Bern University of Applied Sciences | BFH

Department of Engineering and Information Technology

Project 2 (Module BTI7302) 20

Report on

"Planning of the Assignments Information System(PLANA)" Web Application

Author: Kristina SHIRYAGINA (kristina.shiryagina@bfh.ch)
Supervisor: Prof. Marcel PFAHRER (marcel.pfahrer@bfh.ch)

June 12, 2020

Contents

Acknowledgements

I would like to thank my project supervisor - Prof. Marcel Pfahrer for his advice and guidance. I would also like to thank Andrei Herasimau for his corrections and Egger Samuel Sebastian for his advice.

Abstract

The focus of this project is to create the initial phase of a web application PLANA (Planning of assignments for lecturers) using ASP.NET Core Blazor.

To achieve this, the following has been done:

- Requirements Engineering
- Domain Analysis
- System Architecture and System Design
- Research of ASP.NET Core Blazor
- Application Prototype

We measured the framework for popularity.

We came to the conclusion that ASP.NET Core Blazor is completely suitable for us.We have also created prototype for further work.

1 Introduction

Nowadays we have a large selection of technologies for creating a web application. In the world of frameworks, advancements are made at a rapid rate. The most successful of frameworks are fast, facilitate the work of developers, add variety to a user interface, and improve technology as a whole. Our task as developers is to research these technologies, to try them out and use them.

Our goal is to create a web application for assignment planning for lecturers in our school. This work consists of two parts, - the project that is described in this document and the thesis which is planned as a continuation of this work in the next semester. The project has several objectives:

- Employ software engineering techniques which include :
  - Requirements engineering
  - Domain Analysis
  - System Architecture and System Design

- Research the ASP.NET Core Blazor framework, try it out and see if it suits us
- Create an application prototype

  Why did we choose the ASP.NET Core Blazor framework for research and subsequent use?
  ASP.NET Core Blazor is a framework for building interactive client-side web UI with .NET. It has the following interesting features: [3]
  - alows creation of rich interactive UIs using C# instead of JavaScript
  - alows sharing of server-side and client-side application logic, which is written in .NET
  - alows rendering of UI as HTML and CSS for wide browser support, including mobile browsers.

  The result of this work should be an accurate idea of which technologies will be used to create the web application, as well as which software engineering techniques should be utilized for the further implementation of the product.

2 Customer Problem Statement

2.1 Problem Statement

Our plan is to create a system which will perform all the necessary functions for planning tasks for teachers in our school - Berner Fachhochschule (BFH) .

The main users of this system are the study director and teachers. The study director will organize the planning of assignments for teachers and teachers will participate in this planning. Teachers will also manage their project information.

It is a process, which involves the input of certain data for a specific user, certain views for each user, and time limits set by the system.

Planning of assignments includes planning, managing and communication. These are all joint actions.

At our school, actual teacher assignment planning is done using Microsoft Excel tool.

If we look at the requirements described above, we can clearly see that Excel is not the best option. We plan to substitute Excel with our own web application. Compares various Excel and web application we can clearly see that the web application is better suited to our requirements

Features	Excel	Web Application
Enhanced accessibility(teamwork)	Sharing of excel spreadsheets can get out of control, there are many versions of the file. It is impossible to know who has made which changes. Excel is not designed for collaborative work.	It is very important that with the web application, anyone who is authorized can access it. The administrator can specify who can use the application, who can input data. Higher granularity is possible, for example only exposing certain parts of a spreadsheet do different user audiences.
Support various devices	Excel on IOS and Android isn't so powerful as the PC version. For a complex project, it is necessary to have a PC version.	Web applications easily support all the devices connected with the internet. It's also possible to use a smartphone to run web applications.
Data storage	Access to a spreadsheet is sometimes limited to one person at a time. Spreadsheets have made a huge step forward due to the presence of Google Sheets.	Databases have a more robust set of features for handling data. SQL Server and Excel are in completely different weight classes in terms of data handling. SQL is a more powerful tool. Databases are also better atprocessing enormous numbers of records. Databases work with different programming languages , as opposed to Excel. Databases can be used by multiple users at the same time. Data is much safer in a database.
Development (price, time)	Sometimes easy and cheaper to set up.	Can have high development cost. It takes some time to build the web application.

Table 1: Comparing Excel to Web Application

A web application will allow us to implement features, like simultaneous system access by several users. Users will receive only the specific views that they need, which will allow them to enter and confirm data without violating the integrity of the main data.

2.2 Glossary

- **FURPS+**[1] is a system for classifying requirements.
  - Functionality
  - Usability
  - Reliability
  - Performance

  - Supportability

- **SignalIR** is a free and open-source software library for Microsoft ASP.NET that allows server code to send asynchronous notifications to client-side web applications.
- **Blazor** is a free and open-source web framework that enables developers to create web apps using C# and HTML. It is being developed by Microsoft.
- **HTML** HyperText Markup language
- **CSS** Cascading Style Sheets
- **SQL** Structured Query Language
- **JS** JavaScript
- **CRUD** Create, read, update and delete
- **EF** Entity framework
- **UI** User Interface
- **API** Application Programming Interface
- **MS** Microsoft
- **BPMN** Business Process Model and Notation

3 System Requirements

3.1 Product Backlog and user stories

USER STORIES

Id	Story Name	Story Description
Study director		
1.0	Make a plan of assignments	As a study director, I want to be able to make a concrete plan of assignments for lecturers for the coming year as well as a provision plan for the next 2-3 years, so the teachers can be better informed about their work hours and schedule for the next several years. And it will be easier to add new small changes into the employment plan.
2.0	Manage plan of assignments	As a study director, I want to be able to manage information in the assignment's plan, so that I am able to change the information.
Future user stories		
3.0	Get projects information	As a study director, I want to see which lecturers supervise which students in students projects like project1, project2, bachelor thesis, master thesis, so I can use this information for managing processes.
Lecturer		
4.0	See the plan of my assignments	As a lecturer, I want to see my assignment's plan for this year and next two years, so that I know the timetable of classes and I can manage my time better.
5.0	Participate in the planning and management of assignments	As a lecturer, I want to participate in the planning of assignments, so I can add my proposal for the plan and confirm it.
6.0	Manage projects	As a lecturer, I want to be able to manage my projects, so I can easily change any kind of information on my project. (e.g. add student/s to my projects, I know which student(s) is/are working on which project(s)).

Table 2: Product Backlog

The system has been provided with a list of requirements. All requirements are based on consumer needs. There is an identifier in the form of REQ-x and a priority weight from 1 to 5. A higher priority weight indicates that the corresponding requirement is more important for the project's success and for meeting customer needs.

3.2 Enumerated Functional Requirements

	Description	**Priority Weight**
REQ1	The system shall allow the study director to make a plan of assignments, e.g. a concrete plan for the coming year as well as a provisional plan for the next 2-3 years	5
REQ2	The system shall allow the lecturer to participate in the creation of the plan, i.e. manage his own assignments..	5
REQ3	The system shall allow the study director to manage assignment plan.	5
REQ4	The system shall allow the lecturer to manage his projects like project1, project2, bachelor thesis, master thesis	4
REQ5	The system shall allow the lecturer to see his assignment plan for the coming year and provisional for next 2-3 years	5
REQ6	While the user types a question, the system will be able to compare the search string to existing questions and prompt the user to look at those	1
REQ7	The system should show the study director the maximum number of working hours for a given lecturer	1
REQ8	The System should allow the study director to copy assignment plan from one year to other years, from one semester to another	2
REQ9	The system should offer possible replacements of the lecturer in case of sabbaticals and in other special cases	1

Table 3: Functional Requirements

3.3 Enumerated Non-Functional Requirements

Non-functional requirements are based on FURPS+, which stands for **functional usability, reliability, performance, supportability**, and other attributes. These requirements mainly pertain to quality attributes such as capability, compatibility, security, responsiveness, availability, efficiency, and maintainability.

FURPS+	Description	Priority Weight
REQ11	The system shall have an almost constant uptime.	5
REQ12	The search engine shall be reliable and will return relevant and accurate results.	2
REQ13	The system should be easy to navigate and utilize.	2
REQ14	The system must be supported by different browsers, on both web and mobile platforms	1

Table 4: Non-Functional Requirements

4 Functional Requirements Specifications

4.1 Stakeholder Summary

The stakeholders of our system are:

- **Study director**
  The first major stakeholder is a study director.The study director can be both the person who makes an assignment plan and also a lecturer.
- **Lecturer**
  The lecturer is another major stakeholder of our system. He participates in assignment planning, manages his projects and he uses the system to be informed of his assignment plan.
- **Institute Manager**
  The institute manager is a stakeholder of our system. He organises the planning of research projects.

4.2 Actors and Goals

The roles of people that will directly interact with the system and their goals.

- **actor** -Study director
  - Initiating Actor
  - **goal** Uses system to make a plan of assignments for lecturers.
- **actor** -Lecturer
  - Initiating Actor
  - **goal** Uses the system to participate in the planning of his assignments .

– **goal** Uses the system to get his assignment plan .
– **goal** Uses the system to manage his assignments.
– **goal** Uses the system to manage his projects.

4.3 Business Process Model and Notation(BPMN)

In this section, we show a graphical notation of the Planning of assignments process. The model in Figure 1 is provided to make this process easier to understand.

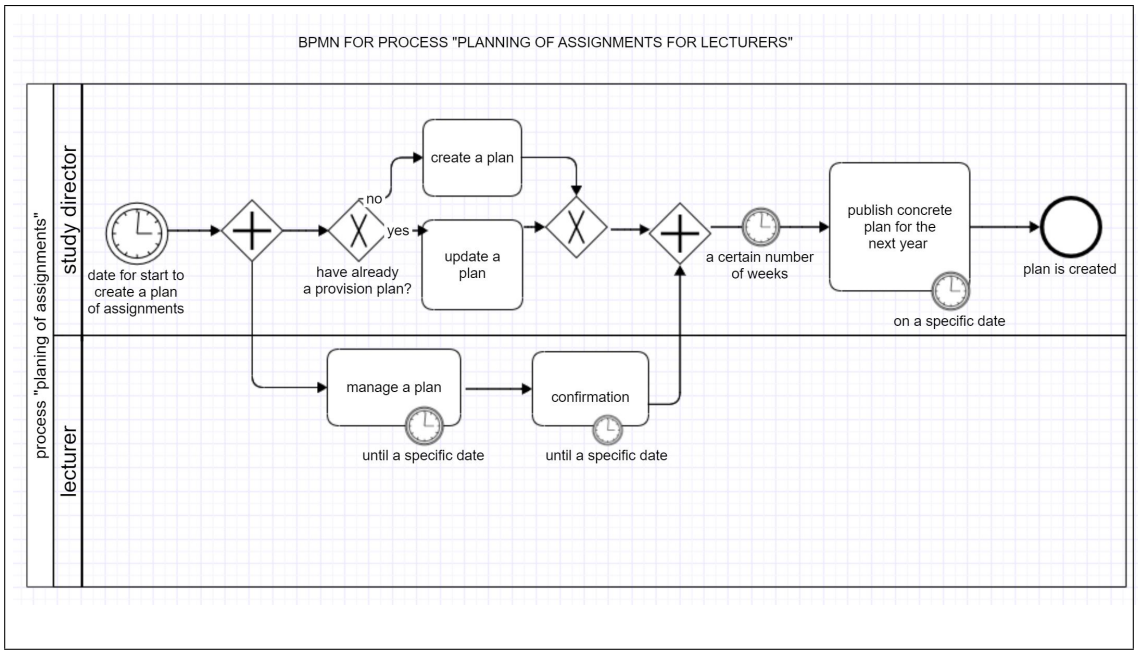

Figure 1: BPMN of "Creating a plan of assignments" 1st version

In Figure 1, we show the process of creating an assignment plan and also the interaction between the study director and lecturer.
The main process of creating an assignment plan shall start at a certain date. As we can see from this diagram, it is a collaborative job.The study director creates a plan with all the necessary details or updates a plan that already exists. At the same time, the lecturers shall manage their assignment plan.
The lecturers shall manage their plan until a specific date. Then, the study director makes some last changes, after which each teacher checks his plan and confirms it until a certain date.
After a certain period of time, the exact assignment plan is published.
After analyzing this model, we came to the conclusion that small changes in it will result in greater clarity in the process. Another minor change is the addition of messages to start certain processes.

Unlike the first version, in Figure 2 it is clear that the director starts the planning process, then joint work on planning takes place, which is completed by a certain time,

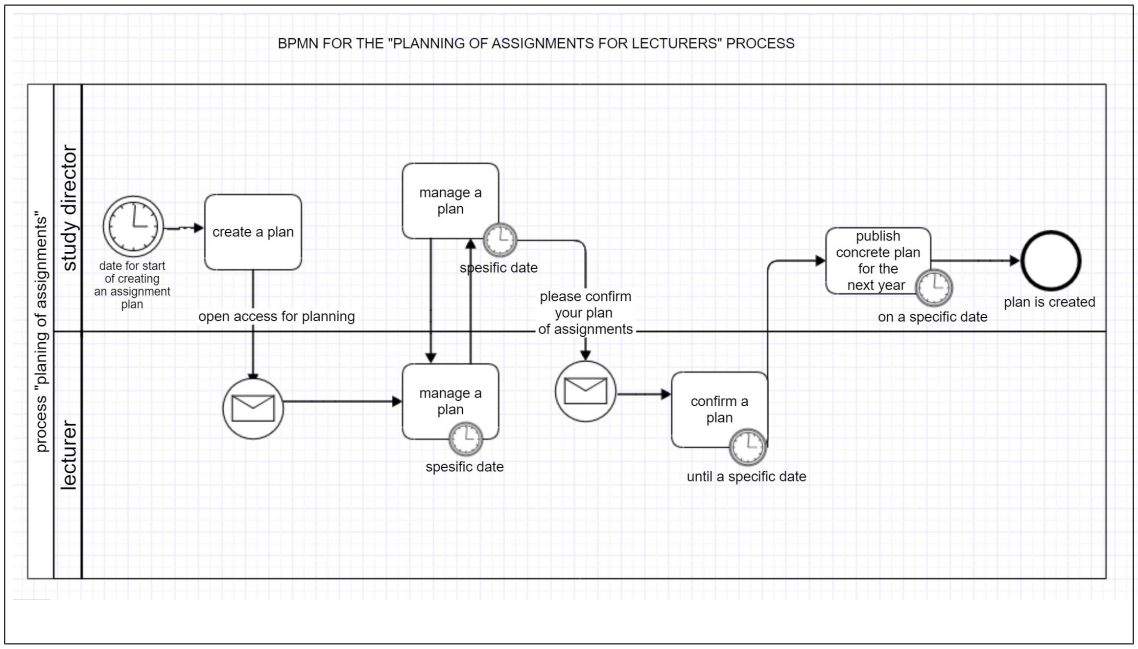

Figure 2: BPMN of ”Creating a plan of assignments” 2nd version

and then the plan is confirmed and published by a certain date.

4.4 Use Cases

i & ii. Casual and Fully-Dressed Descriptions

UC#1.1 Create a concrete plan

The Study director can create a concrete assignment plan of lecturers for the coming year.

UC#1.1	**Use case description of "Create a concrete plan"**
Name	Create a concrete plan .
Summary	The Study director makes a concrete assignment plan for lecturers for the coming year .
Actor	The Study director and The Lecturer
Precondition	The Study director and The Lecturer have logged into the PLANA information system.
Description	1. The Study director opens a assignment plan 2. If there is no provisional plan, create a new plan • for each lecturer: • adds module runs . • adds a percentage/part of employment for concrete module run. • add additional assignments : – Teaching obligations in other departments and schools (e.g. further education, MSE, ...) – Administrative and managerial activities – Research projects – In case of sabbaticals(research semester), adds it 3. if a provisional plan exists, then just update it 4. at the same time, wait for input from lecturer's side (lecturers participated in creating plan), they manage their assignments. 5. after the lecturers have confirmed the plan,the study director publishes the plan on a certain date
Postcondition	Concrete plan of assignments for lecturers is created for coming year.

Table 5: UC#1.1 Create a concrete plan

UC#1.2 Create a provisional plan

Study director can create a provisional assignment plan for lecturers for the next 2-3 years.

UC#2	**Use case description of Create a provisional plan**
Name	Create a provisional plan .
Summary	The Study director makes a provisional assignment plan for lecturers for the the next 2-3 years .
Actor	The Study director and the Lecturer
Precondition	Study director and the Lecturer have logged into the PLANA information system.
Description	1. The Study director opens assignment plan. 2. Makes a copy of concrete plan from one Semester to another one : • chooses assignment plan of modules or whole assignment plan and copies it to another Semester . 3. Or manually makes plan for the next year. 4. In the same time Lecturer opens his own plan of assignments 5. Check the assignment plan 6. Manages his assignments and confirms it. 7. The same procedure for other semesters, so that study director has a plan for the next 2-3 years.
Postcondition	Provisional assignment plan for lecturers is created for the next 2-3 years.

Table 6: UC#1.2 Create a provision plan

UC#1.3 Manage plan

The Study director and the Lecturer can manage an assignment plan. This includes adding, changing or deleting information in the assignment plan.

UC#1.3	**Manage plan**
Name	Manage plan .
Summary	The Study director and the Lecturer can manage plan of assignments for lecturers.
Actor	The Study director and the Lecturer
Precondition	The Study director and the Lecturer have logged into the PLANA information system.
Description	1. Open assignment plan 2. Adds /deletes/updates information in it.
Postcondition	Plan has been changed.

Table 7: UC#1.3 Manage plan

UC#1.3.1 View plan

The Lecturer can see his own plan for the next 2-3 years.

UC#5	**Use case description of "View plan"**
Name	View plan.
Summary	The Lecturer sees his own assignment plan.
Actor	The Lecturer.
Precondition	The Lecturer has logged into the PLANA information system.
Description	1. Opens assignment plan. 2. Sees all modules, additional assignments and other information about assignments for the next 2-3 years.
Postcondition	Lecturer has seen the assignment plan for the next 2-3 years.

Table 8: UC#1.3.1 View plan

UC#2 Manage project information

A lecturer can manage project information. This includes adding or deleting project information like information about which students work on which project.

UC#2	**Use case description of "Manage projects information"**
Name	Manage projects information.
Summary	The Lecturer adds students to his projects
Actor	The Lecturer.
Precondition	The Lecturer has logged into the PLANA information system.
Description	1. Open project list. 2. Add/delete/update information about projects.
Postcondition	Project information has been changed.

Table 9: UC#2 Manage project information

iii Use Case Diagram

Figure 3 shows the use cases of our system.

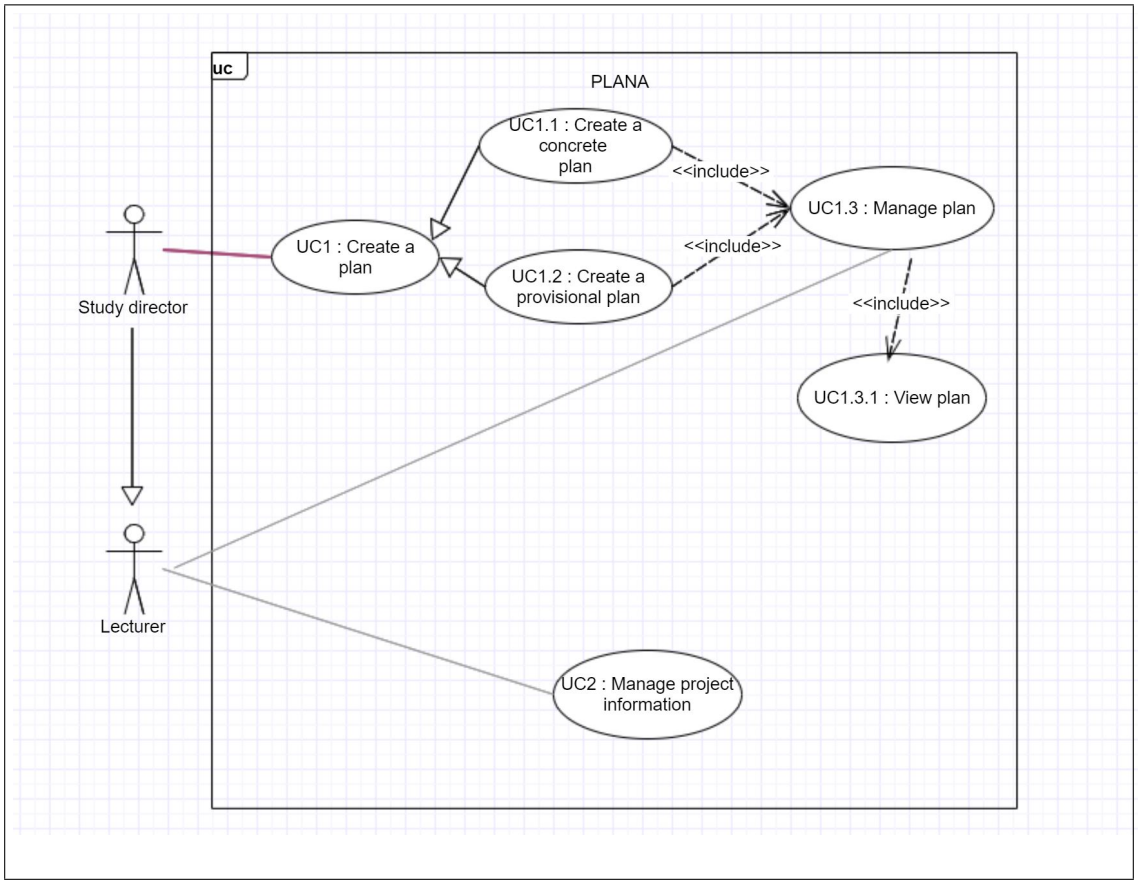

Figure 3: Use Case Diagram for the Study director and the Lecturer

5 Domain Analysis

5.1 Domain model

The domain model (Figure 4) shows us the important concept classes, associations and multiplicities between them.

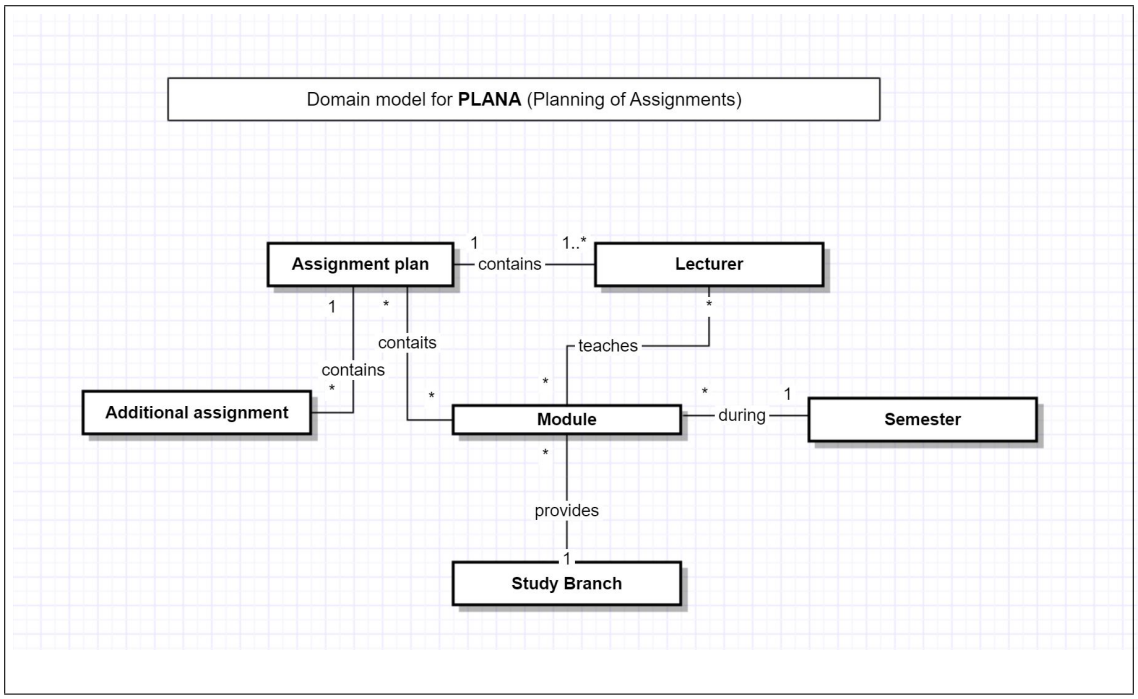

Figure 4: Domain model for PLANA

Figure 5 depicts an updated version of the domain model. In the second version of the domain model, we have made several changes. We deleted the concept class ”assignment plan” and we have connected lecturer with the module run and the additional assignments.

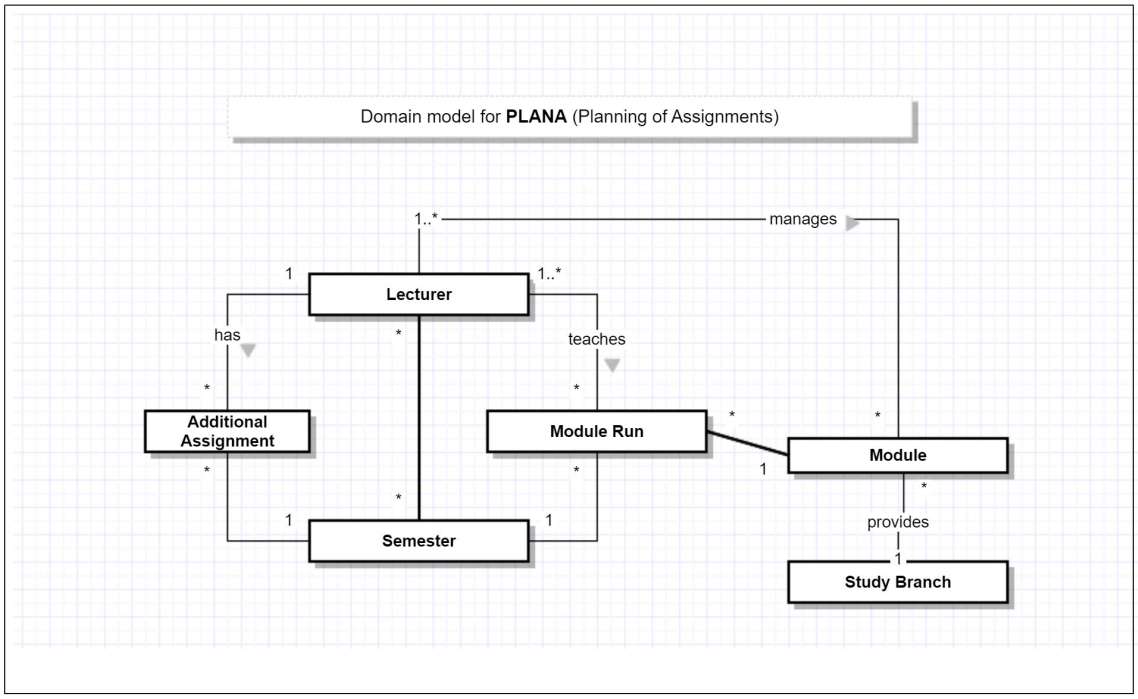

Figure 5: Domain model for PLANA

Figure 6 depicts a domain model with attributes

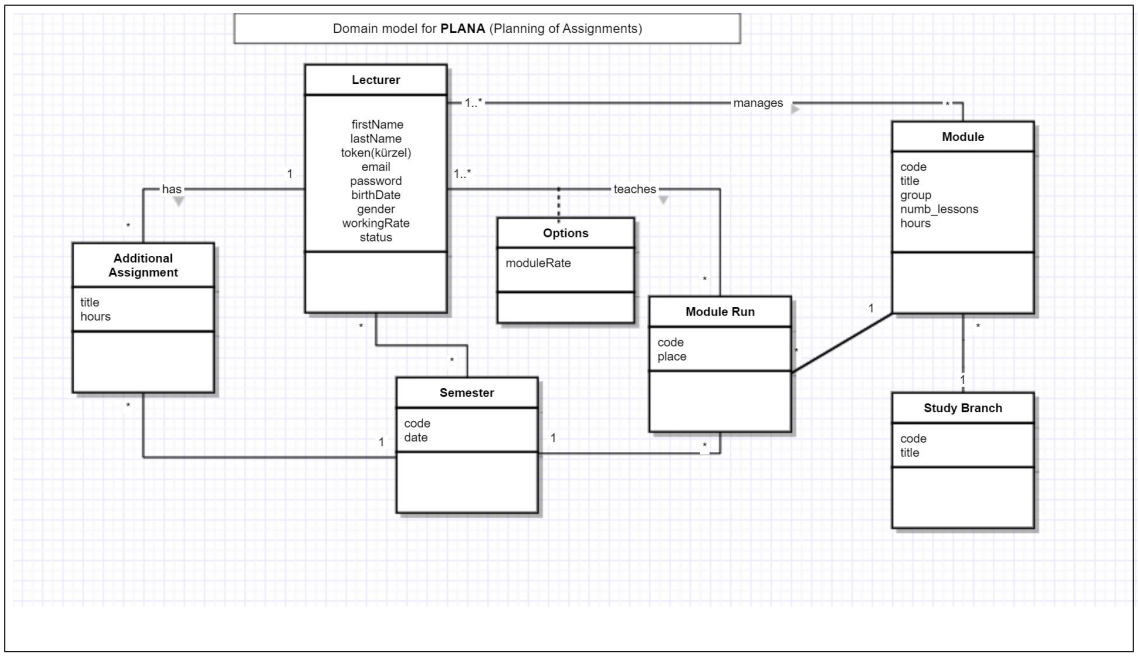

Figure 6: Domain model for PLANA

i. Concept Definition

- Concept class **Lecturer** models a person who teaches in a school.
- Concept class **Study Branch** models a conceptual subdivision of subjects that form a study programme.
- Concept class **Module** models a set of independent units that form a course at the school.
- Concept class **Module Run** models executions of a course in different languages.
- Concept class **Additional Assignment** models a set of independent units that form an additional task for the assignment's plan for the lecturer.
- Concept class **Semester** models the periods in the year, during which the lecturer is present in the school.

ii. Association Definition

Concept Pair	Association Definition	Association Name
Lecturer -> Module Run	Lecturer can teach zero or more Module Runs . Each Module Run can be taught by one or more Lecturers	teaches
Semester -> Module Run	Semester can include zero or more Module Runs .	includes
Semester -> Additional Assignment	Semester can include zero or more Additional Assignments	includes
Lecturer -> Additional Assignment	Lecturer can have zero or more Additional Assignments	has
Lecturer -> Module	Lecturer can manage zero or more Modules. A module can be managed by one or more Lecturers	manages
Module -> Module Run	Module is executed as many as there are module runs or not executed at all. A Module run is executed for one Module.	executes
Study Branch -> Module	Each Study Branch has many modules. These modules belong to exactly one study branch.	has
Semester -> Lecturer	In each Semester, there are many lecturers that are teaching, and these teachers are teaching in more than one Semester	includes

Table 10: Association Definition

6 System Architecture and System Design

6.1 a. Architecture Styles

The Client-Server model is our main architectural design choice. Such an architecture aims to provide access to one or more users to resources on the server.Therefore, we can split our system requirements into two systems.In One of them the client communicates with the server via the user interface, querying it for the data and waiting for the server's response. In the other system the server determines the capabilities of the visible and used data of each user, depending on the authorization. This separation makes it easier to program these systems.

6.2 b. Identifying Subsystems

The client-server model includes many types of tiers to describe the architecture of the system. Often there are three main parts of the tier architecture, i.e. presentation, business logic and data access and database.
In our project, we are using a *layered* architecture. Our layers are:

1. Presentation Layer
   Presentation Layer is the user interface layer, here we design our interface using different technologies

2. Business Logic Layer
   Business or Application Layer contains model classes, where the main business logic is encapsulated .

3. Data Access Layer
   This layer is needed for establishing a connection with the database server. This layer communicates with the business layer.

4. Database Layer
   Database Layer is the layer that contains our database

There are many advantages to layered applications. First of all, it will allow for more code reuse. Also, if the application gets more features, it will still be easy to maintain.

6.3 c. Persistence Data Storage

Our application requires a database. The data will be stored in an MS SQL(Microsoft SQL) database. The MS SQL database will consist of multiple tables. The main tables will be lecturer, module run, additional assignments. The lecturer table will inherit from the abstract class. At the moment, we do not have many users that will be interacting with the system, but the abstraction can be useful for future changes in this system.

6.4 Hardware Requirements

1. **End user** The end user will need an up-to-date web browser.
2. **Back-end** The back-end system will require a database to store all the information, i.e. user data, module-run data, additional assignment data, work-option data, etc.

7 Technologies

For our application, we have chosen the Blazor framework.
Blazor is a framework for building interactive client-side web UIs with .NET:[2]

- Create rich interactive UIs using C# instead of JavaScript.
- Share server-side and client-side app logic written in .NET.
- Render the UI as HTML and CSS for wide browser support, including mobile browsers.
- Integrate with modern hosting platforms, such as Docker.

Using .NET for client-side web development offers the following advantages:

- Write code in C# instead of JavaScript.
- Leverage the existing .NET ecosystem of .NET libraries.
- Share application logic between server and client.
- Benefit from .NET's performance, reliability, and security.
- Stay productive with Visual Studio on Windows, Linux, and macOS.
- Build on a common set of languages, frameworks, and tools that are stable, offer a lot of features, and easy to use. [3]

The Blazor framework offers two hosting models, the **Server side** and **WebAssembly client side** models.The server side hosting model is stable in situations with a small user database and local setups.
We will choose the Server side Blazor application, but a refactoring is possible in the future, when we have the client side ready because it can be preferable if the download time will be small enough.

For the client side Blazor application, we have different options to access the server-side data from a Blazor application. One of the options is to use the architecture that is shown in Figure 7.

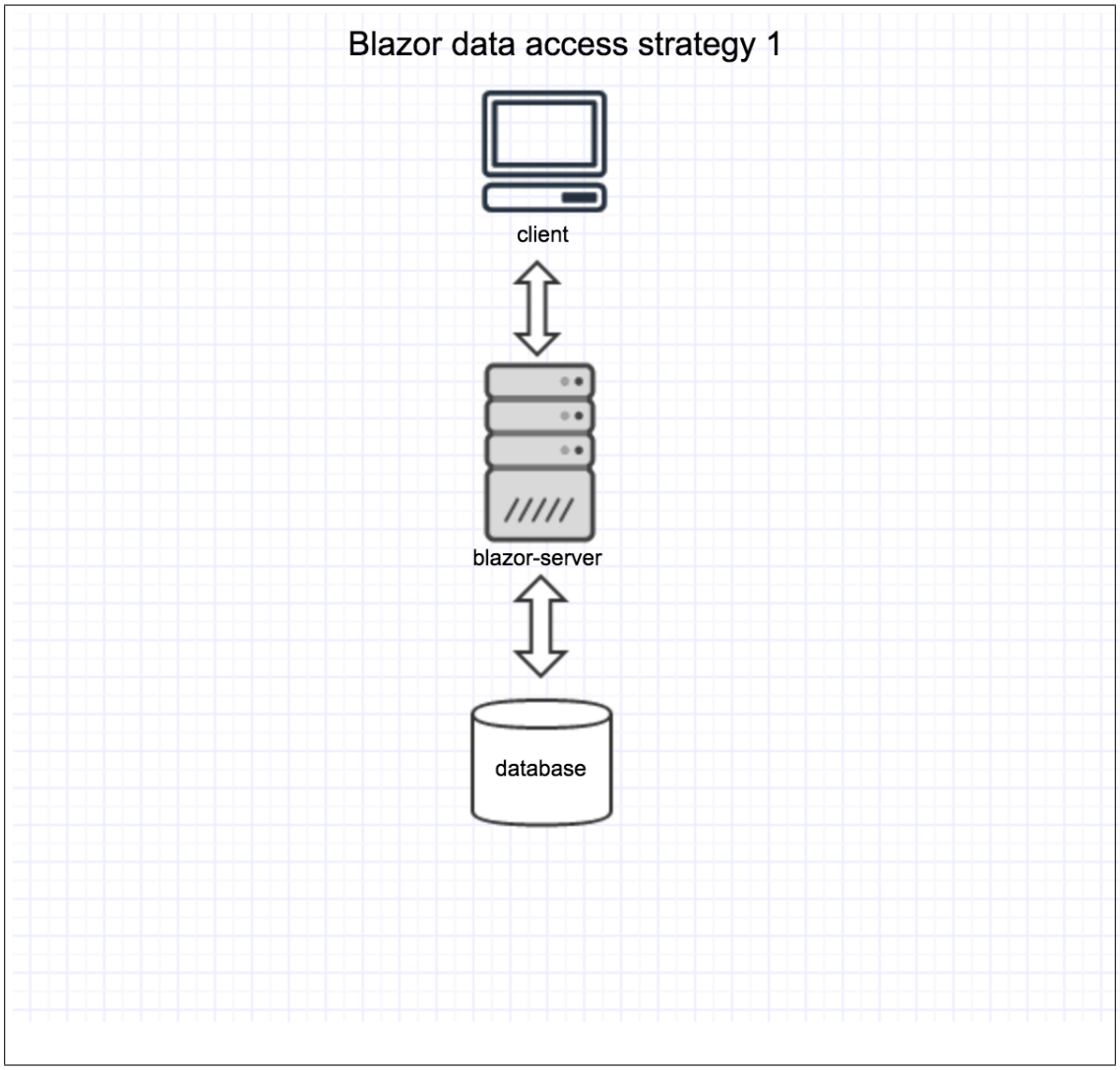

Figure 7: Blazor data access strategy 1

Here, the client calls the server application(Blazor application). Here, we utilize SignalIR, which is used to connect the client and server and exchange data between them. Figure 7 shows the example of the architecture, where the server has access to the database and can directly communicate with it. We can use this type of architecture when we do not plan to make any future changes in our application.
However, we know that it is possible to change our application from Blazor Server to Blazor WebAssembly, it is better to use another architecture, because with this one we would have to do a lot more work. The is the option when the Blazor application will always be running on the server.
Another option is to use the RESTful service in our application. In this case, the communication between the server and database is different. The Blazor application calls the RESTful service. Afterwards, this service calls the database via the Entity Framework

Core.

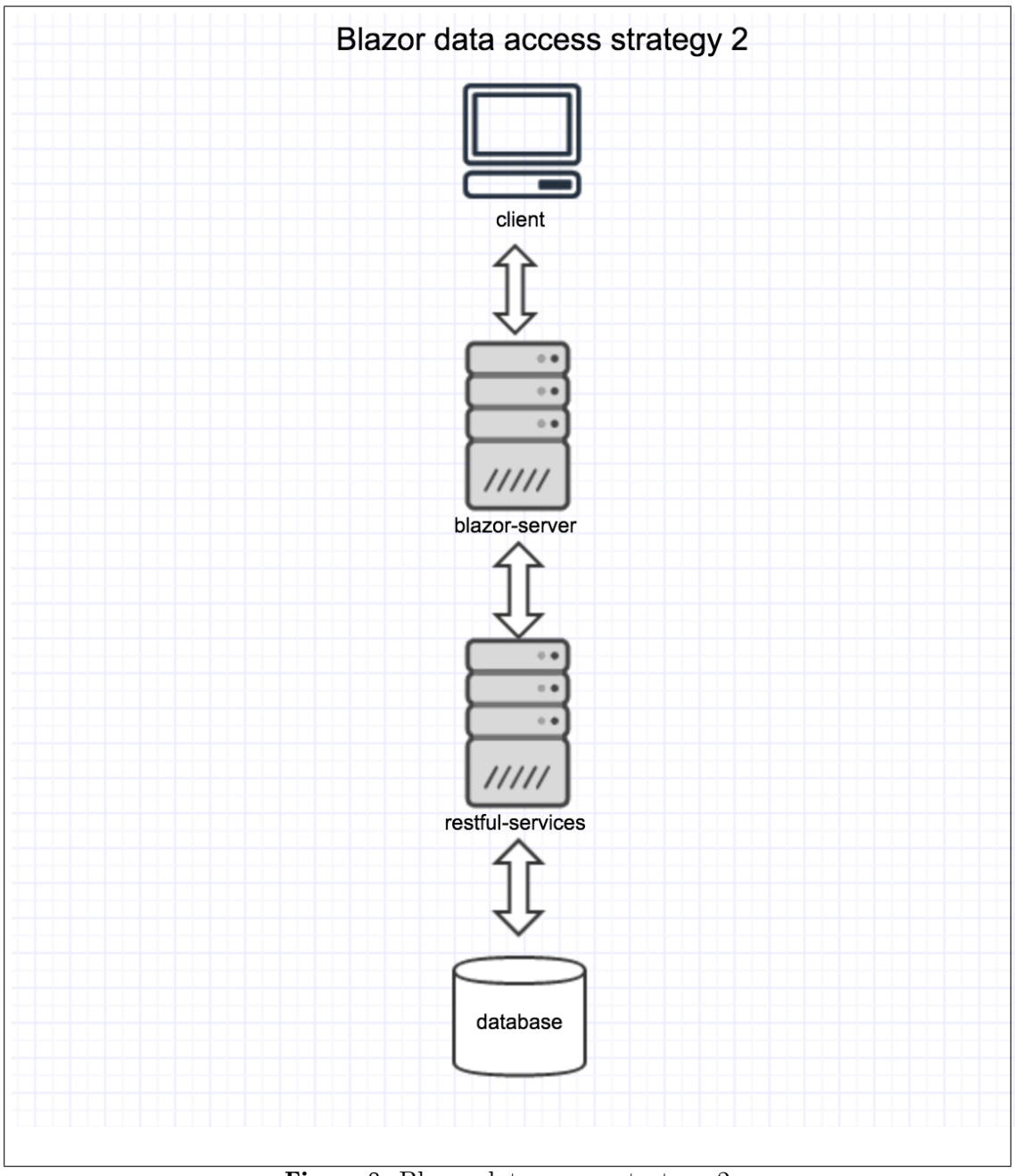

Figure 8: Blazor data access strategy 2

The benefit of this model (depicted in Figure 8) is that we can use this code for both Blazor models: Blazor Server and Blazor WebAssembly with only small changes. This model is more flexible.

Comparing these two options and knowing that in the future it is possible to make a transition to the client side version, we will use the RESTful service model RESTful APIs use the Representational State Transfer pattern to create an API(Application Programming Interface).

7.1 Setup for Blazor

To use the Blazor framework it is necessary to install :

- **.NET Core SDK 3.1 or later** from `http://dotnet.microsoft.com/download`
- **Visual Studio 2019** from `https://visualstudio.microsoft.com/downloads/`

8 Proof of Concepts

In this section, we want to collect information about the technologies which we chose. We also want to describe our first experience with these technologies.

As mentioned in the previous section for our project, we chose the Blazor framework.

Does it suit us? Does it meet the requirements of the system? How popular is it? Is its popularity growing or falling? Popularity is an important indicator of the effectiveness of this framework.

8.1 Popularity of Blazor

Based on information from the Internet, web frameworks like React, Angular, Vue.js are the most popular today.

In Figure 9 we have used the GitHub framework comparison to compare the history of stars of the most popular frameworks and Blazor. [4]

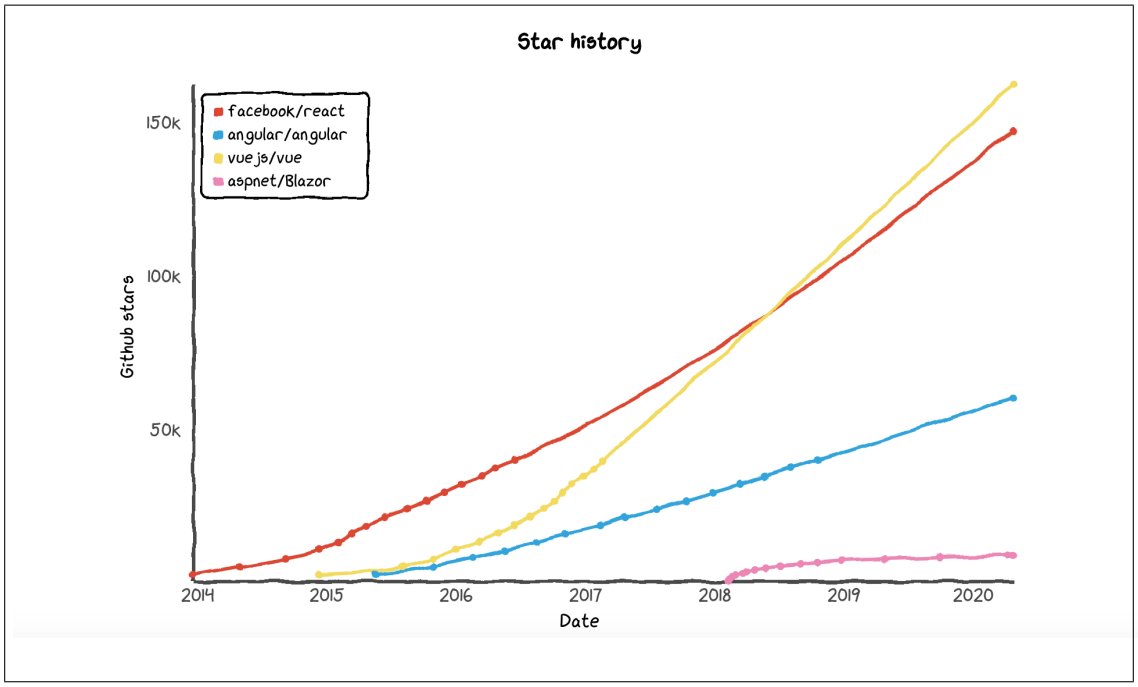

Figure 9: History of stars comparison of frameworks

If we compare the Blazor framework to these three most popular frameworks, we can see that it has a smaller number of stars, but this is due primarily to the fact that Blazor is a young framework.
If we consider Blazor separately, we see a progressive increase in popularity.

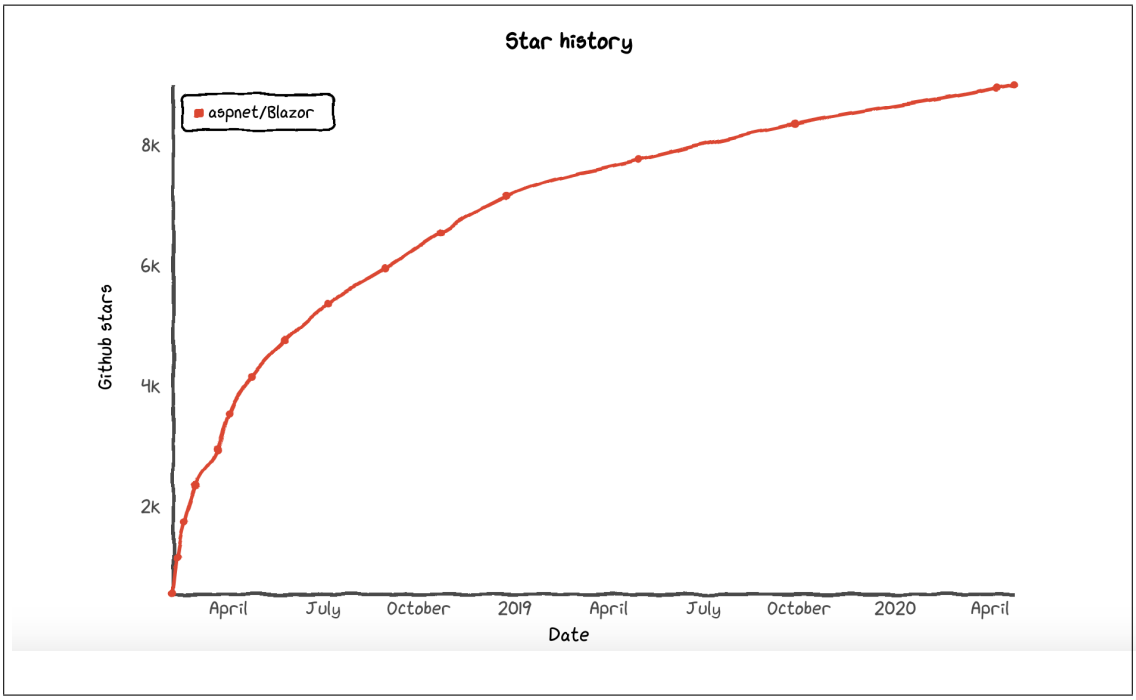

Figure 10: History of stars of Blazor framework

Another tool that we using to see the popularity of the Blazor is Google Trends. The Figure 11 shows us which frameworks are the most searched for in Google Search. [5]

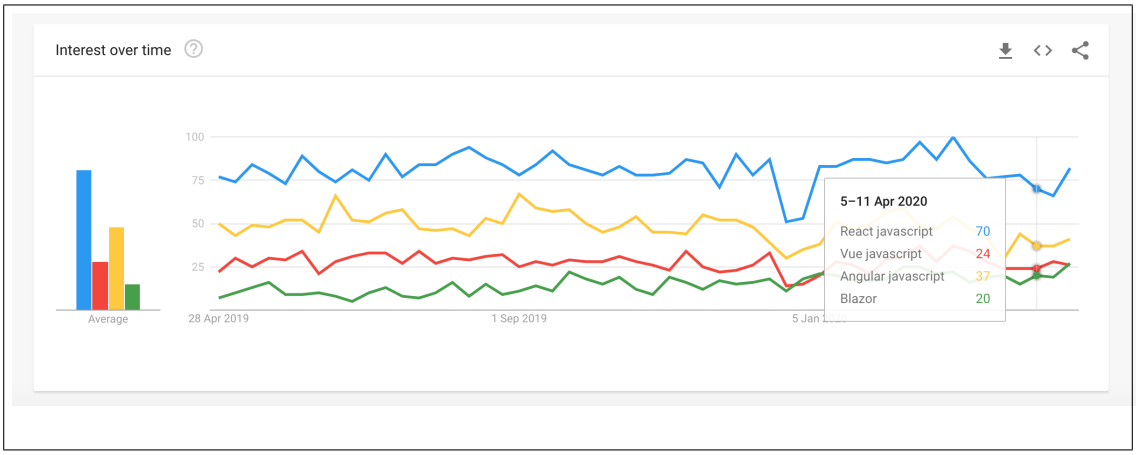

Figure 11: Interest over time of frameworks

As we can see, the React.js is the most popular framework in Google Search. Then comes Angular, followed by VueJs. Blazor is the youngest framework, but people are often interested in it. The Figure 12 shows us how the popularity of Blazor significantly increased in recent years. Which is an important positive indicator.

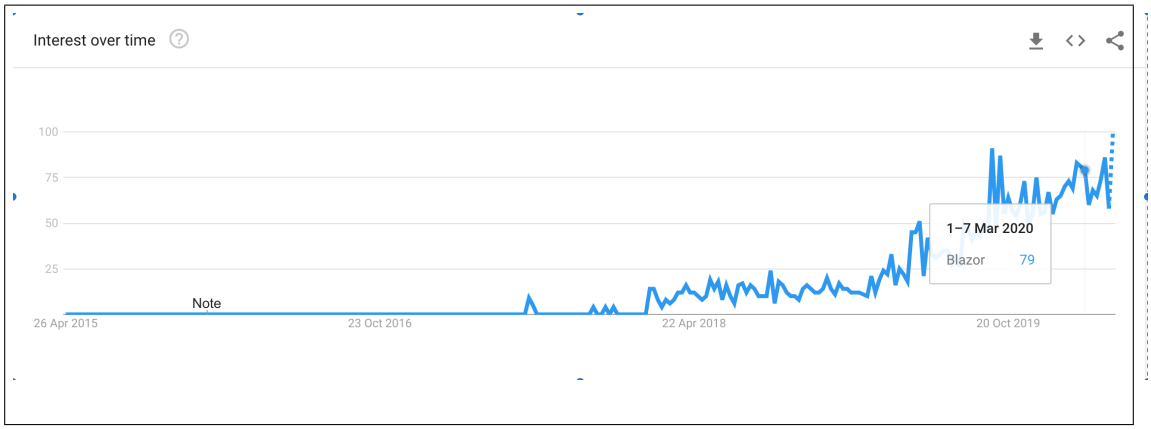

Figure 12: Interest over time in Blazor framework

8.2 Testing the Proposed Project Structure with Selected Technologies

Before starting with the main project, we have tested Blazor on small projects. We tried the WebAssembly client side model and the server side model. Then we have started with the main project and defined the use cases that we will implement as part of this project. All documentation for the implementation is in a separate document **" Visual Review In Initial Phase of Project.docx".**

Our experience with these technologies is quite positive. We managed to do the tasks in the allotted time frame. We have learned a lot and there is still a lot to learn, but despite all this, the result is nevertheless satisfactory.

8.3 Summary of Proof of Concepts

There are so many different possibilities to implement web applications using frameworks. When choosing a framework for our project, we based our decision on factors that are more suitable for us, i.e. functions that we need, architecture, programming language. We decided to create a project using the Blazor framework. It allows us to create a server-side and client-side applications using C#. We did research on the popularity of Blazor. Finally, to create a project using the Blazor framework. Summing up all this, we want to say that Blazor's popularity is growing, working with the Blazor framework is interesting and easy, the Blazor UI is pleasant to look at. We do not need to create separate model classes for the back-end and front-end. The same model classes are shared with both client and server.

9 Work Plan

9.1 Effort Estimation

Project 2 is designed as a 4 ECTS module. This corresponds to a workload of 120 hours. When we are working on a project, we always record our hours of work in an Excel table. At the end of the project, we will compare this time with the time allotted for the project.

9.2 Scrum

The foundation of the project organization was Scrum. Some principles of Scrum could not be achieved since they need a group of more than two people. Our work was based on the principles of Scrum like the Empirical Process of Control, the core of Scrum, self-organization, value-based prioritization, etc. The Empirical Process of Control includes three main ideas, namely transparency, inspection, and adaptation.
Transparency: The work is carried out in full trust of all parties involved. Everyone has the courage to keep each other up to date with both good and bad news.
Inspection: Inspection is carried out by every one in the Scrum Team. The team openly shows the product at the end of each Sprint.
Adaptation: The team asks constant questions about the progress of work, whether we are on the right way. Depending on this, we can adapt an existing product.

At the beginning of the project, we have discussed and estimated all the work that needs to be done. Meetings between supervisor and developer are weekly and sometimes bi-weekly. Each meeting includes a discussion about what has been achieved since the tasks have been assigned, what can be improved, and scheduling of future tasks.

Scrum Roles

- Product Owner: Mr. Pfahrer

- Development Team: Shiryagina Kristina
- Scrum Master: Shiryagina Kristina

Scrum Plan

To discuss the project, were weekly and biweekly meetings held . They included personal meetings, and then meetings using Microsoft Teams. The meetings consisted of:

- Sprint Review. It includes a show of work and its discussion.
- Sprint Planning. It includes the scheduling of future tasks.
- Sprint Retrospective. It includes discussion about what went well and what went wrong, what we should do differently.

Scrum Artefacts

Sprints

The sprints covered a one week period. At the end of each sprint, there was a discussion with the supervisor.

In Table 11 is an overview of what was achieved in which sprint.

Sprint	Goals and Achievements
1	• Make a proposal • Customer Problem Statement description • Asp.net core tutorials • Requirement engineering, User Stories
2	• Requirement engineering, Functional Requirements • Asp.net core tutorials • Problem Statement: Excel vs Web application
3	• Requirement engineering, Non-Functional Requirements • Actors and goals • Define the use cases • Small employee-management asp.net core project
4	• Small employee-management asp.net core project • Use case diagram • Blazor tutorials
5	• Domain model • Blazor tutorials • Blazor WebAssembly CRUDApp small project

Table 11: Sprints

Sprint	Goals and Achievements
6	• Update domain model • Add attributes to the domain model • Add new use cases
7	• Learn about Server side model of Blazor • Small project to-do list with Blazor
8	• Make BPMN2 process diagram with entities. • Update Excel vs Web Application diagram, add table
9	• User interface • Create services, components of Blazor
10	• Update BPMN • Start Chapter Proof of Concepts • Start project, Add initial structure of project, Add model classes
11	• Project: add database, add API project, add repositories and controller class • System Architecture and Design • Make table of comparison of Blazor with the most popular framework

Table 12: Sprints

Sprint	Goals and Achievements
12	• Technologies • Choose the use case for the programming part of the project
13	• Project, Add pages to the Blazor project • Make a description of my experience,which I have acquired while doing this project.
14	• Make a document - code review of initial steps of project • Describe components, services, controller and other important components of project in document
15	• Project: edit Lecturer and view Lecturer use cases • Describe work management
16	• Implement delete and search functions on front-end • Add conclusion part of the report.
17	• Prepare report and review documents for the final delivery • Prepare all documents and project for the final delivery • Add necessary data for the glossary, acronyms ,references.

Table 13: Sprints

10 Conclusions and Future Work

10.1 Conclusions

The purpose of this project is to prepare the necessary environment for the undergraduate project "Planning of the assignments information system(PLANA)" web application. To achieve this goal the following tasks were completed.

- **Requirements engineering**
  We determined the use cases of the system, actors, and all the necessary conditions that the system should satisfy.
- **Domain Analysis**
  We distinguished concept classes, designated interactions between them and created a domain model with its attributes.
- **Architecture and Design of System**
  We have chosen the **client-server** model for our system. We determined the subsystems and selected the MS SQL database.
- **Exploration of Technologies**
  We did a research on ASP.NET Core Blazor and we also made small projects based on this technology . They were executed successfully. The initial structure of the project has been created and the CRUD functions for lecturers have been implemented.

The conclusions that we want to draw are that the technologies that we tried for the project are fully suited to our goals. The project will continue to utilize technologies such as ASP.NET Core Blazor. The project structure has been defined, i.e the environment is prepared for the subsequent implementation of the project.

10.2 Future Work

As we mentioned above the bachelor thesis is a continuation and expansion of this project, for which we are planning the following future extensions:

- The embodiment of our intended use cases
  - Implementation of relationships between model classes
  - Creating the necessary controllers and repositories
  - Creating an easy to understand and user friendly interface
- System Testing implementation
  - Unit-Tests
  - Check-list for front-end
  - Postman for CRUD operation of controllers

- Database test
- Compatibility test
- Usability test

The final product should be a easy to use, easy to understand and user friendly application that will fulfil the main goal - ”Planning of assignments for lecturers”, which will be implemented on the latest technology - ASP.NET Core Blazor.

References

[1] P. Eeles, "Capturing architectural requirements," *IBM Rational developer works*, 2005.

[2] microsoft.com, [Online]. Available: `https://docs.microsoft.com/en-gb/aspnet/core/blazor/`.

[3] M. Pfahrer, *E-business and web – web technologies, blazor, 2019/20.*

[4] Github, *Star history.* [Online]. Available: `https://star-history.t9t.io/`.

[5] Trends.google.com, *Interest over time.* [Online]. Available: `https://trends.google.com/trends/explore?cat=733&date=today%205-y&q=React,Vue,Angular,blazor`.

Document structure
`https://www.ece.rutgers.edu`

11 Protocol

Frequency: (weekly)
Meeting length: (60 minutes)

Agenda

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

Report from 26.02.20
Plan
Future goals are:

- Requirements
- asp.net core tutorials

Lessons learned

Next Meeting: .....

Report from 23.03.20
Plan
Future goals are:

- Domain model update
- Add attributes to the domain model
- Describe actors
- Add 2 more use cases
- Change use case diagram, add requirements for these new use cases
- Add changes to problem statement and make it more understandable.

Lessons learned

Next Meeting: 30.03.20

Report from 30.03.20
Plan
Future goals are:

- Make BMN2 process diagram with entities. This diagram is a good tool to show who does what.
- In the Problem statement, transform the information (Excel in comparison to the web app) from a diagram to a table.
- FURPS+ (source information)
- Better describe the problem statement
- Change uc6 (delete Institute-manager actor or extend uc6 with uc6.1 ”list of research assignments”)
- Domain model will be discussed with the supervisor

Lessons learned
Important to give information in an easy to understand/read format, so that the reader can understand it without reading the report several times, e.g. table form.
Next Meeting: 14.04.20(09:00-10:00)

Report from 14.03.20
Plan
Future goals are:

- Change bpmn, first create a process of s. director, then the joint process of plan managing
- Add Chapter Proof of Concepts (prototyping)
- Start document chapters: 6-8
- Start programming

Lessons learned

Next Meeting: 20.04.20(09:00-10:00)

Report from 20.04.20
Plan
Future goals are:

- Continue to describe chapters 6-8(change 7<->8) For chapter 7 (poc) make a table with comparing different popular frameworks with blazor
- Implement model classes for PLANA

Lessons learned

Next Meeting: 27.04.20(09:00-10:00)

Report from 27.04.20
Plan
Future goals are:

- For a document:
    1. Determine what exactly will we program in project 2
    2. Describe it in the separeted document "Visual Review In Initial Phase of Project"
    3. After trying to write code and implement several functions, describe it and write in the conclusions our experience. Will it be easy to create the application, or are there some problems which we will have to address before proceeding with the bachelor thesis.
-

Lessons learned: How to make Proof of concepts

Next Meeting: 11.05.20

Report from 11.05.20
Plan
Future goals are:

- Put more information regarding implementation description of overview document, add diagram project dependencies. Describe component, services, controller, how they interact with each other.
- Add the work plan in the main document.
- Continue to code the project

Lessons learned
Try to see the content of the documentation from the side of the person who is reading the report. Putting the diagrams in the document makes it easier to understand the given information.
Next Meeting: 25.05.20

Report from 25.05.20
Plan
Future goals are:

- Implement delete and search function on front-end

- Write the conclusion of the report . Prepare report and review documents for the final delivery.
- Prepare all documents and project for the final delivery

Lessons learned
soft-delete
Next Meeting: 08.06.20

Report from 08.06.20
Plan
Future goals are:

- Finish writing code
- Prepare all documents and project for the final delivery

Lessons learned