

Realisierungsbericht

Status	In Prüfung
Projektname	Gradus
Projektleiter	Mirio Eggmann
Auftraggeber	Beat Walter
Autoren	Dario Menzel
Verteiler	Entwicklerteam - Manuel Bieri - Nicolas Brechbühler - Mirio Eggmann - Dario Menzel Beat Walter

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung, Bemerkung	Name oder Rolle
v0.5.0	08.11.2016	Erstellung des Dokumentes, Übernahme unseres Layouts	Dario Menzel
v0.6.0	22.11.2016	Bearbeitung Kapitel «Test»	Dario Menzel
v0.7.0	29.11.2016	Bearbeitung Kapitel «Systemdokumentation»	Nicolas Brechbühler
v0.7.5	29.11.2016	Bearbeitung Kapitel «Systemdokumentation»	Dario Menzel
v0.8.0	06.12.2016	Zusammenfassung geschrieben	Nicolas Brechbühler
v0.8.3	06.12.2016	Testprozedur und Supporthandbuch angepasst	Dario Menzel
v0.8.5	06.12.2016	Technische Detailspezifikation	Mirio Eggmann
V0.9.0	09.12.2016	Konfiguration, Schnittstellendefinition, Systemdesign, Sicherheit, Datenmodell beschrieben.	Mirio Eggmann
V0.9.5	10.12.2016	Benutzerhandbuch, Supporthandbuch, Tests aktualisiert.	Mirio Eggmann
v1.0.0	11.12.2016	Überprüfung gesamtes Dokument und kleine Feinheiten verbessert	Mirio Eggmann, Dario Menzel, Nicolas Brechbühler

Definitionen und Abkürzungen

Begriff / Abkürzung	Bedeutung
SQL	«Structured Query Language» Datenbanksprache
CSRF	«Cross-Site-Request-Forgery» Webseitenübergreifende Anfragefälschung
XSS	«Cross-Site-Scripting» Ausnutzen einer Computersicherheitslücke
PDF	«Portable Document Format» Plattformunabhängiges Datei Format
REST	«Representational State Transfer» Schnittstelle zwischen zwei Systemen
JPA	«Java Persistence API» Datenbankzugriffe und objektrelationales Mapping
Single-page	Eine moderne Webanwendung, die dynamisch Inhalt lädt.
Multi-page	Eine Webanwendung die bei jeder Anfrage den gesamten Inhalt neu lädt.
GIBB	Gewerblich-Industrielle Berufsschule Bern
App	Abkürzung für Applikation / Anwendung.

Referenzen

Referenz	Titel, Quelle
[1]	Angular, https://angular.io
[2]	Hibernate, http://hibernate.org
[3]	MySQL https://www.mysql.de

Inhaltsverzeichnis

1	Zusammenfassung	5
2	Technische Detailspezifikation	5
2.1	Systemdesign	5
2.1.1	Struktur des Systemdesigns	5
2.1.2	Beschreibung der Elemente	8
2.2	Schnittstellendefinitionen	10
2.3	Datenmodel	11
2.3.1	ERD	11
2.4	Sicherheit	12
2.4.1	Informationssicherheit	12
2.4.2	Datenschutz	12
3	Systemdokumentation	13
3.1	Konfigurations-Dokumentation	13
3.1.1	Entwicklung	13
3.1.2	Produktion	13
3.2	Benutzerhandbuch	14
3.2.1	Systemübersicht	14
3.2.2	Anwenderfunktionalität	14
3.3	Integrations- und Installationshandbuch	20
3.3.1	Anwender	20
3.3.2	Systeminstallation	20
3.3.2.1	Benötigte Software:	20
3.3.2.2	Installation	21
3.4	Suppothandbuch	21
4	Test	22
4.1	Unit Test	22
4.2	Systemtest - Testspezifikation	22
4.2.1	Kritikalität der Funktionseinheit	22
4.2.2	Testanforderungen	22
4.2.3	Testverfahren	22
4.2.4	Testkriterien	22
4.2.4.1	Abdeckungsgrad:	22
4.2.4.2	Checklisten:	22
4.2.4.3	Endkriterien:	22
4.2.5	Testfälle	22
4.3	Testprozedur	23
4.3.1	Account Erstellen	23
4.3.2	Anmelden	23
4.3.3	Note hinzufügen	24
4.3.4	Fach hinzufügen	24
4.3.5	Note bearbeiten	24
4.3.6	Fach bearbeiten	25
4.3.7	Benutzer bearbeiten	25
4.3.8	Note löschen	25
4.3.9	Fach löschen	26
4.3.10	Benutzer löschen	26
4.3.11	Semester hinzufügen	26
4.3.12	Semester bearbeiten	27
4.3.13	Semester löschen	27
4.4	Testprotokoll	28
5	Antrag auf Freigabe der nächsten Projektphase	28
A)	Quellcode	29

Abbildungsverzeichnis

Abbildung 1 Backend Diagramm – Model	5
Abbildung 2 Backend Diagramm - Controller	6
Abbildung 3 Backend Diagramm - Repo	6
Abbildung 4 Frontend Diagramm.....	7
Abbildung 5 Backend Mapping.....	10
Abbildung 6 Frontend Mapping Zugriff	10
Abbildung 7 Hibernate Config.....	10
Abbildung 8 Hibernate Mapping	10
Abbildung 9 ERD	11
Abbildung 10: Login-Seite	14
Abbildung 11: Registrations-Seite	15
Abbildung 12: Profil-Seite	15
Abbildung 13: Seite zum Hinzufügen neuer Noten	16
Abbildung 14: Seite zum Hinzufügen neuer Fächer.....	17
Abbildung 15: Seite zum Hinzufügen neuer Semester.....	18
Abbildung 16: Dashboard Übersichts-Seite	18
Abbildung 17: Fach Übersichts-Seite	19

Tabellenverzeichnis

Tabelle 1 Beschreibung der Elemente - Backend	8
Tabelle 2 Beschreibung der Elemente - Frontend.....	9
Tabelle 3 ERD Description	11
Tabelle 4 Informationssicherheit.....	12
Tabelle 5 Datenschutz	12
Tabelle 6 Entwicklungsparameter	13
Tabelle 7 Produktionsparameter	13
Tabelle 8 Benötigte Software	20
Tabelle 9 Testfälle	22
Tabelle 10 Testfall: Account erstellen.....	23
Tabelle 11 Testfall: Anmelden	23
Tabelle 12 Testfall: Note hinzufügen	24
Tabelle 13 Testfall: Fach hinzufügen.....	24
Tabelle 14 Testfall: Note bearbeiten.....	24
Tabelle 15 Testfall: Fach bearbeiten	25
Tabelle 16 Testfall: Benutzer bearbeiten.....	25
Tabelle 17 Testfall: Note löschen	25
Tabelle 18 Testfall: Fach löschen.....	26
Tabelle 19 Testfall: Benutzer löschen	26
Tabelle 20 Testfall: Semester hinzufügen	26
Tabelle 21 Testfall: Semester bearbeiten.....	27
Tabelle 22 Testfall: Semester löschen	27
Tabelle 23 Testprotokoll	28

1 Zusammenfassung

Der Realisierungsbericht fasst die Ergebnisse und Entscheide der Realisierungsphase zusammen. Deswegen enthält er unter anderem Informationen wie technische Detailspezifikationen sowie ein Benutzerhandbuch und Testprotokoll. Für allfällige Nachbesserungen dient dieses Dokument als Grundlage.

2 Technische Detailspezifikation

2.1 Systemdesign

2.1.1 Struktur des Systemdesigns

Backend:

Das Backend Diagramm zeigt die wichtigsten Teile unserer Backend Applikation. Auch hier haben wir uns auf das wesentliche reduziert um die Übersichtlichkeit zu gewährleisten. Weiter konnten die verschiedenen Verbindungen nicht gezogen werden, weil die Diagramme aufgeteilt werden müssen, zur besseren Lesbarkeit.

Model

Dieses Diagramm zeigt die Models des Backendes in Gradus. Diese Java Models werden mit Hibernate gemappt und sind somit mit der Persistenzschicht, also der DB (siehe ERD) verknüpft.

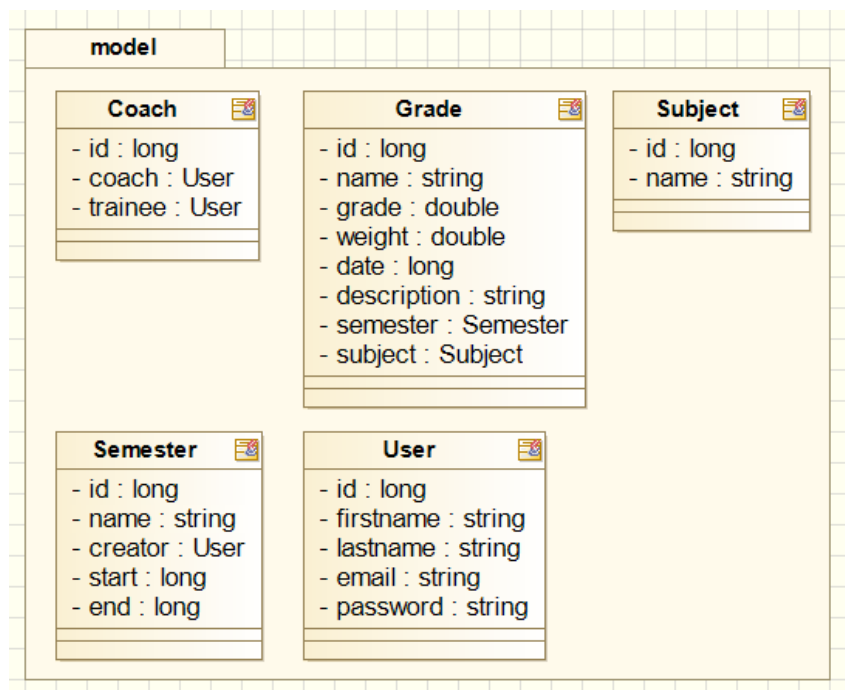


Abbildung 1 Backend Diagramm – Model

Controller

Die Controller machen die Verbindung zwischen dem Backend und dem Frontend. Diese stellen über URLs Daten vom Backend dem Frontend zur Verfügung. Dies nennt man eine sogenannte REST Schnittstelle. Bei dieser Schnittstelle wird über eine URL Daten im JSON Format zur Verfügung gestellt.

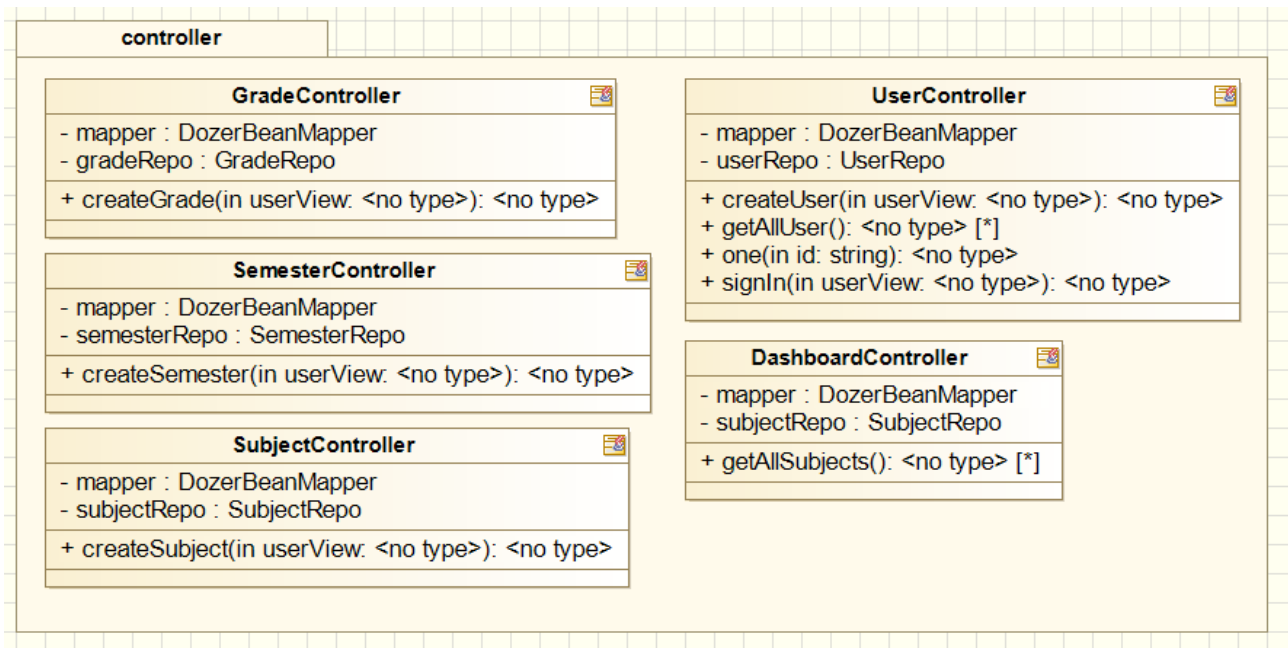


Abbildung 2 Backend Diagramm - Controller

Repo

Die Repos stellen die verschiedenen DB Abfragen zur Verfügung.

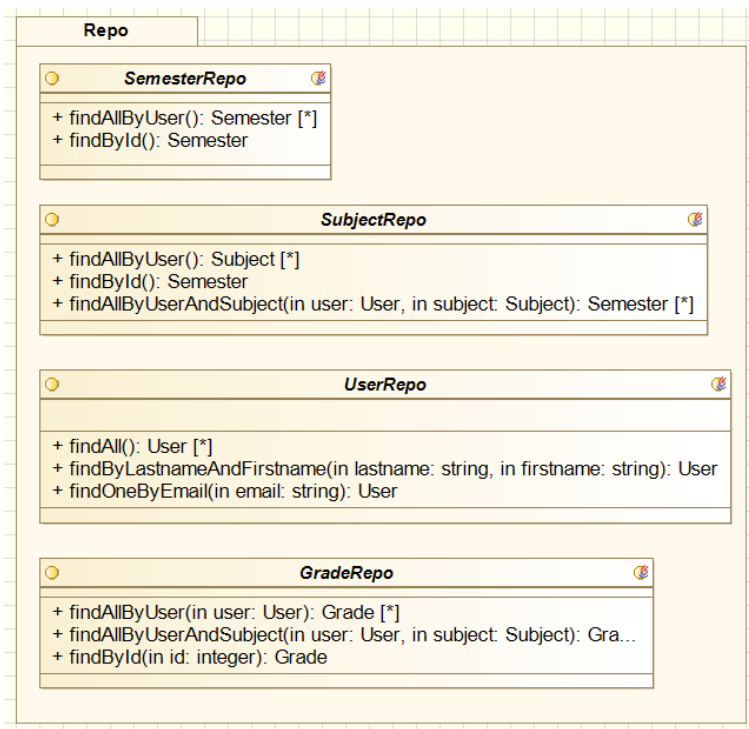


Abbildung 3 Backend Diagramm - Repo

Frontend:

Das Diagramm zeigt auf, wie die das Frontend der Applikation aufgebaut ist. Es wird nur das wichtigste aufgezeigt, damit die Verständlichkeit des Diagrammes nicht leidet. Es zeigt auf wie der Benutzer mit einem Request zugreift und grundsätzlich nur mit der index.html Datei kommuniziert. Dahinter arbeitet dann unsere Frontend Applikation und liefert anschliessend wieder alles an die index.html Datei. Dieser Teil der Applikation kommuniziert über die Services mit dem REST Backend. Dort werden Daten vom Backend geholt oder Daten ans Backend gesendet.

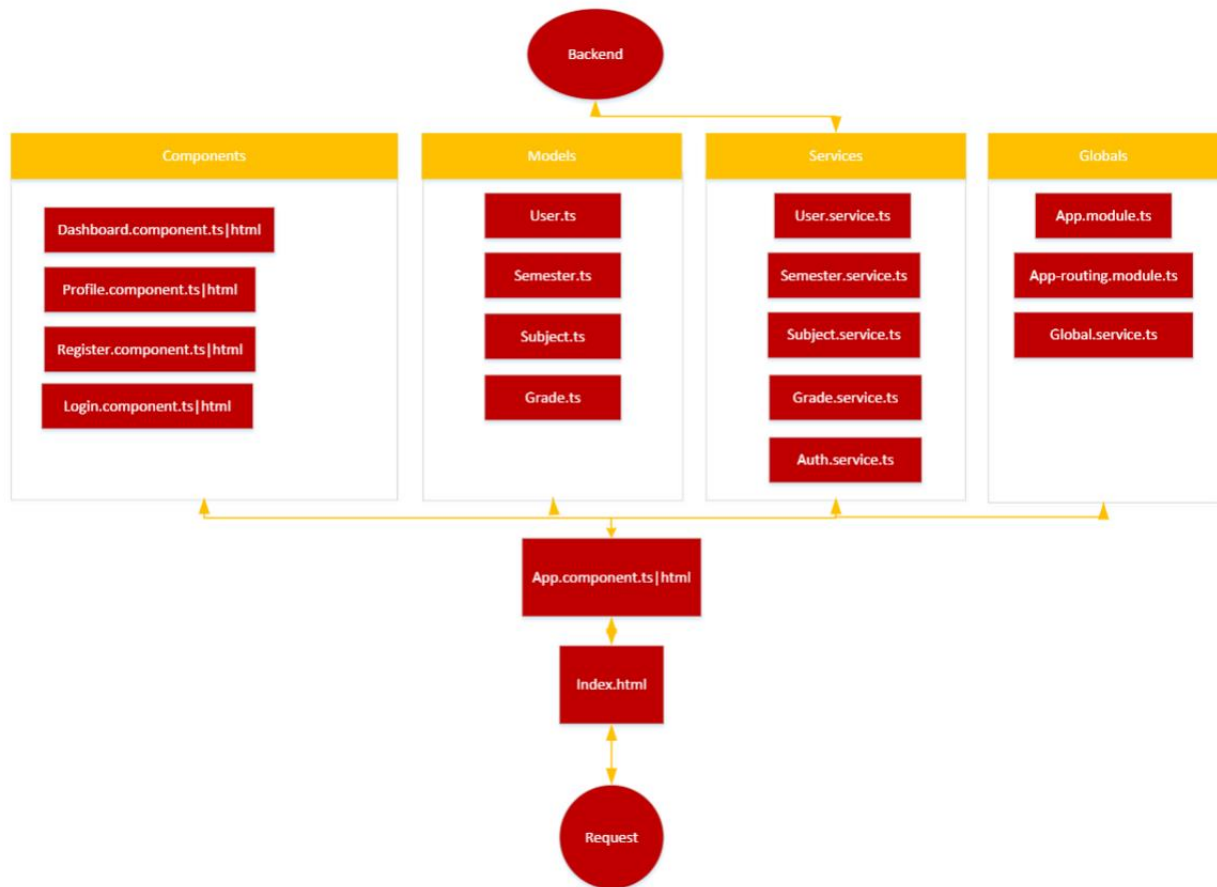


Abbildung 4 Frontend Diagramm

2.1.2 Beschreibung der Elemente

Backend:

Tabelle 1 Beschreibung der Elemente - Backend

Element	Beschreibung
Model: <ul style="list-style-type: none">- Coach- Grade- Subject- User- Semester	Die Models halten die Daten der Applikation. Sie sind mit der DB Verknüpft über Hibernate. <ul style="list-style-type: none">- Coach ist damit ein Benutzer die Noten eines anderen einsehen kann.- Grade ist um Informationen zu den einzelnen Noten zu halten.- Subject ist für ein Fach.- User enthält alle Informationen über den Benutzer, jedoch wird das Passwort nicht über die REST Schnittstelle zur Verfügung gestellt.- Semester enthält Infos zum Semester.
Controller <ul style="list-style-type: none">- GradeController- SubjectController- SemesterController- UserController- DashboardController	Controller sind für die Verbindung zwischen Backend und dem Frontend zuständig. Sie kommunizieren per REST Schnittstelle. Es werden verschiedene URLs zur Verfügung gestellt, welche es dem Frontend ermöglichen Daten zu bekommen und Daten zu schicken. <ul style="list-style-type: none">- GradeController für Verwaltung der Noten.- SubjectController für die Verwaltung der Fächer.- SemesterController für die Verwaltung der Semester.- UserController für die Verwaltung des Benutzers.- DashboardController für die Verwaltung der gesamten Applikation. Sprich von Semester, Note und Fach.
Repo <ul style="list-style-type: none">- SemesterRepo- SubjectRepo- GradeRepo- UserRepo	Stellen verschiedene Abfragen zur Verfügung um Daten aus der DB zu holen oder Daten einzufügen. <ul style="list-style-type: none">- SemesterRepo für die Abfragen des gesamten Semesters oder allen Semestern.- SubjectRepo für die Abfragen von einem Fach oder von allen Fächern.- GradeRepo für die Abfragen von einem oder allen Noten eines Faches.- UserRepo für alles rund um den Benutzer selbst.

Frontend:

Zweck der wichtigsten Elemente:

Tabelle 2 Beschreibung der Elemente - Frontend

File	Beschreibung
frontend/src/index.html	Startpunkt der Angular Applikation. Dort werden alle Stylesheets importiert, das Favicon und der Title gesetzt.
frontend/src/app/app.module.ts	Dieses File importiert Module die in der gesamten Applikation verwendet werden. Weiter bootstrap dieses File die AppComponent.
frontend/src/app/app-routing.module.ts	Dieses File kümmert sich um das Routing (Navigationsmöglichkeiten) der Applikation.
frontend/src/app/app.component.ts html	Beinhaltet Globale Services und das Grundkonstrukt der Seite.
frontend/src/app/login/	Beinhaltet die Login-Komponente.
frontend/src/app/register/	Beinhaltet die Registrierungs-Komponente.
frontend/src/app/dashboard/	Beinhaltet die Hauptkomponente, wo die Noten verwaltet werden.
frontend/src/app/profile/	Beinhaltet die Benutzerprofil Komponente.
frontend/src/app/shared/service/	Beinhaltet alle Services die von den Komponenten genutzt werden können. Diese Services holen Daten vom Backend über die sogenannten REST Schnittstellen.

2.2 Schnittstellendefinitionen

Gradus verfügt über folgende Schnittstellen:

REST (Backend – Frontend)

Die verschiedenen Controller im Backend stellen über URLs Daten dem Frontend zur Verfügung. Die Daten werden im JSON Format bereitgestellt. Heute ist dies eine sehr verbreitete Methode um das Frontend und das Backend vollständig trennen zu können und so bessere Applikationen schreiben zu können. Dies funktioniert wie folgt:

Der Controller im Spring Backend mappt eine Funktion, z.B. `getAllSubjects()` und stellt diese dann über die URL „/webresources/subject“ zur Verfügung. Dies erfolgt in dem Fall über eine GET Abfrage, würde aber auch mit POST funktionieren.

```
@RequestMapping(value = "/webresources/subject", method = RequestMethod.GET)
```

Abbildung 5 Backend Mapping

Anschliessend kann das Frontend über die passende URL auf die nötigen Ressourcen zugreifen. Danach kann man in einer Frontend Komponente auch einfach `getSubjects()` aufrufen und bekommt die Daten vom Backend zugestellt.

```
private baseUrl = 'http://localhost:8080/webresources/subject'; // URL to web api

getSubjects() {
    return this.http.get(this.baseUrl)
        .map(res => <Subject[]> res.json())
        .catch(this.handleError);
}
```

Abbildung 6 Frontend Mapping Zugriff

Hibernate (Backend – DB)

In einer sogenannten properties Datei werden die nötigen Informationen zur Datenbank gespeichert, damit die Applikation mit der DB operieren kann. Dies beinhaltet unter anderem die URL der bereitgestellten DB, der Name und das Passwort des Benutzers und ein paar nötige Daten zu Hibernate selbst.

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/gradus
spring.datasource.username=gradus
spring.datasource.password=gradus
spring.datasource.testOnBorrow=true
spring.datasource.validationQuery=SELECT 1
spring.jpa.hibernate.naming_strategy=org.hibernate.cfg.EJB3NamingStrategy
logging.level.org.hibernate=INFO
spring.jpa.hibernate.ddl-auto=update
```

Abbildung 7 Hibernate Config

Anschliessend werden die Model Klassen mit Hibernate gemappt und somit weiss die Applikation automatisch was sie machen muss, wenn eine Änderung getätigt wird. Im Bild sieht man wie eine Java Klasse mit einer DB Table gemappt wurde.

```
@Entity
@Table (name = "GRAD_user")
public class User {
```

Abbildung 8 Hibernate Mapping

2.3 Datenmodell

2.3.1 ERD

Im folgenden Diagramm sieht man das ERD von der Applikation Gradus. Es handelt sich dabei um eine MySQL Datenbank. Da die Applikation aber mit Hibernate arbeitet, wäre es Problem los möglich eine andere Datenbank, wie z.B. Oracle DB zu verwenden.

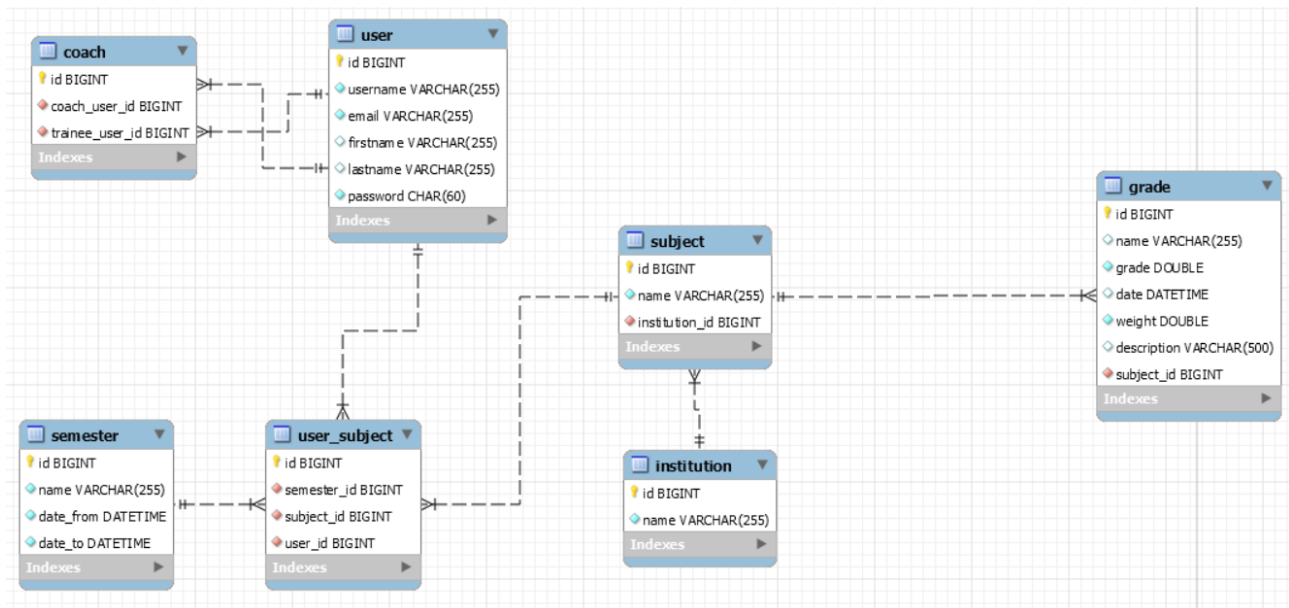


Abbildung 9 ERD

Beschreibung der Tabellen

Tabelle 3 ERD Beschreibung

ID	Tabelle	Beschreibung
T1	user	Beinhaltet alle Informationen über einen Benutzer, damit er identifiziert werden kann.
T2	user_subject	Verbindet den Benutzer mit seinen Noten für das jeweilige Semester. Denn der Benutzer kann für jedes Semester unterschiedliche Fächer haben.
T3	semester	Ermöglicht es Semester zu erstellen, mit einem Namen und einer Laufzeit.
T4	subject	Ein Fach von der Schule im Verbund mit der Schule bei dem das Fach besucht wird.
T5	institution	Beinhaltet die verschiedenen Bildungseinrichtungen.
T6	grade	Beinhaltet die einzelnen Noten mit ihrem Gewicht, Probenname, Probedatum, Gewichtung, Beschreibung und dem dazugehörigen Fach.
T7	coach	Die Tabelle Coach speichert, wenn ein User ein Coach von einem anderen User ist. Dies ermöglicht ihm dann, dass er einen Einblick in die Noten des Schülers haben kann.

2.4 Sicherheit

2.4.1 Informationssicherheit

Tabelle 4 Informationssicherheit

Beschreibung:	Gefährdungen:	Massnahmen:
Organisatorische und technische Massnahmen zur Sicherstellung von Verfügbarkeit und Authentizität der Daten.	<ul style="list-style-type: none">- Löschen von Daten von anderen Benutzern- Ausfall des Servers	<ul style="list-style-type: none">- Berechtigungskonzept implementiert und nur den Benutzern die die Noten gehören können diese auch bearbeiten.- Die Datenbank kann exportiert werden um den Schaden von Verlust von Daten möglichst gering zu halten. Dies sollte der Administrator in einem Cron-Script wöchentlich durchführen.

2.4.2 Datenschutz

Tabelle 5 Datenschutz

Beschreibung:	Schützenswerte Daten:	Massnahmen:
Schutz sensibler Daten vor unbefugtem Zugriff und vor missbräuchlicher Verwendung.	<ul style="list-style-type: none">- Benutzerinformationen- Notendaten- Source-Code	<ul style="list-style-type: none">- Die Benutzerdaten sind nur für den jeweiligen Benutzer einsehbar. Weiter hat auch der Coach Zugriff. Alle anderen sehen diese nicht.- Die Notendaten können ebenfalls nur vom Benutzer selbst oder vom Coach angesehen werden.- Der Source-Code wird in einem externen Repository gesichert, damit er nicht verloren gehen kann. Weiter ist er mit einer Lizenz versehen-

3 Systemdokumentation

3.1 Konfigurations-Dokumentation

Gradus ist eine Applikation, die als Frontend Angular 2 einsetzt, als Backend Spring Boot und als Datenbank MySQL im Verbund mit dem Hibernate Framework. Dies macht das System sehr flexibel, weil die einzelnen Komponenten Frontend, Backend, Datenbank sehr leicht ausgetauscht werden können und somit ist die Applikation auch in Zukunft noch gut erweiterbar. Jedoch folgt daraus auch, dass das Ganze etwas komplexer aufgebaut ist. Die Applikation kann in 2 verschiedenen Umgebungen genutzt werden. Lokal auf dem Entwicklernotebook oder in der Produktion auf einem entsprechenden Server. Wir gehen davon aus das ein Windows Notebook zur Entwicklung genutzt wird und daher wird es für dies beschrieben. Bei einem Linux Benutzer gehen wir davon aus das dieser die kleinen Differenzen in der Konfiguration erkennen kann und somit das gleiche auf einem Linux PC erreichen kann.

3.1.1 Entwicklung

Parameter:

Tabelle 6 Entwicklungsparameter

Frontend (Angular 2) Port:	4200
Backend (Spring Boot) Port:	8080
DB (MySQL) Port:	3306
DB Schema Name:	gradus
DB Benutzer:	gradus
DB Benutzer Passwort:	gradus

3.1.2 Produktion

In der Produktion wird ein Droplet von Digitalocean verwendet. Dies ist aber erst ab Anfang 2017 verfügbar. Bis zu diesem Zeitpunkt kann es selbständig auf einem eigenen Server installiert werden. Einige Parameter werden sich bis 2017 noch ändern.

Parameter:

Tabelle 7 Produktionsparameter

Frontend (Angular 2) Port:	4200
Backend (Spring Boot) Port:	8080
DB (MySQL) Port	3306
DB Schema Name:	gradus
DB Benutzer:	gradus
DB Benutzer Passwort:	WIRD NICHT HIER FESTGEHALTEN! *(Nachfragen bei mirio.eggmann@protonmail.ch)

3.2 Benutzerhandbuch

3.2.1 Systemübersicht

Ziel dieses Systems ist es, die momentane Applikation zur Notenerfassung der Post CH AG, durch unsere Web-Anwendung abzulösen. In Gradus geht es darum, dass der Benutzer seine Schulnoten verwalten kann. Noten können eingetragen, bearbeitet und gelöscht werden. Darüber hinaus hat der Benutzer all seine Noten, Semester und Gesamtschnitte immer komfortabel im Überblick. Die Applikation soll stets verfügbar sein, sogar unterwegs auf dem Smartphone. Jeder User erstellt für sich selbst einen Account mit E-Mail-Adresse und Benutzerpasswort, mit welchem er oder sie sich später über das Web einloggen kann und Zugriff auf den ganz persönlichen Notenerfassungsbereich erhält.

3.2.2 Anwenderfunktionalität

Sobald man das Interface der Anwendung im Webbrowser erreicht hat, kann sich die Benutzerin oder der Benutzer einen neuen Account erstellen oder sich mit bestehenden Benutzerdaten anmelden. Nach erfolgreicher Anmeldung landet man im Hauptbereich der Applikation. Es sind unter anderem Schaltflächen wie «Dashboard», «Profil» und «Logout» vorhanden. Darüber hinaus hat der User einen direkten Überblick über seine bereits eingetragenen Noten und die Möglichkeit, gleich neue Noten hinzuzufügen. Es können ebenfalls Einstellungen aufgerufen werden. Die Optionen welche zur Verfügung stehen in der Anwendung sind alle selbsterklärend und verfügen über eine grafisch passende Schaltfläche.

Login:

Falls sie noch keinen Account erstellt haben müssen Sie zuerst auf Register klicken. Ansonsten können Sie sich mit Ihrer Email und Ihrem Passwort einloggen.

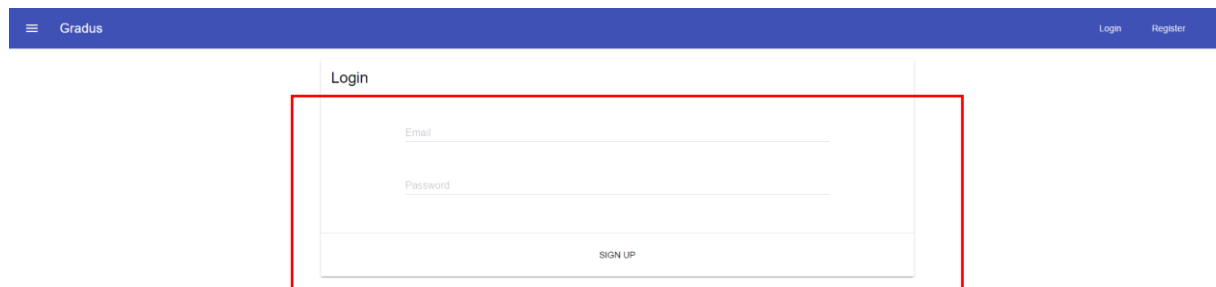


Abbildung 10: Login-Seite

Registration:

Falls Sie nicht bereits einen Account erstellt haben, können Sie dies hier tun. Sie müssen nur Ihren Namen, Vornamen, Ihre Email und das gewünschte Passwort eingeben. Anschliessend können Sie sich damit bei Gradus registrieren.

The registration form is titled 'Register' and is located within a blue header bar that also contains a menu icon and the text 'Gradus'. On the right side of the header, there are links for 'Login' and 'Register'. The form itself has five input fields: 'Firstname', 'Lastname', 'Email', 'Password', and 'Repeat Password'. At the bottom of the form is a 'SIGN UP' button.

Abbildung 11: Registrations-Seite

Profil:

Auf dieser Seite sehen Sie Ihre Benutzerinformationen und über das Stift Symbol können Sie diese bearbeiten. Beim betätigen des Papierkorbes wird Ihr Benutzeraccount gelöscht.

The user profile page is titled 'User information' and is located within a blue header bar that also contains a menu icon and the text 'Gradus'. On the right side of the header, there are links for 'Dashboard', 'Profile', and 'Logout'. The page is divided into two main sections: 'User information' and 'Actions'. The 'User information' section displays the user's details: 'firstname: Manuel', 'lastname: Bieri', and 'email: bierimanu@gmail.com'. The 'Actions' section contains two icons: a pencil icon for editing and a trash can icon for deleting the account. Below these sections is a 'Subjects' section that lists the user's selected subjects: 'Mathematik', 'Deutsch', 'Englisch', and 'M306'. The 'Mathematik' and 'Deutsch' subjects are marked with checkboxes and their respective IDs 'M946' and 'M947' are shown below them.

Abbildung 12: Profil-Seite

Note hinzufügen:

Hier können Sie neue Noten hinzufügen und speichern. Nebst den Noten selbst können auch noch Informationen wie ein Name, die Gewichtung der Note, sowie das Datum und sogar eine Beschreibung hinzugefügt werden. Durch Knopfdruck auf «ADD» werden Ihre Eingaben bestätigt. Beinahe dieselbe Ansicht bekommen Sie auch beim Noten bearbeiten, daher wird sie nicht nochmals dargestellt.

Add grade

Name

Grade

Weight

Date

Description

ADD

Abbildung 13: Seite zum Hinzufügen neuer Noten

Fach hinzufügen:

Auf dieser Seite können neue Fächer hinzugefügt werden. Dazu geben Sie bloss den Namen des Fachs, sowie den der Institution an. Durch einen abschliessenden Klick auf «ADD», wird das neue Fach mit Ihren Angaben gespeichert. Beinahe dieselbe Ansicht bekommen Sie auch beim Fach bearbeiten, daher wird sie nicht nochmals dargestellt.

Add subject

Name

Institution

ADD

Abbildung 14: Seite zum Hinzufügen neuer Fächer

Semester hinzufügen | bearbeiten:

Ein Semester können Sie hier hinzufügen, indem Sie dessen Namen eingeben und danach mit Knopfdruck auf «ADD» bestätigen. Beinahe dieselbe Ansicht bekommen Sie auch beim Semester bearbeiten, daher wird sie nicht nochmals dargestellt.

Add semester

Name

ADD

Abbildung 15: Seite zum Hinzufügen neuer Semester

Dashboard Übersicht:

Hier befinden Sie sich auf der Übersichtsseite Ihrer Noten. Zu jeder Note sieht man das Fach, das Semester und den Durchschnitt aller Noten eines Fachs. Hier stehen Ihnen wieder die Aktionen zum Bearbeiten oder Löschen zur Verfügung. Einen Eintrag bearbeiten können Sie, indem Sie am Ende einer Zeile auf das Stift-Symbol klicken. Wenn Sie eine Zeile löschen wollen, so klicken Sie einfach auf das Papierkorb-Symbol am Ende der entsprechenden Zeile.

Welcome, User!

+







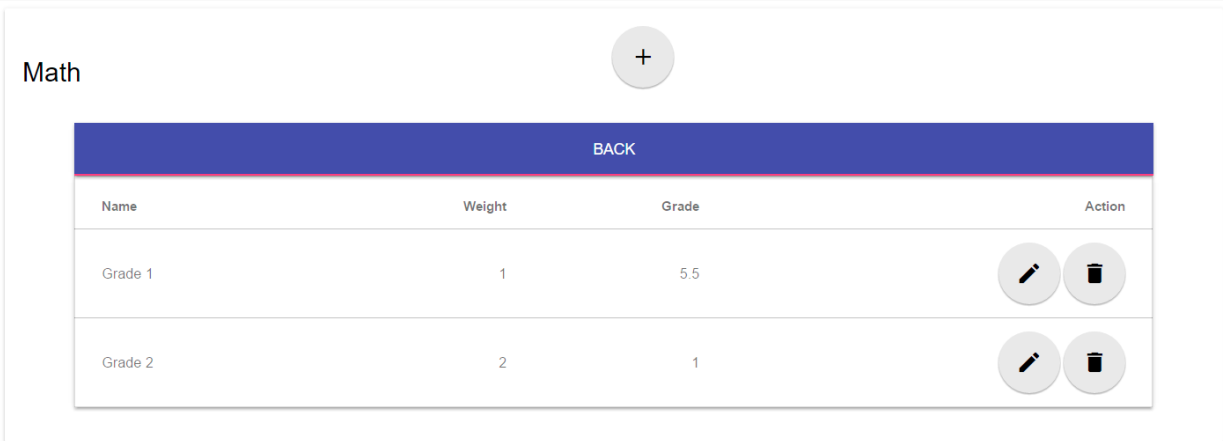
ALL	1. SEMESTER	2. SEMESTER	3. SEMESTER
Subject	Weight	Average	Action
Math	1	5.5	 
German	2	1	 
Informatics	1	6	 

Abbildung 16: Dashboard Übersichts-Seite

Fach Übersicht:

Für die Fächer haben Sie jeweils eine separate Übersicht. Hier werden Ihnen alle Noten zum ausgewählten Fach angezeigt. Auch hier besteht wieder die Möglichkeit, Zeilen zu bearbeiten oder löschen. Einen Eintrag bearbeiten können Sie, indem Sie am Ende einer Zeile auf das Stift-Symbol klicken. Wenn Sie eine Zeile löschen wollen, so klicken Sie einfach auf das Papierkorb-Symbol am Ende der entsprechenden Zeile.








BACK			
Name	Weight	Grade	Action
Grade 1	1	5.5	 
Grade 2	2	1	 

Abbildung 17: Fach Übersichts-Seite

Bei Fehler oder Fehlermeldung:

Aktualisieren Sie die Web-Applikation, indem Sie auf der Tastatur F5 betätigen oder im Webbrowser auf den

 - Knopf drücken (befindet sich meistens gleich rechts von der URL bar). Sollte dies nicht gereicht haben, versuchen Sie sich abzumelden und die Web-Applikation zu schliessen und sich danach erneut anzumelden. Sie können ebenfalls den Verlauf, Cookies und andere Website-Daten in den Browsereinstellungen löschen und es danach erneut probieren.

3.3 Integrations- und Installationshandbuch

3.3.1 Anwender

Um unsere Anwendung zu benutzen müssen zuerst folgende Kriterien eintreffen.

- Ein internetfähiges Gerät muss im Einsatz sein
- Ein Browser, wie Google Chrome muss installiert sein
- Es besteht eine Internetverbindung

Treffen zumindest mal diese Faktoren ein, ist es ein nächster Schritt Gradus zu installieren.

- Systeminstallation durchführen.

Zu einem späteren Zeitpunkt, anfangs 2017, wird es möglich sein Gradus über diese URL aufzurufen und zu nutzen. Dies wird ein Dienst sein, welcher jeder nutzen kann und somit nicht ein eigener Server für die Applikation betrieben werden muss: <https://iet-gibb-gradus.mirioeggmann.ch>

Wenn Gradus erfolgreich installiert wurde oder bereits anfangs 2017 ist und die Webseite online verfügbar ist, können sie sich nun anmelden oder falls sie noch kein Konto haben registrieren.

3.3.2 Systeminstallation

3.3.2.1 Benötigte Software:

Tabelle 8 Benötigte Software

Software	Beschreibung	Link
git	Benötigt es um dem Projekt einen Beitrag zu leisten und um immer die neuste Version herunterladen zu können. Weiter wird die Git Bash benötigt um die angular-cli zu bedienen.	https://git-scm.com/downloads
Java JDK 8	Java wird für das Spring Backend benötigt und muss somit installiert werden.	http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
NodeJS	Es wird die neuste NodeJS Version benötigt, damit Angular-cli funktioniert und die Projekt-Abhängigkeiten heruntergeladen werden können.	https://nodejs.org
Angular-cli	Diese wird benötigt um das Frontend zu starten. Es wäre theoretisch auch mit npm start möglich, jedoch ist es umständlicher.	https://github.com/angular/angular-cli
MySQL Server	Der MySQL Server wird benötigt um die Datenbank der Applikation zu betreiben.	https://dev.mysql.com/downloads/file/?id=466001
MySQL Workbench	Dies kann genutzt werden um einen Einblick in das MySQL Datenbanksystem zu bekommen.	http://dev.mysql.com/downloads/workbench/
IntelliJ IDEA	Mit dieser Entwicklungsumgebung wurde entwickelt und es	https://www.jetbrains.com/idea/

	wird empfohlen dies weiter zu führen.	
Tomcat 8	Dies ist der Server auf dem die Java Spring Boot Applikation ausgeführt wird.	https://tomcat.apache.org/download-80.cgi

3.3.2.2 Installation

Datenbank

Voraussetzungen:

- MySQL Server installiert

Vorgehen:

- Im MySQL Server einloggen als root und ein neues DB Schema «gradus» erstellen.
- Einen neuen Benutzer anlegen mit dem Namen «gradus» und Passwort «gradus»
- Dem Benutzer «gradus» alle Rechte auf das Schema «gradus» geben.

Backend

Voraussetzungen:

- IntelliJ IDEA installiert
- Tomcat 8 wurde installiert
- Git installiert
- Java JDK 8 installiert

Vorgehen:

- Projekt Klonen, falls nicht bereits gemacht, mit der Git Bash: **git clone** <https://github.com/mirioeggmann/gradus.git>
- Projekt konfigurieren, das es den Tomcat 8 Server benutzt
- Projekt mit IntelliJ starten.

Frontend

Voraussetzungen:

- Git installiert
- NPM installiert
- Angular-cli installiert

Vorgehen:

- Projekt Klonen, falls nicht bereits gemacht, mit der Git Bash: **git clone** <https://github.com/mirioeggmann/gradus.git>
- Im Ordner /frontend **npm install** ausführen
- Frontend starten mit **ng serve**

Nun sollten sie in der Lage sein über localhost:4200 auf Gradus zuzugreifen.

3.4 Supporthandbuch

Gradus wird von keiner Support-Organisation unterstützt und somit bieten wir keinerlei Support für die Applikation. Wenn eine dringende Frage offensteht, können Sie mirio.eggmann@protonmail.ch kontaktieren.

4 Test

4.1 Unit Test

In unserer Applikation wurden keine Unit-Tests definiert. Die Funktionalität wurde mit Hilfe der vordefinierten Tests sichergestellt.

4.2 Systemtest - Testspezifikation

4.2.1 Kritikalität der Funktionseinheit

Die Tests werden mit der Stufe «hoch» oder «niedrig» eingestuft.
Ist ein Testfall positiv durchlaufen, so gilt sein Ausmass als «niedrig».
Ist ein Testfall negativ eingestuft, so gilt sein Ausmass als «hoch».

Das Resultat ist im [Testprotokoll](#) ersichtlich.

4.2.2 Testanforderungen

Die Tests sind mit fehlerfreien Werten durchzuführen.
Die Tests sind unter Normalbedingungen auszuführen.

4.2.3 Testverfahren

Das Testing braucht keine spezielle Vorbereitung. Die Schritte, so wie die erwarteten Resultate sind untenstehend aufgeführt. Die Daten werden während des Testdurchganges vom Tester erstellt.

4.2.4 Testkriterien

4.2.4.1 Abdeckungsgrad:

Alle Tests sind gemäss Testprotokoll durchzuführen. Es wird direkt mit der Applikation getestet.
Zum Testen werden Äquivalenzwerte verwendet (Stichprobenartig). Alle funktionalen Anforderungen (UserStories) müssen abgedeckt sein.

4.2.4.2 Checklisten:

Als Checkliste gilt das Testkonzept.

4.2.4.3 Endkriterien:

Alle Testfälle sind erfolgreich durchgeführt worden. Die Testperson hat das Testprotokoll korrekt ausgefüllt und unterzeichnet.

4.2.5 Testfälle

Tabelle 9 Testfälle

Testfall	User Story
Account erstellen	Ich als Benutzer kann einen neuen Account erstellen.
Anmeldung	Ich als Benutzer kann mich mit meinem Account anmelden.
Noten Erfassen	Ich als Benutzer kann Noten erfassen.
Fach hinzufügen	Ich als Benutzer kann ein Fach hinzufügen.
Note bearbeiten	Ich als Benutzer kann falsche Noten bearbeiten.
Fach bearbeiten	Ich als Benutzer kann einen Fachnamen ändern.
Benutzer bearbeiten	Ich als Benutzer kann meine Profildaten bearbeiten.
Note löschen	Ich als Benutzer kann eine unnötige Note löschen.
Fach löschen	Ich als Benutzer kann ein Fach löschen.
Benutzer löschen	Ich als Benutzer kann meinen Account löschen.
Semester hinzufügen	Ich als Benutzer kann ein Semester hinzufügen.
Semester Bearbeiten	Ich als Benutzer kann ein Semester Bearbeiten.
Semester löschen	Ich als Benutzer kann ein Semester löschen.

4.3 Testprozedur

4.3.1 Account Erstellen

Tabelle 10 Testfall: Account erstellen

Beschreibung	Ein neuer Benutzer registriert sich.
Abgedeckte Anwendungsfälle	Registration
Ausgangssituation	Der Benutzer ist erstellt.
Vorbereitungsschritte	<ol style="list-style-type: none"> 1. Bis Anfangs 2017 muss für die URL noch ein Hosts Eintrag erstellt werden für https://iet-gibb-gradus.mirioeggmann.ch auf den localhost:4200 um die Applikation testen zu können. Erst dann ist sie online für alle verfügbar. 2. Starten Sie den Browser Google Chrome oder Mozilla Firefox. (Eine aktuelle Version ist empfohlen)
Testschritte	Erwartetes Resultat
1. Auf dem Testclient im Browser die URL https://iet-gibb-gradus.mirioeggmann.ch/ eingeben	Die Startseite des Notenerfassungstools wird angezeigt.
2. Den Link «Register» wählen.	Das Registrierungsformular wird angezeigt.
3. Eingeben der folgenden Informationen: <ol style="list-style-type: none"> a. Vorname: Gibb b. Nachname: lx c. Email: gibb@ix.ch d. Passwort: gibbiX1234 e. Passwortwiederholung: gibbiX1234 	
4. Den Button «Sign Up» wählen	Das Registrierungsformular wird erneut angezeigt.
5. Der Benutzer wird automatisch zur Login Seite weitergeleitet.	
6.	In der Datenbank ist in der Tabelle user ein neuer Record mit dem Namen Test1 vorhanden.

4.3.2 Anmelden

Tabelle 11 Testfall: Anmelden

Beschreibung	Ein sich bereits registrierter Benutzer meldet sich an.
Abgedeckte Anwendungsfälle	Login
Ausgangssituation	Der Benutzer ist angemeldet und befindet sich auf der Hauptseite.
Vorbereitungsschritte	<ol style="list-style-type: none"> 1. Bis Anfangs 2017 muss für die URL noch ein Hosts Eintrag erstellt werden für https://iet-gibb-gradus.mirioeggmann.ch auf den localhost:4200 um die Applikation testen zu können. Erst dann ist sie online für alle verfügbar. 2. Der Benutzer ist bei Gradus erfolgreich registriert.
Testschritte	Erwartetes Resultat
1. Auf dem Testclient im Browser die URL https://iet-gibb-gradus.mirioeggmann.ch/ eingeben	Die Startseite des Notenerfassungstools wird angezeigt.
2. Eingeben der folgenden Informationen: <ol style="list-style-type: none"> a. Email: gibb@ix.ch b. Passwort: gibbiX1234 	
3. Den Button «Sign Up» wählen	Der Benutzer wird automatisch zu seiner Homepage weitergeleitet.

4.3.3 Note hinzufügen

Tabelle 12 Testfall: Note hinzufügen

Beschreibung	Ein Benutzer fügt eine Note hinzu.	
Abgedeckte Anwendungsfälle	Note erfassen	
Ausgangssituation	Der Benutzer hat eine Note hinzugefügt.	
Vorbereitungsschritte	1. Der Benutzer ist auf der Startseite. 2. Der Benutzer ist angemeldet.	
Testschritte	Erwartetes Resultat	
1. Der Benutzer klickt bei der Burger Navigation auf «Add Grade».	Eine Maske erscheint.	
2. Eingeben der folgenden Informationen: a. Name: 1. Note b. Grade: 6.0 c. Weight: 100 d. Date: 11.12.2016 e. Description: Test Note		
3. Den Button «ADD» wählen	Der Benutzer wird automatisch zu seiner Übersicht weitergeleitet.	
4.	In der Datenbank ist in der Tabelle grade ein neuer Record mit dem Namen 1. Note vorhanden.	

4.3.4 Fach hinzufügen

Tabelle 13 Testfall: Fach hinzufügen

Beschreibung	Ein Benutzer fügt ein Fach hinzu.	
Abgedeckte Anwendungsfälle	Fach erstellen	
Ausgangssituation	Der Benutzer hat ein Fach hinzugefügt.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.	
Testschritte	Erwartetes Resultat	
1. Der Benutzer klickt bei der Burger Navigation auf «Add subject».	Eine Maske erscheint.	
2. Eingeben der folgenden Informationen: a. Name: Fach 1 b. Institution: GIBB		
3. Den Button «ADD» wählen	Der Benutzer wird automatisch zu seiner Übersicht weitergeleitet.	
4.	In der Datenbank ist in der Tabelle subject und institution ein neuer Record mit dem Namen Fach 1 vorhanden.	

4.3.5 Note bearbeiten

Tabelle 14 Testfall: Note bearbeiten

Beschreibung	Ein Benutzer bearbeitet eine Note.	
Abgedeckte Anwendungsfälle	Note erfassen	
Ausgangssituation	Der Benutzer hat seine Note angepasst.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist bei der Notenübersicht.	
Testschritte	Erwartetes Resultat	
1. Der Benutzer klickt bei der Übersicht auf den Stift.	Eine Maske erscheint.	
2. Der Benutzer ändert folgende Attribute: a. Name: 2. Note b. Grade: 5.7		
3. Der Benutzer klickt auf den Button «Save».		
4.	In der Datenbank ist in der Tabelle grade der Eintrag von 1. Note zu 2. Note geändert worden.	

4.3.6 Fach bearbeiten

Tabelle 15 Testfall: Fach bearbeiten

Beschreibung	Ein Benutzer bearbeitet ein Fach.	
Abgedeckte Anwendungsfälle	Fach erstellen	
Ausgangssituation	Der Benutzer hat sein Fach angepasst.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.	
Testschritte		Erwartetes Resultat
1. Der Benutzer klickt bei der Übersicht auf den Stift.		Eine Maske erscheint.
2. Der Benutzer ändert folgende Attribute: a. Name: Fach 2		
3. Den Button «ADD» wählen		Der Benutzer wird automatisch zu seiner Übersicht weitergeleitet.
4.		In der Datenbank ist in der Tabelle subject der Eintrag von Fach 1. zu Fach 2 geändert worden.

4.3.7 Benutzer bearbeiten

Tabelle 16 Testfall: Benutzer bearbeiten

Beschreibung	Ein Benutzer fügt eine Note hinzu.	
Abgedeckte Anwendungsfälle	Registrieren	
Ausgangssituation	Der Benutzer ändert seine Angaben.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.	
Testschritte		Erwartetes Resultat
1. Der Benutzer klickt in der Navigation auf «Profile».		Der Benutzer landet bei seinen Account Informationen.
2. Der Benutzer klickt im Feld Actions auf den Stift.		Eine Maske erscheint.
3. Der Benutzer ändert folgende Attribute: a. Name: Gibbs		
4. Den Button «Save» wählen		Der Benutzer wird automatisch zurück zu seiner Übersicht weitergeleitet.
5.		In der Datenbank ist in der Tabelle user der Eintrag von Gibb zu Gibbs geändert worden.

4.3.8 Note löschen

Tabelle 17 Testfall: Note löschen

Beschreibung	Ein Benutzer löscht eine Note.	
Abgedeckte Anwendungsfälle	Note erfassen	
Ausgangssituation	Der Benutzer hat seine Note gelöscht.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist bei der Notenübersicht.	
Testschritte		Erwartetes Resultat
1. Der Benutzer klickt bei der Übersicht auf den Papierkorb.		Ein Popup Fenster erscheint. Der Benutzer wird gefragt ob er die Note wirklich löschen will.
2. Der Benutzer bestätigt das PopUp mit «ok».		
3.		In der Datenbank ist in der Tabelle grade der Eintrag gelöscht worden.

4.3.9 Fach löschen

Tabelle 18 Testfall: Fach löschen

Beschreibung	Ein Benutzer löscht ein Fach.
Abgedeckte Anwendungsfälle	Fach erstellen
Ausgangssituation	Der Benutzer hat sein Fach gelöscht.
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.
Testschritte	Erwartetes Resultat
1. Der Benutzer klickt bei der Übersicht auf den Papierkorb.	Ein Popup Fenster erscheint. Der Benutzer wird gefragt ob er das Fach wirklich löschen will.
2. Der Benutzer bestätigt das PopUp mit «ok».	
3.	In der Datenbank ist in der Tabelle subject der Eintrag gelöscht worden.

4.3.10 Benutzer löschen

Tabelle 19 Testfall: Benutzer löschen

Beschreibung	Ein Benutzer löscht sein Account.
Abgedeckte Anwendungsfälle	Registrieren
Ausgangssituation	Der Account wurde gelöscht.
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.
Testschritte	Erwartetes Resultat
1. Der Benutzer klickt in der Navigation auf «Profile».	Der Benutzer landet bei seinen Account Informationen.
2. Der Benutzer klickt im Feld Actions auf den Papierkorb.	Ein Popup Fenster erscheint. Der Benutzer wird gefragt ob er seinen Account wirklich löschen will.
3. Der Benutzer bestätigt das PopUp mit «ok».	Der Benutzer wird auf die Login Seite weitergeleitet.
4.	In der Datenbank ist in der Tabelle user der Eintrag gelöscht worden.

4.3.11 Semester hinzufügen

Tabelle 20 Testfall: Semester hinzufügen

Beschreibung	Ein Benutzer fügt ein Semester hinzu.
Abgedeckte Anwendungsfälle	Semester hinzufügen
Ausgangssituation	Ein neues Semester wurde erfasst.
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.
Testschritte	Erwartetes Resultat
1. Der Benutzer klickt bei der Burger Navigation auf «Add semester».	Eine Maske erscheint.
2. Eingeben der folgenden Informationen: a. Name: 1. Semester	
3. Den Button «ADD» wählen	Der Benutzer wird automatisch zu seiner Übersicht weitergeleitet.
4.	In der Datenbank ist in der Tabelle semester der Eintrag vom 1. Semester erstellt worden.

4.3.12 Semester bearbeiten

Tabelle 21 Testfall: Semester bearbeiten

Beschreibung	Ein Benutzer bearbeitet ein Semester hinzu.	
Abgedeckte Anwendungsfälle	Semester bearbeiten	
Ausgangssituation	Ein Semester wurde bearbeitet.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.	
Testschritte		Erwartetes Resultat
1. Der Benutzer klickt bei der Übersicht auf den Stift.		Eine Maske erscheint.
2. Der Benutzer ändert folgende Attribute: a. Name: 2. Semester		
3. Den Button «ADD» wählen		Der Benutzer wird automatisch zu seiner Übersicht weitergeleitet.
4.		In der Datenbank ist in der Tabelle semester der Eintrag vom 1. Semester zu 2. Semester geändert worden.

4.3.13 Semester löschen

Tabelle 22 Testfall: Semester löschen

Beschreibung	Ein Benutzer löscht ein Semester.	
Abgedeckte Anwendungsfälle	Semester löschen	
Ausgangssituation	Das Semester wurde gelöscht.	
Vorbereitungsschritte	1. Der Benutzer ist angemeldet. 2. Der Benutzer ist auf der Startseite.	
Testschritte		Erwartetes Resultat
1. Der Benutzer klickt bei der Übersicht auf den Papierkorb.		Ein Popup Fenster erscheint. Der Benutzer wird gefragt ob er das Fach wirklich löschen will.
2. Der Benutzer bestätigt das PopUp mit «ok».		
3.		In der Datenbank ist in der Tabelle semester der Eintrag gelöscht worden.

4.4 Testprotokoll

Systemtest 1

Getestete Version: 1.0
Tester: Dario Menzel
Datum, Zeit: 11.12.2016, 17:00 – 18:00

Tabelle 23 Testprotokoll

Testfall	Erfüllt	Bemerkung
1. Account erstellen	<input checked="" type="checkbox"/>	
2. Anmeldung	<input checked="" type="checkbox"/>	
3. Noten Erfassen	<input checked="" type="checkbox"/>	
4. Fach hinzufügen	<input checked="" type="checkbox"/>	
5. Note bearbeiten	<input checked="" type="checkbox"/>	
6. Fach bearbeiten	<input checked="" type="checkbox"/>	
7. Benutzer bearbeiten	<input checked="" type="checkbox"/>	
8. Note löschen	<input checked="" type="checkbox"/>	
9. Fach löschen	<input checked="" type="checkbox"/>	
10. Benutzer löschen	<input checked="" type="checkbox"/>	
11. Semester hinzufügen	<input checked="" type="checkbox"/>	
12. Semester Bearbeiten	<input checked="" type="checkbox"/>	
13. Semester löschen	<input checked="" type="checkbox"/>	

5 Antrag auf Freigabe der nächsten Projektphase

Das Projektteam empfiehlt dem Auftraggeber die Freigabe der Phase Einführung.

Anhang

A) Quellcode

Der Quellcode befindet sich auf <https://github.com/mirioeggmann/gradus>

Laufend werden dann noch Aktualisierungen vorgenommen und können dort eingesehen werden.