

## Pflichtenheft

|                    |                         |
|--------------------|-------------------------|
| Projektbezeichnung | Event-Management-System |
| Projektleitung     | Julian Lachenmaier      |
| Erstellt am        | 12.04.2025              |
| Letzte Änderung am | 25.05.2025              |

### Änderungsverlauf

| Nr. | Datum    | Geänderte Kapitel | Art der Änderung | Verantwortlich     |
|-----|----------|-------------------|------------------|--------------------|
| 1   | 18.04.25 | 1 – 2.4           | Verbesserung     | Julian Lachenmaier |
| 2   | 20.04.25 | 2.4 - 3           | Verbesserung     | Julian Lachenmaier |
| 3   | 24.04.25 | 1.4 + 2           | Rollen anpassen  | Julian Lachenmaier |
| 4   | 25.05.25 | 2.16              | Verbesserung     | Rebecca Niklaus    |

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>1. Einleitung .....</b>                            | <b>3</b>  |
| 1.1. Zielsetzung .....                                | 3         |
| 1.2. Endabnahme .....                                 | 3         |
| 1.3. Abnahmekriterien .....                           | 3         |
| 1.4. Benutzergruppen und Stakeholder .....            | 3         |
| 1.5. Produktübersicht .....                           | 4         |
| 1.6. Produktdaten .....                               | 4         |
| 1.7. Externe Systeme .....                            | 5         |
| 1.7.1. Externe Systeme .....                          | 5         |
| 1.7.2. Interne Systeme / Systemkomponenten .....      | 5         |
| 1.7.3. Interne Schnittstellen .....                   | 5         |
| 1.7.4. Datenformat und Kommunikation .....            | 6         |
| <b>2. Funktionale Anforderungen .....</b>             | <b>6</b>  |
| 2.1. Organisation hinzufügen .....                    | 6         |
| 2.2. Event konfigurieren .....                        | 6         |
| 2.3. Event bearbeiten .....                           | 7         |
| 2.4. Event absagen .....                              | 8         |
| 2.5. Templates .....                                  | 8         |
| 2.6. Auf Plattform registrieren .....                 | 9         |
| 2.7. Profil bearbeiten .....                          | 9         |
| 2.8. Benutzer löschen .....                           | 9         |
| 2.9. Organisator einladen .....                       | 9         |
| 2.10. Organisator entfernen .....                     | 10        |
| 2.11. Events anzeigen .....                           | 10        |
| 2.12. Event-Overview .....                            | 10        |
| 2.13. Detaillierte Informationen Event .....          | 11        |
| 2.14. Umgang mit bereits stattgefundenen Events ..... | 11        |
| 2.15. Zu einem Event anmelden/abmelden .....          | 11        |
| 2.16. Event-Prozesssteuerung konfigurieren .....      | 12        |
| <b>3. Nichtfunktionale Anforderungen .....</b>        | <b>12</b> |
| 3.1. Allgemeine Anforderungen .....                   | 12        |
| 3.1.1. Benutzerfreundlichkeit .....                   | 12        |
| 3.1.2. Sicherheit .....                               | 13        |
| 3.2. Technische Anforderungen .....                   | 13        |
| 3.3. Weitere nichtfunktionale Anforderungen .....     | 13        |
| 3.4. Qualitätsziele .....                             | 13        |

## 1. Einleitung

Das vorliegende Pflichtenheft beschreibt die an das zu entwickelnde Produkt gestellten funktionalen und nicht-funktionalen Anforderungen. Es bildet die Grundlage für die Spezifikation der Rahmenbedingungen, die die Entwicklung definieren. Diese Anforderungen wurden von uns (NextGen Development) aus dem Lastenheft weiter ausgearbeitet und präzisiert.

### 1.1. Zielsetzung

Das Ziel ist die Entwicklung einer webbasierten Event-Management-Plattform, mit der Organisationen Events effizient planen, verwalten und evaluieren können. Neben hoher Benutzerfreundlichkeit stehen Sicherheit, Prozessautomatisierung und Skalierbarkeit im Vordergrund.

### 1.2. Endabnahme

Die Endabnahme findet am 27.05.2025 statt.

### 1.3. Abnahmekriterien

- Vollständige Umsetzung aller funktionalen Anforderungen
- Benutzerfreundliche Oberfläche für alle Rollen
- Sicherstellung der Kommunikation und Datenspeicherung gemäß Datenschutzstandards
- Erfolgreiche Durchführung von Testszenarien
- Die Dokumentation ist zugänglich und vollständig

### 1.4. Benutzergruppen und Stakeholder

#### Benutzergruppen (Systemrollen)

Dies sind die Rollen innerhalb der Anwendung, die bestimmten Nutzergruppen zugeordnet sind:

- **Administrator (Admin):** Nutzer mit erweiterten Systemrechten zur Benutzer- und Organisationsverwaltung sowie zur Konfiguration systemweiter Einstellungen.
- **Owner: Nutzer, der eine Organisation besitzt und verwaltet (z.B. Organisatoren hinzufügen oder entfernen, Organisation löschen).**
- **Organizer:** Nutzer, der innerhalb einer Organisation volle Berechtigungen zur Eventverwaltung besitzt. Organizer können beliebig viele Events erstellen, bearbeiten und löschen. Sie haben zudem das Recht, Benutzer zu Event Organisatoren zu ernennen, die anschließend spezifische Rechte für einzelne Events erhalten. Organizer sind für die Verwaltung der Eventlandschaft ihrer Organisation verantwortlich.
- **Event Organizer:** Nutzer, der von einem Organizer für ein bestimmtes Event ernannt wird. Event Organizer besitzen ausschließlich Rechte für das zugewiesene Event: sie können dieses Event bearbeiten, Teilnehmer verwalten und Anmeldungen annehmen oder ablehnen. Ihre Berechtigungen sind auf das jeweilige Event beschränkt und gelten nicht für andere Events oder Organisationseinstellungen.
- **Benutzer (User):** Registrierte Nutzer, die an Events teilnehmen können.

#### Stakeholder (Projektkontext)

Dies sind alle Personen oder Gruppen, die ein Interesse am Projekt haben oder davon betroffen sind:

- **Kunde / Auftraggeber:** Verantwortlich für die Beauftragung und Abnahme der Software. Legt die fachlichen Anforderungen fest.
- **NextGen Development:** Entwicklungsteam, das das System plant, umsetzt und wartet.
- **Endnutzer:** Personen, die die Anwendung im späteren Betrieb tatsächlich nutzen (Admins, Organisatoren, Benutzer).

### 1.5. Produktübersicht

Das zu entwickelnde Produkt ist eine webbasierte Plattform zur Planung, Durchführung und Verwaltung von Veranstaltungen durch Unternehmen und Organisationen. Es ermöglicht eine zentrale und benutzerfreundliche Organisation von Events, wobei sowohl organisatorische als auch teilnehmerbezogene Prozesse digital abgebildet und gesteuert werden.

Das System unterstützt verschiedene Rollen (z. B. Benutzer, Organisator, Administrator) mit spezifischen Rechten und Funktionen. Organisatoren können Events konfigurieren, verwalten, Teilnehmer einladen und prozessgesteuerte Abläufe festlegen. Benutzer wiederum können sich zu Events anmelden, relevante Informationen einsehen und erhalten automatisierte Benachrichtigungen (z. B. Erinnerungen, Bestätigungen).

Die Plattform stellt darüber hinaus eine flexible Template-Verwaltung zur Verfügung, mit der Organisatoren standardisierte Vorlagen für Events, Kommunikationsmails und Prozessschritte erstellen und wiederverwenden können. Durch automatisierte Abläufe und klar definierte Zuständigkeiten wird der Verwaltungsaufwand minimiert und die Nutzererfahrung optimiert.

#### Zentrale Merkmale des Produkts:

- Benutzerregistrierung mit organisationsspezifischer E-Mail-Domain
- Event-Erstellung und -Bearbeitung mit umfangreichen Konfigurationsmöglichkeiten
- Rollen- und Rechteverwaltung (Admin, Organisator, Benutzer)
- Prozesssteuerung mit individuell definierbaren Schritten
- Automatisierte E-Mail-Benachrichtigungen und Erinnerungen
- Archivierung vergangener Events
- Webbasierte, responsive Anwendung ohne zusätzliche Installation

Das System wird vollständig als Webanwendung entwickelt und soll über gängige Browser zugänglich sein. Es richtet sich primär an interne Unternehmensveranstaltungen, Schulungen, Workshops oder ähnliche In-Person-Events.

### 1.6. Produktdaten

Im Rahmen des Betriebs der Plattform fallen verschiedene Arten von Produktdaten an. Diese lassen sich in folgende Kategorien unterteilen:

#### 1. Stammdaten:

- Organisationsdaten (Name, E-Mail-Domain, Ansprechpartner)
- Benutzerprofile (Name, E-Mail, Rollen, Profilbild optional)
- Event-Daten (Titel, Ort, Zeitraum, Beschreibung, Teilnehmergrenzen etc.)

## **2. Bewegungsdaten:**

- Anmeldungen und Abmeldungen zu Events
- Interaktionen mit Prozessschritten (z. B. E-Mail-Versand, Bestätigungen, Dateizugriffe)
- Bearbeitungen von Events oder Benutzerinformationen

## **3. Konfigurationsdaten:**

- Templates für Events, E-Mails, Prozessschritte
- Rollen- und Rechtezuweisungen
- Ablaufpläne und Prozessschritte innerhalb von Events

## **4. System- und Protokolldaten:**

- Zeitstempel für Anmeldungen, Änderungen oder durchgeführte Aktionen
- Protokolle für Eventabsagen, Teilnehmerlimits oder automatische Prozesse

Diese Daten dienen der Sicherstellung der Systemfunktionalität, der Nachvollziehbarkeit von Prozessen sowie der Individualisierung der Benutzererfahrung. Alle personenbezogenen Daten werden gemäß den geltenden Datenschutzanforderungen verarbeitet.

### **1.7. Externe Systeme**

Das zu entwickelnde Online-Event-Management-System wird in einer webbasierten Architektur betrieben und besteht aus mehreren Komponenten, die miteinander kommunizieren. Es ist wichtig, die Schnittstellen zu externen Systemen zu definieren, um eine nahtlose Integration und den Betrieb des Systems sicherzustellen. Die folgenden Schnittstellen und Systemkontexte sind relevant:

#### **1.7.1. Externe Systeme**

Das Event-Management-System interagiert mit folgenden externen Systemen:

- **E-Mail-Server (SMTP):** Für den Versand von Bestätigungs-E-Mails, Benachrichtigungen über Event-Änderungen, Einladungen und Erinnerungen wird ein externer E-Mail-Server über das SMTP-Protokoll genutzt.

#### **1.7.2. Interne Systeme / Systemkomponenten**

Folgende Komponenten werden intern betrieben und sind Teil der Systemarchitektur:

- **Datenbank (PostgreSQL):** Das System verwendet eine relationale Datenbank zur Speicherung sämtlicher relevanter Daten (Benutzerdaten, Event-Details, Anmeldungen, Prozessschritte, Protokolle). Die Datenbank wird im Rahmen der Systembereitstellung selbst gehostet und ausschließlich von der Anwendung genutzt.

#### **1.7.3. Interne Schnittstellen**

Das System wird aus verschiedenen Komponenten bestehen, die über interne Schnittstellen miteinander kommunizieren:

- **Frontend und Backend:** Das Frontend (Web-Anwendung) wird über eine **RESTful API** mit dem Backend kommunizieren. Das Backend stellt auch sicher, dass die richtigen Daten in der Datenbank gespeichert werden.

#### 1.7.4. Datenformat und Kommunikation

- **JSON** wird als bevorzugtes Datenformat für die Kommunikation zwischen Frontend und Backend verwendet. Alle REST-API-Endpunkte werden JSON-Antworten liefern und akzeptieren, um eine einfache und effiziente Datenübertragung zu gewährleisten.
- **HTTPS:** Jegliche Kommunikation zwischen Front- und Backend erfolgt über HTTPS, um die Vertraulichkeit der übertragenen Daten sicherzustellen.

## 2. Funktionale Anforderungen

Die funktionalen Anforderungen legen die spezifischen Funktionen und Abläufe fest, die das zu entwickelnde System erfüllen muss. Sie beschreiben, welche Aufgaben das System ausführt, wie es auf bestimmte Eingaben reagiert und welche Prozesse unterstützt werden, um den Anforderungen der Nutzer gerecht zu werden.

### 2.1. Organisation hinzufügen

- Der Administrator kann eine neue Organisation über eine einfache Weboberfläche mithilfe eines Formulars hinzufügen.
- Die Eingabedaten (Organisationsname, E-Mail-Adresse etc.) werden an eine REST-API übermittelt.
- Das Backend verarbeitet die Anfrage, validiert die Eingaben und legt die Organisation in der Datenbank an.
- Automatisch wird ein Benutzerkonto mit der Rolle "Organisator" erstellt:
  - Der Benutzername entspricht der angegebenen Organisations-E-Mail-Adresse.
  - Es wird ein sicheres, zufällig generiertes Passwort zugewiesen.
- Das initiale Passwort und der Benutzername wird über eine E-Mail an den Organisator übermittelt.
- Das Passwort wird verschlüsselt in der Datenbank gespeichert.
- Beim ersten Login des ernannten Organisators muss das Passwort zwingend geändert werden, bevor Zugriff auf das System gewährt wird.
- Das System prüft bei jeder Anmeldung, ob das Initialpasswort noch aktiv ist, und erzwingt ggf. eine Weiterleitung zur Passwortänderung.
- Der gesamte Vorgang läuft ohne Unterbrechung des Systems ab – die Anwendung bleibt durchgängig verfügbar.
- Die Kommunikation erfolgt ausschließlich über HTTPS.

### 2.2. Event konfigurieren

- Organizer können innerhalb ihrer eigenen Organisation neue Event-Angebote über ein webbasiertes Formular erstellen.
- Jedes Event wird individuell konfiguriert und ist als In-Person-Termin konzipiert; die Durchführung des Events erfolgt außerhalb der Plattform.
- Das Formular zur Event-Erstellung enthält folgende Pflicht- und Eingabefelder:
  - Titel des Events (Pflichtfeld)
  - Coverbild (optional, Upload-Funktion)
  - Event-Kategorie (Auswahl aus vordefinierter Liste)

- Ort (frei wählbar, mit optionaler Kartenanzeige)
- Zeitraum: Datum und Uhrzeit (Start- und Endzeit)
- Optional: mehrtägiges Event, dann muss ein Start- und ein Enddatum eingegeben werden
- Mindest- und Maximalteilnehmeranzahl (Validierung erforderlich,  $\text{Min} \leq \text{Max}$ )
- Beschreibung (Freitextfeld mit Zeichenbegrenzung)
- Anmeldefrist (Datumsauswahl, darf nicht nach dem Event liegen)
- Kontaktperson (Auswahl aus den Mitgliedern der eigenen Organisation, automatisch aus Benutzerliste geladen)
- Optional: Zusätzlichen Organisator welcher alle Rechte an einer Zusammenarbeit des Events bekommt
- Änderung des verantwortlichen Organisators
  - Organizer können weitere Event Organizer zur gemeinsamen Verwaltung eines Events hinzufügen.
  - Bei Ausfall des ursprünglichen Organizer (z. B. aufgrund von Krankheit) kann ein berechtigter Nutzer aus derselben Organisation die Event-Verwaltung übernehmen, sofern er als zusätzlicher Organizer eingetragen ist. Die Eintragung eines weiteren Organisators ist auch nachträglich über die Bearbeitungsfunktion des Events möglich.
- Die Eventdaten werden über eine REST-API an das Backend übermittelt und in der Datenbank gespeichert.
- Organisatoren können für jedes Event optional einen Ablaufprozess mit individuellen Prozessschritten definieren.
- Das System prüft bei der Anmeldung von Teilnehmern, ob die maximale Teilnehmerzahl bereits erreicht ist.
- Sobald die maximale Teilnehmerzahl erreicht wurde, wird die Anmeldung automatisch deaktiviert.
- Die Eingaben werden client- und serverseitig validiert, um fehlerhafte oder unvollständige Eventkonfigurationen zu verhindern.
- Die gesamte Eventkonfiguration erfolgt über eine moderne, responsive Benutzeroberfläche mit klarer Benutzerführung.

### **2.3. Event bearbeiten**

- Organizer können eigen erstellte Events ihrer Organisation nachträglich bearbeiten, sofern das Event noch nicht gestartet ist.
- Die Bearbeitung erfolgt über ein spezielles Event-Bearbeitungsformular in der Weboberfläche, dass die vorhandenen Daten vorausfüllt.
- Bearbeitbare Inhalte sind:
  - Titel
  - Coverbild (Ersetzen oder Entfernen)
  - Event-Kategorie
  - Ort
  - Zeitraum (Start- und Endzeit oder -datum)
  - Mindest- und Maximalteilnehmeranzahl (nur Erhöhung der Maximalanzahl erlaubt)
  - Beschreibung
  - Anmeldefrist
  - Kontaktperson (aus Liste der Organisationsmitglieder)
- Der zugehörige Ablaufplan mit Prozessschritten (vgl. FA-2.16) kann angepasst oder erweitert werden, solange der Prozessschnitt (Start des Prozesses) noch nicht erfolgt ist.
  - Beispiel: Anpassung von E-Mail-Vorlagen oder Hinzufügen von Prozessschritten.

- Änderungen, die für Teilnehmer sichtbar sind (z. B. Ort, Zeit, Beschreibung), werden automatisch per E-Mail an bereits angemeldete Teilnehmer kommuniziert.
- Nach dem Startzeitpunkt eines Events sind nur noch eingeschränkte Bearbeitungen möglich:
  - Sichtbare Änderungen sind gesperrt.
  - Administrative Bearbeitungen zu Dokumentationszwecken oder Feedback-Eingaben bleiben erlaubt.
- Die Prüfung des Eventstatus (noch nicht gestartet, gestartet, abgeschlossen) erfolgt serverseitig vor dem Speichern von Änderungen.
- Alle Änderungen werden versionsbasiert dokumentiert und sind über eine Änderungsverfolgung nachvollziehbar.

#### **2.4. Event absagen**

- Organizer können Events jederzeit absagen, solange das Event noch nicht gestartet wurde.
- Die Absage erfolgt über die Event-Verwaltungs Oberfläche mit einem entsprechenden „Event absagen“-Button.
- Eine automatische Absage erfolgt, wenn die Mindestanzahl an Teilnehmern welche bei der Erstellung des Events angegeben wurde bis zu einem im voraus definierten Zeitpunkt vor Veranstaltungsbeginn nicht erreicht wurde.
- Bei manueller oder automatischer Absage wird eine Benachrichtigungs-E-Mail an alle angemeldeten Teilnehmer sowie den Organisator versendet.
  - Die Mail enthält folgende Informationen: Name des Events, Datum, Uhrzeit, Ort, Grund der Absage (z. B. zu wenige Teilnehmer, durch Organisator abgesagt).
  - Der Organisator kann der automatisierten Mail einen personalisierten Nachrichtentext hinzufügen.
- Die Absage wird in der Datenbank protokolliert und das Event als „abgesagt“ markiert, sodass keine Anmeldung mehr möglich ist.

#### **2.5. Templates**

- Organizer haben Zugriff auf Templates zur Unterstützung bei der Eventgestaltung.
- Initiale Templates (z. B. Beispielprozesse, E-Mail-Vorlagen) werden von NextGen Development bereitgestellt und in das System integriert.
- Organizer können eigene Templates innerhalb der Plattform erstellen, bearbeiten und speichern.
- Templates können z. B. für Prozessschritte und E-Mails verwendet werden.
- Templates enthalten strukturierte Inhalte, die für die Darstellung und Kommunikation eines Events relevant sind.
- Es besteht die Möglichkeit, Platzhalter (z. B. für Namen, Termine) zu integrieren, die beim Einsatz dynamisch ersetzt werden.
- Die Umsetzung erfolgt über ein Formular im Admin-Bereich. Templates werden in einer eigenen Datenbanktabelle gespeichert und verwaltet. Platzhalter innerhalb der Vorlagen (z. B. Name des Events, Kontaktperson) werden beim Versand automatisch ersetzt.
- Jeder Organisation stehen eigene Templates zur Verfügung. Zusätzlich werden systemweit verfügbare Standard-Templates bereitgestellt, die von NextGen Development initial definiert wurden. Diese Standard-Templates dienen als Grundlage und können von jeder Organisation individuell angepasst, aber nicht organisationsübergreifend verändert oder eingesehen werden.



### **2.6. Auf Plattform registrieren**

- Benutzer können sich eigenständig über ein Registrierungsformular auf der Plattform anmelden; bei der Anmeldung wird die Domain der eingegebenen E-Mail-Adresse mit den in der Datenbank hinterlegten Hostnamen verglichen.
- Nur wenn die Domain mit einem registrierten Host übereinstimmt, wird der Benutzer einer Organisation zugeordnet und die Registrierung abgeschlossen.
- Nach erfolgreicher Registrierung kann der Benutzer alle Events seiner Organisation einsehen.
- Nach der Anmeldung erfolgt eine E-Mail-Verifikation zur Aktivierung des Kontos.

### **2.7. Profil bearbeiten**

- Nach der Registrierung können Benutzer über ihr Benutzerprofil ihre E-Mail-Adresse und ihr Passwort ändern; dabei wird bei Änderung der E-Mail eine Verifikation per Mail durchgeführt.
- Optional kann der Benutzer ein Profilbild hochladen; dieses wird clientseitig validiert (Dateityp, Größe) und serverseitig gespeichert.
- Die Änderungen erfolgen über ein Formular im Benutzerprofil-Bereich und werden nach erfolgreicher Validierung in der Datenbank aktualisiert.
- Passwortänderungen erfordern die Eingabe des aktuellen Passworts sowie ein neues Passwort gemäß festgelegter Sicherheitsrichtlinien.

### **2.8. Benutzer löschen**

- Owner können Benutzer ihrer eigenen Organisation über eine Verwaltungstabelle entfernen; die Tabelle listet alle Benutzer der Organisation mit Bearbeitungsoptionen auf.
- Die Entfernung erfolgt durch einen Klick auf den „Entfernen“-Button in der jeweiligen Zeile; nach einer Sicherheitsabfrage wird der Löschvorgang ausgelöst.
- Beim Löschen eines Benutzers wird dieser automatisch aus allen Events entfernt, für die er aktuell angemeldet ist. Dies erfolgt durch das Entfernen seiner Benutzer-ID aus den jeweiligen Teilnehmerlisten.
- Im Anschluss werden alle von der Teilnehmeranzahl abhängigen Prozessschritte aktualisiert. Dazu zählen u. a. das Nachrücken von Wartelistenplätzen, das Erreichen oder Unterschreiten der Mindestteilnehmeranzahl sowie eventuell ausgelöste Benachrichtigungen oder Statusänderungen im Event-Prozess.
- Die Benutzerzuordnung zur Organisation wird ebenfalls gelöscht, und der Vorgang wird zur Nachvollziehbarkeit im Systemprotokoll gespeichert.
- Handelt es sich bei dem Benutzer um einen Organisator, kann er nicht direkt gelöscht werden.
  - Vor dem Löschen wird geprüft, ob er alleiniger Organisator in einem oder mehreren Events ist.
  - Ist dies der Fall, muss zunächst einem dieser Events ein weiterer Organisator zugewiesen werden, bevor die Löschung fortgesetzt werden kann.
- Handelt es sich bei dem Benutzer um einen Administrator, wird geprüft, ob nach der Löschung noch mindestens ein weiterer aktiver Admin im System verbleibt.
  - Die Löschung des letzten verbliebenen Administrators ist nicht zulässig, um die Administrierbarkeit des Systems sicherzustellen.

### **2.9. Organizer einladen**

- Ein Owner kann bestehende Benutzer seiner Organisation über eine Benutzerverwaltungstabelle zu Organizer ernennen; die Tabelle zeigt alle Benutzer der Organisation mit Rollenbearbeitung an.

- Die Berechtigung erfolgt durch Auswahl des gewünschten Benutzers und Zuweisung der Rolle „Organisator“ über ein Dropdown oder einen „Berechtigen“-Button.
- Eingeladene Benutzer erhalten automatisch alle administrativen Rechte innerhalb der Organisation, z. B. zum Erstellen, Bearbeiten und Verwalten von Events.
- Die neue Rolle wird direkt in der Datenbank gespeichert und ist sofort wirksam.
- Änderungen an Rollen werden systemseitig protokolliert und sind im Änderungsverlauf einsehbar.
- Der ernannte Benutzer wird per E-Mail über seinen Statuswechsel benachrichtigt

#### **2.10. Owner entfernen**

- Owner können anderen Organisatoren innerhalb ihrer Organisation die Rolle entziehen; dies erfolgt über die Benutzerverwaltungstabelle per Rollenänderung.
- Beim Entfernen der Owner -Rolle wird der betroffene Benutzer automatisch auf die Rolle „Benutzer“ zurückgestuft, bleibt aber in der Organisation erhalten.
- Vor dem Entzug prüft das System, ob mindestens ein weiterer Owner in der Organisation vorhanden ist.
- Ist der zu entfernende Owner der letzte in der Organisation, wird der Vorgang blockiert, bis ein neuer Organisator zugewiesen wurde.
- Die Änderung der Rolle wird direkt in der Datenbank aktualisiert und im Systemprotokoll dokumentiert.
- Vor dem zurückstufen wird geprüft, ob er alleiniger Owner in einem oder mehreren Events ist. Ist dies der Fall, muss zunächst einem dieser Events ein weiterer Organisator zugewiesen werden, bevor fortgesetzt werden kann.

#### **2.11. Events anzeigen**

- Benutzer können alle verfügbaren Events innerhalb ihrer Organisation über eine Eventübersicht einsehen; diese zeigt nur Events an, die der Organisation des Benutzers zugeordnet sind.
- Die Eventübersicht wird automatisch beim Aufruf der Eventseite geladen und enthält grundlegende Informationen wie Titel, Datum, Ort und Status.
- Eine integrierte Filterfunktion ermöglicht die gezielte Suche nach Events anhand verschiedener Kriterien:
  - Nach Titel mittels Textsuche
  - Nach Datum über eine Datumsauswahl oder einen Zeitraum
  - Optional: nach Event-Kategorie über ein Dropdown-Menü
- Die Filter werden direkt in der Benutzeroberfläche angewendet und wirken sich unmittelbar auf die angezeigte Liste aus; bei größeren Datenmengen wird serverseitig gefiltert, um die Performance zu sichern.
- Alle Daten werden über eine REST-API aus der Datenbank geladen und nach Auswahl dynamisch aktualisiert, ohne dass die Seite neu geladen werden muss.

#### **2.12. Event-Overview**

- Der Benutzer kann eine Übersicht aller Events einsehen, zu denen er aktuell angemeldet ist; diese Übersicht wird aus der Datenbank geladen und auf den angemeldeten Benutzer gefiltert.
- In der Übersicht werden pro Event zentrale Informationen angezeigt: Event-Name, Datum, Uhrzeit, Status (z. B. bevorstehend, laufend, abgeschlossen) und Veranstaltungsort.

- Die Events werden standardmäßig chronologisch sortiert; die Sortierung kann vom Benutzer geändert werden (z. B. aufsteigend oder absteigend nach Datum).
- Über ein Suchfeld und Filteroptionen kann der Benutzer gezielt nach Events suchen – z. B. nach Datum, Titel oder Status.
- Die Filter werden clientseitig angewendet und aktualisieren die Eventliste in Echtzeit.
- Ein Klick auf ein Event führt den Benutzer zur zugehörigen Detailansicht, in der alle weiteren Eventinformationen angezeigt werden.

### **2.13. Detaillierte Informationen Event**

- Der Benutzer kann sich zu jedem Event, für das er berechtigt ist, eine Detailansicht mit allen relevanten Informationen anzeigen lassen; der Zugriff erfolgt über die Event-Übersicht oder Eventliste per Klick.
- Angezeigt werden u. a. die Event-Beschreibung, Datum und Uhrzeit, Veranstaltungsort, Kontaktperson sowie zusätzliche Informationen gemäß den bei der Erstellung hinterlegten Angaben.
- Falls für das Event Prozessschritte definiert wurden, werden diese ebenfalls angezeigt – z. B. Erinnerungen, Aufgaben oder Hinweise zur Teilnahmebestätigung.
- Die Detailansicht wird dynamisch aufgerufen, indem beim Klick auf ein Event dessen ID an die API übergeben wird; die zugehörigen Daten werden aus der Datenbank geladen und im Frontend dargestellt.
- Die Informationen sind ausschließlich lesbar und passen sich responsiv an verschiedene Endgeräte an.

### **2.14. Umgang mit bereits stattgefundenen Events**

- Vergangene Events werden automatisch als „abgeschlossen“ markiert und in einem separaten Archivbereich angezeigt; dieser ist für Benutzer über das Event-Menü zugänglich.
- Der Benutzer kann in diesem Archiv alle Events einsehen, an denen er bereits teilgenommen hat; die Events werden anhand des Benutzerprofils aus der Datenbank gefiltert.
- Abgeschlossene Events bleiben ein Jahr lang in der Datenbank gespeichert und sind weiterhin einsehbar, jedoch schreibgeschützt.
- Nach Ablauf von zwölf Monaten ab dem Event-Enddatum werden die Events endgültig aus der Datenbank gelöscht; dies erfolgt automatisch durch einen geplanten Hintergrundprozess.
- Das Archiv zeigt weiterhin zentrale Informationen wie Titel, Datum, Ort und Status in einer übersichtlichen, chronologisch sortierten Liste.

### **2.15. Zu einem Event anmelden/abmelden**

- Benutzer können sich zu einem Event ihrer Organisation anmelden, sofern die maximale Teilnehmerzahl noch nicht erreicht ist und die Anmeldefrist nicht abgelaufen ist; die Anmeldung erfolgt über einen Button in der Event-Detailansicht.
- Bei erfolgreicher Anmeldung wird der Benutzer in die Teilnehmerliste des Events aufgenommen und erhält eine Bestätigungsmail mit relevanten Eventinformationen.
- Benutzer können sich auch wieder von einem Event abmelden, sofern das Event noch nicht gestartet ist; der Abmeldebutton wird in diesem Fall angezeigt und aktiv geschaltet.
- Die Anmeldung und Abmeldung erfolgt über API-Aufrufe, die die Teilnehmerliste in der Datenbank aktualisieren. Vor dem Speichern prüft das System, ob das Event noch offen ist und die Teilnahmebedingungen erfüllt sind.
- Nach erfolgreicher Abmeldung wird der Benutzer aus der Teilnehmerliste entfernt.

### **2.16. Event-Prozesssteuerung konfigurieren**

- Organizer können für Events individuelle Prozessschritte konfigurieren, die bestimmte automatisierte Aktionen auslösen; die Prozessschritte werden vorab angelegt und während der Event-Konfiguration zugewiesen.
- Die Zuordnung erfolgt über ein Auswahlfeld in der Event-Erstellung, bei dem nur relevante Schritte aktiviert werden können.
- Jeder Prozessschritt enthält folgende konfigurierbare Elemente:
  - Name des Prozessschritts
  - Bedingungen, um festzulegen, wann Aktionen durchgeführt werden sollen
  - Aktion (z. B. automatisierter Mailversand, Deaktivierung der Anmeldung, Erinnerungsfunktion)
  - E-Mail-Vorlagen mit Platzhaltern wie {Teilnehmername}, {Eventtitel}, {Datum}, die beim Versand automatisch ersetzt werden
- Weitere mögliche Funktionen eines Prozessschritts sind:
  - Freischaltung bestimmter Informationen in der Benutzeroberfläche
  - Versand zeitgesteuerter Erinnerungen an Event-Verantwortliche
- Die Prozessschritte werden als separate Entitäten in der Datenbank gespeichert und über eine Weboberfläche gepflegt. Die zugeordneten Aktionen werden durch einen internen Scheduler oder Event-Handler ausgelöst. Platzhalter in Vorlagen werden beim Versand dynamisch ersetzt. Die Konfiguration erfolgt über ein Drag-and-Drop-Interface oder ein Formular mit strukturierter Eingabemaske.

## **3. Nichtfunktionale Anforderungen**

Die nicht-funktionalen Anforderungen definieren die Qualitätsmerkmale des Systems, wie es die gestellten Aufgaben ausführt. Sie umfassen Aspekte wie Leistung, Sicherheit, Benutzerfreundlichkeit und Skalierbarkeit und stellen sicher, dass das System auch unter spezifischen Bedingungen zuverlässig und effizient arbeitet.

### **3.1. Allgemeine Anforderungen**

Die allgemeinen Anforderungen beschreiben grundlegende Eigenschaften und Rahmenbedingungen, die das System unabhängig von konkreten Funktionen erfüllen muss. Sie betreffen insbesondere Aspekte wie Benutzerfreundlichkeit, Sicherheit und allgemeine Erwartungen an das Nutzungserlebnis. Diese Anforderungen stellen sicher, dass das System nicht nur funktional, sondern auch im täglichen Einsatz zuverlässig, verständlich und sicher ist.

#### **3.1.1. Benutzerfreundlichkeit**

- Das System muss eine intuitive und leicht verständliche Benutzeroberfläche bereitstellen, die eine schnelle Einarbeitung für alle Nutzergruppen (Benutzer, Organisatoren, Administratoren) ermöglicht.
- Die Navigation soll klar strukturiert und logisch aufgebaut sein, sodass zentrale Funktionen ohne Umwege erreichbar sind.
- Die Benutzerführung wird durch visuelle Hilfen (z. B. Icons, farbliche Hervorhebungen), Tooltips und einheitliche Designkonventionen unterstützt.
- Die Oberfläche wird responsiv gestaltet, damit das System auf verschiedenen Endgeräten (Desktop, Tablet, Smartphone) benutzerfreundlich bedienbar bleibt.
- Wiederkehrende Bedienelemente (z. B. Buttons, Formulare, Navigationsleisten) werden konsistent im gesamten System verwendet, um Orientierung und Vertrauen zu fördern.

### **3.1.2. Sicherheit**

- Die gesamte Kommunikation zwischen Frontend und Backend muss ausschließlich über HTTPS erfolgen, um Daten während der Übertragung zu schützen.
- Alle personenbezogenen Daten (z. B. Passwort) müssen verschlüsselt gespeichert und übertragen werden; Passwörter sind mittels sicherem Hash-Verfahren zu sichern.
- Ein Autorisierungskonzept muss implementiert werden, das den Zugriff auf APIs und interne Systemfunktionen prüft und unautorisierte Anfragen blockiert. Hierfür kann beispielsweise eine JWT-basierte Autorisierung verwendet werden.
- Rollen- und Berechtigungskonzepte müssen sicherstellen, dass Benutzer nur auf Funktionen und Daten zugreifen können, die ihrer Rolle entsprechen (Benutzer, Organisator, Admin).
- Zugriff auf administrative oder sicherheitsrelevante Bereiche muss zusätzlich gesichert werden.

### **3.2. Technische Anforderungen**

- Die Anwendung muss webbasiert umgesetzt werden, sodass sie ohne lokale Installation direkt über einen Webbrowser aufrufbar ist; der Zugriff erfolgt über eine zentrale Projekt-URL.
- Die Benutzeroberfläche wird mit modernen Webtechnologien (React, Next.js) realisiert und für die Nutzung mit gängigen Browsern (Chrome, Firefox, Safari, Edge) optimiert.
- Durch den Einsatz responsiven Designs (Tailwind) wird sichergestellt, dass die Darstellung auf allen Endgeräten korrekt und benutzerfreundlich ist.
- Die Applikation wird auf einem Webserver gehostet, sodass sie jederzeit erreichbar ist.

### **3.3. Weitere nichtfunktionale Anforderungen**

- Eine verständliche Online-Dokumentation wird direkt in die Benutzeroberfläche integriert. Sie ist ohne Registrierung zugänglich.
- Die Dokumentation wird als statische HTML-Seiten bereitgestellt und erklärt zentrale Funktionen wie Benutzerverwaltung, Event-Konfiguration und Prozesssteuerung.
- Die Inhalte der Dokumentation werden versioniert gepflegt (über ein Git-Repository), um Konsistenz mit der jeweils aktiven Softwareversion zu gewährleisten.

### **3.4. Qualitätsziele**

- Benutzerfreundlichkeit:
  - Die Oberfläche wird mithilfe eines einheitlichen UI-Designs gestaltet (z. B. mit einem Designsystem oder UI-Komponentenbibliothek).
  - Visuelles Feedback (z. B. Ladeanzeigen, Erfolgsmeldungen) und Tooltips unterstützen den Nutzer bei der Bedienung.
- Zuverlässigkeit:
  - Das System basiert auf einem stabilen Backend-Framework, das automatisierte Aufgaben (E-Mail-Versand, Prozessausführungen) zuverlässig steuert.
  - Fehler werden durch ein Logging-System erfasst (Log-Dateien, Monitoring-Tools), um Ausfälle frühzeitig zu erkennen.
- Sicherheit und Datenschutz:
  - Die Anwendung verwendet HTTPS für alle Verbindungen und speichert Passwörter nur gehasht.
  - Ein Rollen- und Rechtesystem regelt alle Zugriffsrechte.
  - Zugriff auf personenbezogene Daten erfolgt nur über gesicherte und autorisierte Schnittstellen.
- Skalierbarkeit:

- Die Architektur folgt einem modularen Aufbau, sodass Last durch horizontale Skalierung bewältigt werden kann.
  - Datenbank und Server sind so konfiguriert, dass bei zunehmender Nutzung keine signifikanten Performanceverluste entstehen.
- Erweiterbarkeit:
  - Die Codebasis wird nach dem Prinzip der Trennung von Logik, Daten und Darstellung aufgebaut, um Erweiterungen ohne tiefgreifende Änderungen zu ermöglichen.
  - Neue Features können über Konfigurationsoptionen oder Plug-ins eingebunden werden.
- Wartbarkeit:
  - Der Quellcode wird nach gängigen Konventionen strukturiert (Clean Code, Namenskonventionen, Modulstruktur).
  - Alle Module werden dokumentiert und versioniert (über Git).
  - Fehlerbehandlung, Logging und automatisierte Tests (Unit- und Integrationstests) ermöglichen eine sichere Weiterentwicklung.