

Zürcher Hochschule für angewandte Wissenschaften  
School of Management and Law

# **Projektdokumentation**

## **StudyBuddy**

Autor: Mustafa Uzdilli, 20HS TZBISa  
uzdilmus@students.zhaw.ch

Modul: Software Engineering 2  
Herbstsemester 2022

# Inhaltsverzeichnis

1 Einleitung (Projektidee).....	1
2 Anforderungen .....	2
2.1 Use-Case Diagramm.....	2
2.2 ER-Modell .....	3
2.3 BPMN-Diagramm .....	4
2.4 UML-Klassendiagramm.....	5
3 Implementation.....	5
3.1 GitHub.....	5
3.2 Technisches Datenmodell.....	7
3.3 Frontend .....	10
3.4 Drittsystem: FormSubmit.co.....	12
4 Testing .....	12
4.1 Modul- und Integrationstests.....	12
4.2 User-Tests .....	14
5 Fazit .....	16

## **1 Einleitung (Projektidee)**

### **Problemstellung:**

Bestimmt wurde schon jeder Studierende mit der Situation konfrontiert, dass man für eine Modulendprüfung lernen muss, es einem jedoch sehr schwerfällt, mit dem Lernen zu beginnen, weil es unzählige andere Tätigkeiten gibt, die einer mehr Spass bereiten als büffeln zu müssen.

Ein weiteres Beispiel könnte sein, dass man begonnen hat sich auf die Prüfung vorzubereiten, jedoch gibt es bestimmte Themen in einem Fach, die man einfach nicht versteht und man wäre dankbar, wenn es eine Person gäbe, die einem genau dieses Thema auf eine verständliche Art und Weise erklären könnte.

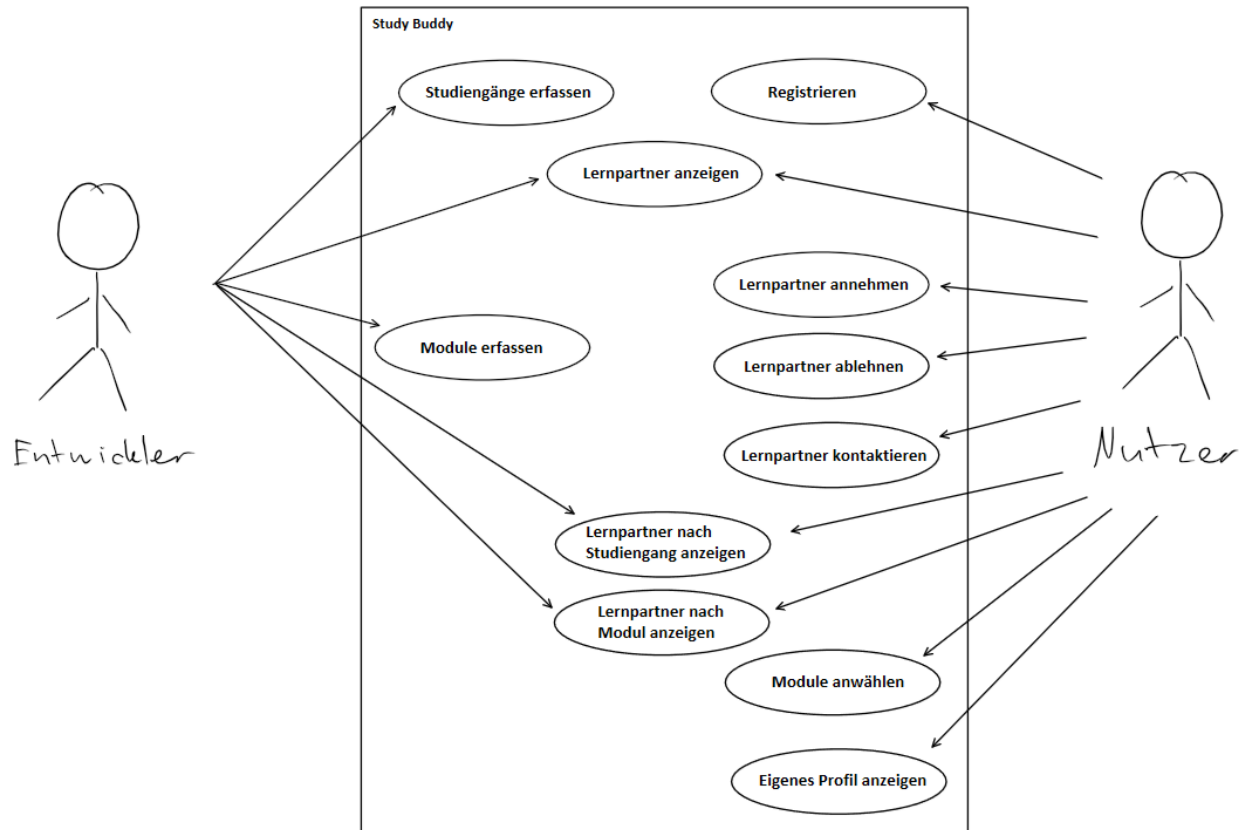
Es könnte auch sein, dass man während dem Lernen alte Prüfungen eines Moduls gelöst hat, jedoch ist einem unklar, ob man es auch richtig gemacht hat, da es keine Lösungen für diese Prüfungen gibt.

### **Projektidee und Benutzergruppe:**

Genau für solche Situationen ist meine Projektidee namens «Study Buddy» gedacht, welches eine sogenannte Lernpartnerbörse darstellen soll. Es ist eine Applikation, welche Studierenden das Lernen einfacher, amüsanter, interessanter gestalten soll, damit sich das Lernen weniger, wie eine Belastung anfühlt. In der Applikation sollen Studierende ihre Ausbildungsstätte und das Modul anwählen können, wofür Sie lernen möchten. Basierend auf den Eingaben soll dann die Applikation passende potenzielle Lernpartner anzeigen, mit dem Sie Kontakt aufnehmen und sich verabreden können um gemeinsam zu Lernen.

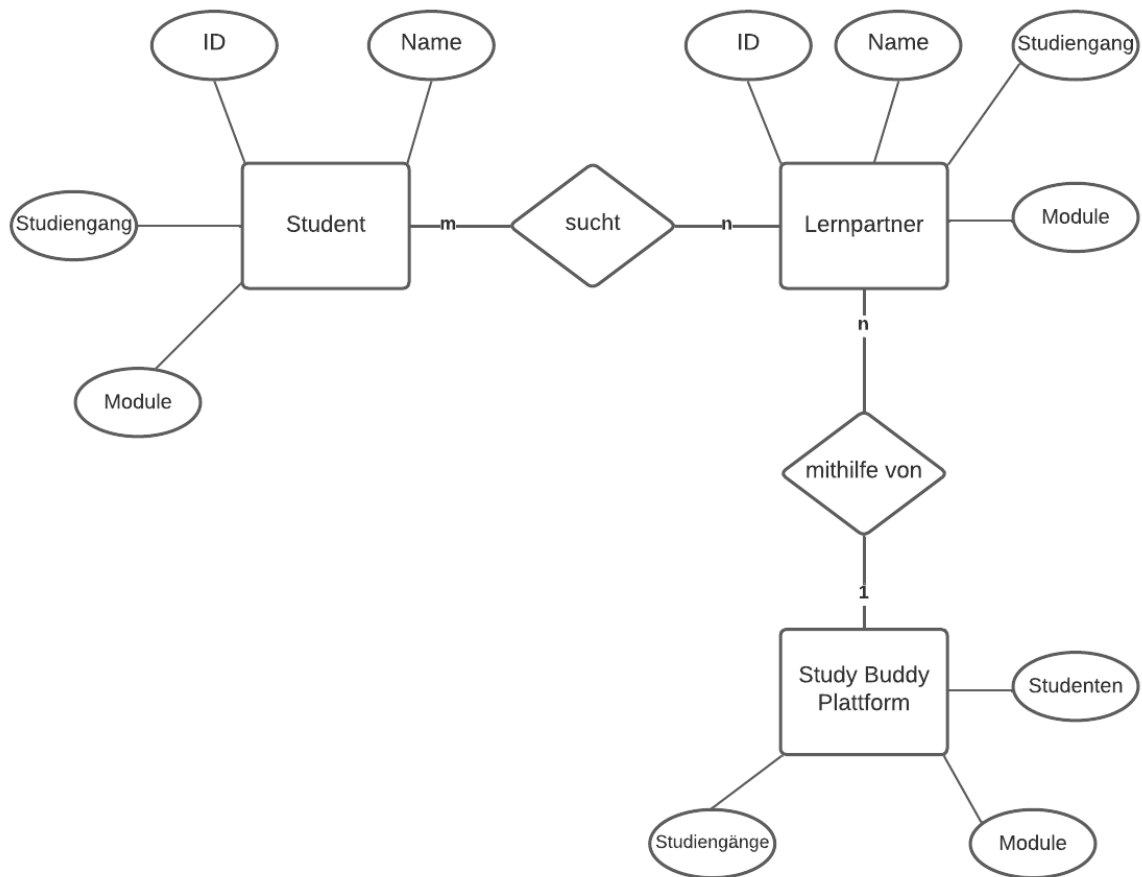
## 2 Anforderungen

### 2.1 Use-Case Diagramm



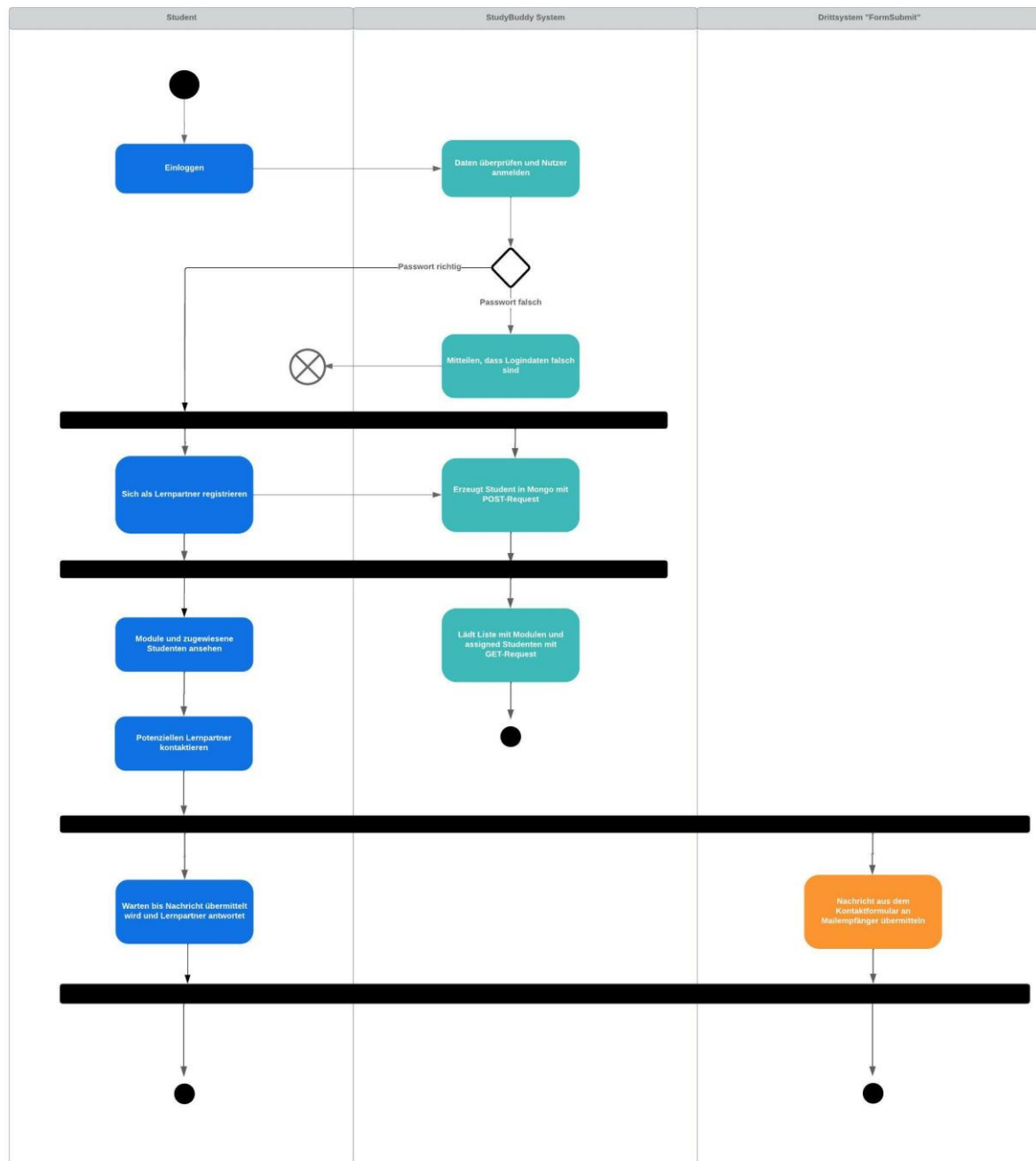
Oben ersichtlich ist die Use-Case-Grafik für meine StudyBuddy Plattform. Auf der einen Seite haben wir den End-User, in meinem Fall einen Studierenden, dem es ermöglicht werden soll, verschiedene Interaktionen durchzuführen, um einen Lernpartner finden zu können. Auf der anderen Seite der Entwickler/Administrator, welcher besondere Rechte hat wie beispielsweise ein neues Modul zu erfassen.

## 2.2 ER-Modell

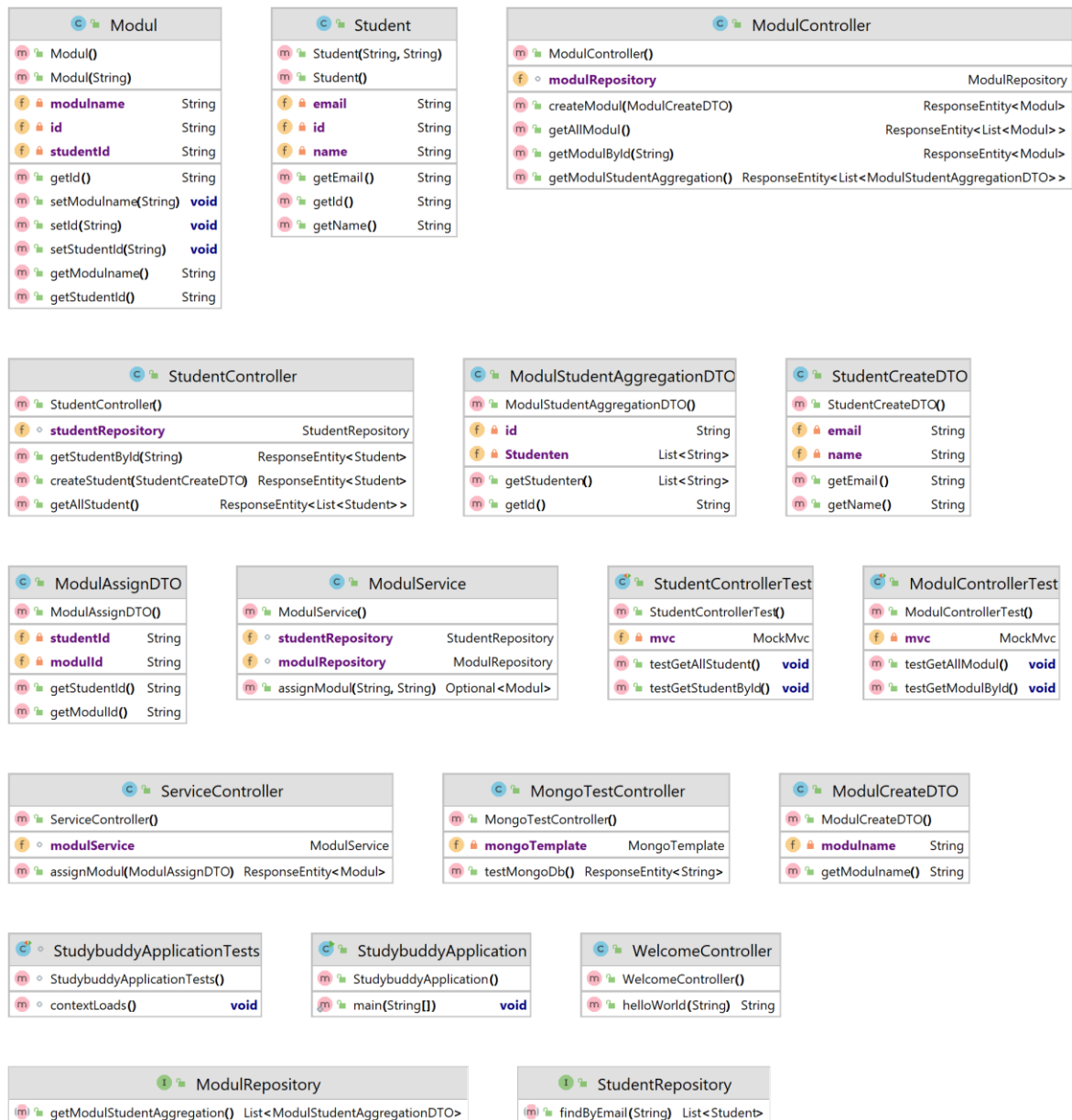


Hier sehen Sie ein mein ER-Diagramm für die StudyBuddy Plattform. Ein Student, der einen bestimmten Studiengang und dessen Module besucht, ist auf der Suche nach einem anderen Studenten (Lernpartner), welcher ebenfalls für dieselben Module lernen muss. StudyBuddy soll als eine sogenannte Vermittlungsbörse dienen und es den Studenten ermöglichen sich gegenseitig zu finden und kontaktieren, um sich zukünftig zusammen auf die Modulendprüfungen vorbereiten zu können.

## 2.3 BPMN-Diagramm



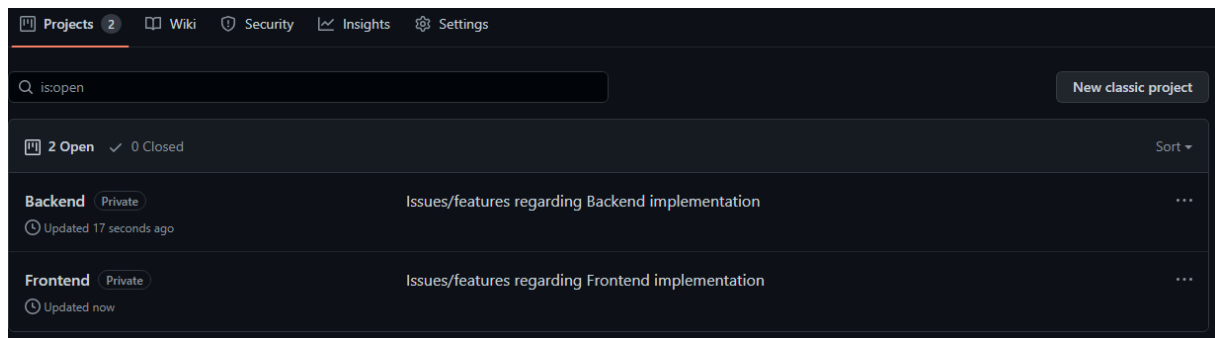
## 2.4 UML-Klassendiagramm



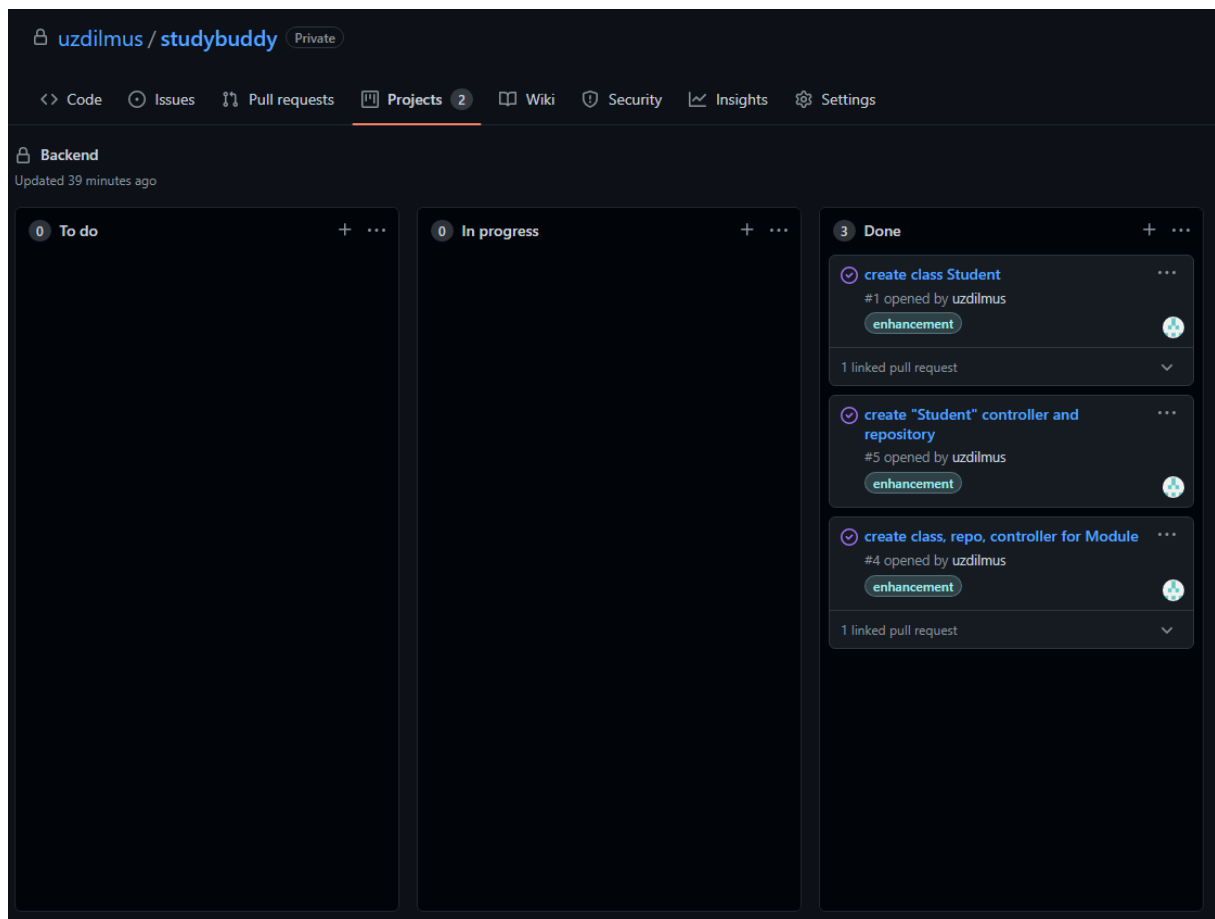
## 3 Implementation

### 3.1 GitHub

Zu Beginn habe ich auf der GitHub Plattform der ZHAW eine neue Repository namens «StudyBuddy» erstellt. Um mir selbst einen besseren Überblick verschaffen zu können, habe ich darin zwei Projekte erstellt, ein Projekt namens «Backend» und ein zweites, welches den Namen «Frontend» trägt:



Anhand der Informationen aus dem ER-Diagramm habe ich anschliessend begonnen, Issues für Klassen zu erstellen und damit zusammenhängend entsprechende Branches.



Wie beim obigen Screenshot bereits ersichtlich, habe ich nicht für jede Klasse ein separates Issue erstellt, sondern habe einige zusammengehörenden Klassen zu einem Issue zusammengeführt wie beispielsweise «create Student Controller and Repository». Für die erstellen Issues habe ich auch entsprechende Branches erstellt, diese mit den Issues verlinkt und mit Pull-Requests zusammengefügt:



## Branches:

All branches				
master	Updated 16 hours ago by uzdilmus	Default		
17-implement-unit-testings	Updated 17 hours ago by uzdilmus	2   0	#18	Merged
15-implement-login-function	Updated 19 hours ago by uzdilmus	4   0	#16	Merged
13-mock-server-svelte-frontend	Updated 20 hours ago by uzdilmus	6   0	#14	Merged
10-create-studentby modul	Updated 4 days ago by uzdilmus	9   0	#12	Merged
9-create-modulservice-and-modulassigndto	Updated 5 days ago by uzdilmus	11   0	#11	Merged
4-create-class-repo-controller-for-module	Updated 7 days ago by uzdilmus	13   0	#8	Merged
5-Student-controller-and-repository	Updated 7 days ago by uzdilmus	16   0	#7	Merged
1-class-student	Updated 7 days ago by uzdilmus	19   0	#6	Merged

## Pull Requests:

0 Open ✓ 8 Closed		Author	Label	Projects	Milestones	Reviews	Assignee	Sort
	added some Testings, all were successful					1		
	login added but button doesn't appear					1		
	13 mock server svelte frontend					1		
	made needed adds/changes for ModulStudentAggregati					1		
	created classes to assign Modul to a Student					1		
	created class, repo etc for Modul					1		
	5 student controller and repository							
	created class Student and StudentCreateDTO					1		

## 3.2 Technisches Datenmodell

Ein DTO ist eine Methode, die es ermöglicht Daten aus einem Teil einer Anwendung in einen anderen Teil zu übertragen, ohne dabei die Informationen aus einer Entwicklungsdatei preisgeben zu müssen. Aus diesem Grund können die Variablen in einer java Klasse auch als private definiert werden. DTOs dienen somit als eine Art Schnittstelle um Daten zwischen verschiedenen Klassen zu übermitteln.

Die Idee meines Projekts setzt voraus, dass man die Möglichkeit hat, Studenten und Module zu erstellen. Im Falle des Studenten hielt ich es für das Beste, wenn man seine Mailadresse (am besten gleich die der Bildungseinrichtung) und seinen Namen angeben muss.

### StudentCreateDTDO:

```
J StudentCreateDTO.java X
src > main > java > studybuddy > studybuddy > model > J StudentCreateDTO.java > ...
1  package studybuddy.studybuddy.model;
2
3  import lombok.Getter;
4  import lombok.NoArgsConstructor;
5
6  @NoArgsConstructor
7  @Getter
8  public class StudentCreateDTO {
9      private String email;
10     private String name;
11
12 }
```

Anschliessend kommt die Erstellung eines Moduls, wobei ein Name für das Modul übergeben werden muss.

### ModulCreateDTO:

```
J ModulCreateDTO.java X
src > main > java > studybuddy > studybuddy > model > J ModulCreateDTO.java > ...
1  package studybuddy.studybuddy.model;
2
3  import lombok.Getter;
4  import lombok.NoArgsConstructor;
5  @NoArgsConstructor
6  @Getter
7
8  public class ModulCreateDTO {
9
10     private String modulname;
11
12 }
```

Damit ein Student eine andere Person finden kann, welche für das gleiche Fach lernen möchte, müssen natürlich diese Personen den entsprechenden Modulen zugewiesen werden. Bei der Erstellung eines Studenten sowie der Erstellung eines Moduls wird durch MongoDB eine ID generiert. Für das Assignment eines Studenten zu einem Modul wird somit die StudentId und ModulId verwendet.

### ModulAssignDTO:

```
J ModulAssignDTO.java X
src > main > java > studybuddy > studybuddy > model > J ModulAssignDTO.java > ModulAssignDTO
1  package studybuddy.studybuddy.model;
2
3  import lombok.Getter;
4  import lombok.NoArgsConstructor;
5
6  @NoArgsConstructor
7  @Getter
8
9  public class ModulAssignDTO {
10
11     private String modulId;
12     private String studentId;
13
14 }
```

Nachdem Studenten verschiedenen Modulen zugewiesen worden sind, gilt es diese auszuweisen. Dies wurde durchgeführt, indem der Controller mithilfe der IDs und Namen aus den vorherigen DTOs eine Liste ausgibt mit dem Namen des Moduls und den zugewiesenen Studenten.

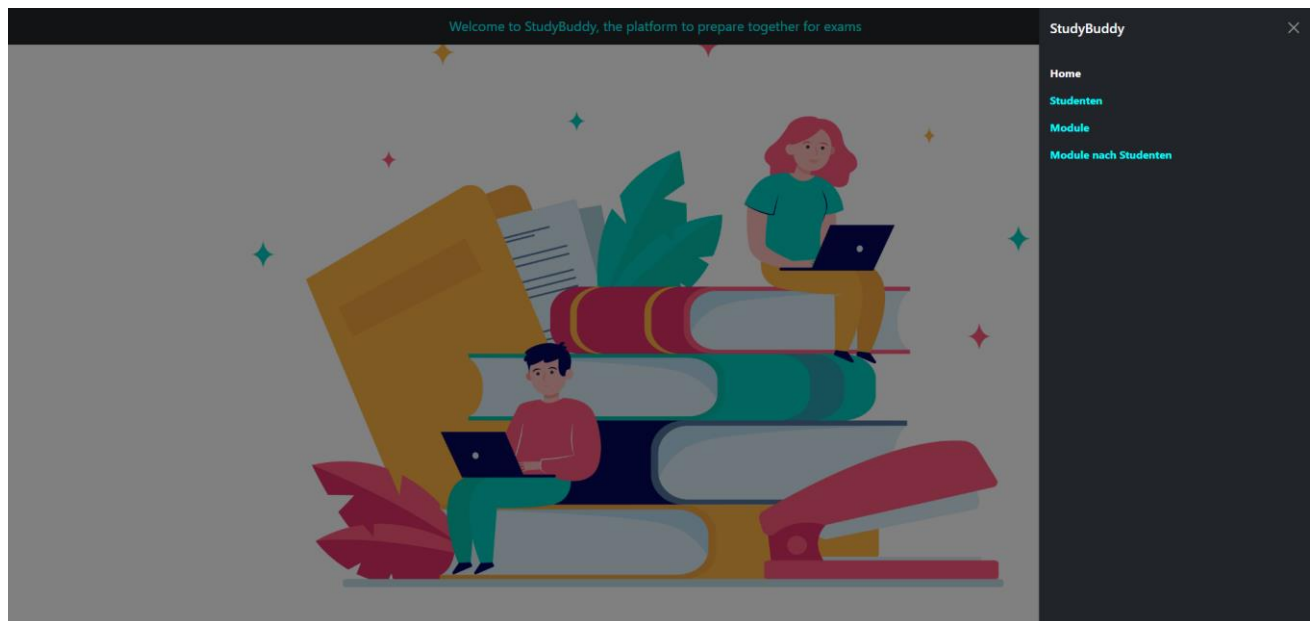
### ModulStudentAggregationDTO:

```
J ModulStudentAggregationDTO.java X
src > main > java > studybuddy > studybuddy > repository > J ModulStudentAggregationDTO.java > ...
1  package studybuddy.studybuddy.repository;
2
3  import java.util.List;
4
5  import lombok.Getter;
6  import lombok.NoArgsConstructor;
7
8  @NoArgsConstructor
9  @Getter
10 public class ModulStudentAggregationDTO {
11
12     private String id;
13     private List<String> Studenten;
14
15 }
16
```

### 3.3 Frontend

#### Home:

Mein Frontend habe ich so gestaltet, dass auf der rechten Seite ein Navigationsmenü erscheint mit den Verlinkungen zu weiteren svelte Seiten:



#### Studenten:

Auf dieser Seite hat man die Möglichkeit eine Liste mit allen bis jetzt registrierten Studenten zu sehen und ebenso die Funktion, sich selbst auch als Lernpartner einzutragen.

Welcome to StudyBuddy, the platform to prepare together for exams

### Als Lernpartner registrieren

Maiadresse

Name

Submit

### Liste registrierter Studenten

E-Mail	Name
uzdilimus@students.zhaw.ch	Mustafa
sabanluk@students.zhaw.ch	LuKa
vasicste@students.zhaw.ch	Stefan
stankdav@students.zhaw.ch	David
kopanbor@students.zhaw.ch	Boris
bakijism@students.zhaw.ch	Ismet
nikoldan@students.zhaw.ch	Daniel
mustafa.uzdillili@maagtechnic.com	Musti

## Module:

Ähnlich wie bei der vorherigen Seite sieht man hier eine Liste der bis jetzt hinzugefügten Module und man hat ebenfalls die Möglichkeit, ein neues Modul hinzuzufügen, falls dieses noch nicht besteht.

Welcome to StudyBuddy, the platform to prepare together for exams

### Modul hinzufügen

Modulname

Hinzufügen

### Alle Module

Modulname	Studentid
Software Engineering 2	63a21217f9d2957c11b8dac1
Data Analytics	63a211aef9d2957c11b8dabd
BWL	63a211d4f9d2957c11b8dabf
VWL	63a211f5f9d2957c11b8dac0
Mathematik	639f0db1e8504f5dc2b0d695
Software Engineering 2	639f0c7fe8504f5dc2b0d694
Mathematik	null
TestAddModul	null
TestNewModul, 25.12.22	null
TestNewModul 2, 25.12.22	null

## Module nach Studenten:

Auf dieser Seite ersichtlich sind die verfügbaren Module und den jeweiligen Studenten IDs, die einem Modul assigned wurden. Weiter unten findet man ein Kontaktformular welches, ein eingebundenes Drittsystem ist.

Welcome to StudyBuddy, the platform to prepare together for exams

### Module und zugewiesene Studenten

Modulname	Studentid
TestAddModul	
TestNewModul, 25.12.22	
Software Engineering 2	63a21217f9d2957c11b8dac1,639f0c7fe8504f5dc2b0d694
BWL	63a211d4f9d2957c11b8dabf
Data Analytics	63a211aef9d2957c11b8dabd
TestNewModul 2, 25.12.22	
Mathematik	639f0db1e8504f5dc2b0d695
VWL	63a211f5f9d2957c11b8dac0

### Student kontaktieren für eine Lernpartnerschaft

Gib hier deine Nachricht ein:

Gib hier deine Mailadresse ein:

Senden

### 3.4 Drittsystem: FormSubmit.co

Für mein Projekt wollte ich nicht eine beliebige API aus dem Internet nehmen und einbauen, sondern ich habe versucht ein Drittsystem zu finden, welches auch zu meinem Projekt passt und einen Sinn hat, einzusetzen. Nach längerer Recherche und gescheiterten Versuchen wie beispielsweise mit «Microsoft Graph API» bin ich irgendwann auf die Seite «[FormSubmit](#)» gestossen, welches ermöglicht ein Kontaktformular zu erstellen und über deren Server eine Mail zu versenden. Dieses Drittsystem habe ich dann eingebaut.

## Student kontaktieren für eine Lernpartnerschaft

Gib hier deine Nachricht ein:

Gib hier deine Mailadresse ein:

Auf meiner Website ist es somit möglich eine Nachricht zu schreiben und die eigene Mailadresse zu hinterlegen. Dies wird als Mail an den Empfänger verschickt, um eine Lernpartnerschaft anzufragen. Die empfangene Mail sieht folgendermassen aus:

Anfrage für Lernpartnerschaft



FormSubmit <submissions@formsubmit.co>  
An Uzdilli Mustafa (uzdillmus)



Antworten



Allen antworten



Weiterleiten



So. 25.12.2022 13:49

Klicken Sie hier, um Bilder herunterzuladen. Um den Datenschutz zu erhöhen, hat Outlook den automatischen Download von Bildern in dieser Nachricht verhindert.

Nachricht übersetzen in: Deutsch | Nie übersetzen aus: Englisch | [Übersetzungseinstellungen](#)

Someone just submitted your form on <http://localhost:8080/>.

Here's what they had to say:

**name:**

Hallo, Interesse an einer Lernpartnerschaft für die BWL Prüfung ?

**Mail\_des\_Absenders:**

[mustafa.uzdilli@outlook.com](mailto:mustafa.uzdilli@outlook.com)

## 4 Testing

### 4.1 Modul- und Integrationstests

Da bei solch einem Projekt Daten das wichtigste Gut darstellen, habe ich mich entschieden die Controller und dazugehörigen Requests zu testen, ob diese auch einwandfrei funktionieren.

## StudentControllerTest:

```
16 @SpringBootTest
17 @AutoConfigureMockMvc
18 public class StudentControllerTest {
19
20     @Autowired
21     private MockMvc mvc;
22
23     @Test
24     public void testGetAllStudent() throws Exception {
25         mvc.perform(get(urlTemplate: "/api/student").header(HttpHeaders.AUTHORIZATION, "Bearer "
26             + "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IktlUW5waEdXckthS0g0bjFmSE9mTCJ9.eyJ1c2VyX3JvbGVzIjpbImFkbWl
27             .andExpect(status().isOk())
28             .andExpect(jsonPath(expression: "$[0].email", is(value: "uzdilmus@students.zhaw.ch"))));
29     }
30
31     @Test
32     public void testGetStudentById() throws Exception {
33         mvc.perform(get(urlTemplate: "/api/student/639f0c7fe8504f5dc2b0d694").header(HttpHeaders.AUTHORIZATION, "Bearer "
34             + "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IktlUW5waEdXckthS0g0bjFmSE9mTCJ9.eyJ1c2VyX3JvbGVzIjpbImFkbWl
35             .andExpect(status().isOk())
36             .andExpect(jsonPath(expression: "$.email", is(value: "uzdilmus@students.zhaw.ch"))));
37     }
38 }
39
40
```

## ModulControllerTest:

```
15
16
17 @SpringBootTest
18 @AutoConfigureMockMvc
19 public class ModulControllerTest {
20
21     @Autowired
22     private MockMvc mvc;
23
24     @Test
25     public void testGetAllModul() throws Exception {
26         mvc.perform(get(urlTemplate: "/api/modul").header(HttpHeaders.AUTHORIZATION, "Bearer "
27             + "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IktlUW5waEdXckthS0g0bjFmSE9mTCJ9.eyJ1c2VyX3JvbGVzIjpbImFkbWl
28             .andExpect(status().isOk())
29             .andExpect(jsonPath(expression: "$[0].modulname", is(value: "Software Engineering 2"))));
30     }
31
32     @Test
33     public void testGetModulById() throws Exception {
34         mvc.perform(get(urlTemplate: "/api/modul/639f3c9e5f45405e8ccc2def").header(HttpHeaders.AUTHORIZATION, "Bearer "
35             + "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IktlUW5waEdXckthS0g0bjFmSE9mTCJ9.eyJ1c2VyX3JvbGVzIjpbImFkbWl
36             .andExpect(status().isOk())
37             .andExpect(jsonPath(expression: "$.modulname", is(value: "VWL"))));
38     }
39
40
41
42
```

## StudyBuddyApplicationTest:

```
src > test > java > studybuddy > studybuddy > StudybuddyApplicationTests.java > Language Support for Java(TM) by Red Hat > StudybuddyApplicationTests
1  package studybuddy.studybuddy;
2
3  import org.junit.jupiter.api.Test;
4  import org.springframework.boot.test.context.SpringBootTest;
5
6  @SpringBootTest
7  class StudybuddyApplicationTests {
8
9      @Test
10     void contextLoads() {
11     }
12
13 }
14
```

## JaCoCo IntegrationTest:

### studybuddy

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
studybuddy.studybuddy.controller	<div><div></div></div>	32%	<div><div></div></div>	20%	11 20	22 37	6 15	0 5
studybuddy.studybuddy.model	<div><div></div></div>	23%	<div><div></div></div>	0%	17 25	9 13	13 21	3 5
studybuddy.studybuddy.service	<div><div></div></div>	8%	<div><div></div></div>	0%	3 4	8 9	1 2	0 1
studybuddy.studybuddy.repository	<div><div></div></div>	0%	<div><div></div></div>	n/a	3 3	2 2	3 3	1 1
studybuddy.studybuddy	<div><div></div></div>	37%	<div><div></div></div>	n/a	1 2	2 3	1 2	0 1
Total	263 of 360	26%	20 of 22	9%	35 54	43 64	24 43	4 13

## 4.2 User-Tests

Für die User-Tests habe ich vier Personen aus meinem Freundeskreis gebeten mir zu helfen. Zuerst habe ich ihnen die Idee meines Projekts erläutert. Anschliessend habe ich Ihnen die Website gezeigt, jedoch bewusst keine Anweisungen gegeben, sondern sie gebeten sich selbst durchzuklicken und von ihren Eindrücken zu erzählen. Alle Freunde für meine User-Tests haben einen Bachelor in Wirtschaftsinformatik, deswegen war es für mich noch interessanter zu sehen, wie ihre Erfahrungen und Eindrücke sind.

### Testperson 1

#### Boris Kopanja

##### Steckbrief:

- Hat den Bachelor in Wirtschaftsinformatik von der ZHAW
- Absolviert zurzeit den Master in Wirtschaftsinformatik
- Tätig als IT-Management Associate bei der Würth Finance International

##### Feedback zum Projekt:

«Was die Funktionalität betrifft scheint alles zu funktionieren, das Frontend könnte man allerdings noch weiterausbauen.»



## Testperson 2

### Farschid Rashidi

#### Steckbrief:

- Hat den Bachelor in Wirtschaftsinformatik von der ZHAW
- Absolviert zurzeit den Master in Wirtschaftsinformatik
- Arbeitet als Application Manager bei der eidg. Finanzmarktaufsicht (FINMA)

#### Feedback zum Projekt:

«Eine vielversprechende Idee. Es macht jedoch den Eindruck, als wäre es noch in seinen Anfangsstadien, man könnte noch weitere Funktionen einbauen wie bestimmte Filter oder eine Chatfunktion.»

## Testperson 3

### Admir Durmo

#### Steckbrief:

- Hat den Bachelor in Wirtschaftsinformatik von der FHSG
- Absolviert zurzeit den Master in Wirtschaftsinformatik
- Arbeitet als DevOps Engineer bei der Swisscom

#### Feedback zum Projekt:

«Das Frontend könnte definitiv noch verschönert werden. Bei den Modulen und zugewiesenen Studenten wird die ID dieser Leute angezeigt und nicht die Namen. Das sieht gar nicht schön aus. Was die Funktionen betrifft, scheint alles ordnungsgemäss zu laufen.»

## Testperson 4

### Veton Rasaj

#### Steckbrief:

- Hat den Bachelor in Wirtschaftsinformatik von der FHSG
- Absolviert zurzeit den Master in Wirtschaftsinformatik
- Arbeitet als Requirements Engineer bei den Helvetia Versicherungen

#### Feedback zum Projekt:

«Scheint mir als hättest du einige Grundfunktionen eingebaut, jedoch könnte man das noch erweitern. Es gibt z.B. keine Möglichkeit mich selbst einem Fach zuzuweisen, das könnte man bei der Registrierung als Lernpartner noch einbauen. Das Frontend könnte man verschönern und weitere Komponenten für eine bessere UX einbauen.»

## **5 Fazit**

### **Stand der Implementation**

Allen Feedbacks, die ich von meinen Freunden erhalten habe, stimme ich zu. Das Projekt enthält aktuell Grundfunktionen und ein Frontend, jedoch ist das Ganze noch in der Entwicklungsphase und hinterlässt auch entsprechend einen solchen Eindruck bei anderen Personen. Die Plattform ist ein guter Anfang, jedoch noch in ihrer Entwicklungsphase, man könnte sowohl noch mehr Funktionen einbauen als auch das Frontend besser gestalten.

Ursprünglich sah es in meiner Vorstellung folgendermassen aus: Studenten können ihre Institution und den Studiengang auswählen, anschliessend wählen sie die gewünschten Module aus einem Dropdown Menü an. Danach hätte man ein Interface wie z.B. bei der App «Tinder», womit ein Student sich durch die potenziellen Lernpartner durch interagieren könnte und bei den Personen, die er kontaktieren möchte, direkt anschreiben kann in einem eingebauten Chat.

### **Persönliches Fazit**

Dieses Projekt hat mir aufgezeigt, dass es einfach ist sich eine Idee vorzustellen, doch die Umsetzung dieser Idee im Kopf fällt sehr viel schwieriger aus in der Realität. Grund dafür ist meiner Meinung nach der enorme Aufwand um solch ein Projekt umzusetzen und das Know-How, welches essenziell ist für die Umsetzung.

Genau dies war auch mein Problem mit diesem Projekt. Es war definitiv interessant und lehrreich zu sehen, wie unzählige Programme, die wir heutzutage auf unseren Smartphones und Computern nutzen, aus der Sicht des Endnutzers simpel zu sein scheinen, doch die Entwicklung dafür sehr viel Zeit beansprucht und das entsprechende Wissen benötigt. Dies war auch mein Problem, denn während der Arbeit an diesem Projekt fühlte ich mich oft überfordert, weil ich fand, dass es einerseits ein enormer Aufwand ist und andererseits mir das nötige Know-How fehlte, um ein solches Projekt auf die Beine zu stellen.

Oft ist es auch vorgekommen, dass man etwas genauso gemacht hat wie in den Übungen beschrieben, danach eine Komponente/Funktion, welche vorher einwandfrei funktionierte, plötzlich nicht mehr lief. Aus diesem Grund fehlt bei meinem Projekt beispielsweise die Funktion mit dem Login und «authenticated». In meinen GitHub Commitments ist ersichtlich, dass ich es eingebaut, jedoch später wieder entfernt habe, weil andere Funktionen die ich als wichtiger eingestuft habe, nicht mehr liefen.

So wie viele meiner Mitstudierenden habe auch ich eine kaufmännische Matura und war noch nie als Entwickler oder ähnliches tätig. Somit war man die meiste Zeit überfordert, weil man

nicht wusste, wie man etwas umsetzen muss oder warum denn an dieser Stelle nun ein Fehler angezeigt wird und wie man diesen Fehler überhaupt beheben kann. Nichtsdestotrotz habe ich und bestimmt auch alle meine Mitstudierenden, unser Bestes gegeben und versucht den Anforderungen gerecht zu werden.