CTU

CZECH TECHNICAL UNIVERSITY IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Computer Science

Bachelor's Thesis

Team Collab

Enhancing the Creation and Collaboration of Team Projects

Hà Trang Phanová
Software Engineering and Technology

May 2023
Supervisor: Ing. Petr Aubrecht, Ph.D

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Phanová** Jméno: **Há Trang** Osobní číslo: **492219**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Team Collab

Název bakalářské práce anglicky:

Team Collab

Pokyny pro vypracování:

Semestrální práce se často dělají ve dvojicích, případně větších týmech. Nicméně na technických školách bývá problém domluva mezi studenty. Navrhovaný systém by měl poskytnout studentům prostředí, kde by mohli rychle a snadno najít kolegy do týmu. Systém by měl umožňovat i skládání heterogenních týmů, kdy je možné specifikovat, kolik lidí je třeba a jaké mají mít schopnosti (znalost nějaké domény, programovacího jazyka, technologie, množství volného času).
Každý projekt mi svůj vlastní popis a specifikace (čeho se týká - např. předmět, v čem je implementován ap.) a podle těchto kritérií by měl být vyhledatelný.
Do budoucna by takový systém mohl sloužit např. opensourcovým projektům při hledání dobrovolníků a na druhou stranu začátečníkům v programování, kde si mohou vyzkoušet technologie podle svého zájmu.
Požadavky:
1. Zanalyzujte a popište aktuální stav software pro organizaci týmů.
2. Implementujte webovou aplikaci podle zadání. Myslete na User Experience jak z pohledu zájemce o spolupráci, tak z pohledu správce týmu. Součástí by měla být i nějaká statistika ukazující, v jakém stavu jsou jednotlivé oblasti, např. kolik týmů hledá partnera pro předmět EAR.
3. Implementaci vyzkoušejte z uživatelského hlediska - projděte kompletní lifecycle projektu – od registrace uživatelů, založení projektu, domluvě, až po finální ukončení projektu. Unit testy zkuste aspoň pro lifecycle projektu. Otestujte REST rozhraní.
4. Řešení implementujte v technologii JakartaEE. Technologie na uživatelské rozhraní je na vůli studenta, musí ji odůvodnit.

Seznam doporučené literatury:

[1] The Jakarta EE 8 Tutorial: https://eclipse-ee4j.github.io/jakartaee-tutorial/
[2] Jakarta EE Cookbook - Second Edition, https://www.packtpub.com/programming/jakarta-ee-cookbook-second-edition
[3] Patterns of Enterprise Application Architecture — Martin Fowler

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Petr Aubrecht, Ph.D. katedra počítačů FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **10.02.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Petr Aubrecht, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

Acknowledgement

I would like to thank my supervisor, Ing. Petr Aubrecht, Ph.D., for his guidance, valuable advice, and expertise. Without his encouragement, this project would have probably never come to life.

My sincerest thanks also belong to my family and friends, who have given me great support throughout the whole journey of writing this thesis. I would not be here without them.

Last but not least, I would like to thank my former teammates and classmates for their help in making this project what it is today.

Declaration

Abstrakt / Abstract

Bakalářská práce se zabývá návrhem a implementací webové aplikace, která poskytne studentům prostředí, kde mohou efektivně najít spolupracovníky do skupinových projektů. Zároveň může sloužit i jako portál pro výběr studentů do školních klubů, akcí a mimoškolních aktivit.

Uživatel si může založit projekt, na kterém by chtěl pracovat a pro který potřebuje najít spolupracovníky do týmu. Každý projekt bude mít svůj popis a specifikace jaké a kolik spolupracovníků zakladatel projektu hledá. Zároveň bude možnost k projektu přidat klíčová slova týkající se hledaných pozic do týmu, které usnadní vyhledávání. Ostatní uživatelé mohou do dostupných projektů zasílat žádosti o přijetí do týmu.

Klíčová slova: týmová spolupráce, vyhledávač projektů, Jakarta EE, Web Components, webová aplikace, Team Collab.

Překlad titulu: Team Collab (Zlepšení tvorby týmových projektů a její spolupráce)

The aim of the bachelor thesis is to design and implement a web application that will provide students with an environment where they can efficiently find collaborators for group projects. It can also serve as a portal for selecting students for school clubs, events, and extracurricular activities.

The user can establish a project that they would like to work on and need to find collaborators for their team. Each project will have a description and specifications of which and how many collaborators the project founder is seeking. There is also an option to add the keywords of positions the team is seeking to make the search easier. Other users can submit applications to the projects to join the team.

Keywords: team collaboration, projects finder, Jakarta EE, Web Components, web application, Team Collab.

/ Contents

Tables / Figures

Chapter 1
Introduction

1.1 Motivation

As part of the academic life of students, there are a variety of collaborative projects to join; like group projects, club events, and large-scale real-time projects, allowing students to experience and acquire knowledge during their studies. Although these opportunities bring plentiful benefits, they are seldom taken advantage of, as few people are aware of them, and no platform provides students with a way to locate these group activities.

This thesis aims to create a web application that provides students with a place to find team members for group work specified in their course work. Having this type of setting would bolster extracurricular activities, such as student organisations, charity work, or even internships. By utilising the application, students can form project teams and meet new people from different backgrounds, potentially leading to valuable, long-term contacts in their field.

1.2 Application Description

In the application Team Collab, the student creates a project that they want to work on and therefore must find team collaborators. Each project will have its description and specifications about which and how many collaborators the project creator is seeking. Simultaneously, the project is assigned to the category during its creation, enabling others to find the project they want to work on more quickly.

Chapter 2
State of the Art

Collaborative group work is a method of learning in which students work together to achieve a common goal. Such team projects encourage group discussions, which helps students learn and analyse the materials more efficiently [1]. Students get to bond with their peers during group activities. Based on National Survey of Student Engagement positive group experiences appear to enhance student learning, retention, and overall university success [2], [3].

Although group projects bring various benefits, it is demanding to work on one. It often becomes uncertain who handles what unless group roles are determined, as well as lack of ongoing discussion on progress can put the team workflow in a stalemate [1]. An innumerate amount of students see themselves when they end up doing most of the work instead of relying on their other teammates.

There are various accessible tools that can assist students in mitigating challenges related to poor collaboration within team projects. For example, word editors that facilitate simultaneous editing offer a solution; however, there is presently no specialised software specifically designed for team formation and creation.

The responsibility of forming teams falls upon individual instructors or students themselves. However, students are frequently unfamiliar with their classmates, which causes resorting to a combination of online discussions, social media platforms, or relying on their existing circle of friends to assemble a team. Finding a suitable group takes several days to finalise. While some students actively seek potential teammates, others leave it to chance, resulting in a hit-or-miss outcome.

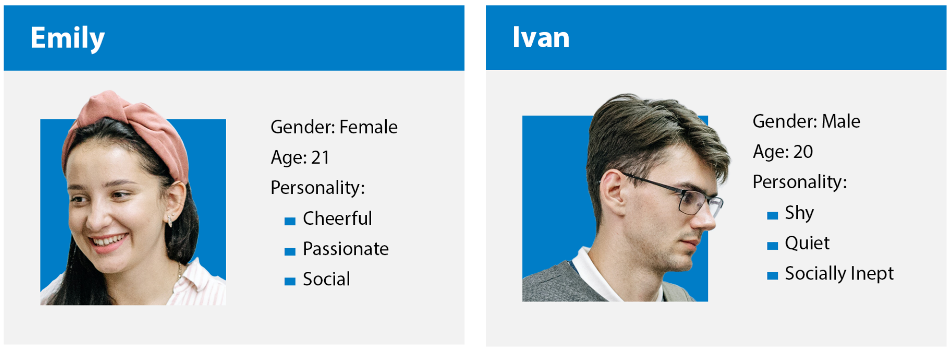
Some would suggest that the students who did not take the initiative reap what they sow, hence they have to handle its consequences, but this problem involves not only one person but the whole team.

2.0.1 Target Personas

A target persona is a detailed profile of somebody who represents the target market. This persona is fictitious and will be henceforth used as a tool to help acquire a better understanding of the struggling students during teammate seeking period.

To start with, there are two types of students as shown in Figure 2.1–**Emily** is a well-spoken individual that prefers taking the reign, and **Ivan** is a silent type that takes the passive role and goes with the flow. Both of them are students in their second year at Czech Technical University and the instructor in one of their classes has just announced that they will work the entire semester in a group of five. They must submit their team members next week at the latest.

Emily had an unpleasant experience with group projects last year. She did not work well with her previous teammates, as they were all struggling with the assignment. There was also an issue with the teammate who, unlike her, did everything at the last second. But besides her former teammates, she does not know anybody else in the class. Emily is not optimistic about working with the same people, so she prefers to assemble


Emily

Gender: Female
Age: 21
Personality:
- Cheerful
- Passionate
- Social

Ivan

Gender: Male
Age: 20
Personality:
- Shy
- Quiet
- Socially Inept

Figure 2.1. Persona cards of Emily and Ivan.

a new team of people this time, but everybody has made a circle of friends/acquaintances that are difficult to break into. She also does not want to offend her previous teammates by rejecting their offer of teaming up together once again. Therefore, Emily instead resorts to contacting two people through a Facebook messaging app. The first classmate replied that their team was already full, and the latter was still hesitating on what to do about this group project, but will work with Emily and help her find more members.

Ivan is socially awkward and does not have the courage to come up and ask his classmates in person. He plans to just see who else does not have a group like him and join them instead, and in worst case scenario, ask his instructor for help.

As no system would help students like Emily and Ivan form a new team for group projects, they will have to brace themselves and speak up to their other classmates, ask to join the people they already know, or just leave everything to the chance.

2.0.2 Alternative Methods and Possible Competition

Based on Emily's situation, one of her choices was to face her old teammates, reject their offer, and then ask other people in the class. This is a strategy where she would talk with people face-to-face and use her social skills to resolve the problem.

If talking in person was not an option, she would then try doing it online by finding other people's contacts and handling the situation through social media platforms and messaging applications available at her disposal. Given that their major or class had a social group on social media, both of them could have made a post visible to all its members proposing a team to join or teammates to join them.

With the stated solutions in mind, the web application Team Collab is considered a niche market, as there is no direct competition. Alternative methods can replace it and social media and messaging apps can be considered its only indirect competition.

Chapter 3
State To Be

The project aims to develop a web application for creating and managing school team projects. The aspiration is to simplify and accelerate the search for team members.

Students can easily create a new team, and add a category, description, and wanted positions. Others can then quickly find a team by name or category and request to be added to the team.

Team Collab web application aspires to provide students with an environment where they can quickly and easily find collaborators in group projects (school, extracurricular, etc.) and other group activities during their academic life. Therefore, it is also suitable for school organisations to make use of the application to recruit members in respective clubs and classes.

3.1 User Roles

The system user roles are as follows:

- Guest user
- Common user
- Project member
- Project owner
- Administrator

Guest user is a non-logged-in visitor who can only register a new account or log in with an existing account.

Common user is an authenticated user, therefore, has access to web application and can browse through lists of projects.

Project member is an authenticated user belonging to a project and can view its members.

Project owner of the project can handle and change its applicants.

Administrator has full access to the functionality of the application and subsequently has all the rights of the other users.

3.2 Requirements

In the web application, the common user creates a project that they would like to work on and needs to find collaborators in the team. Each project will have its description and specifications of which and how many collaborators the project owner is seeking. The project is assigned to the appropriate category during its creation, which will allow others to find the project they want to work on faster. The common user is now the project owner of the project they just created.

If any other common user would like to join in the previously mentioned project, it is required for them to request permission from the project owner first. The common user will have to apply by sending their join request to the project and the project owner can either accept or reject it. The common user can only become an official project member when the project owner approves their request to join.

3.2.1 Functional Requirement

FR01 — The system must allow unregistered visitors to create a new account.

FR02 — The system must allow the user to authenticate with their created account.

FR03 — The system must allow the user to view the list of all projects.

FR04 — The system provides the user tools to search and filter the list of all projects.

FR05 — The system must allow the user to send a join request to the projects.

FR06 — The system must allow the user to view all the projects they are in.

FR07 — The system must allow the user to view all join requests sent by the user.

FR08 — The system must allow the user to view all projects owned by the user.

FR09 — The system must allow the project owner to view all the join requests belonging to the project.

FR10 — The system must allow the project owner to either accept or reject a project join request.

FR11 — The system must allow the project owner to modify the project.

FR12 — The system must allow the project member to view other members in the project.

3.2.2 Non-Functional Requirement

NFR01 — The GUI of the web application must be user-friendly.

NFR02 — The web application must be optimized for the latest version of most used web browsers like Google Chrome, Safari, Microsoft Edge and Mozilla Firefox.

NFR03 — The server must not return a restricted web page to a user who is not authorized to access it.

NFR04 — The web application should perform accordingly well even with a base count of around 30 users.

3.3 Use Cases

Each user role mentioned in section 3.1 serves as an actor in the use-case diagram illustrated in Figure 3.1 which also contains use-cases derived from functional requirements in section 3.2.1

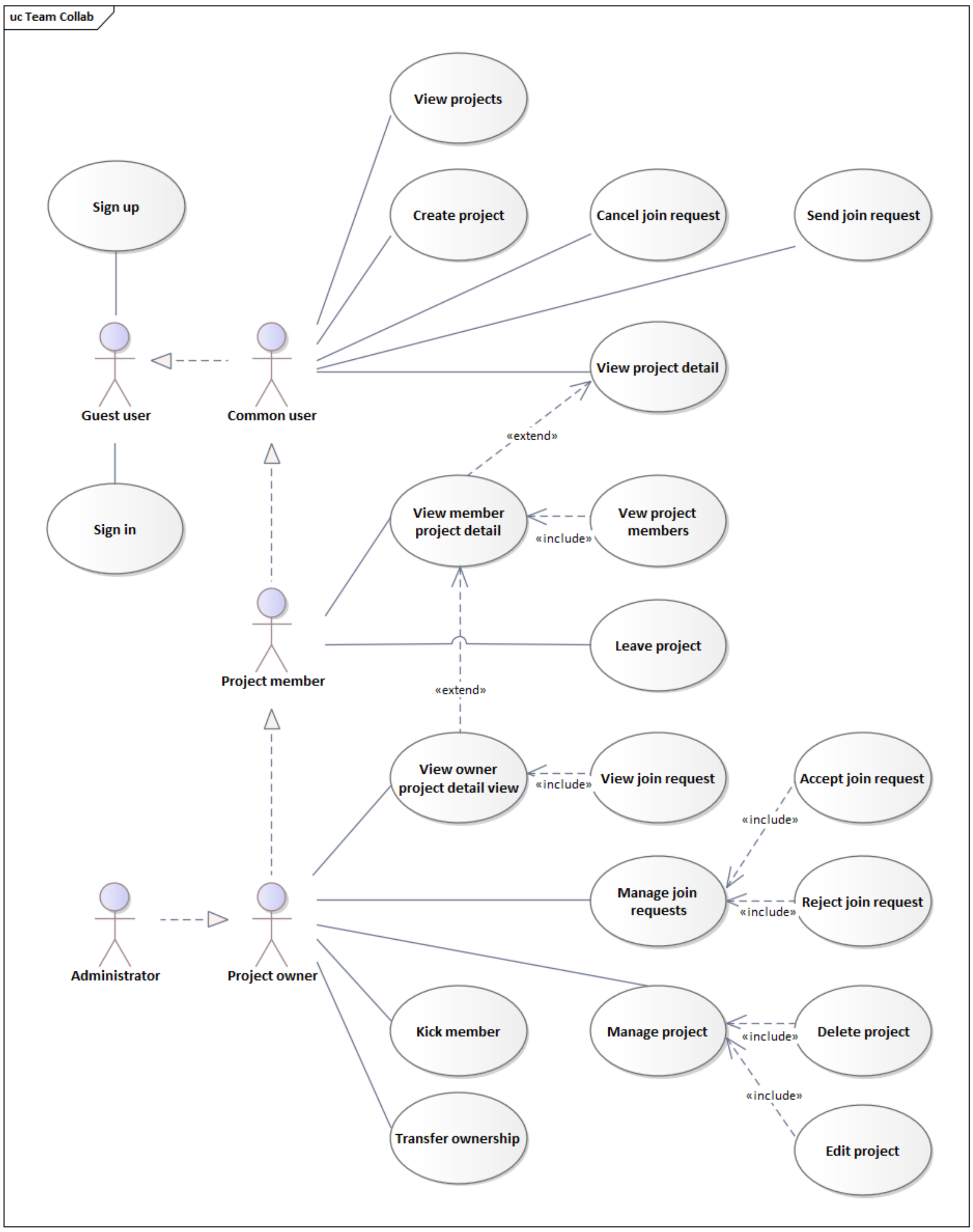


Figure 3.1. Use cases diagram of the web application.

- **UC01 — Sign in**

  After the guest user presses the sign-in button from the landing page, the guest user will be redirected to a sign-in screen, where they can access the web application after authenticating themselves.

- **UC02 — Sign up**

  If the guest user does not have an account they can create a new account by pressing a sign-up button located on the landing page, or by pressing the same button but on the sign-in screen, both of these buttons will take the guest user to the sign-up screen. After successfully creating a new account they will be redirected to the sign-in page.

- **UC03 — Create project**

  The common user can create a new project page by tapping on the button New project on the user's My projects. The common user will be redirected to a page where they have to fill in the required information and it then creates the project detail page.

- **UC04 — View projects**

  The common user can view all the projects in the main Home page of the web application. Each project item in the list will have its category icon, title, the current and maximum number of members in the project, tags of positions they are currently seeking and a short preview of the project description.

- **UC05 — Sort projects**

  The common user can sort the list of projects on the main Home page by newest/oldest addition, lowest/highest number of current members, and lowest/highest maximum limit of members able to join the project.

- **UC06 — Filter projects**

  The common user can filter out the list of projects based on their category, which school subjects they belong to or what positions are they currently seeking.

- **UC07 — View project detail**

  By pressing on one of the projects from the list on the main page, the common user will be viewing the chosen project in greater detail. The Project detail page displays the name of the project owner, the date when it was created, the full project description and a list of its wanted positions.

- **UC08 — Send join request**

  The join request button is located on each page of the project detail and the common user can request to join the project by filling out the join request form.

- **UC09 — Cancel join request**

  The common user can cancel their join request at any time on the Project detail page by pressing the cancel request button.

- **UC10 — View member project detail**

  The project members can view all the projects they belong to as a member.

- **UC11 — View project members**

  The project members can view the list of members the project has on its Project detail page.

- **UC12 — Leave project**

  The project members can leave by pressing the leave project button on the Project detail page.

- **UC13 — View owner project detail**

  The project owners can view all the projects they belong to as an owner.

- **UC14 — View join requests**

  The project owner can view the project's list of incoming requests to join on the Project detail page of the project they are the owner of.

- **UC15 — Manage join requests**

  The project owner can handle all incoming requests to join on the Project detail page of the project they are the owner of.

- **UC16 — Accept join request**

  The project owner can accept a request to join the project and the applicant will become a project member by doing so.

- **UC17 — Reject join request**

  The project owner can decline an applicant's request to join the project. The applicant cannot apply again unless the project owner removes them from the list of rejected requests.

- **UC18 — Manage project**

  The project owners can manage all the projects they belong to as an owner.

- **UC19 — Delete project**

  The project owner can delete the entire project by pressing the delete project button on the Project detail page.

- **UC20 — Edit project**

  The project owner can edit the project information by pressing the edit project button on the Project detail page.

- **UC21 — Kick member**

  The project owner can remove a member from the project.

- **UC22 — Transfer ownership**

  The project owner can transfer their position as the project owner to any of the project members.

The now defined functionality requirements and use cases, gives a clearer understanding of what can Team Collab provide to its users.

3.4 Web Application Design

Personas, sitemaps, and prototypes, among other tools, facilitate the effective documentation and communication of concepts. In the application designing process, this set of deliverables serves as an instrumental tool to document the design choices made [4].

Previous chapter 2.0.1 modelled the target audience using personas and scenarios. Based on the observation of the potential user base, it has been concluded that the main target will be young adults with an age ranging from 18 to 26 who are still involved in the school environment.

The thoroughly done analysis will now prove to be useful and in shaping the design of the web application's UI/UX elements.

3.4.1 UX Design

A comprehensive designing process includes working on its application prototyping. It is essential to determine which kind of screens are needed to provide a visual representation of its purpose. Which for instance can be achieved with the help of user stories. Defining such key screens is a first step whereas filling them with UI components is what comes after.

Deriving from section 3.3 the foundational user stories could be as follows:

- **TASK-1** — Sign Up and First Time Experience of Creating a Project
- **TASK-2** — Send Join Request and Join Project
- **TASK-3** — Accept an Applicant

To progress further within the web application, user login is a prerequisite based on the previously discussed user roles in chapter 3.1. As a result, a landing page becomes essential to provide incoming visitors with pertinent information, including instructions about the web application's functionality, and to prompt them to either sign in or register a new account. Once authenticated, users gain access to the main home page, which presents a comprehensive list of all projects.

- **TASK-1** — Sign Up and First Time Experience of Creating a Project

  Sign Up: User signs up, setting email and password.

  First Project: User's task is to make their first project right after signing up.

Given that the procedure went through a happy path[1], resulting key screens that will be needed in accomplishing this task are outlined below.

Key Screens: Unregistered starting page, Sign Up Screens, Sign Up Confirmation, Sign In Screen, Create Project

The remaining two tasks include more complex states and require engagement of more than just one user. Figure 3.2 depicts how the state of a request to join a project can change and its transitions.

[1] *"...sequences of activities that will be executed if everything goes as expected without exceptions"* [5]

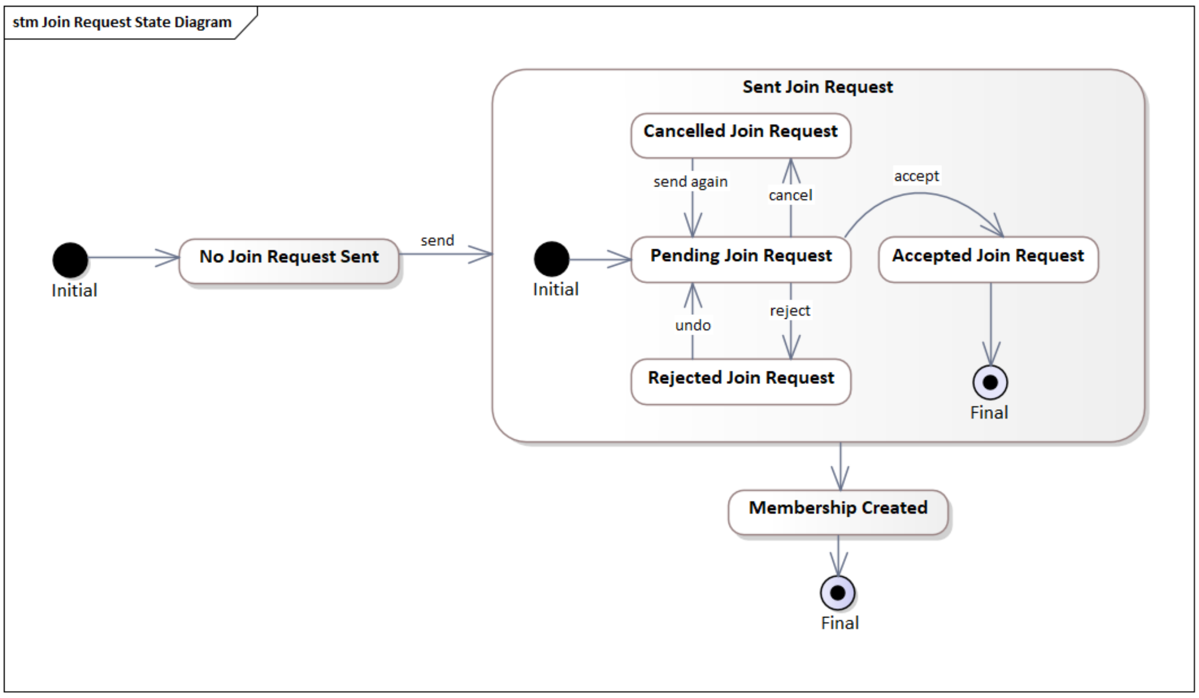


Figure 3.2. State diagram of a Join request.

In consideration of join request states, TASK-2 and TASK-3 can be described in the following manner (for text clarity, the task of signing in has been omitted):

- **TASK-2** — Send Join Request and Join Project

  Send Join Request: User's task is to find a project and send in a join request.

  Join Project: User's task is to wait to be accepted in the project by the project's creator.

  Key Screens: Projects Screen, Project Detail Screen, Join Request Screen, Join Request Confirmation, Join Request Status

- **TASK-3** — Accept an Applicant

  Accept Join Request: User's task is to open one of projects they created and accept a pending join request.

  Key Screens: My Projects Screen, Project Detail Screen, Join Requests Screen, Accept Join Request Screen

The listed items are only a selection from a broader range of tasks that could be done in the application, but there is no need to go through every single one of them. They merely serve as a guide in the thought process of modelling the screens.

"A sitemap is a visual representation of a site's structure. Usually arranged hierarchically, sitemaps indicate how content and information are organized and, consequently, how users will navigate the system. A sitemap documents the system as a whole, pulling back from interface specifics to look from a broader vantage point." [4]

The three tasks' analysis now creates a foundation of key screens that can be assembled together in a sitemap 3.3. Additionally, it has been enriched with more screens on account of the characteristic of a web application.

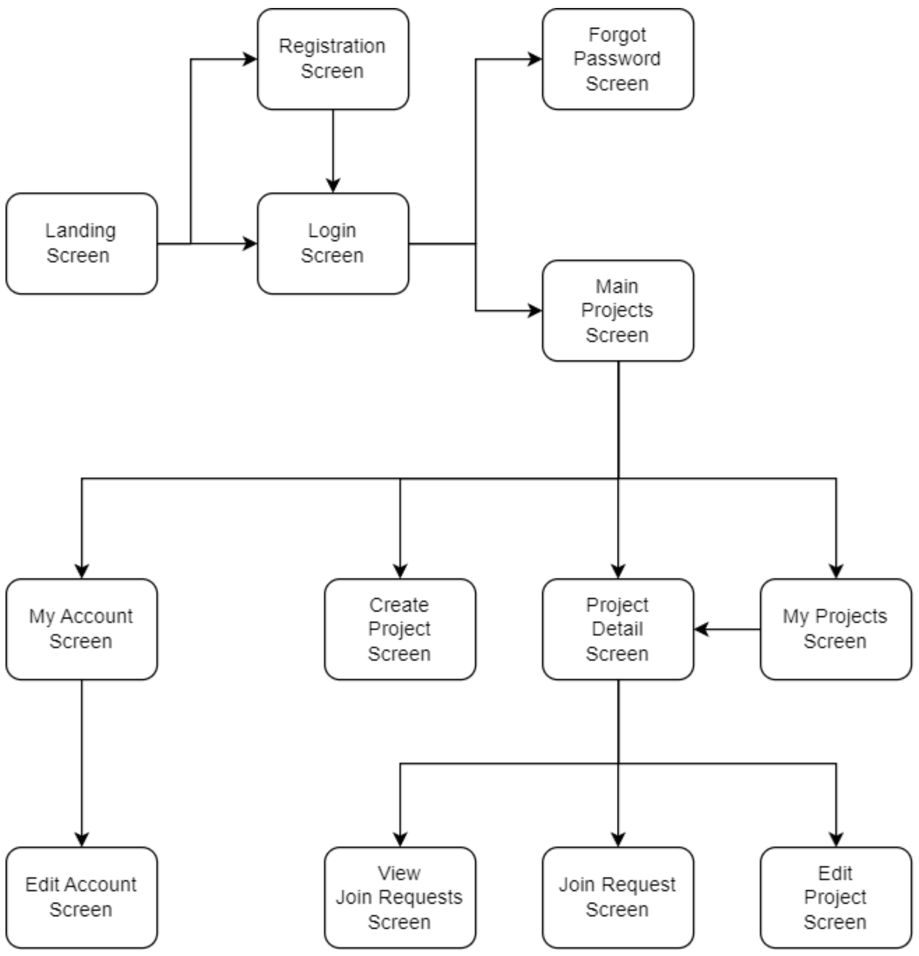


Figure 3.3. Sitemap navigation between screens.

3.4.2 UI Design

"True beauty is the combination of the physical form and desired function operating together in harmony. In digital design, it is not enough that each pixel is perfect but it also must add usefulness, understanding, or delight and often a combination of all three." [6]

The visual appeal of a website plays a crucial role in attracting users and determining their engagement. It significantly influences whether a user chooses to stay or leave. Furthermore, a well-structured website facilitates easier navigation for users. Visitors have specific expectations regarding the placement of various elements on a site, and ensuring proper placement enables efficient searching and faster information retrieval.

Considering that students comprise the primary target audience, the design of the application will prioritise a minimalist look and efficient features. Students are tech-savvy, therefore emphasis will be placed on efficacy rather than displaying detailed explanations for every feature on the screen.

Creating a visually captivating interface that aligns with their age will enhance engagement among student users. Figure 3.4 offers the creation of the icons and logo, in addition to choosing the colour palette.

The icons have been created in Adobe Illustrator as a vector graphics which allows the possibility of exporting them in SVG (Scalable Vector Graphics). This format exhibit practicality in keeping the UI responsive.

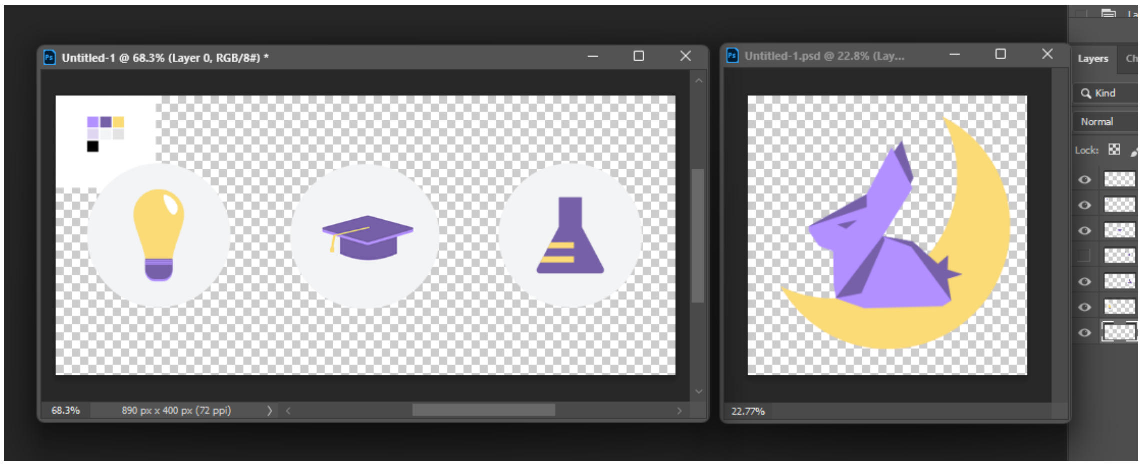
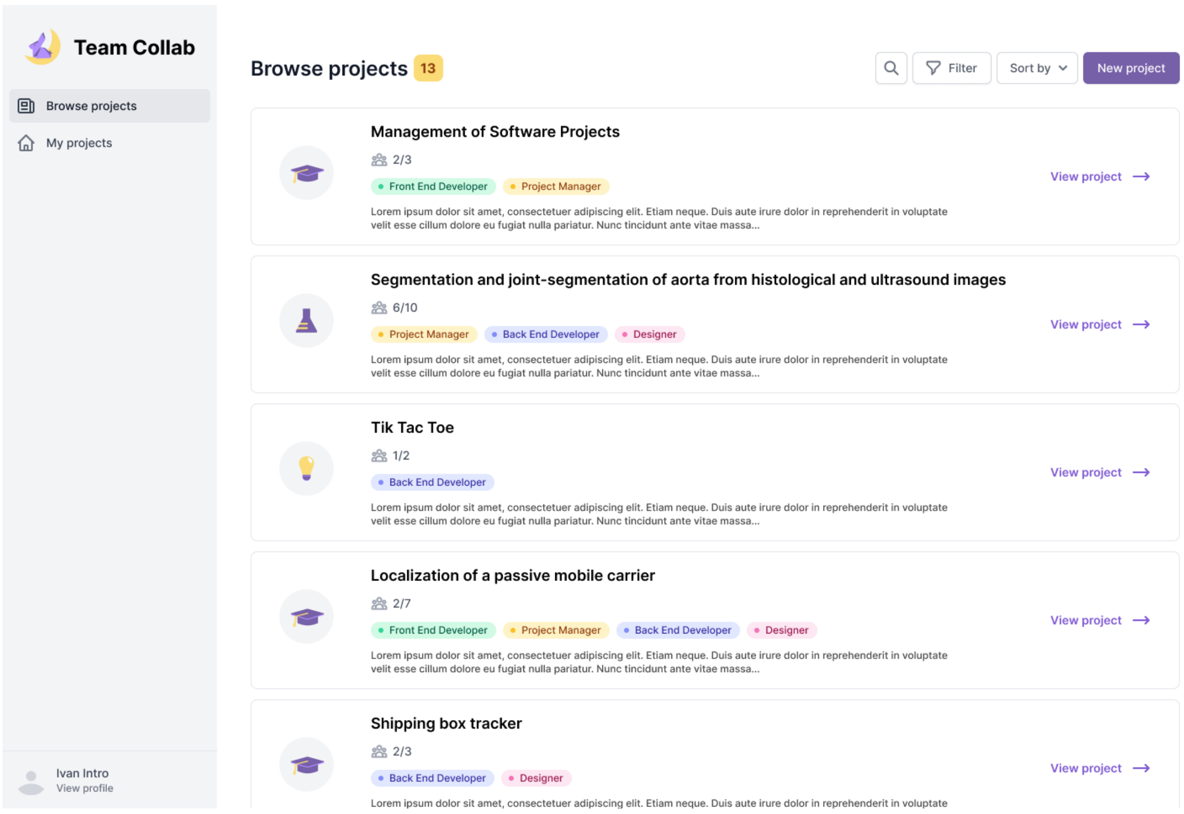

Untitled-1 @ 68.3% (Layer 0, RGB/8#) *
Untitled-1.psd @ 22.8% (Lay...
Layers
Kind
Normal
Lock:
68.3%
890 px x 400 px (72 ppi)
22.77%

Figure 3.4. Displaying logo and icons in Adobe Photoshop 2023.

After a detailed analysis of the UX/UI design, the next step is to translate the concept and ideas into a prototype. To make the implementation easier, a high-fidelity prototype will serve as its template to follow. A suitable tool for such a task is Figma.
Given the present design of the main projects screen 3.5 and project detail screen 3.6, excessive deliberation over style choices during the implementation may be unnecessary or minimal.
Despite the detailed nature of the graphic design, several elements were conveniently accessible through the selection offered in Figma, allowing for a more efficient workflow and the possibility of future alterations.



Management of Software Projects
2/3
Front End Developer · Project Manager
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam neque. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Nunc tincidunt ante vitae massa...

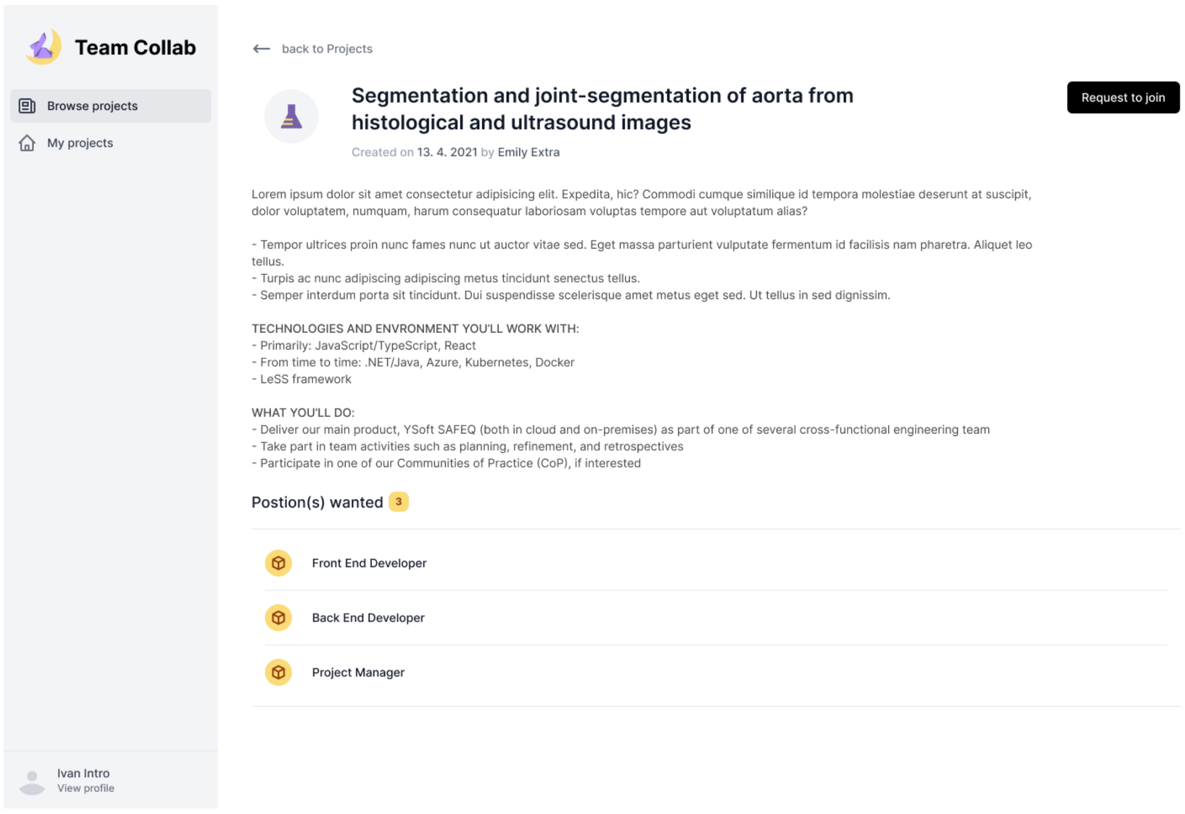
Segmentation and joint-segmentation of aorta from histological and ultrasound images
6/10
Project Manager · Back End Developer · Designer
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam neque. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Nunc tincidunt ante vitae massa...

Tik Tac Toe
1/2
Back End Developer
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam neque. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Nunc tincidunt ante vitae massa...

Localization of a passive mobile carrier
2/7
Front End Developer · Project Manager · Back End Developer · Designer
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam neque. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Nunc tincidunt ante vitae massa...

Shipping box tracker
2/3
Back End Developer · Designer
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam neque. Duis aute irure dolor in reprehenderit in voluptate

Figure 3.5. High-fidelity prototype of the project browsing page.



Segmentation and joint-segmentation of aorta from histological and ultrasound images

Created on 13. 4. 2021 by Emily Extra

Lorem ipsum dolor sit amet consectetur adipisicing elit. Expedita, hic? Commodi cumque similique id tempora molestiae deserunt at suscipit, dolor voluptatem, numquam, harum consequatur laboriosam voluptas tempore aut voluptatum alias?

- Tempor ultrices proin nunc fames nunc ut auctor vitae sed. Eget massa parturient vulputate fermentum id facilisis nam pharetra. Aliquet leo tellus.
- Turpis ac nunc adipiscing adipiscing metus tincidunt senectus tellus.
- Semper interdum porta sit tincidunt. Dui suspendisse scelerisque amet metus eget sed. Ut tellus in sed dignissim.

TECHNOLOGIES AND ENVRONMENT YOU'LL WORK WITH:
- Primarily: JavaScript/TypeScript, React
- From time to time: .NET/Java, Azure, Kubernetes, Docker
- LeSS framework

WHAT YOU'LL DO:
- Deliver our main product, YSoft SAFEQ (both in cloud and on-premises) as part of one of several cross-functional engineering team
- Take part in team activities such as planning, refinement, and retrospectives
- Participate in one of our Communities of Practice (CoP), if interested

Postion(s) wanted 3

- Front End Developer
- Back End Developer
- Project Manager

Figure 3.6. High-fidelity prototype of the project details.

Chapter 4
Architecture

With the usage of a client-side three-tier architecture, the application is divided into following parts:

- Presentation tier – web application
- Application tier – application server
- Data tier – database

4.1 Web Application

4.1.1 Front-end Technologies

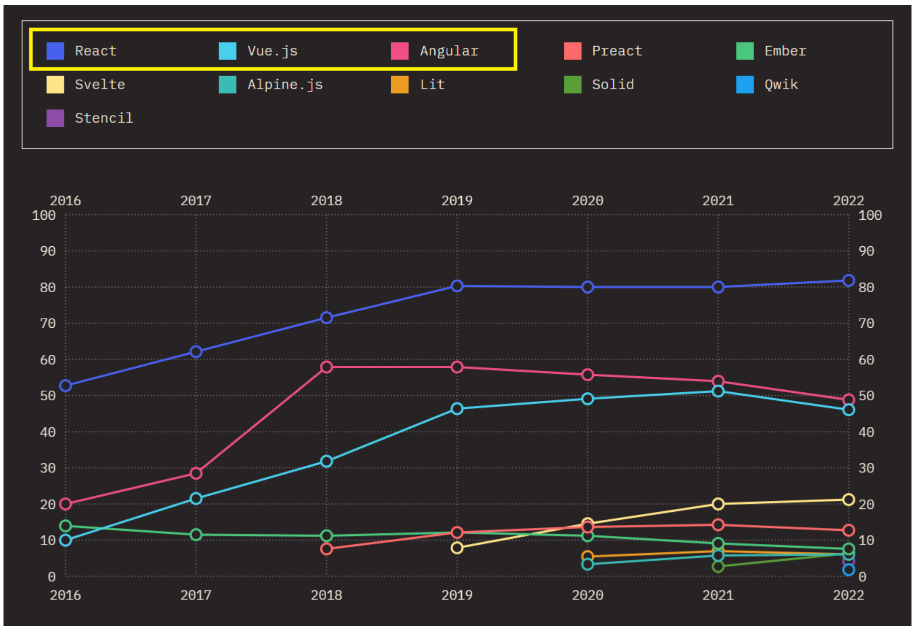


Figure 4.1. Frontend frameworks usage data from State of JavaScript [based on ratio: (would use again + would not use again) / total] [7].

According to data 4.1 from the annual survey on State of JavaScript 2022 [1] the top three most used front-end frameworks are React, Vue.js and Angluar.

React is currently the most popular JavaScript library and with its large-scale user community there is not a lack of online resources to study. When it comes to its learning curve, it is relatively manageable for individuals who possess a foundational understanding of HTML and JavaScript [8].

[1] `https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/`

When examining the second contender, **VueJS**, it presents a comparatively shorter learning period in comparison to React, yet it fails to match React's significantly more robust ecosystem. Unlike React and Angular, VueJS lacks the endorsement of prominent corporations such as Facebook and Google. It relies on community-driven support, which can be regarded as both advantageous and disadvantageous [8].

Angluar, conversely, demands a more comprehensive understanding of TypeScript and is the hardest to learn. As previously stated, its association with Google ensures the longevity of its applications through ongoing maintenance [8].

4.1.2 Overall Thoughts on Frameworks and Libraries

Over the years, there have been significant advances in the idea of autonomously extending the capabilities of HTML in order to create advanced UI elements. This has resulted in the introduction of nowadays highly popular libraries such as jQuery UI, React, Angular, Vue, and others. All of these libraries provide methods for creating UI elements that adhere to the component model. Each of these libraries introduces its approach to creating UI components, as well as attempts with mapping UI elements to markup in its own fashion [9].

It is undeniable that it makes the development of client-side application much easier. Many companies have moved from merely using JavaScript and jQuery to developing applications in such frameworks. Because of that, the skills of programming in frameworks like React, VueJS and Angluar are highly sought for.

The author themselves had learnt programming in React and VueJS and this project has already been attempted and trialed in 2021 with React and bootstrapped with Create React App[2].

The project implemented Tailwind CSS version 2.0 as its CSS framework in 2021. However, a subsequent release of version 3.0 in December of the same year brought unforeseen modifications that caught the author off guard. The extent of these changes proved to be more significant than expected, resulting in a disruption of the application's functionality. Upgrading the application to version 3.0 will require a substantial investment of time and effort.

This situation stands as evidence of the demanding nature of keeping up with and maintaining up-to-date technologies when relying on libraries and frameworks. Consequently, it has brought the author's attention to a technology known as web components.

4.1.3 Web Components

Web components possess cross-browser compatibility since they are built using HTML, CSS, and JavaScript, rendering them platform-independent [10]. Creation of custom elements by merely extending existing HTML elements without the need for any JavaScript library or framework [9]. Moreover, web components demonstrate significant characteristics, such as the ability to be reused, maintained, and encapsulated [10].

[2] `https://github.com/facebook/create-react-app`

Web component development consists of three main building blocks illustrated in Figure 4.2 that enables it to be reusable across platforms. Some sources would also mention about the HTML Imports, but they are deprecated as they are not standard [11]. The HTML element possesses the characteristic of being able to contain HTML snippets without the need for immediate rendering.

Custom elements refer to a collection of JavaScript APIs for creating custom DOM elements associated with specific HTML tags. This shadow DOM is a set JavaScript APIs that enables the encapsulation of the web components. Encapsulated object is protected from end users interacting with it in ways it was not intended [11]. When the component is appended to the DOM, its shadow DOM becomes connected to the global DOM while maintaining its privacy [9].

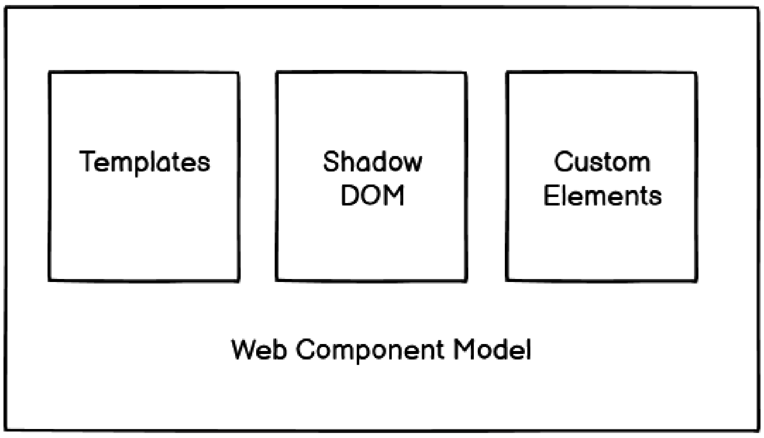


Figure 4.2. Web component model [10]. (Re-created in Balsamiq Wireframes)

4.2 Application Server

4.2.1 Jakarta EE 8

Jakarta EE 8 (previously also known as The Java Platform, Enterprise Edition, Java EE) is a collection of API specifications designed to work together when developing server-side enterprise Java applications. There are numerous Jakarta EE implementations and since code created under the Java EE specification may be deployed to any Java EE-compliant application server with little to no modifications prevents vendor lock-in [12].

4.2.2 Payara Server

Application servers, such as JBoss, Websphere, Weblogic, and GlassFish, are where Java EE applications are commonly deployed. Every application server is viewed as a different Java EE implementation. The author has decided to integrate an existing implementation, which is the open-source application server Payara 5 that supports Jakarta EE 8, instead of creating one from the scratch [12].

4.3 Database

The open-source relational database PostgreSQL is used as a primary database for many web applications. It is proved to be the most reliable option as the author has the most experience with it compared to other technologies. Jakarta EE also provides a robust integration with SQL via JPA.

Chapter 5
Implementation

The application's primary implementation details are covered in this chapter. Starting with the implementation and functionality of the back-end side, then proceeding to the realisation of the front-end.

5.1 Data Model

The data model of the web application is depicted in the Figure 5.1.

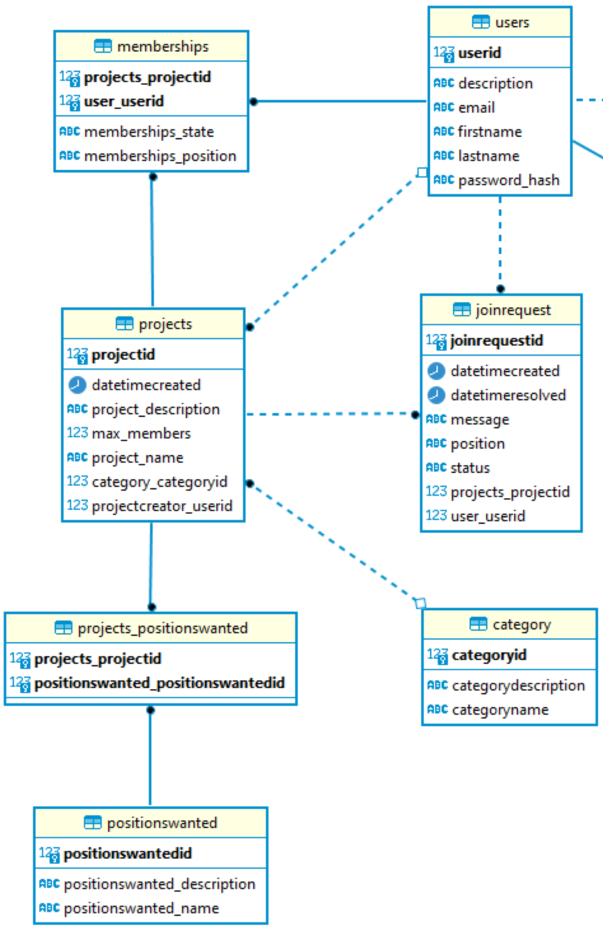


Figure 5.1. ER Diagram of the web application.

5.2 Security

Security of the web application is primarily handled by the Jakarta EE 8 Security API. *"It includes an HTTP authentication mechanism and an identity store abstraction for validating user credentials and group memberships, and also provides a security-context API to programmatically handle security."* [13]

One or more users, roles, and permissions are managed by a realm. A user logs in and is a member of a realm. Realms are separate from one another and are limited in their ability to administer and authenticate people. In our case, the JDBC Security Realm has access to the database through the Payara data source, which is in line with how realms typically relate to data sources [14].

The creation of the JDBC Security Realm also includes the configuration of password hashing.

5.3 REST API

"An API, or application programming interface, is a set of rules that define how applications or devices can connect to and communicate with each other. A REST API is an API that conforms to the design principles of the REST, or representational state transfer architectural style. For this reason, REST APIs are sometimes referred to RESTful APIs." [15]

The client communicates with the server through the RESTful API and utilizes HTTP requests to access and retrieve data. When a third party communicates with an application interface, it makes calls to special endpoints that are identified by a service or server's URL. So for instance a GET request for retrieving all project would be as follows:

```
/api/projects/all:
  get:
    operationId: getProjects
    responses:
      default:
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/ProjectOverviewDto'
        description: Default Response.
```

Table 5.1. Code extract: GET request for retrieving all project (extracted from localhost:4848/openapi)

Following that, an appropriate class and functions are in charge of handling the requests–interpreting the requests, getting the entities ready, and providing responses.

A Data Transfer Object (DTO) is used to pass the data. This enables the creation of many representations of the same data that is adjusted to the requirements of the client. These classes transfer only the necessary attributes and intricate dependencies are intentionally omitted. For example, the client does not need the whole project with all its attributes and information, therefore a shorter overview of the Project is enough. A DTO of such object would look like this:

```
ProjectOverviewDto:
  type: object
  properties:
    projectId:
      type: integer
    projectCreator:
      type: string
    name:
      type: string
    description:
      type: string
    dateTimeCreated:
      type: object
    category:
       $ref: '#/components/schemas/CategoryDto'
    positionsWanted:
      type: array
      items:
        $ref: '#/components/schemas/PositionsWantedDto'
    maxMembers:
      type: integer
    memberCount:
      type: integer
```

Table 5.2. Code extract: DTO of Project (extracted from localhost:4848/openapi).

5.4 Front-End

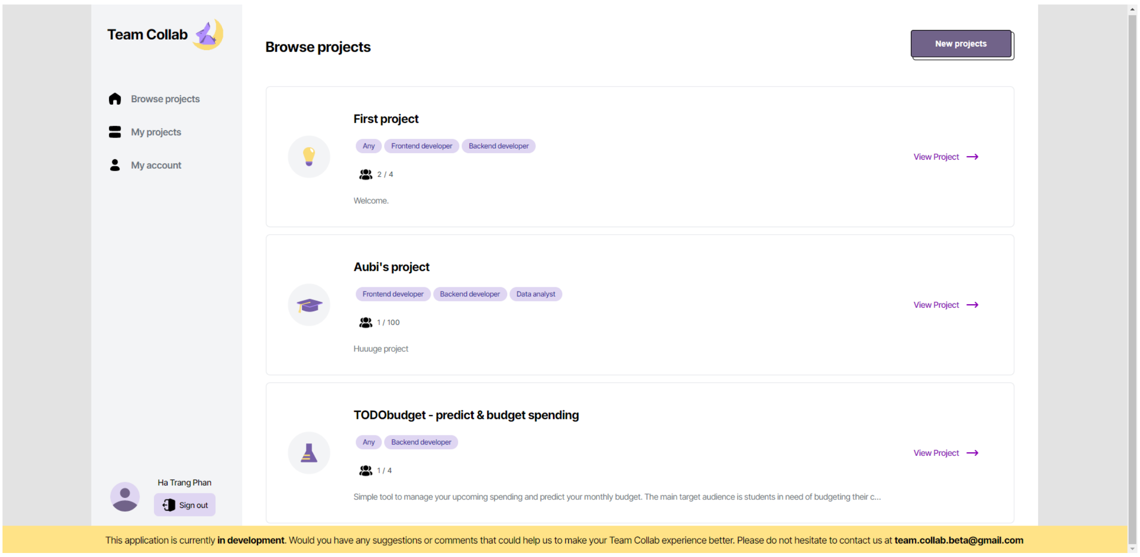
The front-end side of the application is fully implemented in plain JavaScript by using the technology of the Web Components. They enable the reuse of smaller functional units in other areas of the same application. Thus, each HTML page has its JavaScript file that would build the needed UI components accordingly. Components are created by extending the `HTMLElement` class and defined with `customElements` in code 5.3.

```
class ProjectItem extends HTMLElement {
...
}

customElements.define('c-project-item', ProjectItem)
```

Table 5.3. Code extract: Creation of ⟨*c-project-item*⟩ element

Figure 5.2 is a screenshot of the main Home page of the web application. This page serves as a project browser. It fetches `ProjectOverviewDTO` and creates a component ⟨*c-project-item*⟩⟨*/c-project-item*⟩ for each project it gets. The received data is subsequently passed by setting the property of the Web Component `c-project-item` with the data.



Browse projects

First project

Any · Frontend developer · Backend developer

2 / 4

Welcome.

Aubi's project

Frontend developer · Backend developer · Data analyst

1 / 100

Huuuge project

TODObudget - predict & budget spending

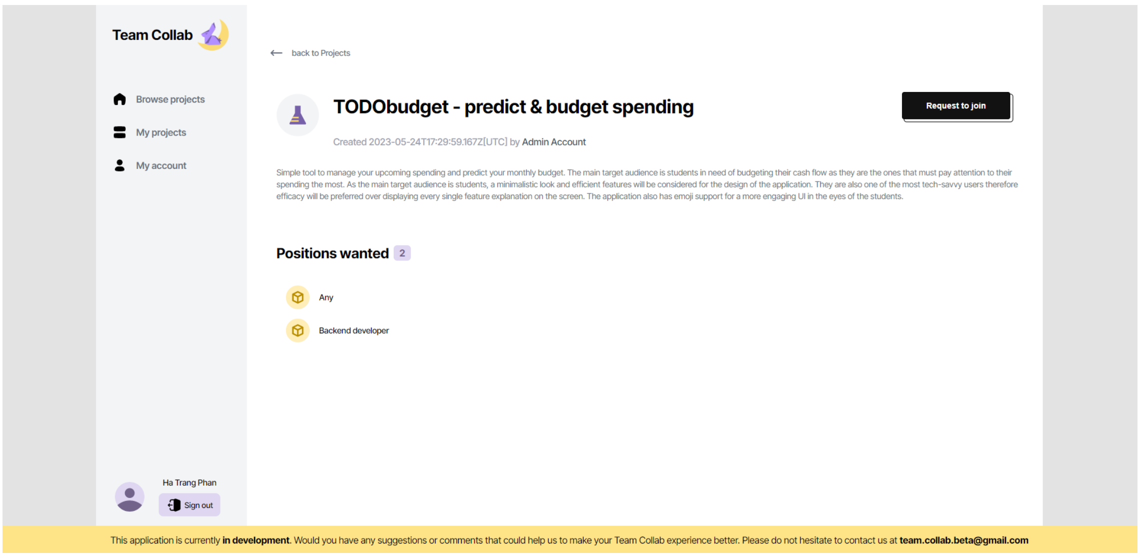
Any · Backend developer

1 / 4

Simple tool to manage your upcoming spending and predict your monthly budget. The main target audience is students in need of budgeting their c...

This application is currently **in development**. Would you have any suggestions or comments that could help us to make your Team Collab experience better. Please do not hesitate to contact us at **team.collab.beta@gmail.com**

Figure 5.2. Screenshot of the Home page.



TODObudget - predict & budget spending

Created 2023-05-24T17:29:59.167Z[UTC] by Admin Account

Simple tool to manage your upcoming spending and predict your monthly budget. The main target audience is students in need of budgeting their cash flow as they are the ones that must pay attention to their spending the most. As the main target audience is students, a minimalistic look and efficient features will be considered for the design of the application. They are also one of the most tech-savvy users therefore efficacy will be preferred over displaying every single feature explanation on the screen. The application also has emoji support for a more engaging UI in the eyes of the students.

Positions wanted 2

Any

Backend developer

This application is currently **in development**. Would you have any suggestions or comments that could help us to make your Team Collab experience better. Please do not hesitate to contact us at **team.collab.beta@gmail.com**

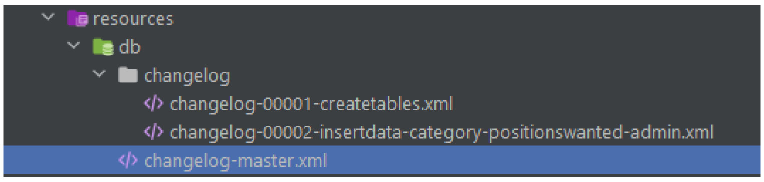
Figure 5.3. Screenshot of the Project detail page from the view of a non-member user.

With the creation of individual custom elements, a different view can be displayed based on its specified condition. For example, a user has different views based on their relation to the project. A user that does not belong to the project would view the Project detail page as shown in Figure 5.3, but if they were a member of the project, they would be able to see different information and functionalities.

5.5 Liquibase

Liquibase helps in tracking, versioning, and deploying database code [16].

"Liquibase All database changes are specified in the Liquibase changelog file. A change is contained in a changeset and changesets are added to the changelog in the order they need to be deployed." [16]



```
resources
  db
    changelog
      changelog-00001-createtables.xml
      changelog-00002-insertdata-category-positionswanted-admin.xml
    changelog-master.xml
```

Figure 5.4. Liquibase directory from Team Collab project

With Liquibase, there is no need to worry about the changes made to the database, as a changelog file is created to track each change. These changelogs are referenced in the root file called `changelog-master.xml` (as shown in Figure 5.4), and they are available in various formats, including XML. The `changelog-00001` file creates the database tables, while the `changelog-00002` file populates them with the necessary data for the application to operate as intended.

Chapter 6
Testing

The front-end of the Team Collab posed significant challenges due to the utilization of unfamiliar web component technology. Several setbacks were encountered during the implementation process.

The process of determining the logic and deciding on data flow between parent and child components proved to be tedious, particularly without the aid of any libraries.

Careful attention had to be given to the rendering of HTML elements on the page, as web components utilise the shadow DOM. Due to their placement in a separate DOM, components remained hidden and needed careful handling.

Complications may arise when using the Fetch API for data retrieval through the `await` Promise, such as when attempting to attach an event listener to an element that is not yet present. Debugging such issues can be extremely time-consuming.

Since web component technology is a fundamental aspect of this project, it is imperative to conduct thorough testing of its components. GUI testing emerges as a suitable approach for this purpose, and options such as Selenium and Playwright are to be considered.

Both Playwright and Selenium provide compatibility with a range of web browsers and programming languages. They facilitate the simulation of user interactions involving mouse and keyboard movements. Therefore, the selection of a testing tool becomes a subjective decision, as their functionalities are comparable.

The author has implemented GUI testing of the web application using Playwright.

6.1 Playwright Testing

Playwright is a testing framework that supports web browsers like Firefox, Chromium and Webkit. Its API is available in .NET, JavaScript, TypeScript, Python and Java.

This project will primarily focus on creating GUI tests in Playwright Java since it is distributed as a maven module and merely needs a dependency to be added to the `pom.xml` file. It also enables running with test-runners like JUnit in the following manner:

```
public class ProjectCreationTest {

    static Playwright playwright;
    static Browser browser;

    BrowserContext context;
    Page page;

    @BeforeAll
    static void launchBrowser() {
        playwright = Playwright.create();
```

```
            browser = playwright.chromium().
            launch(new BrowserType.LaunchOptions().setHeadless(false));
        }

        @AfterAll
        static void closeBrowser() {
            playwright.close();
        }

        @BeforeEach
        void createContextAndPage() {
            context = browser.newContext(new Browser.NewContextOptions().
            setStorageStatePath(Paths.get("appLoginInfo.json")));
            page = context.newPage();
        }

        @AfterEach
        void closeContext() {
            context.close();
        }

        @Test
        public void CreateProjectTest() {
        ...
        }
    }
```

Table 6.1. Code extract: Running Playwright with Junit

In the test setup, Playwright and the Browser are initialized in the `@BeforeAll` method and subsequently terminated in the `@AfterAll` method. As a result, the test methods will share the same Browser instance while each test will have its own Browser-Context and Page [17].

6.2 Testing Scenarios

In chapter 3.4.1 there are three main tasks that can be done in Team Collab. And the testing scenarios for those user stories are similarly planned in following tables 6.2, 6.4, 6.3.

During the testing of test scenario 6.2, the following bugs were identified: the Login screen failed to redirect successfully authenticated users to the Projects screen. This bug has been resolved. The issue originated with the `j_security_check` mechanism, which redirected users to the web application's root instead of the intended page. It was observed during testing that the handling of unsuccessful login attempts is not functioning properly and requires further attention.

Name:	User registration and login
Actor(s)	Guest user
1.	Guest user clicks `Register` button
2.	Guest user fills in the form on the Register page
3.	Guest user clicks `OK` to confirm registration
4.	Guest user fills in login details
5.	Guest user clicks `Sign in` button
Expected results:	Guest user creates a profile and can log in.

Table 6.2. Test scenario for User registration and login

Name:	Project creation
Actor(s)	Common user
1.	Common user clicks `New project` button
2.	Common user clicks fills the form
3.	Common user clicks `Publish project` button
Expected results:	New project item in `My projects` page and Common user has become Project Owner of the project

Table 6.3. Test scenario of creating a project

Name:	Sending and evaluating a join request
Actor(s)	Common user, Project owner
1.	Common user A becomes the Project owner
2.	Common user B clicks `Request to~join` button
3.	Common user B fills a `Message` in request
4.	Common user B clicks `Send request` button
5.	Project owner clicks `View requests` button
6.	Project owner clicks button: a) `Accept` — Common user B becomes Project member b) `Reject` — Common user B sees the request as declined
Expected results:	Project owner approves or declines the user and it is saved accordingly in the project.

Table 6.4. Test scenario for sending and evaluating a join request

Test scenario 6.3 faced some issues during its testing. It raised many questions as the manual user testing worked, but the Playwright tests kept failing. To identify the problem, the tracing feature that Playwright provides was used to troubleshoot. It was observed that during the test assertion method; the components were not yet rendered, therefore the tests kept failing. This issue was resolved by adding `page.waitForLoadState(LoadState.NETWORKIDLE)` before checking the results. The same issue was identified in 6.4 otherwise both testing scenarios successfully passed.

Chapter 7
Conclusion

The primary goal of the project was to create a tool that can help alleviate the problems students come across when they look for teams to join. Searching for teammates can be a difficult task when the student is in an unfamiliar environment. Team Collab intends to make this process easier by providing a space where they can easily find the right projects for them. The purpose of this application is to help the students progress in their academic and professional careers.

After the analysis of the current state of the problem, the future state of the web application has been proposed and broken down in more detail. By establishing user roles, requirements, and use cases and emphasising UX and UI design, the envisioned state of the project was clearly defined.

Subsequently, the back-end of the application was made using Jakarta EE 8 and Payara Server, and the most challenging technology used in the project was creating the user interface with only plain JavaScript using Web Components. The author has struggled the most with data passing between the components. One of the most time-consuming activities during the implementation was sorting the order of creating and using components since many of them were inside each other. Finally, the web application underwent testing scenarios and GUI testing using Playwright.

Despite the laborious work involved with web components as a stand-alone technology, it presents an opportunity to work with other libraries and frameworks thanks to its reusability and being based on existing web standards.

During the initial stages of project analysis, the product features and its high-fidelity prototype were showcased and demonstrated to a group of 30 international students at the University of Seoul. Team Collab received a positive response from these students, with many expressing their enthusiasm for the prototype and saying that it would significantly make group projects much easier. Even though the sample of research is not representative of large-scale projects, the positive feedback from target audience proves this project has potential for further development.

7.1 Future Work

The tool could be further improved by incorporating an ontology-based approach where the searched keyword would show results based on its keyword and semantic relation [18].

Finding a suitable project is crucial to achieving the best results. Implementing a search engine and filtration system means the user can efficiently seek available projects. Creating a full-text search that matches a phrase against all the words in the database facilitates the overall work of result browsing [19].

Furthermore, the research proposes the possible integration of additional communication functionalities, such as the attachment of document files or the inclusion of voice messages in requests to join.

References

[1] Rebecca Jackson. *Why group projects fail*. 2015.
`https://www.psychologytoday.com/us/blog/school-thought/201503/why-group-projects-fail`.

[2] Carnegie Mellon University. *What are the benefits of group work? - eberly center - carnegie Mellon University*.
`https://www.cmu.edu/teaching/designteach/design/instructionalstrategies/groupprojects/benefits.html`.

[3] Megan Lutz, and Steven Culver. The National Survey of Student Engagement: A university-level analysis. *Tertiary Education and Management*. 2010, 16 35-44. DOI 10.1080/13583881003629814.

[4] Cennydd Bowles, and James Box. *Undercover user experience design: Learn how Todo great ux work with tiny budgets, no time, and limited support*. New Riders, 2011.

[5] PWL Bollen, and others. *BPMN: a meta model for the happy path*. Citeseer, 2010.

[6] Jenifer Tidwell, Charles Brewer, and Aynne Valencia. *Designing Interfaces*. 3 ed.. Sebastopol, CA: O'Reilly Media, 2020.

[7] Sacha Greif, and Eric Burel. *State of JavaScript 2022*. 2023.
`https://2022.stateofjs.com/en-US/libraries/front-end-frameworks/`.

[8] Mohit Joshi. *Angular vs react vs Vue: Core differences*. 2022.
`https://www.browserstack.com/guide/angular-vs-react-vs-vue`.

[9] A. Chiarelli. *Exploring Web Components: Build Reusable UI Web Components with Standard Technologies*. BPB PUBN, 2020. ISBN 9789389423976.

[10] Sandeep Kumar Patel. *Learning web component development*. Birmingham, England: Packt Publishing, 2015.

[11] Ben Farrell. *Web Components in Action*. New York, NY: Manning Publications, 2021.

[12] David Heffelfinger. *Java EE 8 Application Development*. Birmingham, England: 2018 . ISBN 9781788297332.

[13] P Spath. *Beginning Jakarta EE : Enterprise Edition for Java: from Novice to Professional, Apress L*. Berkeley, CA. Ebook Central: Apress L. P, 2019.

[14] A. Tijms, T. Bais, and W. Keil. *The Definitive Guide to Security in Jakarta EE: Securing Java-based Enterprise Applications with Jakarta Security, Authorization, Authentication and More*. Apress, 2022. ISBN 9781484279441.

[15] *What is a rest api?*
`https://www.ibm.com/topics/rest-apis`.

[16] Kchappell. *Getting started: Liquibase best practices*. 2022.
`https://www.liquibase.org/get-started/best-practices`.

[17] *Test Runners.*
`https://playwright.dev/java/docs/test-runners`.

[18] Yu Hou, and Lixin Tao. An Ontology-based Ranking Model in Search Engines. *Journal of Computer Science Research.* 2019, 1 DOI 10.30564/jcsr.v1i2.972.

[19] *Co Je to Fulltext / Fulltextové vyhledávání? Definice Pojmu.* 2022.
`https://topranker.cz/slovnik/fulltext-fulltextove-vyhledavani/`.

Appendix A
Acronyms

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
XML	Extensible Markup Language
DOM	Document Object Model
GUI	Graphical User Interface
UI	User Interface
UX	User Experience
JPA	Java Persistence API
DTO	Data Transfer Object
API	Application Programming Interface
SVG	Scalable Vector Graphics