



Student self-organization



Gliederung

1. Mitglieder
2. Service
3. Aktueller Stand des Projektes
4. Bisher erreichte Ziele/Meilensteine
5. Probleme (Java 1)
6. Lessons learned (Java 1)
7. Bisherige Diagramme



Mitglieder

- Felix Zwicker (DevLead)
- Friedemann Taubert
- Jenny Wagner
- Tim Eisenberg
- Maximilian Keller



Service

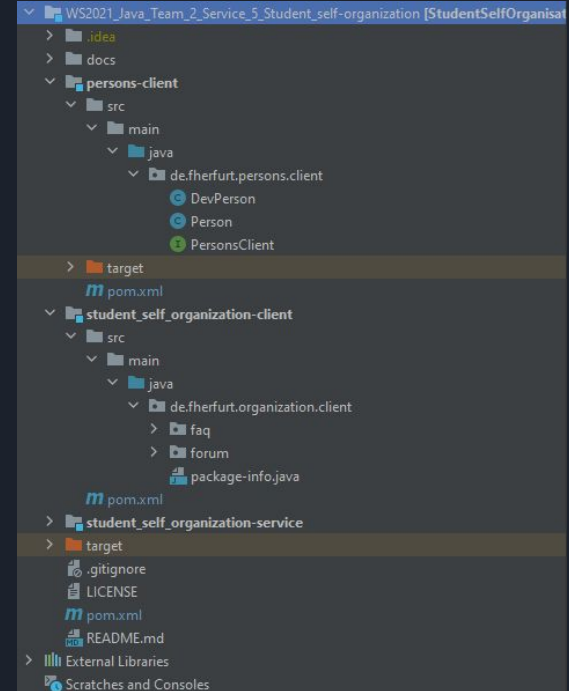
Team 2, Service 5

- beinhaltet Forum, FAQ, To-do-Liste und Infoseite
- Forum:
 - Studenten oder Dozenten können Fragen erstellen
 - andere Nutzer können antworten
- FAQ:
 - wichtigen und oft gestellten Fragen in einer Übersicht
- To-do-Liste:
 - organisatorische Rahmen für das Managen aller Aufgaben eines Studenten
 - Aufzählung von mehreren Tätigkeiten, die der Nutzer selbst hinzufügen, bearbeiten und löschen kann
- Infoseite:
 - zeigt allgemeine Informationen zu Gremien oder StuRa
 - beinhaltet Kontaktpersonen vom Service Persons

Aktueller Stand des Projektes

Gesamtes Projekt:

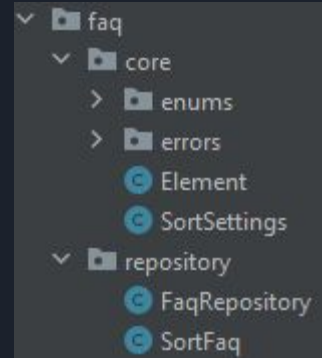
- Projekt unterteilt in drei Module
 - persons-client
 - organization-client
 - organization-service
- Person-Client: Schnittstelle zu Service 2
- Organization-Client: stellt Service anderen zur Verfügung
- Organization-Service: Logik und Verwaltung des Service



Aktueller Stand des Projektes

FAQ:

- besteht aus den elementaren Klassen:
- Element: konstruiert ein Element des FAQ's mit Title, Content...
- SortFaq: nutzt Comparator um FAQ Liste zu sortieren
- FaqRepository:
 - alle FAQ-Elemente werden in einer Liste gespeichert
 - Elemente können hinzugefügt, entfernt und gesucht werden

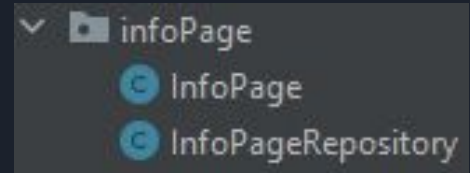




Aktueller Stand des Projektes

Infoseite:

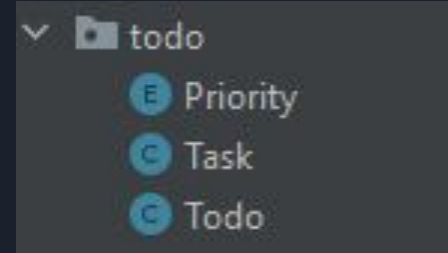
- Infoseite beinhaltet die Klassen InfoPage und InfoPageRepository
- InfoPage: konstruiert eine Infoseite mit Liste von Personen
- Repository: verwaltet erstellte InfoPage



Aktueller Stand des Projektes

ToDo: zwei Klassen und ein Enum

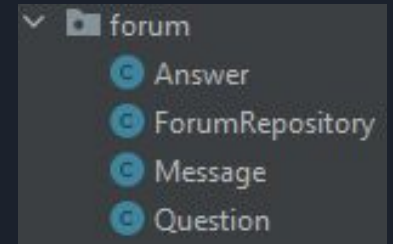
- Priority: beinhaltet Prioritätsstufen “highly important”, “important”, “insignificant”
- Task: Definiert eine Aufgabe in der Liste mit den Variablen title, date, isChecked, taskId, priority
- Todo: beinhaltet Methoden zum
 - erstellen der Todo-Liste
 - anzeigen der Todo-Liste
 - hinzufügen einer Aufgabe
 - suchen einer Aufgabe
 - löschen einer oder aller Aufgabe/n
 - abchecken einer Aufgabe



Aktueller Stand des Projektes

Forum:

- aus vier Klassen aufgebaut: Answer, ForumRepository, Message, Question
- Answer: zeigt die Antworten zu einer Forum Frage
- ForumRepository: beinhaltet Methoden (addNewQuestion, removeQuestion, getByTopic, etc.)
- Message: dient als Datencontainer für eine Standardnachricht
- Question: stellt die Frage eines Forums dar, beinhaltet Antwort und Topic beinhaltet Methode: removeAnswer





Bisher erreichte Ziele/Meilensteine

- Grundstrukturen der Programmiersprache Java mit der Entwicklungsumgebung IntelliJ wurden sich individuell angeeignet
- Grundverständnis für Unit Tests wurde erlangt
- Umgang mit GitHub und Maven wurde im Laufe des letzten Semesters immer organisierter und einfacher
- Aufgabenteilung und -erfüllung innerhalb des Teams war stets gewissenhaft und zuverlässig
- in Java 1 verlangten Ziele/Meilensteine wurden erfolgreich umgesetzt



Probleme (Java 1)

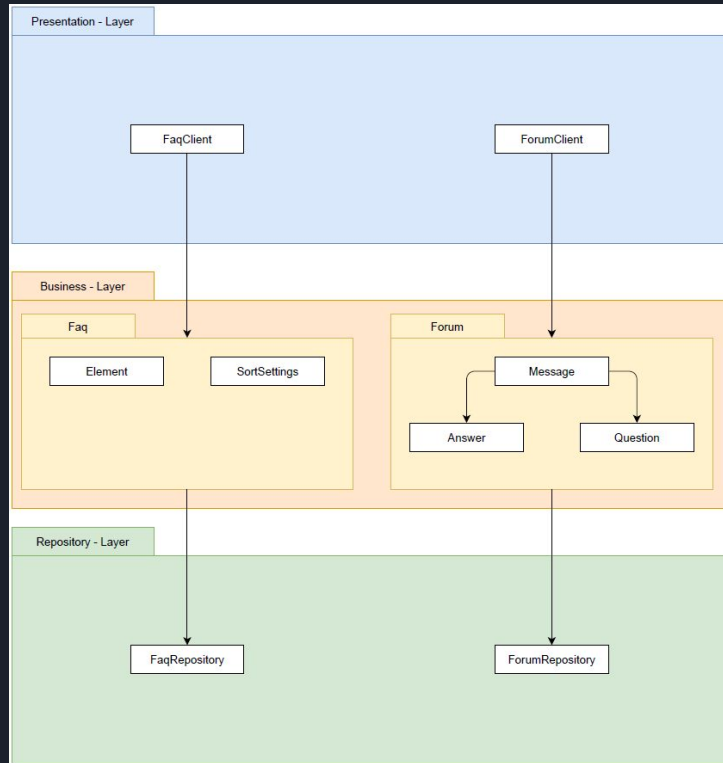
- Genaue Anforderungen unseres Services
- Zeitfindung unter Berücksichtigung der anderen laufenden Projekte und Klausuren, Zusammenarbeit mit anderen Services für Schnittstellen
- Schnittstelle zwischen GitHub und IntelliJ IDEA
- Umsetzung der Speicherung und Darstellung von Einträgen eines Forums
- Kennenlernen und Umgang mit der Programmiersprache Java



Lessons Learned (Java 1)

- problematisch sich bei mangelnden Vorkenntnissen, beispielsweise mit GitHub oder IntelliJ, in den Workflow der Gruppe einzubringen -> alle müssen auf demselben Stand sein
- Wichtigkeit von Termineinhaltung
- die Arbeitsweise sollte sich mehr anhand der im GitHub erstellten To-Do-Listen orientieren
- von vornherein geplanter Arbeiten und sich nicht zu viele Aufgaben vornehmen

Bisherige Diagramme



Top-Down-Sicht

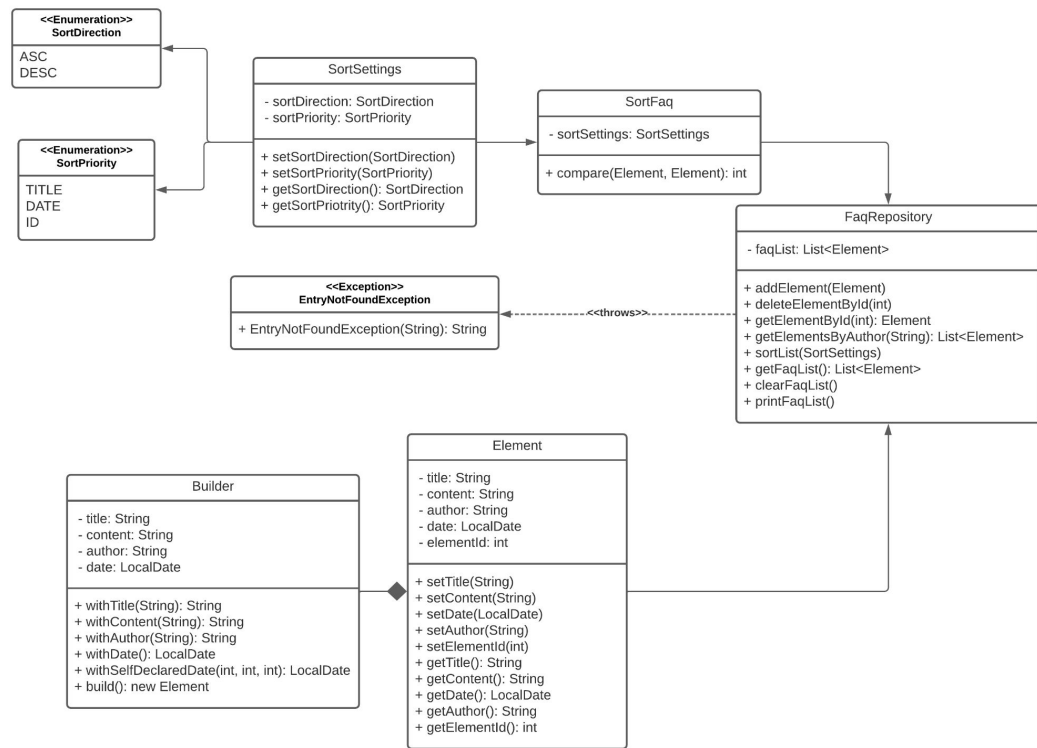
Forum
- topics : ArrayList<String>
- questions : ArrayList<Question>
+ addNewQuestion (...) : void
+ removeQuestion (title) : void
+ getByTopic (topic) : ArrayList<>
+ getbyQuestionTitle (question) : ArrayList<>
+ getByQuestionAuthor (author) : ArrayList<>
+ addNewTopic (newTopic) : boolean
+ removeTopic (topic) : boolean

Question
- message : Message
- topic : String
- answers : ArrayList<Answer>
+ addAnswer (...) : boolean
+ removeAnswer (title) : boolean

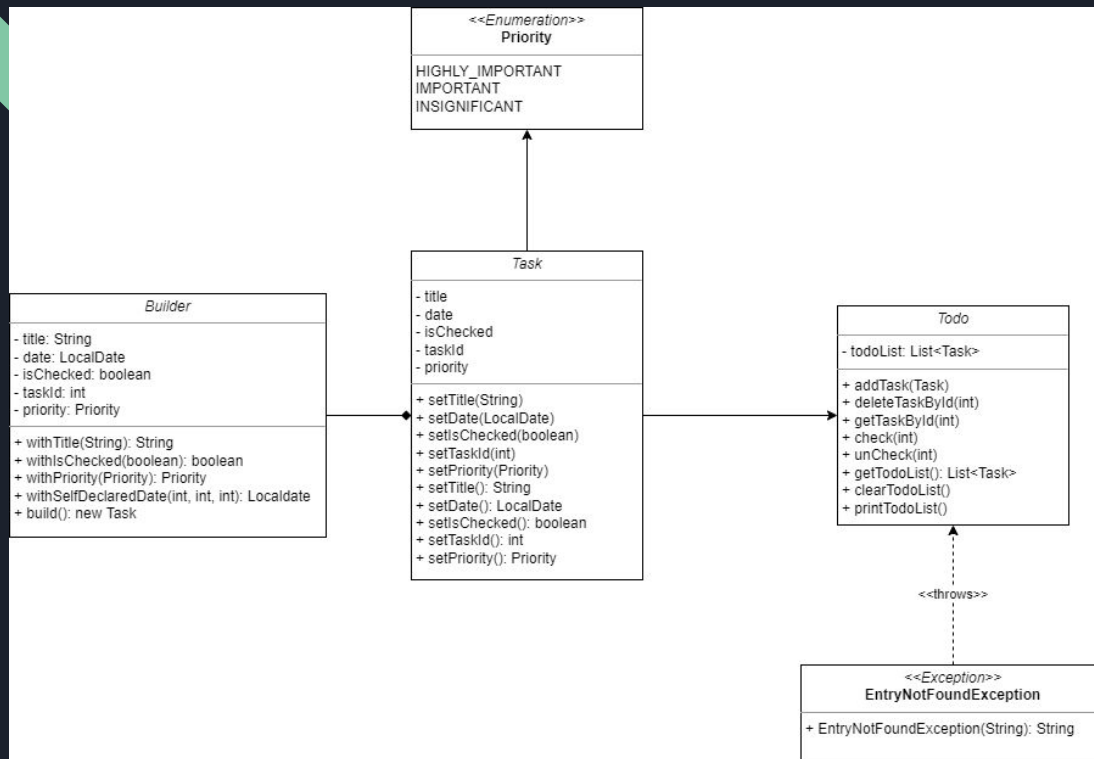
Answer
+ message : Message
+ method(type): type

Message
- title : String
- text : String
- author : String
+ method(type): type

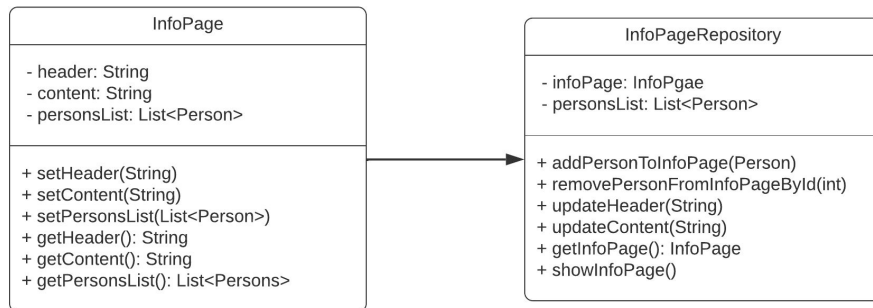
Forum



FAQ



Todo



Info-Page



Vielen Dank für eure Aufmerksamkeit