

# Pflichtenheft: Sauberfix Terminverwaltung

## 1. Einleitung

### 1.1 Projektbeschreibung

Sauberfix ist eine webbasierte Anwendung zur digitalen Verwaltung von Reinigungsterminen, Mitarbeitern und Kunden für Reinigungsunternehmen. Das System ersetzt manuelle Planungsprozesse durch eine zentrale, digitale Lösung mit grafischer Dispo-Tafel.

### 1.2 Zielsetzung

Ziel ist die Effizienzsteigerung bei der Einsatzplanung, die Vermeidung von Terminkollisionen und die Sicherstellung der Kommunikation durch automatische Erinnerungen.

---

## 2. Funktionale Anforderungen (Muss-Kriterien)

### 2.1 Benutzerverwaltung & Rollen

- **FA-010:** Das System muss zwei Benutzerrollen unterstützen: "Admin" und "User" (Mitarbeiter).
- **FA-011:** Admins haben vollen Zugriff auf alle Daten und Funktionen.
- **FA-012:** User (Mitarbeiter) dürfen nur ihre eigenen Termine einsehen und keine Verwaltungsdaten ändern.
- **FA-013:** Authentifizierung erfolgt mittels Benutzername und Passwort.

### 2.2 Kundenverwaltung

- **FA-020:** Anlegen, Bearbeiten und Löschen von Kundenstammdaten.
- **FA-021:** Erfassung von: Vorname, Nachname, Firma, Adresse (inkl. PLZ/Ort), E-Mail, Telefon.
- **FA-022:** Orte (PLZ/Stadt) müssen normalisiert gespeichert werden.

### 2.3 Mitarbeiterverwaltung

- **FA-030:** Anlegen, Bearbeiten und Löschen von Mitarbeiterkonten.
- **FA-031:** Zuweisung von Zugangsdaten und Rollen.
- **FA-032:** Passwörter dürfen nicht im Klartext gespeichert werden.

### 2.4 Terminverwaltung & Kalender

- **FA-040:** Erstellung von Terminen mit Startzeit, Endzeit, zugewiesenem Kunden und Mitarbeiter.
- **FA-041: Kollisionsprüfung:** Das System muss verhindern, dass ein Mitarbeiter zwei Termine zur gleichen Zeit hat.
- **FA-042: Grafische Plantafel:** Visuelle Darstellung aller Termine in einer Kalenderansicht (Ressourcenansicht pro Mitarbeiter).
- **FA-043:** Unterstützung von Drag & Drop zum Verschieben von Terminen.
- **FA-044:** Terminstatus-Verwaltung (Geplant, Erledigt, Storniert).

### 2.5 Benachrichtigungen

- **FA-050:** Automatischer Versand von E-Mail-Erinnerungen an Kunden.
  - **FA-051:** Erinnerungen werden 24 Stunden vor Terminbeginn versendet.
  - **FA-052:** Das System prüft periodisch (Hintergrunddienst) auf anstehende Termine.
-

### 3. Nicht-funktionale Anforderungen

#### 3.1 Sicherheit

- **NFA-010:** Passwörter müssen mit BCrypt gehasht werden.
- **NFA-011:** Die Kommunikation muss über HTTPS verschlüsselt sein.
- **NFA-012:** Schutz vor XSS-Angriffen in allen Eingabefeldern.
- **NFA-013:** API-Zugriff nur mit gültigem JWT-Token.

#### 3.2 Systemarchitektur & Betrieb

- **NFA-020:** Backend: .NET 9.0 (ASP.NET Core Minimal APIs).
- **NFA-021:** Frontend: HTML5, CSS, JavaScript (kein schwergewichtiges Framework).
- **NFA-022:** Datenbank: PostgreSQL.
- **NFA-023:** Containerisierung mittels Docker.
- **NFA-024:** Lauffähigkeit in Kubernetes-Clustern.

#### 3.3 Performance

- **NFA-030:** Die Kalenderansicht muss auch bei vielen Terminen (>100 pro Woche) flüssig bedienbar bleiben.
- 

### 4. Datenmodell

Das System basiert auf folgenden Kern-Entitäten: 1. **Mitarbeiter** (User-Accounts) 2. **Kunde** (Stammdaten) 3. **Ort** (Normalisierte Adressdaten) 4. **Termin** (Verknüpfung von Zeit, Kunde und Mitarbeiter)

---

### 5. Abnahmekriterien

Das System gilt als abgenommen, wenn: 1. Ein Admin einen kompletten Workflow (Kunde anlegen -> Mitarbeiter anlegen -> Termin planen) durchführen kann. 2. Doppelbuchungen vom System aktiv verhindert werden. 3. E-Mail-Erinnerungen für Testtermine korrekt zugestellt werden. 4. Die Anwendung via Docker-Container gestartet werden kann.