
<Company Name>

**SWE-Büroplatzverwaltung
Software Requirements Specification**

Version 1.1

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

Revision History

| Date | Version | Description | Author |
|------------|---------|-------------------|--------|
| 18.10.2024 | 1.0 | Erstellt | alle |
| 26.10.2024 | 1.1 | Mockups eingefügt | Hagen |
| | | | |
| | | | |

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

Table of Contents

| | | |
|-------|---------------------------------------------------------|----|
| 1. | Introduction | 4 |
| 1.1 | Purpose | 4 |
| 1.2 | Scope | 4 |
| 1.3 | Definitions, Acronyms and Abbreviations | 4 |
| 1.4 | References | 4 |
| 1.5 | Overview | 4 |
| 2. | Overall Description | 5 |
| 3. | Specific Requirements | 6 |
| 3.1 | Functionality | 6 |
| 3.1.1 | Platz buchen (Christoph) | 6 |
| 3.1.2 | Use-Case Specification: Stockwerk anzeigen | 7 |
| 3.1.3 | Platz stornieren (Moritz) | 9 |
| 3.1.4 | Buchungszeitraum anpassen (Jannik) | 10 |
| 3.1.5 | Defekt melden (Leon) | 11 |
| 3.1.6 | Plätze verwalten (Christoph) | 12 |
| 3.1.7 | Employees verwalten (Moritz) | 13 |
| 3.1.8 | Defekte auflösen (Jannik) | 16 |
| 3.1.9 | Flurpläne erstellen (Leon) | 17 |
| 3.2 | Usability | 17 |
| 3.3 | Reliability | 17 |
| 3.3.1 | Uptime | 17 |
| 3.3.2 | Keine Einarbeitungszeit erforderlich | 17 |
| 3.3.3 | Vertraute UI/UX | 17 |
| 3.4 | Performance | 17 |
| 3.4.1 | Performance | 17 |
| 3.4.2 | Capacity | 17 |
| 3.5 | Supportability | 17 |
| 3.5.1 | Coding Standards | 17 |
| 3.5.2 | Naming Conventions | 18 |
| 3.5.3 | Testing | 18 |
| 3.6 | Design Constraints | 18 |
| 3.6.1 | Architecture | 18 |
| 3.6.2 | Tech stack | 18 |
| 3.6.3 | Development tools | 18 |
| 3.7 | On-line User Documentation and Help System Requirements | 18 |
| 3.8 | Purchased Components | 18 |
| 3.9 | Interfaces | 18 |
| 3.9.1 | User Interfaces | 18 |
| 3.9.2 | Hardware Interfaces | 18 |
| 3.9.3 | Software Interfaces | 18 |
| 3.9.4 | Communications Interfaces | 18 |
| 3.10 | Licensing Requirements | 19 |
| 3.11 | Legal, Copyright, and Other Notices | 19 |
| 3.12 | Applicable Standards | 19 |
| 4. | Supporting Information | 19 |

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

Software Requirements Specification

1. Introduction

1.1 Purpose

Dieses Dokument beschreibt die funktionalen Anforderungen für Entwicklung eines webbasierten Systems zur Verwaltung und Vermietung von Büroplätzen (Shared Office).

1.2 Scope

Die Software ermöglicht es Benutzern, verfügbare Büroplätze (grafisch) anzuzeigen und zu buchen.

1.3 Definitions, Acronyms and Abbreviations

n/a

1.4 References

[Github](#)

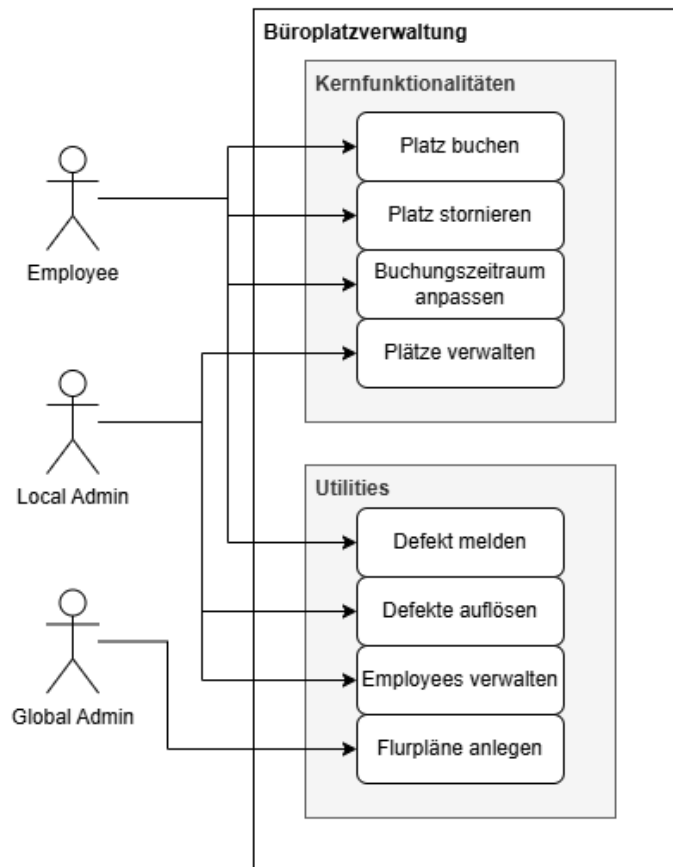
[Blog](#)

1.5 Overview

Diese Spezifikation beschreibt die allgemeine Funktionsweise des Software-Produkts und listet die Kern-Funktionalitäten zusammen mit ihren use-case Definitionen auf.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

2. Overall Description



- *product perspective*

Als Startup wollen wir Unternehmen mit Shared Desks eine Lösung anbieten, um als Mitarbeiter Arbeitsplätze reservieren zu können.

- *product functions*

Anzeigen und buchen von Plätzen in einer Weboberfläche.

- *user characteristics*

Normale Benutzer sind Mitarbeiter in Firmen, die Sitzplätze und/oder Räume einsehen und buchen können.

Local Admins sind ausgewählte Mitarbeiter der Firma, die das Zugriffsmanagement innerhalb der Firma regeln.

Global Admins sind Mitarbeiter unseres Startups, die für Kunden die unternehmensspezifische Umgebung anlegen.

- *constraints*

Belegung funktioniert auf "Trust-Basis": MAs können sich ohne Buchung hinsetzen oder buchen und nicht beanspruchen.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

- *assumptions and dependencies*

Das System erfordert eine stabile Internetverbindung.
Es wird auf einem Cloud-Hosting-Service bereitgestellt.

3. Specific Requirements

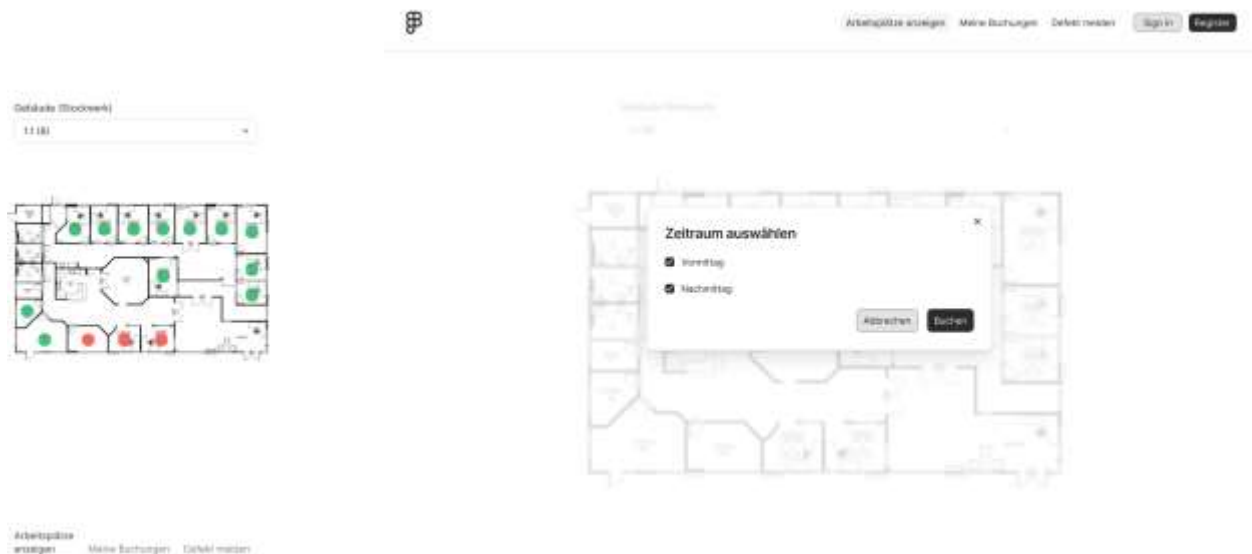
3.1 Functionality

3.1.1 Platz buchen (Christoph)

1. Kurze Beschreibung

Dieser Use case ermöglicht es den Nutzern/Mitarbeitern einen von Ihnen ausgewählten Büroplatz zu buchen. Der Mitarbeiter muss hierzu seinen Platz auswählen sowie den Zeitraum, an dem er diesen belegen möchte.

1.2 Mockup



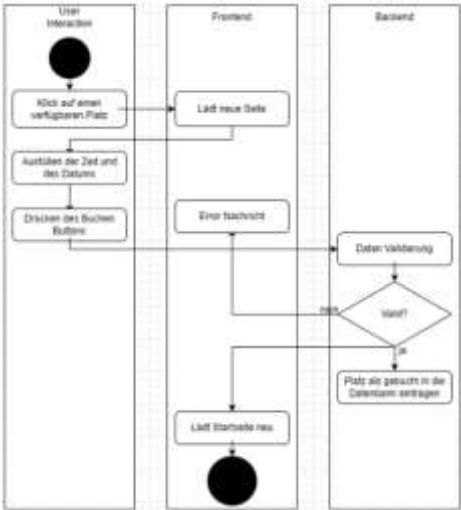
1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

- Nutzer ist angemeldet
- Klickt auf einen verfügbaren Platz
- Füllt Datum und Uhrzeit aus
- Drückt den Buchen Knopf
- Platz wird gebucht

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |



Aktivitätsdiagramm 1

2.2 Alternativer Ablauf

n/a

3. Besondere Anforderungen

n/a

4. Vorbedingungen

Der Nutzer ist angemeldet.

5. Nachbedingungen

Der Büroplatz wird für die anderen Mitarbeiter gesperrt.
Der Mitarbeiter bekommt eine Bestätigung über die Buchung

5.1 Änderungen speichern/ Synchronisation mit dem Server

Der Server muss sicherstellen, dass die Buchung gespeichert wird und der Büroplatz nicht mehr für die anderen buchbar ist.

6. Aufwandschätzung

3.1.2 Use-Case Specification: Stockwerk anzeigen

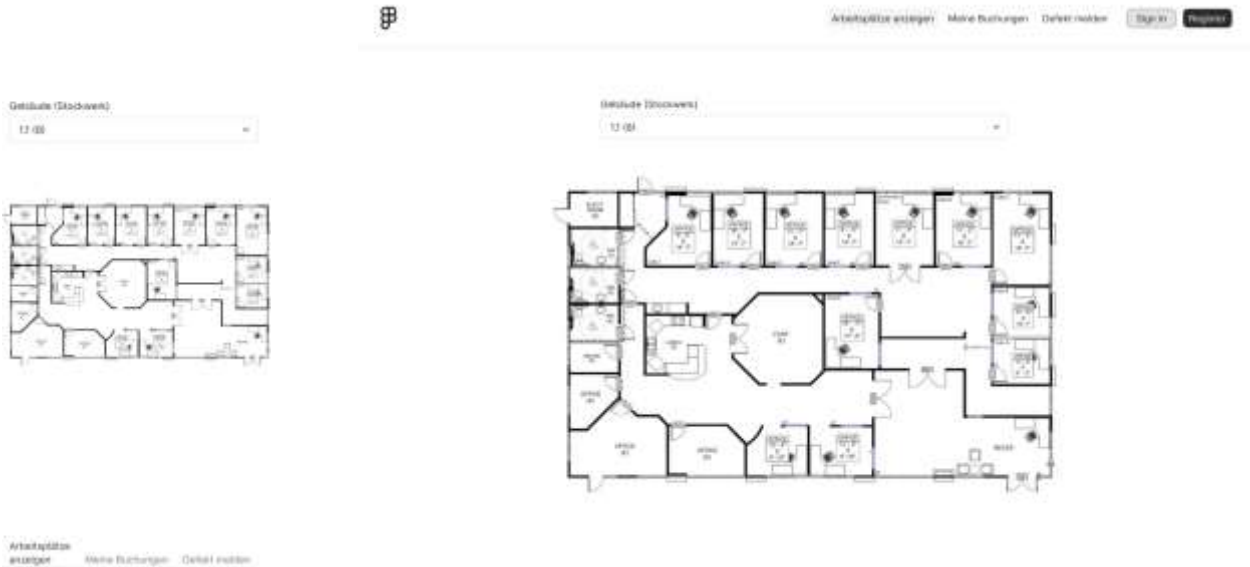
1. Stockwerk anzeigen

1.1 Kurze Beschreibung

Dieser Use-Case ermöglicht es einem Benutzer, ein bestimmtes Stockwerk eines ausgewählten Gebäudes anzuzeigen. Der Benutzer wählt zunächst ein Gebäude auf der Hauptseite aus. Danach wird eine Übersicht der Etagen dieses Gebäudes angezeigt, und der Benutzer kann auf eine Etage klicken, um den Grundriss und die verfügbaren Büroplätze einzusehen.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

1.2 Mockup



1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

Benutzer navigiert auf die Hauptseite der Website.
Benutzer wählt ein Gebäude aus der Liste der verfügbaren Gebäude aus.
Die Etagenübersicht für das gewählte Gebäude wird angezeigt.
Benutzer klickt auf eine Etage, um den Grundriss dieser Etage anzuzeigen.
Der Benutzer kann dann einen verfügbaren Büroplatz anklicken, um weitere Details zu diesem Platz zu sehen (z.B. Ausstattung, Verfügbarkeit).
Ein Popup mit den Büroplatzdetails wird angezeigt.

2.2 Alternative Abläufe

n/a

3. Besondere Anforderungen

n/a

4. Vorbedingungen

Der Benutzer hat die Website gestartet.
Der Benutzer befindet sich auf der Hauptseite der Anwendung.
Der Benutzer ist angemeldet (falls erforderlich, abhängig von den Berechtigungen).

5. Nachbedingungen

Der Grundriss des gewählten Stockwerks wird korrekt angezeigt.
Es wird ein Popup mit den Details des ausgewählten Büroplatzes angezeigt, wenn ein Büroplatz angeklickt wird.

5.1 Änderungen speichern / Synchronisation mit dem Server

Der Server muss sicherstellen, dass die Frontend-Anwendung die korrekten Daten zum gewählten Gebäude und den zugehörigen Stockwerken bereitstellt. Es wird keine Änderung an den gespeicherten Büroplatzdaten vorgenommen.

6. Funktionspunkte

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

3.1.3 Platz stornieren (Moritz)

Use-Case Specification: Platz Stornieren

1. Platz Stornieren

1.1 Kurze Beschreibung

Dieser Use-Case ermöglicht es einen gebuchten Platz zu stornieren

1.2 Mockup



1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

Der Nutzer navigiert zur Übersicht seiner Buchungen, die in einem speziellen Bereich für gebuchte Plätze angezeigt wird.

Das System zeigt eine Liste aller zukünftigen und laufenden Buchungen des Nutzers an, zusammen mit Optionen zur Stornierung.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

Der Nutzer wählt die Buchung aus, die er stornieren möchte.

Das System zeigt Details der ausgewählten Buchung an.

Der Nutzer klickt auf die Schaltfläche “Buchung stornieren”.

Das System storniert die Buchung und markiert den entsprechenden Arbeitsplatz als verfügbar. Der Nutzer wird mit einer Erfolgsmeldung benachrichtigt.

2.2 Alternative Abläufe

Netzwerk- oder Systemfehler

Sollte während des Stornierungsprozesses ein technisches Problem auftreten, wird der Nutzer durch eine Fehlermeldung informiert und kann die Aktion später erneut versuchen.

3. Besondere Anforderungen

Die Stornierung sollte möglichst in Echtzeit ablaufen, um sicherzustellen, dass der Arbeitsplatz sofort wieder für andere Nutzer verfügbar ist.

4. Vorbedingungen

Der Nutzer ist erfolgreich im System eingeloggt.

Der Nutzer hat bereits einen Arbeitsplatz gebucht, der noch nicht abgelaufen ist (Buchung in der Zukunft oder für den aktuellen Tag).

5. Nachbedingungen

Die Buchung wird erfolgreich storniert und der Arbeitsplatz steht wieder zur Verfügung.

3.1.4 *Buchungszeitraum anpassen (Jannik)*

1. Kurze Beschreibung

Dieser Use-Case ermöglicht es einem Benutzer, den Buchungszeitraum eines bereits reservierten Büroplatzes zu ändern. Der Benutzer kann nach Auswahl eines Büroplatzes den Start- und Endzeitpunkt der Buchung anpassen. Die Verfügbarkeit des Platzes wird vor der Bestätigung überprüft.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

1.2 Mockup

1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

Benutzer navigiert zu seiner Buchungsliste.

Benutzer wählt eine bestehende Buchung aus.

Die Details der Buchung werden angezeigt, einschließlich des aktuellen Buchungszeitraums.

Benutzer klickt auf "Buchungszeitraum anpassen".

Der Benutzer ändert den Start- und/oder Endzeitpunkt der Buchung.

Die Verfügbarkeit des Platzes für den neuen Zeitraum wird überprüft.

Benutzer bestätigt die Anpassung.

Die angepasste Buchung wird gespeichert, und der Benutzer erhält eine Bestätigung.

2.2 Alternative Abläufe

Wenn der Büroplatz für den neuen Zeitraum nicht verfügbar ist, wird eine Fehlermeldung angezeigt, und der Benutzer muss einen anderen Zeitraum auswählen.

3. Besondere Anforderungen

n/a

4. Vorbedingungen

Der Benutzer ist angemeldet.

Der Benutzer hat eine bestehende Buchung.

Der Benutzer befindet sich in der Buchungsliste.

5. Nachbedingungen

Der angepasste Buchungszeitraum wird korrekt gespeichert.

Der Benutzer erhält eine Bestätigung über die erfolgreiche Anpassung.

5.1 Änderungen speichern / Synchronisation mit dem Server

Der Server muss sicherstellen, dass die neuen Buchungsdaten korrekt synchronisiert und gespeichert werden. Es darf keine Überschneidung mit anderen Buchungen bestehen.

6. Funktionspunkte

3.1.5 Defekt melden (Leon)

Use-Case Specification: Defekt melden

1. Defekt melden

1.1 Kurze Beschreibung

Dieser Use-Case ermöglicht es einem Benutzer, einen Defekt an einem gebuchten Büroplatz zu melden. Wenn der Benutzer an einem Platz einen Defekt feststellt, kann er diesen über die Plattform melden. Dies führt dazu, dass der Platz vorübergehend gesperrt wird, bis der Defekt behoben ist.

1.2 Mockup

1.3 Screenshots

2. Flow of Events

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

2.1 Grundlegender Ablauf

Der Benutzer navigiert zur Detailansicht eines von ihm gebuchten Büroplatzes.

Der Benutzer entdeckt einen Defekt und klickt auf den Button "Defekt melden".

Ein Formular öffnet sich, in dem der Benutzer den Defekt beschreiben kann (z.B. kaputte Ausrüstung, fehlende Ausstattung).

Der Benutzer sendet die Meldung ab.

Der Büroplatz wird vorübergehend gesperrt, bis der Defekt behoben ist, und der Administrator wird über die Meldung informiert.

Eine Bestätigung wird dem Benutzer angezeigt, dass der Defekt gemeldet wurde.

2.2 Alternative Abläufe

Defekt nicht bestätigbar: Sollte der Defekt nicht bestätigt werden (z.B. technische Probleme bei der Meldung), erhält der Benutzer eine Fehlermeldung und wird gebeten, es erneut zu versuchen.

3. Besondere Anforderungen

Die Defektmeldungs-Funktion sollte nur für Benutzer verfügbar sein, die den Büroplatz gebucht haben.

4. Vorbedingungen

Der Benutzer ist angemeldet.

Der Benutzer hat den Büroplatz für den angegebenen Zeitraum gebucht.

Der Benutzer befindet sich auf der Detailseite des Büroplatzes.

5. Nachbedingungen

Der Defekt wird erfolgreich gemeldet und an das System übermittelt.

Der Büroplatz wird gesperrt, bis der Defekt behoben wurde.

Der Administrator erhält eine Benachrichtigung über den gemeldeten Defekt.

5.1 Änderungen speichern / Synchronisation mit dem Server

Der Server muss sicherstellen, dass der gemeldete Defekt gespeichert wird und der Büroplatz für andere Benutzer nicht mehr buchbar ist, bis die Meldung bearbeitet wurde.

6. Funktionspunkte

3.1.6 Plätze verwalten (Christoph)

1. Kurze Beschreibung

Dieser Use case ermöglicht es den Nutzern/Mitarbeitern einen von Ihnen ausgewählten Büroplatz zu buchen. Der Mitarbeiter muss hierzu seinen Platz auswählen sowie den Zeitraum an dem er diesen belegen möchte.

1.2 Mockup

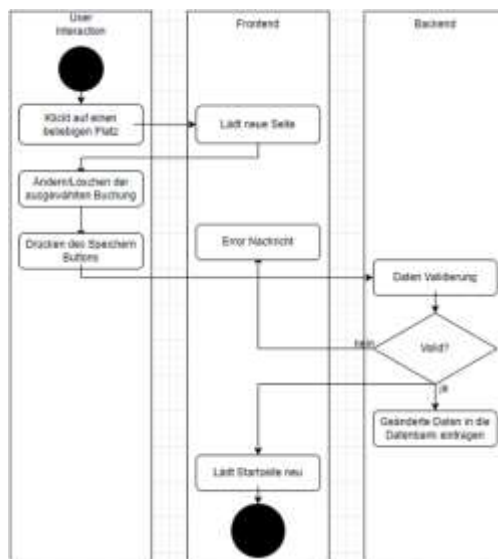
1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

- Admin ist angemeldet
- Hat Überblick über alle Buchungen
- Klickt eine Buchung an
- Kann diese ändern/löschen

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |



Aktivitätsdiagramm 2

2.2 Alternativer Ablauf

n/a

3. Besondere Anforderungen

Berechtigung eines local Admin oder höher nötig

4. Vorbedingungen

Als local Admin oder höher angemeldet

5. Nachbedingungen

n/a

6. Aufwandschätzung

3.1.7 Employees verwalten (Moritz)

Use-Case Specification: Employees verwalten

1. Employees verwalten

1.1 Kurze Beschreibung

Der Office Manager möchte die Mitarbeiter im System verwalten. Dazu gehören das Hinzufügen, Bearbeiten, Entfernen von Mitarbeitern, die Zugang zum Buchungssystem haben.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

1.2 Mockup

1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

Der Office Manager navigiert zur Verwaltungsoberfläche, um die Liste der bestehenden Mitarbeiter anzuzeigen.

Das System zeigt eine Liste aller im System registrierten Mitarbeiter, zusammen mit ihren aktuellen Rollen und Status (aktiv, inaktiv).

Der Office Manager wählt eine der folgenden Aktionen:

Mitarbeiter hinzufügen

Mitarbeiter bearbeiten

Mitarbeiter deaktivieren oder entfernen

Hinzufügen eines neuen Mitarbeiters:

Der Office Manager klickt auf die Option „Mitarbeiter hinzufügen“ und gibt die erforderlichen Informationen ein (z. B. Name, E-Mail-Adresse, Abteilung).

Das System überprüft die eingegebenen Daten auf Vollständigkeit und Konflikte (z. B. doppelte E-Mail-Adresse oder bereits existierende Konten).

Bearbeiten eines bestehenden Mitarbeiters:

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

Der Office Manager wählt einen bestehenden Mitarbeiter aus der Liste aus, den er bearbeiten möchte (z. B. Rollenänderung, Abteilungswechsel).

Das System zeigt die aktuellen Informationen des Mitarbeiters in einem Bearbeitungsformular an.

Der Office Manager ändert die gewünschten Informationen (z. B. neue Abteilung).

Deaktivieren oder Entfernen eines Mitarbeiters:

Der Office Manager wählt einen Mitarbeiter aus der Liste, den er deaktivieren oder komplett aus dem System entfernen möchte.

Das System fordert eine Bestätigung für die Deaktivierung oder das Löschen an, um unbeabsichtigte Aktionen zu vermeiden.

Der Office Manager bestätigt die Deaktivierung oder das Löschen.

2.2 Alternative Abläufe

Wenn der Office Manager versucht, einen Mitarbeiter mit einer bereits registrierten E-Mail-Adresse hinzuzufügen, zeigt das System eine Fehlermeldung an und bietet Optionen zum Zusammenführen oder Bearbeiten bestehender Profile.

3. Besondere Anforderungen

Der Office Manager benötigt die entsprechenden Zugriffsrechte, um Mitarbeiter zu verwalten.

4. Vorbedingungen

Der Office Manager ist erfolgreich im System eingeloggt und hat die notwendigen Berechtigungen, um Mitarbeiter zu verwalten.

5. Nachbedingungen

Mitarbeiterprofile werden erfolgreich erstellt, geändert, deaktiviert oder gelöscht.

Die betroffenen Mitarbeiter werden über Änderungen (z. B. Deaktivierung oder neue Zugangsdaten) benachrichtigt.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

3.1.8 Defekte auflösen (Jannik)

1. Defekte auflösen

1.1 Kurze Beschreibung

Dieser Use-Case ermöglicht es einem lokalen Administrator, einen gemeldeten Defekt an einem Büroplatz aufzulösen, nachdem der Defekt in der realen Welt behoben wurde. Der Büroplatz wird nach der Auflösung des Defekts wieder zur Buchung freigegeben.

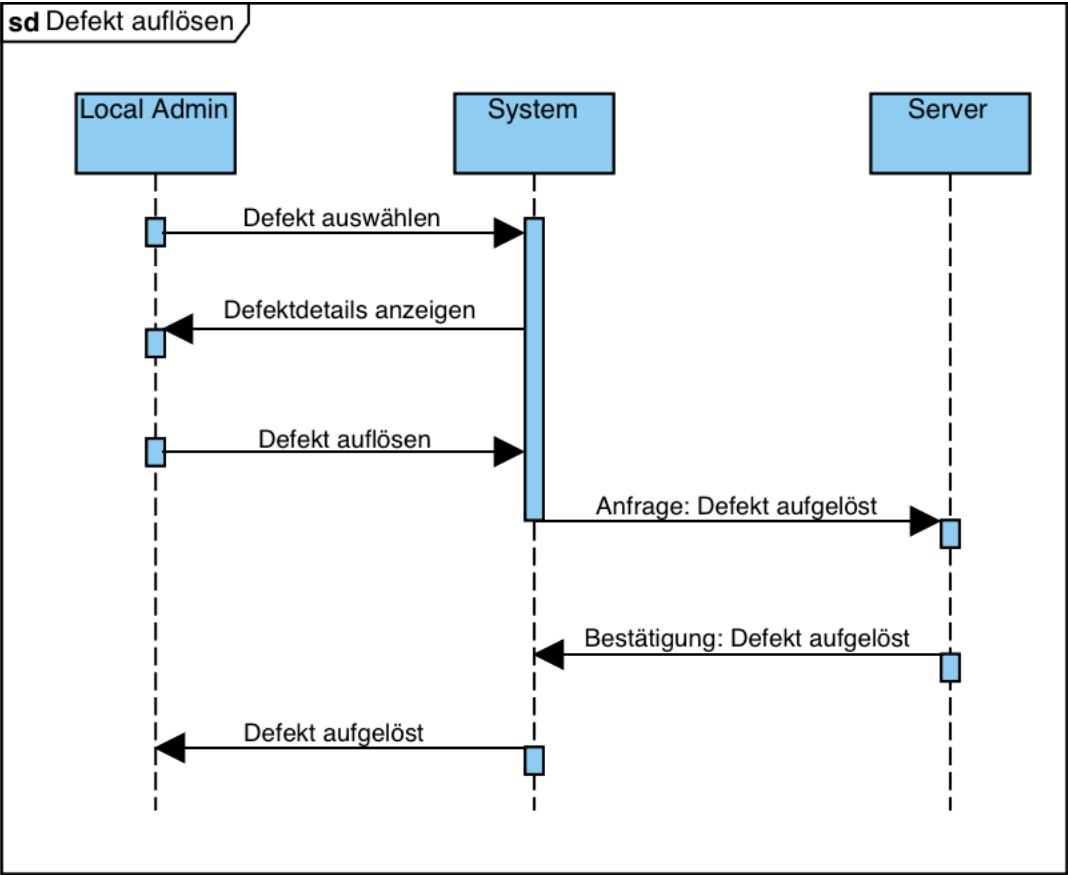
1.2 Mockup

1.3 Screenshots

2. Flow of Events

2.1 Grundlegender Ablauf

Der lokale Administrator navigiert zur Liste der gemeldeten Defekte.
Der lokale Administrator wählt den Büroplatz mit dem behobenen Defekt aus.
Die Details des Defekts werden angezeigt.
Der lokale Administrator klickt auf "Defekt auflösen".
Der Büroplatz wird wieder zur Buchung freigegeben.
Der Administrator erhält eine Bestätigung, dass der Defekt erfolgreich aufgelöst wurde.



2.2 Alternative Abläufe

n/a

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

3. Besondere Anforderungen

n/a

4. Vorbedingungen

Der Benutzer ist als lokaler Administrator angemeldet.

Der Defekt wurde in der realen Welt behoben.

Der Büroplatz ist aktuell gesperrt.

5. Nachbedingungen

Der Defekt ist in der Anwendung erfolgreich aufgelöst.

Der Büroplatz ist wieder buchbar.

5.1 Änderungen speichern / Synchronisation mit dem Server

Der Server muss sicherstellen, dass der Büroplatz nach der Auflösung des Defekts wieder als verfügbar markiert und für Buchungen freigegeben wird.

6. Funktionspunkte

3.1.9 Flurpläne erstellen (Leon)

3.2 Usability

n/a

3.3 Reliability

3.3.1 Uptime

Up-time sollte während Geschäftszeiten garantiert sein.

3.3.2 Keine Einarbeitungszeit erforderlich

Unser Ziel ist es, dass ein Benutzer die Website öffnet und alle Funktionen ohne jegliche Erklärung oder Hilfe nutzen kann.

3.3.3 Vertraute UI/UX

Wir möchten eine Website mit ansprechendem Design und Funktionen implementieren. Auf diese Weise kann der Benutzer mit der Website interagieren, ohne sich an neue UX gewöhnen zu müssen.

3.4 Performance

3.4.1 Performance

Die Ladezeiten für die Anwendung und Aktionen sollen so kurz wie möglich sein.

3.4.2 Capacity

Das System soll min. 100 Buchungen pro Sekunde bearbeiten können.

3.5 Supportability

3.5.1 Coding Standards

Wir werden den Code nach gängigen Clean-Code-Standards schreiben. Zum Beispiel werden wir unsere Variablen und Methoden entsprechend ihrer Funktion benennen. Dadurch bleibt der Code für alle leicht lesbar und zukünftige Weiterentwicklungen werden deutlich vereinfacht.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

3.5.2 Naming Conventions

Für das Backend sollen die API Endpoints genau nach Richtlinien benannt werden.

3.5.3 Testing

Wir möchten die Endpoints auf Validität und Funktionalität testen, fällt unter den Prioritäten aber nach hinten.

3.6 Design Constraints

3.6.1 Architecture

Die Software soll über Webarchitektur funktionieren; ein Backend-System für den gehosteten Service und ein Frontend-System für die Benutzeroberfläche.

3.6.2 Tech stack

Backend: C# mit ASP.NET, Datenbank mit sqllite

Frontend: JavaScript mit VueJS und TailwindCSS

Die Interaktion zwischen Front- und Backend läuft über eine REST API.

3.6.3 Development tools

- Visual Studio (Code)
- Git
- Draw.io / Figma

3.7 On-line User Documentation and Help System Requirements

Es soll eine Hilfeseite zur Verfügung stehen, die alle Funktionen erklärt. Allerdings sollen diese selbstverständlich und intuitiv implementiert sein, sodass eine Hilfe nicht notwendig ist.

3.8 Purchased Components

Wir haben bisher keine gekauften Komponenten. Sollten in der Zukunft gekaufte Komponenten hinzukommen, werden wir sie hier auflisten.

3.9 Interfaces

3.9.1 User Interfaces

Die zu implementierenden Benutzeroberflächen für unsere Büroplatzvermietungs-Website sind:

- Hauptseite: Ermöglicht die Auswahl des Gebäudes, in dem der Büroplatz gemietet werden soll.
- Etagenübersicht: Nach der Auswahl des Gebäudes können die einzelnen Etagen mit ihren Grundrissen aufgerufen werden.
- Büroplatz-Details: Durch Klicken auf einen einzelnen Platz öffnet sich ein Popup mit detaillierten Informationen über den Büroplatz.

3.9.2 Hardware Interfaces

n/a

3.9.3 Software Interfaces

n/a

3.9.4 Communications Interfaces

Der Server und die Hardware kommunizieren über das HTTP-Protokoll.

| | |
|-------------------------------------|------------------|
| SWE-Büroplatzverwaltung | Version: 1.1 |
| Software Requirements Specification | Date: 2024-10-18 |
| rup_srs.docx | |

3.10 Licensing Requirements

n/a

3.11 Legal, Copyright, and Other Notices

Wir übernehmen keine Verantwortung für falsche Daten oder Fehler in der Anwendung.

3.12 Applicable Standards

Die Entwicklung wird den gängigen Clean-Code-Standards und Namenskonventionen folgen. Keine DIN/ISO/IEC-Normen.

4. Supporting Information

Teammitglieder:

- Hagen Zagola
- Leon Rother
- Moritz Mürle
- Christoph Niederer
- Jannik Filpe