**ZHAW School of Engineering**

**Bachelor of Science in Informatik**

**Software-Projekt PM3 HS24**

**Projektskizze // Technischer Bericht xx**

**Team 0x**

**Titel des Produkts**

Abgabedatum

Namen der Team-Mitglieder in alphabetischer Reihenfolge:

Inhaltsverzeichnis

[1. Projektidee 1](#_Toc176794766)

[1.1. Ausgangslage 1](#_Toc176794767)

[1.2. Idee 1](#_Toc176794768)

[1.3. Kundennutzen 1](#_Toc176794769)

[1.4. Hauptablauf 1](#_Toc176794770)

[2. Analyse 1](#_Toc176794771)

[2.1. Use-Case-Modell 1](#_Toc176794772)

[2.1.1. Domäne User 1](#_Toc176794773)

[2.2. Wichtigste Use-Cases „Fully-Dressed” 1](#_Toc176794774)

[2.3. Casual Use-Cases 1](#_Toc176794775)

[2.4. Standardszenario 2](#_Toc176794776)

[2.5. Zusätzliche Anforderungen 2](#_Toc176794777)

[2.5.1. Nicht funktionale Anforderungen 2](#_Toc176794778)

[2.6. Domänenmodell 2](#_Toc176794779)

[3. Design 2](#_Toc176794780)

[3.1. Softwarearchitektur 2](#_Toc176794781)

[3.2. Design-Klassendiagramme 2](#_Toc176794782)

[4. Implementation 2](#_Toc176794783)

[4.1. Lieferergebnisse 2](#_Toc176794784)

[4.2. Testkonzept 2](#_Toc176794785)

[5. Resultate 2](#_Toc176794786)

[5.1. Erreichte Ziele 2](#_Toc176794787)

[5.2. Offene Punkte 2](#_Toc176794788)

[5.3. Ausblick auf mögliche Weiterentwicklungen 2](#_Toc176794789)

[6. Quellenverzeichnis 4](#_Toc176794790)

[7. Abbildungs- und Tabellenverzeichnisse 4](#_Toc176794791)

[7.1. Abbildungsverzeichnis 4](#_Toc176794792)

[7.2. Tabellenverzeichnis 4](#_Toc176794793)

[8. Glossar 5](#_Toc176794794)

# Projektidee

## Ausgangslage

…….

Um diese Probleme anzugehen, wird im Rahmen einer Semesterarbeit die Software xxx entwickelt, die alle notwendigen Funktionalitäten integriert …..

## Idee

…. …..

## Kundennutzen

…..

## Hauptablauf

….

# Analyse

## Use-Case-Modell

……

### Domäne User

Abbildung 1 zeigt das Use-Case-Modell der Domäne User, …….

…..

Abbildung 1: Use-Case-Modell für die Domäne User

## Wichtigste Use-Cases „Fully-Dressed”

## Casual Use-Cases

|  |  |
| --- | --- |
| Domäne: User | |
| Use Case: | **Beschreibung:** |

## Standardszenario

## Zusätzliche Anforderungen

### Nicht funktionale Anforderungen

…..

## Domänenmodell

……

Abbildung 5: Domänenmodell

# Design

## Softwarearchitektur

## Design-Klassendiagramme

……..

# Implementation

## Lieferergebnisse

….

## Testkonzept

…..

# Resultate

## Erreichte Ziele

….

## Offene Punkte

….

## Ausblick auf mögliche Weiterentwicklungen

….

# Quellenverzeichnis

[1]

# Abbildungs- und Tabellenverzeichnisse

## Abbildungsverzeichnis

Abbildung 1: Use-Case-Modell für die Domäne User 2

Abbildung 7: UI-Sketch Projekt-Administration 2

Abbildung 8: UI-Sketch Personenvorschläge 2

Abbildung 9: UI-Sketch Tasks 2

Abbildung 5: Domänenmodell 2

## Tabellenverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**

# Glossar

**A**

* **Anwendungsfall**: Eine Beschreibung einer Interaktion zwischen einem Akteur und dem System, die ein bestimmtes Ziel erreichen soll.
* **API (Application Programming Interface):** Eine API definiert die Schnittstelle zwischen verschiedenen Softwarekomponenten. Sie legt fest, wie Programme oder Komponente miteinander interagieren können, indem sie bestimmte Funktionen und Daten zugänglich macht.

**D**

* **DAO (Data Access Object):** DAO ist ein Designmuster in der Softwareentwicklung, das die Trennung von Datenbankzugriff und Geschäftslogik fördert. DAOs stellen Methoden bereit, um auf Datenbanken zuzugreifen und Operationen wie das Lesen, Schreiben, Aktualisieren und Löschen von Daten durchzuführen.
* **Domänenmodell**: Eine abstrakte Darstellung der relevanten Konzepte, Eigenschaften und Beziehungen innerhalb eines bestimmten Problemgebiets.
* **DTO (Data Transfer Object):** DTOs sind Objekte, die verwendet werden, um Daten zwischen verschiedenen Teilen einer Anwendung zu übertragen. Sie enthalten typischerweise nur die benötigten Datenfelder und dienen dazu, die Kommunikation effizienter zu gestalten, indem sie die Datenübertragung optimieren.

**G**

* **Gruppe**: Als Gruppe wird eine Sammlung von Benutzern verstanden, welche im Rahmen eines Projektes zusammenarbeiten.

* **Gruppenbildung**: Die Gruppenbildung entspricht das Verfahren, mit welcher die Projektleitung eine Gruppe für einen Projekt zusammenstellt.

* **Gruppenvorschlag**: Ein Gruppenverschlag bezeichnet eine Sammlung von Benutzern, welche die gestellten Anforderungen für die Gruppenbildung erfüllt (benötigte Skills für das Projekt).

**M**

* **Mitglieder**: Mitglieder sind Benutzer von *TeamForge*, welche einer Gruppe angehören.

**P**

* **Projekt**: Ein *TeamForge*-Projekt gehört zu einem konkreten Projekt (z.B. beruflich, akademisch). Innerhalb eines *TeamForge*-Projektes werden bestimmte Aspekte eines realen Projektes verwaltet, wie z.B. Terminplanung oder Gruppenbildung.

* **Projektleitung**: Die verantwortliche Person für die Gruppenbildung, Taskverwaltung und Terminfindung innerhalb eines Projektes.

**R**

* **React**: React ist eine JavaScript-Bibliothek zur Entwicklung von Benutzeroberflächen (UIs). Sie ermöglicht die Erstellung von interaktiven und dynamischen Webanwendungen durch die Verwendung von wiederverwendbaren Komponenten.

* **REST (Representational State Transfer**): REST ist ein Architekturstil für die Entwicklung von verteilten Systemen, insbesondere für Webanwendungen. Er basiert auf dem Prinzip, dass Ressourcen über standardisierte HTTP-Methoden (wie GET, POST, PUT, DELETE) manipuliert werden.

**S**

* **Skill**: Ein Skill (Fähigkeit) bezieht sich auf das Wissen, die Erfahrung oder die Fertigkeiten, die ein (Team) Mitglied besitzt, bezogen auf eine bestimmte Technologie bzw. ein bestimmtes Arbeitsumfeld.
* **Springboot:** Spring Boot ist ein Framework für die Entwicklung von Java-Anwendungen. Es bietet eine schnelle und einfache Möglichkeit, Produktionsbereitschaft (Production-Readiness) in Java-Anwendungen zu erreichen, indem es Konventionen über Konfiguration stellt und eine Vielzahl von Funktionen für die Entwicklung bereitstellt.
* **SSD (System-Sequenzdiagramm)**: UML-Diagramm, das den Ablauf einer Interaktion zwischen einem externen Akteur und einem System darstellt. Es fokussiert sich auf die Reihenfolge der Nachrichten oder Aktionen zwischen dem Akteur und dem System, ohne detaillierte interne Prozesse des Systems zu zeigen.

**T**

* **Task**: Ein Task (Aufgabe) ist eine spezifische Einheit von Arbeit oder Aktivität, die innerhalb eines Projekts von einem Mitglied ausgeführt werden muss. Die Projektleitung definiert, wer welchen Task übernimmt (Taskverwaltung).

* **Taskverwaltung**: Die Taskverwaltung umfasst die Zuweisung von Tasks an bestimmte Mitglieder innerhalb eines Projektes.

* **Termine**: Termine beziehen sich auf Meetings/Treffen zwischen mehreren Mitgliedern eines Teams.

* **Terminfindung**: Der Prozess für die Festlegung von geeigneten Zeitrahmen für Termine, um sicherzustellen, dass alle Beteiligten verfügbar sind.
* **Typescript**: TypeScript ist eine von Microsoft entwickelte Programmiersprache, die als Superset von JavaScript fungiert. Sie fügt statische Typisierung, Klassen, Schnittstellen und andere Features hinzu, um die Entwicklung von grösseren JavaScript-Anwendungen zu erleichtern und Fehler frühzeitig zu erkennen.

**U**

* **UC (Use-Case)**: siehe Anwendungsfall.