.

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

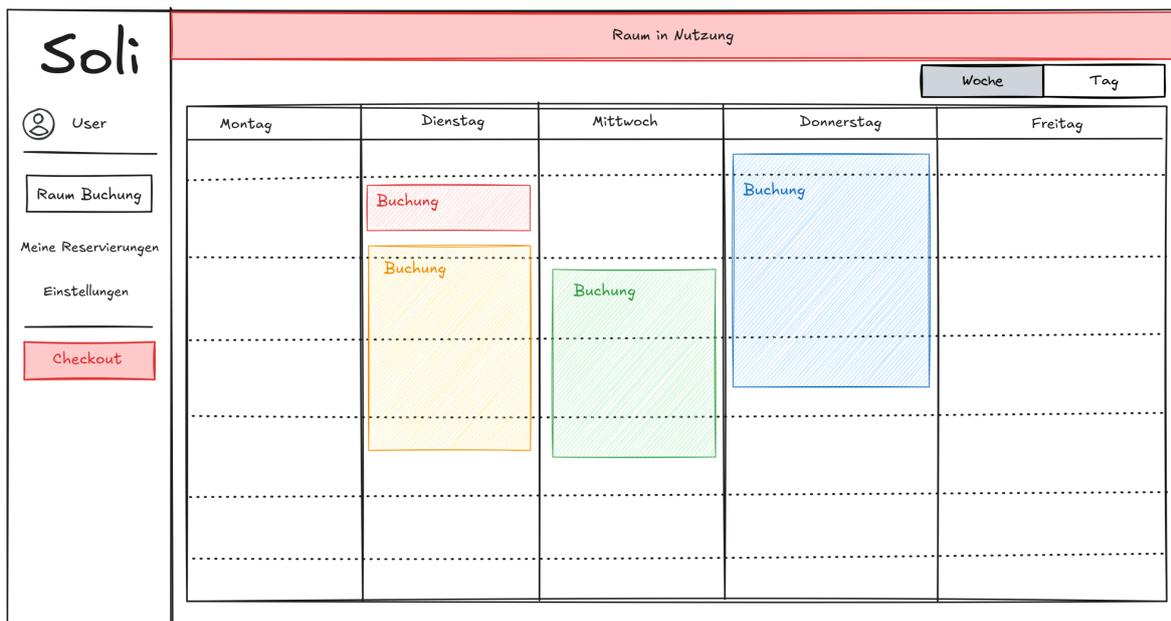


Abbildung 6.9: Quick Checkout



Abbildung 6.10: Implementierung: Quick Checkout

6.2 Terminübersicht

Nutzende, die eine Buchung vorgenommen haben, können diese in der Terminübersicht, die in Abbildung 6.11 dargestellt ist, einsehen und verwalten.

Die Implementierung der Terminübersicht ist in Abbildung 6.12 zu sehen und die der Terminansicht in Abbildung 6.14.

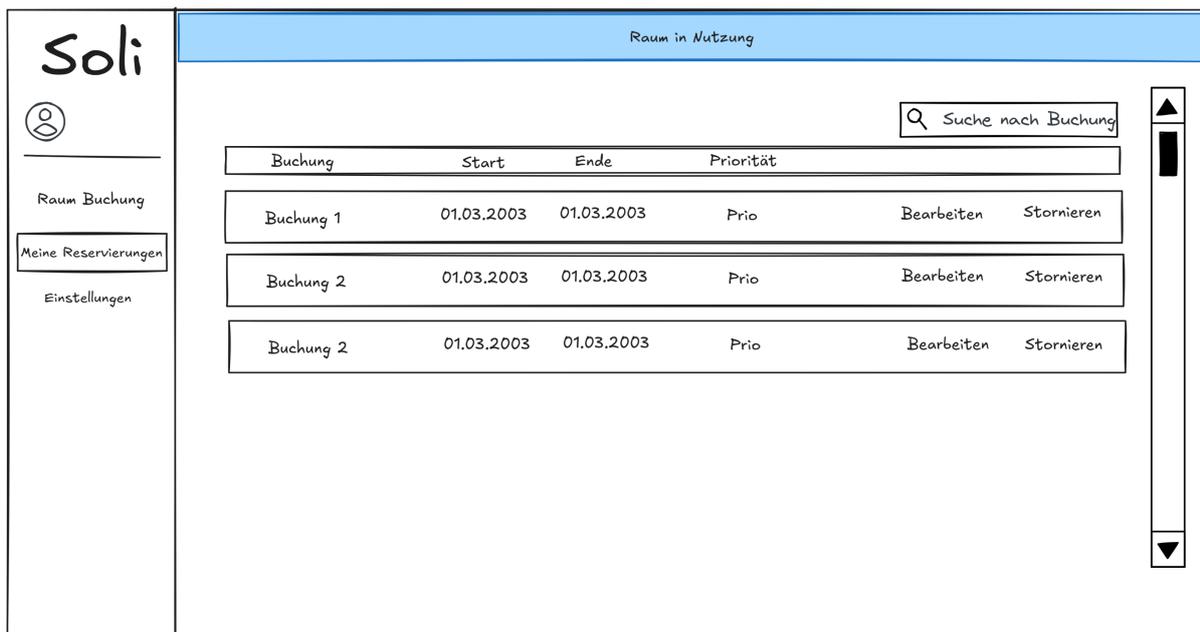
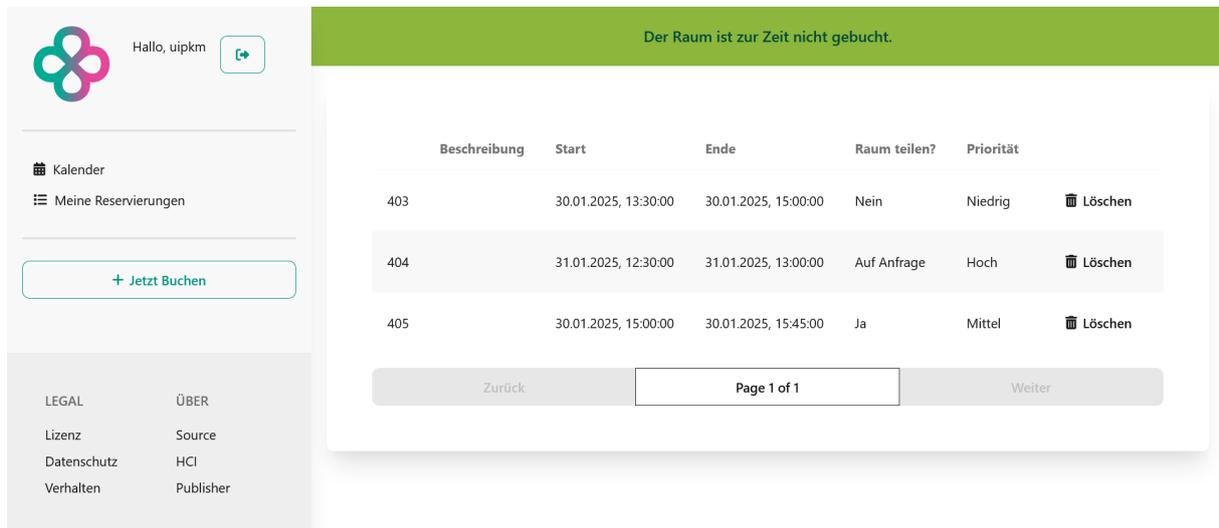


Abbildung 6.11: Reservierungsübersicht



Hallo, uipkm

Kalender
Meine Reservierungen

+ Jetzt Buchen

LEGAL ÜBER
Lizenz Source
Datenschutz HCI
Verhalten Publisher

Der Raum ist zur Zeit nicht gebucht.

Beschreibung	Start	Ende	Raum teilen?	Priorität	
403	30.01.2025, 13:30:00	30.01.2025, 15:00:00	Nein	Niedrig	Löschen
404	31.01.2025, 12:30:00	31.01.2025, 13:00:00	Auf Anfrage	Hoch	Löschen
405	30.01.2025, 15:00:00	30.01.2025, 15:45:00	Ja	Mittel	Löschen

Zurück Page 1 of 1 Weiter

Abbildung 6.12: Implementierung: Reservierungsübersicht

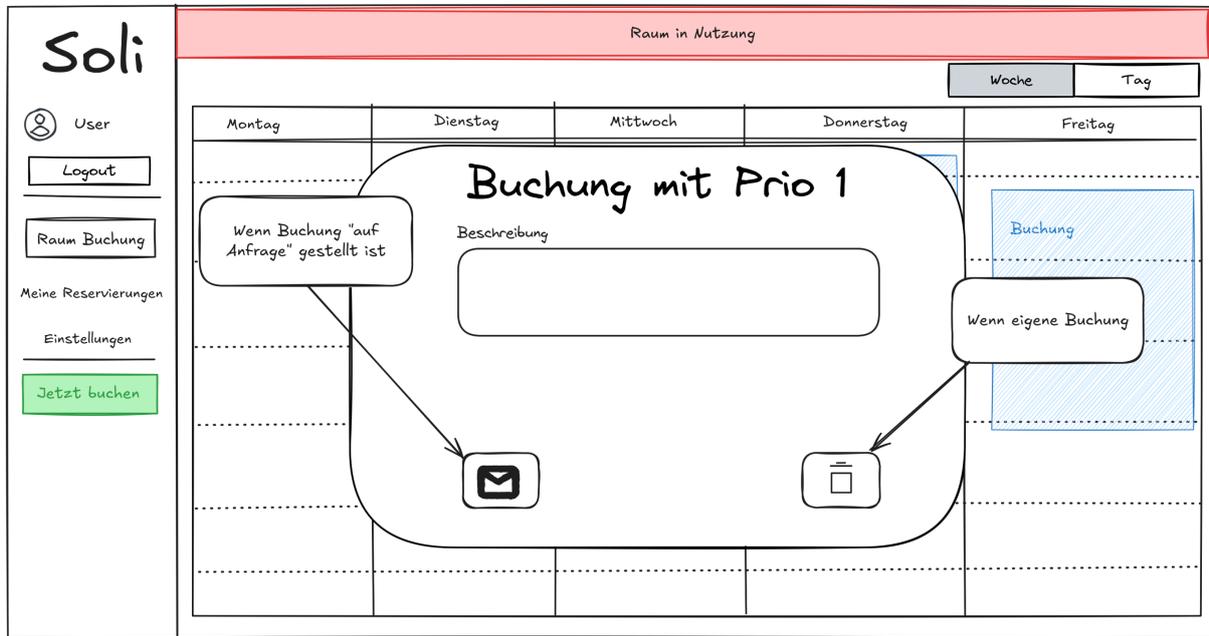


Abbildung 6.13: Reserverierung im Kalender

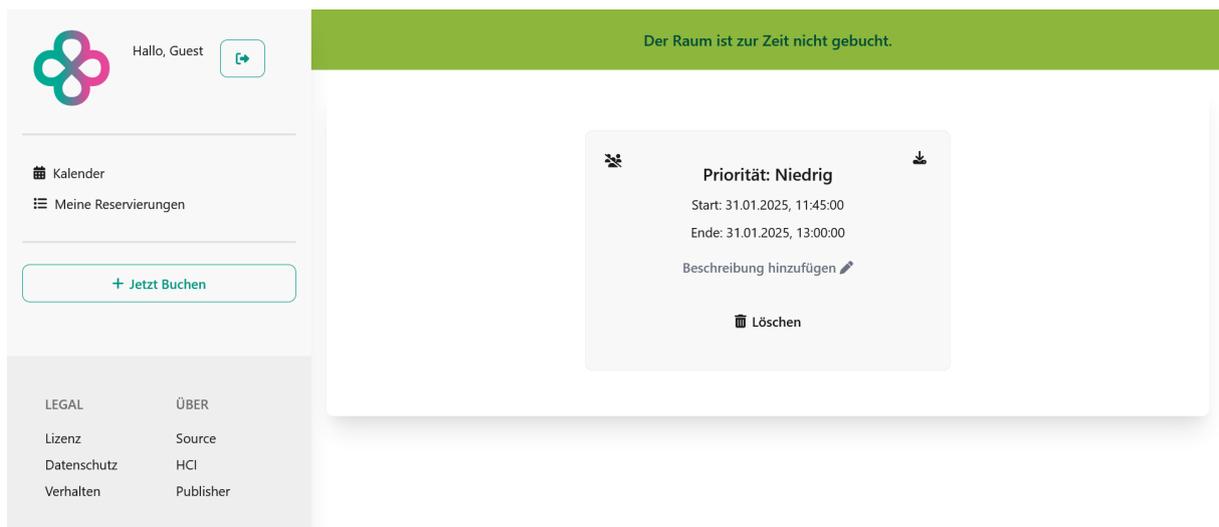


Abbildung 6.14: Implementierung: Reserverierung im Kalender

6.3 Administration

Ein/e Administrator*in hat die Möglichkeit, über die Benutzeradministrationsoberfläche, die in Abbildung 6.15 dargestellt ist, Nutzende einzusehen und zu verwalten.

Die Implementierung der Kontenliste ist in Abbildung 6.16 dargestellt.



Abbildung 6.15: Benutzeradministrationsoberfläche

The screenshot displays a user administration interface. At the top left, there is a logo and a greeting 'Hallo, admin' with a user icon. A green banner at the top right contains the message 'Der Raum ist zur Zeit nicht gebucht.' Below this, a red button labeled 'Gastlogin deaktivieren' is positioned above a table of guest users. The table has five columns: an index, 'Benutzername', 'Email', 'ID', and 'Deaktiviert'. The table contains five rows of data. On the left side, there is a sidebar with navigation links: 'Konten', 'Statistiken', 'Räume', 'Öffnungszeiten', and 'Kalender'. At the bottom of the sidebar, there are links for 'LEGAL' and 'ÜBER', with sub-links for 'Lizenz', 'Source', 'Datenschutz', 'HCI', 'Verhalten', and 'Publisher'.

	Benutzername	Email	ID	Deaktiviert
52	Guest	fakemail@example.com	guest/fakemail@example.com	false
102	Guest	abc@example.com	guest/46751950-4899-489f-908d-77aa41f5b354/ abc@example.com	false
2	Guest	foo@bar	guest/foo@bar	false
202	Guest	dimi.al@gmx.de	guest/8a5635c9-9845-4abe-94d8-5f867c3364cf/ dimi.al@gmx.de	false
203	Guest	ol2208@kit.edu	guest/2beea480-9454-48f6-8809-1ec81a7ad7af/ ol2208@kit.edu	false

Abbildung 6.16: Implementierung: Benutzeradministrationsoberfläche

Die Funktionalität der einstellbaren Öffnungszeiten für einen Raum, welche nur von Administrator*innen genutzt werden kann ist in seiner implementierten Form in Abbildung 6.17 zu sehen.

Die implementierte Visualisierung der Öffnungszeiten in der *Kalender* Ansicht in Abbildung 6.18 zu sehen ist.

The screenshot shows a configuration page for a room. At the top, a green banner displays the message "Der Raum ist zur Zeit nicht gebucht." Below this is a table with three columns: "Day", "Start Time", and "Closing Time". The rows correspond to the days of the week, with input fields for the start and end times. A "Speichern" (Save) button is located at the bottom left of the table.

Day	Start Time	Closing Time
Montag	10:00	18:00
Dienstag	10:00	18:00
Mittwoch	11:00	18:00
Donnerstag	12:00	19:00
Freitag	10:00	17:00

Abbildung 6.17: Implementierung: Konfiguration

The screenshot shows a calendar view for the week of February 3-7, 2025. The calendar is a grid with time slots on the y-axis (from 10:00 to 13:30) and days on the x-axis (Mo. 03.02., Di. 04.02., Mi. 05.02., Do. 06.02., Fr. 07.02.). Shaded cells indicate booked time slots: from 10:00 to 11:00 on Wednesday (Mi. 05.02.) and from 11:00 to 12:00 on Thursday (Do. 06.02.).

	Mo. 03.02.	Di. 04.02.	Mi. 05.02.	Do. 06.02.	Fr. 07.02.
10:00					
10:30					
11:00					
11:30					
12:00					
12:30					
13:00					
13:30					

Abbildung 6.18: Implementierung: Visualisierung der Öffnungszeiten

Die Implementierung des Wunschkriteriums einer Statistikansicht für ein/e Administrator*in ist in Abbildung 6.19 zu sehen.



Abbildung 6.19: Implementierung: Statistikansicht

Glossar

iCal Dateiformat zur Speicherung von Kalenderdaten. 8, 9

OIDC (OpenID Connect) Authorisierungsframework welches vom KIT genutzt wird um dritten Webseiten Logins basierend auf KIT-Konten bereitzustellen. 6

UI (User Interface) Typischerweise visuelle Oberfläche, mit welcher der Nutzende interagiert. 8, 9