Dokumentation - Java 1 Projekt - Appointments

Im Rahmen des Projektes Java 1 ist den Studierenden des 3. Fachsemesters der Angewandten Informatik die Aufgabe gestellt worden die bis dato bestehende App der Fachhochschule Erfurt, die in der bisherigen Funktionalität und Benutzerfreundlichkeit nicht wenige Wünsche offen lässt, nachzubilden, sowie zu verbessern und ggf. zu erweitern.

Dafür wurden 2 Teams gebildet, die konkurrierend jeweils ein gemeinsames Projekt erarbeiten.

Einzelne Bestandteile dieses Projekts wurden in sog. Services, 7 insgesamt, aufgeteilt, die jeweils kleineren Gruppen zugewiesen wurden. Diese hatten jeweils die Aufgabe u.a. die Logik und das Datenmodell für eine Java-basierte Serveranwendung (Service) zu entwickeln.

Unsere Gruppe war für den Service Appointments zuständig.

Arbeitskürzel des Serviceteils des Gesamtprojekts:

WS2021_Java_Team_1_Service_6_Appointments

Projektteam

Jonas Helmboldt (Dev)
Stephan Teichmüller
Nadine Hütter
Artur Jadranski

Anforderungsbeschreibung

Die bestehende Art und Weise des Terminplans soll in diesem Projekt neu ausgearbeiteten und ersetzt werden. Geplant ist ein allgemeiner Terminplan zum Anzeigen, Erstellen, sowie Bearbeiten von Terminen. Mitarbeiter in der Rolle des Terminerstellers sollen Termine erstellen können und diese veröffentlichen. Darüber hinaus soll bei Erstellung dem Termin eine Auswahl an Studenten zugeordnet werden. Die hier partizipierenden Studenten sollen dann über den Termin mit einer Nachricht informiert werden können. Die Bearbeitung der eigens erstellten Termine soll Fähigkeit des Terminerstellers sein. Ebenso soll nur der Terminersteller einen Termin löschen können.

Die Studenten wiederrum müssen nach den relevanten Terminen suchen, die erstellten relevanten Termine sehen, diese filtern und sortieren können. Bei Bedarf sollen sie auch über einen Termin bei Veröffentlichung informiert werden.
* Mitarbeiter sollen Termine erstellen können

* Mitarbeiter sollen Termine veröffentlichen können
* Mitarbeiter sollen Termine bearbeiten können
* Mitarbeiter sollen Termine löschen können
* Terminen wird bei Erstellung eine Wiederholrate zugeordnet
* Terminen wird bei Erstellung ein Veranstaltungsort zugeordnet
* Terminen wird bei Erstellung bei Bedarf eine Liste an partizipierenden Studenten zugeordnet
* Studenten können sich alle bestehenden relevanten Termine/Veranstaltungen ihrer Fakultät/Fachrichtung anzeigen lassen
* Studenten können sich alle relevanten Informationen und Zeiträume ihres Semesters anzeigen lassen
* Studenten können Termine nach Datum, Fakultät, Campus und Ersteller sortieren
* Studenten können Termine nach Datum, Fakultät, Campus, Ersteller, Terminname filtern

---

Aufsetzung und Nutzung des Programms

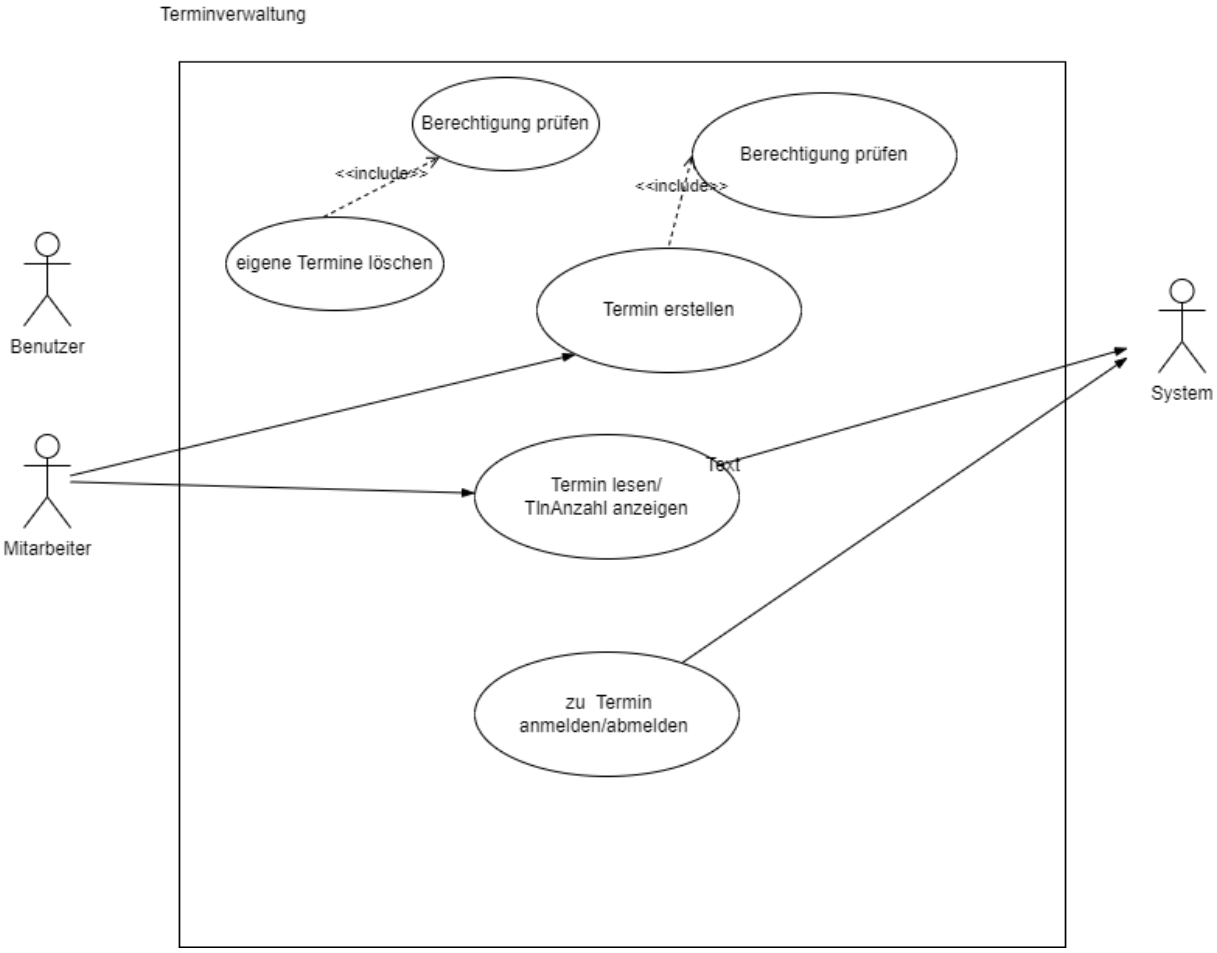
Da der Projektumfang sich zuerst auf die Datenstruktur und Logik des Services beschränkt, existieren sowohl keine reale Datenbank, als auch eine interaktive Benutzeroberfläche jeglicher Art.
Die Fähigkeit und die Notwendigkeit einer Aufsetzung des Programms wird Teil von Java 2 sein.

Eine Möglichkeit der Interaktion wird ebenfalls Teil von Java 2 werden.

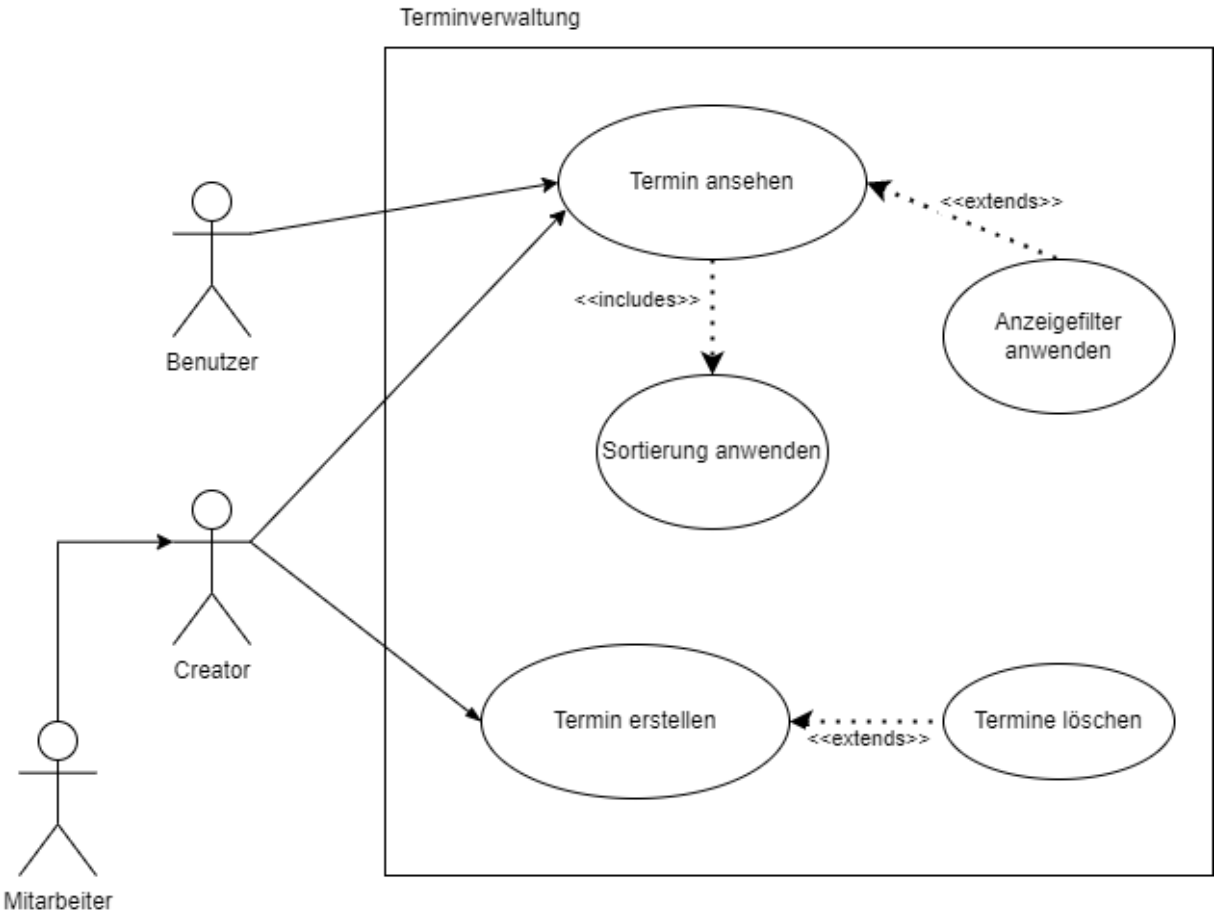
Diagramme

Hier sind zusammengefasst alle bisher erstellten relevanten Diagramme und Schemata, sowie die vorhergehenden Entwürfe.

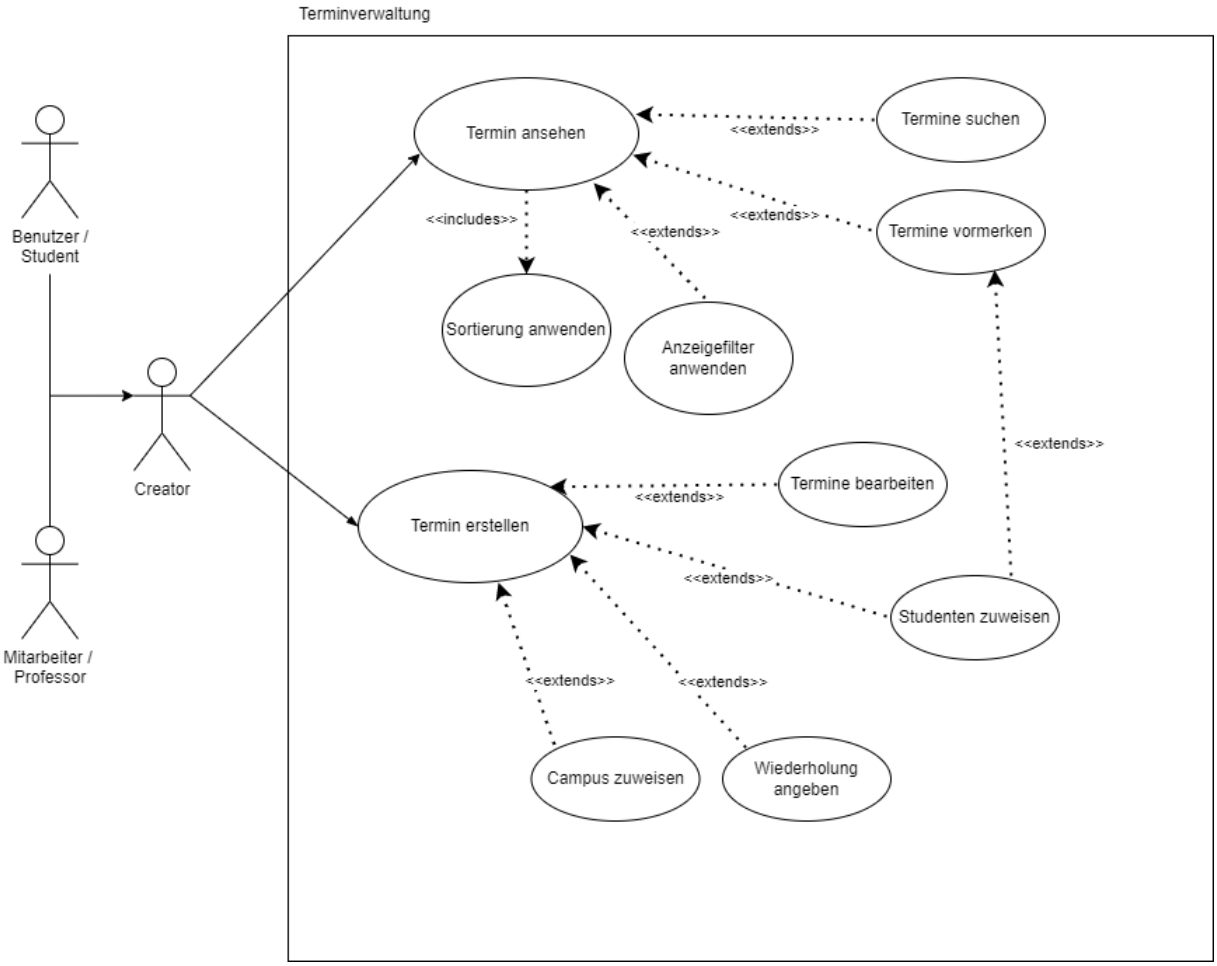
UC



Im ersten Entwurf wurde sich auf die notwendigsten Funktionalitäten beschränkt, die für die Realisierung des Gesamtprojekts sinnvoll wären. Eine grundlegende Terminverwaltung, die im vollen Umfang einem erstellenden Mitarbeiter zugeordnet wird, erschien als notwendig. Darüber hinaus war auch Teil des Entwurfsprozesses die Überlegung, dass eine Art rudimentäres Rechtesystem existieren könne. Hier gäbe es die Möglichkeit die Erstellung und sonstige Verwaltung eines Termins explizit nur Mitarbeitern zuzuordnen. Zusätzlich sollte ein Benutzer sich die bestehenden Termine anzeigen lassen können.

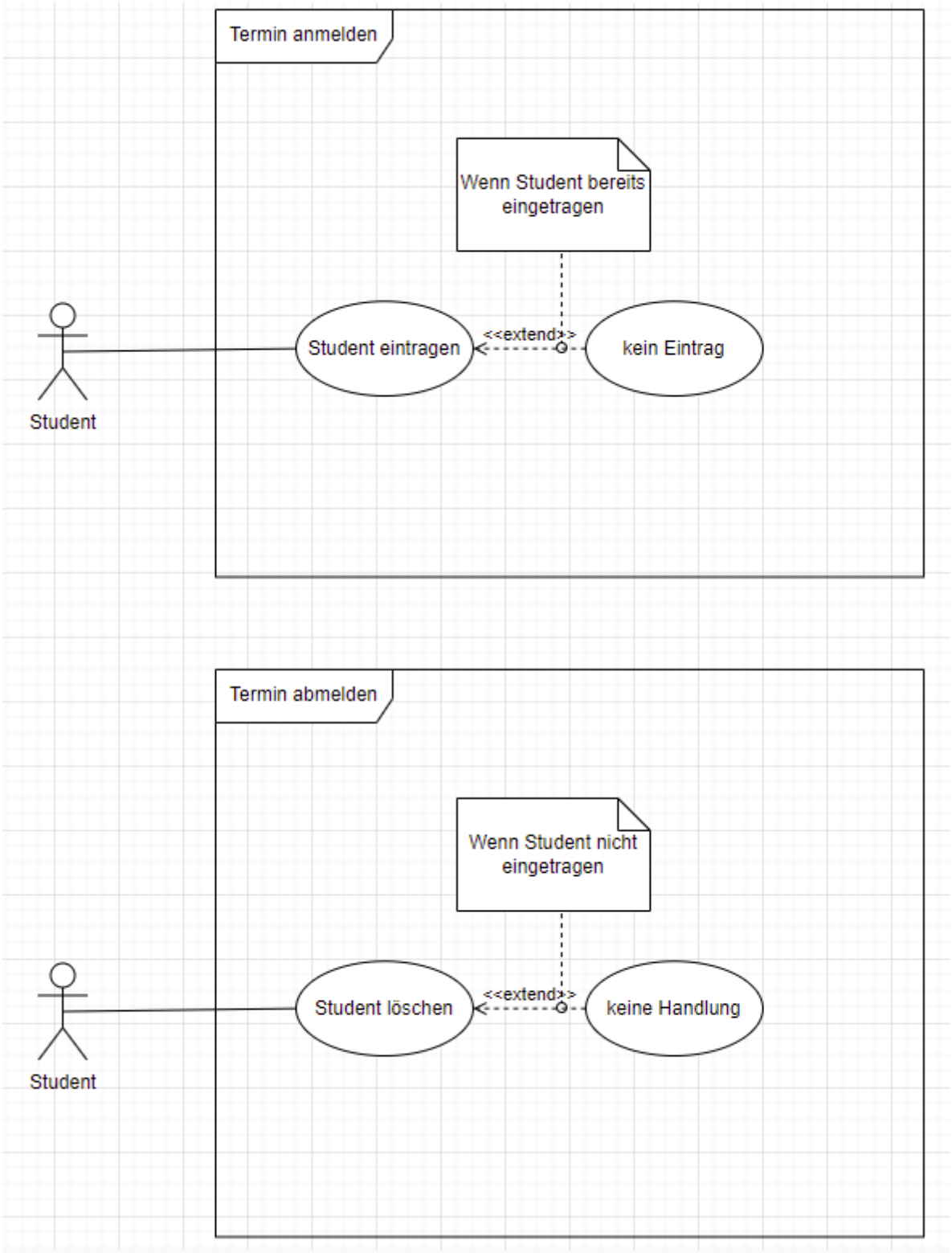


Im Hinblick auf den Projektumfang und die erarbeiteten Entwürfe anderer Services wurde die Realisierung des Service Appointments mit den hier gezeigten Fähigkeiten realisiert. Grundlegend existieren zwei große Rollen. Ein Benutzer, dies sind Studenten. Sowie Mitarbeiter, zu diesen zählen auch Professoren. Teil der Mitarbeiter sind ebenfalls Terminersteller, die einen Termin erstellen und löschen können. Zudem können Termine angesehen und sortiert/gefiltert werden können. Benutzer können wiederum nur Termine ansehen, jedoch nicht erstellen oder löschen.

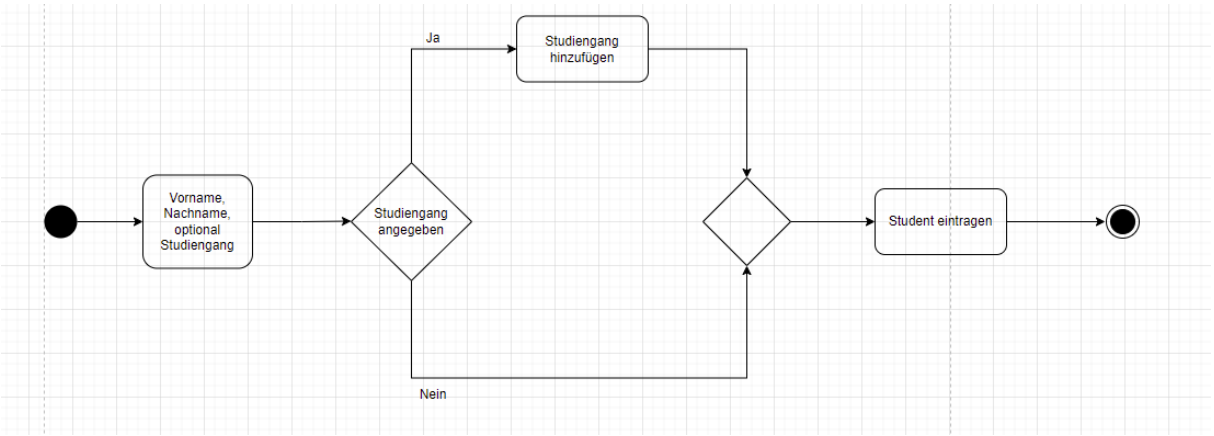


Dies ist das finale Use Case des Service Appointments.
Benutzer und Mitarbeiter sind vorerst über eine gemeinsame, Rolle in ihren Berechtigungen, vereinigt.

UML

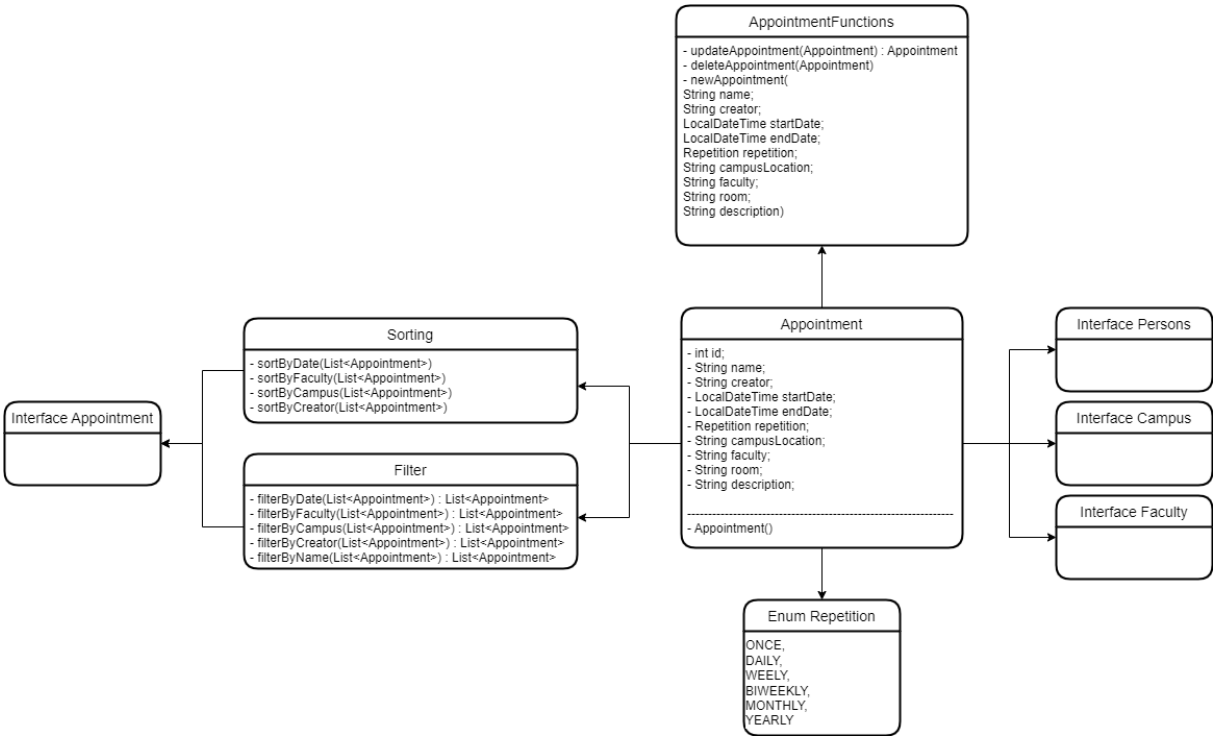


Dies ist der Erste Entwurf einer Prüfungsan- und abmeldung.

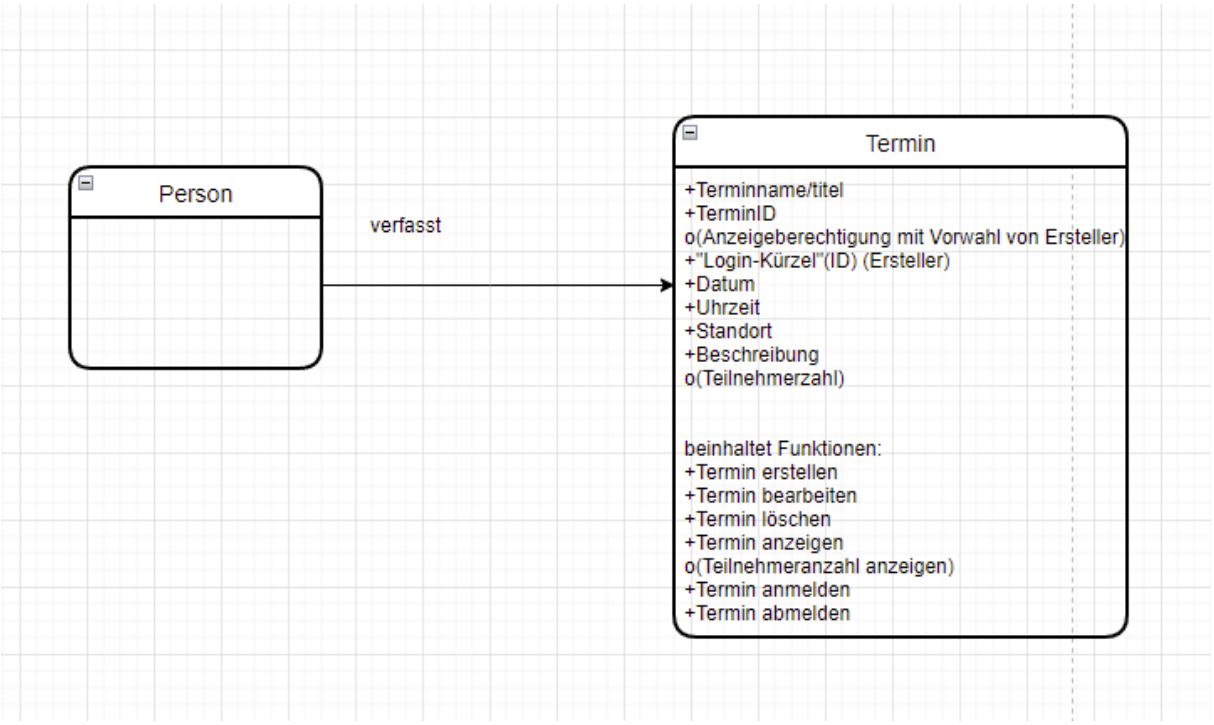


Erste Überlegung zu Eingrenzung der Termine u.a. mit Studienganginformationen.

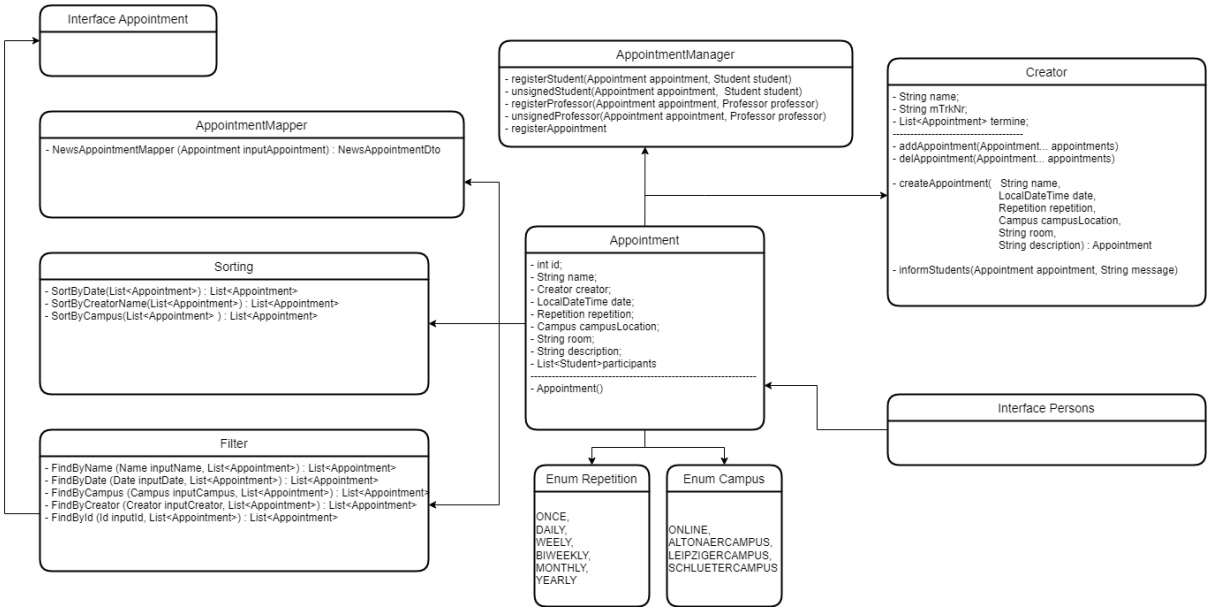
Klassenmodelle



Dies ist der ausgearbeitete Entwurf des Klassenmodells mit den zu implementierenden Funktionen und Klassen.



Hier wird beschreiben, welche relevanten Informationen unser Service Appointments vom Service Persons entgegennehmen soll. Da dies ein Entwurf ist und nicht final, haben wir uns auf die Realisierung unseres Services auf grundlegendste und relevanteste Informationen begrenzt.



Das hier vorliegende Klassenmodell entspricht der finalen Form.

Ideensammlung und Meilensteine

Die Arbeit am Projekt kann in 3 grundlegende Abschnitte eingeteilt werden:

- Entwurfsphase
- Ausbau-/Erweiterungsphase
- Finalisierungsphase

Entwurfsphase

* Erarbeitung des Projekts geplant methodisch wöchentlich durchgeführt
* Sammlung erster Projektideen
* Überlegung des Funktionsumfangs
* Aufteilung Rollen in Student <=> Mitarbeiter
* Überlegung relevanter Zuarbeiten von / zu anderen Services
* Schnittstelle zu Datenbank (provisorisch wird im Weiteren ausgetauscht)
* Checkfuntion ob Termin vorhanden
* Erstellung einer Präsentation über unsere Gruppe und den zu erstellenden Service
* erste Präsentation

Ausbau-/Erweiterungsphase

* Erstellung eines Repository in Github, sowie erste Konfiguration
* erstes Erstellen einer Ausgangssituation:
    * Überlegung: die Art und Weise der Terminverwaltung:
        * benötigte weitere grobe Differenzierung Studenten <=> Dozenten
        * Termine sollten nur von Dozenten erstellt, bearbeitet und gelöscht werden können
        * Dozenten sollten auch nur ihre eigenen Termine verwalten können
        * eine Art Superrolle für allgemeine Organisation der Termine könnte ebenfalls im weiteren Projektverlauf implementiert werden
        * Studenten sollen allgemein nur für sie relevanten Termine angezeigt werden
        * Möglichkeit der Terminsuche soll auf alle Termine ausgeweitet werden
* Erstellung erster Entwürfe zum Use Case

* Erste Überlegungen zum Klassenmodell
* Erste Überlegungen zu Funktionalitäten:
* Bereitstellung des Appointment-Interfaces mit grundlegenden Funktionalitäten
* Termine ausgeben / exportieren
* Termine filtern und sortieren
* Ausarbeitung benötigter Informationen für das Anlegen von Terminen aus anderen Services
* Standort (Campus)
* Betreffende Fakultät (Faculty)
* Ersteller (Mitarbeiter) verknüpfen mit >Persons<

Finalisierungsphase

* Erstellung einer gemeinsamen Rolle >Creator<
* Zusammenführung der Berechtigungen und Fähigkeiten
* zweite Präsentation der gesammelten Ideen
* weitere Ausarbeitung der geplanten Funktionen
* Klassenmodelle sind erstellt
* Ordnerstruktur implementiert
* Funktionalitäten erstellt und getestet
* Beginn der Implementierung
* erneute Absprachen mit anderen Services
* Erstellung und Implementierung von Tests
* Fehlerbehebung

Verwendete Programme

- Intellij - IDE für Java
- draw.io - für Diagramme
- SharePoint - vorläufige Dokumentation, Präsentationen & Veranschaulichung, Projektplanung und Aufgabenverteilung
- Github - Versionskontrolle, Projektplanung und Aufgabenverteilung
- Discord - Kommunikationsmittel

Mit diesem Projekt soll eine solide Grundlage für das hierauf aufbauende- Java 2, werden.