Dokumentation Gruppe 4

**Thema: üK 223**

**Dokumentinformationen**

Dateiname: Dokumentation Gruppe 4.docx

Speicherdatum: 05.11.2021

**Autoreninformationen**

Autor: Anojan Pirabakaran | Enes Spahiu

E-Mail: anojan.pirabakaran@noseryoung.ch | enes.spahiu@noseryoung.ch

**Inhaltsverzeichnis**

[1 Einleitung 4](#_Toc87617681)

[Sinn und Zweck 4](#_Toc87617682)

[Abkürzungen 4](#_Toc87617683)

[2 Aufgabenstellung 5](#_Toc87617684)

[Funktionale Anforderungen 5](#_Toc87617685)

[Nicht funktionale Anforderungen 5](#_Toc87617686)

[Implementation 5](#_Toc87617687)

[Testing 5](#_Toc87617688)

[Multiuserfähigkeit 5](#_Toc87617689)

[Dokumentation 6](#_Toc87617690)

[Github Repository 7](#_Toc87617691)

[Domänenmodell 7](#_Toc87617692)

[Klassen-Diagramm 8](#_Toc87617693)

[Use-Cases 9](#_Toc87617694)

[Sequenz-Diagramm 9](#_Toc87617695)

Abbildungsverzeichnis

Tabellenverzeichnis

[Tabelle 1 Versionen 2](#_3znysh7)

[Tabelle 2 Abkürzungen 3](#_4d34og8)

Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Autor** | **Details** |
| 1.0 | 05.11.2021 | Anojan | Dokument erstellt |

**Tabelle 1 Versionen**

# Einleitung

## Sinn und Zweck

Das vorliegende Dokument beschreibt unsere Dokumentation zu unserem Abschlussprojekt im üK Modul 223. In diesem Modul geht es, um Multi-User-Applikationen. Wir werden unseren Schwerpunkt auf Spring Boot und Datenbank setzen.

## Abkürzungen

|  |  |
| --- | --- |
| **Abkürzung** | **Beschreibung** |
| NYP | **N**oser **Y**oung **P**rofessionals |

**Tabelle 2 Abkürzungen**

# Aufgabenstellung

## Funktionale Anforderungen

Unsere Aufgabe als Team 4 ist es ein Gruppen-System zu implementieren, bei dem man User in eine Gruppe einteilen kann und die User haben dann auch als einziges Zugriff auf die Daten der Gruppe.

Eine Bedingung ist, dass User nur in maximal einer Gruppe sein können, sind sie also schon in einer Gruppe dürfen sie nicht auch noch in einer anderen Gruppe sein. Die andere Bedingung besagt, dass eine Gruppe nur von einem Admin erstellt und verwaltet werden kann, also jemandem mit Admin rechten.

Nur User mit Adminrechte oder User, die in einer Gruppe sind, können Informationen der Gruppen anschauen, verwalten und löschen.

Mithilfe der Pagination, sollten wir Endpoints in unserem Programm implementieren, um alle Mitglieder einer Gruppe anzuschauen.

Die einzelnen Gruppen brauchen Informationen, wie Mitglieder, Gruppenname, Motto etc.

## Nicht funktionale Anforderungen

## Implementation

Die einzelnen Daten müssen in einem PostgreSQL Datenbank speichert und mit OR-Mapping und JPA realisiert werden.

## Testing

Ein Backend-Endpoint muss definitiv getestet werden. Funktionalitäten, die wir implementiert haben, müssen mit Component-Tests in Post getestet werden. Die Endpoints werden mit mehreren Usern und Rollen getestet und mindestens ein Erfolgsfall und ein Error Fall muss getestet werden.

## Multiuserfähigkeit

Die ACID-Prinzipien werden eingehalten.

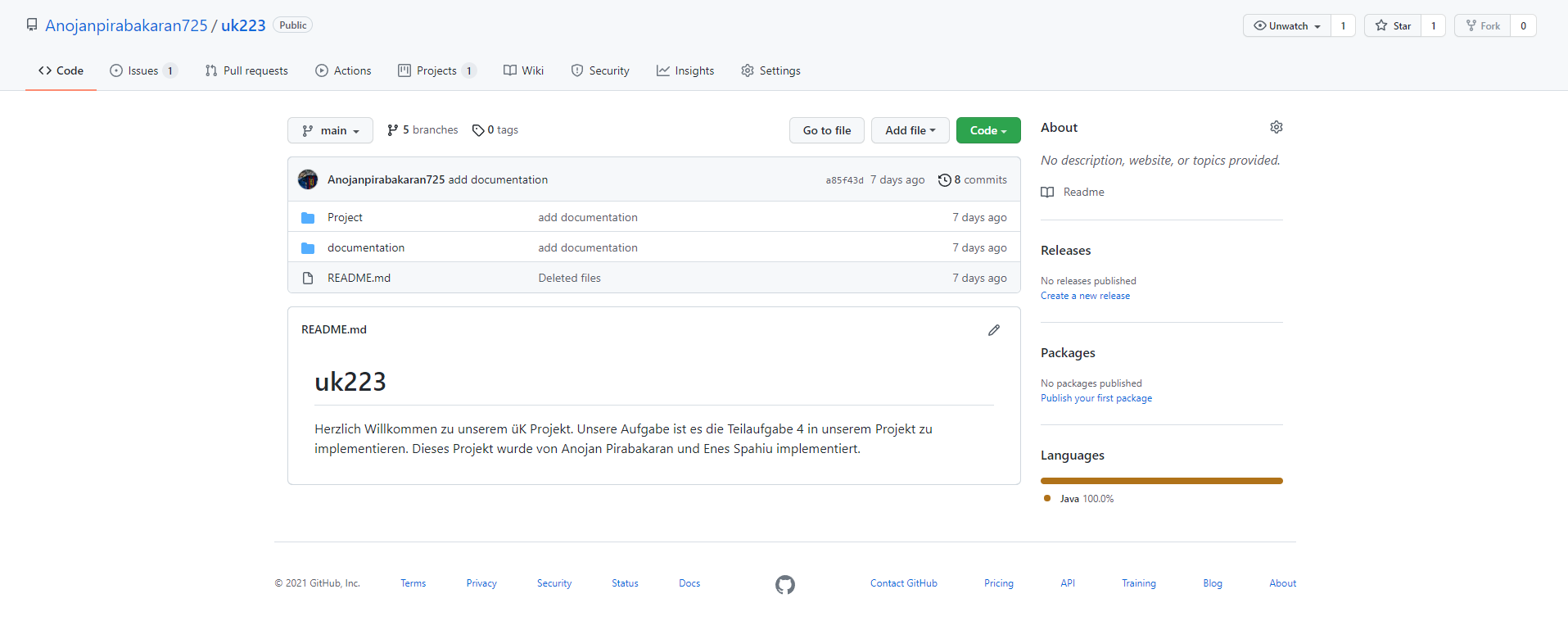
## Dokumentation

Die Funktionen werden genau und verständlich dokumentiert. Die Dokumentation erhält:

* ein Readme File im Repo
* ein Domänenmodell
* ein Klassen-Diagramm
* ein Sequenz-Modell
* UseCase

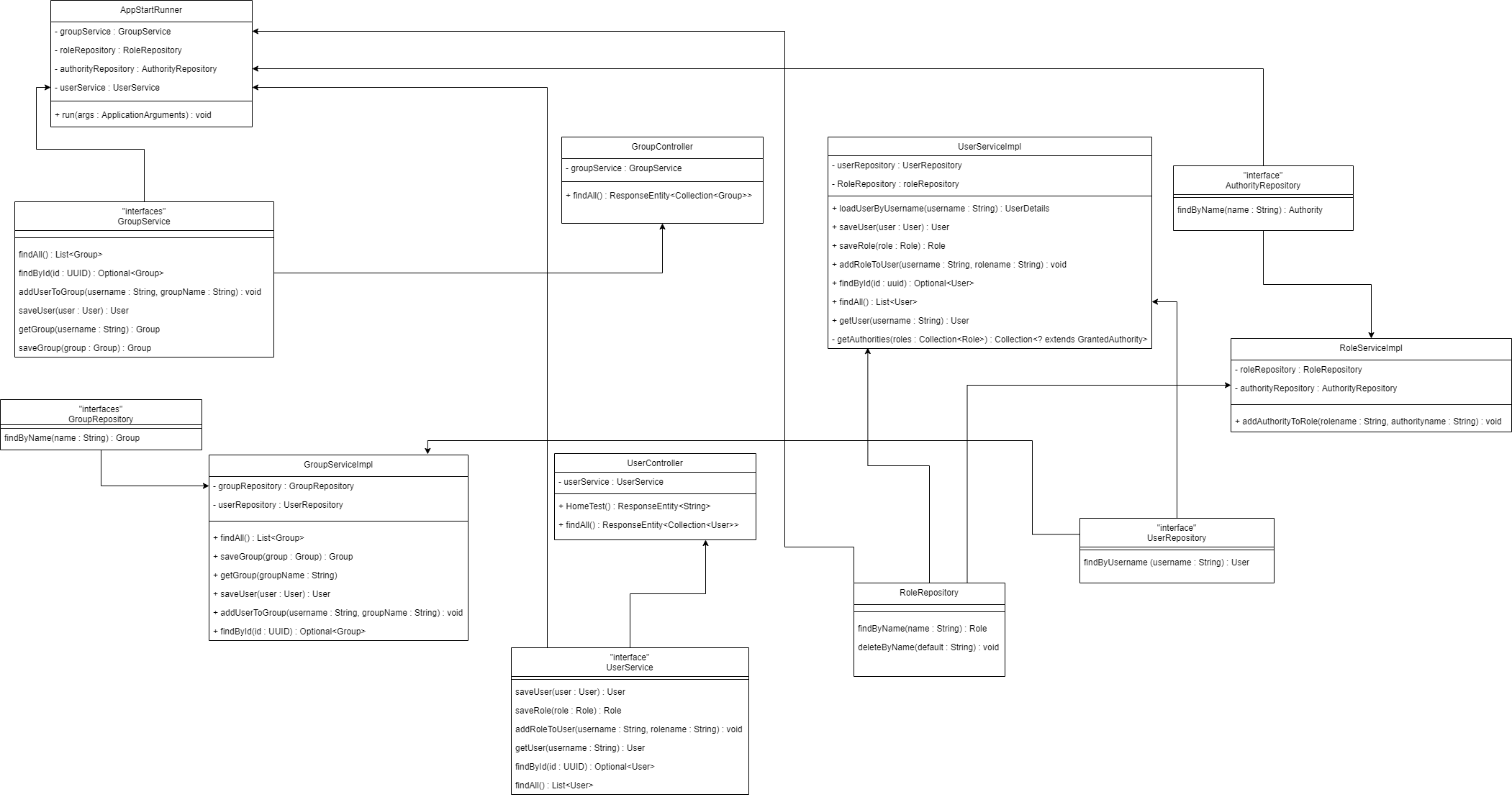
Administratives

## Github Repository

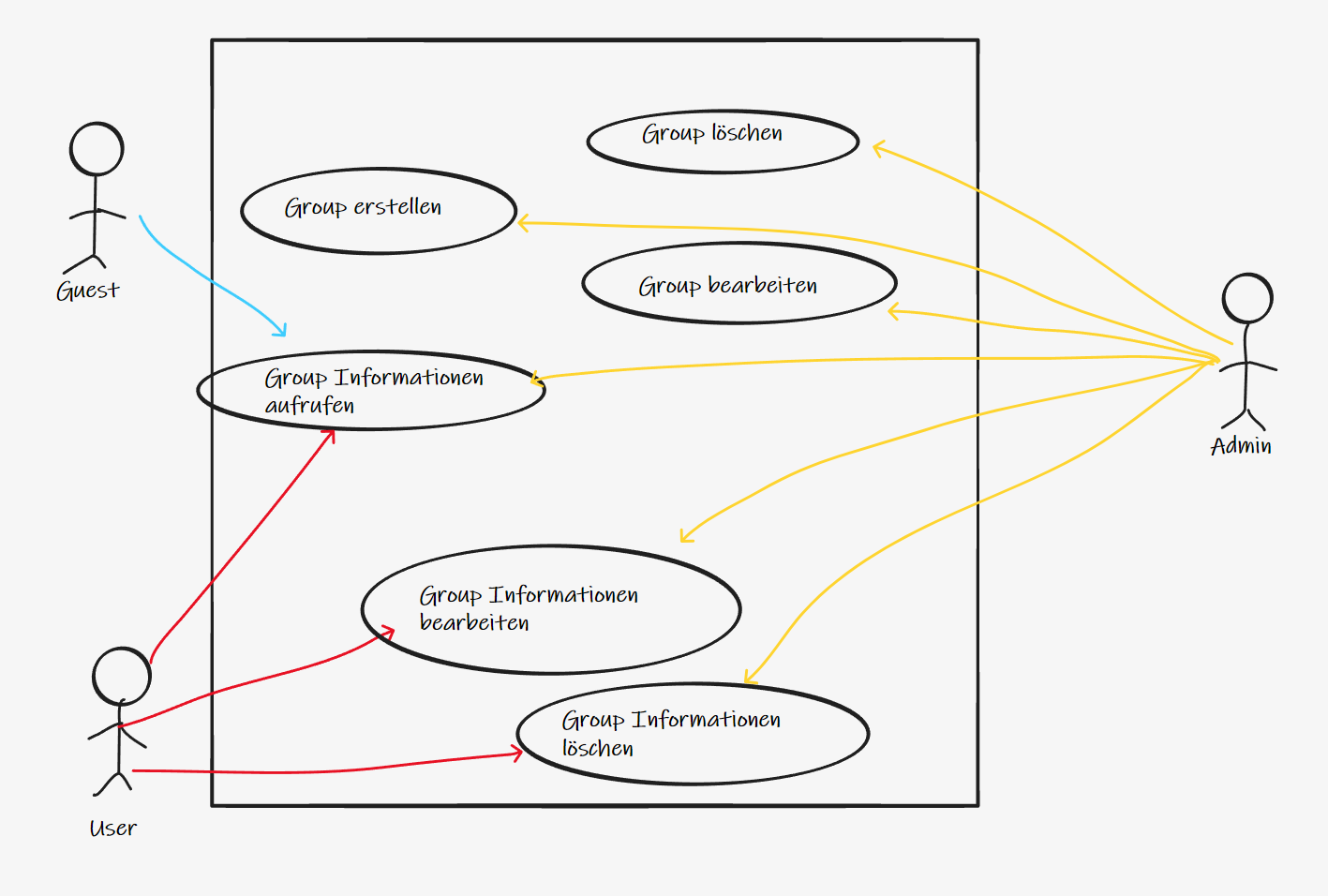


## Domänenmodell

## Klassen-Diagramm



## Use-Cases



## Sequenz-Diagramm