Technische Dokumentation des Projekts „Taskitory“

Programmentwurf für die Vorlesung Software Engineering II

für die Prüfung zum

Bachelor of Science

des Studiengangs Informatik

Studienrichtung Angewandte Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

Fabian Schwickert

16. Mai 2022

Matrikelnummer 4439027

Kurs TINF19 B4

Ausbildungsfirma Siemens AG, Karlsruhe

Gutachter der Studienakademie Mirko Dostmann

Inhaltsverzeichnis

[Inhaltsverzeichnis II](#_Toc93146817)

[Abbildungsverzeichnis III](#_Toc93146818)

[Tabellenverzeichnis IV](#_Toc93146819)

[Abkürzungsverzeichnis V](#_Toc93146820)

[1 Problemstellung, Ziel und Aufbau 1](#_Toc93146821)

[2 Entwurfsmuster 1](#_Toc93146822)

[3 Domain Driven Design 1](#_Toc93146823)

[3.1 Ubiquitous Language 1](#_Toc93146824)

[3.2 Taktische Muster des Domain Driven Design 1](#_Toc93146825)

[4 Clean Architecture 1](#_Toc93146826)

[5 Programming Principles 1](#_Toc93146827)

[6 Testing 1](#_Toc93146828)

[6.1 Unit Testing 1](#_Toc93146829)

[6.2 Mocks 2](#_Toc93146830)

[7 Refactoring 2](#_Toc93146831)

[Literaturverzeichnis XI](#_Toc93146832)

Abbildungsverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Tabellenverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Abkürzungsverzeichnis

AG Aktiengesellschaft

# Problemstellung, Ziel und Aufbau

Es gibt eine Reihe von Aufgabenverwaltungs-Systemen wie z. B. Jira, Youtrack oder Mi­cro­soft Planner, die mit einem Kanban Board arbeiten. Die hier dokumentierte Klau­sur­er­­­satz­leis­tung besteht in der Entwicklung einer App­li­ka­tion, die eine solche Auf­ga­ben­ver­wal­­­tung mo­delliert. Es wird kei­ne graphische Benutzeroberfläche (GUI) entwickelt und le­­diglich die not­wendige An­wen­dungs­logik pro­grammiert und über einen REST-Service (Re­­pre­sentational State Transfer) verfügbar ge­macht. Für die Implementierung des Rest-Ser­vices wird die Pro­gram­miersprache Java mit dem Package Manager „Ma­­ven“ und dem „Spring-Boot-Framework“ verwendet.

Mit der Anwendung soll es möglich sein, die Aufgaben in einem Projekt gemeinsam in ei­nem Projekt-Team zu verwalten. Dazu müssen Benutzer Accounts anlegen, die in den Kon­­­text gemeinsamer Projekte gebracht werden. Das Identity- und Accessmanagement (IAM) soll dabei durch einen Keycloak umgesetzt werden. Ei­ne persistente Da­ten­spei­che­rung wird mit einer Postgresql-Datenbank (DB) um­ge­setzt. In der DB werden die Daten des Keycloaks und ein eigenes DB-Modell gehalten. Die benannten Komponenten (Key­cloak, DB und REST-Service) der App­likation sollen do­cke­risiert werden und mit „Docker-Com­pose“ konfiguriert und be­trie­ben werden.

Um die Bedienung des REST-Services zu vereinfachen, wird eine Sammlung von Anfragen und Konfigurationen für einen HTTP-Client zur Verfügung gestellt. Dazu wird eine „Collection“ für Anwendung Postman zur Verfügung gestellt.

Alle Inhalte der hier dokumentierten Klausurersatzleistung sind auf GitHub unter dem Link „https://github.com/Krayaty/Taskitory“ erreichbar.

# Anforderungsanalyse

Die Anforderungen an die Applikation, die im Rahmen dieser Klausurersatzleistung ent­wi­ckelt wird, sind weniger umfangreich als bei Referenz-Applikationen wie z. B. Youtrack oder Jira. Allerdings orientiert sich die die eigene Applikation an solchen Referenz-App­li­ka­­tionen. Nachfolgend werden allgemeine funktionale Anforderungen definiert und zen­tra­le Use-Cases festgehalten, die implementiert werden sollen. Nicht-Funktionale An­for­de­rungen werden nur dann betrachtet, wenn entsprechende Probleme auftreten.

**Projekte und Benutzer**

Die Projekt-Entitäten sind der Haupt-Bezugspunkt für die anderen Entitäten. Benutzer sind ei­nem oder mehreren Projekten zugeordnet und können über eine Projekt-Entität auf die Res­sourcen eines Projekts („Kanban Boards“, „Aufgaben“ und „Tags“) zugreifen. Dabei kön­nen User die Rollen eines einfachen „Users“ oder eines Projekt-Administrators („Ad­min“) innehaben. Es muss zu jeder Zeit mindestens einen Projekt-Administrator ge­ben. Es gibt keine Obergrenze für die Anzahl von Projekt-Administratoren. Die Rollen be­schrän­­ken den Zu­griff auf die Ressourcen eines Projekts. Einfache User können die At­tri­bu­te einer Projekt-Entität nicht verändern. Zudem sollen einfache User keine be­ste­hen­den User entfernen oder neue hinzufügen können. Einfache User können Datensätzen der an­deren Entitäten hinzufügen, verändern und löschen. Projekt-Administratoren sollen al­le Zu­griffs-Rechte auf Ressourcen und Funktionen eines Projekts haben. Das schließt das Hin­zufügen, Befördern und Entfernen von Benutzern ein.

Über die Zeit sollen Benutzer ein Projekt verlassen und wieder beitreten können. Es ist nicht mög­lich einem Projekt, ohne die Zustimmung des Projekt-Teams beizutreten. Dafür wird ein geheimer Projekt-Schlüs­­sel verwendet, der nur von Projekt-Administratoren einsehbar ist.

1. User anlegen

Benutzer müssen einen Account für die Applikation besitzen und einem Projekt zu­ge­ordnet sein, um die zugehörigen Informationen einsehen zu können. Ein Be­nut­zer-Account soll durch einen Username identifizierbar sein und mit einem Passwort au­thentifiziert werden. Weitere Userdaten sind für diese Applikation nicht relevant. Da­mit ein Benutzer über die REST-API auf die Applikation zugreifen kann, werden zu­nächst Username und Passwort in einer HTTP-Nachricht übermittelt. Später kann mit einem IAM-Server ein Access-Token erzeugt werden.

1. Projekt anlegen

Ein Projekt ist der gemeinsame Kontext für alle anderen Entitäten. Ein Projekt hat da­her auch Informationen über alle beteiligten Objekte dieser Entitäten. Neben all­ge­meinen At­tributen wie z. B. einer Bezeichnung kennt das Projekt sein Projekt-Team, seine Aufgaben, sei­ne Kanban Boards und seine Statistiken. Ein Projekt wird von einem Benutzer mit Be­zeichnung und Projekt-Team angelegt und enthält au­to­ma­tisch eine leere Auf­ga­ben­samm­lung.

**Aufgaben**

Eine zentrale Funktion der Applikation ist die Erstellung und Verwaltung von Aufgaben. Die Aufgaben-En­­tität ist dem entsprechend wichtig. Die Aufgaben werden neben Entitäs-At­­­tri­bu­ten auch mit Meta-Informationen in „Tags“ be­schrie­ben. Aufgaben haben die At­tri­bute „Be­zeich­nung“, „Beschreibung“, „Erstellungs-“ und „Fertigstellungs-Zeit­punkt“, „Kom­­ple­xi­tät“, „Bearbeitungs-Status“ und „zuständige Person“ besitzen. Die Bezeichnung einer Aufgabe ist in einem Projekt eindeutig. Es sollen Aufgaben-Kompositionen und Abhängigkeiten zwischen Aufgaben mög­lich sein. Auf­gaben sind nach der Erzeugung automatisch in dem „Backlog“ eines Pro­jekts ein­sehbar, von wo sie einem „Kanban Board“ hinzugefügt werden können.

**Kanban Board**

Ein Kanban Board be­trifft einen be­stimm­ten Zeitraum („Sprint“). Nach einem Sprint besteht das Kan­ban Board weiter, kann aber nicht mehr bearbeitet werden. Ein altes Kan­ban Board kann in ein neues Kanban Board überführt werden.

Ein Kanban Board kann im Rah­men dieser Klausurersatzleistung aus drei bis zu fünf Spalten bestehen, die den Status ei­ner Auf­gabe be­schreiben. Die Spalten „ToDo“, „In Progress“, und „Done“ sind ob­li­ga­to­risch. Die Spalten „Review“ und „Testing“ können bei Bedarf hinzugefügt und wieder ent­fernt werden. Dabei sollen dort eingeordnete Aufgaben ihren Bearbeitungs-Status nicht ver­lieren. Die Spalte und die darin enthaltenen Aufgaben werden lediglich ausgeblendet. Auf­gaben können einem Kanban Board hin­zu­­gefügt und auf dem Kanban Board in einen an­deren Bearbeitungs-Status ver­scho­­ben wer­­den.

Über einen Sprint können Statistiken an­­ge­legt werden. Dafür sollen ent­spre­chen­de Kanban Boards als Repräsentation eines Sprints genutzt werden. Es soll z. B. aus­ge­wer­tet werden, welcher Anteil der Aufgaben auf ei­nem Kanban Board in einem be­stimm­ten Bearbeitungs-Status ist.

Use Cases

1. Kanban Board anlegen

Für die Verwaltung von Aufgaben, werden bei dieser Applikation Kanban Boards ein­gesetzt. Kanban Boards müssen in einem bestehenden Projekt angelegt werden und für einen bestimmten Zeitraum erzeugt werden. Ein Kanban Board ist bei Er­zeu­gung leer und besitzt die oben benannten Spalten. Es sollen mehrere Kanban Boards gleichzeitig bestehen können. Daher muss ein Kanban Board über eine Be­zeich­nung o. Ä. eindeutig identifizierbar sein.

1. Kanban Board migrieren

Aus einem „abgelaufenen“ Kanban Board soll ein neues Kanban Board erzeugt wer­den. Das ist dann interessant, wenn sich auf dem alten Kanban Board noch nicht ab­­ge­schlos­se­ne Aufgaben befinden. Diese Aufgaben sollen in den gleichen Status des neu­en Kanban Boards über­nommen werden.

1. Aufgaben anlegen

Aufgaben sind das Herzstück dieser Applikation. Eine Aufgabe soll die oben ge­nann­ten Attribute enthalten und im Stan­dard­fall in die Aufgabensammlung eines Pro­jekts eingefügt werden. Es soll aber möglich sein, die Aufgabe direkt einem be­stimm­ten Kanban Board hinzuzufügen.

1. Aufgaben zu einem Kanban Board hinzufügen

Es soll möglich sein, eine Aufgabe nachträglich aus der Aufgabensammlung zu einem Kanban Board hinzuzufügen.

1. Aufgabe auf dem Kanban Board verschieben

Die zentrale Funktion eines Kanban Boards ist, Aufgaben über ihren Bearbeitungs-Sta­­tus zu verwalten. Diese Funktion soll auch mit der Applikation möglich sein. Die Aufgaben auf einem Kanban Board sollen ihren Status auf Anweisung eines Be­nut­zers ändern.

1. Aufgaben bearbeiten

Es soll möglich sein, alle Attribute einer Aufgabe zu bearbeiten.

1. Statistik erstellen

Wenn alle oberen Use Cases implementiert wurden, ist es möglich Statistiken über ein Kanban Board zu erstellen. Mögliche Anwendungsfälle sind z. B. Verteilung von Aufgaben nach Status oder durchschnittliche Bearbeitungsdauer.

Blau 🡪 text überarbeiten

Rot 🡪 noch nicht implementiert

# Entwurfsmuster

a

# Domain Driven Design

## Ubiquitous Language

A

## Taktische Muster des Domain Driven Design

A

# Clean Architecture

A

# Programming Principles

A

# Testing

## Unit Testing

A

## Mocks

A

# Refactoring

A

Literaturverzeichnis

1. a