

Dokumentation Java Projekt

TO-DO

ALEX ZIEGLER, JULIAN KÜHNE

25.01.2022

Inhaltsverzeichnis

1	Projektbeschreibung	2
1.1	Github	2
2	Use Case Diagram	2
3	Klassendiagramm	3
3.1	Vor dem coden	3
3.2	Nach dem coden.....	4
4	Aktivitätsdiagramm	5
5	JDK.....	7
6	IDE	7
7	Gradle	7
8	Vererbung und Abstrakte Klasse.....	7
8.1	Abstrakte Klasse	7
8.2	Vererbung	7
9	JavaDoc.....	8
10	Anwendung.....	8
11	Reflexion.....	8

1 Projektbeschreibung

Wir codieren eine To-Do Applikation, mit der man nach einem Benutzerlogin Tasks erstellen, bearbeiten und löschen kann. Diese Tasks kann man zudem an Personen zuweisen, und jeweils die eigenen Tasks anzeigen lassen. Zugewiesene Tasks werden den Personen angezeigt, welche diese assigned haben.

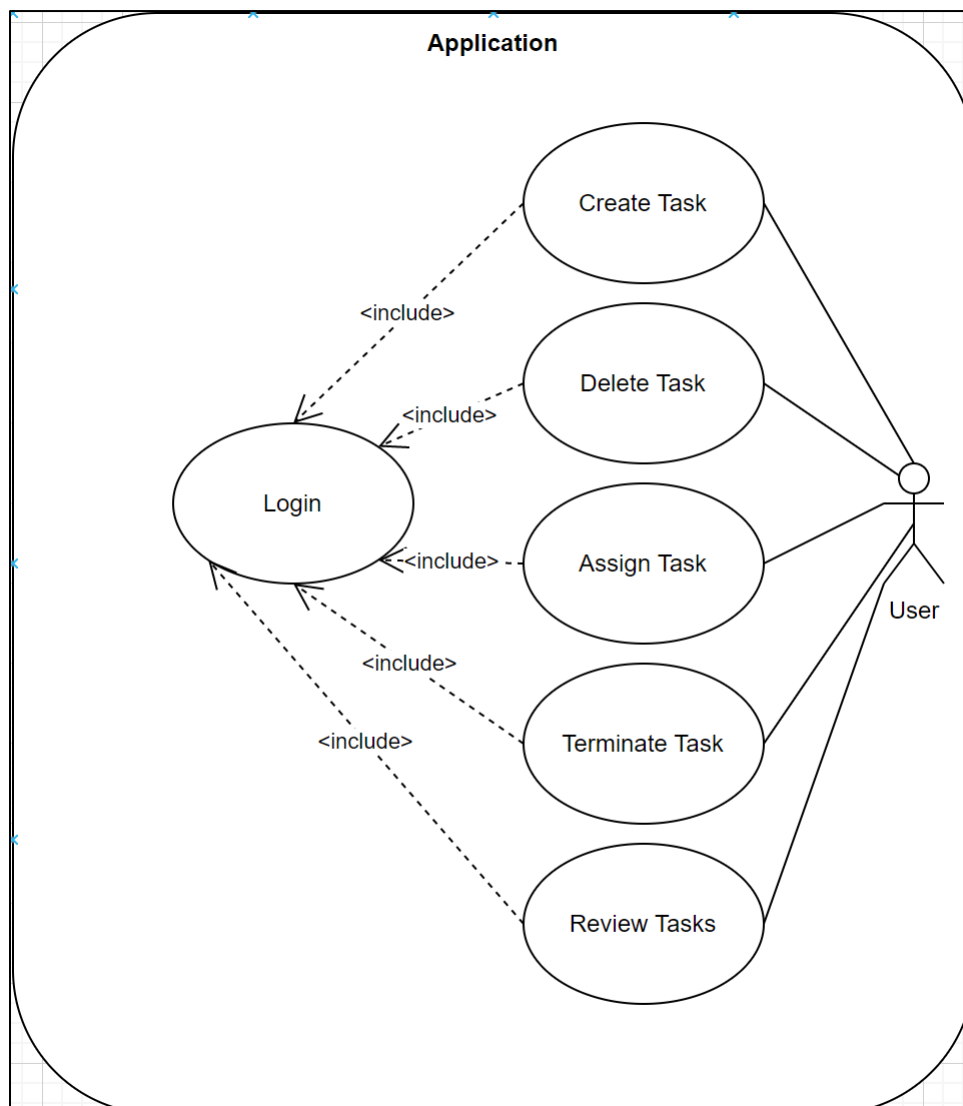
1.1 Github

Das Projekt wurde auf Github veröffentlicht und kann unter dem folgenden Link abgerufen werden:

<https://github.com/alxeex/M226-Todo>

2 Use Case Diagramm

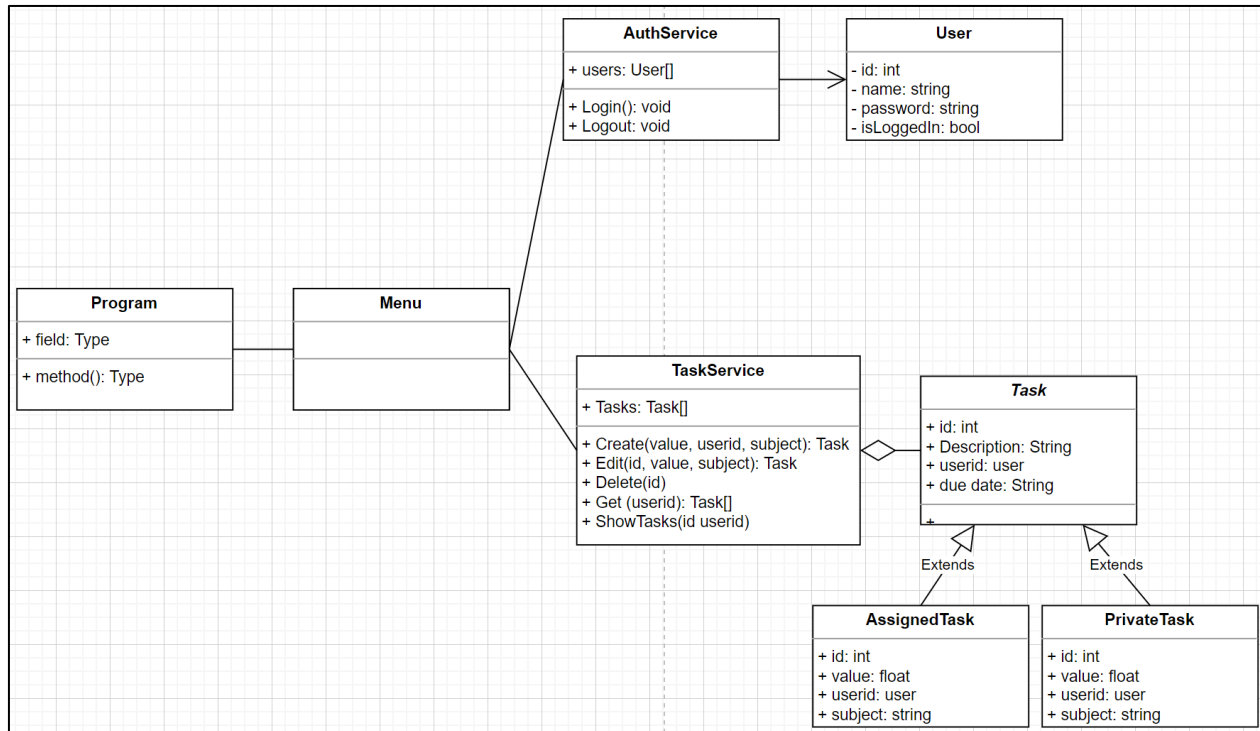
Achtung dies ist unsere Planung am Anfang des Projektes!



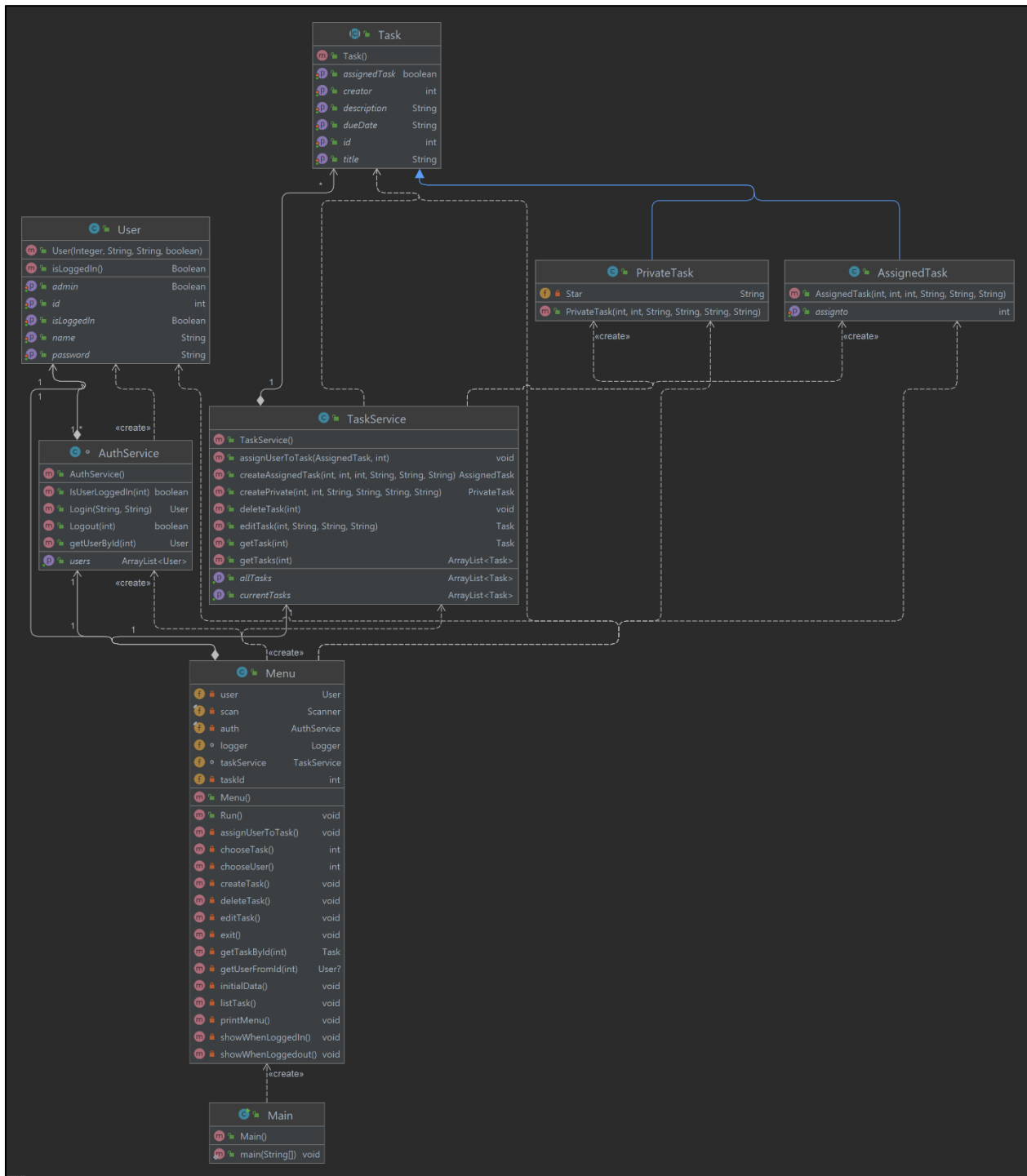
3 Klassendiagramm

3.1 Vor dem coden

Die Klasse *Task* ist eine Abstrakter Oberklasse. Die Klassen «AssignedTask» und «PrivateTask» erben von der Klasse *Task*. Das es nicht sehr sinnvoll ist, war uns bewusst, jedoch mussten wir Vererbung einbauen.

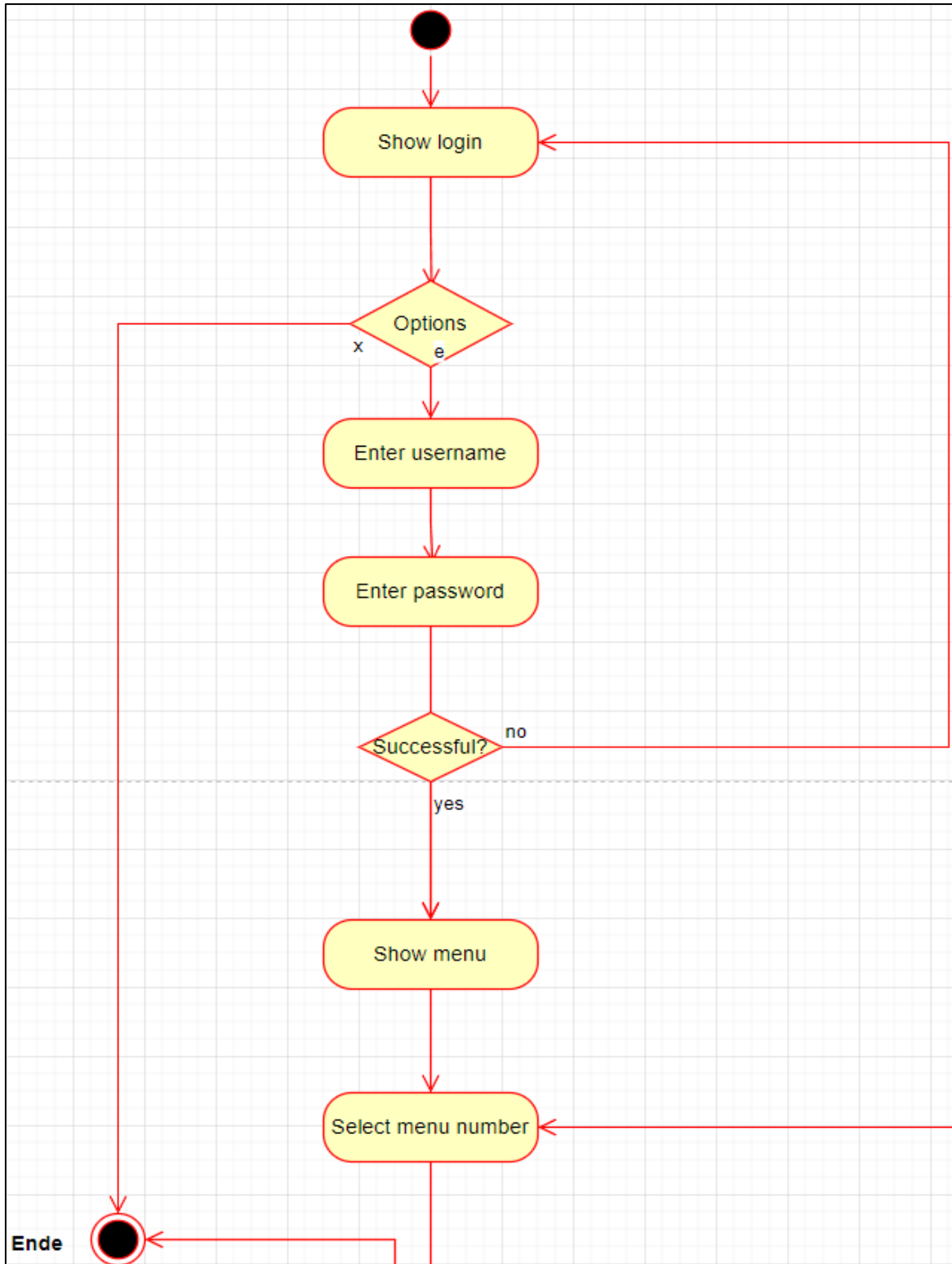


3.2 Nach dem coden

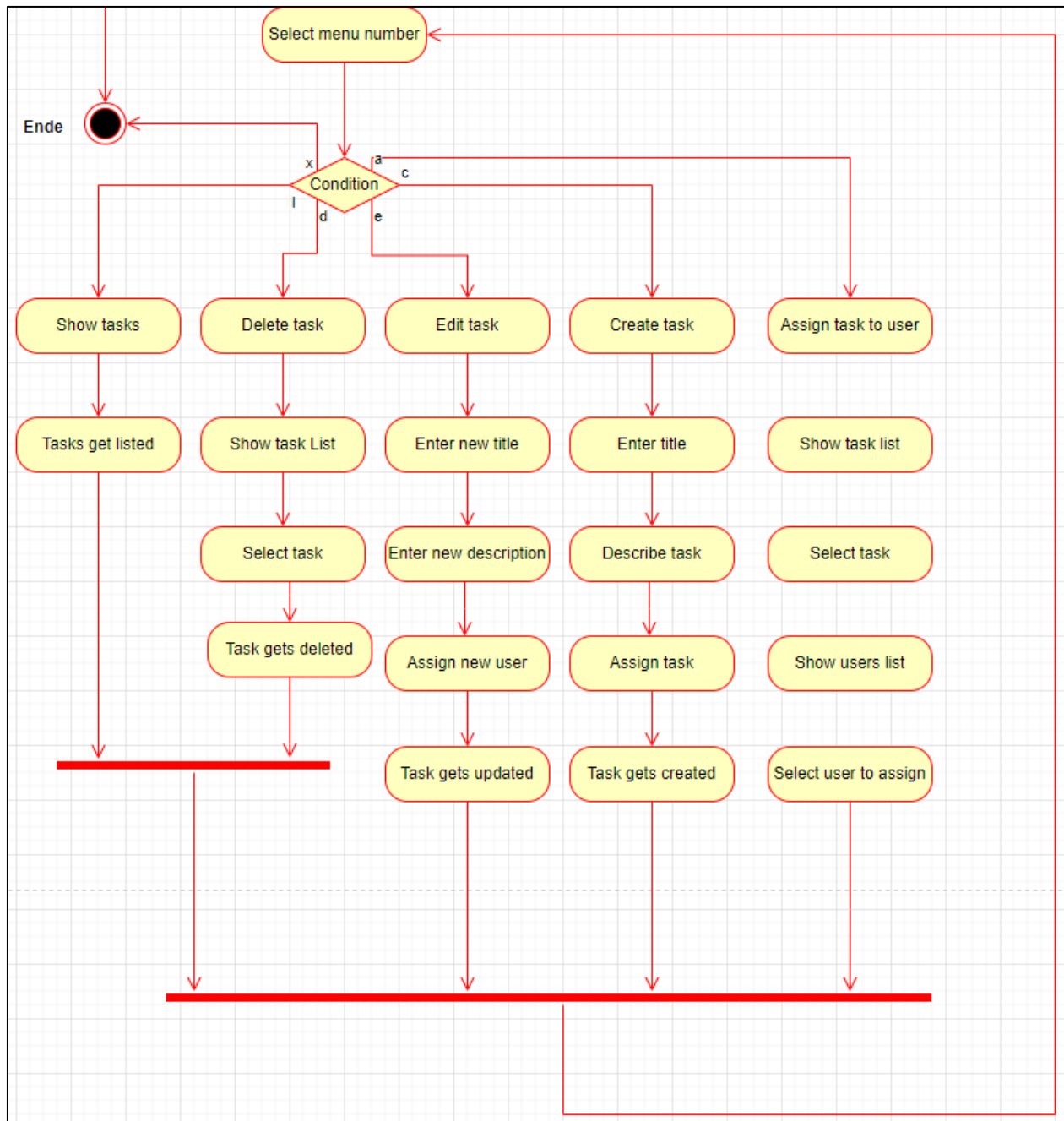


4 Aktivitätsdiagramm

Wenn das Programm gestartet wird, werden zwei Optionen angezeigt. Mit «e» kann man sich einloggen, stimmen die User und das Passwort überein, wird das Menü angezeigt. Mit «x» kann man das Programm direkt beenden.



Nach einem erfolgreichen Login wird das Menu mit den folgenden Optionen angezeigt.



5 JDK

Für dieses Projekt haben wir Java JDK 16.0.2 verwendet.

6 IDE

Als IDE haben wir IntelliJ IDEA Ultimate verwendet.

IntelliJ IDEA ist eine integrierte Entwicklungsumgebung des Softwareunternehmens JetBrains für die Programmiersprachen Java, Kotlin, Groovy und Scala. Es existieren ab der Version 9.0 zwei verschiedene Editionen, die kostenpflichtige Ultimate Edition und die kostenfreie Open Source Community Edition.

[IntelliJ IDEA – Wikipedia](#)

7 Gradle

Für unser Projekt haben wir das Build-Management-Automatisierungs-Tool Gradle verwendet.

Achtung bei Gradle müssen die Dependencis hinzugefügt werden, um den Code zu kompilieren!

Gradle ist ein auf Java basierendes Build-Management-Automatisierungs-Tool, vergleichbar mit Apache Ant und Apache Maven. Gradle nutzt eine auf Groovy basierende domänenspezifische Sprache zur Beschreibung der zu bauenden Projekte. Im Gegensatz zu Maven-Projektdefinitionen sind Gradle-Skripte direkt ausführbarer Code.

[Gradle – Wikipedia](#)

8 Vererbung und Abstrakte Klasse

Bereits im Klassendiagramm erwähnt, haben wir die abstrakte Klasse Task erstellt. Diese vererbt in die Subklassen AssignedTask und PrivateTask. Untenstehend sind die Screenshots des Codes eingefügt.

8.1 Abstrakte Klasse

```
/**
 * Abstrakte Klasse Task
 * Vererbung zu AssignedTask und PrivateTasks
 */
public abstract class Task {
```

8.2 Vererbung

```
public class AssignedTask extends Task {
    @Getter
    @Setter
```

```
public class PrivateTask extends Task{
    private String Star;
```


9 JavaDoc

Für den Zusatzpunkt haben wir das Kommentarschema angewendet.

```
/**
 * Diese Methode returned ob ein User eingeloggt ist
 * @param userId Das ist die Userid vom Benutzer.
 * @return Return true wenn der User eingeloggt ist.
 */
```

10 Anwendung

Mit folgenden Benutzernamen und Passwörtern kann man auf die Applikation zugreifen.

Benutzernamen	Passwörter
alex	ziegler
julian	kuehne
test	test

11 Reflexion

Dieses Modul ist uns schwerer gefallen als andere. Beim 226a bekamen wir Hilfe vom Betrieb und konnte auch auf Arbeitszeit daran arbeiten. Des Weiteren hatten wir beide während der Programmierphase viel im Betrieb zu tun. Die IPA rückt immer näher und wir beide führten im Betrieb in dieser Zeit ein Probe-IPA durch. Daher fehlte uns die Zeit, sowie die Kapazität und die Motivation für dieses Modul Herzblut zu investieren. Wir haben trotzdem versucht, alles zu geben, um ein möglichst gutes Endprodukt zu erhalten. Auch wenn das Programm nicht perfekt ist, sind wir mit unserer Arbeit zufrieden.

Die Zusammenarbeit ist in der Schule ist, wie üblich in diesem Team sehr gut gelaufen. Wir stehen uns nicht nur in der Schule nahe, sondern auch im privaten. Aus diesem Grund kennen wir die Gegenseitigen Vorlieben und Arbeitsweisen. Wir haben uns gut abgesprochen und organisiert. Da es unser letztes Modul an der TBZ ist, wurden wir mit Motivationsproblemen konfrontiert. Jedoch haben wir bei unseren Online-Zusammentreffen darauf geachtet, dass wir die Kamera stetig anhatte und somit bei Betracht unserer Gesichter uns gegenseitig mit Glücksgefühlen überfluten konnten. Dies resultierte in einem gewaltigen Motivationsboost.