



Departement of Engineering and Information Technology

Project2 (Modul BTI7302) 19/20

Report on management system for:

"Academy of the handsome men and beautiful woman(BEA)" web application

author: Kristina SHIRYAGINA (kristina.shiryagina@bfh.ch)

supervisor: Prof.Dr Olivier BIBERSTEIN (cccc.cccc@bfh.ch)

December 4, 2019

Contents

1	Abstract	3
2	Main section	3
2.1	Vision	5
2.1.1	Introduction	5
2.1.2	Problem Statement	6
2.1.3	Stakeholder Summary	6
2.1.4	Product Overview	6
2.1.5	Summary of System Features	7

2.1.6	Summary of Future Features	7
2.1.7	User Summary	8
2.2	Software development methology	8
2.2.1	Agile development, Scrum.	9
2.3	Feasibility	10
2.3.1	Technical Feasibility	10
2.3.2	Financial Feasibility	10
2.3.3	Resource and Time Feasibility	10
2.4	Analysis	11
2.4.1	Domain model	11
2.4.2	Product Backlog	13
2.5	Design	13
2.5.1	System Sequence Diagrams	13
2.6	Implementation	14
2.6.1	Why Spring Boot And Angular ?	15
2.7	Integration and System testing	17
3	Conclusion and future work	17
3.0.1	Summary	17
3.0.2	Interpretation	17
3.0.3	Outlook	17
4	Project planning	18
4.1	Meetings calendar	18
4.2	Spring Backlog	20
4.2.1	Protocol	21

1 Abstract

/**The abstract may not be what you write first, as it might be easiest to summarize your whole paper after it's been completed. You could draft it from your outline, but you'll want to double-check later that you have included the most important points from your article and that there's nothing in the abstract that you decided not to include in your report..., max 250 words*/ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

For that we read the article book of **Diniz:2010:UG0:525452.452352** /** just an example

2 Main section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer

sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac

habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

2.1 Vision

2.1.1 Introduction

/** The introduction gives the reader the necessary background info. can include: -a description of purpose and objective -a statement of the problem -background info -a review of previous work -an indication of the scope and limitations of study -an outline of material presented in the rest of report */

As the head of information system for Online academy we are tasked with developing a part of new Online Management System. As the idea of online education is getting more popular day by day.

The proposed software product (Online Beauty academy) is an online education system. The system will be used for online-education, to download lectures, conducting online quizzes, course registration, exam reservation, managing results. The system must be right protected.

This product let persons who have interest to study beauty , who want to make themselves more beautiful to do it. Our product includes different online courses. There are beauty-course , beauty-instructor course and other future courses. Each course will have topics and lessons for each topic. The context of lessons are text, video-tutorials, etc. If participant want to get a certificate he shall do exams.

Online Exams.

This application will establish a network between the lecturers and participants. Academia enter the questions they want in the exam. These questions are displayed as a test to the eligible participant. The answers enter by the participant are then evaluated and their score is calculated and saved. This score then can be accessed by the Academy to evaluate the performance of participants. Exam details.

Each topic will have small exam(quizzes). Making exams the participant will collect a points. The sum of all point for all small exams are 30% of final grade. After making all small exams participant will be able to make a final exam that has weight 70% of final

grade. The participant must make a reservation for final interactive exam with lecturers. He has to prenote the available Date for this exam. ´

The application has an administrator who keeps an eye on the overall functioning of the system.

2.1.2 Problem Statement

Now academies are running various programs as full time courses. The academy timing sometimes make it difficult to study for person who are doing some jobs. The online education would help such person who live far away from education institutes. The other problem is that the current online management system doesn't have the needed flexibility and is not modern enough. The capabilities are limited. Online management system is effective, reduce time and cost in courses and exam management process.

2.1.3 Stakeholder Summary

Similar to other technology applications, the success of online-learning is dependent on the extent to which it satisfies the needs and addresses the concerns of its stakeholders.

- Participant: use the system to register for the course or exam, view information.
- Instructors: they could give ideas on the solution for the system's development and improvement.
- Administrator: manage the system after it is built.
- Education Institutions
- Content Providers
- Development team: include all software engineers, business analysts, system analysts, system designers, implementers, testers, QA, and project management. They are tasked to build the system.
- Employers

2.1.4 Product Overview

/**Think of questions a customer might ask Always include the details: dimensions, size, materials, etc. Tell the story of your product to make it feel unique Make text easy to scan and read quickly Add testimonials and social proof Optimize product descriptions for SEO too.who, what, where, when, why and how before writing. */ This section provides a high-level overview of the BEA(Beautiful academy) features for various types of roles.

.Our user can easily register hisself for courses he likes and follow the program of this course.Each course has topics and lessons for each topic. The context of lessons are text,

video-tutorials, etc.

The user can test himself with mini exams. The user can also prenote a final exam with lecturer. Our academy gives possibility to have a certificate. For this the user have to make all required exams for course and if he passed es well he will become a certificate from our academy.

This application is also very useful for lecturers. The lecturer can use this system for uploade a content of courses, topics, lessons. The lecturer can make online tutorials for participants. He can gives an online exams. He can make evaluation. He can prepare content of quizzes needed for mini exams.

The administrator can create, change, delete, update and control information about courses, lectures, users, date and time, etc.

2.1.5 Summary of System Features

- Online registration.
- Log in.
- Manage user information.
- Manage Offering Courses.
- Communication via mails.
- Manage Lecturer information.
- Course registration.
- Exam registration.

2.1.6 Summary of Future Features

- Access the system as lecturer
- Manage Financial Activities.
- Uploading course content.
- Course evaluation.
- Downloading course content.
- Video conferencing.
- Info service.
- Information library.

2.1.7 User Summary

2.2 Software development methodology

The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines is called software engineering.

Software engineering is the discipline whose aim is:

1. Production of quality software
2. software that is delivered on time
3. cost within the budget
4. satisfies all requirements.

Software process is the way in which we produce the software. Apart from hiring smart, knowledgeable engineers and buying the latest development tools, effective software development process is also needed, so that engineers can systematically use the best technical and managerial practices to successfully complete their projects.

A **Software life cycle** is the series of identifiable stages that a software product undergoes during its lifetime .A software lifecycle model is a descriptive and diagrammatic representation of the software life cycle .A life cycle model represents all the activities required to make a software product transit through its lifecycle phases .It also captures the order in which these activities are to be taken .

Life Cycle Models

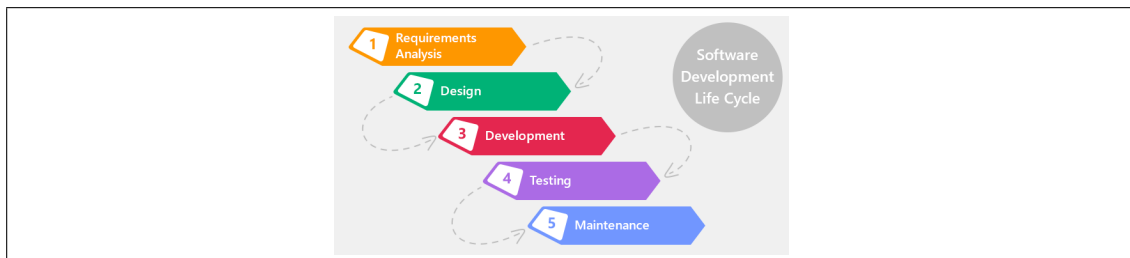


Figure 1: This is a Life Cycle Model

There are various life cycle models to improve the software processes. And we have used the WATERFALL MODEL.

This model contains **6 phases**:

- o **Feasibility study** The feasibility study activity involves the analysis of the problem and collection of the relevant information relating to the product. The main aim of the feasibility study is to determine whether it would be financially and technically feasible to develop the product.
- o **Requirement analysis** and specification The goal of this phase is to understand the exact requirements of the customer and to document them properly.

o **Design** The goal of this phase is to transform the requirement specification into a structure that is suitable for implementation in some programming language.

o **Implementation** and unit testing

During this phase the design is implemented. Initially small modules are tested in isolation from rest of the software product.

o **Integration and system testing**

In this all the modules are integrated and then tested altogether.

o **Operation and maintenance.** Release of software inaugurates the operation and life cycle phase of the operation.

2.2.1 Agile development, Scrum.

Scrum was in the center of developing process. As scrum projects make progress in a series of “sprints”, we have divided the whole process into 4 sprints. Product was first analysed, designed , then code and tested during the sprint.

Scrum Roles

- Scrum Master – Prof.Dr. Olivier Biberstein
- Developer – Kristina Shiryagina

Artifacts

- Product Backlog
- Sprint Backlog

Meetings have included :

- Product/release planning
- Sprint planning
- Weekly Scrum
- Sprint review
- Sprint retrospective

Scrum Meetings

Each meeting between developer and scrum master was made of several steps, that were repeated each time:

- Attendance: all
- Product Owner presents Product Backlog with all relevant user stories with their priority
- Discussions and clarifications if needed
- Results:

Prioritized Product Backlog

- Specifies what to build
- Final decision by the Master
- Vision, high level architecture, most important non-functional requirements

Release planning (if product is to be delivered in releases):

- Select and prioritize items of Product Backlog for the next Release Backlog

2.3 Feasibility

2.3.1 Technical Feasibility

This project is a web-based application. The main technologies and tools:

- HTML
- CSS
- TYPESCRIPT
- FREE MAKER(ftl)
- JAVA
- SPRING-BOOT FRAMEWORK
- ANGULAR FRAMEWORK
- POSTGRESQL
- INTELIJ IDEA
- GIT
- GIFFY(DIAGRAM DRAWING TOOL)
- LATEX
-

2.3.2 Financial Feasibility

Being a web application BEA will have a hosting cost.

2.3.3 Resource and Time Feasibility

- Laptop (programming device)
- Hosting
- Programming tools

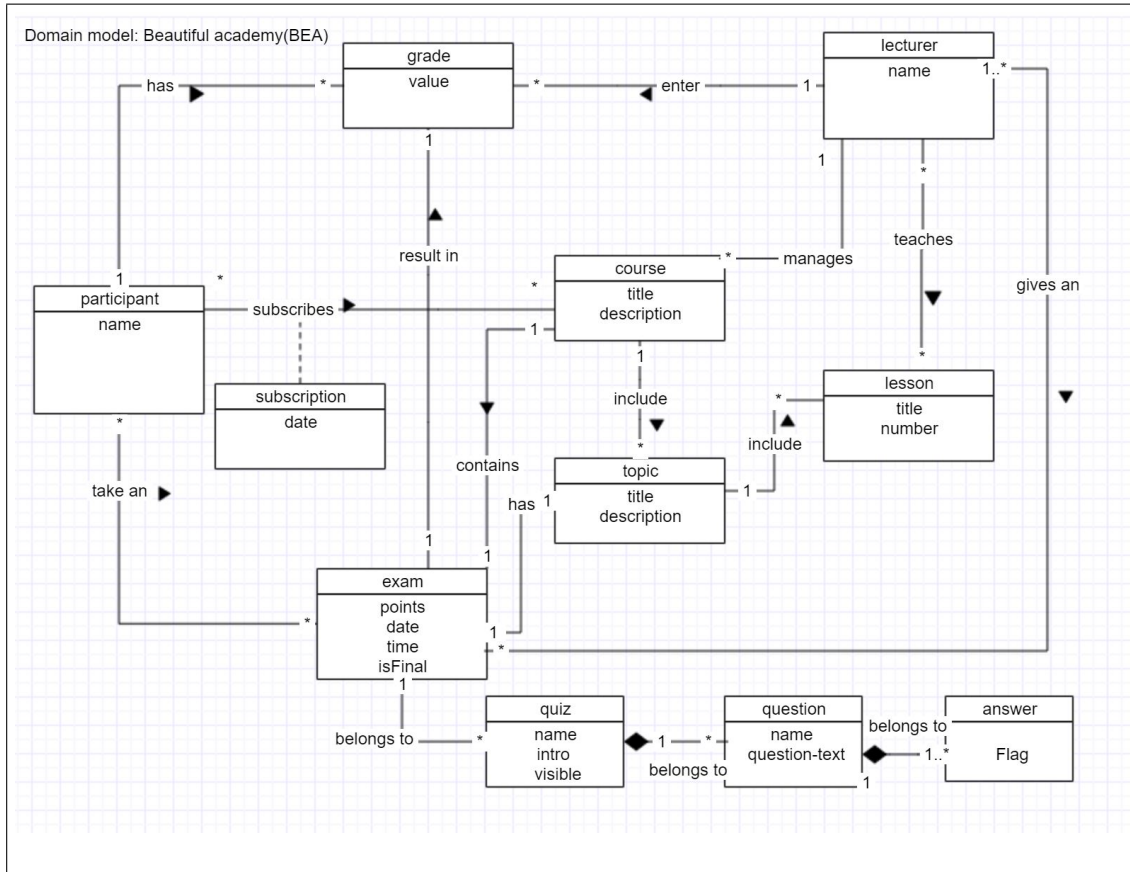


Figure 2: This is a Domain model for Beautiful academy(BEA)

2.4 Analysis

2.4.1 Domain model

This document describes the domain model of the Beautiful Academy BEA. It introduces the most important concepts and the associations among them. It also introduces the respective multiplicities.

Concept Classes

Concept class Participant models a person who is taking the courses.

Concept class Lecturer models a person who is teaching and give an exam for the participant.

Concept class Course models the main courses that offer application.

Concept class Grade models the grades of a participant.

Concept class Topic models the sub course of the course.

Concept class Lesson models a set of lessons that contains each topic.

Concept class Exam models a exam that can be taken by participant.

Concept class Subscription models a subscription, that has a date when participant is subscribed.

Concept class Quiz models a quiz that belongs to exam.

Concept class Question models a question that belong to quiz.

Concept class Answer models a possible answer to questions.

Associations

Association take an between Participant and Exam denotes the fact that Participant can make many exams, hence (*) multiplicity, and (*) multiplicity at the Exam side means that the Exam can be done by many Participants.

Association has between Participant and Evaluation denotes the fact that Participant can get zero or more Evaluations, hence multiplicity (0..*), and (1)multiplicity at the Participant side means that each Participant has its own set of Evaluation.

Association subscribes between Participant and Course denotes the fact that that zero or more Participant can be entered to the Course depending of the number of participant, hence multiplicity(*), and each Participant can be subscribed to zero more courses, hence multiplicity(*)

Association result in between Exam and Evaluation denotes the fact that each exam has it's own unique evaluation, hence multiplicity 1, and 1 multiplicity at the evaluation side means that each evaluation belongs to unique exam.

Association has between Exam and Topic denotes the fact that each topic has it's unique an intermediate exam.

Association contains between Exam and Course denotes the fact that each exam belongs to it unique course , and each course has it own unique final exam.

Association include between Course and Topic denotes the fact that course has many topics, hence multiplicity (*), and each topic belongs to exactly one course, hence multiplicity (*).

Association include between Topic and Lesson denotes the fact that topic has many lessons, hence multiplicity (*), and each lesson belongs to exactly one topic, hence multiplicity (*).

Association teaches between Lecturer and Lesson denote the fact that Lecturer can teaches zero or more lessons, hence multiplicity (*), and each lesson can have zero or more lecturer, hence multiplicity (*).

Association gives an between Lecturer and Exam denotes the fact that Lecturer can give zero or more exam, hence multiplicity (*), and each exam can be done by 1 or more lecturer, hence multiplicity (1..*).

Association enter between the Lecturer and Evaluation denotes the fact that each lecturer can enter zero or more evaluations, hence multiplicity (*), and evaluation can be entered by exactly one lecturer, hence multiplicity (1).

Association manages between the Lecturer and Course denotes the fact that each lecturer can manage zero or more courses, hence multiplicity (*), and each course can be manage by exactly one lecturer, hence multiplicity (1).

Association belongs to between Quiz and Exam denotes the fact that each Quiz be-

longs to exactly one exam, hence multiplicity (1), and each exam can have zero or more quizzes, hence multiplicity(0..*).

Association belongs to between Quiz and Question denotes the fact that each Question belongs to exactly one Quiz, hence multiplicity (1), and each Quiz can have zero or more questions, hence multiplicity(0..*).

Association belongs to between Question and Answers denotes the fact that each Answer belongs to exactly one Question, hence multiplicity (1), and each exam can have one or more answer, hence multiplicity(1..*).

2.4.2 Product Backlog

The product backlog is a list of user stories which is used to implement the product vision. It is sorted according to the priority of the user stories according to the product owner. The priority of these stories will be modified during the process of the project.

id	Story name	Story description
1.0	Log in	As a user, I want to be able to login into the system
2.0	List courses	As a Participant I want to list courses I'm entitled to
3.0	View information	As a Participant, I want to see the information of the courses
4.0	List results	As a Participant, I want to see my marks of the courses
5.0	View schedule	As a Participant or a Lecturer I want to be able to view the schedule
6.0	Exam reservation	As a Participant, I want to be able to select an exam
7.0	Cancellation of exam	As a Participant, I want to be able to cancel the exam
8.0	Personal information	As a Participant or a lecturer I want to see my personal information
9.0	Manage course	As a System Administrator, I want to be able to manage the courses
10.0	List participants and corresponding courses	As a lecturer, I want to see a list of participants and their corresponding courses

Table 1: Product Backlog

2.5 Design

2.5.1 System Sequence Diagrams

Use case "Log in"

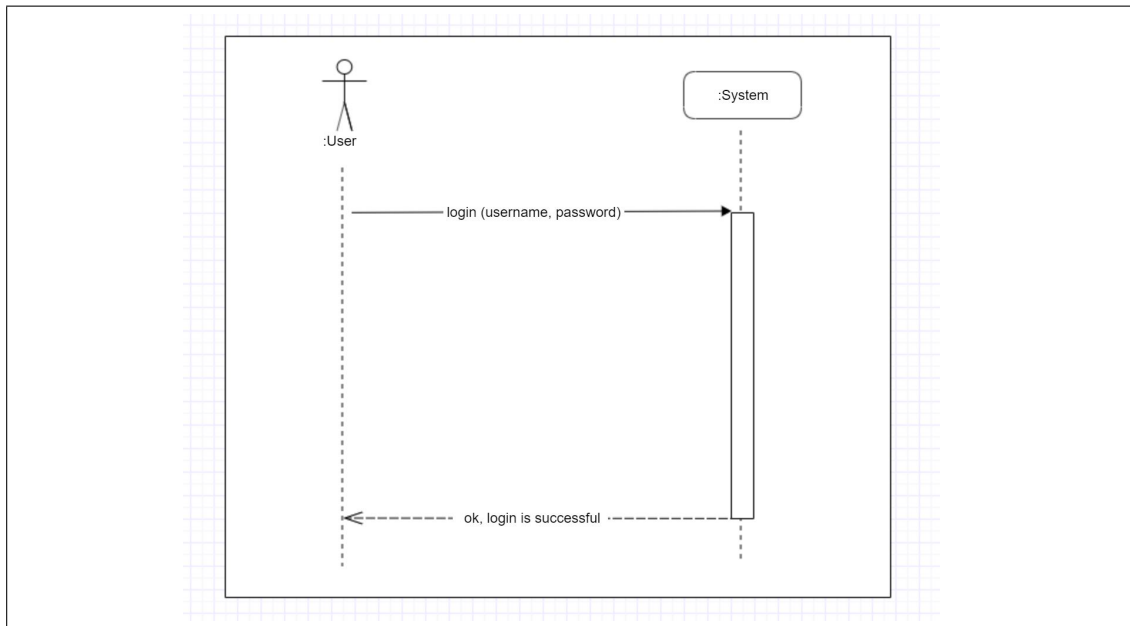


Figure 3: ssd for login

2.6 Implementation

Backend - Srpring-boot

- Integration/ Data layer (entity models and repository)
- Application/ Service layer (contains logic)
- Web layer (RESTful controllers)

What is restful? Building REST services with Spring REST has quickly become the de-facto standard for building web services on the web because they're easy to build and easy to consume.

Why REST? REST embraces the precepts of the web, including its architecture, benefits, and everything else. What **benefits?** The web and its core protocol, HTTP, provide a stack of features:

Suitable actions (GET, POST, PUT, DELETE, ...)

Caching

Redirection and forwarding

Security (encryption and authentication)

So building on top of HTTP, REST APIs provide the means to build flexible APIs that can:

Support backward compatibility

Evolvable APIs

Scaleable services

Securable services

A spectrum of stateless to stateful services

Frontend – Angular

- Angular MVC Framework by Google
- Component based
- Main programming Language is Typscript
- Own structure
- CSS (Design the Gui)
- Angular Material Design

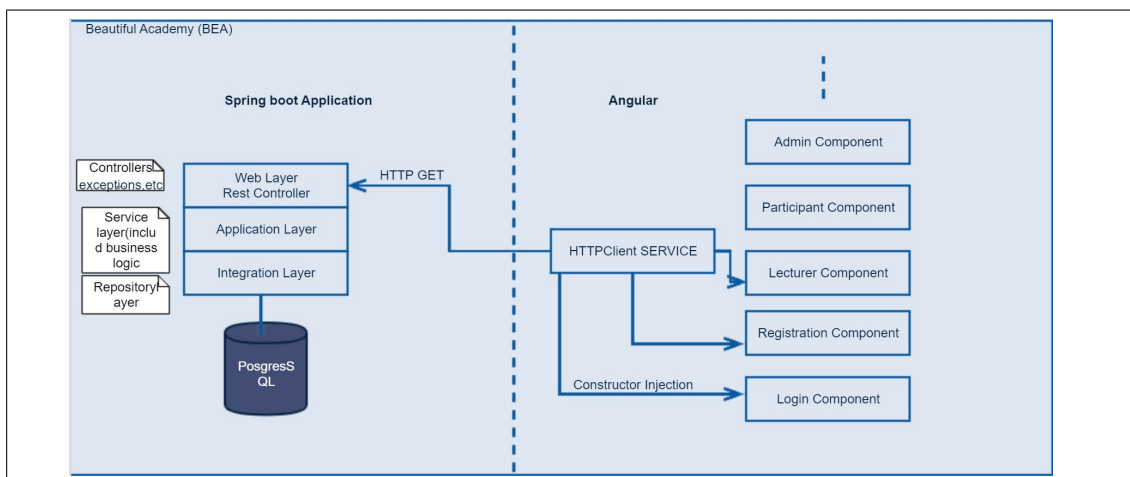


Figure 4: Layer model of application implementation

2.6.1 Why Spring Boot And Angular ?

BUILD Backend WITH SPRING BOOT

Before starting with spring boot, it was clear for me about spring boot that spring boot does not provides any extra features on functionality on top of spring framework. Rather, it provides unlimited defaults configurations and useful conventions to create a stand-alone, production grade web applications in no time.

Spring Boot is the starting point for building all Spring-based applications. Spring Boot is designed to get you up and running as quickly as possible, with minimal upfront configuration of Spring.

- Get started in seconds using Spring Initializer
- Build anything: REST API, WebSocket, web, streaming, tasks, and more
- Simplified security
- Rich support for SQL and NoSQL
- Embedded runtime support: Tomcat, Jetty, and Undertow
- Developer productivity tools such as Live Reload and Auto Restart
- Curated dependencies that just work
- Production-ready features such as tracing, metrics, and health status
- Works in your favourite IDE: Spring Tool Suite, IntelliJ IDEA, and NetBeans

Using these features it has really made building a production grade Spring applications very easy and faster for developers. Also, no XML configurations required anymore with spring boot.

ANGULAR FOR FRONT-END

The technology for front-end is Angular.

Angular helps build interactive and dynamic single page applications (SPAs) with its compelling features including templating, two-way binding, modularization, RESTful API handling,

dependency injection, and AJAX handling. We can use HTML as template language and even extend HTML' syntax to easily convey the components of the application.

Angular applications are built using TypeScript language, a superscript for JavaScript, which ensures higher security as it supports types (primitives, interfaces, etc.). It helps catch and eliminate errors early when writing the code or performing maintenance tasks.

Angular has a lot of pros

- simplicity
- efficiency
- Developers find AngularJS very effective especially in creating dynamic, single page apps, and supporting MVC (Model View Controller) programming structure.
- time-saving Projects that previously used to take many months with other frameworks can now be completed faster with AngularJS. All that AngularJS framework requires is splitting the app into several MVC components. From there, the framework takes over because you do not require additional coding.
- the app is easy to learn and get started.
- data binding in AngularJS is very easy.

I like in Angular that it gives our application a clean structure, that is easy to understand and easy to maintain.

It brings a lot of utility code that we can reuse, for example users navigation. Angular applications are more testable.

2.7 Integration and System testing

For testing our application we have use different tests: • JUNIT is used as the main testing tool that automates the testing process. • Checklist to check if we have all function we need in UI. • Postman to test crud operation in controller

3 Conclusion and future work

/** In the conclusion of the project, you first summarize the most important results of your project in an understandable way. The main thing here is to filter out the essential content from the previous chapters and get to the point in a compact, clear and concise way. */ Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

3.0.1 Summary

/* Summary of the main results of your work/thesis*/

3.0.2 Interpretation

/Interpretation of my results

/**. In the theoretical part of your project, you should research and present the current state of research in detail in order to draw the right conclusions from the conclusion of your project. Even if your question could not be answered satisfactorily, you should present it honestly. Check what you could and could not achieve, what you would do differently in retrospect and which methods have proven to be useful.*/

3.0.3 Outlook

/** outlook of my project and future research recommendation*/

References

- [1] Walter Abish *The Alphabetical Africa*,1974
- [2] Prof.Dr. Olivier Biberstein, Prof.Dr Eric Dubuis *software Engineering and Design*,
- [3] Spring.io/guides •,2019
- [4] •,

4 Project planning

4.1 Meetings calendar



Figure 5: Meetings calendar

4.2 Spring Backlog

Sprint Backlog of Sprint 1

	id	Story name	Story description
	0	Domain model	Domain model
	1	Documentation	Initial structure of main document
	2	User stories	Write some userstories
	3	Vision	Start vision of project
	4	Infrastructure	Set up infrastructure- Gitlab
	5	User stories	Define more user stories and enter them to the main document
	6	User stories	Extend user stories with "success" and "failure"
	7	Infrastrcuture	Sprint Backlog. Make a document with sprint plan
	8	Dcumentation	Extend documentation with explanation about Angular
	9	Documentation	Make clear vision document and put it to the main document

Sprint Backlog of Sprint 2

10	Postgres SQL	Add postgres SQL to the project, make connection
11	UML diagrams	System sequence diagram for login
12	User stories	Implement login
13		

Sprint Backlog of Sprint3

15	UML diagrams	ssd for registration
16	User stories	Registration
17	Front end	Start with frontend
18	Documentation	word->latex
19	User stories	Add email confirmation to the registration
20	User stories	Domain classes
21	User stories	Repositories and Service classes
22	User stories	Add ssd for new feature
23	User stories	Add Controller for new feature
24	Front end	Continue with Angular front end

Spring Backlog of Sprint 4

[illegible]

Table 2: Sprint Backlog

4.2.1 Protocol

Frequency: (weekly)

Meeting length: (35-45 minutes)

Agenda

- Demo and Discuss Deliverables(Demo)
- Planning next Goals(Plan)
- Lessons learned (Lessons)
- Date, time of the next meeting(next meeting)

Report from 09.10.19

Plan

Next goals are:

- Introduction of a Vision make clear. Write about an application I want to build. I have to write a Vision that can make a good picture about the functionality of this application.
- Change the problem statement:
- Join 2 Systems in 1 . Rename system in functions. And write that these functions just a part of this system we want to build.
- Write concrete user stories to these functions, group the stories according the function.
- Analyze domain model, put attributes to each conceptual class, make description for each association. Rebuild domain model according of new clear representation of necessary functions of the system.
- Sequence diagram of first function we want to implement (probably log in)
- Try to make class-diagram.
- Implement of log in function.
- Write a protocol in the main doc.

Lessons learned

To documentation plays decisive role in Software Engineering Project. The first analysing phase have to be done well to make a good start for design and implementation of the IT Product. Next Meeting: 16.10.19, 14:30

Report from 16.10.19

Plan:

The next goals are:

- A domain model – last version. +
- Vision complete. +
- Put spring plan in the main document (Time planning is the first preference).
- After completing 3d point, merge sprint1 and make a del1.
- Start with Design part: 1st we start with design for login feature.+
- SSD for login +
- Write good document for Spring Boot and Angular. Describe the most successful aspects,that has spring boot , angular. Describe why we have chosen it for this project. Describe how can be implement login with spring boot. +
- Make a Product Backlog(list of user stories) and divide to 4 sprints. +
- Point 1 and 2 have highest priority. Just when these 2 points successfully completed I will continue my to-dos. +
- Change the style of document, make headers numerable, and the style more readable. +

Lesson Learned

With this practical work I become always clearer the main principles of building software.

Next meeting: 23.10.19, 14:30

Report from 23.10.19

Plan:

- Make Vision clear(Describe how people will use this application. Describe what is it exactly the course(content, PDF, Unterricht) The vision needs to be loner ,about 1 A4 page. Vision-Solution belongs to Vision, here I have to write what we will produce.
- Domain model. Change word entity to concept by description. Add two concepts: subscription with a Date attribute . Add to exam concept boolean , that will show the difference between final and intermediate exam.
- Change product backlog (failure and success)

- Make ssd for login with all needed classes(show exactly what's happened in system).
- Implement login function, test it. Have to be able to make a demo.
- After making all 5 points and if I'll have more time make ssd for other features.

Lesson learned:

More experience about software engineering diagrams

Next meeting: 30.10.19, 14:30

Report from 30.10.19

Plan:

- Start frontend with Angular. Make frontend for login and registration.

Lesson learned:

Next meeting: 06.11.19, 14:30

Report from 06.11.19

Plan:

- Documentation
- Most important thing in a documentation is to write it so that the reader can easily understand the main scope of project, how it will be realised, etc.
- Angular and spring . Make description of angular and spring , reader have to understand for what we use exactly this frameworks. (have to notice in bibliography everything I use from other authors)
- Make a build of architecture of angular and spring(probably layers build)
- Vision more
- User-stories
- Agile and scrum -
- Make registration with via email address. (read about registration via facebook, ect if I have more time)
- Start to learn about microservices. Try to add first documentation about it.

Lesson learned:

Today I've learned that I have to plan my work better, means that first I have to realise the most difficult and most important tasks, and then make tasks less difficult.

Next meeting: 13.11.19, 14:30

Report from 13.11.19

Plan:

- Documentation: Describe:
What is it monolith architecture, what is different from monolith and microservices?
What is restful? (in layer diagram)
- Code
Make registration via email. Start other features. This step I have to do just when I have finished the documentation. For next time be able to make a demo of what we have.

Lesson learned:

Be able to define clear the features I want to realise in this project.

Next meeting: 27.11.19, 14:30

Report from Plan:

-
-

Lesson learned:

Next meeting:

Report from Plan:

-
-

Lesson learned:

Next meeting:

Report from Plan:

-
-

Lesson learned:

Next meeting:

Report from Plan:

-
-

Lesson learned:

Next meeting:

Report from Plan:

-
-

Lesson learned:

Next meeting: