

---

# **Final Report**

**PR Multimediale Systeme  
WS/SS 2015/2016 – Gruppe K2B**

---

---

## Document Control

### Contributors

Person	Role	Company	Contribution
Uygur Senocak		UNIVIE	Template draft

### Revision History

Issue	Author	Date	Description
V0.1	Uygur Senocak	21-Marz-2016	1 <sup>st</sup> version written
V0.2	Uygur Senocak	12-Mai-2016	2 <sup>st</sup> version written
V0.3	Uygur Senocak	13-Jun-2016	3 <sup>st</sup> version written

### Table of Contents

1	Objectives .....	3
2	Features .....	3
3	System Architecture .....	4
3.1	Design Overview .....	4
3.2	Detailed Design .....	4
3.3	Technologies and Third-Party Software .....	10
4	User Documentation .....	11
4.1	Installation Guidelines .....	11
4.2	Quick Start Tutorial .....	11
5	Progress Report .....	11
5.1	Summary .....	11
5.2	Status per Milestone .....	12
5.3	Risk Analysis .....	12
5.4	Activities per Person .....	13
6	Feedback .....	13

---

## 1 Objectives

*This section describes the project's overall objectives and contains an updated and final version of the ones given within the requirements document.*

*Dieses Projekt wird für die Studenten von der Universität-Wien entwickelt. Das bedeutet, dass wenn ein User dieses System benutzen will, dann braucht er eine Emailadresse von der Universität-Wien, weil das System während der Anmeldung mit einer Bestätigung überprüfen wird, ob der User eine realen Emailadresse von der Universität hat. Ein angemeldeter User kann sich in der System mit einem oder mehreren Studienrichtungen und Lehrfächern anmelden. Nachdem Anmeldung der Lehrfächer wird das System automatisch einen Kalender erstellen und die Lehrfächer, die der Student ausgewählt haben, in diesen Kalender hinzufügen. Danach kann der User in diesen Kalender sie anschauen, wo und wann genau diese Lehrfächer stattfinden.*

*In der System kann ein User auch einen Semesterplan erstellen. Nachdem Auswählen des Studienrichtungens wird das System die getroffene Module auflisten. Da kann ein User einfach es eintragen, in welcher Semester welche Module teilnehmen möchte, so kann ein User für sich selbst einen Plan erstellen und danach kann er in diese Liste eintragen, ob er ein Module absolviert hat oder nicht. Wenn ein User für sich einen Semesterplan erstellt hat, dann kann er in diesen Plan sie auch anschauen, wie viele Module er insgesamt hat, wie viele er davon geschafft und wie viele er davon noch nicht geschafft.*

*Ein User von der System kann seinem Semesterplan einen anderen User absenden. Dafür braucht der User nur die Emailadresse von den anderen User eingeben, weil das System mit einem Email seinen Semesterplan schicken wird.*

---

## 2 Features

*This section contains a short description and a list summarizing which features have been fulfilled.*

Das Programm ist eine Web-Applikation. Ich habe es in Java geschrieben, deswegen kann man dieses Programm in jeder Plattform ausführen. Es ist auch in Objektorientierte Programmierung geschrieben, dafür habe ich die Entwurfsmuster „Model-View-Controllers“ benutzt.

Für die Backend habe ich die Technologie „Spring Boot“ benutzt, damit ich mich um viele Konfigurationen nicht kümmern werde. Für die Security habe ich die Technologie „Spring Security“ ausgewählt. Wenn ein User sich nicht registrieren wird, dann kann er das System nicht zugreifen. Um die Datenbank zu kommunizieren habe ich die Technologie „Spring Data JPA“ ausgewählt, weil die Kommunikation zwischen Backend und Datenbank mit dieser Technologie sehr einfach und ganz sicher ist. Dieses Projekt ist auch ein Maven-Projekt, das bedeutet, dass man dieses Programm ohne irgendwelche Abhängigkeiten von einer Bibliothek ausführen kann, da die gebrauchte Bibliotheken automatisch nachdem Ausführen des Programms heruntergeladen wird. Um eine XML-Datei zu lesen habe ich die Technologie „JAXB“ benutzt. Mit Hilfe dieser Library kann man XML Daten automatisiert in Java-Objekte / POJOs deserialisieren. Für die Datenbank habe ich „Postgresql“ benutzt.

Für die Frontend habe ich nur mit HTML5 und CSS3 gearbeitet. Um einen Button, Table, Panels usw. zu erstellen habe ich „Bootstrap“ benutzt, deswegen sind die Seiten automatisch Responsive.

Das bedeutet, dass man die Seiten mit einem Rechner oder mit einem Handy anschauen und benutzen kann. Ich habe für die Symbole die Technologie „Fontawesome“ benutzt. Um die Teil von Javascript zu vereinfachen habe ich JQuery benutzt. Mit JQuery habe ich einige Sachen mit der Technologie „Ajax“ implementiert.

---

## 3 System Architecture

---

### 3.1 Design Overview

*This part should describe the overall in terms of an architectural overview and a breakdown of the major software components (e.g. by functional area ("vertically") or by layer/tier ("horizontally"))*

This document is organised according to the main components of the system, as follows...

---

### 3.2 Detailed Design

#### 3.2.1 Component 1..n

##### 3.2.1.1 [Description](#)

Ich habe für dieses System die Entwurfsmuster "Model View Controller Pattern" benutzt. Aber ich habe für die weiteren Schritten zwei neue Package "Logik und Security" hinzugefügt, damit ich den Programm besser aufteilen kann. Logik ist mit der Sache "XML-Auslesen, die Daten in die Klassen hinzufügen, Datenbank speichern usw." beschäftigt und die Security beinhaltet drei verschiedene Klassen, die mit der Sicherheit beschäftigt sind.

##### 3.2.1.2 [Documentation](#)

Ich habe für dieses System 6 folgende Controller implementiert, die Controller sind Schnittstelle zwischen den Backend und Frontend.

1. CalendarController
2. LehrfachController
3. LoginController
4. ModuleAuswählenController
5. MyModulController
6. StudienrichtungAuswahl

Ich habe eine Klasse „MyXMLReader“ für meine Logik implementiert. Diese Klasse wird den Server von der Universität verbinden und die XML-Datei auslesen und die Daten in einer entsprechenden Klasse hinzufügen.

Ich habe für dieses System 6 eigene und für die Bibliothek „JAXB“ 12 Model-Klasse implementiert.

Eigene Klasse:

1. CalendarEvents
2. LehrfachAnmelden
3. ModulAuswahl
4. User
5. UserRole
6. UserStadiumInformation

Das System beinhaltet 11 folgende Interfaces „Repositorys“. Diese Repositorys beschäftigen die Daten in der Datenbank zu speichern oder auslesen.

1. CourseRepository
2. FachGebietRepository
3. GroupRepository

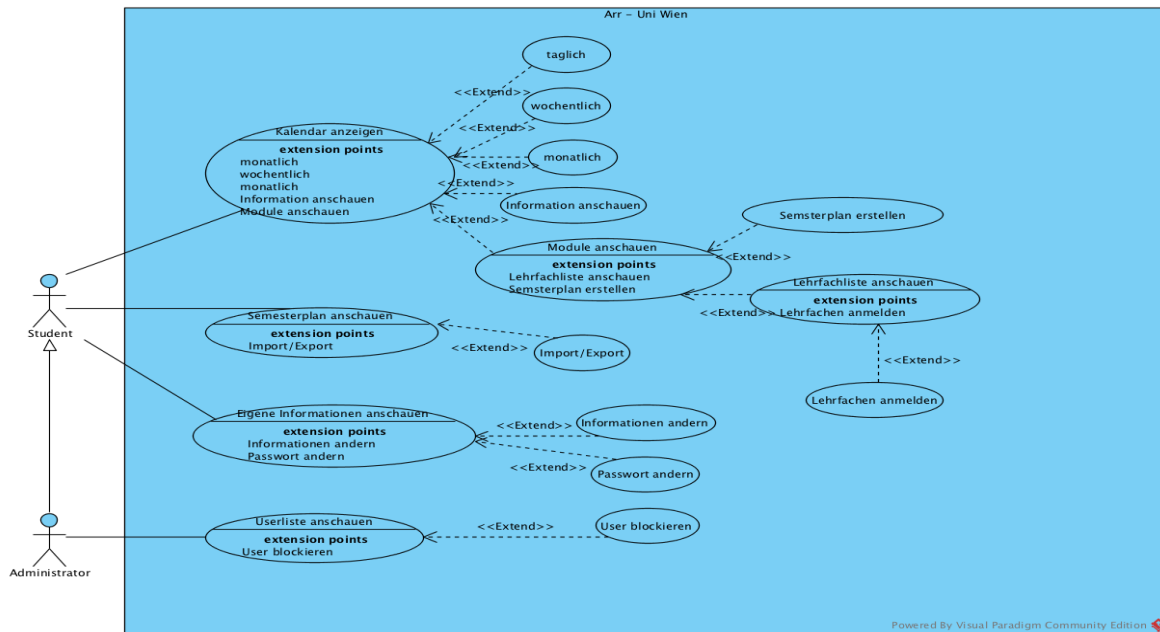
4. LehrfachAnmeldenRepository
5. ModulAuswahlRepository
6. ModulRepository
7. StudienRichtungRepository
8. UserRepository
9. UserRolesRepository
10. UserStudiumInformationRepository
11. ZeitRepository

Das System beinhaltet für die Security 3 folgende Klassen:

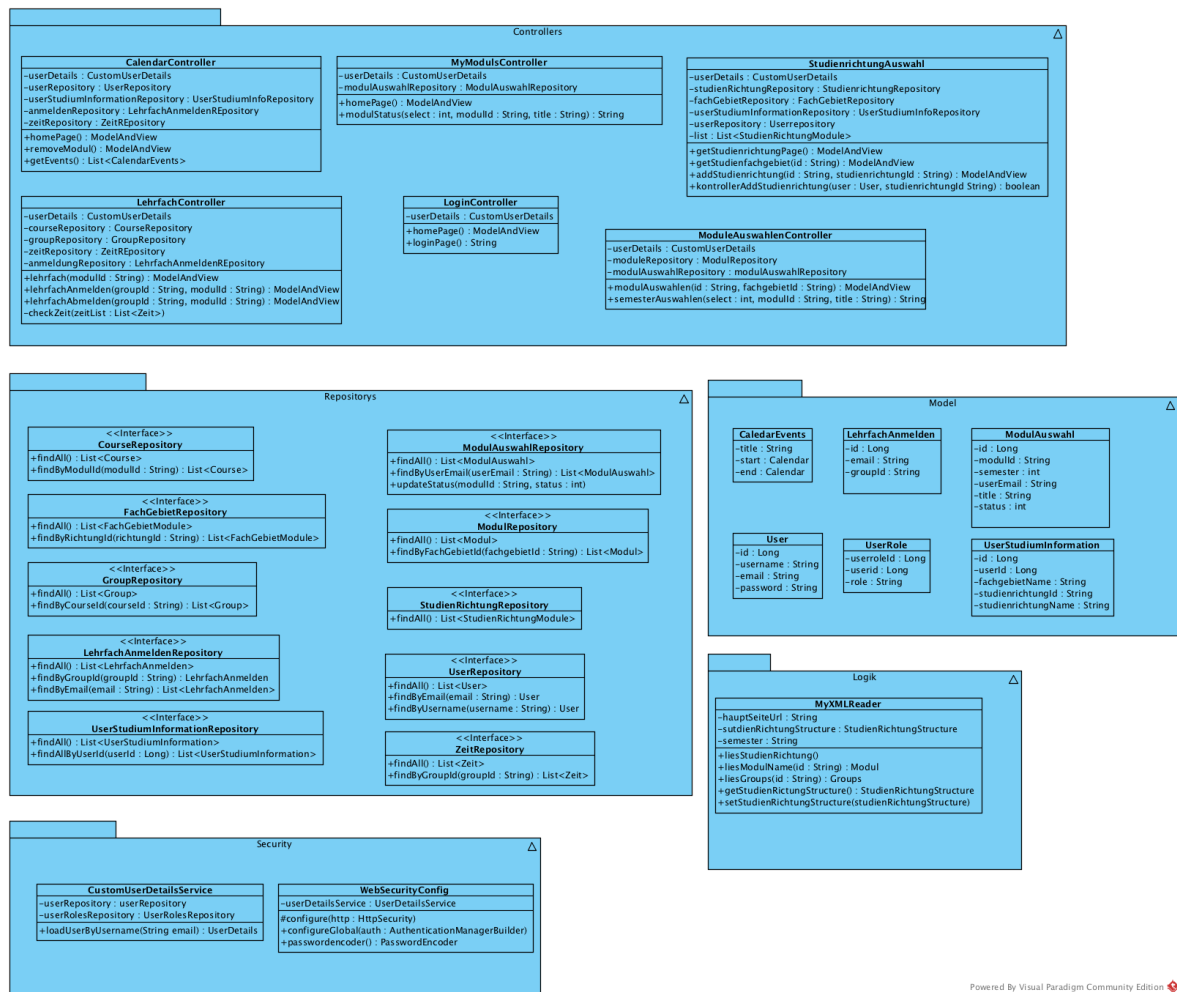
1. CustomUserDetails
2. CustomUserDetailsService
3. WebSecurityConfig

### 3.2.1.3 Architecture

#### Use Case Diagramm:



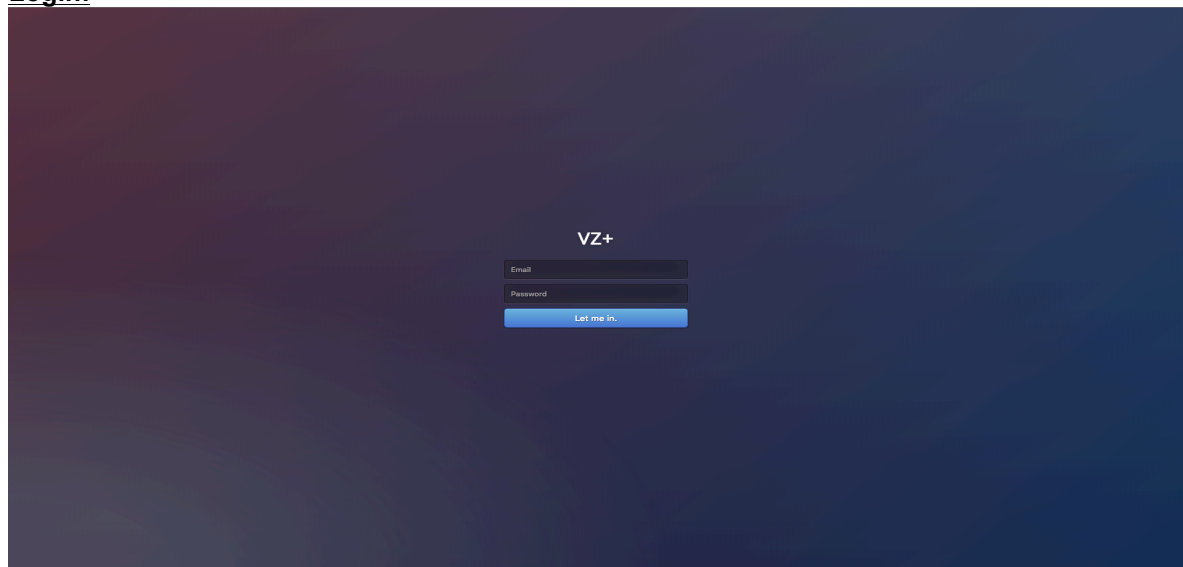
## Klassendiagramm:



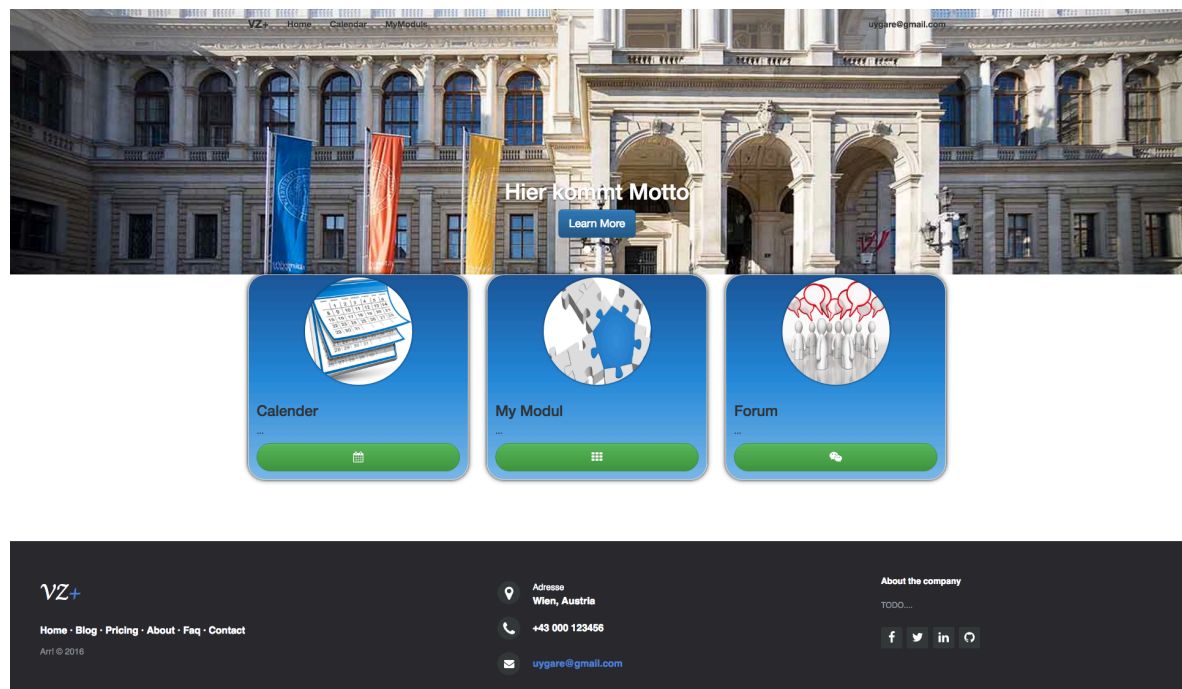
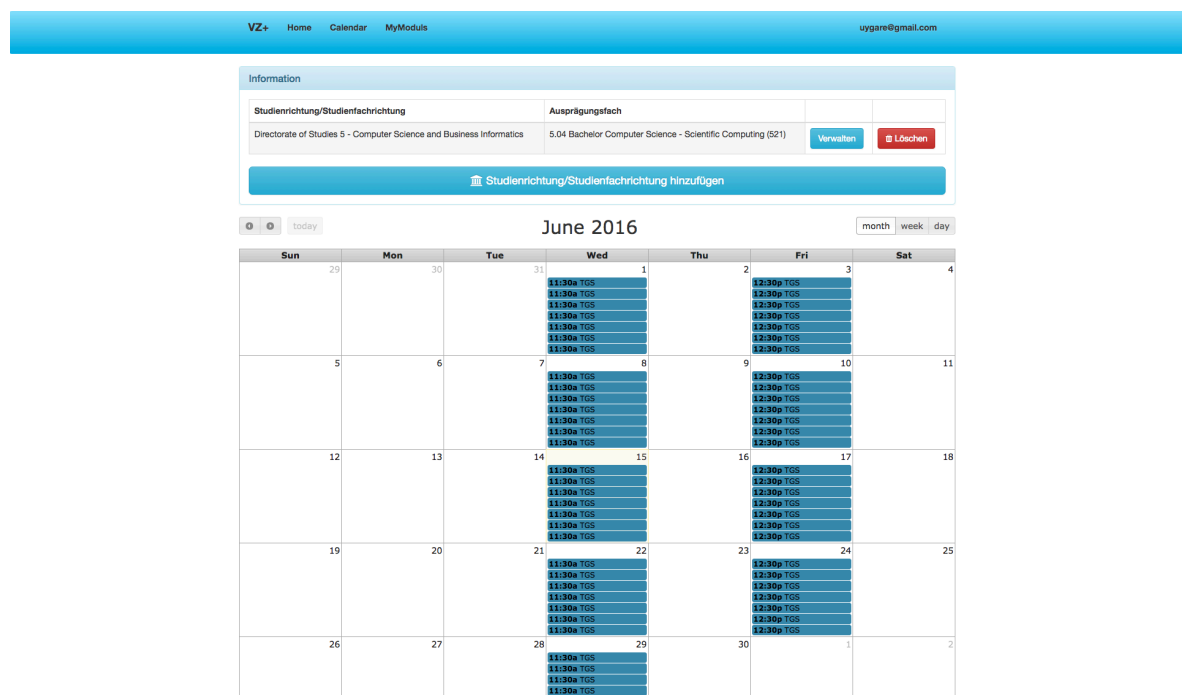
Powered By Visual Paradigm Community Edition

### 3.2.1.4 Screenshots

#### Login:



**Home-Seite:**

**Kalender:**

## Kalender taglich:

VZ+ Home Calendar MyModules

uygare@gmail.com

Information

Studienrichtung/Studienfachrichtung

Ausprägungsfach

Directorate of Studies 5 - Computer Science and Business Informatics

5.04 Bachelor Computer Science - Scientific Computing (S21)

Verwalten

Löschen

Studienrichtung/Studienfachrichtung hinzufügen

today

June 15, 2016

month week day

Wednesday

all-day

6am

7am

8am

9am

10am

11am

12pm

1pm

2pm

3pm

4pm

5pm

6pm

7pm

8pm

11:30 TGS

11:30 TGS

11:30 TGS

11:30 TGS

11:30 TGS

11:30 TGS

11:30 TGS

11:30 TGS

## Module-Auswählen:

VZ+ Home Calendar MyModules

uygare@gmail.com

Module

Scientific Computing and Formal and Natural Sciences (72 ECTS)

---Select Semester---

Lehrfach

5.04 Bachelor Computer Science - Scientific Computing (S21)

---Select Semester---

Lehrfach

STEOP Orientation Phase (18 ECTS)

---Select Semester---

Lehrfach

STEOP 1 - Module EIN Introduction to Computer Science (6 ECTS)

Semester: 1

---Select Semester---

Lehrfach

STEOP 2 - Module PRG Programming (6 ECTS)

Semester: 1

---Select Semester---

Lehrfach

STEOP 3 - Module TGS Technical Basics and System Software (6 ECTS)

Semester: 1

---Select Semester---

Lehrfach

Information Technology (36 ECTS)

Semester: 1

---Select Semester---

Lehrfach

Module ADG Algorithms and Data Structures (6 ECTS)

Semester: 2

---Select Semester---

Lehrfach

Module NET Computer and Communication Networks (6 ECTS)

Semester: 3

---Select Semester---

Lehrfach

Module THI Theoretical Computer Science (6 ECTS)

Semester: 3

---Select Semester---

Lehrfach

Module DBS Database Systems (6 ECTS)

Semester: 2

---Select Semester---

Lehrfach

Module SWE Software Engineering (6 ECTS)

Semester: 3

---Select Semester---

Lehrfach

Module SVA Software Architectures (6 ECTS)

Semester: 2

---Select Semester---

Lehrfach

General Basics (18 ECTS)

Semester: 2

---Select Semester---

Lehrfach

Module SOP Social Competence and Project Management (6 ECTS)

Semester: 3

---Select Semester---

Lehrfach

Module MOD Basics of Modelling (6 ECTS)

Semester: 2

---Select Semester---

Lehrfach

Module HCI Human-Computer-Interaction and Psychology (6 ECTS)

Semester: 4

---Select Semester---

Lehrfach

Structural Sciences (24 ECTS)

---Select Semester---

Lehrfach

Module MBT Basic Techniques of Mathematics (6 ECTS)

Semester: 1

---Select Semester---

Lehrfach

Module GMA Basics of Mathematics and Analysis (6 ECTS)

Semester: 2

---Select Semester---

Lehrfach



## Gruppe von einer Lehrfach Listen und Anmelden:

VZ+ Home Calendar MyModules uygare@gmail.com

Information

**Technical Basics and System Software** VO 4.00 ECTS

Group 050041-1

Anmelden Group 050041-1

**Technical Basics and System Software** UE 2.00 ECTS

Group 050042-1

Anmelden Group 050042-1

Group 050042-2

Beginn Zeit	End Zeit	Ort	Raum
Fri Mar 18 12:30:00 CET 2016	Fri Mar 18 13:15:00 CET 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Apr 08 12:30:00 CEST 2016	Fri Apr 08 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Apr 15 12:30:00 CEST 2016	Fri Apr 15 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Apr 22 12:30:00 CEST 2016	Fri Apr 22 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Apr 29 12:30:00 CEST 2016	Fri Apr 29 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri May 06 12:30:00 CEST 2016	Fri May 06 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri May 13 12:30:00 CEST 2016	Fri May 13 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri May 20 12:30:00 CEST 2016	Fri May 20 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri May 27 12:30:00 CEST 2016	Fri May 27 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Jun 03 12:30:00 CEST 2016	Fri Jun 03 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Jun 10 12:30:00 CEST 2016	Fri Jun 10 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Jun 17 12:30:00 CEST 2016	Fri Jun 17 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG
Fri Jun 24 12:30:00 CEST 2016	Fri Jun 24 13:15:00 CEST 2016	Währinger Straße 29	Seminarraum 6, Währinger Straße 29 1.OG

## Studienrichtung/Studienfachrichtung Auswählen:

VZ+ Home Calendar MyModules uygare@gmail.com

Information

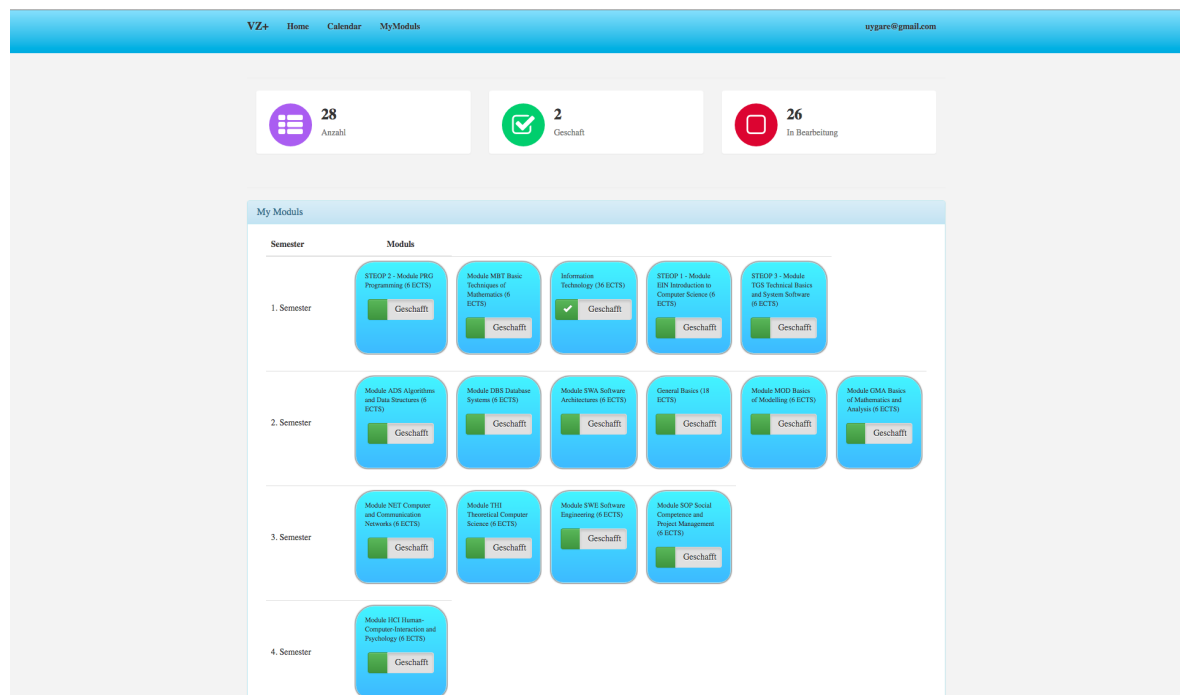
<b>Studienrichtung/Studienfachrichtung</b>	<b>Ausprägungsfach</b>	
Directorate of Studies 5 - Computer Science and Business Informatics	5.04 Bachelor Computer Science - Scientific Computing (521)	<a href="#">Löschen</a>

**Studienrichtung/Studienfachrichtung hinzufügen**

Nothing selected

[Wähle Ausprägungsfach](#)

## Semesterplan:



### 3.2.1.5 [Objects and Actions](#)

### 3.2.1.6 [Noteworthy](#)

Ich glaube dass es noch viele Sachen für dieses System gibt, dass man in der weitere Schritten entwickeln kann und ich denke dass wenn man in der Universität studiert dann kann man diese Idee brauchen wird. Ich muss auch für die Technologie „Spring“ sagen, dass es wirklich sehr hilfreich und sehr gute Technologie um einen Web-Application zu entwickeln. Was man wirklich für solche Systemen braucht, bietet Spring dafür einen einfachen Möglichkeit.

## 3.3 Technologies and Third-Party Software

*Which technologies have been used for building the system? This section should define a list of third-party software which is used within any software component of the preceding chapters. This should contain a brief description, its functionality, the possible features which will get used, issues as licensing, etc. and its intended use within the system. This should also include reasons why alternatives are rejected.*

1. Spring Boot
2. Spring Data JPA
3. Spring Security
4. PostgreSQL
5. Bootstrap
6. Thymeleaf
7. JAXB
8. HTML 5 / CSS3
9. Javascript
10. JQuery

---

## 4 User Documentation

---

### 4.1 Installation Guidelines

Wenn man das System in der eigenen Computer installieren will, dann soll man zuerst einen Datenbank von "Postgresql" erstellen und in der Konfigurationsdatei "application.properties" von der System die Zugriffsdaten von der Datenbank richtig eingeben. Sonst kann der System nicht mit der Datenbank kommunizieren. Dafür kann man einen IDE zb. "Eclipse" benutzen. Man kann mit einen IDE den Git-Repository verbinden und den Projekt herunterladen. Danach kann man in der Ordner "src/main/resources" die Konfigurationsdatei "application.properties" erreichen. Wenn man erstes mal den Programm ausführen und die Daten von der Uni-Server auslesen will, dann soll man in der Klasse "App.java" einige Parts von der Code, die gerade Kommentar sind, nicht mehr als Kommentar einstellen, damit das Programm das Ausführen des Programms die entsprechende Funktionalitäten aufrufen und die Daten auslesen wird. Danach kann man einfach den Programm ausführen. Das Programm benutzt für die Bibliotheken Maven, deswegen wird das Programm zuerst einige Bibliotheken herunterladen.

---

### 4.2 Quick Start Tutorial

Wenn man den Datenbank erstellt und in der Konfigurationsdatei geändert hat, dann kann man entweder direkt mit der IDE das Programm ausführen oder man soll zuerst mit der Hilfe von Maven einen JAR-Datei erstellen, damit man ohne IDE das Programm ausführen kann. Danach kann man in einer Browser die Adresse "localhost:8080" eingeben. Wenn man alles richtig gemacht hat, dann bekommt man die Index-Seite von der Programm. Hier soll man sich zuerst mit einen Emailadresse von der Universität Wien registrieren. Danach kann man sich einloggen. In der Homeseite hat man zwei Möglichkeiten, entweder kann man seinen eigene Kalender-Seite eingehen und die Kalender anschauen oder die Einstellungen von ihren Studienrichtungen verwalten oder man kann in der Seite „MyModuls“ hingehen und seinen Semesterplan anschauen.

Aber wenn man erstes mal das System benutzt, dann soll man zuerst die Kalender-Seite hingehen und für sich selbst einen Studienrichtung auswählen. Nachdem Studienrichtung auswählen, soll man in der Kalender-Seite auf der Button "Verwalten" drücken und seinen Semesterplan erstellen, dafür soll man einfach in der Seite auf der Button "Select Semester" drücken und die gewünschte Semester eintippen oder man kann auch in dieser Seite auf der Button "Lehrfach" drücken und für sich selbst die gewünschte Gruppen von einer Lehrfach auswählen, damit das Programm den Ort und die Zeit von diesen Lehrfächern in der Kalender anzeigen wird.

---

## 5 Progress Report

---

### 5.1 Summary

*What's the overall progress? Which activities have been undertaken? What has gone well – where have problems occurred?*

Ich habe bis jetzt alle Meilensteine fertig implementiert. Es fehlt jetzt nur Import/Export von einen Semesterplan und das werde ich auch in einer Woche fertig implementieren.

Das Programm hat jetzt insgesamt 18 Modul-Klassen, 3 Klasse für die Security, 11 Interfaces für die Repositories, 1 Klasse für die Logik, 6 Klasse für die Kontroller und 6 HTML-Seiten gebildet.

Der User kann sich in der System registrieren oder einloggen. Für die Sicherheit benutzt das System Spring-Security, damit der nicht registrierte Users den System nicht zugreifen dürfen. Ein registrierte User kann sich jetzt in der System einloggen und sich für mehreren Studienrichtungen und das Ausprägungsfach von diesen Studienrichtungen anmelden. Ein registrierte User kann auch die Module anschauen und davon sich einige Lehrfächern anmelden, welche er/sie dieses

Semester besuchen will. Er kann auch von diesen Modulen für sich Selbst einen Semesterplan erstellen und eingeben, ob er einen Modul geschafft hat oder nicht. Wenn er die Lehrfächer, die er geschafft hat, eingeben, wird das Programm ihm auszeigen, wie viele Moduls er in seinem Semesterplan geschafft und nicht geschafft hat. Für die Kalender habe ich die Library „<http://fullcalendar.io>“ benutzt. Der User kann sich im Kalender die Termine gruppiert nach täglich, wöchentlich oder monatlich anschauen. Um die Daten zu lesen habe ich die Library „JAXB“ benutzt. Mit Hilfe dieser Library kann man XML Daten automatisiert in Java-Objekte / POJOs deserialisieren. Für die allgemeine Aufgaben kann ich schon sagen, dass ich bis jetzt über 90% fertig implementiert habe und es hat alles ganz gut funktioniert. Ich habe nur Problem mit der Usability, weil ich mich darüber nicht so gut entscheiden kann, wie die Seiten ausschauen soll, damit sie wirklich einfacher sein würde. Ich habe am Anfang einige Schwierigkeiten gehabt, weil ich nicht so gut verstanden habe, wie die XMLs ausschauen und wie die Daten in einer Models hinzufügen werde, deswegen konnte ich die Daten nicht so einfach auslesen und in einer Datenbank speichern.

---

## 5.2 Status per Milestone

*What's the progress on a milestone basis? Have all activities been started and terminated on time? What's the percentage of completion for every given requirement?*

*Ich kann diese Projekt in 9 folgende Meilensteine unterteilen:*

1. XML auslesen
2. Modelle-Klassen erstellen
3. Datenbank speichern und wieder auslesen
4. Security
5. Die Funktionalität von User und Administrator
6. Html-Seiten und die Kontroller erstellen
7. Kalender auszeigen und die Daten in der Kalender hinzufügen
8. Die Funktionalität von User und Administrator
9. Import/Export

Ich habe bis jetzt alle Aufgaben fertig implementiert. Es fehlt bei mir nur "Import/Export" und diese Aufgabe werde ich auch in einer Woche fertig implementieren. Danach will ich mich noch um die Usability kümmern und die Projekt fertig stellen.

---

## 5.3 Risk Analysis

*Which risks have been identified? Which have occurred? What actions have been undertaken?*

Es gibt nur einen Risiko dafür, dass die Einstellungen von der XML des Universität geändert werden. Wenn so etwas passieren wird, dann kann das System die Daten nicht mehr einlesen, deswegen wird das System nicht mehr funktionieren. Dafür werde ich das System so einstellen, dass das System den gecachte Daten benutzen wird. So kann ich auch bisschen Zeit sparen, dass ich die Konfiguration von meinem Programm ändern werde.

## 5.4 Activities per Person

*Which roles / activities / responsibilities have the individual group members taken over within the last period. Which technologies have been explored? Which components have been implemented? Distribution of work between the team members.*

### **Implementiert:**

1. XMLs auslesen
2. Model-Klassen erstellen und fertig implementieren
3. Datenbank erstellen, die Daten speichern und wieder auslesen
4. Security
5. Die Funktionalität von User implementieren
6. Html-Seiten und CSS implementieren
7. Kalender auszeigen und die Daten in der Kalender hinzufügen
8. Semesterplan erstellen

*Ich habe Import/Export noch nicht fertig implementiert. Aber wie ich schon früher geschrieben habe, dass ich in einer Woche Import/Export auch fertig implementieren.*

### 5.4.1 Report on working hours per person

*Please enter your time-recordings for every team member.*

<b>Meilenstein</b>	<b>Zeit</b>
<b>XML-Reader</b>	10 Stunden
<b>Models implementieren</b>	5 Stunden
<b>Datenbank und Repositorys erstellen</b>	7 Stunden
<b>Html-Seiten und Controllers</b>	25 Stunden

---

## 6 Feedback

*Any additional feedback or notes you want to give at this time*

Ich finde den Ziel für dieses Lehrfach wirklich ganz gut, dass wir selbst in einer echten Projekte arbeiten und davon was lernen können. Aber es wäre vielleicht besser, wenn wir auch von jemanden, der viele Erfahrungen von zB. „Web-Application“ hat, bisschen noch was neues lernen und in dieser Projekte es benutzen können. Wir sind natürlich Wissenschaftler und wir können viele Sachen selbst lernen aber ich glaube dass es schneller funktionieren wird, wenn jemand uns mindestens einen Überblick von der neuen Technologien erklären wird.