

Meilenstein 3

Software Architektur

Gruppe 03

Martina Rutsch
Samira Salija
Markus Tal
Mohamed Benbelkacem
Max Böttinger
Ardian Fejzullahu

main ▾

1 branch

0 tags

Go to file

Add file ▾

Code ▾

maxboettinger Update README.md	78aa441	7 days ago	4 commits
.vscode	init commit	7 days ago	
public	init commit	7 days ago	
src	init commit	7 days ago	
.gitignore	init commit	7 days ago	
README.md	Update README.md	7 days ago	
package.json	init commit	7 days ago	
yarn.lock	init commit	7 days ago	

README.md

Software Projekt HdM WS20

Martina Rutsch, Samira Salija, Markus Tal, Mohamed Benbelkacem, Max Böttinger, Ardian Fejzullahu,

About

No description, website, or topics provided.

[Readme](#)

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

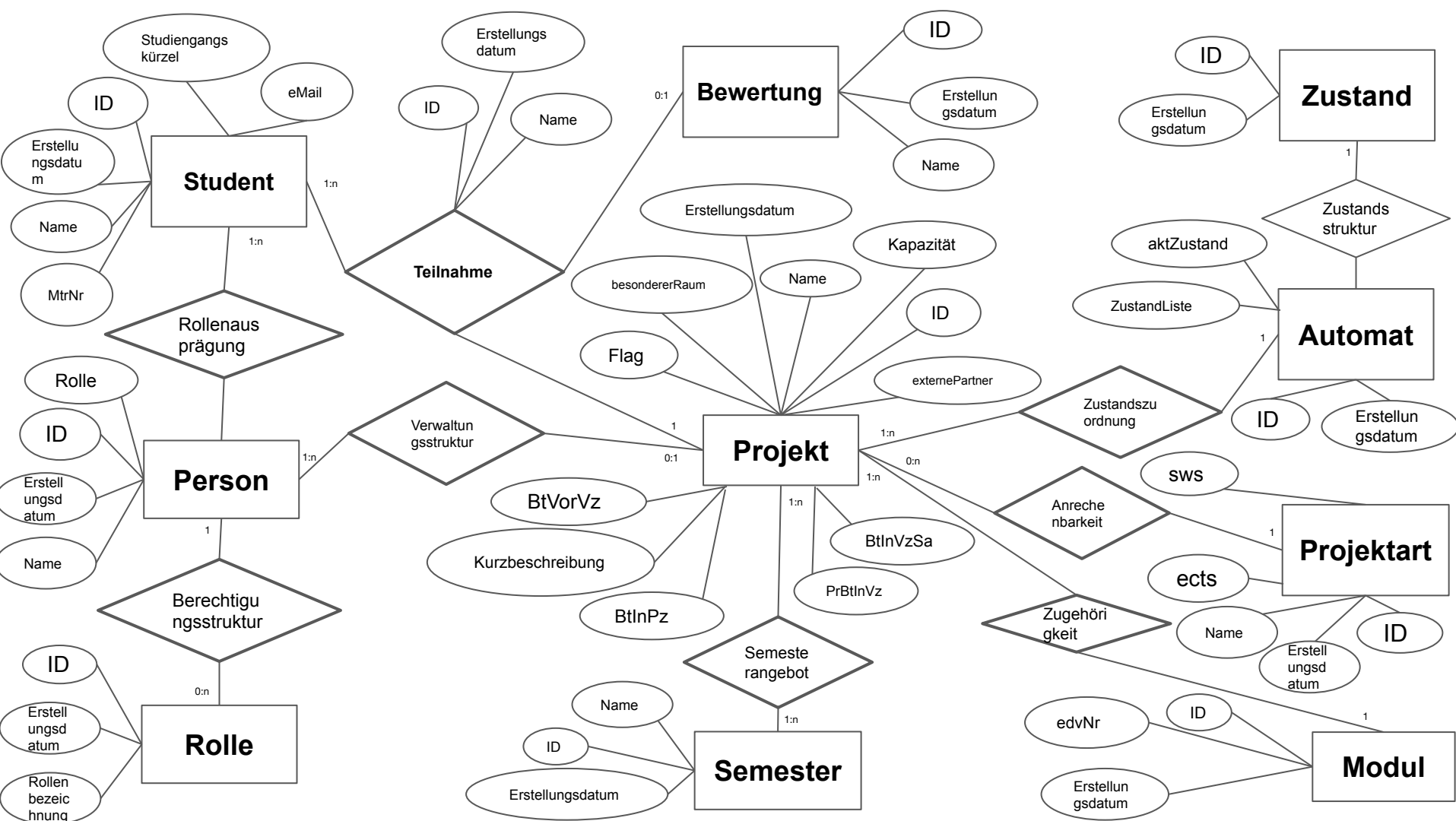


<https://github.com/maxboettinger/hdm-projektverwaltung>



Verantwortlichkeiten

Präsentationsschicht	Max, Ardian
Applikationsschicht	Samira, Martina
Datenbankschicht	Mohamed, Markus
Dokumentation	alle Gruppenmitglieder



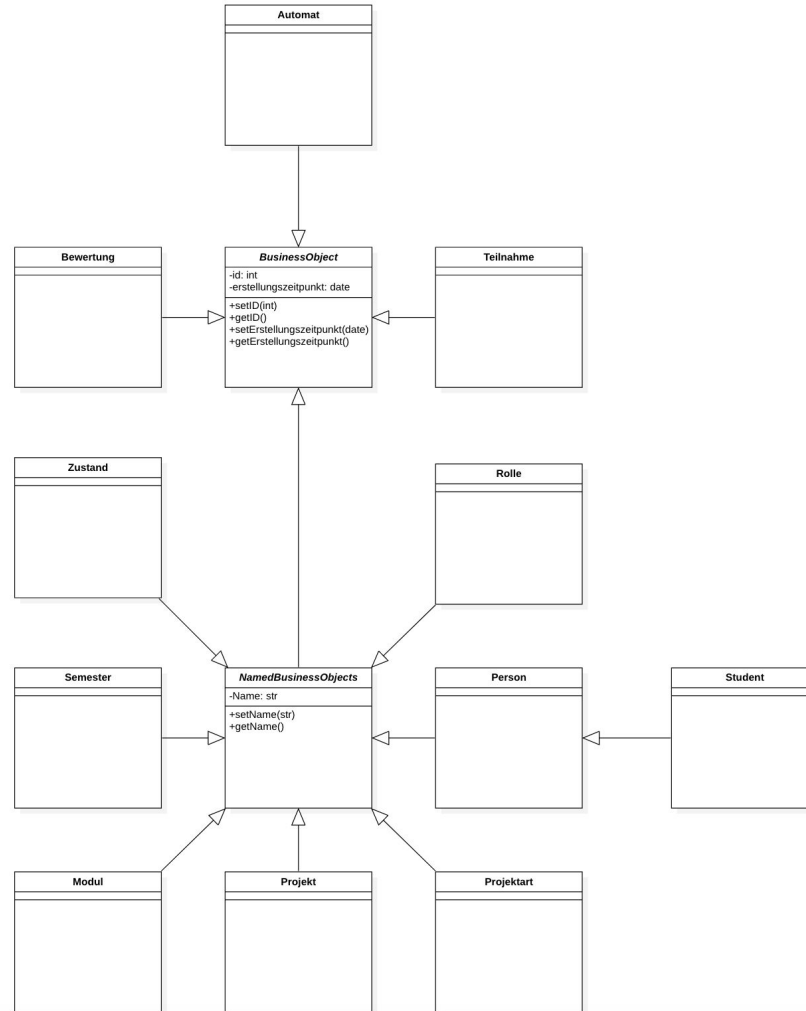
BusinessObjects und NamedBusinessObjects

Superklassen

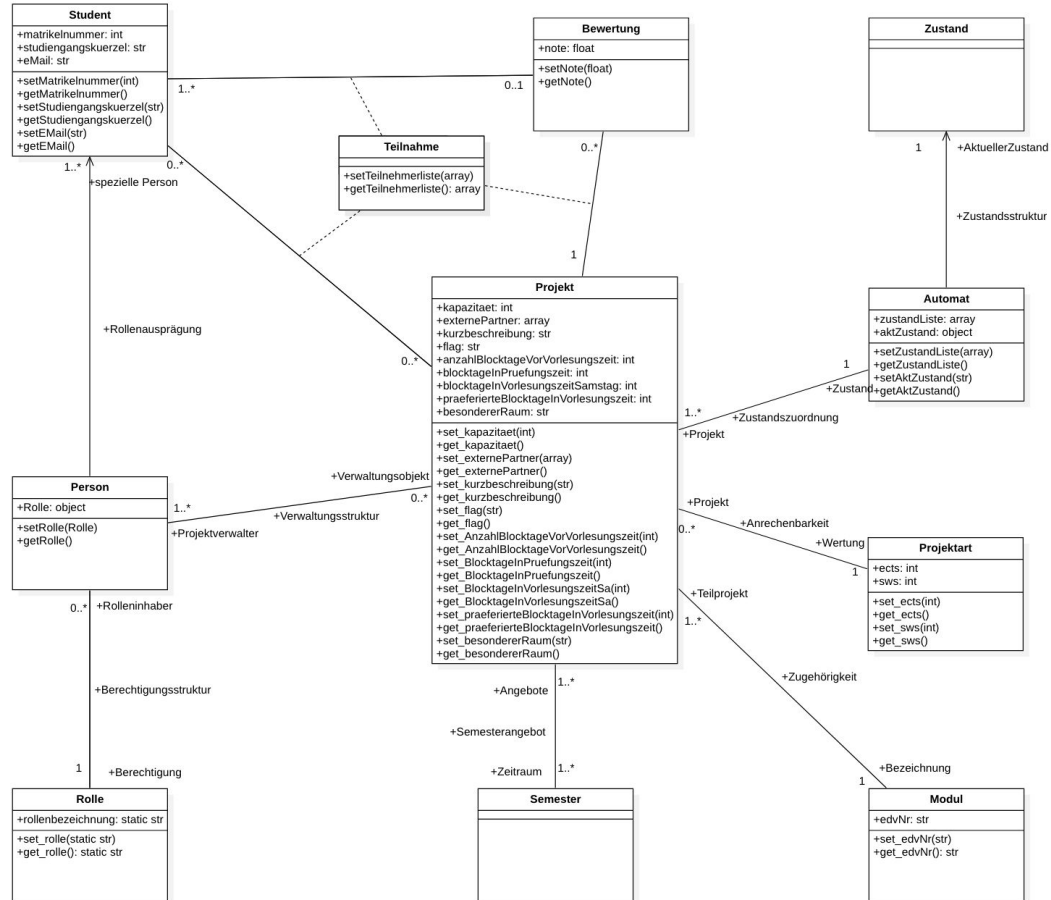
- BusinessOb.py und NamedBusinessOb.py sind die Superklassen der Klassen in unserem System.
- Die Subklassen von BusinessOb.py und NamedBusinessOb.py repräsentieren reale Objekte
- Alle BusinessObjects und NamedBusinessObjects haben als Primary Key das Attribut ID. Darüber hinaus dienen sie der eindeutigen Identifikation in der relationalen Datenbank.

Business Objects

Superklasse

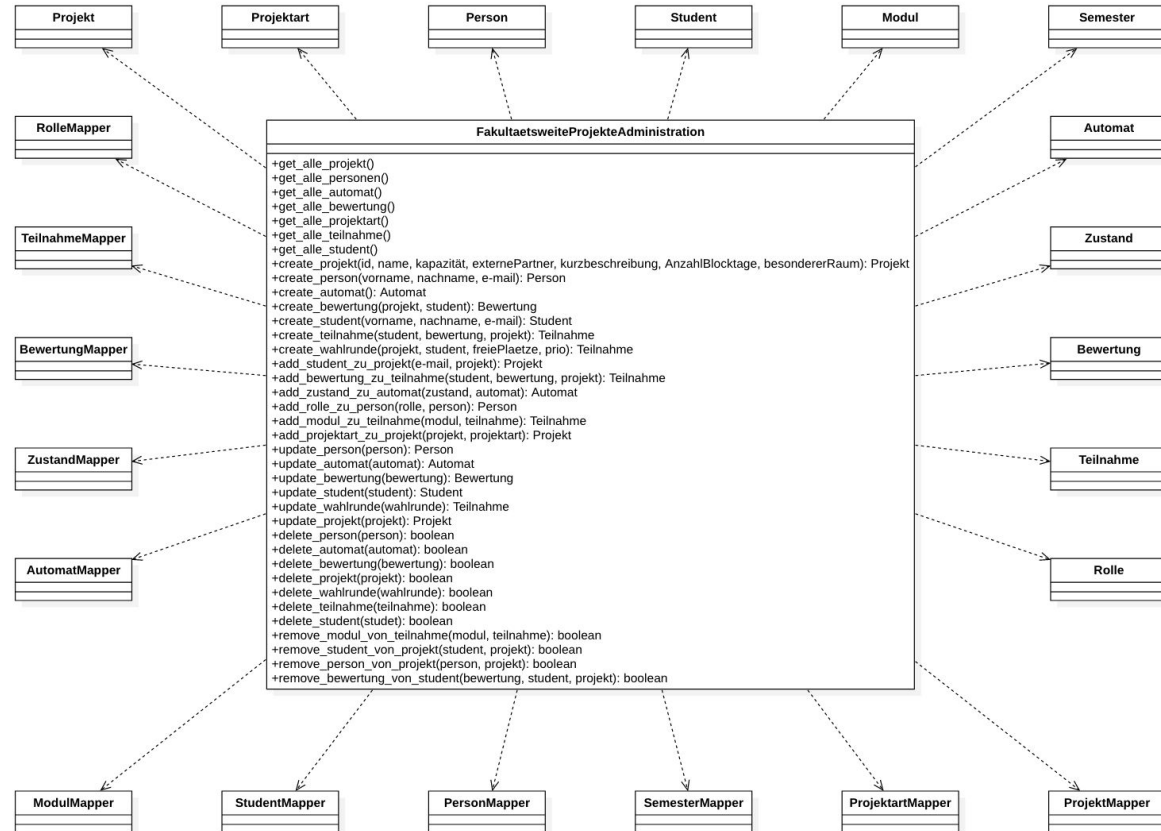


Subklassen



Remote Procedure Calls

Administration der fakultätsweiten Projekte



Remote Procedure Calls

Administration der fakultätsweiten Projekte

- Die Klasse **fakultaetsweiteProjektAdministration** zeigt die gesamte Applikationslogik des Systems. Es zeigt die relevanten BusinessObjects und NamedBusinessObjects, welche zum Erstellen, Bearbeiten, Entfernen und Löschen benötigt werden. Es bestehen Zusammenhänge zwischen den Mapper-Klassen und den BusinessObjects bzw. NamedBusinessObjects.
- Die **get-Methode** beschreibt wie Daten der BusinessObjects bzw. NamedBusinessObjects zurückgegeben werden.
- Die **create-Methode** beschreibt wie ein BusinessObjects bzw. NamedBusinessObjects erstellt wird.
- Die **update-Methode** definiert wie einzelne Informationen der BusinessObjects bzw. NamedBusinessObjects verändert werden.
- Die **delete-Methode** beschreibt wie ein BusinessObject bzw. NamedBusinessObjects gelöscht wird.
- Die **remove-Methode** beschreibt ebenfalls wie BusinessObjects bzw. NamedBusinessObjects gelöscht werden, jedoch beschreibt diese Methode wie Abhängigkeiten zwischen BusinessObjects bzw. NamedBusinessObjects aufgelöst werden.

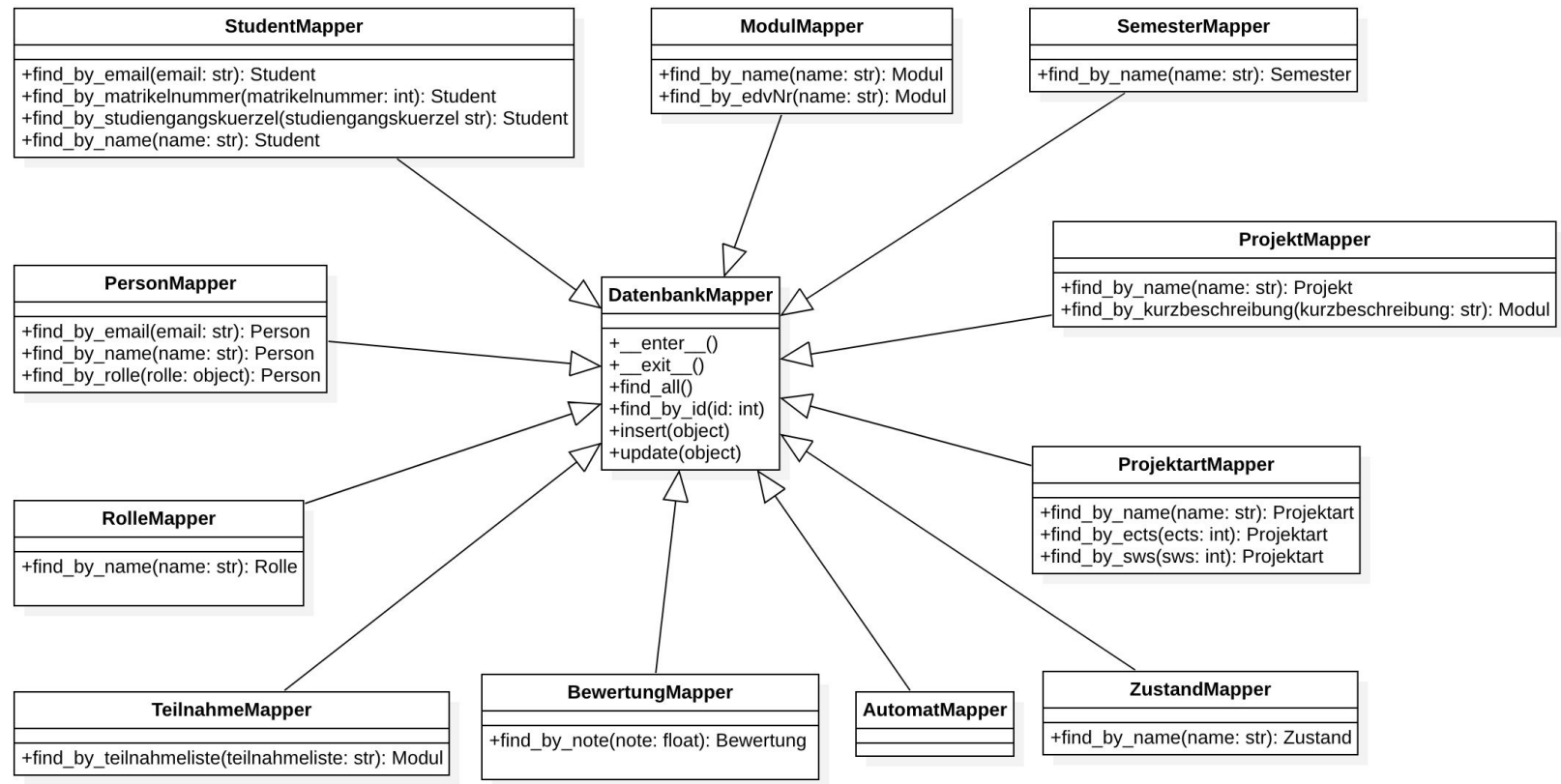
Datenbankschicht

Mapper-Klassen

- 12 Mapperklassen: 1 Superklasse (DatenbankMapper), 11 Subklassen
- Subklassen erben alle Methoden der Superklasse
- Alle Klassen haben als Primary Key IDs, die hier nicht dargestellt sind
- Objekte der Klasse Student und Person lassen sich über Name, Email und Matrikelnummer finden
- Objekte der Klasse Modul können über die edvNummer oder den Namen zugegriffen werden
- Auf Objekte der anderen Klassen kann über den Namen (wenn vorhanden) zugegriffen werden

Datenbankschicht

Mapper-Klassen



Main.py / REST Operations

Das Modul main.py enthält 7 Klassen in der sich die RESTful Services befinden. Durch ihre Methoden **get**, **post**, **put** und **delete** können Nachrichten zwischen Server und Client versendet werden.

Folgende Klassen werden definiert:

- *PersonOperations*
- *ProjektOperations*
- *ProjektartOperations*
- *TeilnahmeProjektOperations*
- *SemesterStudentOperations*
- *ModulOperations*
- *BewertungStudentOperations*
-

Die **Packages** *flask-restx* und *flask-cors* werden zur Durchführung der Vorgänge importiert.

Main.py / REST Operations

