

## # Anforderungen: Allgemeine Verwaltung

Dieses Dokument beschreibt die detaillierten Anforderungen für den Bereich "E. Allgemeine Verwaltung" des Autovermietungssystems. Es basiert auf dem Lastenheft "DV-Projekt Winfo - Anwendungssystem Autovermietung SS 2025".

### ## E1: Geschäftsanwendungsfall: Mietstationen verwalten

#### ### Beschreibung

In der Unternehmenszentrale werden die Mietstationen verwaltet. [cite: 247] Dies umfasst die Pflege von Stammdaten, die Überwachung und Optimierung des Fahrzeugbestands sowie die Durchführung von Fahrzeugüberführungen.

#### #### Funktionale Anforderungen

##### \* \*\*E1.1 Stammdatenpflege:\*\*

- \* Das System soll es ermöglichen, Stammdaten für jede Mietstation zu erfassen und zu pflegen. [cite: 247]

- \* Zu den Stammdaten gehören:

- \* Name der Mietstation

- \* Adresse

- \* Telefonnummer

- \* E-Mail-Adresse

- \* Kapazität (Anzahl Stellplätze für PKW)

- \* Informationen zur Fahrzeugaufbereitung (ja/nein, Art der Aufbereitung)

- \* Informationen zur Schadensregulierung (ja/nein)

- \* Das System soll eine Benutzeroberfläche bieten, um neue Mietstationen anzulegen, bestehende zu bearbeiten und zu löschen.

- \* Das System soll Datenvalidierungen durchführen, um sicherzustellen, dass alle erforderlichen Stammdaten vorhanden und im richtigen Format sind.

##### \* \*\*E1.2 Fahrzeugbestand überwachen und optimieren:\*\*

- \* Das System soll den aktuellen Fahrzeugbestand jeder Mietstation erfassen und anzeigen. [cite: 247]

- \* Der Fahrzeugbestand umfasst:

- \* Anzahl der verfügbaren Fahrzeuge (nach Fahrzeugtyp)

- \* Anzahl der vermieteten Fahrzeuge (nach Fahrzeugtyp)

- \* Anzahl der Fahrzeuge in der Aufbereitung/Reparatur

- \* Das System soll Warnmeldungen ausgeben, wenn der Fahrzeugbestand einer Mietstation einen bestimmten Schwellenwert unter- oder überschreitet.

- \* Das System soll Vorschläge zur Optimierung des Fahrzeugbestands geben (z. B. Überführung von Fahrzeugen von einer Station zu einer anderen).

##### \* \*\*E1.3 Fahrzeugüberführung durchführen:\*\*

- \* Das System soll die Überführung von Fahrzeugen zwischen Mietstationen unterstützen. [cite: 248]
- \* Dies umfasst:
  - \* Erfassung der Überführungsaufträge (von welcher Station zu welcher Station, Anzahl und Typ der Fahrzeuge)
  - \* Planung der Überführung (Termin, Fahrer)
  - \* Dokumentation der Überführung (Status, ggf. Abweichungen)
- \* Das System soll die Auswirkungen der Überführung auf den Fahrzeugbestand der beteiligten Stationen aktualisieren.

### ### Nicht-funktionale Anforderungen

- \* \*\*E1.N1 Performance:\*\*
  - \* Das System soll Anfragen zur Stammdatenpflege, Bestandsabfrage und Überführungsverwaltung innerhalb von X Sekunden beantworten (z.B. 2 Sekunden).
- \* \*\*E1.N2 Sicherheit:\*\*
  - \* Der Zugriff auf die Funktionen zur Mietstationsverwaltung soll auf autorisierte Benutzer beschränkt sein (z.B. Mitarbeiter der Unternehmenszentrale).
  - \* Das System soll die Datenintegrität gewährleisten (z.B. durch Transaktionsmanagement).
- \* \*\*E1.N3 Benutzerfreundlichkeit:\*\*
  - \* Die Benutzeroberfläche soll intuitiv und einfach zu bedienen sein.
  - \* Fehlermeldungen sollen klar und verständlich sein.
- \* \*\*E1.N4 Skalierbarkeit:\*\*
  - \* Das System soll in der Lage sein, eine wachsende Anzahl von Mietstationen und Fahrzeugen zu verwalten.

### ## E2: Geschäftsanwendungsfall: Performance-Bericht erstellen

#### #### Beschreibung

In der Unternehmenszentrale wird ein Performance-Bericht erstellt, um die Leistung der einzelnen Mietstationen zu analysieren. [cite: 249]

#### #### Funktionale Anforderungen

- \* \*\*E2.1 Leistungskennziffern definieren:\*\*
  - \* Das System soll es ermöglichen, verschiedene Leistungskennziffern (KPIs) zu definieren. [cite: 250]
  - \* Beispiele für KPIs sind:
    - \* Auslastung der Mietstation (nach Fahrzeugtyp)
    - \* Umsatz pro Mietstation
    - \* Durchschnittliche Mietdauer
    - \* Anzahl der Schäden pro Mietstation
  - \* Das System soll es ermöglichen, KPIs zu gruppieren und zu kategorisieren.

\* \*\*E2.2 Leistungskennziffern der einzelnen Mietstationen auswerten:\*\*

- \* Das System soll die definierten KPIs für jede Mietstation berechnen und anzeigen. [cite: 251, 252]

- \* Die Daten für die Berechnung der KPIs sollen aus dem laufenden Buchungssystem exportiert und in das Reporting-System importiert werden. [cite: 251]

- \* Das System soll es ermöglichen, die KPIs für verschiedene Zeiträume (z.B. täglich, wöchentlich, monatlich, jährlich) auszuwerten.

\* \*\*E2.3 Performance-Dashboard erstellen:\*\*

- \* Das System soll ein Dashboard zur Visualisierung der Unternehmensleistung bereitstellen. [cite: 253]

- \* Das Dashboard soll die wichtigsten KPIs in übersichtlicher Form darstellen (z.B. in Form von Diagrammen, Tabellen, Grafiken).

- \* Das Dashboard soll es ermöglichen, Drill-Down-Analysen durchzuführen, um detailliertere Informationen zu erhalten.

- \* Das Dashboard soll exportierbar sein (z.B. als PDF oder Excel).

#### ### Nicht-funktionale Anforderungen

\* \*\*E2.N1 Performance:\*\*

- \* Die Generierung von Performance-Berichten soll innerhalb eines angemessenen Zeitraums erfolgen (z.B. X Minuten).

\* \*\*E2.N2 Sicherheit:\*\*

- \* Der Zugriff auf die Performance-Berichte soll auf autorisierte Benutzer beschränkt sein.

\* \*\*E2.N3 Datenqualität:\*\*

- \* Das System soll sicherstellen, dass die Daten für die Berechnung der KPIs korrekt und vollständig sind.

\* \*\*E2.N4 Flexibilität:\*\*

- \* Das System soll es ermöglichen, neue KPIs bei Bedarf hinzuzufügen.

#### ## E3: Geschäftsanwendungsfall: Personaleinsatz planen

##### ### Beschreibung

Die HR-Abteilung des Unternehmens plant den Personaleinsatz zentral für alle Mietstationen. [cite: 253]

##### ### Funktionale Anforderungen

\* \*\*E3.1 Notwendige Stammdaten und Auslastungsdaten bereitstellen:\*\*

- \* Das System soll die notwendigen Stammdaten (Personal, Mietstation) und Auslastungsdaten bereitstellen. [cite: 253]

- \* Stammdaten Personal:

- \* Mitarbeitername

- \* Mitarbeiter-ID

- \* Rolle (z.B. Stationsleiter, Sachbearbeiter, Fahrzeugaufbereiter)
- \* Verfügbarkeit
- \* Qualifikationen
- \* Stammdaten Mietstation: (siehe E1.1)
- \* Auslastungsdaten:
  - \* Anzahl der erwarteten Ankünfte/Abholungen
  - \* Anzahl der erwarteten Rückgaben
  - \* Anzahl der Reservierungen
- \* \*\*E3.2 Personaleinsatz planen:\*\*
  - \* Das System soll die Planung des Personaleinsatzes auf Basis der Auslastungsdaten ermöglichen. [cite: 254, 255]
  - \* Das System soll Vorschläge für die Personalplanung machen (z.B. Anzahl der benötigten Mitarbeiter pro Rolle pro Station).
  - \* Das System soll es ermöglichen, den Personaleinsatz manuell anzupassen.
  - \* Das System soll verschiedene Planungszeiträume unterstützen (z.B. täglich, wöchentlich, monatlich).
- \* \*\*E3.3 Personaleinsatz überwachen:\*\*
  - \* Das System soll die Überwachung des Personaleinsatzes ermöglichen. [cite: 256]
  - \* Das System soll es ermöglichen, den geplanten Personaleinsatz mit dem tatsächlichen Personaleinsatz zu vergleichen.
  - \* Das System soll Warnmeldungen ausgeben, wenn es Abweichungen gibt (z.B. Unterbesetzung).
  - \* Das System soll die kurzfristige Anpassung des Personaleinsatzes bei Bedarf ermöglichen (z.B. bei Krankheitsfällen).

### ### Nicht-funktionale Anforderungen

- \* \*\*E3.N1 Performance:\*\*
  - \* Die Planung des Personaleinsatzes soll innerhalb eines angemessenen Zeitraums erfolgen.
- \* \*\*E3.N2 Sicherheit:\*\*
  - \* Der Zugriff auf die Funktionen zur Personaleinsatzplanung soll auf autorisierte Benutzer beschränkt sein (z.B. Mitarbeiter der HR-Abteilung).
- \* \*\*E3.N3 Integration:\*\*
  - \* Das System soll mit anderen Systemen integriert werden können (z.B. mit dem Buchungssystem, dem Personalverwaltungssystem).
- \* \*\*E3.N4 Flexibilität:\*\*
  - \* Das System soll verschiedene Arbeitszeitmodelle und Schichtplanungen unterstützen.

---

### \*\*Hinweise:\*\*

- \* Diese `anforderungen.md`-Datei ist eine detaillierte Ausarbeitung der im Lastenheft genannten Geschäftsanwendungsfälle.

- \* Sie enthält sowohl funktionale (was das System tun soll) als auch nicht-funktionale (wie das System sein soll) Anforderungen.
- \* Die nicht-funktionalen Anforderungen sind wichtig, um die Qualität des Systems sicherzustellen.
- \* Diese Datei sollte im Laufe des Projekts weiter verfeinert und aktualisiert werden.
- \* Die Anforderungen sind so formuliert, dass sie testbar sind.
- \* Die Nummerierung der Anforderungen (z.B. E1.1, E2.N3) dient der besseren Referenzierung.

## # Bereitstellungsanleitung: Allgemeine Verwaltung

Dieses Dokument beschreibt die Schritte zur Bereitstellung des Moduls "Allgemeine Verwaltung" für das Autovermietungssystem.

### ## 1. Voraussetzungen

Stellen Sie sicher, dass die folgenden Komponenten installiert und konfiguriert sind:

- \* **Webserver:** XAMPP (mit Apache) [cite: 8, 9, 10]
- \* **Datenbank:** MySQL [cite: 8, 9, 10]
- \* **PHP:** Version 7.4.29 oder höher (falls PHP verwendet wird) [cite: 30, 31]
- \* **Entwicklungsumgebung:** Visual Studio (oder eine andere bevorzugte IDE) [cite: 8, 9, 10]
- \* **Git:** Zur Versionskontrolle und zum Klonen des Repositorys
- \* **Betriebssystem:** Windows (oder ein anderes geeignetes System für den Server)

### ## 2. Repository klonen

1. Öffnen Sie die Eingabeaufforderung oder ein Terminal.
2. Navigieren Sie zu dem Verzeichnis, in dem Sie das Repository klonen möchten (z. B. `xampp/htdocs` für XAMPP).
3. Führen Sie den folgenden Befehl aus, um das Repository zu klonen:

```
``bash
git clone <URL_DES_REPOSITORYS>
``
```

Ersetzen Sie ``<URL_DES_REPOSITORYS>`` durch die tatsächliche URL des GitHub-Repositorys für das Modul "Allgemeine Verwaltung".

### ## 3. Datenbank einrichten

1. Starten Sie den MySQL-Dienst über XAMPP Control Panel oder einen anderen Dienstmanager.

2. Öffnen Sie ein MySQL-Client-Tool (z. B. phpMyAdmin).
3. Erstellen Sie eine neue Datenbank (z. B. `autovermietung`).
4. Führen Sie das SQL-Skript zum Erstellen der Datenbankstruktur aus. Das Skript befindet sich im Verzeichnis `daten/` des Repositories (z. B. `daten/initialdaten.sql`).
5. Stellen Sie sicher, dass die Datenbankverbindungseinstellungen in der Konfigurationsdatei korrekt sind (z. B. `config/db\_config.php` oder ähnliches).

#### ## 4. Webserver konfigurieren

1. Stellen Sie sicher, dass der Apache-Webserver gestartet ist (über XAMPP oder einen anderen Dienstmanager).
2. Wenn sich der Code nicht im Root-Verzeichnis des Webserver befindet, konfigurieren Sie einen Virtual Host oder passen Sie die Document Root entsprechend an.

#### ## 5. Aufgabenplanung (falls zutreffend)

Falls Aufgabenplanung für bestimmte Funktionen erforderlich ist (z.B. für das Erstellen von Performance-Berichten oder andere Hintergrundprozesse):

1. Öffnen Sie die Aufgabenplanung des Betriebssystems (z. B. "Aufgabenplanung" unter Windows)[cite: 30, 31].
2. Importieren Sie die bereitgestellten Aufgabenplanungsdateien (falls vorhanden). Die Dateien könnten im Verzeichnis `config/` oder einem anderen geeigneten Ordner liegen.
3. Stellen Sie sicher, dass die Pfade und Einstellungen in den Aufgaben richtig konfiguriert sind.

#### ## 6. Zugriff auf die Anwendung

Öffnen Sie einen Webbrowser und navigieren Sie zur URL der Anwendung, um auf das Modul "Allgemeine Verwaltung" zuzugreifen. Zum Beispiel:

#### # Design Dokumentation: Allgemeine Verwaltung

Dieses Dokument beschreibt die Designentscheidungen für den Bereich "E. Allgemeine Verwaltung" des Autovermietungssystems.

#### ## 1. Architektur

##### ### 1.1 Architekturüberblick

Das Modul "Allgemeine Verwaltung" wird als Teil einer mehrschichtigen Architektur implementiert. Die Architektur umfasst die folgenden Schichten:

- \* **Präsentationsschicht:** Zuständig für die Darstellung der Benutzeroberfläche und die Interaktion mit dem Benutzer. (z.B. HTML, CSS, JavaScript)
- \* **Anwendungsschicht:** Enthält die Geschäftslogik des Moduls. (z.B. PHP, Java, Python)
- \* **Datenhaltungsschicht:** Zuständig für den Zugriff auf die Datenbank. (z.B. MySQL)

### ### 1.2 Komponenten

Die Hauptkomponenten des Moduls "Allgemeine Verwaltung" sind:

- \* **Mietstationsverwaltung:**
  - \* Ermöglicht die Verwaltung der Stammdaten von Mietstationen.
  - \* Stellt Funktionen zum Erstellen, Anzeigen, Bearbeiten und Löschen von Mietstationen bereit.
- \* **Performance-Berichtserstellung:**
  - \* Generiert Performance-Berichte für die Unternehmenszentrale.
  - \* Berechnet und visualisiert Leistungskennzahlen.
- \* **Personaleinsatzplanung:**
  - \* Unterstützt die Planung des Personaleinsatzes in den Mietstationen.
  - \* Verwaltet Personalstammdaten und plant den Einsatz basierend auf der Auslastung.
- \* **Gemeinsame Dienste:**
  - \* Stellt gemeinsame Funktionen für die anderen Komponenten bereit (z.B. Datenbankzugriff, Fehlerbehandlung).

### ### 1.3 Diagramme

- \* Ein **Komponentendiagramm** (UML) wird die Beziehungen und Abhängigkeiten zwischen den Hauptkomponenten des Moduls "Allgemeine Verwaltung" darstellen.
- \* Ein **Deploymentdiagramm** (UML) wird die physische Verteilung der Komponenten auf die Infrastruktur zeigen (z.B. Webserver, Datenbankserver).

## ## 2. Datenmodellierung

### ### 2.1 Entity-Relationship-Diagramm (ERD)

Ein ERD wird die wichtigsten Entitäten und ihre Beziehungen im Modul "Allgemeine Verwaltung" darstellen.

- \* **Mietstation:**
  - \* `StationsID` (PK)
  - \* `Name`
  - \* `Adresse`
  - \* `Telefonnummer`
  - \* `Kapazität`
  - \* ...

```

* **PerformanceBericht:**
  * `BerichtID` (PK)
  * `Datum`
  * `MietstationID` (FK)
  * `Umsatz`
  * `Auslastung`
  * ...
* **PersonalEinsatzPlan:**
  * `PlanID` (PK)
  * `Datum`
  * `MietstationID` (FK)
  * `MitarbeiterID` (FK)
  * `Rolle`
  * `Schicht`
  * ...
* **Mitarbeiter:**
  * `MitarbeiterID` (PK)
  * `Name`
  * `Rolle`
  * `...`

```

## ### 2.2 Datenbankschema

Das Datenbankschema wird die Tabellen und Beziehungen in der MySQL-Datenbank definieren. Es wird auf dem ERD basieren.

## ## 3. Benutzeroberfläche (UI) Design

### ### 3.1 Allgemeine Richtlinien

- \* Die Benutzeroberfläche soll benutzerfreundlich, intuitiv und konsistent sein.
- \* Es sollen die im Lastenheft genannten Mock-up-Tools verwendet werden (z.B. Claritee.io), um die UI zu entwerfen.
- \* Die UI soll responsive sein und auf verschiedenen Geräten (Desktop, Tablet, Smartphone) funktionieren.
- \* Es sollen die GUI-Design-Standards beachtet werden (siehe Lastenheft). [cite: 131]

### ### 3.2 Mockups

- \* Für jeden Geschäftsanwendungsfall (Mietstationen verwalten, Performance-Bericht erstellen, Personaleinsatz planen) werden Mockups der wichtigsten Bildschirme erstellt.
- \* Die Mockups zeigen das Layout, die Navigation und die Interaktionselemente der UI.
- \* Die Mockups werden im Verzeichnis `doku/mockups/` gespeichert.



### ### 3.3 Navigationsdesign

- \* Die Navigation innerhalb des Moduls "Allgemeine Verwaltung" soll einfach und klar sein.
- \* Es wird ein Navigationsmenü verwendet, um zwischen den Hauptfunktionen zu wechseln.
- \* Breadcrumbs helfen dem Benutzer, seinen aktuellen Standort innerhalb der Anwendung zu bestimmen.

### ### 3.4 Formulare

- \* Formulare zur Dateneingabe sollen benutzerfreundlich gestaltet sein.
- \* Es werden geeignete Eingabefelder (z.B. Textfelder, Dropdown-Menüs, Datumsfelder) verwendet.
- \* Es werden Validierungen durchgeführt, um sicherzustellen, dass die eingegebenen Daten korrekt sind.
- \* Fehlermeldungen sollen klar und verständlich sein.

## ## 4. Schnittstellen

### ### 4.1 Schnittstellen zu anderen Modulen

Das Modul "Allgemeine Verwaltung" interagiert mit den folgenden Modulen:

- \* **Buchungssystem:** Um Daten für die Performance-Berichte und die Personaleinsatzplanung abzurufen.
- \* **Personalverwaltungssystem:** Um Mitarbeiterdaten abzurufen und zu aktualisieren.

### ### 4.2 Schnittstellen zu externen Systemen

Es sind keine direkten Schnittstellen zu externen Systemen geplant.

## ## 5. Technologien

Die folgenden Technologien werden für die Implementierung des Moduls "Allgemeine Verwaltung" verwendet:

- \* **Frontend:** HTML5, CSS, JavaScript
- \* **Backend:** PHP (oder eine andere im Lastenheft erlaubte Sprache)
- \* **Datenbank:** MySQL

## ## 6. Designentscheidungen

- \* Die Entscheidung für eine mehrschichtige Architektur ermöglicht eine klare Trennung der Zuständigkeiten und eine bessere Wartbarkeit.

\* Die Verwendung von MySQL als Datenbank entspricht den Empfehlungen des Lastenhefts.  
[cite: 9]

\* Die UI-Gestaltung orientiert sich an den GUI-Design-Standards, um eine benutzerfreundliche Anwendung zu gewährleisten. [cite: 131]

---

**\*\*Hinweise:\*\***

\* Diese `design.md`-Datei gibt einen Überblick über die wichtigsten Designaspekte des Moduls "Allgemeine Verwaltung".

\* Sie wird im Laufe des Projekts weiter verfeinert und mit detaillierteren Designinformationen ergänzt.

\* Diagramme (z.B. Komponentendiagramm, ERD) und Mockups werden separat erstellt und in die Dokumentation eingebunden oder verlinkt.

\* Die Datei soll die Designentscheidungen nachvollziehbar machen und die Kommunikation innerhalb des Entwicklungsteams erleichtern.

## # Planung Dokumentation: Allgemeine Verwaltung

Dieses Dokument beschreibt die Planung für die Entwicklung des Moduls "Allgemeine Verwaltung" für das Autovermietungssystem.

### ## 1. Projektplan

#### ### 1.1 Projektziele

Die Hauptziele des Projekts für das Modul "Allgemeine Verwaltung" sind:

\* Implementierung der im Lastenheft definierten Geschäftsanwendungsfälle:

\* E1: Mietstationen verwalten

\* E2: Performance-Bericht erstellen

\* E3: Personaleinsatz planen

\* Entwicklung einer benutzerfreundlichen und effizienten Anwendung.

\* Einhaltung der im Lastenheft genannten Qualitätsanforderungen (z.B. Performance, Sicherheit, Benutzerfreundlichkeit).

\* Lieferung einer gut dokumentierten und testbaren Anwendung.

#### ### 1.2 Zeitplan

Der Gesamtzeitplan für das Projekt ist in Sprints unterteilt. Jeder Sprint hat eine Dauer von [z.B. 2 Wochen].

#### \* \*\*Sprint 1:\*\*

- \* Analyse der detaillierten Anforderungen
- \* Design der Datenbank und der grundlegenden Architektur
- \* Implementierung der Stammdatenpflege für Mietstationen (E1.1)
- \* Einrichtung der Entwicklungsumgebung

#### \* \*\*Sprint 2:\*\*

- \* Implementierung der Fahrzeugbestandsüberwachung (E1.2)
- \* Implementierung der Fahrzeugüberführung (E1.3)
- \* Erstellung von Mockups für die Performance-Berichte
- \* Beginn der Tests für E1

#### \* \*\*Sprint 3:\*\*

- \* Implementierung der Leistungskennzifferndefinition (E2.1)
- \* Implementierung der Auswertung der Leistungskennziffern (E2.2)
- \* Erstellung des Performance-Dashboards (E2.3)
- \* Beginn der Tests für E2

#### \* \*\*Sprint 4:\*\*

- \* Implementierung der Stammdatenbereitstellung für Personaleinsatzplanung (E3.1)
- \* Implementierung der Personaleinsatzplanung (E3.2)
- \* Implementierung der Personaleinsatzüberwachung (E3.3)
- \* Beginn der Tests für E3

#### \* \*\*Sprint 5:\*\*

- \* Integrationstests
- \* Fehlerbehebung
- \* Dokumentation
- \* Vorbereitung der Präsentation

### ### 1.3 Meilensteine

\* \*\*Meilenstein 1:\*\* Abschluss der Anforderungsanalyse und des Designs (Ende Sprint 1)

\* \*\*Meilenstein 2:\*\* Fertigstellung der Implementierung von E1 (Ende Sprint 2)

\* \*\*Meilenstein 3:\*\* Fertigstellung der Implementierung von E2 (Ende Sprint 3)

\* \*\*Meilenstein 4:\*\* Fertigstellung der Implementierung von E3 (Ende Sprint 4)

\* \*\*Meilenstein 5:\*\* Abschluss der Tests und der Dokumentation (Ende Sprint 5)

### ### 1.4 Ressourcenplanung

#### \* \*\*Entwicklungsteam:\*\*

- \* [Namen der Teammitglieder]
- \* Rollen:
  - \* Projektleiter
  - \* Entwickler (Frontend, Backend)
  - \* Tester
  - \* Dokumentation

#### \* \*\*Hardware:\*\*

- \* Entwicklungsrechner
- \* Testserver

#### \* \*\*Software:\*\*

- \* Entwicklungsumgebung (z.B. Visual Studio)
- \* Versionskontrollsystem (Git)
- \* Projektmanagement-Tool (z.B. Jira, Trello)
- \* Kommunikationstools (z.B. Slack, Microsoft Teams)

## ## 2. Sprintplanung

### ### 2.1 Sprint-Backlog

Für jeden Sprint wird ein Sprint-Backlog erstellt, der die Aufgaben enthält, die in diesem Sprint bearbeitet werden sollen.

- \* Die Aufgaben werden aus dem Produkt-Backlog abgeleitet (d.h. den Anforderungen in der `anforderungen.md`).
- \* Jede Aufgabe wird geschätzt (z.B. in Stunden oder Story Points).
- \* Die Aufgaben werden dem Sprint-Backlog basierend auf der Priorität und der verfügbaren Kapazität des Teams hinzugefügt.

### ### 2.2 Sprint-Ziele

Jeder Sprint hat ein klar definiertes Ziel, das beschreibt, was am Ende des Sprints erreicht werden soll.

- \* Die Sprint-Ziele sind messbar und erreichbar.
- \* Sie helfen dem Team, sich auf die wichtigsten Aufgaben zu konzentrieren.

### ### 2.3 Sprint-Review und Retrospektive

- \* Am Ende jedes Sprints findet ein Sprint-Review statt, in dem das Team die abgeschlossenen Aufgaben präsentiert und Feedback von den Stakeholdern einholt.
- \* Nach dem Sprint-Review findet eine Sprint-Retrospektive statt, in der das Team den Sprintprozess reflektiert und Verbesserungsmöglichkeiten identifiziert.

## ## 3. Aufgabenverteilung

- \* Die Aufgaben werden den Teammitgliedern basierend auf ihren Fähigkeiten und ihrer Verfügbarkeit zugewiesen.
- \* Die Aufgabenverteilung wird im Projektmanagement-Tool dokumentiert.
- \* Der Projektleiter ist für die Koordination der Aufgaben und die Überwachung des Fortschritts verantwortlich.

## ## 4. Kommunikationsplan

### \* \*\*Regelmäßige Meetings:\*\*

- \* Tägliche Stand-up-Meetings (15 Minuten)
- \* Sprint-Planungsmeetings (zu Beginn jedes Sprints)
- \* Sprint-Review-Meetings (am Ende jedes Sprints)
- \* Sprint-Retrospektiven (am Ende jedes Sprints)

### \* \*\*Kommunikationskanäle:\*\*

- \* Projektmanagement-Tool (für Aufgabenverwaltung und Fortschrittsverfolgung)
- \* Kommunikationstool (WhatsApp und die Universität)

### \* \*\*Berichterstattung:\*\*

- \* Der Projektleiter erstellt regelmäßig Fortschrittsberichte für die Stakeholder.

## ## 5. Risikomanagement

- \* Mögliche Risiken werden identifiziert und bewertet.
- \* Für jedes Risiko werden Maßnahmen zur Risikominderung oder -vermeidung definiert.
- \* Der Fortschritt der Risikomanagementmaßnahmen wird regelmäßig überwacht.

---

Datum 12.06.2025

Wichtigste Info: abschlussdokumentation über ein Gantt-Diagramm erstellen!

Passendes template: <https://www.vertex42.com/ExcelTemplates/excel-gantt-chart.html>

Jeweils für excel und Google Sheets verfügbar

Hier ist der Projektauftrag!

[Projektauftrag](#)