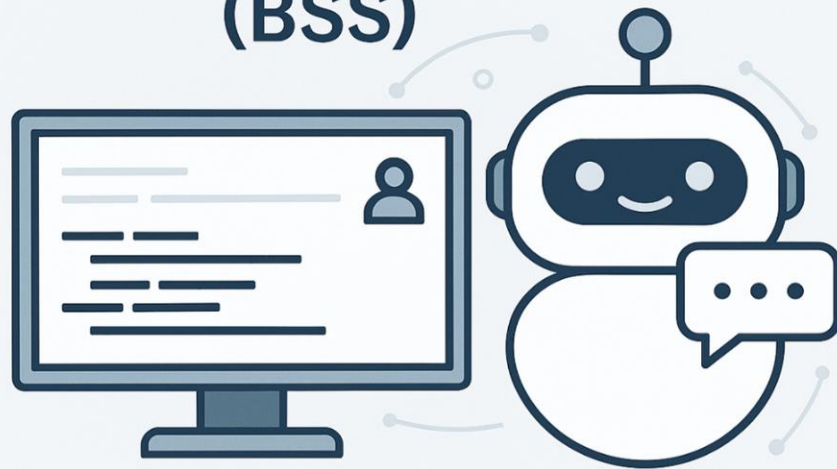


# KI-Agenten für die Personaleinsatzplanung (BSS)



## Technische Spezifikation Softwareentwicklung

KI-Agenten für die Personaleinsatzplanung (BSS)

**Autoren des Dokuments**

Ayham Salha  
Gürhan Arabaci  
Koutaibe Alhassan

**Ort, Datum**

Berlin, 04.05.2025

## Inhalt

1	Analyse .....	3
1.1	Kontext und Überblick .....	3
1.1.1	Externe Schnittstellen .....	4
1.2	Anwendungsfälle .....	5
1.3	Qualitätsanforderungen .....	13
1.4	Testszenarien .....	14
2	Entwurf .....	17
2.1	Fachliche Strukturen und Modelle .....	17
2.2	Benutzeroberfläche .....	18
2.3	Systemarchitektur .....	20
2.3.1	Komponentenbeschreibungstabelle .....	21
2.4	Technologiestack und Lösungsansätze für die Umsetzung der Qualitätsanforderungen	22
2.5	Lizenzmodell .....	22
3	Aufteilung der Kapitel .....	23

# 1 Analyse

## 1.1 Kontext und Überblick

Ziel des Softwareprojekts ist es, ein KI-gestütztes Assistenzsystem zu entwickeln, das die Personaleinsatzplanung unterstützt. Das System soll Führungskräften helfen, Arbeitszeiten besser zu planen und zu koordinieren. Dabei nutzt es vorhandene Planungsdaten, um sinnvolle Vorschläge zu machen und auf mögliche Probleme wie Überschneidungen oder fehlende Qualifikation hinzuweisen.

Im Mittelpunkt steht ein sogenannter KI-Agent, der die Nutzer beim Planen begleitet. Er gibt kontextbezogene Tipps, erkennt typische Fehler und schlägt automatisch passende Mitarbeitende für bestimmte Aufgaben oder Schichten vor.

Die Bedienung erfolgt über eine Chat-Oberfläche, in der die Nutzer einfach Fragen stellen oder Informationen eingeben können. Der KI-Agent antwortet direkt im Chat, gibt Erklärungen und reagiert auf Eingaben, ähnlich wie ein Chatbot, aber mit deutlich mehr Funktionen. Dadurch wird die Planung einfacher, verständlicher und interaktiver gestaltet.

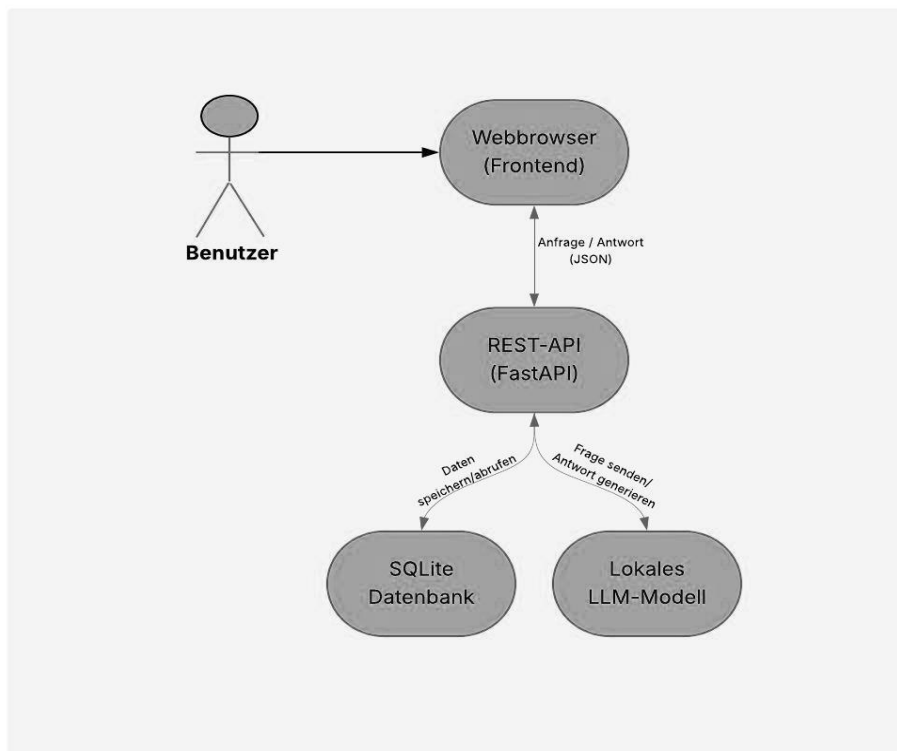


Abbildung 1: Kontextdiagramm

## 1.1.1 Externe Schnittstellen

### 1.1.1.1 Datenbank – SQLite (lokale Datenbank)

Für die Speicherung von Chatverläufen, Benutzeranfragen und KI-Antworten wird eine lokale SQLite-Datenbank eingesetzt.

SQLite ist leichtgewichtig und ermöglicht die schnelle Speicherung ohne die Notwendigkeit eines separaten Datenbankservers.

Im späteren Verlauf kann eine Umstellung auf ein vollwertiges DBMS wie Microsoft SQL Server in Betracht gezogen werden, wie im Lastenheft beschrieben.

### 1.1.1.2 Webbrowser (Frontend)

Die Benutzerinteraktion erfolgt über einen aktuellen Webbrowser (z.B. Chrome, FireFox).

Das Frontend wird mit HTML, CSS und JavaScript umgesetzt und kommuniziert mit der API des KI-Agenten über HTTP-Anfragen (Fetch-API).

### 1.1.1.3 REST-API (Representational State Transfer) - FASTAPI (schnelles Web-Framework für Python)

Der KI-Agent stellt eine REST-API auf Basis von FastAPI bereit.

Diese API verarbeitet Anfragen im JSON-Format und ermöglicht:

- das Senden von Benutzeranfragen (POST /chat),
- das Abrufen von Chatverläufen (GET /history).

Die API kommuniziert intern mit dem KI-Modell und der Datenbank.

### 1.1.1.4 Lokaler Server – Uvicorn (Entwicklungsumgebung)

Während der Entwicklung wird die Anwendung über den lokalen Uvicorn-Server betrieben. Dieser Server ermöglicht das lokale Testen der FastAPI-Anwendung.

Die Kommunikation erfolgt ebenfalls über HTTP.

### 1.1.1.5 KI-Backend – Lokales LLM Modell (Large Language Modell)

Für die Generierung von Antworten wird ein lokal betriebenes KI-Modell verwendet.

In diesem Fall basiert das System auf einem lokal gestarteten LLM (Large Language Model) wie z.B. gemma:2b, welcher über eine interne API (Port 11434) angesprochen wird. Das Modell ist jedoch nicht fest in die Architektur integriert, sondern modular angebunden. Dadurch kann bei Bedarf problemlos auf ein anderes, kompatibles Sprachmodell umgestiegen werden, bei Bedarf. Die gesamte Anwendung wird in Python entwickelt und über Visual Studio Code lokal getestet und betrieben.

## 1.2 Anwendungsfälle

Das Use-Case-Diagramm visualisiert die Kernfunktionen und Benutzerinteraktionen des KI-Agenten für die Personaleinsatzplanung.

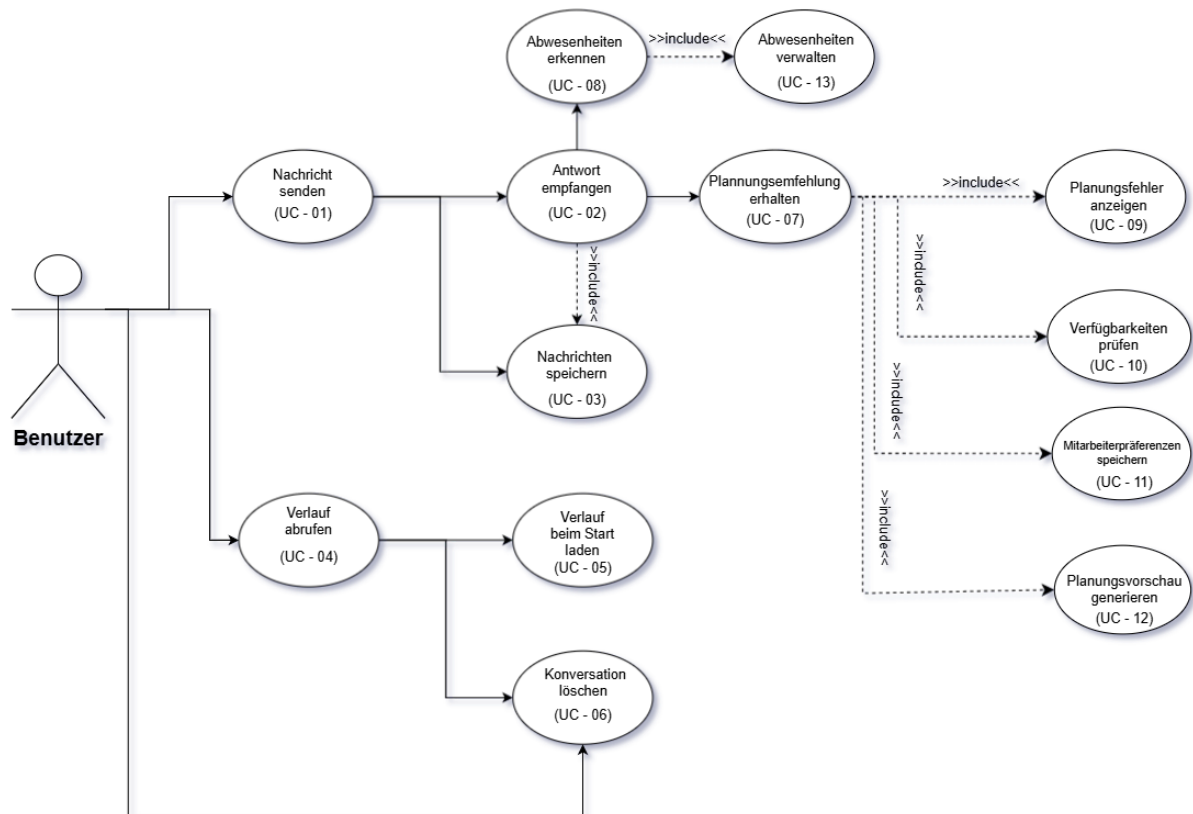


Abbildung 2: Use-Case-Diagramm

## Use Case 01: Nachricht senden

Der Benutzer sendet eine Nachricht an den KI-Agenten, um eine Anfrage zu stellen.

**Tabelle 1: Nachricht senden**

Information	Beschreibung
Identifikation	UC-01
Beschreibung	Der Benutzer sendet eine Anfrage an den KI-Agenten.
Akteure	Benutzer, System
Voraussetzung	Benutzer greift auf die Benutzeroberfläche zu.
Standardablauf	1. Benutzer gibt eine Nachricht ein. 2. Nachricht wird an die API übermittelt.
Alternative Abläufe	API nicht erreichbar: Fehlermeldung wird angezeigt.
Nachbedingung	Nachricht wird verarbeitet und gespeichert.
Bemerkung	Nur gültige Eingaben werden akzeptiert.

## Use Case 02: Antwort empfangen

Nach dem Absenden einer Anfrage erhält der Benutzer eine Antwort vom KI-Agenten.

**Tabelle 2: Antwort erhalten**

Information	Beschreibung
Identifikation	UC-02
Beschreibung	Der Benutzer erhält eine Antwort vom KI-Agenten.
Akteure	Benutzer, System
Voraussetzung	Erfolgreich gesendete Anfrage.
Standardablauf	1. System empfängt Antwort. 2. Antwort wird angezeigt.
Alternative Abläufe	Antwort dauert zu lange: Ladesymbol wird angezeigt.
Nachbedingung	Antwort wird im Verlauf gespeichert.
Bemerkung	Antwort ist direkt sichtbar im Chat.

### Use Case 03: Nachricht speichern

Nach dem Empfangen einer Nachricht wird der Gesprächsverlauf automatisch gespeichert.

**Tabelle 3: Nachricht speichern**

Information	Beschreibung
Identifikation	UC-03
Beschreibung	Die Unterhaltung wird automatisch in der Datenbank gespeichert.
Akteure	System
Voraussetzung	Erfolgreiche Übertragung von Nachrichten und Antwort.
Standardablauf	1. Nachricht und Antwort werden gespeichert
Alternative Abläufe	Datenbankfehler: Speicherung wird erneut versucht.
Nachbedingung	Datenbank enthält aktuellen Chatverlauf.
Bemerkung	Speicherung läuft im Hintergrund.

### Use Case 04: Verlauf Abrufen

Der Benutzer kann jederzeit vergangene Unterhaltungen über die Benutzeroberfläche abrufen.

**Tabelle 4: Verlauf Abrufen**

Information	Beschreibung
Identifikation	UC-04
Beschreibung	Der Benutzer ruft alte Nachrichtenverläufe ab.
Akteure	Benutzer, System
Voraussetzung	Vorhandene Nachrichten in der Datenbank
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer klickt auf "Verlauf abrufen".</li> <li>2. Verlauf wird geladen.</li> </ol>
Alternative Abläufe	Keine gespeicherten Nachrichten: Leere Anzeige.
Nachbedingung	Verlauf ist sichtbar.
Bemerkung	Verlauf kann gescrollt werden.

## Use Case 05: Verlauf beim Start laden

Beim Start der Anwendung wird automatisch der zuletzt gespeicherte Chatverlauf angezeigt

**Tabelle 5: Verlauf beim Start laden**

Information	Beschreibung
Identifikation	UC-05
Beschreibung	Der bestehende Verlauf wird automatisch beim Start der Anwendung geladen.
Akteure	System
Voraussetzung	Existierende Datenbankeinträge.
Standardablauf	<ol style="list-style-type: none"> <li>1. Anwendung wird geöffnet.</li> <li>2. Verlauf wird angezeigt.</li> </ol>
Alternative Abläufe	Keine Einträge vorhanden: Hinweis wird angezeigt.
Nachbedingung	Benutzer sieht die letzten Nachrichten.
Bemerkung	Läuft automatisch im Hintergrund.

## Use Case 06: Konversation löschen

Der Benutzer hat die Möglichkeit, den gesamten Chatverlauf über die Oberfläche zu löschen.

**Tabelle 6: Konversation löschen**

Information	Beschreibung
Identifikation	UC-06
Beschreibung	Der Benutzer kann den gesamten Verlauf löschen.
Akteure	Benutzer, System
Voraussetzung	Bestehender Chatverlauf vorhanden.
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer klickt auf "Konversation löschen".</li> <li>2. Verlauf wird entfernt.</li> </ol>
Alternative Abläufe	Benutzer bricht Löschung ab.
Nachbedingung	Verlauf wird geleert.
Bemerkung	Bestätigungsabfrage notwendig.



## Use Case 07: Planungsempfehlung erhalten

Basierend auf den eingegebenen Anforderungen schlägt der KI-Agent passende Einsatzoptionen vor.

**Tabelle 7: Planungsempfehlung erhalten**

Information	Beschreibung
Identifikation	UC-07
Beschreibung	Der KI-Agent schlägt Planungsoptionen basierend auf Eingaben vor.
Akteure	Benutzer, System
Voraussetzung	Vorherige Anfrage erfolgreich bearbeitet.
Standardablauf	1. Benutzer erhält Vorschlag. 2. Benutzer bestätigt oder lehnt ab.
Alternative Abläufe	Keine sinnvolle Planung möglich: Fehlermeldung.
Nachbedingung	Planungsvorschlag wird gespeichert.
Bemerkung	Funktion für die spätere Version.

## Use Case 08: Abwesenheiten erkennen

Der KI-Agent überprüft, ob geplante Einsätze mit Abwesenheiten kollidieren.

**Tabelle 8: Abwesenheiten erkennen**

Information	Beschreibung
Identifikation	UC-08
Beschreibung	Der KI-Agent erkennt mögliche Überschneidung oder Abwesenheiten.
Akteure	Benutzer, System
Voraussetzung	Zugang zu entsprechenden Kalender- oder Arbeitszeitdaten.
Standardablauf	1. Anfrage wird analysiert. 2. Warnung bei Abwesenheit.
Alternative Abläufe	Keine Abwesenheit erkannt: Keine Warnung.
Nachbedingung	Benutzer wird rechtzeitig informiert.
Bemerkung	Erweiterung der Funktionalität möglich.

## Use Case 09: Planungsfehler anzeigen

Der KI-Agent identifiziert automatisch mögliche Planungsfehler und stellt diese visuell dar.

**Tabelle 9: Planungsfehler anzeigen**

Information	Beschreibung
Identifikation	UC-09
Beschreibung	Der KI-Agent zeigt erkannte Planungsfehler visuell an.
Akteure	Benutzer, System
Voraussetzung	Planungsempfehlung wurde angefordert.
Standardablauf	<ol style="list-style-type: none"> <li>1. System analysiert Planungsdaten.</li> <li>2. Fehlerhafte Einsätze werden markiert.</li> </ol>
Alternative Abläufe	Keine Fehler vorhanden: Hinweis auf fehlerfreie Planung.
Nachbedingung	Benutzer sieht erkannte Probleme.
Bemerkung	Fehlerarten z. B. Überschneidungen, fehlende Qualifikationen.

## Use Case 10: Verfügbarkeiten prüfen

Zur Planungsausführung prüft das System, ob Mitarbeitende für bestimmte Zeiträume verfügbar sind

**Tabelle 10: Verfügbarkeiten prüfen**

Information	Beschreibung
Identifikation	UC-10
Beschreibung	Prüft die Verfügbarkeit von Mitarbeitenden für geplante Einsätze.
Akteure	System
Voraussetzung	Zugriff auf Abwesenheiten und Kalenderdaten
Standardablauf	<ol style="list-style-type: none"> <li>1. System greift auf Kalenderdaten zu.</li> <li>2. Verfügbare Zeiträume werden markiert</li> </ol>
Alternative Abläufe	Keine Daten vorhanden: Nutzer wird informiert.
Nachbedingung	Verfügbarkeiten stehen der Planung zur Verfügung.
Bemerkung	Grundlage für Empfehlungen und Konfliktanalysen.

## Use Case 11: Mitarbeiterpräferenzen speichern

Benutzer können individuelle Einsatzwünsche eingeben, um ihre Planung zu beeinflussen

**Tabelle 11: Mitarbeiterpräferenzen speichern**

Information	Beschreibung
Identifikation	UC-11
Beschreibung	Der Benutzer hinterlegt individuelle Wünsche zur Einsatzplanung.
Akteure	Benutzer, System
Voraussetzung	Benutzer ist authentifiziert.
Standardablauf	<ol style="list-style-type: none"> <li>1. Benutzer gibt Präferenzen ein.</li> <li>2. System speichert die Angaben.</li> </ol>
Alternative Abläufe	Eingabe unvollständig: Warnhinweis wird angezeigt.
Nachbedingung	Präferenzen sind mit Benutzer verknüpft gespeichert.
Bemerkung	Kann z.B. Wunsch-Schichten, freie Tage, Rollenpräferenzen enthalten.

## Use Case 12: Planungsvorschau generieren

Der KI-Agent erstellt auf Basis vorhandener Daten eine Vorschau des Einsatzplans

**Tabelle 12: Planungsvorschau generieren**

Information	Beschreibung
Identifikation	UC-12
Beschreibung	Der KI-Agent erstellt eine Vorschau des geplanten Einsatzplans.
Akteure	Benutzer, System
Voraussetzung	Alle relevanten Daten (Verfügbarkeiten, Präferenzen etc.) sind vorhanden.
Standardablauf	<ol style="list-style-type: none"> <li>1. System generiert Vorschau.</li> <li>2. Benutzer sieht Ergebnis</li> </ol>
Alternative Abläufe	Fehlende Daten: System gibt Hinweis aus.
Nachbedingung	Vorschau steht für Freigabe oder Bearbeitung zur Verfügung.
Bemerkung	Optionalen Schritt vor dem endgültigen Plan.

## Use Case 13: Abwesenheiten verwalten

Benutzer können Abwesenheiten eintragen oder bearbeiten, um die Einsatzplanung aktuell zu halten.

**Tabelle 13: Abwesenheiten verwalten**

Information	Beschreibung
Identifikation	UC-13
Beschreibung	Benutzer kann eigene Abwesenheiten eintragen oder bestehende bearbeiten.
Akteure	Benutzer, System
Voraussetzung	Benutzer ist angemeldet.
Standardablauf	<ol style="list-style-type: none"> <li>1. Abwesenheit hinzufügen oder ändern.</li> <li>2. System aktualisiert Daten.</li> </ol>
Alternative Abläufe	Ungültiges Datum oder Überschneidung: Warnung erscheint.
Nachbedingung	Abwesenheitsdaten sind aktualisiert.
Bemerkung	Wichtige Grundlage für die Verfügbarkeitsprüfung.

### 1.3 Qualitätsanforderungen

Die folgenden Qualitätsziele beschreiben grundlegende Anforderungen an die Funktion und Bedienbarkeit des Systems.

**Tabelle 14:**

Ziel	Beschreibung
<b>Zuverlässigkeit</b>	<p>Der KI-Agent muss zuverlässig arbeiten und in über 95 % der Anwendungsfälle korrekte Ergebnisse liefern. Dazu zählt insbesondere die:</p> <ul style="list-style-type: none"> <li>• fehlerfreie Erkennung von Abwesenheiten und Zeitüberschneidungen</li> <li>• richtige Auswahl geeigneter Mitarbeiter basierend auf Qualifikationen,</li> <li>• konsistente Kommunikation mit der Datenbank ohne Datenverlust.</li> </ul>
<b>Leistungsfähigkeit</b>	<p>Der KI-Agent muss in der Lage sein, mindestens 5 gleichzeitige Mitarbeitereinsätze (Testdaten) innerhalb von &lt;1 Sekunde Antwortzeit pro Anfrage zu verarbeiten. Das System soll bei maximaler Last (z. B. 100 gleichzeitige Planungen) stabil bleiben.</p>
<b>Skalierbarkeit</b>	<p>Das System muss skalierbar sein, um mit einer steigenden Anzahl von:</p> <ul style="list-style-type: none"> <li>• bis zu 100 Mitarbeitern,</li> <li>• und 1000 Einsatzplänen pro Jahr umgehen zu können.</li> </ul>
<b>Benutzerfreundlichkeit und Nutzbarkeit</b>	<p>Die Benutzeroberfläche muss intuitiv bedienbar sein und den Benutzer bei der Planung durch klare Hinweise unterstützen. Der Fokus liegt auf funktionaler Klarheit, nicht auf grafischer Perfektion.</p> <p>Anforderungen:</p> <ul style="list-style-type: none"> <li>• verständliche Fehlermeldungen</li> <li>• Tooltips bei Planungskonflikten</li> <li>• responsives Layout</li> </ul>

## 1.4 Testszenarien

Zur Sicherstellung der Systemfunktionalität werden die folgenden Testszenarien definiert, die zentrale Anwendungsfälle des KI-Agenten abdecken.

**Tabelle 15:**

Use Case	Testfall	Beschreibung	Ziel	Erwartetes Ergebnis
UC-01 Nachricht	1.1	Nachricht mit gültigem Text	Erfolgreich Senden	Nachricht wird korrekt gesendet
	1.2	Nachricht mit leerem Text senden	Fehlerbehandlung prüfen	Fehlermeldung erscheint
	1.3	Nachricht mit Sonderzeichen senden	Eingabepfung	Nachricht wird korrekt verarbeitet
UC-02 Antwort empfangen	2.1	Antwort auf normalen Prompt empfangen	Antwortverarbeitung prüfen	Antwort wird richtig angezeigt
	2.2	Antwort bei Serverfehler empfangen	Fehlerbehandlung testen	Fehlermeldung wird angezeigt
	2.3	Sehr lange Antwort empfangen	Systemreaktion bei großen Antworten	Antwort wird vollständig angezeigt
UC-03 Nachrichten speichern	3.1	Nachricht in DB speichern	Sicherung der Konversation prüfen	Nachricht ist korrekt gespeichert
	3.2	Fehlerhafte Speicherung simulieren	Fehlerfall testen	Fehlermeldung bei Speicherung
UC-04 Verlauf abrufen	4.1	Verlauf mit Einträgen abrufen	Verlaufsanzeige prüfen	Nachrichten werden korrekt angezeigt
	4.2	Verlauf ohne Einträge abrufen	Verhalten bei leerem Verlauf testen	Leere Verlaufsanzeige oder Hinweis

**Tabelle 16:**

Use Case	Testfall	Beschreibung	Ziel	Erwartetes Ergebnis
UC-05 Verlauf beim Start laden	5.1	Verlauf direkt beim Start laden	Automatische Verlaufsanzeige testen	Verlauf wird automatisch angezeigt
UC-06 Konversation löschen	6.1	Einzelne Konversation löschen	Löschfunktion testen	Nachricht wird gelöscht
	6.2	Alle Konversationen auf einmal löschen	Massenlöschung testen	Alle Nachrichten werden entfernt
UC-07 Planungsempfehlung erhalten	7.1	Planungsempfehlung bei normalem Verlauf	Qualität der Planung testen	Sinnvolle Planung wird vorgeschlagen
	7.2	Empfehlung bei fehlenden Daten	Verhalten bei unvollständiger Datenlage	Fehlerhinweis oder alternative Planung
UC-08 Abwesenheit erkennen	8.1	Abwesenheit korrekt erfassen	Erkennungstest	Abwesenheit wird richtig erkannt
	8.2	Falsche oder fehlende Abwesenheit	Fehlerbehandlung	Keine oder angepasste Meldung
UC-09	9.1	Planung mit Konflikt anzeigen	Fehlererkennung testen	Fehler wird korrekt angezeigt
	9.2	Planung ohne Konflikt	Fehlerfreiheit bestätigen	Keine Fehler werden angezeigt

**Tabelle 17:**

Use Case	Testfall	Beschreibung	Ziel	Erwartetes Ergebnis
UC-10	10.1	Verfügbarkeit vorhanden	Prüfung auf Einsatzfähigkeit	Verfügbarkeit wird bestätigt
	10.2	Keine Verfügbarkeit vorhanden	Einschränkungen erkennen	Hinweis auf fehlende Verfügbarkeit
UC-11	11.1	Gültige Präferenz speichern	Nutzerwunsch registrieren	Präferenz wird gespeichert
	11.2	Ungültige Eingabe	Fehlerbehandlung testen	Fehlermeldung wird angezeigt
UC-12	12.1	Vorschau mit vollständigen Daten	Ergebnisvorhersage prüfen	Vorschau wird korrekt angezeigt
	12.2	Vorschau mit fehlenden Daten	Verhalten bei Fehler prüfen	Fehlerhinweis oder leere Vorschau
UC-13	13.1	Abwesenheit hinzufügen	Verwaltung testen	Abwesenheit wird gespeichert
	13.2	Abwesenheit entfernen	Bearbeitung testen	Abwesenheit wird gelöscht



## 2 Entwurf

### 2.1 Fachliche Strukturen und Modelle

Das folgende ER-Diagramm stellt die zentralen Entitäten und Beziehungen des Systems dar und bildet die Grundlage für die Datenbankstruktur des KI-Agenten.

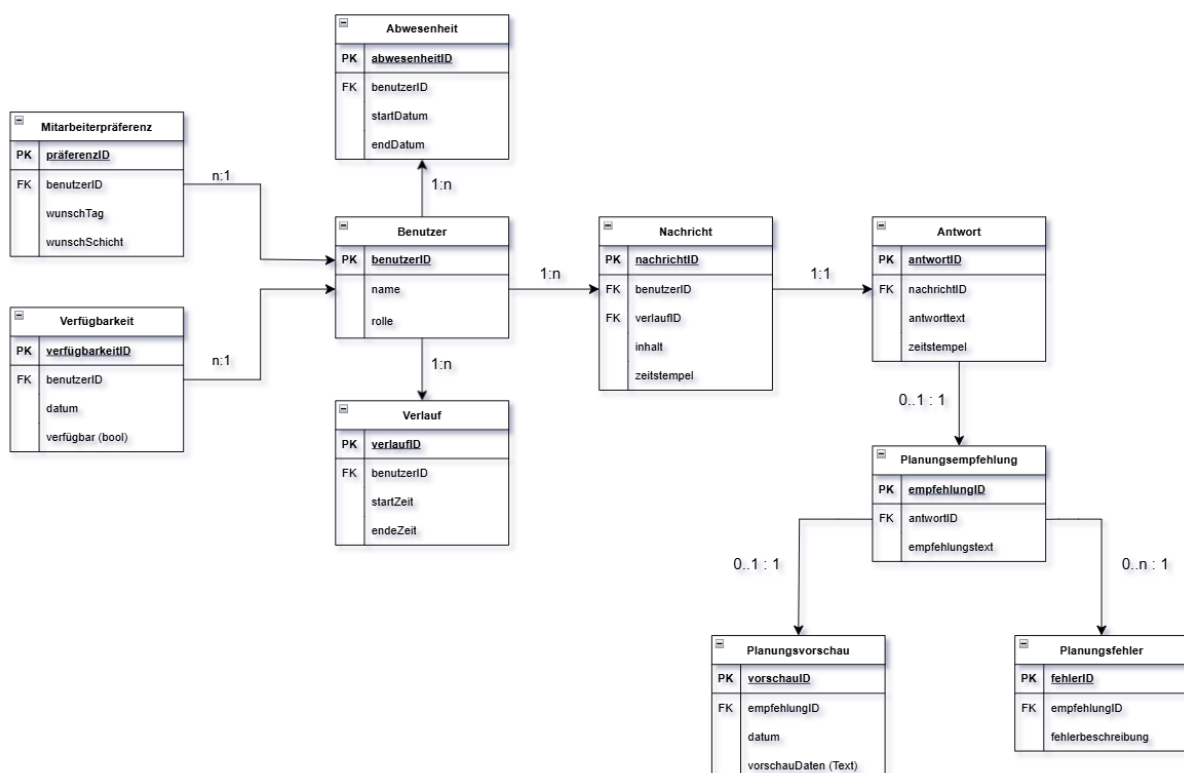


Abbildung 3: ER-Diagramm

## 2.2 Benutzeroberfläche

Die Benutzeroberfläche ermöglicht die intuitive Kommunikation mit dem KI-Agenten und zeigt Planungshinweise, Warnungen und Empfehlungen übersichtlich im Chatformat an.

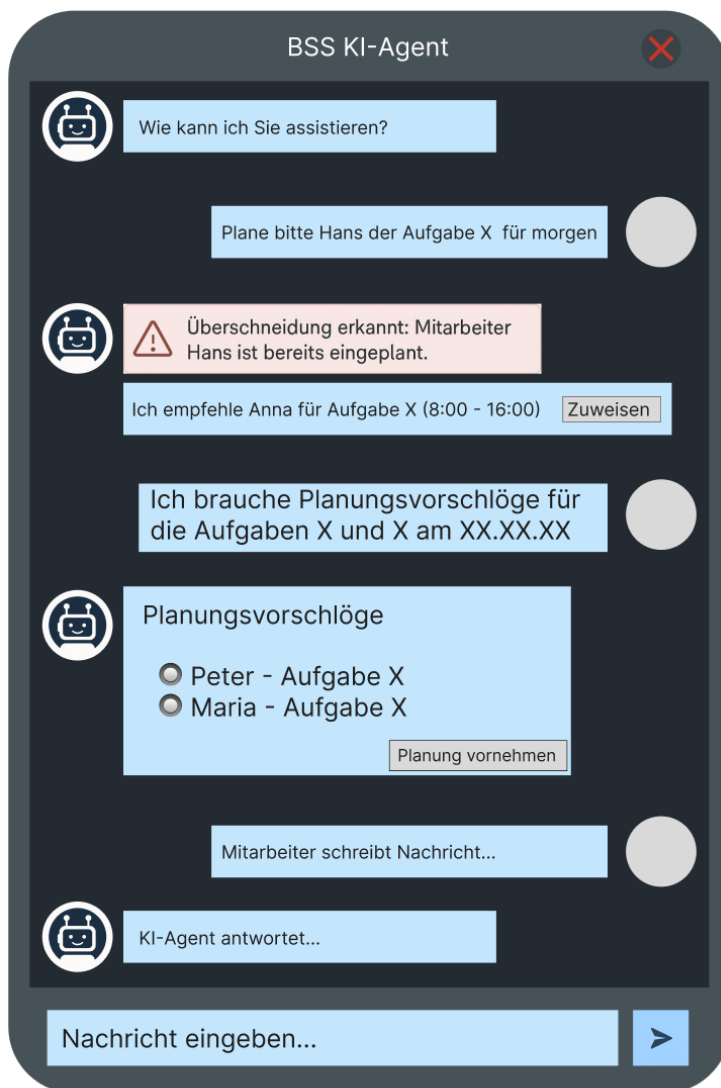


Abbildung 4: Mockup (Benutzeroberfläche)

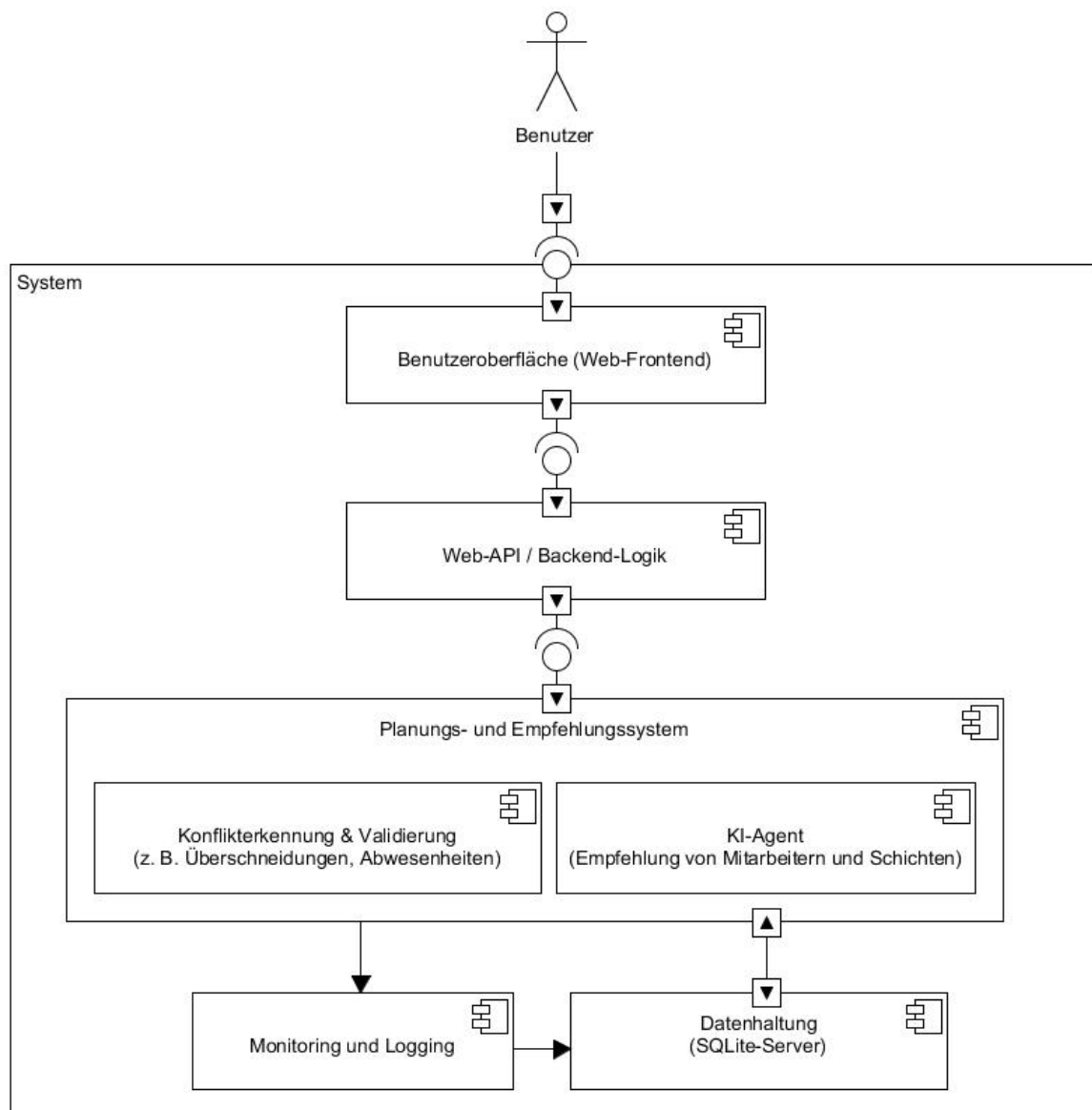




Abbildung 5: Mockup mit der Webseite der Personaleinsatzplanung von BSS

## 2.3 Systemarchitektur

Die Systemarchitektur zeigt den modularen Aufbau des KI-gestützten Planungssystems, von der Benutzeroberfläche bis zur Datenhaltung und dem KI-Agenten



**Abbildung 6: Systemarchitektur**

### 2.3.1 Komponentenbeschreibungstabelle

Die folgende Tabelle gibt einen Überblick über die zentralen Systemkomponenten und ihre jeweiligen Aufgaben im Gesamtsystem.

**Tabelle 18:**

Subsystem	Kurzbeschreibung
Benutzeroberfläche (Web-Frontend)	Ermöglicht es dem Benutzer, Einsätze einzugeben, zu planen und Empfehlungen abzurufen. Benutzerfreundliche und übersichtliche Darstellung von Plänen und Warnungen.
Web-API / Backend-Logik	Schnittstelle zwischen Benutzeroberfläche und dem Planungs-/Empfehlungssystem. Verantwortlich für die Steuerung von Benutzeranfragen und den Datenaustausch.
Planungs- und Empfehlungssystem	Beinhaltet die zentrale Logik zur Einsatzplanung: Überprüft Verfügbarkeiten, erkennt Konflikte (Überschneidungen, Fehlzeiten) und generiert KI-gestützte Empfehlungen für Mitarbeiterereinsätze.
Konflikterkennung & Validierung	Modul zur Erkennung von Planungsfehlern wie Überschneidungen, Qualifikationsmängeln oder Abwesenheiten.
KI-Agent (Empfehlungssystem)	Nutzt historische Einsatzdaten und Präferenzen, um personalisierte Vorschläge für Einsatzpläne zu generieren. Lernt kontinuierlich aus neuen Daten.
Datenhaltung (SQLite-Datenbank)	Speichert alle Mitarbeiterprofile, Schichtdaten, Einsatzpläne sowie Logs. Erlaubt effiziente Abfragen und Analysen.
Monitoring & Logging	Überwacht die Nutzung und Leistung des Systems. Erfasst relevante Ereignisse zur Optimierung, Fehlersuche und Sicherheit

## 2.4 Technologiestack und Lösungsansätze für die Umsetzung der Qualitätsanforderungen

Die folgende Tabelle zeigt, welche Technologien eingesetzt werden, um die zuvor definierten Qualitätsanforderungen zuverlässig zu erfüllen.

**Tabelle 19:**

Ziel	Passende Lösungsansätze
Zuverlässigkeit	Betrieb auf HTW-Server, Fehlerbehandlung mit Python-Fehlerkontrolle
Leistungsfähigkeit	Schnelle Datenbank mit SQL, asynchrone Verarbeitung in Python
Skalierbarkeit und Erweiterbarkeit	Modularer Backend-Aufbau mit Microservices; REST-APIs zur einfachen Integration und Erweiterung
Sicherheit	HTTPS-Verbindungen, sichere Anmeldung mit Tokens, verschlüsselte Passwörter
Benutzerfreundlichkeit und Nutzbarkeit	Frontend mit Bootstrap; responsive Design; klare Benutzerführung und sprachlich reduzierte Oberfläche

## 2.5 Lizenzmodell

**Gewählte Lizenz:** MIT-Lizenz

**Begründung:** Die MIT-Lizenz erlaubt die freie Nutzung, Veränderung und Verbreitung der Software bei gleichzeitigem Erhalt des Urheberrechtshinweises. Sie bietet maximale Flexibilität, ist sehr verbreitet im Open-Source-Bereich und kompatibel mit den von uns genutzten Technologien wie FastAPI, SQLite. Damit unterstützt sie unser Ziel, eine offene und leicht erweiterbare Lösung zur Personaleinsatzplanung zu entwickeln.

**Hinweis:** Das LLM-Modell gemma:2b unterliegt einer separaten Lizenz von MetaAI, die ausschließlich eine nicht-kommerzielle Nutzung erlaubt, somit ist sie nicht Bestandteil dieser Software und wird lediglich über eine lokale API angebunden. Bei Bedarf kann das Modell durch ein anderes kompatibles LLM ersetzt werden. Die Hauptsoftware bleibt dadurch unabhängig und ist vollständig unter der MIT-Lizenz nutzbar.

### 3 Aufteilung der Kapitel

Die technische Spezifikation ist durch enge Zusammenarbeit und gegenseitige Unterstützung aller Teammitglieder entstanden.

**Tabelle 20:**

Kapitel	Titel	Unterpunkte	Verantwortlicher
1	Analyse	1.1 Kontext und Überblick	Koutaibe Alhassan
		1.1.1 Externe Schnittstellen	Koutaibe Alhassan & Gürhan Arabaci
		1.2 Anwendungsfälle	Gürhan Arabaci
		1.3 Qualitätsanforderungen	Ayham Salha
		1.4 Abnahmeszenarien und -kriterien	Gürhan Arabaci
2	Entwurf	2.1 Fachliche Struktur	Gürhan Arabaci
		2.2 Benutzeroberfläche	Ayham Salha
		2.3 Systemarchitektur	Ayham Salha
		2.4 Technologiestack und Lösungsansätze für die Umsetzung der Qualitätsanforderungen	Koutaibe Alhassan
		2.5 Lizenzmodell	Alle Teammitglieder
3	Aufteilung der Kapitel		Alle Teammitglieder