

EventHub

**Felix Franke 1318821
Markus Wurms 50106907**

6. Februar 2026

Inhaltsverzeichnis

| | |
|--|-----------|
| 1 Fachliche Grundlagen und Anforderungen | 5 |
| 1.1 Zielsetzung des Softwareprodukts | 5 |
| 1.1.1 Produktname und Begründung | 5 |
| 1.2 Akteure und Rollen | 5 |
| 1.2.1 Nicht registrierte / unangemeldete Akteure | 5 |
| 1.2.2 Registrierte Akteure | 5 |
| 1.2.3 Sekundäre Akteure | 7 |
| 1.3 Glossar der Fachdomäne | 7 |
| 1.3.1 Event | 7 |
| 1.3.2 Gruppe | 7 |
| 1.3.3 Teilnehmer (Privatnutzer) | 7 |
| 1.3.4 Organisator (Gruppenleiter) | 7 |
| 1.3.5 Dienstleister | 7 |
| 1.3.6 Aufgabe | 8 |
| 1.3.7 Notiz | 8 |
| 1.3.8 Umfrage | 8 |
| 1.3.9 Budget | 8 |
| 1.3.10 Rolle | 8 |
| 1.3.11 Institution | 8 |
| 1.3.12 Institutionsadministrator | 8 |
| 1.3.13 Ressource | 8 |
| 2 Funktionale Anforderungen | 8 |
| 2.1 Funktionale Gruppen Übersicht | 8 |
| 2.2 FG1 Account und Authentifizierung | 9 |
| 2.3 FG2 Profil und Einstellungen | 9 |
| 2.4 FG3 Gruppen- und Eventmanagement | 9 |
| 2.5 FG4 Notizen und Umfragen | 10 |
| 2.6 FG5 Finanzen | 11 |
| 2.7 FG6 Aufgabenmanagement | 11 |
| 2.8 FG7 Medien und Dokumentation | 11 |
| 2.9 FG8 Rollen- und Rechtesystem | 12 |
| 2.10 FG9 Öffentliche Eventseite und QR-Code | 12 |
| 2.11 FG10 Integration und Benachrichtigungen | 12 |
| 2.12 FG11 Dienstleister und Anfragen | 13 |
| 2.13 FG12 Ressourcenplanung | 14 |
| 3 Nicht-funktionale Anforderungen (ISO 25010) | 14 |
| 3.1 Performance Efficiency | 14 |
| 3.2 Compatibility | 14 |
| 3.3 Usability (Gebrauchstauglichkeit) | 15 |
| 3.4 Reliability (Zuverlässigkeit) | 15 |
| 3.5 Security (Sicherheit) | 16 |
| 3.6 Maintainability (Wartbarkeit) | 16 |
| 3.7 Portability (Übertragbarkeit) | 17 |
| 3.8 Datenschutz (nicht aus der ISO) | 17 |

| | |
|---|-----------|
| 4 Graphische Benutzungsschnittstelle | 17 |
| 4.1 Designprinzipien und UX-Grundlagen | 17 |
| 4.1.1 Zielsetzung der Benutzeroberfläche | 17 |
| 4.1.2 Konsistenz und Navigationskonzept | 18 |
| 4.1.3 Rollenbasierte Darstellung | 18 |
| 4.1.4 Responsives Design und Geräteunabhängigkeit | 18 |
| 4.1.5 Fehlervermeidung und Systemfeedback | 18 |
| 4.1.6 Barrierearme Bedienbarkeit | 18 |
| 4.2 Navigationsmodelle | 18 |
| 4.2.1 Zustandsdiagramm – Privatnutzer | 19 |
| 4.2.2 Zustandsdiagramm – Organisator | 19 |
| 4.2.3 Zustandsdiagramm – Dienstleister | 20 |
| 4.3 Webbrowser-Mockups | 21 |
| 4.3.1 Login & Registrierung | 21 |
| 4.3.2 Dashboard | 22 |
| 4.3.3 Gruppenübersicht | 23 |
| 4.3.4 Event-Detailseite | 24 |
| 4.3.5 Finanzen | 25 |
| 4.3.6 Notizen & Umfragen | 26 |
| 4.3.7 Aufgabenplanung | 27 |
| 4.3.8 Profil & Einstellungen | 28 |
| 4.3.9 Dienstleister-Dashboard | 29 |
| 4.3.10 Rollen- und Rechteverwaltung | 30 |
| 4.3.11 Eventanfrage | 31 |
| 4.3.12 Eventübersicht (Lesemodus) | 32 |
| 4.4 Mobile-App-Mockups | 33 |
| 4.4.1 Dashboard (Mobil) | 33 |
| 4.4.2 Event-Details (Mobil) | 34 |
| 4.4.3 Aufgaben (Mobil) | 35 |
| 4.4.4 Finanzen (Mobil) | 36 |
| 5 Logische Sicht | 36 |
| 5.1 Ableitung der Domänenklassen aus den User Stories | 36 |
| 5.2 Fachliches Domänenmodell (UML-Klassendiagramm) | 37 |
| 5.3 Beziehungen, Multiplizitäten und Vererbungen | 38 |
| 5.4 Logisches Datenmodell | 38 |
| 5.4.1 Benutzer, Gruppen und Events | 38 |
| 5.4.2 Notizen & Umfragen | 39 |
| 5.4.3 Finanzen & Aufgaben | 40 |
| 5.4.4 Dienstleister & Abrechnung | 42 |
| 5.4.5 Ressourcenplanung | 44 |
| 5.5 Übersicht der Teilmodelle und Querverweise | 45 |
| 5.6 CRUD-Matrix | 46 |
| 6 Prozess-Sicht | 47 |
| 6.1 Zustandsdiagramm: Event | 47 |
| 6.2 Zustandsdiagramm: Rechnung | 48 |
| 6.3 Sequenzdiagramm: Gruppe per Beitrittscode beitreten | 49 |
| 6.4 Sequenzdiagramm: Einzahlung erfassen | 50 |

| | |
|--|-----------|
| 7 Entwicklungs-Sicht | 50 |
| 7.1 Drei-Schichten-Architektur (Client – Server – Datenbank) | 50 |
| 7.2 Client-Schicht (Präsentation) | 50 |
| 7.3 Server-Schicht (Applikation / Geschäftslogik) | 51 |
| 7.4 Datenbank-Schicht (Persistenz) | 51 |
| 7.5 Datenfluss | 51 |
| 7.6 Vorteile der Architektur | 52 |
| 7.7 Komponentendiagramm | 52 |
| 7.8 REST-API-Spezifikation | 53 |
| 7.8.1 API – Konventionen | 53 |
| 7.8.2 Ressourcen und Endpunkte | 53 |
| 8 Verteilungs-Sicht | 54 |
| 8.1 UML-Verteilungsdiagramm | 54 |
| 9 Sicherheit und Datenschutz | 55 |
| 9.1 STRIDE-Sicherheitsanalyse | 55 |
| 9.2 Schutzmaßnahmen | 55 |
| 9.2.1 Authentifizierung | 55 |
| 9.2.2 Autorisierung | 55 |
| 9.2.3 Eingabevalidierung und Datenintegrität | 55 |
| 9.2.4 Transport- und Datensicherheit | 55 |
| 9.2.5 Logging und Monitoring | 56 |
| 9.3 Datenschutz und DSGVO-Umsetzung | 56 |
| 9.3.1 Zugriffsschutz (Privacy by Design) | 56 |
| 9.3.2 Datenminimierung | 56 |
| 9.3.3 Einwilligung und Transparenz | 56 |
| 9.3.4 Löschkonzept und Aufbewahrung | 56 |
| 9.3.5 Betroffenenrechte | 56 |
| 10 Arbeitsteilung | 56 |
| 10.1 Aufgabenteilung Felix Franke | 56 |
| 10.1.1 Fachliche Grundlagen | 56 |
| 10.1.2 Funktionale Anforderungen | 56 |
| 10.1.3 Nicht-funktionale Anforderungen | 57 |
| 10.1.4 GUI & Usability | 57 |
| 10.2 Aufgabenteilung Markus Wurms | 57 |
| 10.2.1 Domänen- & Datenmodellierung | 57 |
| 10.2.2 Zustände & Datenpersistenz | 57 |
| 10.2.3 Systemarchitektur & Schnittstellen | 57 |
| 10.2.4 Sicherheit & Datenschutz | 57 |
| 11 Übersicht verwendeter Hilfsmittel | 58 |
| 11.1 OpenAI ChatGPT (openai.com, chatgpt.com) | 58 |
| 11.1.1 Nutzung von ChatGPT zur Formatierung | 58 |
| 11.1.2 Generierung von Mockups | 58 |
| 11.1.3 Prompts | 58 |

1 Fachliche Grundlagen und Anforderungen

1.1 Zielsetzung des Softwareprodukts

Die Zielsetzung des Softwareprodukts ist es, eine zentrale Plattform zur Planung, Organisation, Durchführung und Nachbereitung von Veranstaltungen zu erstellen. Dazu sollen viele verschiedene Tools zusammengelegt werden, wie zum Beispiel verschiedene Messenger, E-Mail, Excel und Ticketsysteme. Das System sammelt alle relevanten Informationen an einem Ort bzw. auf einer Plattform.

Das System soll Transparenz über Termine, Aufgaben, Finanzen und Teilnehmer bieten. Dabei soll das Softwareprodukt sowohl für private als auch kommerzielle Nutzung dienen. Weitere Anforderungen sind die Endgeräteunabhängigkeit und die Skalierbarkeit des Produktes. Das Produkt bietet eine Kombination aus Eventplanung, Finanzverwaltung und Dienstleisterintegration.

1.1.1 Produktnname und Begründung

Wir nennen das Softwareprodukt EventHub. Der Name ist leicht merkbar und simpel, spiegelt den Zweck der Software gut wieder und ist international verständlich. Außerdem ist er nicht zu technisch, aber gleichzeitig professionell.

1.2 Akteure und Rollen

1.2.1 Nicht registrierte / unangemeldete Akteure

Ein nicht angemeldeter Akteur hat folgende Optionen:

- Registrierung durchführen
- Login durchführen
- öffentliche Event- und Dienstleisterseiten aufrufen

1.2.2 Registrierte Akteure

Privatnutzer/Teilnehmer Das sind die Nutzer, die Events beitreten und daran teilnehmen. Sie können:

- sich registrieren/anmelden
- Gruppen beitreten/verlassen
- Notizen, Termine, Umfragen erstellen und daran teilnehmen
- abstimmen & kommentieren
- Einzahlungen vornehmen
- Aufgaben übernehmen
- Medien hochladen
- Kalender synchronisieren

Gruppenmitglied Ein Gruppenmitglied kann:

- Inhalte einer Gruppe lesen
- Notizen erstellen/bearbeiten/löschen
- an Umfragen teilnehmen
- Aufgaben übernehmen

- Einzahlungen vornehmen
- Multimedia hochladen
- Gruppe verlassen

Organisator/Gruppenleiter

- Gruppe/Event erstellen und löschen
- Mitglieder einladen/entfernen/sperren
- Beitragscode verwalten
- Mindestteilnehmerzahl festlegen
- Mindestbudget festlegen
- Aufgaben zuweisen
- Rückzahlungen veranlassen
- Eventseite verwalten
- Übersicht über Finanzen
- Moderationsmöglichkeiten

Institutionsadministrator Das können zum Beispiel Lehrer oder Firmenverantwortliche sein:

- mehrere Events verwalten
- Rollen und Berechtigungen vergeben
- Benutzerkonten innerhalb der Institution verwalten
- Übersicht über alle Veranstaltungen der Organisation
- Nutzungslimits überwachen
- Abrechnungen/Events archivieren

Institutionsmitglied

- meist nur Leserechte
- Teilnahme an Events
- Aufgaben ausführen

Veranstaltungs-Dienstleister

- eigenes Profil/Homepage verwalten
- Eventanfragen empfangen
- Angebote erstellen
- Rechnungen generieren
- Ressourcen planen
- Material und Personal verwalten
- Budgets kalkulieren
- durch Kunden bewerten lassen

Mitarbeiter eines Dienstleisters

- persönliche Verfügbarkeit pflegen
- Einsätze/Arbeitspläne einsehen
- Benachrichtigungen bei Zuweisung erhalten
- Qualifikationen verwalten

Kunde/Veranstalter (extern)

- Event anfragen
- Budget angeben
- Angebote erhalten
- Angebote annehmen und ablehnen
- Rechnungen bezahlen
- Dienstleister bewerten

1.2.3 Sekundäre Akteure

Externe Systeme

- Kalenderdienste (z. B. Outlook, Google Kalender)
- Kommunikationsdienste (E-Mail, SMS, Messenger)
- Zahlungsdienste
- Ticketing-Systeme
- Datenaustausch über APIs

Systemadministrator

- Benutzer sperren/verwalten
- Systemkonfiguration
- Backups/Wartung

1.3 Glossar der Fachdomäne

1.3.1 Event

Eine geplante Veranstaltung innerhalb des Systems, die organisatorische Informationen wie Termine, Aufgaben, Finanzen, Teilnehmer und Dokumentation bündelt.

1.3.2 Gruppe

Eine Menge von Benutzern, die gemeinsam ein Event planen oder daran teilnehmen. Gruppen dienen als organisatorische Einheit zur Zusammenarbeit.

1.3.3 Teilnehmer (Privatnutzer)

Ein registrierter Benutzer, der einer oder mehreren Gruppen beitritt und aktiv an Events teilnimmt, beispielsweise durch Abstimmungen, Aufgabenübernahme oder Einzahlungen.

1.3.4 Organisator (Gruppenleiter)

Ein Benutzer mit erweiterten Rechten innerhalb einer Gruppe, der Events erstellt, Mitglieder verwaltet und organisatorische Entscheidungen trifft.

1.3.5 Dienstleister

Ein externer Anbieter von Leistungen oder Ressourcen für Events, der Anfragen empfängt, Angebote erstellt, Rechnungen ausstellt und Material sowie Personal plant.

1.3.6 Aufgabe

Ein konkretes To-do innerhalb eines Events, das einer oder mehreren Personen zugewiesen werden kann und einen Status wie offen, in Bearbeitung oder erledigt besitzt.

1.3.7 Notiz

Ein freier Informationseintrag innerhalb eines Events zur Dokumentation von Ideen, Absprachen oder Hinweisen.

1.3.8 Umfrage

Ein strukturiertes Abstimmungswerkzeug mit mehreren Antwortoptionen, das zur Entscheidungsfindung innerhalb einer Gruppe genutzt wird.

1.3.9 Budget

Der für ein Event verfügbare Geldrahmen, bestehend aus Einzahlungen und Ausgaben, der zur Planung und Kontrolle der Finanzierung dient.

1.3.10 Rolle

Eine Sammlung von Berechtigungen, die festlegt, welche Funktionen ein Benutzer im System ausführen darf, beispielsweise Lesen, Bearbeiten oder Administrieren.

1.3.11 Institution

Eine organisatorische Einheit wie Schule oder Unternehmen, die mehrere Events verwaltet und eigene Administratoren besitzt.

1.3.12 Institutionsadministrator

Ein Benutzer mit globalen Verwaltungsrechten innerhalb einer Institution, der Rollen vergibt und Benutzerkonten verwaltet.

1.3.13 Ressource

Materielle oder personelle Mittel (z. B. Geräte oder Mitarbeiter), die einem Event zugeordnet und eingeplant werden.

2 Funktionale Anforderungen

2.1 Funktionale Gruppen Übersicht

- FG1 Account und Authentifizierung
- FG2 Profil und Einstellungen
- FG3 Gruppen- und Eventmanagement
- FG4 Notizen und Umfragen
- FG5 Finanzen
- FG6 Aufgabenmanagement
- FG7 Medien und Dokumentation

- FG8 Rollen- und Rechtesystem
- FG9 Öffentliche Eventseite und QR-Code
- FG10 Integration und Benachrichtigungen
- FG11 Dienstleister und Anfragen
- FG12 Angebote, Rechnungen und Mahnungen
- FG13 Ressourcenplanung

2.2 FG1 Account und Authentifizierung

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|-----------------------------|------------------------|--------------------------------------|----------------------------------|---|-----------|
| US-01 Registrierung | Gast | mich registrieren | ich das System nutzen kann | Pflichtfelder sind ausgefüllt und Account wird erstellt | Muss |
| US-02 Login | Benutzer | mich anmelden | ich Zugriff auf meine Daten habe | korrekte Zugangsdaten erlauben Anmeldung | Muss |
| US-03 Logout | Benutzer | mich abmelden | niemand mein Konto missbraucht | Sitzung wird beendet | Muss |
| US-04 Passwort zurücksetzen | Benutzer | mein Passwort zurücksetzen | ich wieder Zugriff bekomme | Reset-Link oder Code ermöglicht neues Passwort | Muss |
| US-05 2FA aktivieren | Benutzer | Zwei-Faktor-Authentifizierung nutzen | mein Account sicherer ist | zusätzlicher Faktor wird beim Login geprüft | Sollte |

2.3 FG2 Profil und Einstellungen

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|--------------------------------|------------------------|--|-----------------------------------|-------------------------------------|-----------|
| US-06 Profil anzeigen | Benutzer | mein Profil sehen | ich meine Daten prüfen kann | gespeicherte Daten werden angezeigt | Muss |
| US-07 Profil bearbeiten | Benutzer | meine Daten ändern | sie aktuell bleiben | Änderungen werden gespeichert | Muss |
| US-08 Kommunikationsweg wählen | Benutzer | bevorzugte Benachrichtigungen einstellen | ich passende Nachrichten erhalten | gewählter Kanal wird gespeichert | Sollte |
| US-09 Sprache wählen | Benutzer | die Sprache ändern | ich die Oberfläche verstehen | UI-Texte werden angepasst | Sollte |

2.4 FG3 Gruppen- und Eventmanagement

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|-----------------------------------|------------------------|-----------------------------------|---|--|-----------|
| US-10 Gruppe erstellen | Organisator | eine Gruppe anlegen | ich ein Event planen kann | Name und Beschreibung werden gespeichert | Muss |
| US-11 Gruppe löschen | Organisator | meine Gruppe löschen | alte Events entfernt werden | Gruppe wird archiviert oder gelöscht | Muss |
| US-12 Gruppe beitreten | Benutzer | einer Gruppe beitreten | ich teilnehmen kann | gültiger Beitrittscode erlaubt Beitritt | Muss |
| US-13 Gruppe verlassen | Benutzer | eine Gruppe verlassen | ich nicht mehr beteiligt bin | Mitgliedschaft endet | Muss |
| US-14 Mitglieder anzeigen | Benutzer | alle Mitglieder sehen | ich Überblick habe | Liste wird angezeigt | Muss |
| US-15 Mitglieder verwalten | Organisator | Mitglieder entfernen oder sperren | Regeln durchgesetzt werden | Zugriff wird entzogen | Muss |
| US-16 Mindestteilnehmer festlegen | Organisator | eine Mindestzahl definieren | Event nur bei genug Teilnehmern stattfindet | Status wird automatisch geprüft | Sollte |

2.5 FG4 Notizen und Umfragen

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|-------------------------|------------------------|-------------------------------------|------------------------------------|-------------------------------------|-----------|
| US-17 Notiz erstellen | Benutzer | eine Notiz anlegen | Infos dokumentiert sind | Notiz erscheint in der Liste | Muss |
| US-18 Notiz bearbeiten | Benutzer | Notizen ändern | Fehler korrigiert werden | Änderungen werden gespeichert | Muss |
| US-19 Notiz löschen | Benutzer | Notizen entfernen | irrelevante Infos verschwinden | Notiz wird entfernt | Muss |
| US-20 Notizen filtern | Benutzer | nach Typ filtern | ich schneller finde, was ich suche | Filter zeigt passende Einträge | Sollte |
| US-21 Termin erfassen | Benutzer | Termine mit Datum und Ort speichern | alle informiert sind | Termin enthält strukturierte Felder | Muss |
| US-22 Umfrage erstellen | Benutzer | eine Umfrage anlegen | Entscheidungen getroffen werden | Antwortoptionen frei wählbar | Muss |
| US-23 Abstimmen | Benutzer | an Umfragen teilnehmen | meine Meinung zählt | Stimme wird gezählt | Muss |

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|-------------------------|------------------------|------------------|---------------------|----------------------------------|-----------|
| US-24 Ergebnis anzeigen | Benutzer | Ergebnisse sehen | Transparenz besteht | Stimmenverteilung wird angezeigt | Muss |

2.6 FG5 Finanzen

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|------------------------------|------------------------|-----------------------------------|----------------------------|---|-----------|
| US-25 Einzahlung tätigen | Benutzer | Geld einzahlen | Event finanziert wird | Betrag wird gespeichert | Muss |
| US-26 Zahlungsstand anzeigen | Benutzer | Gesamtbetrag sehen | ich Überblick habe | eingegangenes Geld und Zielsumme werden angezeigt | Muss |
| US-27 Ausgabe erfassen | Benutzer | Ausgaben dokumentieren | Kosten transparent sind | Betrag und Beschreibung werden gespeichert | Muss |
| US-28 Kas senübersicht | Organisator | alle Einnahmen und Ausgaben sehen | ich abrechnen kann | Summen werden korrekt berechnet | Muss |
| US-29 Rückzahlung auslösen | Organisator | Rückzahlungen durchführen | Guthaben ausgeglichen wird | Rückzahlung wird dokumentiert | Sollte |

2.7 FG6 Aufgabenmanagement

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|---------------------------|------------------------|---------------------------------|--------------------------|--|-----------|
| US-30 Aufgabe erstellen | Organisator | Aufgaben anlegen | To-dos geplant sind | Aufgabe erscheint in Liste | Muss |
| US-31 Aufgabe übernehmen | Benutzer | Aufgaben übernehmen | Zuständigkeit klar ist | Aufgabe wird mir zugeordnet | Muss |
| US-32 Aufgabe abschließen | Benutzer | Aufgaben als erledigt markieren | Fortschritt sichtbar ist | Status ändert sich auf erledigt | Muss |
| US-33 Erinnerung senden | System | an offene Aufgaben erinnern | nichts vergessen wird | Benachrichtigung wird automatisch gesendet | Sollte |

2.8 FG7 Medien und Dokumentation

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität | |
|---------|------------------------|------------------|--|----------------------------|--|--------|
| US-34 | Medien hochladen | Benutzer | Fotos oder Videos hochladen | das Event dokumentiert ist | Dateien werden gespeichert und angezeigt | Sollte |
| US-35 | Kommentare schreiben | Benutzer | Bewertungen oder Kommentare hinterlassen | Erfahrungen geteilt werden | Text wird gespeichert | Könnte |

2.9 FG8 Rollen- und Rechtesystem

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität | |
|---------|-------------------------|--------------------|------------------|---------------------------|--|--------|
| US-36 | Rollen vergeben | Institutions-admin | Rollen zuweisen | Rechte gesteuert werden | Berechtigungen werden gespeichert | Muss |
| US-37 | Leserechte einschränken | Institutions-admin | nur lesen können | Regeln eingehalten werden | Schreibfunktionen deaktiviert | Muss |
| US-38 | Lizenzlimit prüfen | Institutions-admin | Events begrenzen | Community-Regeln gelten | mehr als drei Events werden verhindert | Sollte |

2.10 FG9 Öffentliche Eventseite und QR-Code

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität | |
|---------|------------------------|------------------|------------------------------|----------------------------------|---------------------------|--------|
| US-39 | Eventseite anzeigen | Organisator | eine öffentliche Seite haben | ich das Event bewerben kann | feste URL ist erreichbar | Sollte |
| US-40 | QR-Code generieren | Benutzer | einen QR-Code erhalten | ich den Link schnell teilen kann | Code wird korrekt erzeugt | Sollte |

2.11 FG10 Integration und Benachrichtigungen

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität | |
|---------|--------------------------|------------------|-------------------------|--|------------------------------|--------|
| US-41 | Kalender synchronisieren | Benutzer | Termine synchronisieren | sie automatisch im Kalender erscheinen | Synchronisation funktioniert | Könnte |

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|-------------------------------------|------------------------|---|---|--|-----------|
| US-42 Benachrichtigungen senden | System | Nachrichten versenden | Nutzer informiert bleiben | Versand über gewählten Kanal erfolgt | Sollte |
| US-42a Sprachassistent verbinden | Benutzer | einen digitalen Sprachassistenten koppeln | ich per Sprache auf meine Events zugreifen kann | Verbindung wird autorisiert und Konto ist verknüpft | Könnte |
| US-42b Termine per Sprache abfragen | Benutzer | meine nächsten Termine per Sprachbefehl abrufen | ich Informationen freihändig erhalten kann | Assistent gibt korrekte Termine aus | Könnte |
| US-42c Aktion per Sprache ausführen | Benutzer | einfache Aktionen wie Aufgaben abhaken oder Einzahlungen bestätigen per Sprache ausführen | ich schnell reagieren kann ohne die App zu öffnen | Aktion wird korrekt ausgeführt und im System gespeichert | Könnte |

2.12 FG11 Dienstleister und Anfragen

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|--------------------------|------------------------|-------------------------------|--------------------------------|---------------------------------------|-----------|
| US-43 Anfrage stellen | Kunde | ein Event anfragen | ich ein Angebot erhalte | Formular speichert alle Pflichtdaten | Muss |
| US-44 Angebot erstellen | Dienstleister | ein Angebot generieren | ich Kunden informieren kann | Angebotsdokument wird erstellt | Muss |
| US-45 Rechnung erstellen | Dienstleister | eine Rechnung erzeugen | ich abrechnen kann | Rechnung enthält alle Pflichtangaben | Muss |
| US-46 Zahlung erinnern | System | an offene Rechnungen erinnern | Zahlungen rechtzeitig eingehen | Erinnerung wird automatisch versendet | Sollte |

2.13 FG12 Ressourcenplanung

| Name/ID | In meiner Rolle als... | ...möchte ich... | ..., so dass... | Akzeptiert, wenn... | Priorität |
|-----------------------------|------------------------|-------------------------------|----------------------------------|--|-----------|
| US-47 Material verwalten | Dienstleister | Material anlegen oder ändern | Bestand aktuell bleibt | Materialdaten werden gespeichert | Muss |
| US-48 Material buchen | Dienstleister | Material einem Event zuordnen | Doppelbuchungen vermieden werden | Material ist im Zeitraum gesperrt | Muss |
| US-49 Mitarbeiter verwalten | Dienstleister | Mitarbeiter planen | Personal verfügbar ist | Einsatzplan berücksichtigt Verfügbarkeit | Muss |
| US-50 Wartungshinweis | System | Wartungen anzeigen | Geräte sicher bleiben | Hinweis erscheint bei fälligem Zyklus | Sollte |

3 Nicht-funktionale Anforderungen (ISO 25010)

3.1 Performance Efficiency

Die Plattform soll gemäß der Aufgabenstellung auch bei sehr vielen gleichzeitigen Benutzern eine hohe Leistungsfähigkeit aufweisen. Das System muss skalierbar sein und darf auch bei paralleler Nutzung mehrerer Gruppen und Events keine spürbaren Verzögerungen zeigen. Aktionen sollen innerhalb von maximal zwei Sekunden verarbeitet werden und der Seitenaufbau soll auch bei normaler Netzwerklatenz nur wenige Sekunden dauern. Das System soll mehrere tausende gleichzeitige Benutzer unterstützen und bei steigender Last einfach weiter skalierbar sein, z. B. durch weitere Serverinstanzen.

Zusätzlich werden zur Sicherstellung einer stabilen Performance sinnvolle ergänzende Maßnahmen definiert. Ressourcenintensive Prozesse wie Datei-Uplands oder größere Berechnungen sollen asynchron im Hintergrund ausgeführt werden, damit die Benutzeroberfläche weiterhin reaktionsfähig bleibt. Der Ressourcenverbrauch von CPU, Arbeitsspeicher und Datenbank soll überwacht und regelmäßig optimiert werden, um auch bei wachsendem Datenbestand eine gleichbleibende Leistung zu gewährleisten.

3.2 Compatibility

Das System soll auf unterschiedlichen Endgeräten und Softwareumgebungen nutzbar sein. Gemäß der Aufgabenstellung muss die Anwendung sowohl im Webbrowser als auch auf mobilen Endgeräten lauffähig sein und funktional konsistent sein. Alle wesentlichen Funktionen der Webversion sollen somit auch in der mobilen Anwendung in gleicher oder gleichwertiger Form zur Verfügung stehen.

Die Webanwendung soll in aktuellen Versionen der gängigen Browser Google Chrome, Microsoft Edge, Mozilla Firefox und Safari ohne zusätzliche Plugins oder Erweiterungen nutzbar sein. Für mobile Endgeräte soll eine Unterstützung der Betriebssysteme Android und iOS gewährleistet werden.

Darüber hinaus soll das System die Integration externer Dienste ermöglichen. Es sollen Kalenderdienste wie Google Calendar, Outlook und iCal angebunden werden können, sodass Termine automatisch synchronisiert werden. Weitere Schnittstellen, beispielsweise zu Zahlungs- oder Ticketingsystemen, sollen über REST-APIs realisiert werden.

Ergänzend werden folgende technische Maßnahmen geplant, um eine hohe Geräteunabhängigkeit sicherzustellen: Die Benutzeroberfläche soll responsiv gestaltet sein und sich automatisch an unterschiedliche Bildschirmgrößen und Auflösungen anpassen. Damit wird die explizit geforderte Endgeräteunabhängigkeit umgesetzt und durch zusätzliche Maßnahmen zur praktischen Nutzbarkeit abgesichert.

3.3 Usability (Gebrauchstauglichkeit)

Die Anwendung soll gemäß der Aufgabenstellung ohne lange Einarbeitung nutzbar sein. Neue Nutzer sollen grundlegende Funktionen wie das Erstellen oder Beitreten von Gruppen, das Anlegen von Notizen sowie das Verwalten von Aufgaben und Zahlungen intuitiv bedienen können, ohne zuvor eine Schulung oder umfangreiche Anleitung zu benötigen.

Die Benutzeroberfläche soll klar strukturiert und konsistent aufgebaut sein. Navigationselemente, Bezeichnungen und Strukturen sollen über alle Bereiche der Anwendung hinweg einheitlich verwendet werden, damit sich Nutzer schnell orientieren können. Zentrale Funktionen sollen mit wenigen Interaktionen erreichbar sein, sodass typische Aufgaben ohne nervige Zwischenschritte ausgeführt werden können.

Systemmeldungen und Fehlermeldungen sollen verständlich formuliert sein und dem Nutzer konkrete Hinweise zur Behebung von Problemen geben. Unterschiedliche Inhalte, wie beispielsweise Notiztypen oder Statusinformationen, sollen visuell eindeutig unterscheidbar dargestellt werden, etwa durch Farben oder Symbole.

Zusätzlich werden sinnvolle ergänzende Maßnahmen zur Verbesserung der Gebrauchstauglichkeit berücksichtigt. Die Oberfläche soll responsiv gestaltet sein, ausreichend große Bedienelemente besitzen und auch auf mobilen Geräten gut lesbar und bedienbar bleiben. Damit wird die explizit geforderte einfache Nutzbarkeit umgesetzt und durch weitere Maßnahmen zur praktischen Benutzerfreundlichkeit ergänzt.

3.4 Reliability (Zuverlässigkeit)

Das System soll zuverlässig betrieben werden können und die zentralen Funktionen der Plattform dauerhaft bereitstellen. Da über die Anwendung unter anderem organisatorische Informationen und Finanzdaten (Einzahlungen und Ausgaben) verwaltet werden, muss eine konsistente Speicherung dieser Daten gewährleistet sein, sodass keine Daten durch technische Störungen verloren gehen oder in einen widersprüchlichen Zustand geraten.

Zur Konkretisierung werden folgende Qualitätsziele festgelegt: Änderungen an relevanten Daten (z. B. Zahlungen, Ausgaben, Aufgabenstatus) sollen dauerhaft und konsistent gespeichert werden. Bei temporären Netzwerkproblemen oder kurzfristigen Serverstörungen sollen Benutzeraktionen nicht zu unvollständigen oder doppelten Einträgen führen. Außerdem sollen regelmäßig automatische Datensicherungen durchgeführt werden, sodass eine Wiederherstellung nach Systemausfällen möglich ist.

Ergänzend werden sinnvolle technische Maßnahmen definiert, um die Zuverlässigkeit

praktisch abzusichern. Fehlerzustände sollen protokolliert werden (Logging), und kritische Systemzustände sollen überwacht werden (Monitoring), um Ausfälle frühzeitig zu erkennen. Für zentrale Prozesse sollen geeignete Mechanismen vorgesehen werden, die eine sichere Verarbeitung auch bei hoher Last unterstützen.

3.5 Security (Sicherheit)

Die Aufgabenstellung legt einen besonderen Fokus auf Sicherheitsanforderungen. Die Datenkommunikation zwischen Client und Server muss verschlüsselt erfolgen, damit Inhalte nicht von Unbefugten abgefangen werden können. Passwörter dürfen nicht im Klartext gespeichert werden und müssen in der Datenbank ausschließlich in gehashter und gesalzener Form vorliegen. Zusätzlich sollen Nutzer zu starken Passwörtern ermutigt werden und es soll eine Zwei-Faktor-Authentifizierung, beispielsweise per SMS oder vergleichbar, unterstützt werden.

Weiterhin ist aufgrund des Rollen- und Rechtesystems sicherzustellen, dass Benutzer nur auf diejenigen Gruppen- und Eventdaten zugreifen können, für die sie berechtigt sind. Insbesondere müssen administrative Aktionen wie das Entfernen oder Sperren von Mitgliedern auf Organisatoren bzw. Institutionsadministratoren beschränkt sein.

Ergänzend werden technische Schutzmaßnahmen festgelegt, die in modernen Web- und App-Systemen üblich sind. Dazu zählen der Schutz vor typischen Angriffen wie SQL-Injection, Cross-Site Scripting (XSS) und Cross-Site Request Forgery (CSRF) sowie sichere Sitzungsverwaltung (z. B. Timeout bei Inaktivität). Sicherheitsrelevante Ereignisse wie fehlgeschlagene Loginversuche oder Änderungen sicherheitskritischer Einstellungen sollen protokolliert werden.

3.6 Maintainability (Wartbarkeit)

Für eine langfristige Nutzung und Weiterentwicklung soll die Plattform wartbar und gut erweiterbar sein. Gemäß der Aufgabenstellung wird ein Client-Server-System umgesetzt, das auf mobilen Endgeräten und im Webbrowser verfügbar ist. Daraus ergibt sich eine klare Trennung zwischen Client, Server und Datenhaltung, sodass Änderungen an einzelnen Komponenten möglichst isoliert vorgenommen werden können.

Zur Konkretisierung werden folgende Qualitätsziele festgelegt: Schnittstellen zwischen den Systemkomponenten sollen klar definiert und dokumentiert sein. Insbesondere ist eine dokumentierte REST-API bereitzustellen, die die Kommunikation zwischen Client und Server nachvollziehbar beschreibt. Die Struktur des Systems soll so gestaltet sein, dass neue Funktionen (z. B. zusätzliche Integrationen oder neue Notiztypen) ohne grundlegende Umstrukturierung ergänzt werden können.

Ergänzend werden Maßnahmen vorgesehen, die Wartbarkeit praktisch unterstützen. Dazu gehören ein konsistenter Code-Stil, eine nachvollziehbare Struktur der Komponenten sowie die Möglichkeit, zentrale Funktionen automatisiert zu testen. Konfigurationswerte (z. B. Rollenrechte oder Systemparameter) sollen soweit sinnvoll ohne Codeänderungen anpassbar sein.

3.7 Portability (Übertragbarkeit)

Das System soll auf aktuellen Plattformen betrieben werden können und nicht von spezieller Hardware abhängig sein. Da die Anwendung sowohl im Webbrowser als auch auf mobilen Endgeräten genutzt wird, soll die Lösung auf den gängigen Betriebssystemen lauffähig sein. Dazu gehören insbesondere Android und iOS für mobile Geräte sowie aktuelle Desktop-Umgebungen für die Webnutzung.

Zur Konkretisierung wird festgelegt, dass die Serverkomponenten auf unterschiedlichen Serverumgebungen betrieben werden können sollen. Das Deployment soll standardisiert möglich sein, sodass ein Umzug zwischen unterschiedlichen Umgebungen (z. B. verschiedene Hosting-Anbieter) ohne große Anpassungen realisierbar bleibt.

Ergänzend werden sinnvolle technische Maßnahmen berücksichtigt, um die Übertragbarkeit zu erhöhen. Dazu zählen standardisierte Deploymentverfahren (z. B. Containerisierung) sowie der Export und Import von Daten in gängigen Formaten, sofern dies für administrative oder datenschutzbezogene Zwecke erforderlich ist.

3.8 Datenschutz (nicht aus der ISO)

Neben den ISO-Qualitätsmerkmalen sind in der Aufgabenstellung explizite Anforderungen an Datenschutz und Vertraulichkeit genannt. Nutzerdaten müssen gemäß Datenschutzgrundverordnung (DSGVO) behandelt werden, und die Datenschutzrichtlinie muss von den Nutzern bestätigt werden, bevor die Software vollständig genutzt werden kann. Private Informationen, insbesondere Kontaktinformationen und ausgetauschte Nachrichten, dürfen nicht in falsche Hände geraten. Die Aufgabenstellung fordert daher eine verschlüsselte Speicherung und einen verschlüsselten Austausch sensibler Daten; darüber hinaus wird ein Ende-zu-Ende-Verschlüsselungssystem als mögliche Lösung genannt.

Zur Konkretisierung werden folgende Anforderungen festgelegt: Es soll nur die für den Betrieb notwendige Menge personenbezogener Daten gespeichert werden (Datenminimierung). Nutzer sollen ihre eigenen Daten einsehen können und es soll ein Verfahren vorgesehen werden, um personenbezogene Daten auf Anfrage zu löschen. Der Zugriff auf personenbezogene Daten muss durch geeignete Berechtigungen geschützt werden.

Ergänzend werden sinnvolle Maßnahmen vorgesehen, um Datenschutzanforderungen praktisch umzusetzen. Dazu gehören ein nachvollziehbares Lösch- und Aufbewahrungskonzept für Daten, sowie eine sichere Protokollierung, ohne dabei vertrauliche Inhalte im Klartext zu speichern.

4 Graphische Benutzungsschnittstelle

4.1 Designprinzipien und UX-Grundlagen

4.1.1 Zielsetzung der Benutzeroberfläche

Die Benutzeroberfläche von EventHub soll eine einfache und intuitive Bedienung der Plattform ermöglichen. Nutzer sollen typische Aufgaben wie Gruppen beitreten, Events verwalten, Aufgaben bearbeiten oder Zahlungen durchführen ohne Schulung ausführen können. Die Oberfläche ist daher klar strukturiert, reduziert gestaltet und auf die wesentlichen Funktionen fokussiert.

4.1.2 Konsistenz und Navigationskonzept

Alle Bereiche der Anwendung verwenden ein einheitliches Layout und wiederkehrende Navigationselemente. Gleichartige Funktionen sind an denselben Positionen erreichbar, sodass sich Nutzer schnell orientieren können. Zentrale Aktionen sind direkt sichtbar und nicht in tiefen Menüstrukturen versteckt. Häufig genutzte Funktionen sind mit maximal zwei bis drei Interaktionen erreichbar.

4.1.3 Rollenbasierte Darstellung

Die Oberfläche ist rollenabhängig gestaltet. Benutzer sehen ausschließlich Funktionen, für die sie berechtigt sind. Teilnehmer erhalten eine vereinfachte Ansicht mit Fokus auf Teilnahme und Interaktion, während Organisatoren und Institutionsadministratoren zusätzliche Verwaltungsfunktionen erhalten. Dadurch wird die Komplexität reduziert und Fehlbedienung vermieden.

4.1.4 Responsives Design und Geräteunabhängigkeit

Die Anwendung ist responsiv gestaltet und passt sich automatisch an unterschiedliche Bildschirmgrößen und Endgeräte an. Die Webversion ist für Desktop-Browser optimiert, während mobile Endgeräte eine vereinfachte Navigation und größere Bedienelemente erhalten. Alle wesentlichen Funktionen stehen auf beiden Plattformen in gleichwertiger Form zur Verfügung.

4.1.5 Fehlervermeidung und Systemfeedback

Pflichtfelder werden eindeutig gekennzeichnet und Eingaben vor dem Speichern validiert. Kritische Aktionen wie Löschen oder Rückzahlungen werden durch Bestätigungsdialoge abgesichert. Das System informiert Nutzer jederzeit über den aktuellen Status, beispielsweise durch Ladeanzeigen, Erfolgsmeldungen oder verständliche Fehlermeldungen. Dadurch bleibt das Systemverhalten nachvollziehbar und transparent.

4.1.6 Barrierearme Bedienbarkeit

Texte sind gut lesbar gestaltet und Bedienelemente ausreichend groß dimensioniert, sodass die Anwendung auch auf mobilen Geräten oder Touchscreens komfortabel nutzbar ist. Kontraste und klare Beschriftungen unterstützen eine schnelle Erfassbarkeit der Inhalte.

4.2 Navigationsmodelle

Die folgenden Zustandsdiagramme zeigen Navigation und mögliche Zustandsübergänge der Benutzeroberfläche.

4.2.1 Zustandsdiagramm – Privatnutzer

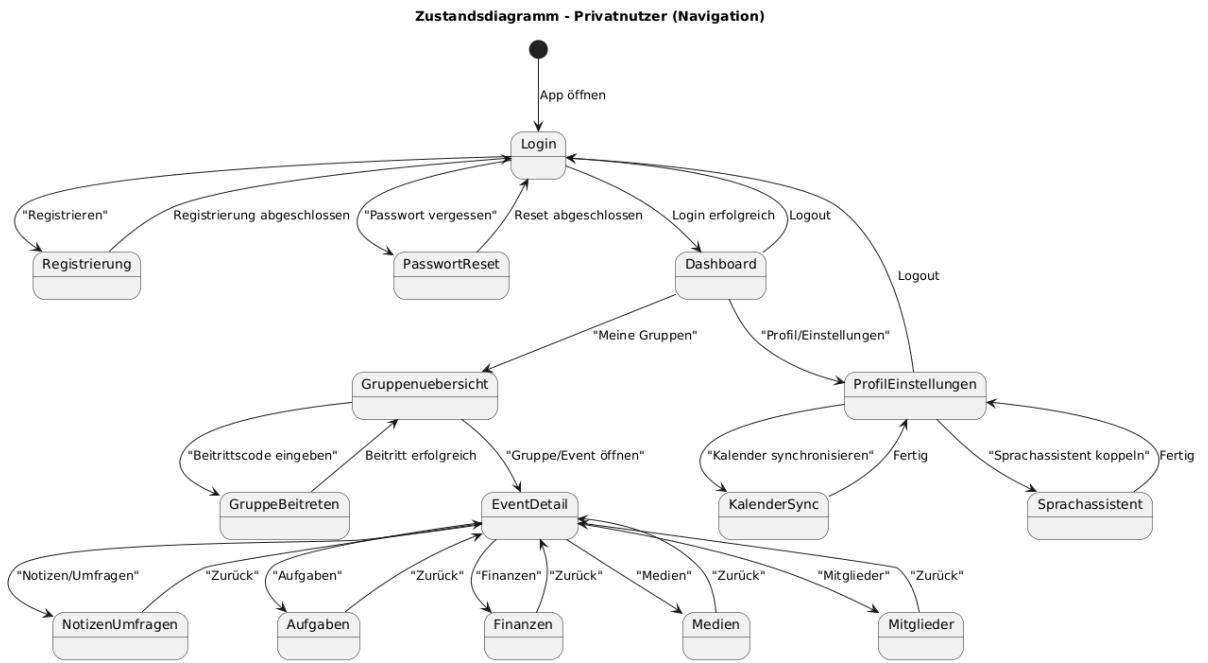


Abbildung 1: Zustandsdiagramm Privatnutzer

4.2.2 Zustandsdiagramm – Organisator

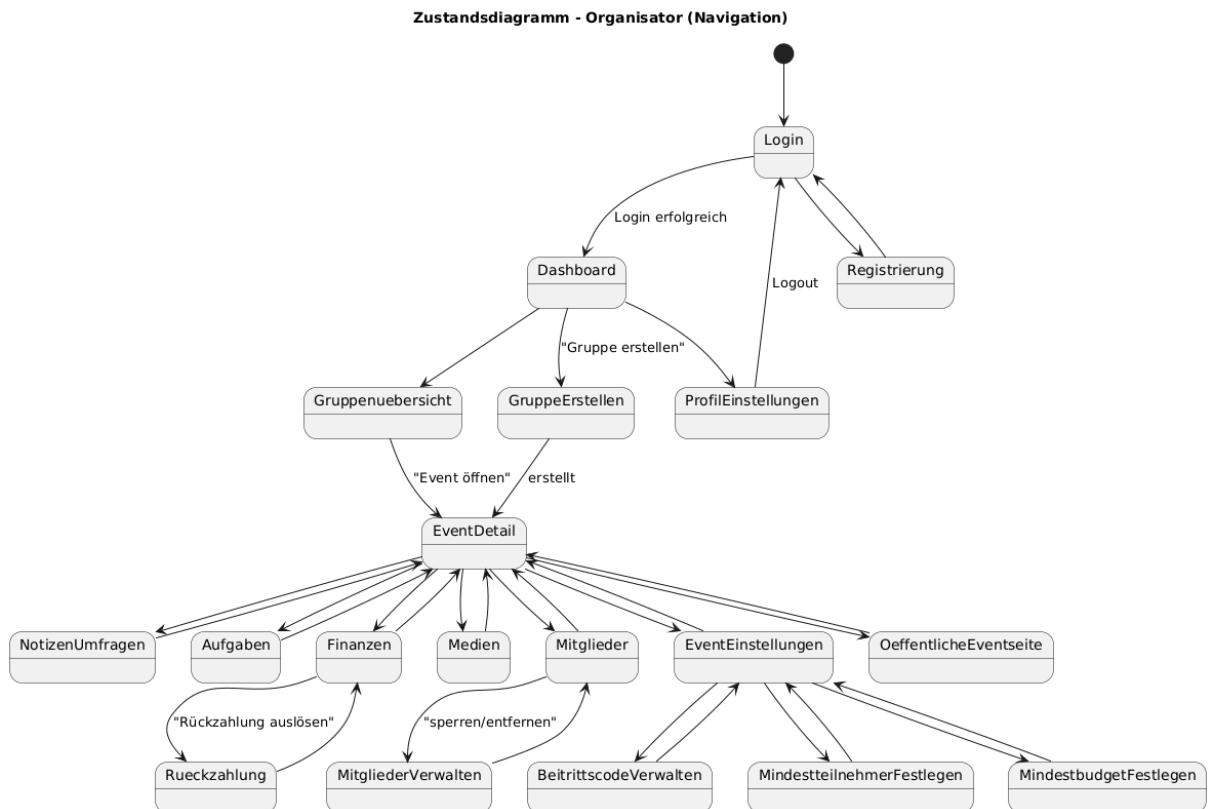


Abbildung 2: Zustandsdiagramm Organisator

4.2.3 Zustandsdiagramm – Dienstleister

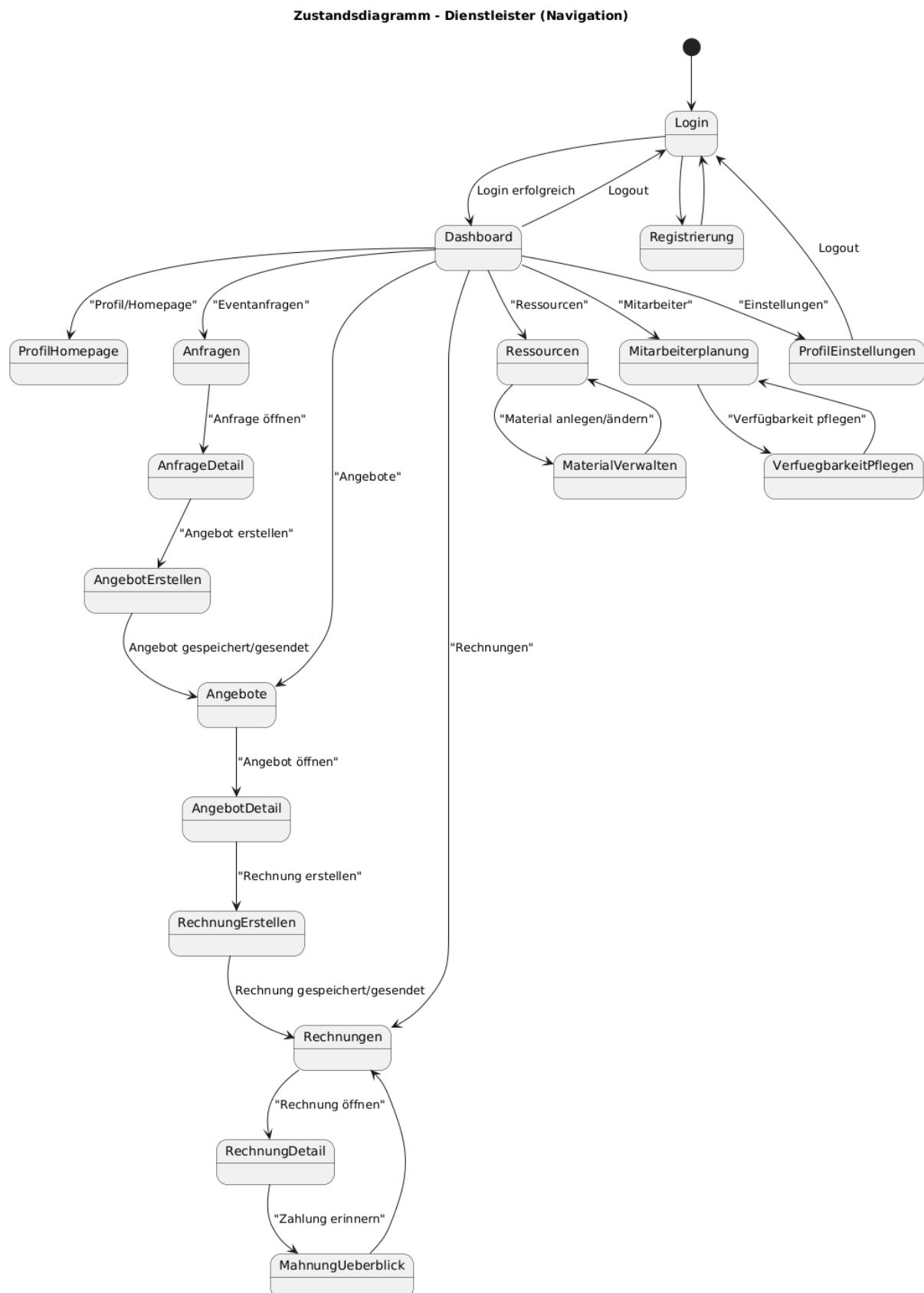


Abbildung 3: Zustandsdiagramm Dienstleister

4.3 Webbrowser-Mockups

4.3.1 Login & Registrierung

The UI-Mockup for the EventHub login and registration page is displayed. At the top, there is a logo consisting of two overlapping squares and the text "EventHub". Below the logo, the word "Login" is centered above a horizontal line. The form is divided into sections:

- E-Mail:** A text input field.
- Passwort:** A text input field.
- Login:** A large, prominent button.
- Checkboxes:** Three buttons: "Angemeldet bleiben" (checkbox), "Registrieren", and "Passwort vergessen".
- 2-Faktor-Authentifizierung Code:** A text input field.
- Anmelden:** A button at the bottom right.

Abbildung 4: UI-Mockup Login und Registrierung (US-01, US-02, US-03, US-04, US-05)

4.3.2 Dashboard

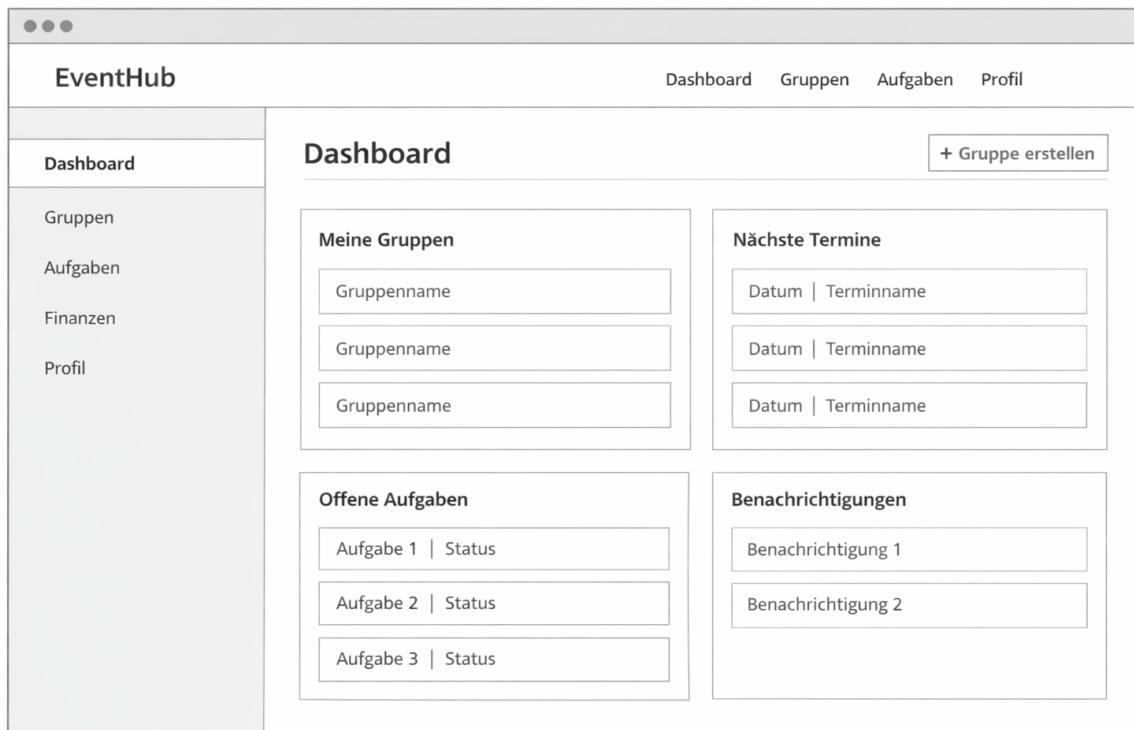


Abbildung 5: UI-Mockup Dashboard (US-10, US-12, US-30, US-42)

4.3.3 Gruppenübersicht

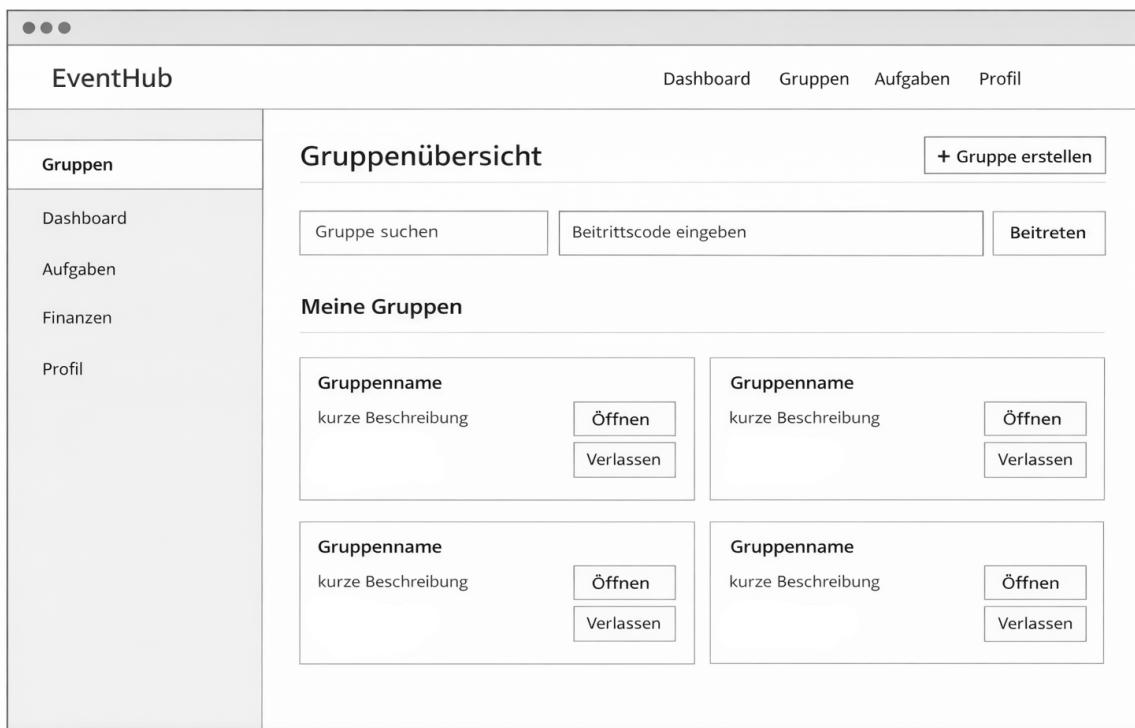


Abbildung 6: UI-Mockup Gruppenübersicht (US-10, US-11, US-12, US-13, US-14)

4.3.4 Event-Detailseite

The UI-Mockup for the Event-Detailseite (EventHub) is displayed. At the top, there is a header bar with three dots on the left and navigation links: Dashboard, Gruppen, Aufgaben, and Profil on the right. Below the header is a sidebar on the left containing links: Übersicht, Notizen/Umfragen, Aufgaben, Finanzen, Medien, and Mitglieder. The main content area is titled "Event-Detail". It shows an "Event-Titel | Ort, Datum" section with three buttons: "+ Notiz erstellen", "+ Aufgabe hinzufügen", and "Mitglieder verwalten". Below this is an "Übersicht" section with three boxes labeled "Notiz / Aufgabe 1", "Notiz / Aufgabe 2", and "Notiz / Aufgabe 3", each containing the placeholder text "Beschreibung der Notiz /Aufgabe."

Abbildung 7: UI-Mockup Event-Detailseite (US-14, US-16, US-17–US-24, US-25–US-32, US-34, US-35, US-39, US-40)

4.3.5 Finanzen

The UI-Mockup displays the EventHub application interface. The top navigation bar includes three dots, the title "EventHub", and links to "Dashboard", "Gruppen", "Aufgaben", and "Profil". On the left, a sidebar lists "Übersicht", "Notizen/Umfragen", "Aufgaben", "Finanzen" (which is currently selected), "Medien", and "Mitglieder". The main content area is titled "Finanzen" and contains three summary boxes: "Budget-Ziel", "Aktueller Stand", and "Restbetrag". Below these are two buttons: "+ Einzahlung" and "- Rückzahlung". A table lists financial transactions:

| Datum | Typ | Beschreibung | Betrag | Bearbeiten | Löchen |
|------------|-------------|--------------|----------|------------|--------|
| TT.MM.JJJJ | Einzahlung | | 100,00 € | Bearbeiten | Löchen |
| TT.MM.JJJJ | Einzahlung | | 100,00 € | Bearbeiten | Löchen |
| TT.MM.JJJJ | Rückzahlung | | -50,00 € | Bearbeiten | Löchen |

Abbildung 8: UI-Mockup Finanzen (US-25, US-26, US-27, US-28, US-29)

4.3.6 Notizen & Umfragen

The UI-Mockup displays the 'EventHub' application interface. The top navigation bar includes three dots, the 'EventHub' logo, and links for 'Dashboard', 'Gruppen', 'Aufgaben', and 'Profil'. A vertical sidebar on the left contains links for 'Übersicht', 'Notizen/Umfragen', 'Aufgaben', 'Medien', and 'Mitglieder'. The main content area is titled 'Notizen & Umfragen' and features a 'Notizen & Umfragen' button. Below it are two buttons: '+ Notiz erstellen' and '+ Umfrage erstellen'. There are also 'Typ filtern' and 'Suchen' search filters. The main list displays four entries, each consisting of a title, a short description, and actions: 'Bearbeiten', 'Löschen', and 'Öffnen'. The first entry shows 'Titel der Notiz/Umfrage', 'Kurzer Beschreibungstext', and a status section with '45 Stimmen' and 'Ja'/'Nein' buttons. The other three entries follow a similar structure.

Abbildung 9: UI-Mockup Notizen und Umfragen (US-17, US-18, US-19, US-20, US-21, US-22, US-23, US-24)

4.3.7 Aufgabenplanung

The UI-Mockup shows the 'Aufgabenmanagement' (Task Management) section of the EventHub application. On the left, a sidebar menu lists 'Übersicht', 'Notizen/Umfragen', 'Aufgaben' (selected), 'Finanzen', 'Medien', and 'Mitglieder'. The main area has a header 'Aufgabenmanagement' with a 'Status filtern' dropdown, a search bar, and a 'Suchen' button. A large 'Offen' section displays a task card with 'Aufgabenname', 'kurze Beschreibung', and 'Zugewiesen: Name' fields, along with 'Übernehmen', 'Bearbeiten', and 'Erledigt' buttons. Below it are sections for 'In Bearbeitung' and 'Erledigt', each with similar task cards. A 'Neue Aufgabe erstellen' button is located at the top right of the main area.

Abbildung 10: UI-Mockup Aufgabenübersicht (US-30, US-31, US-32, US-33)

4.3.8 Profil & Einstellungen

The UI-Mockup shows the 'EventHub' application interface. At the top, there is a navigation bar with three dots on the left, followed by the 'EventHub' logo, and menu items: Dashboard, Gruppen, Aufgaben, and Profil (which is highlighted in bold). On the left side, there is a vertical sidebar with links: Übersicht, Notizen/Umfragen, Aufgaben, Finanzen, and Profil (also highlighted in bold). The main content area is titled 'Profil & Einstellungen'. It contains several sections: 'Profilinformationen' with fields for Name, E-Mail, and Passwort ändern, and a 'Speichern' button; 'Benachrichtigungen' with checkboxes for E-Mail, Push, and SMS; 'Sprache' with a dropdown menu labeled 'Sprache auswählen'; and 'Integrationen' with buttons for 'Kalender synchronisieren' and 'Sprachassistent koppeln', each accompanied by a 'Verbinden' button. A 'Logout' button is located at the bottom left of the main content area.

Abbildung 11: UI-Mockup Profil & Einstellungen (US-03, US-06, US-07, US-08, US-09, US-41, US-42a, US-42b, US-42c)

4.3.9 Dienstleister-Dashboard

The UI-Mockup for the Dienstleister Dashboard is displayed within a main window titled "EventHub". The top navigation bar includes "EventHub", "Dashboard", "Anfragen", "Angebote", "Rechnungen", "Ressourcen", and "Profil". A sidebar on the left lists "Dashboard", "Anfragen", "Angebote", "Rechnungen", "Material", "Mitarbeiter", and "Profil". The main content area is titled "Dienstleister Dashboard" and contains four primary sections:

- Neue Anfragen**: Three items listed with "Öffnen" buttons:
 - Eventname TT.MM.JJJJJ
 - Eventname TT.MM.JJJJJ
 - Eventname TT.MM.JJJJJ
- Offene Angebote**: Two items listed with "Status" and "Öffnen" buttons:
 - 1.000,00 € Status Öffnen
 - 800,00 € austehend Öffnen
- Offene Rechnungen**: Two items listed with "Fällig bis TT.MM.JJJJJ" and "Öffnen" buttons:
 - 1.500,00 € Fällig bis TT.MM.JJJJJ Öffnen
 - 750,00 € Fällig bis TT.MM.JJJJJ Öffnen
- Ressourcenstatus**: Three items listed with status indicators:
 - Material 1 Verfügbar
 - Material 2 Nicht verfügbar
 - Mitarbeiter A Verfügbar

Abbildung 12: UI-Mockup Dienstleister-Dashboard (US-43, US-44, US-45, US-46, US-47, US-48, US-49)

4.3.10 Rollen- und Rechteverwaltung

The UI-Mockup shows a dashboard titled "EventHub" with navigation links for Dashboard, Events, Benutzer, and Admin. On the left, a sidebar lists Übersicht, Events, Benutzer, and Rollen & Rechte. The main content area is titled "Rollen- und Rechteverwaltung" and includes a dropdown menu "Institution auswählen". A button "+ Rolle vergeben" is located in the top right. Below is a table with columns: Benutzer, Rolle, Lesen, and Schreiben. The table contains six rows of data. A "Speichern" button is at the bottom right.

| Benutzer | Rolle | Lesen | Schreiben |
|--------------|------------|-------------------------------------|-------------------------------------|
| Benutzername | Rollenname | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Benutzername | Rollenname | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Benutzername | Rollenname | <input type="checkbox"/> | <input type="checkbox"/> |
| Benutzername | Rollenname | <input type="checkbox"/> | <input type="checkbox"/> |
| Benutzername | Rollenname | <input type="checkbox"/> | <input type="checkbox"/> |

Abbildung 13: UI-Mockup Rollen- und Rechteverwaltung (US-36, US-37)

4.3.11 Eventanfrage

The UI-Mockup shows a clean, modern interface for submitting an event inquiry. At the top left, there are three small black dots. To the right, the text "EventHub" is displayed. Below this, the main title "Eventanfrage stellen" is centered. The form consists of several input fields: "Firmenname / Name", "Kontakt E-Mail", "Eventdatum", "Ort", "Budget", and a large text area for "Beschreibung / Anforderungen". At the bottom center is a button labeled "Anfrage senden".

Abbildung 14: UI-Mockup Eventanfrage (US-43)

4.3.12 Eventübersicht (Lesemodus)

The UI-Mockup displays the 'EventHub' application interface. On the left, a vertical sidebar contains navigation links: 'Übersicht', 'Notizen', 'Aufgaben', and 'Finanzen'. The main content area is titled 'Eventübersicht (Lesemodus)' and includes a note: 'Schreibrechte deaktiviert – nur Lesen möglich'. Below this, there are five input fields: 'Firmenname / Name', 'Kontakt E-Mail', 'Eventdatum', 'Ort', and 'Budget'. A large text area labeled 'Beschreibung / Anforderungen' follows. In the top right corner of the main area is a button '+ Rolle vergeben'. At the bottom right is a button 'Anfrage senden'.

Abbildung 15: UI-Mockup Eventübersicht (Lesemodus) (US-37)

4.4 Mobile-App-Mockups

4.4.1 Dashboard (Mobil)



Abbildung 16: UI-Mockup Dashboard Mobil (US-10, US-12, US-30, US-42)

4.4.2 Event-Details (Mobil)

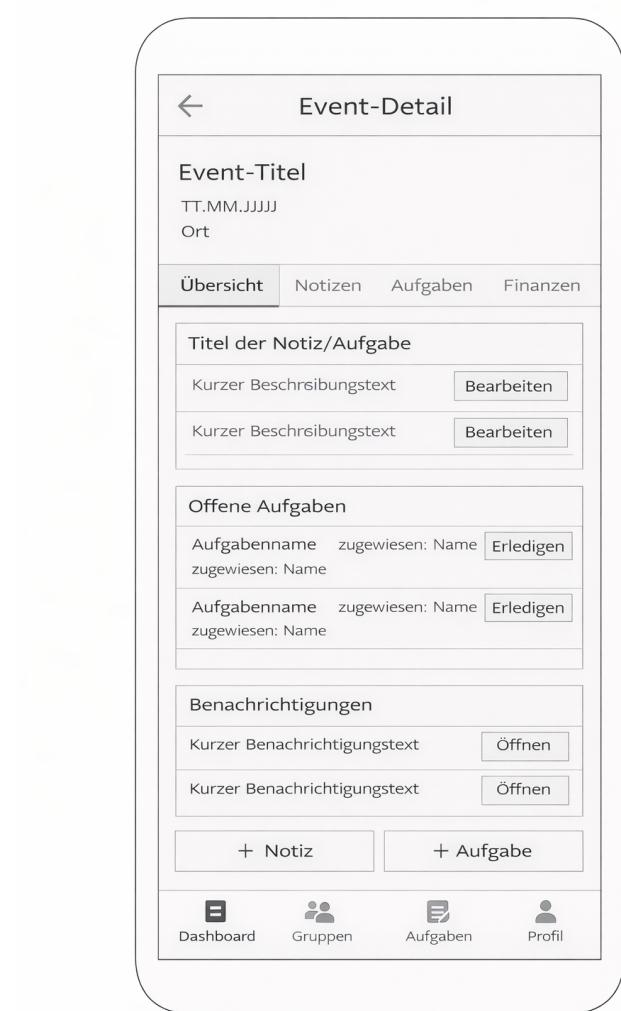


Abbildung 17: UI-Mockup Event-Details Mobil (US-14, US-17, US-18, US-19, US-21, US-22, US-23, US-24, US-25, US-26, US-30, US-31, US-32)

4.4.3 Aufgaben (Mobil)

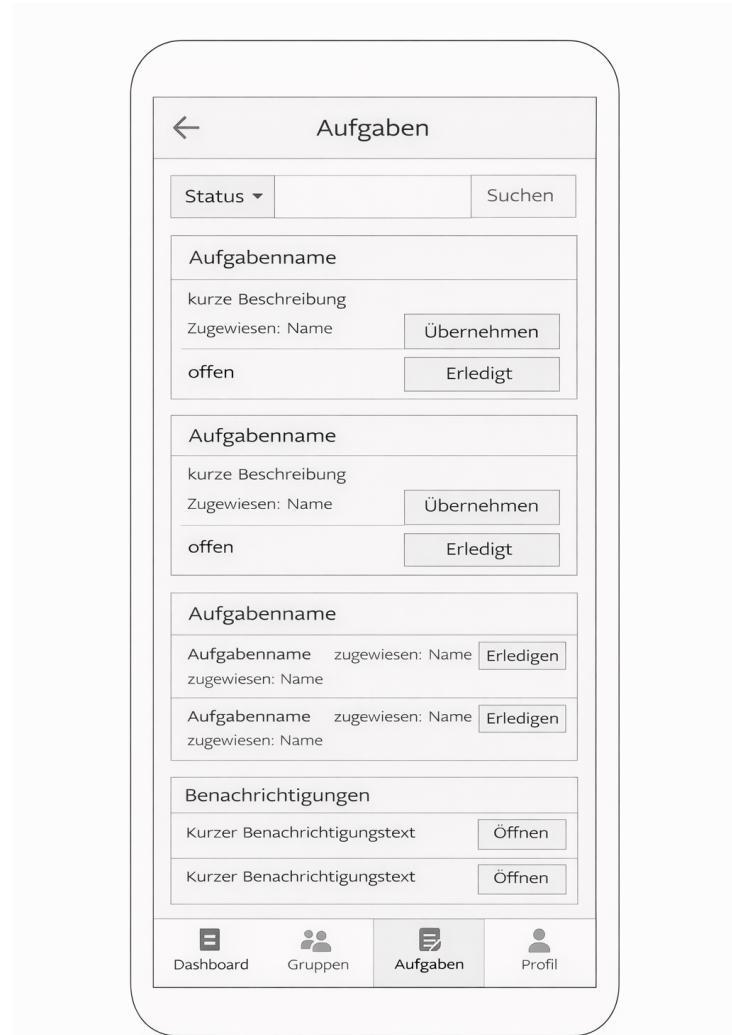


Abbildung 18: UI-Mockup Aufgaben Mobil (US-30, US-31, US-32, US-33)

4.4.4 Finanzen (Mobil)



Abbildung 19: UI-Mockup Finanzen Mobil (US-25, US-26, US-27, US-28, US-29)

5 Logische Sicht

5.1 Ableitung der Domänenklassen aus den User Stories

Die Domänenklassen des Systems EventHub wurden systematisch aus den fachlichen Anforderungen und User Stories abgeleitet. Ziel war es, zentrale Konzepte des Anwendungsbereichs als eigenständige Klassen zu identifizieren und ihre Beziehungen zueinander fachlich korrekt abzubilden.

Aus den User Stories zur Benutzer- und Gruppenverwaltung (z. B. Registrierung, Gruppenerstellung, Gruppenbeitritt) ergeben sich die Klassen *Benutzer*, *Gruppe*, *Mitgliedschaft* und *Rolle*. Die Mitgliedschaft modelliert dabei explizit die Beziehung zwischen Benutzern und Gruppen inklusive Status und Rolle.

User Stories zur Eventplanung und -durchführung führen zur Klasse *Event* sowie zu

abhängigen Konzepten wie *Aufgabe*, *AufgabenZuweisung*, *Medium* und *Kommentar*. Diese Klassen ermöglichen die strukturierte Organisation und Dokumentation von Events.

Funktionen zur Kommunikation und Abstimmung innerhalb von Gruppen werden durch Notizen und Umfragen abgebildet. Daraus ergeben sich die Klassen *Notiz* mit ihren Spezialisierungen (Freitext-, Termin- und Ausgaben-Notiz) sowie *Umfrage*, *UmfrageOption* und *Stimme*.

Die User Stories zur Finanzverwaltung (Einzahlungen, Rückzahlungen, Kassenübersicht) resultieren in den Klassen *Einzahlung*, *Rueckzahlung* und berichtenden Strukturen wie dem *Kassensturz*.

Erweiterte User Stories für Dienstleisterprozesse und Ressourcenplanung führen zu Domänenklassen wie *Dienstleister*, *Anfrage*, *Angebot*, *Rechnung*, *Mahnung* sowie *Material*, *Mitarbeiter* und den zugehörigen Buchungs- und Einsatzklassen.

Die so identifizierten Domänenklassen bilden die fachliche Grundlage für das Klassendiagramm und dienen als Ausgangspunkt für das logische Datenmodell.

5.2 Fachliches Domänenmodell (UML-Klassendiagramm)

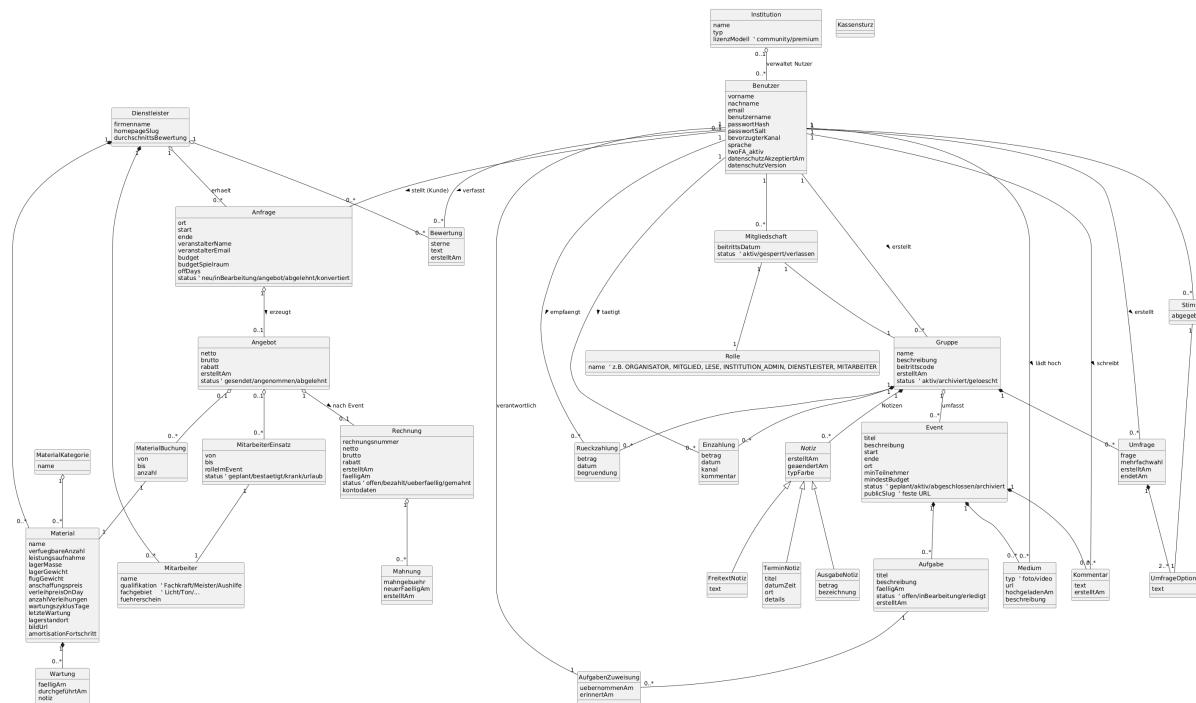


Abbildung 20: Klassendiagramm1.png

5.3 Beziehungen, Multiplizitäten und Vererbungen

5.4 Logisches Datenmodell

5.4.1 Benutzer, Gruppen und Events

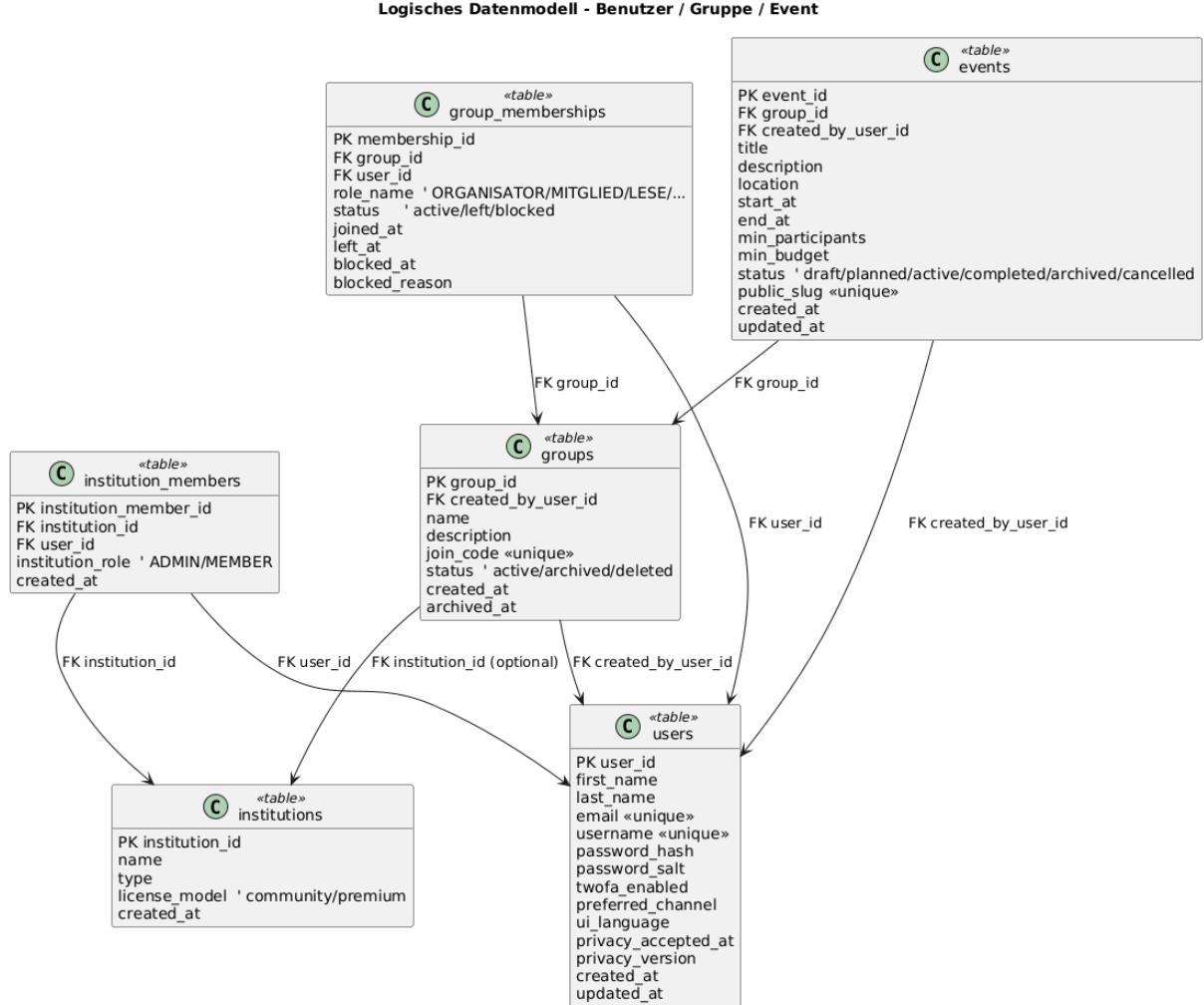


Abbildung 21: LDM Benutzer, Gruppen und Events

Das logische Datenmodell bildet die zentralen Stammdaten des Systems EventHub ab. Die Tabellen **users**, **groups** und **events** repräsentieren Benutzer, Gruppen sowie geplante oder durchgeführte Events und stellen damit die fachliche Grundlage der Anwendung dar.

Die Beziehung zwischen Benutzern und Gruppen wird über die Tabelle **group_memberships** modelliert. Dadurch können zusätzliche Informationen wie Rolle und Status einer Mitgliedschaft (z. B. aktiv, gesperrt oder verlassen) explizit gespeichert werden. Rollen werden separat verwaltet, um eine flexible Zuordnung von Rechten innerhalb einer Gruppe zu ermöglichen.

Institutionen werden über die Tabelle **institutions** abgebildet. Die Mitgliedschaft von Benutzern in einer Institution erfolgt über **institution_members**. Gruppen können optional einer Institution zugeordnet sein, wodurch sowohl institutionelle als auch private Gruppen unterstützt werden.

Events sind eindeutig einer Gruppe zugeordnet und enthalten neben zeitlichen Angaben auch fachliche Attribute wie Mindestteilnehmerzahl, Mindestbudget und Status. Eindeutige Felder wie E-Mail-Adresse, Benutzername, Beitrittscode und öffentliche Event-URLs werden durch `unique`-Constraints abgesichert. Alle Beziehungen sind über Fremdschlüssel eindeutig definiert.

5.4.2 Notizen & Umfragen

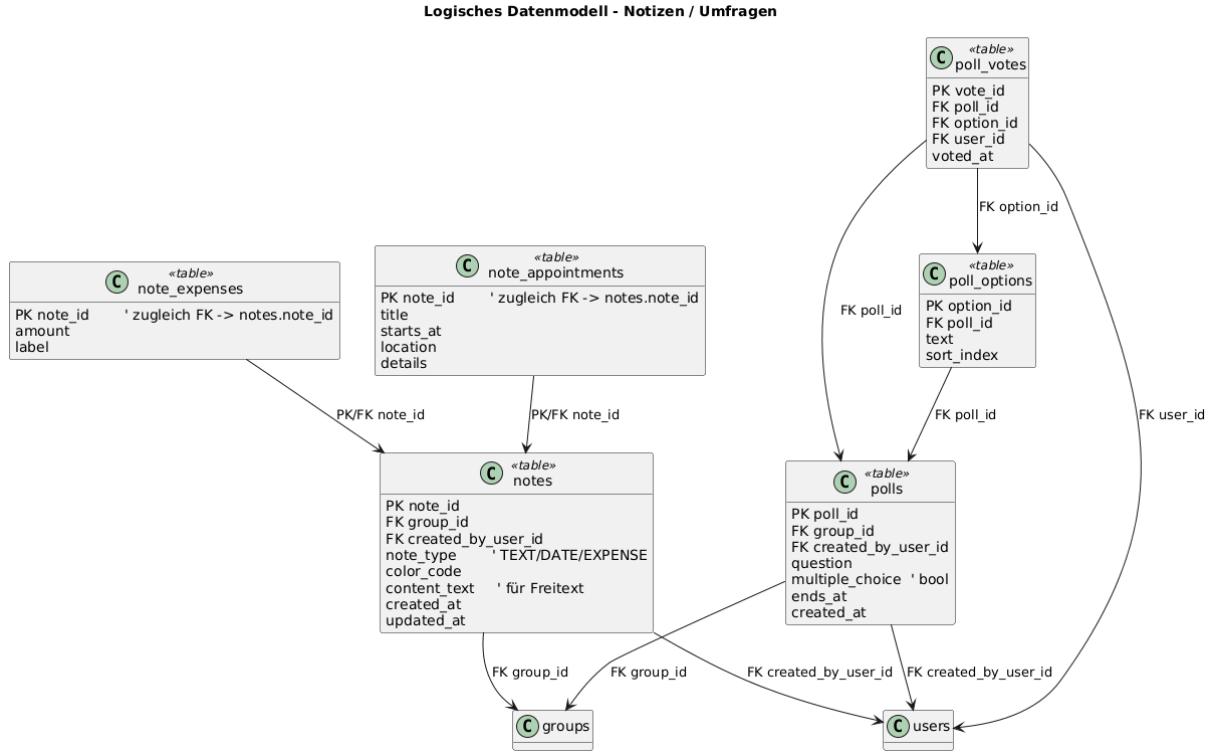


Abbildung 22: LDM Notizen & Umfragen

Abbildung 22 zeigt das logische Datenmodell für gruppenbezogene Notizen und Umfragen. Die Tabelle `notes` speichert alle Notizen als gemeinsame Basisstruktur und unterscheidet über `note_type` zwischen Freitext-, Termin- und Ausgaben-Notizen.

Termin- und Ausgaben-spezifische Attribute werden in den Tabellen `note_appointments` bzw. `note_expenses` abgelegt. Diese Tabellen verwenden `note_id` gleichzeitig als Primärschlüssel und Fremdschlüssel auf `notes`, wodurch eine 1:1-Beziehung zwischen Basisnotiz und Spezialisierung erzielt wird.

Umfragen werden über `polls` modelliert und sind eindeutig einer Gruppe sowie einem Ersteller zugeordnet. Die Antwortmöglichkeiten sind in `poll_options` abgebildet; abgegebene Stimmen werden als Bewegungsdaten in `poll_votes` gespeichert. Dadurch sind Auswertungen (z. B. Ergebnisdarstellung) möglich, ohne Umfragedaten nachträglich zu verändern.

5.4.3 Finanzen & Aufgaben

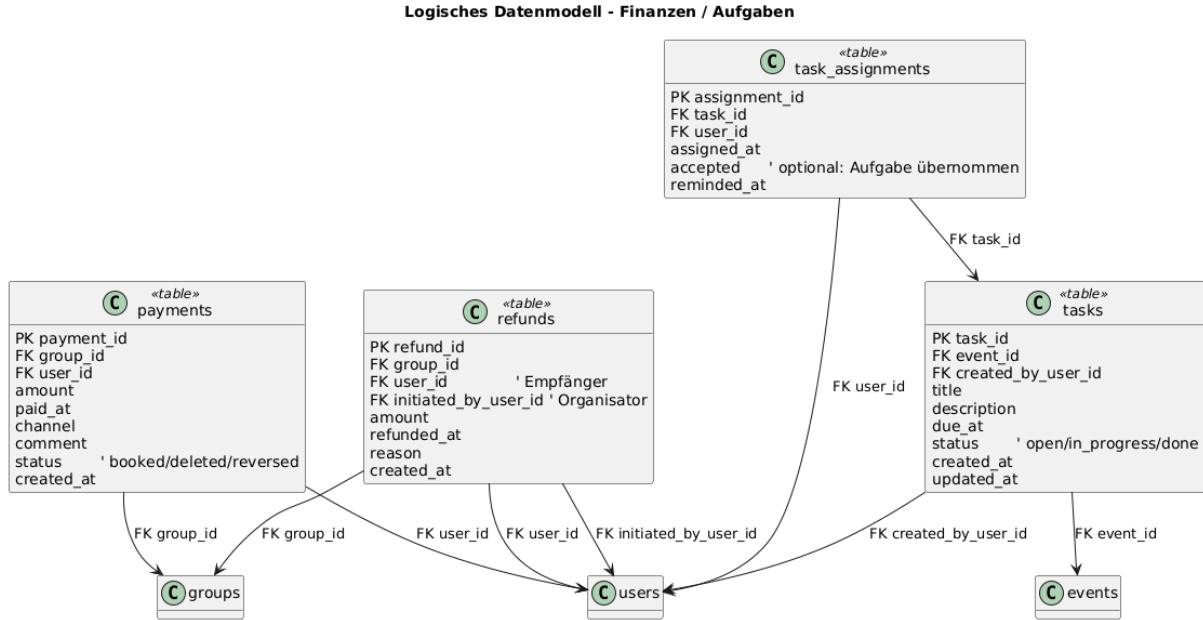


Abbildung 23: LDM Finanzen & Aufgaben

Das logische Datenmodell für Finanzen und Aufgaben bildet die zentralen Bewegungsdaten im Kontext von Gruppen und Events ab. Finanzielle Transaktionen werden über die Tabellen **payments** und **refunds** modelliert und ermöglichen die Nachverfolgung von Einzahlungen sowie Rückzahlungen innerhalb einer Gruppe.

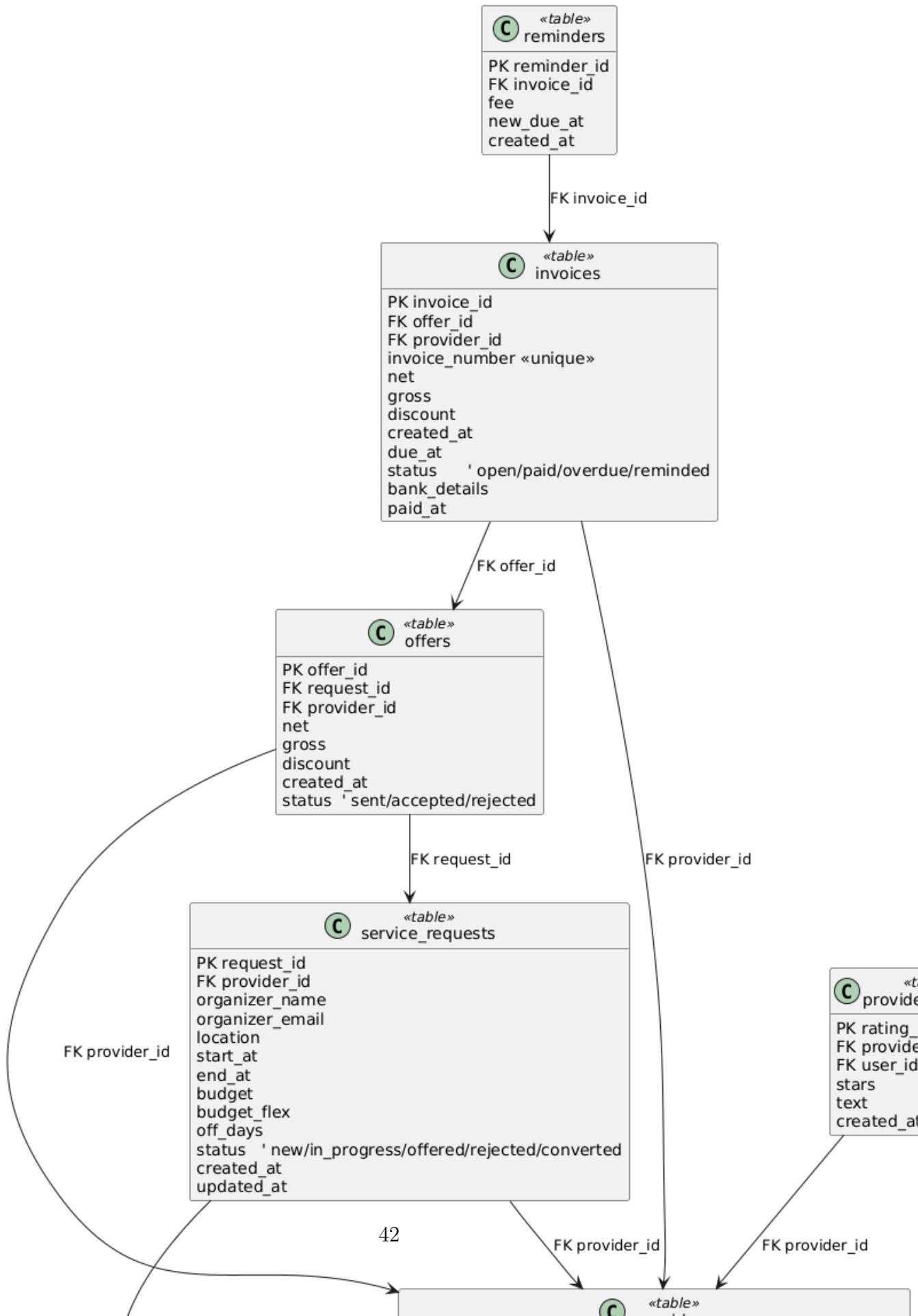
Einzahlungen (**payments**) sind jeweils einem Benutzer und einer Gruppe zugeordnet und enthalten neben dem Betrag auch Informationen zum Zahlungszeitpunkt, Zahlungskanal und Status. Rückzahlungen (**refunds**) referenzieren ebenfalls eine Gruppe und einen Benutzer als Empfänger; zusätzlich wird der auslösende Benutzer (z. B. ein Organisator) separat gespeichert, um Verantwortlichkeiten nachvollziehbar abzubilden.

Aufgaben werden über die Tabelle **tasks** modelliert und sind eindeutig einem Event sowie einem Ersteller zugeordnet. Die konkrete Zuweisung von Aufgaben an Benutzer erfolgt über **task_assignments**. Dadurch können Aufgaben mehreren Benutzern zugewiesen und Zustände wie Übernahme oder Erinnerung zeitlich dokumentiert werden.

Durch die Trennung von Stammdaten (Aufgaben) und Bewegungsdaten (Zuweisungen, Zahlungen, Rückzahlungen) bleibt das Datenmodell flexibel, erweiterbar und konsistent.

5.4.4 Dienstleister & Abrechnung

Logisches Datenmodell - Dienstleister / Abrechnung



Das logische Datenmodell für Dienstleister und Abrechnung bildet den vollständigen B2B-Prozess von der Anfrage über Angebot und Rechnung bis hin zu Mahnungen ab. Dienstleister werden in der Tabelle **providers** verwaltet und können optional mit einem Benutzerkonto verknüpft sein, sofern ein eigener Dienstleister-Login existiert.

Anfragen an Dienstleister werden über **service_requests** modelliert und enthalten organisatorische sowie zeitliche Rahmenbedingungen eines geplanten Events. Auf Basis einer Anfrage kann ein oder mehrere Angebote (**offers**) entstehen, die Preisangaben und einen Angebotsstatus enthalten.

Wird ein Angebot angenommen, kann daraus eine Rechnung (**invoices**) erzeugt werden. Rechnungen enthalten eindeutige Rechnungsnummern, Beträge, Zahlungsfristen sowie Statusinformationen. Mahnungen werden in **reminders** separat gespeichert und referenzieren jeweils eine Rechnung, wodurch mehrere Mahnstufen abgebildet werden können.

Bewertungen von Dienstleistern erfolgen über **provider_ratings** und sind einem Benutzer sowie einem Dienstleister zugeordnet. Durch die klare Trennung der einzelnen Prozessschritte bleibt das Datenmodell nachvollziehbar, erweiterbar und fachlich konsistent.

5.4.5 Ressourcenplanung

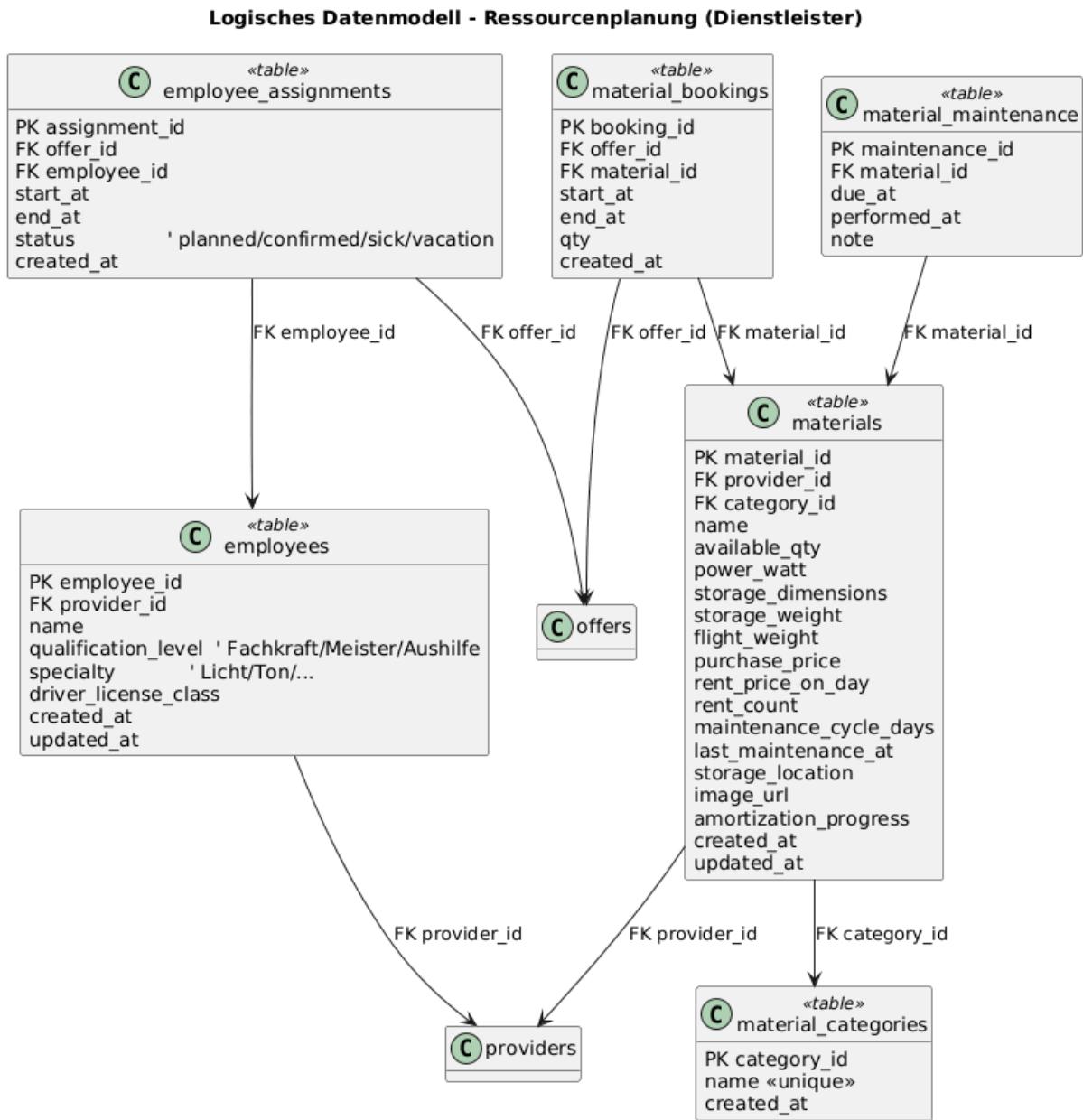


Abbildung 25: LDM Ressourcenplanung

Das logische Datenmodell zur Ressourcenplanung bildet Materialeinsatz im Kontext von Dienstleisterangeboten ab. Materialien werden in **materials** verwaltet und sind eindeutig einem Dienstleister (**provider_id**) zugeordnet. Über **material_categories** können Materialien thematisch klassifiziert werden, wobei Kategoriennamen durch einen **unique**-Constraint eindeutig sind.

Konkrete Materialreservierungen für ein Angebot werden über **material_bookings** modelliert. Jede Buchung referenziert ein **offer** sowie ein **material** und enthält Zeitraum und Menge, wodurch parallele Buchungen und wiederholte Ausleihen abbildbar sind. Wartungsereignisse werden in **material_maintenance** als eigene Bewegungsdaten gespeichert und referenzieren das jeweilige Material, um Fälligkeiten und durchgeführte Wartungen

nachvollziehbar zu dokumentieren.

Mitarbeiter werden in `employees` verwaltet und ebenfalls einem Dienstleister zugeordnet. Die Einsatzplanung erfolgt über `employee_assignments`, die einen Mitarbeiter mit einem Angebot verknüpft und Zeitraum sowie Einsatzstatus (z. B. geplant, bestätigt, krank, Urlaub) speichert. Insgesamt ermöglicht das Modell eine konsistente Planung und Nachverfolgung von Ressourcen pro Angebot.

5.5 Übersicht der Teilmodelle und Querverweise

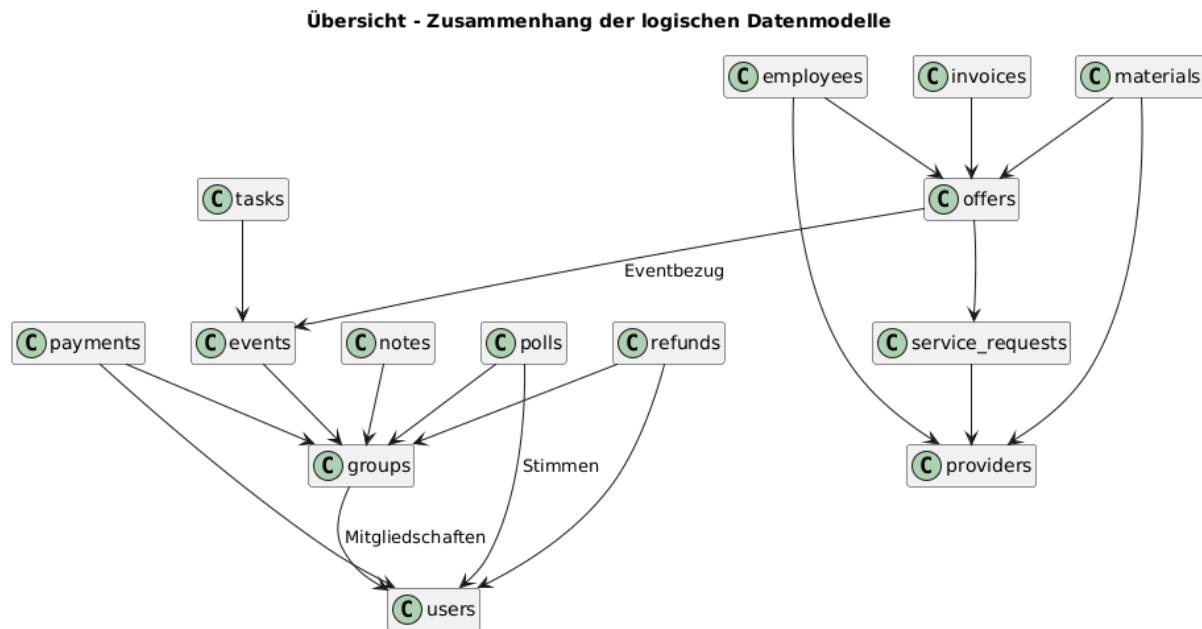


Abbildung 26: LDM Zusammenhang der LDM's

Abbildung 26 zeigt eine kompakte Übersicht über den Zusammenhang der fünf logischen Teilmodelle. Zentrale Entitäten sind `users`, `groups` und `events`, auf denen die Community-Funktionen wie Notizen, Umfragen, Finanzen und Aufgaben aufbauen.

Notizen und Umfragen sind gruppenbezogen modelliert, ebenso Einzahlungen und Rückzahlungen. Aufgaben sind an Events gekoppelt und werden damit indirekt über die zugehörige Gruppe kontextualisiert. Der Dienstleisterprozess verläuft von `service_requests` über `offers` zu `invoices` und ist über `providers` mit Dienstleistern verknüpft.

Die Ressourcenplanung (Material und Mitarbeiter) ist dem Dienstleister zugeordnet und wird über Angebote konkret für einzelne Aufträge/Events gebucht bzw. eingeplant. Die Übersicht dient als Navigationshilfe und stellt sicher, dass die Teilmodelle konsistent aufeinander referenzieren.

5.6 CRUD-Matrix

Tabelle 13: CRUD-Matrix

| Entität (Tabelle) | Create (C) | Read (R) | Update (U) | Delete (D) |
|---------------------|--|--------------|--|-----------------------------|
| users | US-01 | US-02, US-06 | US-04, US-05, US-07, US-08, US-09 | – |
| institutions | – | US-38 | – | – |
| institution_members | – | – | US-36, US-37 | – |
| groups | US-10 | US-14 | US-15 (Code verwalten implizit), US-16 (Min. Teilnehmer/Budget) | US-11 |
| group_memberships | US-10 (Organisator wird Mitglied), US-12 | US-14 | US-13 (Austritt als Statuswechsel), US-15 (sperren/entfernen), US-36, US-37 | – |
| events | US-10 (Event im Rahmen der Gruppe anlegen), US-39 (öffentliche Seite implizit) | US-39 | US-16 | US-11 (archivieren/löschen) |
| notes | US-17, US-21, US-27 | US-20 | US-18 | US-19 |
| note_appointments | US-21 | US-21, US-41 | US-18 (wenn Termin-Notiz bearbeitet wird) | US-19 (via Notiz löschen) |
| note_expenses | US-27 | US-26, US-28 | US-18 (wenn Ausgabe-Notiz bearbeitet wird) | US-19 (via Notiz löschen) |
| polls | US-22 | US-24 | – | – |
| poll_options | US-22 | US-24 | – | – |
| poll_votes | US-23 | US-24 | – | – |
| payments | US-25 | US-26, US-28 | – | – |
| refunds | US-29 | US-28 | – | – |
| tasks | US-30 | – | US-32 | – |
| task_assignments | US-31 | – | US-33 (Erinnerung/Statusdaten) | – |
| media | US-34 | US-34 | – | – |
| comments | US-35 | – | – | – |
| providers | – | – | – | – |
| service_requests | US-43 | – | – | – |
| offers | US-44 | – | – | – |
| invoices | US-45 | – | – | – |

| Entität (Tabelle) | Create (C) | Read (R) | Update (U) | Delete (D) |
|----------------------|---------------------------------|--------------|------------|--------------------------------------|
| reminders | US-46 | — | — | — |
| provider_ratings | — | — | — | — |
| material_categories | — | — | — | — |
| materials | US-47 | US-47, US-50 | US-47 | US-47 (falls “entfernen” vorgesehen) |
| material_maintenance | — | US-50 | — | — |
| material_bookings | US-48 | — | — | — |
| employees | US-49 | US-49 | US-49 | US-49 (falls “entfernen” vorgesehen) |
| employee_assignments | US-49 (Einsatzplanung implizit) | US-49 | US-49 | — |

Die CRUD-Matrix zeigt die Zuordnung der User Stories zu den persistierten Entitäten. Bewegungsdaten (z. B. Zahlungen, Abstimmungen, Buchungen) werden ausschließlich erzeugt (Create) und gelesen (Read), jedoch nicht verändert oder gelöscht.

6 Prozess-Sicht

6.1 Zustandsdiagramm: Event

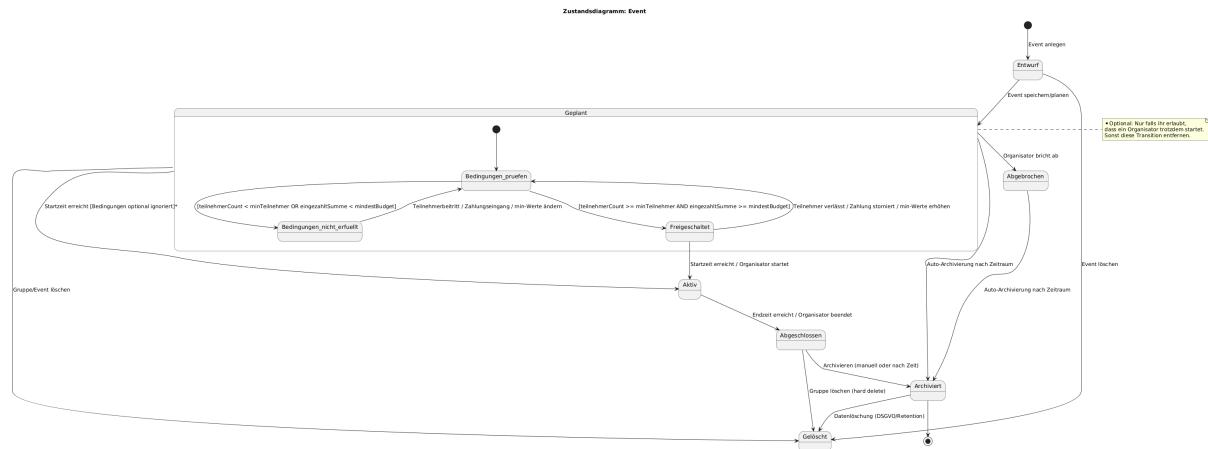


Abbildung 27: Zustandsdiagramm_Event.png

Das Zustandsdiagramm beschreibt den Lebenszyklus eines Events von der initialen Erstellung bis zur Archivierung oder Löschung. Nach dem Anlegen befindet sich ein Event zunächst im Zustand *Entwurf* und kann geplant, geändert oder verworfen werden.

Im Zustand *Geplant* werden fachliche Bedingungen wie Mindestteilnehmerzahl und Mindestbudget geprüft. Sind diese erfüllt, wird das Event freigeschaltet und kann zum geplanten Zeitpunkt oder manuell in den Zustand *Aktiv* übergehen. Nach Abschluss der Veranstaltung wechselt das Event in den Zustand *Abgeschlossen* und kann anschließend archiviert oder gemäß definierter Regeln gelöscht werden.

Das Diagramm stellt sicher, dass alle fachlich zulässigen Zustandsübergänge eindeutig definiert sind und der Eventstatus jederzeit konsistent ist.

6.2 Zustandsdiagramm: Rechnung

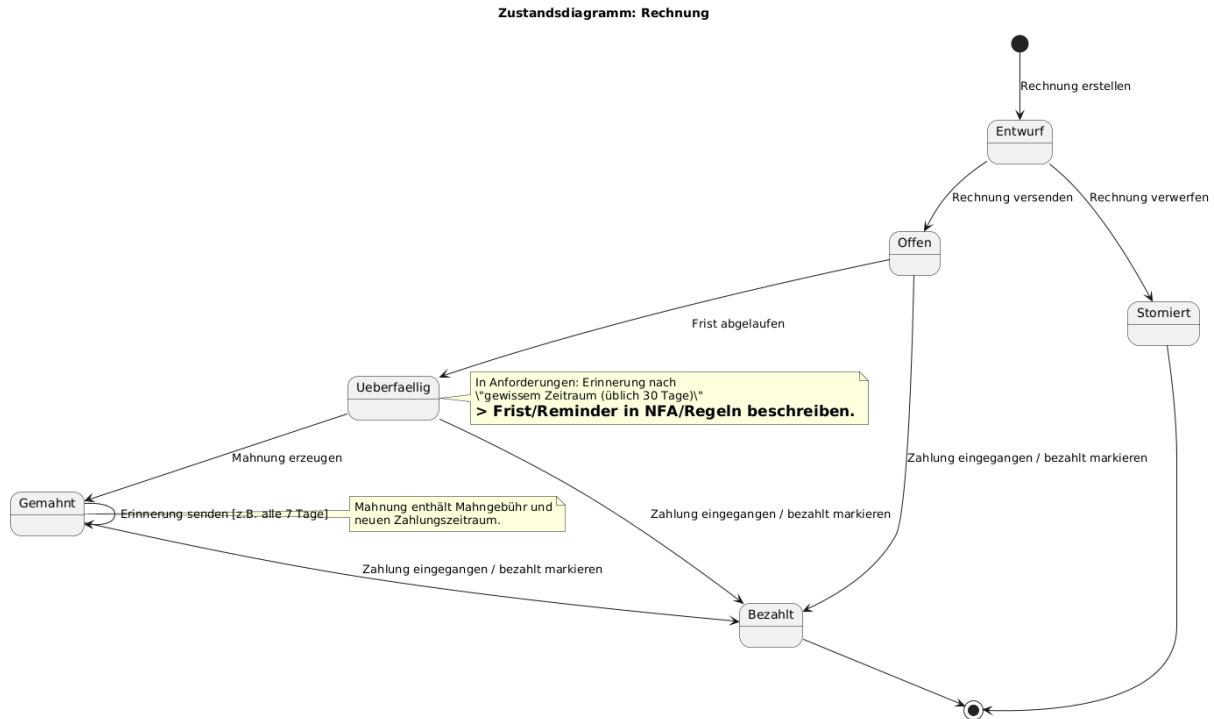


Abbildung 28: Zustandsdiagramm Rechnung

Das Zustandsdiagramm für Rechnungen modelliert den fachlichen Ablauf von der Erstellung einer Rechnung bis zu deren Abschluss. Eine Rechnung wird zunächst im Zustand *offen* angelegt und kann durch Zahlung in den Zustand *bezahlt* übergehen.

Erfolgt innerhalb der definierten Frist keine Zahlung, wechselt die Rechnung in den Zustand *überfällig*. In diesem Zustand können Mahnungen erzeugt werden, wodurch der Status entsprechend aktualisiert wird. Nach vollständigem Zahlungseingang gilt die Rechnung als abgeschlossen.

Durch die Modellierung der Zustände wird sichergestellt, dass Rechnungen nachvollziehbar bearbeitet werden und finanzielle Prozesse transparent und regelkonform ablaufen.

6.3 Sequenzdiagramm: Gruppe per Beitrittscode beitreten

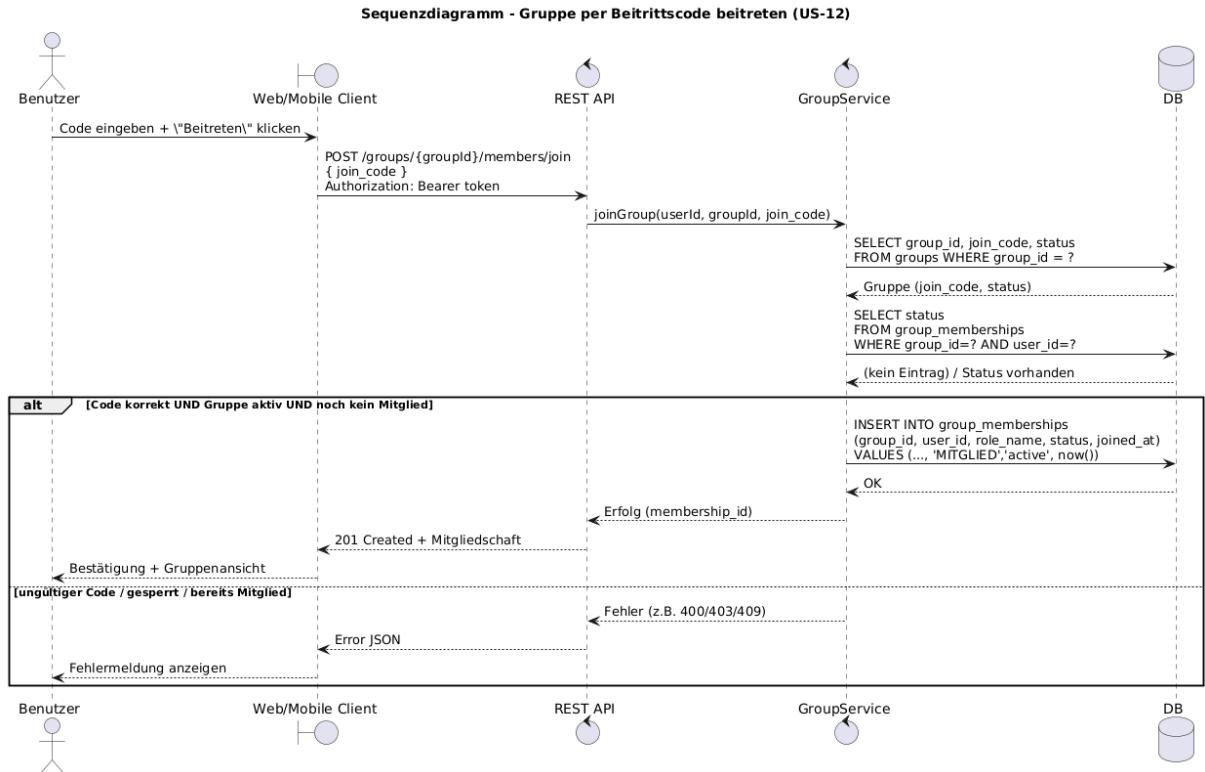


Abbildung 29: Sequenzdiagramm Gruppe per Beitrittscode beitreten

Abbildung 29 zeigt den Ablauf eines Gruppenbeitritts über einen Beitrittscode. Der Benutzer initiiert den Vorgang über den Web- oder Mobile-Client, welcher einen REST-Request an die Server-Schnittstelle sendet.

Die Server-Schicht prüft die Gültigkeit des Beitrittscodes, den Status der Gruppe sowie bestehende Mitgliedschaften. Ist der Beitritt zulässig, wird eine neue Mitgliedschaft mit der Rolle *Mitglied* angelegt undpersistiert. Andernfalls wird eine geeignete Fehlermeldung an den Client zurückgegeben.

Das Diagramm verdeutlicht die Trennung von Präsentation, Geschäftslogik und Datenhaltung sowie die serverseitige Durchsetzung fachlicher Regeln.

6.4 Sequenzdiagramm: Einzahlung erfassen

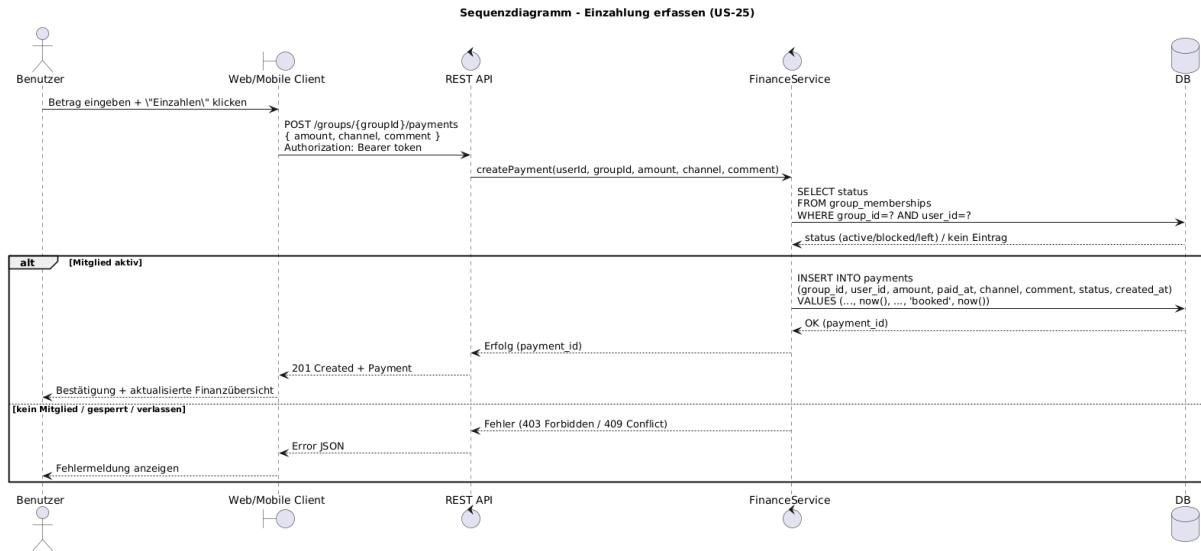


Abbildung 30: Sequenzdiagramm Einzahlung erfassen

Abbildung 30 zeigt den Ablauf zur Erfassung einer Einzahlung durch ein Gruppenmitglied. Der Benutzer gibt den Betrag im Client ein, woraufhin ein authentifizierter REST-Request an die Server-Schnittstelle gesendet wird.

Die Server-Schicht prüft zunächst, ob der Benutzer aktives Mitglied der angegebenen Gruppe ist. Ist dies der Fall, wird die Einzahlung als Bewegungsdatum in der Datenbankpersistiert und dem Client eine erfolgreiche Bestätigung zurückgegeben. Andernfalls wird der Vorgang mit einem Fehlerstatus abgelehnt.

Das Sequenzdiagramm verdeutlicht die serverseitige Durchsetzung fachlicher Regeln sowie die klare Trennung zwischen Client, Geschäftslogik und Datenhaltung.

7 Entwicklungs-Sicht

7.1 Drei-Schichten-Architektur (Client – Server – Datenbank)

EventHub wird als klassische Drei-Schichten-Architektur umgesetzt. Ziel ist eine klare Trennung von Verantwortlichkeiten zwischen Darstellung, Geschäftslogik und Datenhaltung. Dadurch werden Wartbarkeit, Skalierbarkeit und Sicherheit verbessert.

7.2 Client-Schicht (Präsentation)

Die Client-Schicht umfasst die Webanwendung (Browser) sowie die Mobile App. Sie ist verantwortlich für:

- Benutzerinteraktion (Formulare, Navigation, Validierung auf UI-Ebene)
- Darstellung von Gruppen, Events, Notizen, Umfragen, Finanzen und Aufgaben
- Aufruf der Backend-Schnittstellen über HTTPS (REST)
- Token-basierte Authentifizierung (z. B. Speicherung eines Access Tokens)

Die Logik (z. B. Rollenprüfung oder Finanzberechnungen) liegt nicht im Client, sondern in der Server-Schicht, um eine Umgehung fachlicher Regeln zu verhindern.

7.3 Server-Schicht (Applikation / Geschäftslogik)

Die Server-Schicht stellt die zentrale REST-API bereit und enthält die vollständige Geschäftslogik der Anwendung. Sie übernimmt unter anderem:

- Authentifizierung (Login, Zwei-Faktor-Authentifizierung) und Autorisierung (Rollen und Rechte wie Organisator, Mitglied, Institutionsadministrator, Dienstleister)
- Validierung fachlicher Regeln (z. B. Beitrittscode, Mindestteilnehmer, Mindestbudget, Sperrstatus)
- Orchestrierung komplexer Use Cases (z. B. Anfrage → Angebot → Rechnung → Mahnung)
- Aggregationen und Berichte (z. B. Kassenübersicht aus Einzahlungen, Ausgaben und Rückzahlungen)
- Integration externer Dienste (z. B. Kalendersynchronisation oder Benachrichtigungsanäle) über Adapter oder Services

Die Server-Schicht kapselt sämtliche Datenzugriffe (z. B. über Repository- oder DAO-Konzepte) und gibt nach außen ausschließlich kontrollierte Datenobjekte (DTOs/JSON) zurück.

7.4 Datenbank-Schicht (Persistenz)

Die Datenbank-Schicht ist für die persistente Speicherung aller relevanten Daten verantwortlich. Dazu zählen unter anderem:

- Stammdaten: Benutzer, Gruppen, Events, Dienstleister, Material, Mitarbeiter
- Bewegungsdaten: Einzahlungen, Abstimmungen, Buchungen, Einsätze, Mahnungen
- Status- und Historieninformationen (z. B. Mitgliedschaft aktiv, gesperrt oder verlassen)

Zugriffe auf die Datenbank erfolgen ausschließlich über die Server-Schicht. Ein direkter Datenbankzugriff vom Client ist nicht möglich. Dadurch können Rechteprüfungen, Datenschutzmaßnahmen und Konsistenzregeln zentral durchgesetzt werden.

7.5 Datenfluss

Der typische Datenfluss innerhalb der Drei-Schichten-Architektur gestaltet sich wie folgt:

1. Der Client sendet einen Request (HTTPS/REST) an den Server, inklusive Authentifizierungs-Token.
2. Der Server prüft Token und Berechtigungen und führt die entsprechende Geschäftslogik aus.
3. Der Server liest oder schreibt Daten in der Datenbank.
4. Der Server antwortet mit einem JSON-Ergebnis, das vom Client dargestellt wird.

7.6 Vorteile der Architektur

Die gewählte Drei-Schichten-Architektur bietet für EventHub folgende Vorteile:

- Sicherheit: Rechte- und DSGVO-Regeln werden zentral im Server durchgesetzt und können nicht clientseitig umgangen werden.
- Wartbarkeit: Benutzeroberfläche, Geschäftslogik und Datenhaltung können unabhängig voneinander weiterentwickelt werden.
- Skalierbarkeit: Die Server-Schicht kann horizontal skaliert werden, während die Datenbank separat optimiert werden kann.
- Portabilität: Web- und Mobile-Clients nutzen dieselbe REST-API.

7.7 Komponentendiagramm

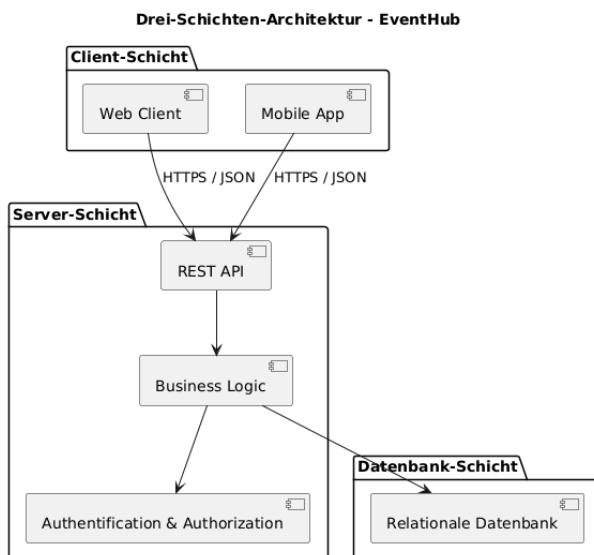


Abbildung 31: Komponentendiagramm

Abbildung 31 visualisiert die Drei-Schichten-Architektur von EventHub als Komponentendiagramm. Die Client-Schicht (Web Client und Mobile App) kommuniziert ausschließlich über HTTPS/JSON mit der REST-API der Server-Schicht.

In der Server-Schicht kapselt die *Business Logic* die fachlichen Regeln und Use-Case-Abläufe, während *Authentification & Authorization* die Authentifizierung und rollenbasierte Zugriffsprüfung zentral durchsetzt. Persistente Daten werden in der Datenbank-Schicht gespeichert; der Zugriff darauf erfolgt ausschließlich serverseitig, wodurch Konsistenz- und Sicherheitsanforderungen zentral kontrolliert werden können.

7.8 REST-API-Spezifikation

7.8.1 API – Konventionen

- Base URL: /api/v1
- Datenformat: JSON (UTF-8)
- Authentifizierung: Token-basiert (Authorization: Bearer Token)
- Autorisierung: Rollen- und Rechteprüfung serverseitig
- Statuscodes: 200 OK, 201 Created, 204 No Content, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 409 Conflict

7.8.2 Ressourcen und Endpunkte

Authentifizierung und Benutzer

| Endpoint | Methode | Beschreibung |
|---------------------------|---------|--|
| /auth/register | POST | Benutzer registrieren (US-01) |
| /auth/login | POST | Benutzer anmelden (US-02) |
| /auth/2fa/verify | POST | Zwei-Faktor-Authentifizierung bestätigen |
| /auth/logout | POST | Benutzer abmelden |
| /users/me | GET | Eigenes Benutzerprofil abrufen (US-06) |
| /users/me | PATCH | Profil bearbeiten (US-04, US-07, US-08, US-09) |
| /users/me/privacy-consent | POST | DSGVO-Zustimmung speichern |

Gruppen und Mitgliedschaften

| Endpoint | Methode | Beschreibung |
|--------------------------------|---------|---------------------------------------|
| /groups | POST | Gruppe anlegen (US-10) |
| /groups | GET | Eigene Gruppen anzeigen (US-14) |
| /groups/groupId | GET | Gruppendetails anzeigen |
| /groups/groupId | DELETE | Gruppe löschen/archivieren (US-11) |
| /groups/groupId/join-code | POST | Beitrittscode verwalten (US-15) |
| /groups/groupId/members/join | POST | Gruppe per Code beitreten (US-12) |
| /groups/groupId/members/userId | PATCH | Mitglied sperren/Rolle ändern (US-15) |
| /groups/groupId/leave | POST | Gruppe verlassen (US-13) |

Events

| Endpoint | Methode | Beschreibung |
|-------------------------|---------|--------------------------------|
| /groups/groupId/events | POST | Event anlegen |
| /groups/groupId/events | GET | Events einer Gruppe anzeigen |
| /events/eventId | GET | Eventdetails anzeigen |
| /events/eventId | PATCH | Event bearbeiten (US-16) |
| /events/eventId/archive | POST | Event archivieren |
| /events/eventId/public | GET | Öffentliche Eventseite (US-39) |

Notizen, Umfragen und Finanzen

| Endpoint | Methode | Beschreibung |
|-----------------------|---------|-------------------------------------|
| /groups/groupId/notes | POST | Notiz anlegen (US-17, US-21, US-27) |
| /groups/groupId/notes | GET | Notizen anzeigen (US-20) |

| Endpoint | Methode | Beschreibung |
|---------------------------------|---------|-----------------------------|
| /notes/noteId | PATCH | Notiz bearbeiten (US-18) |
| /notes/noteId | DELETE | Notiz löschen (US-19) |
| /groups/groupId/polls | POST | Umfrage anlegen (US-22) |
| /polls/pollId/vote | POST | Abstimmen (US-23) |
| /groups/groupId/payments | POST | Einzahlung erfassen (US-25) |
| /groups/groupId/finance/summary | GET | Kassenübersicht (US-28) |

Die REST-API stellt alle Systemfunktionen als versionierte Ressourcen bereit. Die Kommunikation erfolgt ausschließlich über HTTPS mit JSON als Austauschformat. Geschäftslogik sowie Sicherheits- und Datenschutzregeln werden serverseitig durchgesetzt.

8 Verteilungs-Sicht

8.1 UML-Verteilungsdiagramm

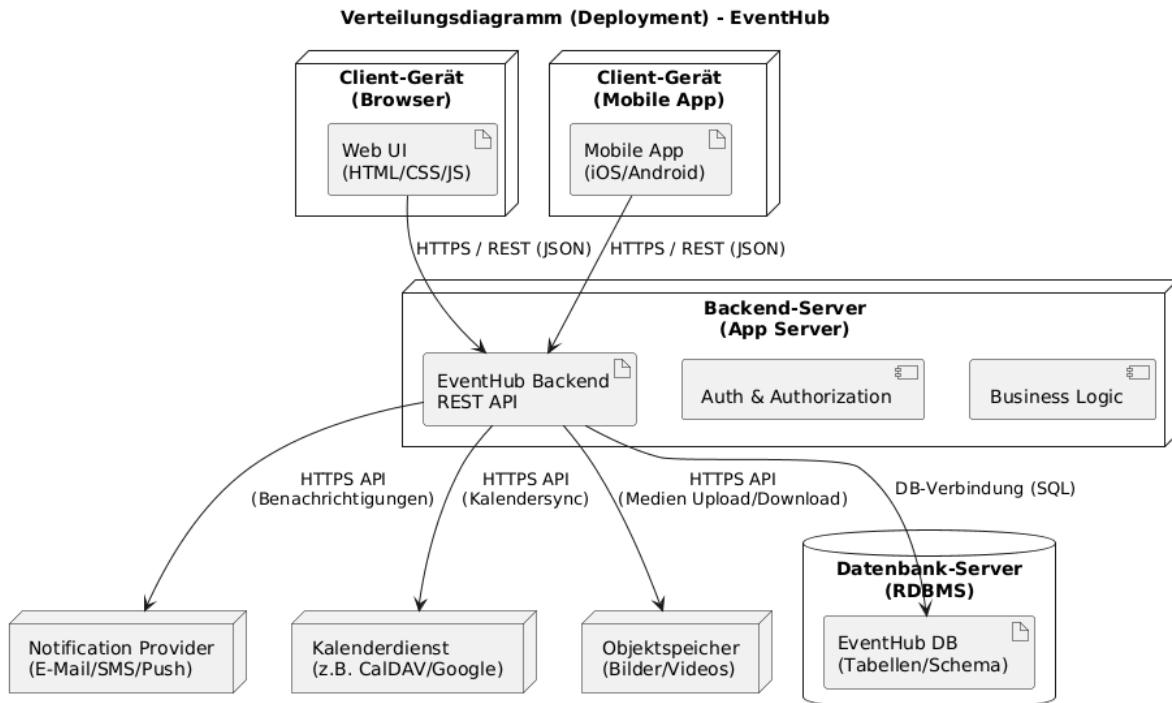


Abbildung 32: Verteilungsdiagramm

Abbildung 32 zeigt die physische Verteilung der Systemkomponenten von EventHub zur Laufzeit. Web-Client (Browser) und Mobile App laufen auf Client-Geräten und kommunizieren ausschließlich über HTTPS/REST mit dem Backend-Server.

Der Backend-Server hostet die REST-API sowie zentrale Komponenten für Geschäftslogik und Authentifizierung/Autorisierung. Persistente Daten werden auf einem separaten Datenbank-Server gespeichert, der ausschließlich vom Backend angesprochen wird.

Zusätzlich sind optionale externe Dienste angebunden, darunter ein Benachrichtigungsdienst (E-Mail/SMS/Push), ein Kalenderdienst sowie ein Objektspeicher für Medieninhalte. Diese werden über gesicherte HTTPS-Schnittstellen vom Backend aus integriert.

9 Sicherheit und Datenschutz

9.1 STRIDE-Sicherheitsanalyse

Die Sicherheitsanalyse des Systems erfolgt anhand des STRIDE-Modells. Betrachtet wird der Systemkontext bestehend aus Web-/Mobile-Client, REST-API (Backend) und Datenbank.

| STRIDE | Risiko | Beispiel | Gegenmaßnahmen |
|----------------------------|------------------------------|--|--|
| S (Spoofing) | Identitätsvortäuschung | Account-Übernahme durch gestohlene Zugangsdaten | Passwort-Hashing (Argon2/bcrypt), HTTPS, 2FA, Rate-Limiting, Token mit begrenzter Laufzeit |
| T (Tampering) | Manipulation von Daten | Manipulierte Requests mit fremden IDs | Serverseitige Validierung, Autorisierungsprüfungen, Prepared Statements, Ownership-Checks |
| R (Repudiation) | Abstreitbarkeit von Aktionen | Benutzer bestreitet Zahlung oder Sperre | Audit-Logging (wer/was/wann), serverseitige Zeitstempel, Request-IDs |
| I (Information Disclosure) | Informationsleck | Unbefugter Zugriff auf Gruppen- oder Profildaten | RBAC, Datenminimierung, sichere Fehlerausgaben, Verschlüsselung in Transit und at Rest |
| D (Denial of Service) | Dienstverweigerung | Brute-Force-Login oder API-Spam | Rate-Limits, Timeouts, Paging, WAF/Reverse-Proxy |
| E (Elevation of Privilege) | Rechteausweitung | Mitglied wird unberechtigt Organisator | Serverseitige Rollenprüfung, getrennte Admin-Endpunkte, gültige Rollen-Constraints |

9.2 Schutzmaßnahmen

9.2.1 Authentifizierung

Die Authentifizierung erfolgt über ein tokenbasiertes Verfahren (Bearer Token). Passwörter werden ausschließlich gehasht und gesalzen gespeichert. Optional kann eine Zwei-Faktor-Authentifizierung (2FA) aktiviert werden.

9.2.2 Autorisierung

Die Autorisierung erfolgt serverseitig über rollenbasierte Zugriffskontrollen (RBAC). Zusätzlich wird bei jedem Zugriff geprüft, ob der Benutzer zur angeforderten Ressource (z. B. Gruppe oder Event) gehört.

9.2.3 Eingabevalidierung und Datenintegrität

Alle Eingaben werden serverseitig validiert. Datenbankzugriffe erfolgen ausschließlich über vorbereitete Statements oder ORM-Mechanismen, um Manipulationen zu verhindern.

9.2.4 Transport- und Datensicherheit

Die gesamte Kommunikation erfolgt über TLS (HTTPS). Persistente Daten werden verschlüsselt gespeichert oder auf verschlüsselten Datenträgern abgelegt. Backups werden ebenfalls verschlüsselt gespeichert.

9.2.5 Logging und Monitoring

Sicherheitsrelevante Aktionen wie Login-Versuche, Rollenänderungen oder Zahlungen werden auditierbar protokolliert. Monitoring-Mechanismen erkennen ungewöhnliche Zugriffsmuster.

9.3 Datenschutz und DSGVO-Umsetzung

9.3.1 Zugriffsschutz (Privacy by Design)

Der Zugriff auf personenbezogene Daten ist ausschließlich berechtigten Benutzern möglich. Öffentlich zugängliche Inhalte zeigen nur explizit freigegebene Informationen.

9.3.2 Datenminimierung

Es werden nur Daten erhoben und gespeichert, die für den Betrieb der Anwendung erforderlich sind. Sensible Informationen wie Passwörter werden niemals im Klartext gespeichert.

9.3.3 Einwilligung und Transparenz

Die Zustimmung zur Datenschutzerklärung wird mit Zeitstempel und Versionsnummer (`privacy_accepted_at`, `privacy_version`) gespeichert und ist nachvollziehbar.

9.3.4 Löschkonzept und Aufbewahrung

Benutzer können ihr Konto löschen lassen. Personenbezogene Daten werden dabei gelöscht oder anonymisiert. Beim Austritt aus Gruppen werden zugehörige Einzahlungen gemäß fachlicher Vorgabe entfernt oder logisch gelöscht (Soft-Delete).

9.3.5 Betroffenenrechte

Das System unterstützt die Rechte auf Auskunft, Berichtigung und Löschung personenbezogener Daten. Ein Export der eigenen Daten kann in strukturierter Form bereitgestellt werden.

10 Arbeitsteilung

10.1 Aufgabenteilung Felix Franke

10.1.1 Fachliche Grundlagen

- Analyse der Aufgabenstellung und fachlichen Anforderungen
- Festlegung eines aussagekräftigen Produktnamens
- Identifikation und Beschreibung aller Akteure und Rollen
- Erstellung des Glossars der Fachdomäne

10.1.2 Funktionale Anforderungen

- Definition funktionaler Gruppen (z. B. Benutzerverwaltung, Gruppenmanagement, Eventplanung)
- Ausformulierung aller User Stories je funktionaler Gruppe unter Angabe von Rolle, Ziel, Nutzen, Akzeptanzkriterien und MoSCoW-Priorisierung
- Ergänzung fehlender funktionaler Anforderungen (Plausibilitätsprüfung)

10.1.3 Nicht-funktionale Anforderungen

- Beschreibung der Qualitätsmerkmale nach ISO 25010 (Usability, Sicherheit, Performance, Wartbarkeit etc.)
- Rechtliche Anforderungen (DSGVO)

10.1.4 GUI & Usability

- Entwurf der GUI-Mockups für Webbrowser
- Entwurf der GUI-Mockups für mobile Endgeräte
- Zustandsdiagramme für die Navigation
- Zuordnung der Screens zu den jeweiligen User Stories
- Beschreibung der angewandten UI-/UX-Designprinzipien

10.2 Aufgabenteilung Markus Wurms

10.2.1 Domänen- & Datenmodellierung

- Ableitung der Domänenklassen aus den User Stories
- Erstellung des fachlichen Domänenmodells (UML-Klassendiagramm)
- Modellierung von Beziehungen, Multiplizitäten und Vererbungen
- Ergänzung sinnvoller Attribute und Klassen

10.2.2 Zustände & Datenpersistenz

- Zustandsdiagramme für zentrale Klassen (z. B. Event, Rechnung, Aufgabe)
- Überführung des fachlichen Modells in ein logisches Datenmodell
- Definition von Bewegungsdaten (Zahlungen, Buchungen, Statusänderungen)
- Erstellung der CRUD-Matrix (User Stories ↔ Klassen)

10.2.3 Systemarchitektur & Schnittstellen

- Beschreibung der Drei-Schichten-Architektur (Client – Server – Datenbank)
- UML-Verteilungsdiagramm (Hardware-/Systemlandschaft)
- Definition der REST-API (Ressourcen, Endpunkte, HTTP-Methoden)
- Sequenzdiagramme für ausgewählte Kern-Use-Cases

10.2.4 Sicherheit & Datenschutz

- Sicherheitsanalyse mittels STRIDE
- Beschreibung der Schutzmaßnahmen (Authentifizierung, Autorisierung, Verschlüsselung)
- Technische Umsetzung von DSGVO-Anforderungen (Zugriffsschutz, Datenminimierung, Löschkonzepte)

11 Übersicht verwendeter Hilfsmittel

11.1 OpenAI ChatGPT (openai.com, chatgpt.com)

11.1.1 Nutzung von ChatGPT zur Formatierung

ChatGPT wurde zur Hilfe gezogen beim Formatieren dieses Dokuments, insbesondere bei den Tabellen der User-Stories. Außerdem haben wir mit ChatGPT im ganzen Dokument auf Rechtschreibfehler kontrolliert.

11.1.2 Generierung von Mockups

ChatGPT wurde zur Generierung von Mockups genutzt. Die jeweils verwendeten Prompts werden hier ausgelistet. Jedes generierte Mockup wurde per Bildbearbeitungsprogramm meist noch ein wenig verändert. Die Prompts selber wurden per Hand erstellt und in ein einfaches Eingabefeld in ChatGPT 5.2 eingefügt (Version Stand 05.02.2026).

11.1.3 Prompts

Login & Registrierung Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
zentrierter content-bereich ohne sidebar,  
  
oben logo und schriftzug "EventHub",  
  
darunter rechteckige login-box mit title "Login",  
  
in der box mehrere rechteckige eingabefelder:
```

E-Mail

Passwort

darunter button "Login",

darunter text-links:

"Registrieren"

"Passwort vergessen",

checkbox "Angemeldet bleiben",

kleiner hinweisbereich "2-Faktor-Authentifizierung Code",

unter der login-box eine zweite kleinere box oder link "Anmelden"
}

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Dashboard Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug und Logo links und Navigation mit  
    "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts }  
    sidebar navigation on the left { "Dashboard" (active), "Gruppen", "Aufgaben",  
    "Finanzen", "Profil" }  
    mittiger Content-Bereich rechts neben der Sidebar {  
        title: "Dashboard",
```

darunter mehrere übersichts-karten in einem grid-layout:

karte "Meine Gruppen" -> liste von 3 rechteckigen listen-objekten mit Gruppennamen

karte "Nächste Termine" -> liste von 3 rechteckigen listen-objekten mit Datum + Titel

karte "Offene Aufgaben" -> liste von 3 rechteckigen listen-objekten mit Aufgabennamen + Status

karte "Benachrichtigungen" -> liste von 2-3 rechteckigen meldungen

```
    oben rechts ein button "+ Gruppe erstellen"  
}  
}
```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Gruppenübersicht Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug und Logo links und Navigation mit  
    "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts }  
    sidebar navigation on the left { "Gruppen" (active), "Dashboard", "Aufgaben",  
    "Finanzen", "Profil" }  
    mittiger Content-Bereich rechts neben der Sidebar {  
        title: "Gruppenübersicht",
```

oben eine suchleiste "Gruppe suchen" und ein eingabefeld "Beitrittscode eingeben" mit button "Beitreten",
 darunter sektion "Meine Gruppen" → liste von 4 rechteckigen listen-objekten mit:
 Gruppenname
 kurze Beschreibung
 1x Button "Öffnen"
 1x Button "Verlassen",
 oben rechts ein button "+ Gruppe erstellen"
 }
 }

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Event-Detailseite Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label. Layout: desktop browser window { top header bar { "Eventhub" Schriftzug und logo links und navigation mit "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts} sidebar navigation on the left {"Übersicht" (active), "Notizen/Umfragen", "Aufgaben", "Finanzen", "Medien", "Mitglieder"}Mittiger Content-Bereich recht neben der sidebar {title: "Event-Detail", darunter eine Zeile mit "Event-Titel" und "Ort,Datum", darunter action buttons: "+Notiz erstellen", "+Aufgabe hinzufügen", "Mitglieder verwalten", darunter Sektion "Übersicht" -> untergeordnet liste von 3 rechteckigen listen-objekten mit "Notizen/Aufgaben" als Titel und "Beschreibung der Notiz/Aufgabe" als untertext"}}

Finanzen Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```

desktop browser window {
  top header bar { "EventHub" Schriftzug und Logo links und Navigation mit
    "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts }
  sidebar navigation on the left { "Übersicht", "Notizen/Umfragen", "Aufgaben",
    "Finanzen" (active), "Medien", "Mitglieder" }
  mittiger Content-Bereich rechts neben der Sidebar {
    title: "Finanzen",
  }
}
    
```

oben budget-übersicht als rechteckige info-boxen:

"Budget-Ziel"

"Aktueller Stand"

"Restbetrag"

darunter action buttons: "+ Einzahlung hinzufügen", "+ Ausgabe erfassen", "Rückzahlung auslösen",

darunter tabelle mit mehreren rechteckigen listen-objekten:

Datum

Typ (Einzahlung/Ausgabe)

Beschreibung

Betrag

Buttons "Bearbeiten", "Löschen"

optional einfache fortschrittsleiste als rechteckiger balken für
Budgetfortschritt
}
}

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches
UI-design, kein schatten, keine farben, nur wireframe.

Notizen & Umfragen Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub".
Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug und Logo links und Navigation mit  
    "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts }  
    sidebar navigation on the left { "Übersicht", "Notizen/Umfragen" (active),  
    "Aufgaben", "Finanzen", "Medien", "Mitglieder" }  
    mittiger Content-Bereich rechts neben der Sidebar {  
        title: "Notizen & Umfragen",
```

oben action buttons: "+ Notiz erstellen", "+ Umfrage erstellen",

darunter filterleiste mit dropdown "Typ filtern" und suchfeld,

darunter liste von 4 rechteckigen listen-objekten mit:

Titel der Notiz/Umfrage

kurzer Beschreibungstext

Status oder Anzahl Stimmen

Buttons "Bearbeiten", "Löschen",

bei einer geöffneten Umfrage zusätzlich rechteckige Antwortoptionen mit Button "Ja" und Button "Nein"

}

}

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Aufgabenplanung Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug und Logo links und Navigation mit  
    "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts }  
    sidebar navigation on the left { "Übersicht", "Notizen/Umfragen", "Aufgaben"  
    (active), "Finanzen", "Medien", "Mitglieder" }  
    mittiger Content-Bereich rechts neben der Sidebar {  
        title: "Aufgabenmanagement",
```

oben action button: "+ Aufgabe erstellen",

darunter filterleiste mit dropdown "Status filtern" und suchfeld,

darunter aufgabenliste als mehrere rechteckige listen-objekte mit:

Aufgabenname

kurze Beschreibung

zugewiesene Person

Status-Anzeige (offen / in Bearbeitung / erledigt)

Buttons "Übernehmen", "Bearbeiten", "Erledigt",

```
optional gruppiert in sektionen "Offen", "In Bearbeitung", "Erledigt"  
}  
}
```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Profil & Einstellungen Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug und Logo links und Navigation mit  
    "Dashboard", "Gruppen", "Aufgaben", "Profil" rechts }  
    sidebar navigation on the left { "Dashboard", "Gruppen", "Aufgaben", "Finanzen",  
    "Profil" (active) }  
    mittiger Content-Bereich rechts neben der Sidebar {  
        title: "Profil & Einstellungen",  
  
        sektion "Profilinformationen" mit mehreren rechteckigen eingabefeldern:  
  
        Name  
  
        E-Mail  
  
        Passwort ändern  
  
        Speichern Button,  
  
        sektion "Benachrichtigungen" mit checkboxen oder toggles:  
  
        E-Mail  
  
        Push  
  
        SMS,  
  
        sektion "Sprache" mit dropdown auswahl,  
  
        sektion "Integrationen" mit zwei rechteckigen boxen:  
  
        "Kalender synchronisieren" mit Button "Verbinden"  
  
        "Sprachassistent koppeln" mit Button "Verbinden",  
  
        unten Button "Logout"  
    }  
}
```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Dienstleister-Dashboard Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug und Logo links und Navigation mit  
    "Dashboard", "Anfragen", "Angebote", "Rechnungen", "Ressourcen", "Profil" rechts  
}
```

```

sidebar navigation on the left { "Dashboard" (active), "Anfragen", "Angebote",
"Rechnungen", "Material", "Mitarbeiter", "Profil" }
mittiger Content-Bereich rechts neben der Sidebar {
title: "Dienstleister Dashboard",

darunter mehrere übersichts-karten in einem grid-layout:

karte "Neue Anfragen" → liste von 3 rechteckigen einträgen mit Eventname + Datum
+ Button "Öffnen"

karte "Offene Angebote" → liste von 2 rechteckigen einträgen mit Preis + Status

karte "Offene Rechnungen" → liste von 2 rechteckigen einträgen mit Betrag +
Fälligkeitsdatum

karte "Ressourcenstatus" → einfache liste von Material oder Mitarbeitern mit
Verfügbarkeit

}

}

```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Dashboard – Mobil Prompt:

Erstelle ein wireframe-ui mockup einer mobilen app-version der website mit dem Namen "EventHub". Hochformat, smartphone-format, Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```

smartphone screen im hochformat {

oben eine schmale header bar mit schriftzug "EventHub",

mittiger content-bereich mit vertikal gestapelten karten:

karte "Meine Gruppen" → liste von 2-3 rechteckigen einträgen

karte "Nächste Termine" → liste von 2 rechteckigen einträgen

karte "Offene Aufgaben" → liste von 2 rechteckigen einträgen

karte "Benachrichtigungen" → liste von 2 rechteckigen meldungen

unten feste bottom-navigation bar mit 4 icons/text-labels:
"Dashboard" (active), "Gruppen", "Aufgaben", "Profil"

alle elemente groß und touch-geeignet, einfache rechtecke, viel abstand zwischen
elementen
}
```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Event-Detail – Mobil Prompt:

Erstelle ein wireframe-ui mockup einer mobilen app-version der website mit dem Namen "EventHub". Hochformat, smartphone-format, Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
smartphone screen im hochformat {
```

oben header bar mit titel "Event-Detail" und zurück-button,

darunter event-informationen als block:

Event-Titel

Datum

Ort

darunter horizontale tab-leiste oder segmentierte navigation mit:
"Übersicht" (active), "Notizen", "Aufgaben", "Finanzen",

darunter scrollbarer content-bereich mit mehreren rechteckigen listen-objekten
(Notizen/Aufgaben),

unten zwei große touch-buttons:

"+ Notiz"

"+ Aufgabe"

alle elemente groß, vertikal gestapelt, fingerfreundliche abstände, einfache rechtecke, kein desktop-layout
}

Stil: flach, minimal, nur einfache rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Aufgaben – Mobil Prompt:

Erstelle ein wireframe-ui mockup einer mobilen app-version der website mit dem Namen "EventHub". Hochformat, smartphone-format, Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
smartphone screen im hochformat {
```

oben header bar mit titel "Aufgaben",

darunter filterleiste mit dropdown "Status" und suchfeld,

darunter vertikal gestapelte aufgabenliste mit mehreren großen rechteckigen einträgen:

Aufgabename

kurze Beschreibung

zugewiesene Person

Statusanzeige

rechts oder unten pro eintrag große touch-buttons:

"Übernehmen"

"Erledigt",

unten feste bottom-navigation bar mit:

"Dashboard", "Gruppen", "Aufgaben" (active), "Profil"

alle elemente groß, fingerfreundlich, viel abstand, nur einfache rechtecke
}

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Finanzen – Mobil Prompt:

Erstelle ein wireframe-ui mockup einer mobilen app-version der website mit dem Namen "EventHub". Hochformat, smartphone-format, Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

smartphone screen im hochformat {

oben header bar mit titel "Finanzen",

darunter budget-übersicht als große info-boxen:

"Budget-Ziel"

"Aktueller Stand"

"Restbetrag"

darunter zwei große touch-buttons nebeneinander oder gestapelt:

"+ Einzahlung"

"+ Ausgabe",

darunter vertikal gestapelte liste von einträgen mit:

Datum

Beschreibung

Betrag (+ oder -)
kleiner button "Details",
unten feste bottom-navigation bar mit:
"Dashboard", "Gruppen", "Aufgaben", "Profil"
alle elemente groß, vertikal, fingerfreundlich, nur einfache rechtecke und text
}

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Rollen- und Rechteverteilung Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
    top header bar { "EventHub" Schriftzug links, Navigation "Dashboard", "Events",  
    "Benutzer", "Admin" }  
    sidebar navigation on the left { "Übersicht", "Events", "Benutzer", "Rollen &  
    Rechte" (active) }
```

mittiger Content-Bereich rechts neben der Sidebar {
title: "Rollen- und Rechteverwaltung",

oben dropdown "Institution auswählen",

darunter tabelle mit mehreren rechteckigen zeilen:

Spalten:

Benutzer | Rolle | Lesen | Schreiben

Spalte Benutzer enthält Benutzernamen,
Spalte Rolle enthält Platzhalter Rollenname,
Spalten lesen und schreiben sind checkboxen,

oben rechts button "+ Rolle vergeben",
unten rechts button "Speichern"
}
}

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Eventanfrage Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
zentrierter content-bereich ohne sidebar {  
  
title: "Eventanfrage stellen",  
  
mehrere rechteckige formularfelder:  
  
Firmenname / Name  
  
Kontakt E-Mail  
  
Eventdatum  
  
Ort  
  
Budget  
  
Beschreibung / Anforderungen (großes Textfeld)  
  
darunter button "Anfrage senden"  
}  
}
```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.

Eventübersicht (Lesemodus) Prompt:

Erstelle ein wireframe-ui mockup einer desktop-website mit dem Namen "EventHub". Grauskala, keine Designelemente, nur einfache Rechtecke und Text-Label.

Layout:

```
desktop browser window {  
top header bar { "EventHub" }  
  
sidebar navigation on the left { "Übersicht" (active), "Notizen", "Aufgaben",  
"Finanzen" }  
  
mittiger Content-Bereich {  
title: "Eventübersicht (Lesemodus)",  
  
liste von einträgen ohne buttons oder bearbeitungsoptionen,  
  
kleiner hinweistext oben: "Schreibrechte deaktiviert - nur Lesen möglich"  
}  
}
```

Stil: flach, minimal, nur rechtecke und platzhaltertext, kein realistisches UI-design, kein schatten, keine farben, nur wireframe.