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](#_Toc97828054)

[Abbildungsverzeichnis III](#_Toc97828055)

[Tabellenverzeichnis IV](#_Toc97828056)

[Abkürzungsverzeichnis V](#_Toc97828057)

[1 Problemstellung, Ziel und Aufbau 1](#_Toc97828058)

[2 Anforderungsanalyse 1](#_Toc97828059)

[3 Domain Driven Design 8](#_Toc97828060)

[3.1 Ubiquitous Language 8](#_Toc97828061)

[3.2 Taktische Muster des Domain Driven Design 9](#_Toc97828062)

[4 Clean Architecture 9](#_Toc97828063)

[5 Entwurfsmuster 9](#_Toc97828064)

[6 Programming Principles 9](#_Toc97828065)

[7 Testing 9](#_Toc97828066)

[7.1 Unit Testing 9](#_Toc97828067)

[7.2 Mocks 9](#_Toc97828068)

[8 Refactoring 9](#_Toc97828069)

[Literaturverzeichnis XI](#_Toc97828070)

Abbildungsverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Tabellenverzeichnis

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

Abkürzungsverzeichnis

DTO Datentransferobjekt

IAM Identity and Access Management

REST Representational State Transfer

# 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.

# Clean Architecture

## Grundlagen

Unter „Clean Architecture“ versteht man einen Architektur-Ansatz für Software-Applikationen. Die Clean Architecture fußt auf der Idee, dass Applikationen in mehrere Schichten aufgeteilt werden können, die eigene Aufgabenbereiche abdecken. Wie bei einer Zwiebel umschließt eine Schicht die andere und nur die äußerste Schicht ist für Benutzer der Applikation zugänglich. Die Funktionen der inneren Schichten sind über definierte Schnittstellen für die jeweils folgende Schicht zugänglich. Die äußeren sind dadurch von den inneren Schichten abhängig; umgekehrt sind die inneren jedoch nicht von den äußeren Schichten abhängig. Daher wird es schwerer den Code auf einer Schicht zu ändern, je weiter innen sie liegt. Die Clean Architecture ermöglicht durch den be­schrie­be­nen modularen Aufbau, dass die für eine Domäne und Applikation spezifische Ge­schäfts­logik nicht geändert werden muss, wenn konkrete technische Implementierungen ei­ner Applikation wie geändert werden müssen. Auf diese Weise kann ein Applikation lang­fristig mit wenig Aufwand gewartet und angepasst werden.

Im Allgemeinen werden die folgenden fünf Schichten der Clean Architecture definiert (von innen nach außen):

1. Abstraktions-Schicht:

Die Abstraction-Schicht ist der Kern einer Applikation. In dieser Schicht wird sog. do­mä­nen­­übergreifendes Wissen implementiert. Das meint Algorithmen und Konzepte, die nicht an die Domäne einer Applikation gebunden sind. Zudem werden hier Software-Werk­­zeuge und -Bibliotheken ein­gebunden, die auf allen folgenden Schichten verwendet wer­­den und technologische Standards umsetzen. Eine Applikation muss keine Ab­strak­tions-Schicht besitzen.

1. Domänen-Schicht:

Die Domänen-Schicht umschließt die Abstraktions-Schicht und hängt daher von der ihr ab. In der Domänen-Schicht wird die Fachlichkeit einer Applikation implementiert. Hier wird die übergreifende Geschäftslogik implementiert, die für die Abbildung einer Domäne in vielen verschiedenen App­li­kation benötigt wird. Die Domänen-Schicht wird durch das Domain Driven Desing (siehe Ka­pitel 3) modelliert.

1. Applikations-Schicht:

Die Applikations-Schicht folgt direkt auf die Domänen-Schicht und hängt daher von ihr ab. In der Applikations-Schicht werden die Use Cases einer Applikation abgebildet. In diesem Teil einer Applikation werden die für den Benutzer erfahrbaren Funktionen implementiert. Das umfasst auch die Implementierung der Applikations-spezifischen Geschäftslogik einer Domäne.

1. Adapter-Schicht:

Die Adapter-Schicht baut auf der Applikations-Schicht auf und hängt daher von ihr ab. Der Zweck der Adapter-Schicht ist es, Benutzerinteraktionen an die Applikations-Schicht wie­ter­zugeben. Dabei dient die Adapter-Schicht als „Anti Corruption Layer“, der Domänen- und Applikations-spezifischen Quellcode von festen technischen Implementierungen ab­kop­pelt. Das meint auch die Konvertierung der Daten aus eingehenden Anfragen in ein für die darunter liegenden Schichten verständliches Format.

1. Plugin-Schicht:

Die Plugin-Schicht liegt ganz außen und hängt von der darunter liegenden Adapter-Schicht ab. Auf der Plugin-Schicht werden technische Implementierungen wie z. B. das Ein­binden einer Datenbank oder eines IAM-Services vorgenommen. Außerdem werden auf dieser Schicht die Funktionen einer Applikation den Benutzern zugänglich gemacht.

## Implementierung in der Applikation „Taskitory“

Die Applikation Taskitory verwendet das Konzept der Clean Architecture mit allen oben be­schriebenen Schichten. Um die Eigenschaft der nach innen gerichteten Abhängigkeiten in der Applikation Taskitory sicherzustellen, wird für jede Schicht ein eigenes Maven- Mo­dul angelegt. Die Maven-Module werden wie oben beschrieben über Ab­hän­gig­kei­ten miteinander verbunden. Nachfolgend soll auf alle Schichten und deren Aufgabenfelder ein­gegangen werden.

1. Abstraktions-Schicht:

Die Abstraction-Schicht der Applikation Taskitory enthält keinen eigenen Quellcode. Lediglich die folgenden Abhängigkeiten werden auf dieser Schicht eingebunden:

* Java Persistence API (JPA)
* Apache Commons-Lang
* Spring-Context
* Lombok

Die „Java Persistence API“ (JPA) ist ein de fakto Standard für die Integration einer Da­ten­bank in eine Java-Applikation. Da die JPA eine weit verbreitete, beständige Software-Bib­lio­thek ist, ist es nicht notwendig sie als Teil der Plugin-Schicht einzubinden. Denn das wür­­de umfassende Implementierungen auf der Adapter-Schicht nach sich ziehen. Die JPA hat keinen Bezug zu der Domäne der Applikation Taskitory und kann als Standard-Tech­nik angesehen werden. Daher gehört diese Abhängigkeit auf die Abstraktions-Schicht.

Die Software-Bibliothek Apache Commons-Lang stellt Funktionalitäten für die Ver­ar­bei­tung von Komponenten der Java Standard-Bibliotheken zur Verfügung, die nicht von die­sen Java Standard-Bibliotheken mitgeliefert werden. Mit Apache Commons-Lang werden all­­gemeine Problemstellungen wie z. B. das Hashen erledigt. Diese allgemeinen Funk­tio­nen können für alle folgenden Schichten re­le­vant sein. Daher wird diese Abhängigkeit auf der Abstraktions-Schicht ein­ge­bun­den.

Die Abhängigkeiten „Spring-Context“ und „Lombok“ sind zwei Software-Bibliotheken, de­ren Funktionen allen anderen Schichten zur Verfügung stehen soll. Mit Spring-Context wer­­den Funktionen zur Verfügung gestellt, die das Betreiben einer eigenen Applikation mit dem Spring-Framework ermöglichen. Lombok ist eine Software-Bibliothek mit der Quellcode vereinfacht werden kann. Lombok ermöglicht es, Stan­dard-Konstrukte in Java wie z. B. Getter und Setter durch eine Annotation zu ersetzen. Die Funk­tionen der beiden Software-Bibliotheken Spring-Context und Lombok beziehen sich auf allgemeine Pro­gram­mier-Konzepte und Konstrukte der Programmiersprache Java. Da­her gehören beide Abhängigkeiten auf die Abstraktions-Schicht.

1. Domänen-Schicht:

Die Domänen-Schicht ist das Herzstück der Applikation Taskitory. Hier wird die grund­le­gen­de Geschäftslogik der Domäne definiert. Es wird ein statisches Modell auf­gestellt, das die fachlichen Akteure der Applikation Taskitory definiert. Das umfasst besonders die En­ti­täten „User“, „Projekt“, „Projekt-Mitgliedschaft“, „Auf­ga­be“, „Kanbanboard“, „Tag“, „Nach­richt“. In Kapitel 3 wird die Domänen-Schicht näher beschrieben.

1. Applikations-Schicht:

Auf Grundlage des statischen Modells aus der Domänen-Schicht wird in der App­likations-Schicht ein dynamisches Modell der Abläufe aufgestellt. Hier werden die Use Ca­ses umgesetzt. Dabei wird der Informationsfluss von Akteur zu Akteur unter Be­rück­sich­tigung der Geschäftslogik aus der Domänen-Schicht abgebildet. Die Use Cases werden selbst durch Applikations-spezifische Geschäftslogik gelenkt. Die wichtigsten Use Cases der Applikation Taskitory sind in Kapitel 4 beschrieben.

1. Adapter-Schicht:

Die Applikation Taskitory benutzt für Benutzerinteraktionen eine REST-Schnittstelle. Es gibt kein Frontend, das die Validierung von Anfrage-Parametern übernimmt. Da­her müssen die eingehenden HTTP-Anfragen in dem Backend validiert und unter Um­stän­den umgewandelt werden, bevor sie an die inneren Schichten der Applikation weiter­ge­ge­ben werden. Für diesen Zweck ist die Adapter-Schicht gedacht. Hier werden Da­ten­trans­ferobjekte (DTO) und die nötige Validierungs-Logik definiert.

1. Plugin-Schicht:

Die Plugin-Schicht der Applikation Taskitory enthält die technischen Implementierungen für die Kommunikation mit einer Postgresql Datenbank und einem Keycloak IAM-Server. Zu­dem werden die Funktionen der Applikation Taskitory auf dieser Schicht den Benutzern über eine REST-Schnittstelle verfügbar gemacht.

# Domain Driven Design

## Problemdomäne

Die Problemdomäne einer Applikation setzt sich aus einer Kerndomäne, mehreren un­ter­stüt­zenden und mehreren ge­nerischen Domänen zusammen. Die Kerndomäne bestimmt die zentralen Funk­tio­na­li­tä­ten und den Zweck einer Applikation. Unterstützenden Do­mä­nen beziehen sich auf Aufgaben-Felder, die nicht direkt Teil der Kerndomäne sind, aber die Kerndomäne mit ihren Funktionen unterstützen. Die Kerndomäne verwendet die un­ter­stützenden Domänen, um die zentralen Funktionen einer Applikation umzusetzen. Die Ge­nerischen Domänen beziehen sich auf Aufgabenfelder, die nicht direkt mit dem Zweck ei­ner Applikation in Verbindung stehen, jedoch für das Betreiben einer Applikation not­wen­dig sind. Die Generischen Domänen liefern Basis-Funktionalitäten, ohne die eine App­li­kation nicht lauffähig ist.

Die Problemdomäne der Applikation „Taskitory“ ist die Aufgaben­ver­waltung in Projekten mit mehreren Projekt-Teilnehmern. Dabei ist die Kerndomäne das Verwalten von Auf­ga­ben. Das Umfasst nicht nur das Erstellen, Verändern und Löschen von Aufgaben, sondern auch die Funktionen eines Kanban Boards.

Die über die Kerndomäne zur Verfügung gestellten Daten und Funktionen werden Be­nut­zern der Applikation über die Domäne IAM zur Verfügung gestellt. Ohne eine Be­nut­zer­ver­waltung könnte das Konzept von Projekt-Teams und damit die ganze Applikation nicht um­gesetzt werden. IAM hat keine inhaltliche Verbindung zu der Kerndomäne und wird daher als generische Domäne eingestuft.

Die Verwaltung von Pro­jekten ist kein zwingend notwendiger Aufgabenbereich für eine Auf­­­gabenverwalten. Allerdings ist dieser Aufgabenbereich bei der Applikation „Tas­ki­tory“ relevant für die Funktionen der Kerndomäne und inhaltlich mit dieser verbunden. Denn alle Objekte, die in der Kerndomäne verwendet, erzeugt oder gelöscht werden, sind ei­­­nem Projekt zugeordnet. Auch die Rolle eines Benutzers in einem Projekt kann in diese Do­­­mäne gezählt werden. Denn der Zugriff auf die Funktionen der Kerndomäne wird durch die Rolle eines Be­­nutzers in einem Projekt eingeschränkt. Auf Grund der in­halt­li­chen Verbindung zur Kerndomäne wird die Verwaltung von Projekten und der damit as­so­ziierten Projekt-Teams als un­ter­stüt­zen­­de Domäne eingestuft.

Das Erstellen von Statistiken ist keine für eine Aufgabenverwaltung notwendige Funktion. Al­­lerdings werden mit Statistiken die Daten von Kanban Boards ver­ar­bei­tet, die aus der Kern­domäne stammen. Es besteht damit eine klare inhaltliche Verbindung zur Kern­do­mä­ne. Das Erstellen von Statistiken ist damit eine unterstützende Domäne.

Die Applikation „Taskitory“ informiert Benutzer mit Nachrichten über bestimmte Er­eig­nis­­se, die für den Benutzer relevant sind. Diese Ereignisse betreffen i. d. R. nicht die Kern­do­­mäne, sondern unterstützende Domänen. Es besteht daher keine inhaltliche Ver­bin­dung zur Kerndomäne. Das Messaging ist eher eine Basisfunktionalität und wird daher als ge­nerische Domäne eingestuft. Abbildung 1 zeigt noch einmal die Problemdomäne der App­likation „Taskitory“ mit allen Bestandteilen.

**Problemdomäne**

IAM

**Generische Domänen**

**Unterstützende Domänen**

Projekt-Verwaltung

**Kerndomäne**

Aufgabenverwaltung

Statistiken

Messaging

Abb. 1: Die Problemdomäne der Applikation „Taskitory“

## Ubiquitous Language

Der Begriff „Ubiquitous Language“ bezeichnet eine Sammlung von Begriffen, die ver­wen­det werden, um sich über eine Applikation zu unterhalten. Das umfasst Fachbegriffe der Pro­blem­domäne und der technischen Implementierung. Es müssen nicht triviale Objekte, Kon­zepte und Regeln eindeutig definiert werden.

Die Ubiquitous Language der Applikation Taskitory umfasst die folgenden Be­griffe für das Be­­schreiben problemdomänenspezifischer Objekte, Konzepte und Regeln:

1. Projekt

Ein Projekt ist ein zielgerichtetes Vorhaben, das aus einer Menge von Aufgaben besteht, die von einer Gruppe von Benutzern verwaltet und bearbeitet werden. Ein Projekt hat eine eindeutige Bezeichnung und eine Beschreibung, die von Pro­jekt-Ad­mi­­nistratoren festgelegt und verändert werden kann. Für die Erfüllung die­ser Aufgaben, stellen Projekte Kanbanboards und Tags zur Verfügung. Projekte sind der zen­trale Kontext, in dem ein Benutzer die Applikation Taskitory verwendet.

Über die Zeit können Benutzer ein Projekt verlassen und wieder beitreten. Für das Ver­las­­sen eines Projekts wird keinerlei Zustimmung von Seiten des Projekt-Teams benötigt. Al­­lerdings kann ein Benutzer einem Projekt nur dann beitreten, wenn er den geheimen Pro­­jekt-Schlüs­­sel kennt. Dieser Projekt-Schlüssel kann nur von Projekt-Administratoren ein­­­gesehen werden. Der Projekt-Schlüssel kann ausschließlich von Projekt-Ad­mi­nis­tra­to­ren und nur auf einen zufällig generierten Wert geändert werden.

1. Benutzer

Ein Benutzer wird durch ein Benutzer-Konto repräsentiert. Dieses Benutzer-Konto be­steht aus einem Benutzernamen und einem Passwort. Beide Werte können durch den Be­nutzer selbst verändert werden.

Be­nutzer müssen keinem Projekt angehören, können aber beliebig vielen Projekten bei­tre­­­ten. Benutzer können die Applikation Taskitory ohne eine Projektmitgliedschaft nicht sinn­­voll nutzen. Daher können Benutzer neue Projekte erstellen, in denen sie automatisch als Projekt-Administrator eingetragen werden.

Es werden der Beginn von Projektmitgliedschaften und die eingenommene Rolle in einem Pro­­jekt festgehalten. Benutzer können die Rollen „Mitglied“ und „Admin“ einnehmen.

1. Projekt-Administrator

Pro­jekt-Administratoren sind Benutzer mit der Admin-Rolle. Wie der Name schon sagt, ist diese Eigenschaft projektspezifisch. Benutzer mit der Admin-Rolle haben alle Rechte eines Mitglieds und zu­sätzliche Rechte für die Projekt-Verwaltung.

Es muss zu jeder Zeit einen Ad­mi­nis­tra­tor in einem Projekt ge­ben. Wenn der letz­te Ad­mi­nis­trator ein Projekt verlässt, muss er einen neuen Projekt-Administrator aus dem Pro­jekt-Team bestimmen. Wenn es keine anderen Team-Mitglieder gibt, wird das Pro­jekt ge­löscht. Es gibt keine Ober­gren­­ze für die Anzahl von Projekt-Administratoren. Nur Projekt-Ad­­­ministratoren können Benutzer ein­la­den, entfernen und deren Rolle än­dern.

1. Projekt-Mitglieder

Projekt-Mitglieder sind Benutzer mit der Mitglieds-Rolle. Diese Eigenschaft ist pro­jekt­spe­zifisch. Die Projekt-Mitgliedschaft ermächtigt einen Benutzer Aufgaben, Kanban­boards und Tags in einem Pro­­jekt zu verwalten.

1. Aufgabe

Auf­gaben sind Teil eines Projekts und beschreiben eine Tätigkeit, die durchgeführt werden soll. Die Erfüllung aller Aufgaben führt zum Erreichen eines über­ge­­ordneten Pro­jekt-Ziels.

Auf­gaben sind einem Projekt zugeordnet und haben eine im Projekt eindeutige Be­zeich­nung und eine nicht eindeutige Beschreibung. Über einen Erstellungs- und einen Fer­tig­stel­lungs-Zeit­punkt wird der Bearbeitungszeitraum einer Aufgabe festgehalten. Der Be­ar­beitungs-Sta­tus einer Aufgabe ist eng damit verbunden. Der Bearbeitungs-Status kann die Werte der Spalten eines Kanbanboards (siehe unten) und einen neutralen Wert für die Zuweisung zum Backlog haben. Über einen Kom­­ple­xi­täts-Wert und eine Priorität kann die Wichtigkeit und eine damit verbundene Bearbeitungsreihenfolge festgestellt werden. Ei­ne Aufgabe enthält Informationen über den Autor der Aufgabe und die für die Be­ar­bei­tung aktuell zuständige Person. Die Aufgaben werden neben diesen At­­­tri­­bu­ten auch durch Tags mit Meta-Informationen beschrieben.

Aufgaben sind immer einem Projekt zugeordnet. Aufgaben können sich entweder auf ei­nem oder keinem Kanbanboard befinden. Wenn eine Aufgabe nicht auf einem Kann­ban­board zu finden ist, liegt sie im Backlog. Aufgaben können entweder direkt auf einem Kan­ban­­board erzeugt werden oder vorerst im Backlog abgelegt werden.

1. Backlog

Das Backlog eines Projekts ist die Sammlung aller Aufgaben, die keinem Kanbanboard zu­ge­ordnet sind. Aus dem Backlog können Aufgaben einem beliebigen Kanbanboard des glei­chen Projekts zugewiesen werden. Aufgaben, die von einem Kanbanboard entfernt wer­den, werden automatisch in das Backlog zurückgelegt.

1. Kanban Board

Eine Kanbanboard ist eine Tabelle, bei der jede Spalte einen Bearbeitungs-Status re­prä­sen­­tiert. Aufgaben können in Form einer Karteikarte in diese Spalten einsortiert werden, um den Bearbeitungs-Status der Aufgabe zu visualisieren. Um einen Fortschritt ab­zu­bil­den, werden die durch Karteikarten repräsentierten Aufgaben von einer in eine andere Spal­­te geschoben.

Kanbanboards sind einem Projekt zugeordnet und haben eine im Projekt eindeutige Be­zeich­nung und eine nicht eindeutige Beschreibung. Ein Kanban Board be­trifft einen be­stimm­ten „Sprint“ genannten Zeitraum. Dieser Zeit­raum wird durch Attribute für den Start- und den End-Zeitpunkt festgehalten. 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.

Kanbanboards können grundsätzlich eine beliebige Anzahl Spalten haben. Weit verbreitet sind die folgenden fünf Spalten. Die Spalten werden in der unten genutzten Reihenfolge auf einem Kanbanboard angeordnet.

* Todo

In dieser Spalte werden alle Aufgaben abgelegt, die noch nicht bearbeitet wurden und sich nicht aktuell in Bearbeitung befinden. Einem Kanbanboard hin­­zu­ge­fügte werden hier abgelegt, sofern ihr Bearbeitungs-Status kei­nen anderen Wert aufweist. In der App­­likation Taskitory ist diese Spalte immer vor­han­den.

* In Progress

In dieser Spalte werden alle Aufgaben abgelegt, die aktuell bearbeitet werden. In der App­­likation Taskitory ist diese Spalte immer auf einem Kanbanboard vorhanden.

* Review

In dieser Spalte werden alle Aufgaben abgelegt, die bereits bearbeitet wurden und de­ren Er­gebnis durch eine verantwortliche Person abgenommen werden muss. In der App­­li­ka­tion Taskitory muss diese Spalte nicht auf einem Kanbanboard vorhanden sein. Die Spalte kann ein- und ausgeblendet werden. Dabei werden die in diesen Spalten be­find­lichen Auf­ga­ben nicht entfernt. Allerdings können Aufgaben in einer aus­ge­blen­de­ten Spalte nicht ver­waltet werden.

* Testing

Diese Spalte enthält alle Aufgaben, die bereits bearbeitet wurden und noch Tests un­ter­­zo­gen werden. In der Applikation Tas­ki­tory muss diese Spalte nicht auf einem Kann­ban­­board vorhanden sein. Die Spalte kann ein- und ausgeblendet werden. Dabei wer­den die in diesen Spalten befindlichen Aufgaben nicht entfernt. Allerdings können Auf­ga­ben in ei­ner ausgeblendeten Spalte nicht ver­wal­tet werden.

* Done

Diese Spalte enthält alle Aufgaben, die abgeschlossen wurden. In der Applikation Tas­ki­­to­ry muss diese Spalte immer auf einem Kanbanboard vorhanden sein.

1. Sprint

Der Begriff „Sprint“ entstammt den agilen Vorgehensmodellen. Sprints sind feste Zeit­räu­me, in die ein Projekt eingeteilt wird. In diesen Zeiträumen nimmt sich das Projekt-Team be­stimmte Aufgaben vor, die mit Hilfe eines Kanban Boards verwaltet werden können. Nach Beendigung eines Sprints werden die Arbeitsergebnisse Qualitätsgesichert. Auf die­se Weise wird ein Produkt iterativ weiterentwickelt und kann mit den Anforderungen ab­ge­glichen werden.

1. Statistiken

Es sollen einfache statistische Analysen auf Kanbanboards durchgeführt werden können. Es soll mindestens ei­­ne Statistik erzeugt werden können. Ein Beispiel ist die Berechnung des Anteils der Aufgaben auf ei­nem Kanban Board, der zum Ende des Sprints in einem be­­stimm­ten Bearbeitungs-Sta­­tus ist. Statistische Analysen sollen nur von Projekt-Ad­mi­nis­tratoren erstellt und ein­ge­­sehen werden können.

1. Tag

Ein Tag ist ein Schild, das einer Aufgabe angeheftet werden kann, um diese Aufgabe mit Me­ta-Informationen zu versehen. Tags bestehen aus einer global eindeutigen Be­zeich­nung und einer nicht eindeutigen Beschreibung.

1. Nachricht

Eine Nachricht dient der Übermittlung von für einen Benutzer relevanten Informationen zu einem Projekt, dem er angehört. Solche relevanten Informationen sind z. B. eine Än­derung der Projekt-Bezeichnung, die Änderung der Rolle oder das Löschen des Pro­jekts. Auch die Einladung in ein Projekt wird über eine Nachricht umgesetzt.

Nachrichten werden nicht von einem Benutzer verschickt und können daher nicht dazu ge­nutzt werden, um eine Konversation zu führen. Nachrichten werden von einem Projekt ver­sendet und von einem Benutzer empfangen. Der Inhalt von Nachrichten kann nicht per­so­na­lisiert werden und ist für den jeweiligen Anwendungsfall vorbereitet. Über ein Attribut kön­nen Nachrichten als „gelesen“ markiert werden.

Die Ubiquitous Language der Applikation Taskitory umfasst die folgenden Be­griffe für das Be­­schreiben technischer Objekte, Konzepte und Regeln für die Implementierung der App­li­­kation:

1. Backend

Der Begriff „Backend“ meint im Umfeld der Applikation Taskitory die Java-Anwendung, in der die Geschäftslogik, die Datenbank- und IAM-Anbindung sowie die REST-Schnitt­stel­le der Applikation Taskitory implementiert ist. Das steht im Widerspruch zu der ei­gent­li­chen Bedeutung des Begriffs, da Benutzer direkt auf das Backend zugreifen. Alle anderen Eigenschaften sind jedoch typisch für ein Backend. Wenn zukünftig die Applikation Tas­ki­to­ry durch ein Frontend mit GUI erweitert wird, kann der direkte Zugriff der Benutzer auf das Back­end verhindert werden. Dann handelt es sich um ein echtes Backend.

1. Representational State Transfer (REST)

Representational State Transfer (REST) ist ein Konzept für die Architektur von verteilten Soft­ware-Systemen. Ein REST-Service stellt Hypermedia-Ressourcen über HTTP im In­ter­net zur Verfügung. Hypermedia-Ressourcen sind Informationen, die nicht auf ein Medium fest­gelegt sind und sich aus mehreren per Hyperlink verbundenen Daten-Objekten zu­sam­mensetzen. Eine solche Ressource wird über eine global eindeutige Adresse und eine Zu­griffsmethode (HTTP-Verben) zugänglich gemacht.

1. Java Persistence API (JPA)

Die „Java Persistence API“ (JPA) ist ein de fakto Standard für die Integration verschiedener Da­­ten­banken in eine Java-Applikation. Durch das Konzept vom Objektrelationalen Map­ping (OR-Mapping) können in einer Applikation erzeugte Objekte ohne großen Pro­gram­mieraufwand direkt in einer Datenbank persistiert werden.

1. Objektrelationales Mapping

Objektrelationales Mapping bezeichnet eine Technologie, mit der die Objekt-Struktur ei­ner objektorientierten Programmiersprache auf die Relationen-Struktur eines re­la­tio­na­len Datenbank-Systems abgebildet wird. Dabei werden Tabellen mit Spalten durch Klassen mit Attributen repräsentiert. Die Datensätze entsprechen im OR-Mapping den In­stan­zen der Klassen.

1. IAM-/Keycloak-Server

Unter Identity und Access Management (IAM) werden allgemein die Techniken ver­stan­den, mit denen Benutzer einer Software-Applikation authentifiziert und autorisiert wer­den. Das umfasst die Konfiguration von Authentifizierungs- sowie Au­to­ri­sie­rungs­me­tho­den, Rechtegruppen uvm.

Mit IAM-Server ist im Kontext der Applikation Taskitory einer Anwendung gemeint, die die­se IAM-Funktionen für eine andere Applikation übernimmt. Für die Applikation Tas­ki­to­ry wird dabei konkret die Applikation „Keycloak“ verwendet. Diese Applikation spei­chert ihre Daten in der gleichen Datenbank, in der auch das Backend Daten persistiert.

1. Datenbank/Postgresql

Eine Datenbank ist eine Software-Anwendung, die für die persistente Speicherung von Da­ten verwendet wird. Die Daten werden dabei hauptsächlich von anderen An­wen­dun­gen zur Verfügung gestellt. Mit der Datenbank-Sprache SQL können die Datenbestände ei­ner Datenbank erweitert, verkleinert und bearbeitet werden.

Für die Applikaiton Taskitory wird eine Postgresql-Datenbank verwendet. Postgresql ist eine umfangreiche Open Source Anwendung mit allen notwendigen Funk­tio­nen.

1. Docker-Container

Docker ist eine Software-Applikation, mit der andere Software-Applikationen in „Con­tai­nern“ betrieben werden können. Diese Container sind virtualisiert Laufzeitumgebungen für Soft­­ware-Applikation. Mit „Dockerfiles“ können diese Laufzeitumgebungen kon­fi­gu­riert und auf verschiedenen Computern exakt wiederhergestellt wer­­den. Dadurch können gan­ze Applikationen schnell auf andere Server bewegt und wie­­der hergestellt werden. Um den Zustand einer Applikation zu speichern und in einem neu­en Container wieder her­zustellen, können „Volumes“ eingebunden werden. Volumes sind Datei-Ordner, die die den Zustand repräsentierenden Dateien beinhalten.

1. Docker Compose

Docker-Compose ist eine Technologie, mit der mehrere Docker-Container zusammen verwaltet wer­den können. Das umfasst die Konfiguration der einzelnen Container, aber auch die Kommunikationswege der Container untereinander und nach au­ßen. Auf diese Weise können mehrere Container zu einem System ver­bun­den werden. Mit einer „Do­cker-Compose.yml“ Datei können solche Systeme beschrieben werden.

Durch einbinden einer „.env“ Datei ist es möglich sensible Informationen wie Passwörter aus dem Docker-Compose.yml und anderen Applikations-internen Dateien in Um­ge­bungs­­variablen auszulagern.

1. Postman Collection

Postman ist eine Software-Applikation mit der HTTP-Anfragen gezielt konfiguriert, ver­sen­det und empfangen werden können. Eine „Collection“ ist eine Sammlung von HTTP-An­fragen, die mit einer Verzeichnis-Struktur verwaltet und durch übergreifende Kon­fi­gu­ra­tionen wie z. B. die Authentifizierung von HTTP-Anfragen erweitert werden. Für die App­likation Taskitory wird eine solche Collection angelegt, um die Bedienung der App­li­ka­tion zu vereinfachen.

1. Hashing

Beim Hashen einer Zeichenfolge wird eine Einweg-Hash-Funktion auf diese Zeichenfolge an­gewendet. Hash-Funktionen bilden Zeichenfolgen beliebiger Länge auf Zeichenfolgen mit einer festen Länge ab.

Eine Untergruppe der Hash-Funktionen sind die kryptographischen Hash-Funktionen. Der erzeugte Hash-Wert einer kryptographischen Hash-Funktion darf praktisch nicht auf sei­nen Ausgangswert zurückführbar sein. Zudem darf es in annehmbarer Zeit nicht mög­lich sein zwei Zeichenketten zu finden die den gleichen Hash-Wert haben.

## Taktische Muster des Domain Driven Design

Entities

Aggregates

Repositories

Domain Services

Value Objects

# Use Cases

In diesem Kapitel werden zentrale Use Cases der für die Applikation Taskitory the­ma­tisch sortiert aufgeführt. Die hier geschilderten Informationen sind be­son­ders für die App­li­ka­tions-Schicht der Clean Architecture relevant.

**Benutzer und Projekte**

1. Benutzer-Konto registrieren

Um ein Benutzer-Konto zu registrieren, müssen ein Benutzername und ein Passwort bestimmt werden. Bei erfolgreicher Registrierung soll der Keycloak dem Benutzer direkt einen Access-Token zurückgeben. Bei einem Fehler wird ein passender Fehlercode zu­rückgegeben. Das Erstellen eines Eintrags in der Datenbank für das neue Be­nut­zer­kon­to wird von dem Keycloak erledigt und kann über die REST-API des Keycloaks getriggert wer­den. Für das Registrieren eines Benutzerkontos muss das Backend eine Anfrage le­dig­lich an den Keycloak weiterleiten.

1. Benutzer-Konto löschen

Wenn ein Benutzer sich entscheidet, sein Konto zu löschen, werden alle damit ver­bun­de­nen Daten unwiederbringlich gelöscht. Das Ergebnis des Löschvorgangs wird durch einen passenden HTTP-Status quittiert. Das Löschen des Be­nut­zer­kon­to-Datensatzes in der Da­ten­bank wird von dem Keycloak erledigt und kann über die REST-API des Keycloaks ge­trig­gert wer­den. Für das Löschen eines Benutzerkontos muss das Backend eine Anfrage le­­dig­lich an den Keycloak weiterleiten.

1. Projekt anlegen

Ein Projekt wird von einem Benutzer mit einer Be­zeichnung und einer Beschreibung an­ge­legt. Der geheime Gruppen-Schlüssel wird automatisch generiert. Optional kann der Be­nut­zer direkt ein Projekt-Team aus Benutzern zusammenstellen. Dabei kann er ent­schei­den, wel­che Rolle ein Benutzer-Konto erhält. Der Ersteller eines Projekts wird au­to­ma­tisch als Ad­mi­nistrator dem Projekt zugeordnet. Ein Projekt beinhaltet nach der Ini­tia­li­sie­rung keine Aufgaben und keine Kanban Boards.

1. Benutzer-Konto zu einem Projekt hinzufügen

Das Projekt-Team kann auch nach der Initialisierung erweitert werden. Um einem Projekt beitreten zu können, muss ein Benutzer den geheimen Schlüssel eines Projekts angeben. Diesen geheimen Schlüssel können Projekt-Administratoren per Nachricht an den jeweiligen Benutzer schicken. Damit der Schlüssel nicht so einfach kompromittiert werden kann, wird der Schlüssel gehasht übertragen. Wenn ein Benutzer einem Projekt beigetreten ist, werden alle Projekt-Mitglieder und auch der neu hinzugefügt Benutzer darüber in einer Nachricht informiert.

1. Ein Benutzer-Konto befördern

Projekt-Administratoren sollen einfache Mitglieder zu Projekt-Administratoren be­för­dern können. Der beförderte Benutzer und alle Mitglieder eines Projekt-Teams sollen über die Beförderung informiert werden.

1. Benutzer-Konto aus Projekt-Team entfernen

Ein Benutzer kann aus eigenem Antrieb aus einem Projekt austreten. Ein Benutzer mit der Rolle „Mitglied“ kann auch gegen seinen Willen von einem Projekt-Administrator aus einem Projekt entfernt werden. Dadurch werden die Referenzen auf den Benutzer aus dem Projekt entfernt. Der entfernte Benutzer soll per Nachricht informiert werden, wenn ein Administrator ihn aus einem Projekt entfernt hat.

1. Projekt-Attribute anpassen

Die Bezeichnung, die Beschreibung und der geheime Projekt-Schlüssel eines Projekts sollen von einem Projekt-Ad­mi­nis­tra­tor geändert werden können. Benutzer mit der Rolle „Mitglied“ können diese At­tri­bute nicht ändern. Der geheime Schlüssel eines Projekts kann nur auf einen zufälligen Wert geändert werden.

**Aufgaben**

1. Aufgaben anlegen

Aufgaben sind immer einem Projekt zugeordnet. Aufgaben können daher nur angelegt werden, wenn ein Benutzer mindestens einem Projekt angehört. Im Stan­dard­fall wird eine neue Aufgabe zu dem Backlog eines Pro­jekts hinzugefügt. Wenn bereits ein Kanban Board erstellt wurde, kann eine Aufgabe auch direkt dem Kanban Board hinzugefügt wer­den. Dort wird die Aufgabe in der Spalte „Todo“ abgelegt. In diesem Fall wird die Auf­ga­be nicht mehr Im Backlog angezeigt.

1. Benutzern Aufgaben zuweisen

Eine bestehende Aufgabe kann einem Mitglied des Projekt-Teams zur Bearbeitung zu­ge­wie­sen werden. Dabei ist es egal, ob die Aufgabe vorher bereits einer Person zugewiesen war oder nicht. Es ist auch möglich die Zuweisung aufzuheben und keine neue Zuweisung vor­zunehmen („NULL“). Der zu­gewiesene Benutzer wird über eine Nachricht informiert.

1. Aufgaben zu einem Kanban Board hinzufügen

Aufgaben können aus dem Backlog zu genau einem Kanban Board hinzugefügt werden. Dazu muss eines aus vielen möglichen Kanban Boards eines Projekts gewählt werden. Außerdem muss festgelegt werden, in welcher Spalte die Aufgabe abgelegt werden soll. Der Standardfall ist, dass die Aufgabe in der Spalte „Todo“ abgelegt wird.

1. Aufgaben bearbeiten

Die Attribute „Be­zeich­nung“, „Beschreibung“, „Komplexität“ und „zuständige Person“ (siehe oben) kön­nen nach der Erstellung einer Aufgabe jederzeit durch einen Benutzer verändert werden. Es bleibt zu beachten, dass die Bezeichnung der Aufgaben in einem Pro­jekt eindeutig ist.

1. Aufgaben löschen

Aufgaben können von allen Benutzern gelöscht werden. Dabei spielt es keine Rolle, ob die Aufgabe auf einem Kanban Board oder dem Backlog liegt.

**Tags**

1. Tag erstellen

Tags können mit einer Bezeichnung und einer Beschreibung erzeugt werden und sind einem Benutzer zugeordnet. Die Bezeichnung ist global eindeutig.

1. Tag bearbeiten

Die Beschreibung eines Tags kann im Nachhinein vom Erzeuger des Tags geändert wer­den.

1. Tag löschen

Ein Tag kann von seinem Erzeuger gelöscht werden. Dadurch werden automatisch alle Re­ferenzen von Auf­ga­ben entfernt.

1. Tags zuweisen

Aufgaben können über Tags mit Metadaten wie z. B. Versionsnummern uvm. versehen wer­den. Dabei ist der Zugriff nicht auf die eigens erzeugten Tags beschränkt.

**Kanban Board**

1. Kanban Board anlegen

Kanban Boards können mit einer für ein Projekt eindeutigen Bezeichnung, einer Beschreibung und einem festgelegten Sprint-Zeitraum erzeugt werden. Optional können noch die zwei Spalten „Review“ und „Testing“ aktiviert werden.

Ein Kanban Board ist bei Er­zeu­gung leer und besitzt die oben benannten Spalten. Es sollen mehrere Kanban Boards in einem Projekt gleichzeitig bestehen können.

1. Attribute eines Kanban Boards bearbeiten

Die Bezeichnung und die Beschreibung eines Kanban Boards können durch die Benutzer des entsprechenden Projekts angepasst werden. Dabei bleibt zu beachten, dass die Bezeichnung in einem Projekt eindeutig sein muss.

Außerdem ist es möglich die Spalten „Review“ und „Testing“ auszublenden. Alle in diesen Spalten befindlichen Aufgaben werden daraufhin ausgeblendet. Allerdings behalten sie den Status und können anschließend wieder eingeblendet werden.

1. Kanban Board löschen

Kanban Boards können durch Benutzer des entsprechenden Projekts gelöscht werden. Die darauf befindlichen Aufgaben werden daraufhin in das Backlog gelegt. Die Aufgaben behalten ihren Bearbeitungs-Status.

1. Aufgabe auf einem Kanban Board verschieben

Die zentrale Funktion eines Kanban Boards ist, Aufgaben über ihren Bearbeitungs-Sta­­tus zu verwalten. Be­nut­zer können die Aufgaben auf einem Kanban Board in einen anderen Be­arbeitungs-Status verschieben. Der Bearbeitungs-Status einer Aufgabe im Backlog ei­nes Projekts kann nicht geändert werden.

1. Kanban Board überführen

Das Kanban Board eines vergangenen Sprints kann in ein neues Kanban Board überführt 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. Statistiken erstellen

Es soll möglich sein, Statistiken über ein bestehendes Kanban Board anzulegen. Mögliche Anwendungsfälle sind z. B. die Verteilung von Aufgaben nach Bearbeitungs-Status oder die Berechnung einer durchschnittlichen Bearbeitungsdauer.

**Nachrichten**

1. Benutzer zu Projekt einladen

Projekt-Administratoren können Benutzer, die noch kein Teil des Projekt-Teams sind zu dem Projekt einladen. Dazu muss der geheime Schlüssel eines Projekts von einem Projekt-Ad­ministrator an diesen Benutzer in gehashter Form geschickt werden. Der Benutzer muss diesen gehashten Schlüssel vorweisen, um einem Projekt beizutreten.

1. Nachrichten empfangen

Benutzer können die erhaltenen Nachrichten empfangen und lesen. Dabei ist es möglich Nachrichten einzeln und in größeren Gruppen zu empfangen. Wurde eine Nachricht einmal empfangen, wird sie unwiederbringlich gelöscht.

# Entwurfsmuster

A

# Programming Principles

A

# Testing

## Unit Testing

A

## Mocks

A

# Refactoring

A

Literaturverzeichnis

1. a