PROEP 2020

Process Report

Teacher: Mariëlle Fransen
Date: 27-01-2021

Project Group
Rahul Annadurai - 3413632
Anh Le - 3274950
Barra Abd Al Fattah - 3004503
Bao Quoc Nguyen - 3477347
Lauren van Loo - 3238849

Summary

This document gives an overview of the process of the entire project. The project mainly focuses on implementing the student networking system, which helps students easily find and work together on a project. The system also represents the main purpose of this course, implementing a distributed system.

The project consisted of nine sprints divided into the four phases of the project: Initialization, Research & Design, Implementation & Testing, and Finalization. At the start of each sprint, we had a team meeting to summarize and keep track of our progress in our project. At the end of the sprint, we had a meeting with our mentor to keep her up to date with our progress and make plans for the incoming sprint.

Initialization
During the first phase and sprint, we brainstormed for ideas and decided to go for the one with the highest potential and interest. The initial definition, scope, and functionalities of the project were written down in a project plan, and we made the first two levels of the C4 model.

Research & Design
In this phase, we put our focus on researching possibilities, tools that would help us carry out the project efficiently. For the design part we made documentation such as Use case document, Wireframe, e.g. to help us visualize the flow of the application.

Implementation & Testing
During this phase, firstly, we implemented all the must have features of our application. Afterwards, we spent our time implementing some of the extra features for our application.

Finalization
For the final phase of our project, the team spent time finalizing our documentation, application and preparing for our final presentation.

The result from this project was that all of the agreed functionalities were completed, and we also implemented some of the extra features for our application. All the agreed documentations were delivered to our mentor at the end of our project.

Overall, we think that this is definitely an interesting project. Throughout the project, we have acquired valuable knowledge in technical skills as well as interpersonal skills. We got a chance to learn new frameworks and tools that would benefit us in our future project. In addition, we gained much experience in planning a project. This would definitely be a valuable skill for us in our upcoming graduation

Glossary

Name	Description
Azure cognitive search	A cloud based search service provided by Microsoft
ASP.net core	A free and open-source web framework provided by Microsoft
Fontys API	Is an application programming interface that enables third party applications to interact with fontys programs and data.
Laravel	An open-source PHP web framework
Lucene library	A free and open-source search engine software library
PHPMyadmin	A free open source administration tool for MySQL and MariaDB
ReactJS	An open-source, front end, JavaScript library for building user interfaces or UI components

Table of Contents

Chapter 1: Project Overview

This section provides an overview of the programming languages used, the working methodologies and briefly explains the features.

The application is intended for students to find teammates for personal interest projects. So there are quite a few key features that had to be included. Search functionality was of high priority because it makes the website so much more easier to work with and search for projects. Another high priority feature for the website was for authentication to make sure that only Fontys students can join the website and that was achieved by using Fonty's sign up service. Besides these main functionality all the other basic but essential functions are pressend such as login, delete account, retrieve password, create project, delete project, update project, adding members to projects and many as such.

As far as the programming languages being used are concerned we used React JS for the front end development. The reason for using React JS for the front end was the ease of reusability of components. And for the back end development Laravel was used because of its useful functionalities and prior knowledge of the framework.

The team decided to follow the Waterfall model combined with the SCRUM framework in development. With the waterfall model, the project was divided into 4 phases. The phases include Initialization, Research and Design, Implementation and Testing, Finalization. SCRUM framework was applied in the implementation phase.

Chapter 2: Process and Results

This section discusses the process and the results for each of the nine sprints in the project. The project was divided into four phases; Initialization, Research & Design, Implementation & Testing, and Finalization. The table below shows an overview of the project phasing and the sprints. Each sprint takes around two weeks.

Project Phase	**Sprint(s)**	**From**	**To**
Initialization	1	Week 1	Week 2
Research & Design	2 - 3	Week 3	Week 6
Implementation & Testing	4 - 8	Week 7	Week 16
Finalization	9	Week 17	Week 18

2.1. Initialization

Sprint 1

Task

Sprint one represents the beginning of the project, during this sprint we had to come up with a project idea and make a project plan.

Process

During the first few days of the sprint, we came up with three ideas for the project. The first idea was more health- and food-related and focused on connecting the community with the farmer. The other ideas were related to students; a student networking application for activities like parties and an application for finding side projects and/or finding teammates for projects. For each of the ideas, we discussed the possibilities and functionalities. Eventually, we decided on creating an application intended for students to find teammates for projects. This decision was made based on the potential of the idea and preference vote from each team member.

After we decided on the project and it got approved by Mariëlle Fransen, we started defining the project in the second week. We discussed and determined the basic components, the features, and the scope of the application, which was then written down in the project plan. Besides the project plan we also worked on the C4 model (Level 1 and 2) to outline the software architecture of our application.

Result

By the end of the first sprint, we decided on the general idea of the project, got it approved, and completed the project plan. Besides, we finished Level 1 and 2 of the C4 model. The project will be more defined in the next phase Research & Design.

2.2. Research & Design

This is the second phase of the project. During this phase, we mainly focused on research and design, made documentation and planned our project in more detail. With this we came up with the following list:

- Research:
    - React JS library
    - Convert react web application to mobile application
    - Entity framework design
    - Azure Cognitive Search
    - Swagger
    - Research about GDPR
    - Fontys API
- Documentation:
    - User case document
    - Wireframes
    - Style Guide
    - Class Diagram
    - Database design: ERD
    - API documentation

Sprint 2

Task

Research:

- Swagger
- GDPR
- Entity framework design
- Fontys API

Documentation:

- Use case document
- Wireframes
- Class diagrams
- ERD

Process

Firstly, we tended to the use case document as we thought it would be the most important part of the sprint. With the use case document, we could visualize the whole application flow. In addition, it also acted as an agreement between all the stakeholders. For this, we had several team meetings to discuss and decide how our application should behave. After that, we had a meeting with our mentor to ask for feedback and possible improvements.

After finishing the use case documentation, we moved on to work with the Wireframes, class diagrams and the ERD. All these designs followed closely with our use case document.

During the process, we also made our research about Swagger, GDPR and Entity framework design. The main reasons were that:

- Swagger: provide a tool for describing Restful APIs expressed in JSON. This would benefit us when we connected our front-end and back-end application
- GDPR: This is because the European Union had a very strict data protection regulation. In our project we were going to use and process personal data, and thus we needed to carefully research about this regulation to see what rules we need to oblige to.
- Entity framework design: this was mainly used to apply a good design pattern to our WebAPI application. With EFD, we can easily maintain and extend our application.

During this sprint, we also made research about the possibilities of using Fontys API in our application. The main reason for this was that we get the students' information without letting them insert it manually as well as restrict the application to the students of Fontys. We asked Fontys for a developer account that we can use during the research as well as we read the documentation of Fontys API.

Result

At the end of the sprint, we finished the Use case documentation and wireframes. For the Class diagrams and the ERD, we finished with a draft version, and they would be improved in the latter sprint. In addition, we made a meeting with our mentor and from the meeting we got approved for our use case documentation and a suggestion for making the endorsement feature of our application. In addition we understood how the Fontys API works and what data we are allowed to get and use in our application.

Sprint 3

Task

Research:

- Convert web application to mobile application
- Azure Cognitive Search
- Studiekeuzedatabase API
- Design the application API

Document:

- Finishing the class diagram and the ERD
- API documentation
- Style guide

Process

We were creating the concept of API documentation containing API endpoints and their explanation. After discussion and agreement, we finalized the API documentation.

From the draft version of ERD and wireframes in the sprint two, we pointed out the weaknesses of the current design and conducted the improvements so that ERD, wireframes and API endpoints can be consistent. When the wireframes were improved and finished, the team made a style guide to use during the development of the front-end. The decision to create a style guide

was made so that everyone would immediately implement the correct styling and there would not be any major differences.

The group came up with the idea of converting the web application to the mobile application. Our application is built in ReactJS and the corresponding mobile application is supposed to be built in React Native. The question is how to convert a ReactJS app to React Native. After researching, the team recognized that it was impossible to use the whole code from the web application to the mobile application. While business-logic can be reused, most components need to be re-written. The quickest way is to follow the React Native architecture and develop our user interface using reactive native components. Due to the time restriction, the team decided to focus on the web application and the promised features in the project plan.

During our process, we found out that our traditional database (PHPMyAdmin) was not good enough to provide a good search engine. Thus, we made an investigation into search services provided by third party companies. For this we derived the following list:

- Algolia: a U.S. startup company offer web search product
- ElasticSearch: provides a search engine based on Lucene Library
- Azure Cognitive Search: a search engine provided by Microsoft

In general, all these search services provide all the necessary features for us to implement a good search engine for our website. One of the main differences that we took into consideration was the pricing tier of each of these products. With Algolia and ElasticSearch, the features provided come with different pricing tier. In contrast, the only difference in AzureCS pricing tier was the amount of storage for our search index. Thus, we decided to use AzureCS as we could use all the search features and our data storage was not that large.

The team researched different API options, one of which was the Studiekeuzedatabase REST-API. This API was interesting for our project as it could provide us with all the study programs and universities in the Netherlands. We wanted to add a functionality to the website so that the user can select his or her specific study program and university. However, in the end we did not get access to the Studiekeuzedatabase API.

During this sprint, we also did research about how to design an API in such a way that it is easy to support multiple clients.

Result

The end of the sprint 3 is also the end of the research and design phase. By the end of this sprint we finished all the tasks that we planned. We made a small meeting with our mentor to deliver the result and a team meeting for planning the next phase.

2.3. Implementation & Testing

Finishing the Research and Design phase, the team proceeded to the Implementation and Testing phase. The focus was to develop our web application and test the developed features. The team fixed the bugs and enhanced our code qualities.

To complete the application, we broke down the required functionalities into different parts. At the beginning of each sprint, the team created the sprint plan to concentrate on a part of features.

Sprint 4

Task

Sprint 4 marks the starting time of the implementation and testing phase. For the front-end, the team aimed to select the design framework and created the basic layout of the website. For the back-end, the team selected the back-end framework, created the database and initialized the project.

Process

One of our biggest challenges is that most team members did not have any experience with ReactJS - the framework in the front-end and needs to study the framework and work on the project at the same time. The reason that this framework was picked is that ReactJS offers fast rendering, clean abstraction, gentle learning curve, and reusable components. We spent a large amount of effort and time learning this library during the sprint 4.

To approach the project, the team decided to pick a front-end design framework. Some ideal options are Bootstrap, Skeleton and Material UI. Each of these frameworks exposes both advantages and disadvantages. Bootstrap offers fast prototyping, responsive design and a large collection of components. Bootstrap is the most popular design framework and easy to use. However, the built-in designs are too familiar and basic. Like Bootstrap, Material UI‘s advantages are rapid prototyping and has a large range of components. The cons of Material UI are that the documentation and community is not as large as Bootstrap. Skeleton is a small CSS library that offers a limited collection of theming and is most suitable for a small-scaled project. Considering both pros and cons of all libraries or frameworks, the team decided to utilize Bootstrap for the basic layouts as header, footer, panel, etc. To enhance the design of the website, Material UI and Skeleton are used for small details naming loading, autocomplete, etc.

The next step was to decide on the structure of the application and create the basic layouts of the web application. Another challenge was that React does not require a specific architecture as other front-end frameworks (eg: AngularJS, VueJS). For example, AngularJS uses Model-View-Controller Architecture or VueJS works with Model-View-View-Model Architecture. It was a challenge for the team to decide on the structure of the application. After trials, the team decided to stick with the idea of creating one component folder for each web page. This approach offered the easy to divide the task among members and refactor the code.

Then we had to decide which back-end framework we are going to use. We had two opinions, the first option was to use Laravel, and the second was to use asp.net core. Each framework has its own advantages and disadvantages. Because Laravel is powerful when it comes to making an API service and it was taught at Fontys, we decided to work with Laravel. After that we created the database and initialized the project. Next, we had to connect the Laravel project with the database and define the relations between the tables in Laravel so the team can easily work with the database in the project. Another challenge is that the back-end is handled by two team members.

Result

At the end of sprint 4, we finished the layouts of the web application and agreed on the structure, the design framework of the application. This was the gateway for every member to have clear ideas of the next working steps. In addition, we created the database and initialized the backend project.

Sprint 5

Task

During the fifth sprint the team was focused on getting the front-end completed without the connection to the back-end API. For the back-end part of the project we focused on creating the project endpoints and connecting our API with Fontys API.

Process

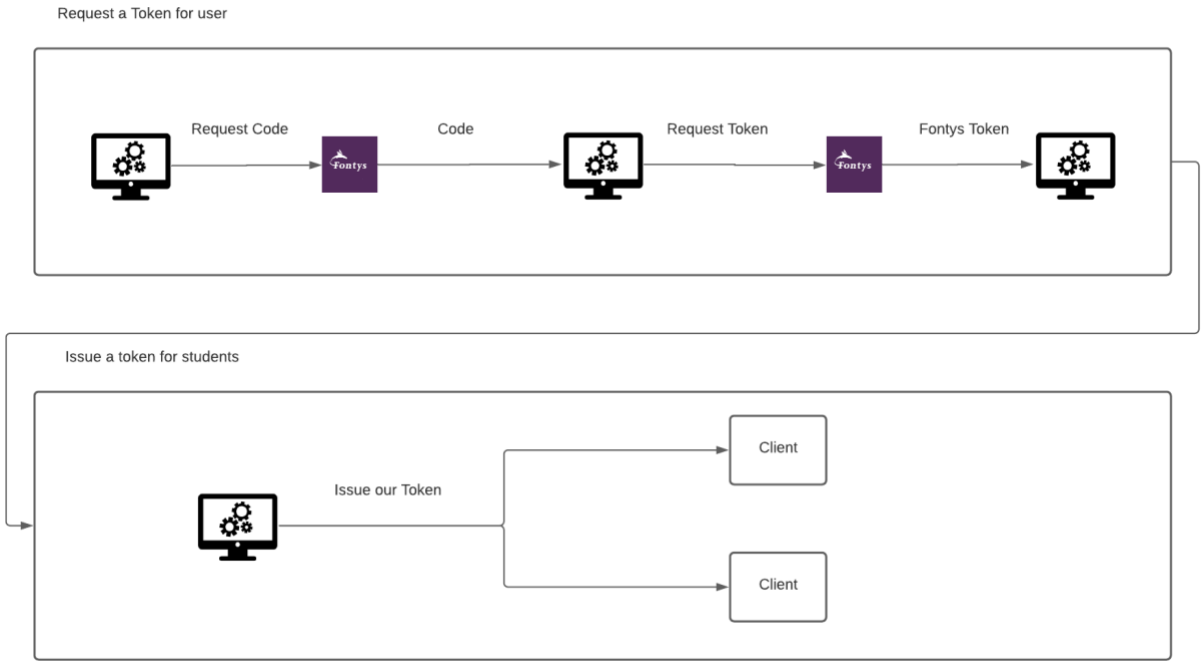
To ensure a smooth work process we divided the project into separate parts and assigned them to each team member.

For the frontend we had a small issue that some code was written with functional components and others with class-based components. After carefully considering the two, we decided to go for functional components to remain the consistency within the frontend. In either case we had to change some of the code that was already written. The team decided on functional components as they are not only getting used more often in React projects, but they are also less complex and allow the developer to achieve the same goal with less code.

During this phase we also made some changes to the initial idea of the student skills. The initial idea was to let the user add a level of proficiency for each skill, however the implementation and design for this did not work out nicely in the end. Therefore, we decided to drop this functionality and replace it with endorsements. This entails that the user has a number of skills that other users can endorse if they think the user is good at a certain skill. As a result, we started to make some changes in the frontend and backend to add the endorsements' functionality. For implementing the skills, we decided to go with Material-UIs chip component. It has some great advantages over Bootstraps' version, like prebuild click and delete functionalities, and it is fully customizable. Besides, Material-UI has an autocomplete search component that allows us to show the selected values as the chips. We decided to use the autocomplete for searching skills in the add and delete functionalities, which meant changing the design slightly.

For the back-end, we started to make the connection between Fontys API and our application and make needed logic and endpoints to it works. We used Fontys API for student registration. We did it in such a way that the student can only register using their Fontys account. After registration, our website saves the needed information of the registered student from Fontys server like the first name, last name, title, and the studentId. After that the information is successfully saved, we issue a token to that student. We made the token in such a way that our clients can only use it in our project and it can not be used to perform actions on Fontys server. In that way we provide an extra security layer to our application as well as we protect the students from having someone else performing unwanted actions using their names on Fontys server in case the token is stolen.

The photo below describes how our application interacts with Fontys API and issues tokens for the students.



After the registration in the back-end was made, the login logic, and endpoints were also made. In addition, we managed to finish all of the project routes for the application. These route including create project, create join request from a student to the project, e.g. We also managed to send an email to the students to notify whether they were accepted or rejected from the project. Overall, we did not have much problem during the implementation of this part of the application

Result

At the end of Sprint 5, we converted all class components into functional components to acquire the consistency in the front-end code and accomplished the code without the connection with the backend API in the following webpages: login, signup, landingPage, homePage, ProjectDetails and ProfileDetails. The result of this sprint was making the connection to Fontys API as well as

making the routes and the logic of signup and login. In addition the routes related to a project in our application were also made.

Sprint 6

Task

At the sprint 6, the team finalized the front-end code in ReactJS without the connection with our API and merged our individual parts into a complete website. For the back-end we made the connection to AzureSearchService, created a search index, and implemented our search feature.

Process

To complete the front-end code without the communication with the back-end, the team needed to make adjustments on the business logics. Initially, the idea was to add skills when an user creates a project. However, we recognized that we must add the spots when creating a project. We decided to create the project first, then create the spot and add skills to the spots. In the other words, we will save the array of spots once the project is already created. To guarantee the responsiveness of the web in the view of a mobile app or tablet, we needed to adjust the design and tested the adjustment in the screen size of a mobile application or a tablet. Completing the must-requirements, the team developed the searching function and suggestion feature. In the front-end, we had to separate the logics and view into different components to ensure the code quality and extensibility of the feature. In the back-end, we use AzureCS - a Python wrapper for the client-side APIs of Microsoft Azure Cognitive Services - to handle this feature.

For connecting to AzureCS, we first needed to create an AzureCS resource with our Fontys account and get the API key. With the search index, since out DB is not supported by Azure we could not automatically import the structure of it. Thus, for this part, we created a small Console Application to handle the search index creation and data upload to AzureCS. In this index, we discussed which features that we would need to implement for our web search engine and design the structure of the index accordingly. During designing the search index structure, we had a small team meeting to decide how it should be created fit our whole application. In this meeting we made sure that our design went accordingly with the application and what we agreed before in the documentation. After finishing with the search index, we implemented our search function with AzureCS search API. We did not encounter any major problem during this part of the project. There were some minor mistakes when connecting to Azure, but we managed to fix it by the end of this sprint.

Result

Completing the sprint 6, we were done with all the code in ReactJS for the whole website except for the connection with the API. Regarding the back-end, we finished the basic endpoints such as project, student, skill and were done with the extra feature: AzureSearchService.

Sprint 7

Task

Finishing the code for front-end, the front-end team proceeded to connect the front-end with the back-end API. At the same time, the back-end created new routes for students, spots and skill matching, implemented filtering projects (based on a category), typeahead (autocomplete) feature and improved the code quality. Protecting the routes as well as making the input validation are also tasks of this sprint.

Process

The communication between the front-end and back-end is the most challenging with us. Due to the lack of experience in ReactJS, the team spent a large amount of time investigating the most efficient way to place the code and ensure the extensibility. The team decided to create a config file that contains the domain and header of the URL. This helps us to reuse and maintain the code in case the domain needs to change. We had a lot of bugs in the connections. However, we communicated regularly though WhatsApp and MS Teams to test the features and fix the bugs.

Being inspired by the functionalities of LinkedIn, the team came up with an idea of developing the skill matching feature. The logic behind is that the feature provides the user with the projects matching with the user‘s skills. The challenge is based on which standards or logics to decide the matching projects. After discussion and investigation, the team decided to compare the user skills with the required skills of each spot in all projects. Once the project contains a skill of the user, it will be displayed in the list of projects found.

We had some issues with the implementation of the student skills in the frontend. The user was able to add the same skill twice and delete skills that were not in his profile, resulting in errors in the backend. To solve these issues, we changed the code in such a way that it will only show skills that are not added to the user’s profile and add a validation that will notify the user in case the skill has already been added. For deleting we made sure it only shows the skills the user has added. This however resulted in another issue causing the view not to update whenever the user adds and deletes a skill from his profile. As a result, the add and delete parts were showing the wrong skills. In the end, this issue was easily fixed by adding a special React Hook (UseEffect) which will re-render the component in case the data changes. To finish up the skill part, we needed to add the endorsements. Implementing the endorsements in the frontend was a bit challenging as a user can only endorse a skill once and cannot endorse himself. Besides, we required different skill types as some skills do not have endorsements. To be able to show everything nicely, we decided to use the following logic:

- If the skill is endorsed, we always show endorsements as a number next to the name of the skill.
- If the student has different types of skills, we will divide them in “Personal & Interpersonal Skills” and “Industry Knowledge”.
- If the user has six or more skills and at least three endorsed skills, we will show three highlighted skills with a picture and name of an endorser.

For the data of the skill we used a dataset from LinkedIn with over 300 skills and skill types

With autocomplete functionality, we allowed users to gradually discover their search term. For example, if a user typed “soft”, we could return software as a possible result. Our autocomplete feature operated on the project search index. In particular, project name, description and category. The reason we chose this field for our autocomplete feature was that we thought that this would be in the users mind when they searched for projects in our website

With the filtering feature, we allowed a user to filter projects based on their category in the database. For instance, a user can filter a project based on the “Software” category, and all software projects would be returned in the result. For skill matching, we made use of AzureCS search function. We provided the search API with a list of student skills and AzureCS would return a list of project recommendations for the student based on his or her skills. In the returned results, the most relevant projects(the skills of that project has the best match for the student skill), would be returned as top results, the higher the order of the project in the result list, the less relevant(or less match) of that project to the student skills. Overally, we did not encounter much issue for this part of the application. The main reason was due to the fact that we had a good discussion and a well-structured search index that allowed us to easily apply AzureCS APIs to our website search functionalities.

The logic behind creating, reading, updating and deleting a student, a skill and a spot was implemented as well as the endpoints related to it. The input validation for each update and delete endpoint was also implemented in order to protect our routes and database from SQL injection and to keep the data in the database clean and well structured. In case the front-end sends an unvalid request to the back-end, the back-end returns a response with a message related to that invalid input. The route protection was also made to check if the student has the right privilege to perform an action on the data. For example, only the account owner can delete or update his/her information.

Result

At the end of sprint 7, the team completed the Homepage, Landing, Login, ProjectDetailsPublic, ProfileDetailsPublic, ProjectDetailsPrivate, ProfileDetailsPrivate and was in the process of developing the extra features. For the back-end part of the application, we managed to finish autocomplete, filtering and skill match as extra features for our web application. In addition, routes of student, skill and spot were implemented with input validation as well as route protection.

Sprint 8

Task

In the last sprint of implementation and testing phase, the team finalized the application and tested all the developed features. We concentrated on fixing the bugs and improved the code quality. Making the endorsement as well as sending email to students were also tasks for this sprint.

In addition to the application, Trello and 'Work division' excel file are our tools for keeping track of each team member 's progress.

Process

To improve the user experience, we created responsive content loaders in the frontend that show up if the application is waiting for the data. These loaders ensure that the user is aware that the application is still working and also is not looking at a broken website. Initially we went with a react-content-loader package as it came with a website in which we could easily design the loaders specifically for our website. However, in the end these loaders were not responsive and really difficult to adjust. As a solution we switched to the React Placeholder extension. React Placeholder allowed use to not only create custom loaders for our application, but also make them responsive.

As it sometimes takes a second or two for the API call to finish and to update the view, we have added toast notifications. These notifications will show up whenever the user adds, deleted, or updates something, and tells the user his action was successful.To enhance the user usability, the team added the extra 3 pages: OwnedProjects, joinedProjects and joinedRequests to help the user keep track of the projects which they are owning or joining and sending the pending requests.

We noticed that the student's profile pictures coming from the Fontys API were not always working, which caused errors in the frontend. To prevent these errors, we created a fallback that checks the Fontys picture after the student signs up and before the data gets send to the database. When the picture doesn't work, we call a third-party API UI Avatars to create a profile picture for the user with its initials. This picture will then be saved to the database instead of the Fontys picture. We needed to add additional logic to determine if the user has multiple names in his/her surname, as this would cause the API request to fail. If the user has multiple names, by default we take the 'last' name in the surname.

In the backend of the application, our main focus was on connecting with the front-end application, fixing any bugs that we got and improving our coding quality.

The endorsements feature was implemented. We had to change the database design to make it possible for the students to endorse the skills of other students. Furthermore, sending an email to the student when they want to change their passwords, register on the website and delete their account. In case a project is deleted by the project owner, the joined students also receive an email informing them about the deleted project.

Result

At the end of sprint 8, the team managed to finish our whole applications(system). The website was able to connect with the distributed service without any errors. Our code in all applications was also improved as best as we could at the time. Overall, we managed to finish all our must have features as well as some of the extra features and improvements we could implement for our application. We tested the features and ensured that all featured work correctly.

2.4. Finalization

Sprint 9

Task

- Finalize the documentation and application.
- Prepare the final presentation.

Process

Week 17 was the week of the presentation and also the final sprint. During this sprint we as a team went through documentation such as the project plan, design document and the C4 model and updated it according to the completed application. Besides, we also created this document to outline the process and the results of the entire project.

We also practiced and prepared for the final presentation/demo by creating a PowerPoint presentation explaining the key points of the project and practicing presenting among ourselves. Finally, we send the updated documents (Project Plan, Design Document and C4 Model), the Process Report and the final source code to our client.

Result

The sprint 9 marks the end of the project. In this sprint, we completed the whole application and the process report. Besides, we were done updating the documents: Project Plan, Design Document, C4 model. We prepared the final presentation for our project.

Chapter 3: Conclusions and Recommendations

Requirements	Must	Should	Could
A web application	x		
A web API application	x		
Database	x		
Search service index		x	
Log in/Register an account	x		
Create the project	x		
Delete the project	x		
Join the project	x		
Update the project	x		
Update the student account	x		
Delete the student account	x		
Reset password	x		
Search for projects, students		x	
Filter for projects, students		x	
A mobile application		x	
Administration overview		x	
Discussion network			x
Autocomplete/ Suggestion for projects		x	
Email verification		x	
Email notification for matching projects			x

Extra features
Skill matching
View a personal profile of the project owner
Accept, reject, create request to join a spot in a project
Send email notification when a project is deleted, student removed from project, request accepted or rejected
Send an email when the student signs up with the email and password
Send an email when the student deletes his account.
Send an email to the student with a reset code when he forgets the password.
Input validation
Console application for creating, uploading search index
Data consistent with Azure Search when creating or delete project
View the pending requests of the current user
View the joined projects of the current user
View the owned projects of the current user
Endorsement
Fallback profile picture with user's initials
Kick a member out of the spot

In conclusion, we have delivered a working web application including the following **7 components**: **the front-end application, back-end API application** connecting with **Fontys API, search service index, profile picture API (UI Avatars), email system** and **a MySQL database**. From the idea of creating a student networking platform for Fontys students, we developed successfully the ITEAM platform - a distributed system for students to create outside-school projects and recruit fellow Fontys students. The web application fulfilled all the required functionalities and offered a large number of advanced features to optimize the usability and deliver most efficiencies.

The web application offers the basic features namely signing in, registering an account, creating a project, updating a project, joining a project, updating a student account, deleting a student account and reset password. These features offer **advancement** and **optimization**. We allow the

user to reset password, change the profile description, add skill and delete skill when updating a student account. When a student joins a project, the project owner can view the student ‘s profile and either accept or refuse the join request. In the MoSCoW table, we highlighted the features that were accomplished. We decided not to implement the mobile application and discussion network because these required a large amount of time and our group lacks one member. We also decided not to develop the feature: search for a student or filter for a student to prevent a project owner from choosing the student he knows.

In addition to the features in the MoSCoW table, the application provides the user with advanced functionalities listed in the above **Extra features** table. These features are developed because it offers convenience and efficiency for the users.

Beside the web application, we have delivered the following documentations: Process Report, C4 model, Project Plan and Design Document.

To accomplish the project, the team follows the combination of Agile and Waterfall working methodology. The project was divided into 4 phases: Initialization, Research and Design, Implementation and Testing, Finalization. Each phase contains a number of sprints. During the implementation and testing, we developed and tested the features. By fixing the bugs and re-testing, we accomplish the features and proceed to the next stages. We believe that this methodology is the most efficient for a big-scale project.

The biggest challenge the team confronted was to be in lack of one member. For the first three weeks, our 6th member was working with us . However, due to the academic rule, he could not continue with the project. Handling the tasks of 6 members is quite challenging for us, especially the back-end team. The workload was extremely high, especially when we needed to connect the front-end with the api back-end. The team members put double effort into the project to acquire the best results.

In the future, it is ideal to develop the following ideal features such as cancelling endorsement or creating a working network where students can message, send files or update the files.

Chapter 4: Personal Reflection

Rahul Annadurai
Personally speaking, I found this project an interesting and useful one. This was mainly due to the fact that we had multiple people working on different parts of the project, so we needed a lot of planning and had to work with each other to put together the end product. Besides the time/planning aspect of the project, I also benefited from learning to use ReactJS. All in all, I found this project really useful for both learning new skills and an useful addition to my curriculum vitae.

Anh Le
Despite a software engineer student, I did not have much experience in the web development field because my focus was on desktop development. I was curious and doubted if my passion and strengths matched with web development. This curiosity and doubt urged me to choose the web application for the end project of my bachelor. Having experience in Laravel in Web3 course, I would like to try myself in working in ReactJS - the totally new framework to me. After 5 months of developing the software, I discovered my special interest in web development in general and front-end development in specific. My coding skill in the ReactJS is enhanced strongly and this prepares me for my upcoming graduation project. Working in a big team is not an easy task. However, we as a team managed to achieve the best result at the end. My teamwork skill is boosted thanks to this project. I have grown not only technically but also personally. I have learned an effective way of planning and executing the project. As a project leader, I have also sharpened my organizing skill and communication skill and approached the issues from both the overview and detailed perspective. I believed that the valuable skills I have achieved helped me to grow into a good software developer.

Barra Abd Al Fattah
I found the project very interesting and I managed to enhance my back-end skills as well as communication skills. Because we were working on two teams, front-end team and back-end team, we had to make a good plan in order to connect the front-end with back-end without having big issues. Therefore, my planning skills have improved.

Bao Quoc Nguyen
Overall, the project is quite interesting to me. I got an opportunity to apply my knowledge in distributed systems and learned more about Laravel framework. However, for me, the most interesting part of this project was our team communication and planning. We had a very good and well formed plan to carry out this project. Every step was planned carefully, and many possibilities for the application were considered. The main success of this project, from my perspective, is that we worked as a team and had good communication among team members. Every task of each member is made clear with Trello, the work division and weekly team meeting. Everyone in the team really worked diligently to achieve our common goals. Even though there were some minor parts that we did not have enough time to carry out, we managed to finish most of the parts that we planned to do.

Lauren van Loo

When they explained ProEP and what the project was supposed to be, it was, as business students, very unclear what our role should be within the projects. However, as I did not necessarily need to do IT & Business related tasks, I decided to take this opportunity to extend my knowledge of web development and focus more on the programming rather than the business aspects. Before this semester, I was not very familiar with front-end frameworks and back-end, which meant I had to learn while doing. Throughout this project and semester, I learned a lot about the front-end and back-end and which aspects/parts come to play when developing a distributed system. Besides, I have gained a lot of knowledge from learning ReactJS and immediately applying it to the project. All in all, I can say that I enjoyed working on this project and everyone was really devoted to the project and worked hard to develop a working end product.

Appendix I Work Division

Week	Task	Anh	Rahul	Barra	Quoc	Lauren
1 (04/09-11/09)	Research on the system context	2	2	2	2	2
	Come up with the component of system	4	4	4	4	4
	Team meeting	4	4	4	4	4
2 (11/09-18/09)	Decide on the system context	2	2	2	2	2
	Make two levels of C4	2	2	2	2	2
	Project Plan	3	3	3	3	3
	Team meeting	7	7	7	7	7
3 (18/09-25/09)	Use cases: Login, Delete account		2			
	Wireframes: Login, Project Details, Forgot Password		4			
	Use cases: Register, Update student account					3
	Wireframes: Sign up, Profile					4
	Use cases: Create Project, Delete Project	4				
	Wireframes: Landing page, Homepage, My Projects	3				
	Research about GDPR	1	1	1	1	1
	Update Project Plan	3	1	1	2	1
	Use cases: Join Project, Update Project			1		
	Use cases: Search, Filter				3	
	Class Diagram			3	3	
	Team meeting	4	4	4	4	4
4 (25/09-02/10)	Improve wireframes	2	2			2
	Improve class diagram, ERD			2	2	
	Decide on the extra component (C# app)	3	3	3	3	3
	Team meeting	4	4	4	4	4
5 (02/10-9/10)	URS (1st draft)	1				
	Improve class diagram, ERD			2	2	
	Work division	3				
	Research on React to React Native/PWA				2	2
	Team meeting	4	4	4	4	4
6 (9/10-16/10)	Make a demo on converting React to React Native	2	2	2	2	2
	learning react					2
	Team meeting	4	4	4	4	4
7 (16/10-23/10)	Create style guide					3
	learning React	3				3
	Create the basic layout of website with header, footer	4				
	Team meeting	4	4	4	4	4
8 (26/10-2/11) (Exam week)	Learning React	4	3			3
	Research on React libraries	2	8			
9 (2/11-9/11) (Exam week)	Code on the LandingPage and ProjectList in the HomePage	4				
	Learning React	4				2
	Code Login, reset password, forgot password, email sent pages		9			
10 (9/11-16/11)	Create the database with its relation on Laravel			6		
	learning React					2
	Improving wireframes, adding interaction, creating demo presentation					3
11 (16/11-23/11)	learn React					1
	Improve the database design			2		
	Team meeting + presentation	4	4	4	4	4
	Code on Filtering and Searching	8				
	Working on Project Url endpoints			6	8.5	
12 (23/11-30/11)	Making profile page public and private					6
	Code project members skills		5			
	Code Login, reset password, forgot password, email sent pages		9			
	Team meeting + mentore meeting	4	4	4	4	4
	Connecting to fontys API			3		
	Connecting login page to API		8			
	Implementing the connection to Fontys API			4		
	Making work devision table					1
	Refactor the code, separate the folder	3				

Week	Task	Anh	Rahul	Barra	Quoc	Lauren
13 (30/11-7/12)	Add 'Load more' function to 'Project Page'	3				
	Converting to functional components, adding the 'Create New Project' function	3.5				
	Code owners project details page		9			
	Implementing search service for project + import data				8.5	
	Meeting frontend	1	1			1
	Team Meeting	5	5	5	5	5
	Connecting to fontys API			3	2	
	Implementing the connection to Fontys API			4		
	Issuing tokens			4		
	Make the autentecation for API endpoints			4	8	
	Converting to functional components,making differrent types of skill					7
	add delete account + edit account functions					4
	Connecting reset Password to API		8			
	Make the forget password feature			3		
	internal meeting (3/12)	2	2	2	2	2
	delete skills					6
14 (7/12 - 14/12)	Implement the AddSlotSkillForm	5				
	Develop JoinedRequest	6				
	Implement CreateNewProject Form	2				
	Research React routing		5			
	Research sharing data between React pages		5			
	Skill matching + improve code quality + add endpoints for spot			10	15	
	Create new routes (profile, skills) for the logged in student.			8		
	Connect Private project page to API		12			
	Make the reset passowrd routes			8		
	endorsements - Backend			8		
	researching uploading picture			2		1
	Create new routes (profile, skills) for the logged in student.			8		1
	Code on SkillMatching	8				
15 (14/12 - 21/12)	Connect Public project page to API		14			
	Code on AddSpot, AddSkill	8				
	Make the reset passowrd routes	8		7		
	Connect CreateProject, CreateSpot, AddSkill with API	6				
	Connect API to front end + fix routes			8	15	
Christmas Holiday	endorsements - Backend			10		4
	Fix CreateProjectPage	4				
	Connect Homepage with API	6				
	setting up api locally					6
	Fixing bugs with public and privage project details page		16			
	Connect API to front end + fix routes				14	
	integrating api in frontend + fixing errors + lazy loading, description, skills					12
	Add loading to Homepage, OwnedProjects, JoinedProjects, JoinedRequests	10				
16 (4/1 - 11/1)	internal meeting	4	4	4	4	4
	documentation	2	2	2	2	2
	Fix bugs in LandingPage, add Validation to the Project Form	8				
	fixing errors + delete account + change password + skills add + skills delete			12	12	17
17 (11/1 - 18/1)	Connect API to front end + fix routes			3		
	documentation	2	2	2	2	2
	internal meeting	4	4	4	4	4
	Refactor and fix bugs in Create new Project, My Owned Projecting Page					
20 (18/1 - 22/1)	Fixing bugs with public and privage project details page, login and reset page		15			
	internal meeting	4	4	4	4	4
	documentation	2	2	2	2	2
	fixing issues + endorsements + skills					7
	integrating api					8
20 (25/1 - 29/1)	Connect API to front end + fix routes + fix import project to azure service					
	integrating api + fixing erros + sign up + redirect delete + default photo				8	
	Connect API to front end + fix routes			4	15	
	Add constraint, test functionalities, fix bugs, refactor	8				
	profile public + projects overview profile + endorsements + skills					14
	Team meeting	12	12	12	12	12
	Connect API to front end + fix routes + update document				15	
	finish endorsements, profile public/private + create skills data + responsive loaders					10
	documentation, presentation, demo video, preparing for demo and presentation	10	10	10	10	10

Member	Total Hours Worked	Mininum Required Hours
Rahul	241	196
Barra	241	196
Quoc	240	196
Lauren	241	196
Anh	239.5	196