Interactive Gamified Portfolio

TCS - Module 11 - Design Project

Stijn Broekhuis
s2121697

Matthias Sleurink
s2109336

Frans de Boer
s2191202

Team 6
2021-04-16
m.sleurink@student.utwente.nl

Adarsh Aaron Denga
s2074591

Geronimo Wolf
s2179245

1 Table of Contents

2 Requirement Specification

2.1 Users

2.1.1 Student

A student is able to have badges and titles, as well as export a CV-like file that displays these.

2.1.2 Teacher

A teacher is allowed to assign badges, and by virtue of adding badges can also assign titles.

2.1.3 Module Coordinator

A module coordinator is allowed to add and edit badges and titles. They are also able to do everything a teacher can.

2.1.4 Administrator

An administrator is allowed to assign the administrator role to teachers, as well as a teacher role to students (like TA's and the like). As well as seeing the logs the application creates for actions done through the system. They are also able to do everything a module coordinator can.

2.2 Pages

2.2.1 Overview

This is the students' main page. They will see an overview of their modules, and a listing of the titles they have, as well as a toggle that allows them to also see the titles that they *can* get.
The students can click on the modules to see an overview of the badges (and titles) they have gotten in that module.

2.2.2 Badge Creation and Editing

This is a page for the administrators. They can search existing badges and edit them, or create completely new badges.

2.2.3 Badge Assign

This is a page for the teachers and administrators. They can search for a badge, and edit what students 'have' that badge, and what level the student has. They can also export a file, where they can perform these actions outside of the browser, in a tool of their liking. This file will provide a template that the user can fill in so that, upon uploading the file, the server can take the file and apply the changes made.

2.2.4 Title Creation and Editing

This is a page for module coordinators and administrators. They can search existing titles and edit them, or create completely new titles. They can also link badges to titles to state which badges are required for a title to be earned.

2.2.5 Teachers

This is a page for module coordinators and administrators. They can add or remove the privilege of assigning badges from teachers and TAs.

2.2.6 Logs (Not implemented)

This is also a page for module coordinators and administrators. They can see all changes to the badges, titles, and assignments of those elements that have been made and by who. This makes it possible to spot unfair or even fraudulent behaviour.

3 Global design

3.1 General Use

This canvas plugin is a Badge-Title system for students to get a better understanding of the topics or skill they have mastered over the course and years during their study.

These badges and titles will serve as a measurement for themselves, to know what they are good at. This can then be used to help them create a resume that shows these skills and topics. For example, a “Best Presentation” badge can give a student confidence to put “Good Presentation Skills” on their resume.

In section *3.2 Roles* we explain the difference between Student, TA, Teachers, and more. This will further explain what each role can do in the system. For example, Teachers can assign badges to students from the module that they launched the system, but they cannot create badges.

The system is module-based. This means that users approach the plugin via a link inside a module. For teachers, and administrators functionality is limited to the module that they have open in canvas. For students this is not the case. They open the plugin the same way, but can then choose what module they want to see the badges from inside the plugin.

3.1.1 Badges

Badges are assigned to students. They have a list of levels, each of which has a description and an icon. Students will see the badges (and the level of that badge) that they have received, as well as an uncollected/default image for badges that they have not received (yet).

3.1.2 Titles

Titles are not assigned by anyone, they are assigned automatically based on the badges that a student has collected. Titles have requirements for badges and once they are met, a student will receive said title.

3.2 Roles

	View Owned Badges & Titles	Assign Badges	View Logs (Not implemented)	Assign Roles	Create Badges & Titles
Student	X				
Teacher/TA		X			
Admin & Module Coordinator		X	X	X	X

- **View Owned Badges**, See badges that you have or haven't collected in a module and what rank of the badge you have.
- **Assign Badges**, Set the rank of a badge for all students from a module.
- **View Logs**, any actions, such as assigning badges, are remembered and shown on a log page.
- **Assign Roles**, Teachers/TA are allowed to set badges for students in a module by *default*, if for some reason one does not want this or want a specific Teacher/TA not be able to set badges, they can do so.
- **Create Badges & Titles**, the ability to create badges and titles is limited to roles that design the modules that they are used in. Therefore, not everyone needs to be able to create them.

3.3 Canvas Style

Since this plugin is built into canvas we made the system to have a style unified with the one canvas uses (Instructure, n.d.). We used the canvas style guide extensively when creating visuals in our pages. This includes buttons - text - page layout and icons. In section *4.6 Screenshots*, a preview can of the system can be found.

4 Detailed Design

4.1 LTI Authentication

LTI is used for authentication. When a user tries to access the app from canvas, first a request gets sent to an endpoint on canvas. This request is automatically filled with data about the currently logged in user. If canvas replies with an OK response, the user gets directed to our plugin, and the correct pages will be displayed. All of this happens invisibly and seamlessly for the user, and it allows the backend to obtain information about the user and the current module without having to store this itself.

4.2 Diagrams

4.2.1 Action Diagram

In *Appendix A*, images can be found of the Action Diagrams for the different user groups.

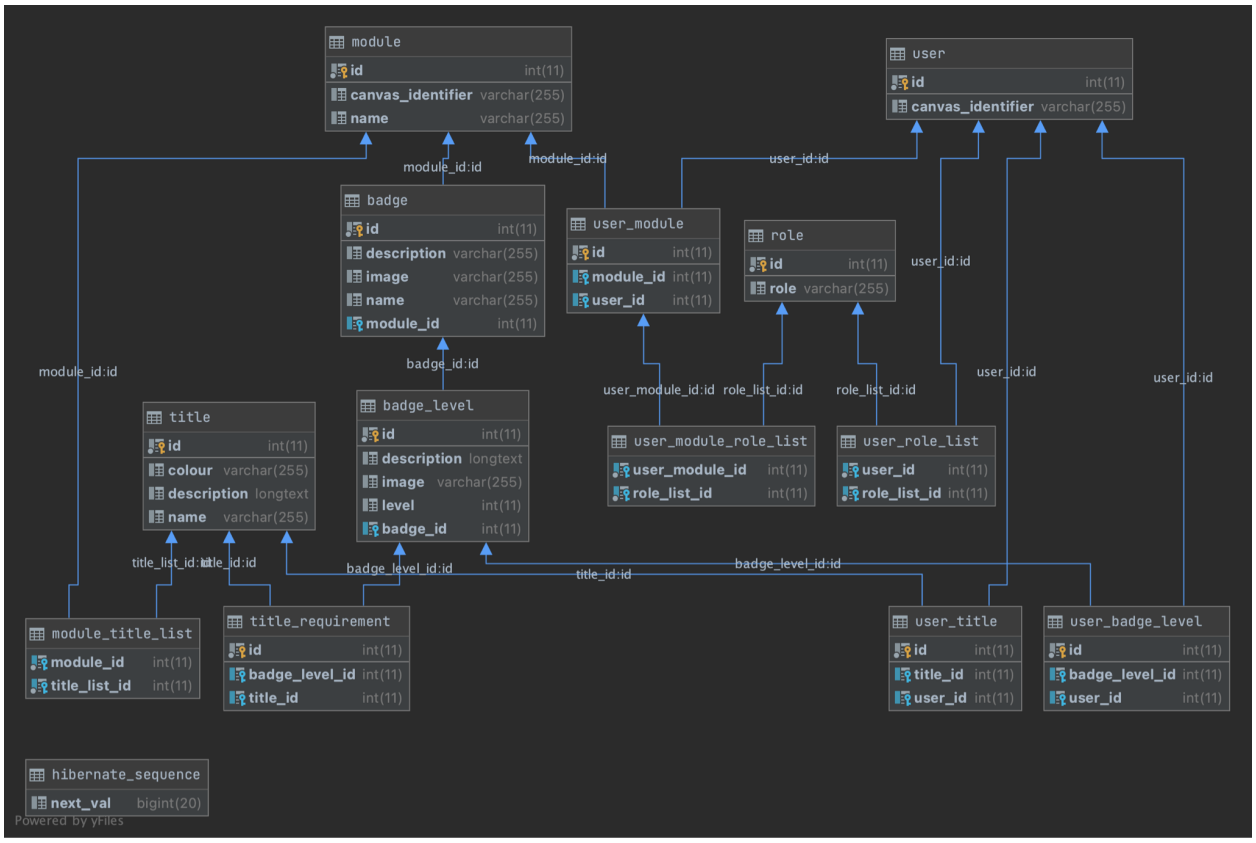
Before starting any programming or design work, action diagrams were created based on the requirement specifications. These action diagrams helped to more accurately describe the way students, teachers, and admins interact with the system. Because these action diagrams were created at such an early stage of the project, small changes had to be made, but overall the action diagrams stayed very accurate, which made it easy to implement the functionality they described, and to check if all requirements were completed.

4.2.2 Class Diagram

Before starting work on the database, a class diagram was created. This class diagram saw many iterations, and even in the final weeks of the project additions were made. A few important things to note are that no personal data is stored in the user (more on this in the next section). Users can have titles, and badges. A user can have a role in the entire system, or a role in one module only. This is used for giving/taking away badge assign privileges in a module.

One thing that is still missing is a SuperModule, something that stands above a module that holds most of the important information about the module. Right now the same module will have a different database entry for each year it is given, meaning that the badges will have to be either remade or copied over from the previous year. This could be avoided by making a SuperModule above it, although this comes with its own problems. If a badge has to be deleted in a new instance of a module, it will also be deleted from all previous years, and students might lose their badges.

There is one class that doesn't have much to do with the project. It is called `Hibernate_sequence`s, and is automatically generated by spring. It can be ignored by the developer.



4.3 Data Storage in Database:

When authenticated, Canvas sends over quite a lot of personal data, such as a person's full name, email address, and student number. In accordance with the data minimization principle none of this is stored, and instead LTI fills in this data from the authentication before sending a response to the frontend.

We use a `canvas_identifier`, which is a unique identifier the authentication sends for each user, to connect a user in our database back to canvas. One thing that is stored from the authentication is the name of the current module. This is then used to display to the user a list of all modules they have participated in that used this plugin.

4.4 Frameworks

For the backend Java Spring was used, and for the frontend Angular. While most of the team did have some experiences with both frontend and backend frameworks, no members of the team have any past experiences with these frameworks. The decision was made to use these frameworks to make it easier for the university's technical staff to maintain the plugin in the future, which would increase the likelihood of the plugin being used and further developed after the project has been completed.

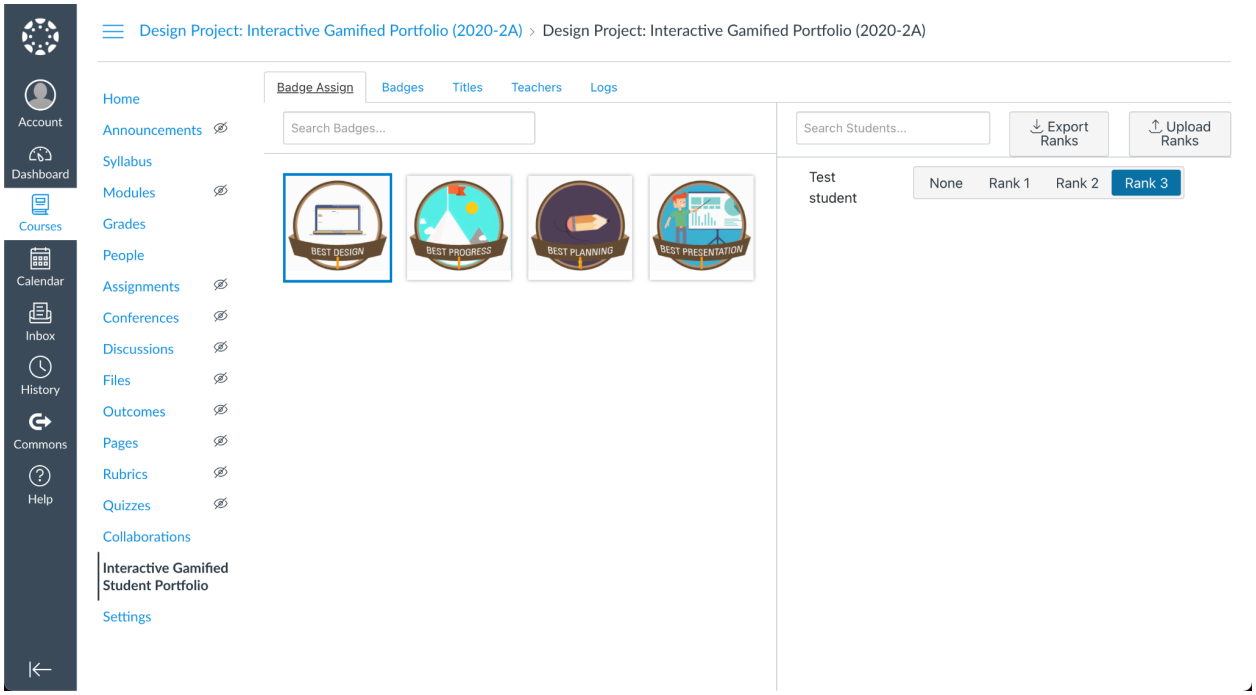
Angular's component based structure allowed for easy page creation, and the routing system was used to display different pages depending on the roles of the user. This, combined with the Single Page Application structure that Angular's routing provides, made for pages that load instantly (after the initial loading time).

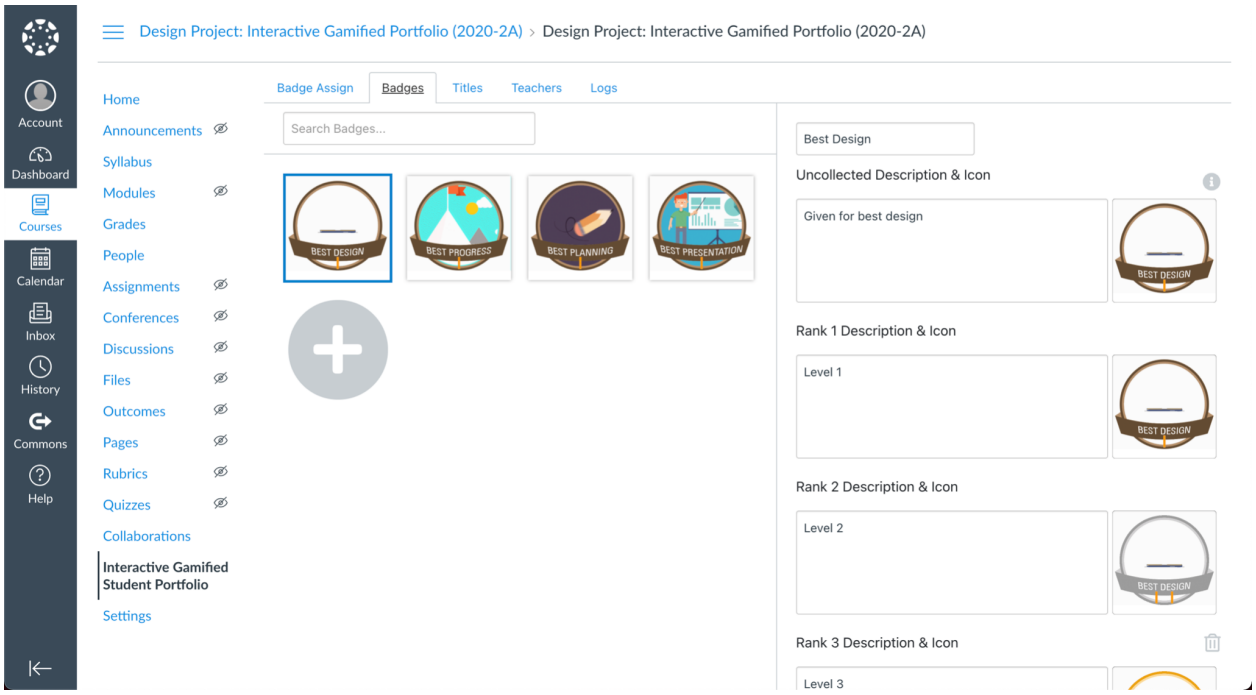
Spring made security extremely easy. After some initial struggle with setting up the roles, it was possible to disable any API endpoint depending on if the user is logged in with a sufficient account, and hide the sensitive data it contains. After getting over the learning curve of a completely new framework it became very nice to work with, and things like creating a new entity (database entry), and new routes was just a few seconds of work.

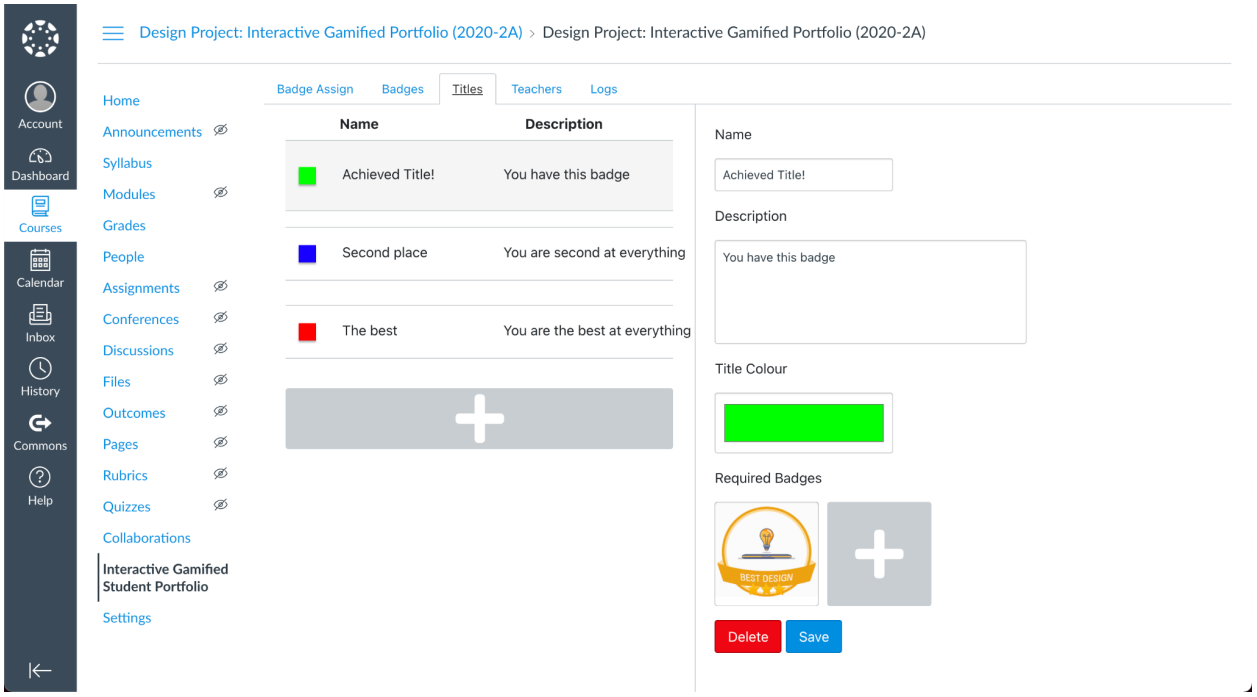
4.5 Rest API (Frontend and Backend Disconnected)

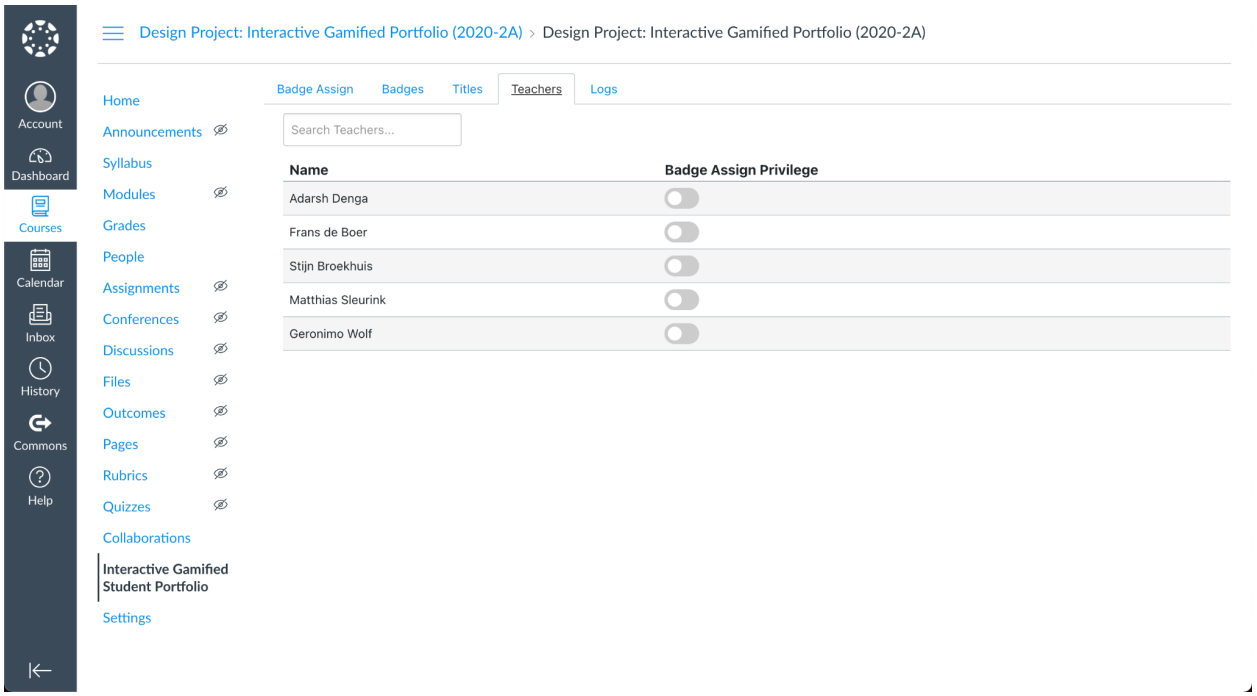
Another technical decision that was made was to have the frontend and backend as two separate projects, and use a REST architecture to connect them. This was done so a Single Page Application could be created using Angular, that can dynamically retrieve all data necessary from the API. Working with the frontend and backend disconnected made some things in the workflow more difficult, for example testing the plugin properly required building the entire frontend which could take upwards of a minute, and adding a page and its links to the backend required good communication between the frontend and backend team. Once something is implemented though this way of working makes it very pleasant to use, and tasks appear to be executed instantly for the user.

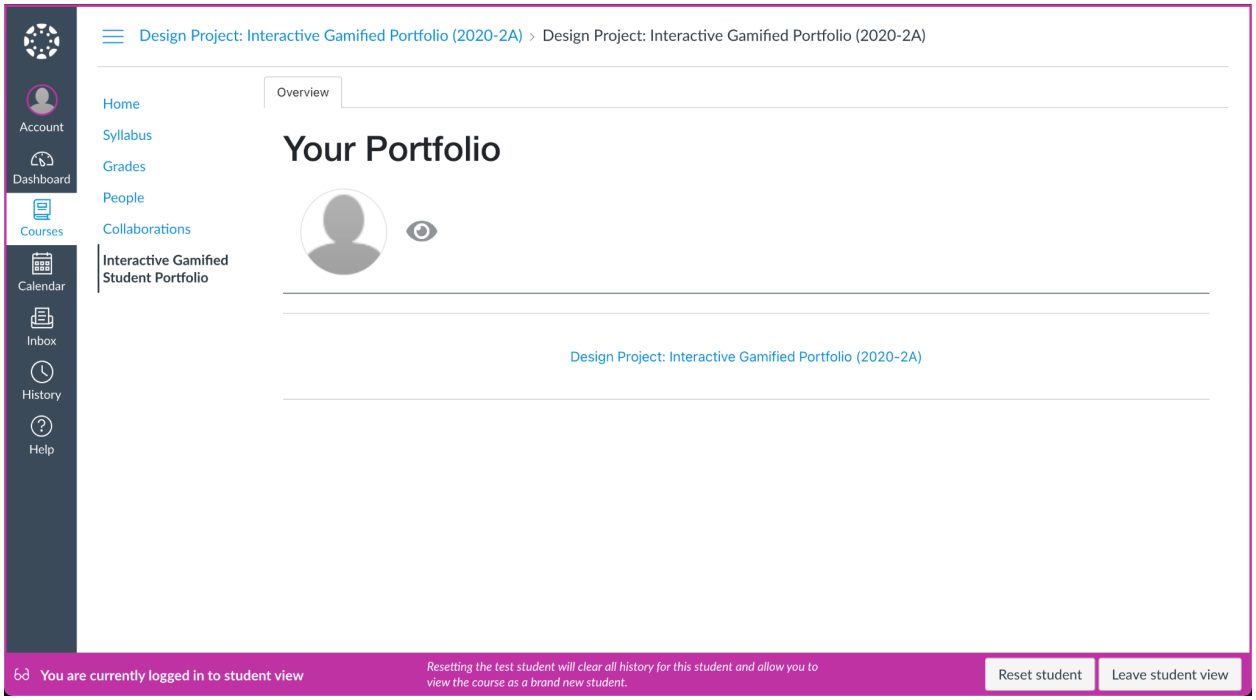
4.6 Screenshots

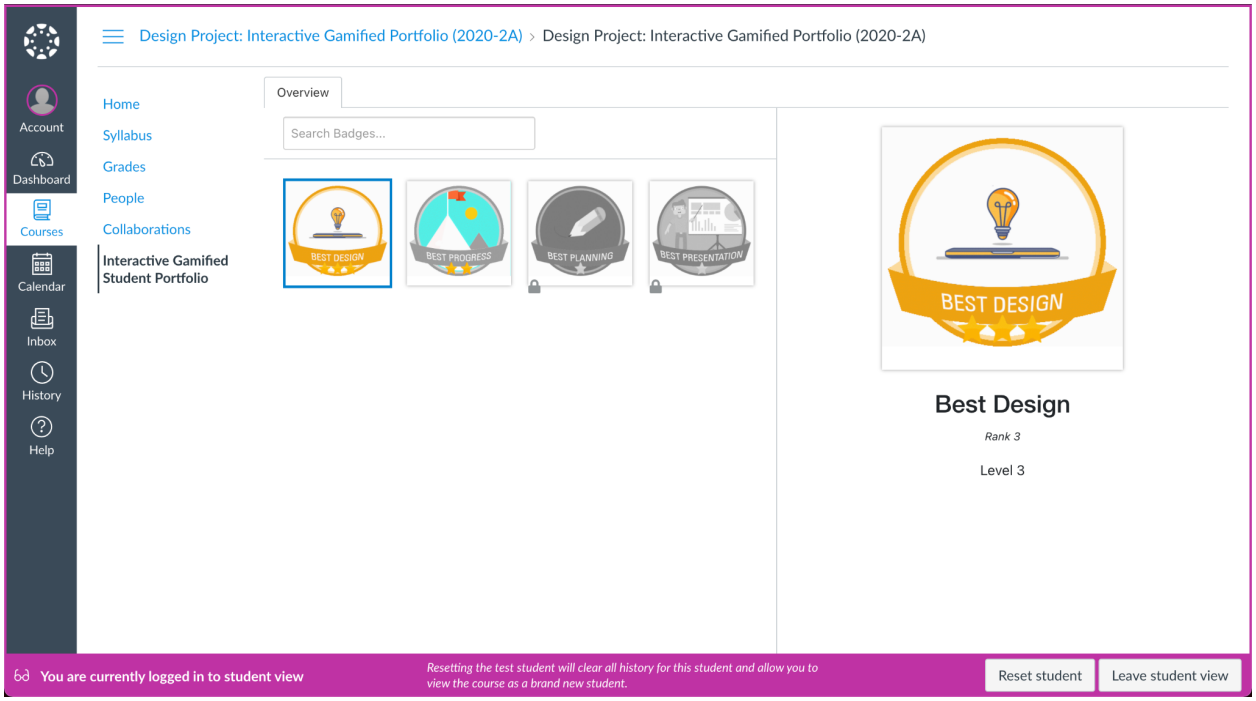





Design Project: Interactive Gamified Portfolio (2020-2A) › Design Project: Interactive Gamified Portfolio (2020-2A)
Account
Dashboard
Courses
Calendar
Inbox
History
Commons
Help
Home
Announcements
Syllabus
Modules
Grades
People
Assignments
Conferences
Discussions
Files
Outcomes
Pages
Rubrics
Quizzes
Collaborations
Interactive Gamified Student Portfolio
Settings
Badge Assign
Badges
Titles
Teachers
Logs
Name
Description
Achieved Title!
You have this badge
Second place
You are second at everything
The best
You are the best at everything
Name
Achieved Title!
Description
You have this badge
Title Colour
Required Badges
BEST DESIGN
Delete
Save


Design Project: Interactive Gamified Portfolio (2020-2A) › Design Project: Interactive Gamified Portfolio (2020-2A)
Account
Dashboard
Courses
Calendar
Inbox
History
Commons
Help
Home
Announcements
Syllabus
Modules
Grades
People
Assignments
Conferences
Discussions
Files
Outcomes
Pages
Rubrics
Quizzes
Collaborations
Interactive Gamified Student Portfolio
Settings
Badge Assign
Badges
Titles
Teachers
Logs
Search Teachers...
Name
Badge Assign Privilege
Adarsh Denga
Frans de Boer
Stijn Broekhuis
Matthias Sleurink
Geronimo Wolf


Design Project: Interactive Gamified Portfolio (2020-2A) › Design Project: Interactive Gamified Portfolio (2020-2A)
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Syllabus
Grades
People
Collaborations
Interactive Gamified Student Portfolio
Overview
Your Portfolio
Design Project: Interactive Gamified Portfolio (2020-2A)
You are currently logged in to student view
Resetting the test student will clear all history for this student and allow you to view the course as a brand new student.
Reset student
Leave student view


Design Project: Interactive Gamified Portfolio (2020-2A) › Design Project: Interactive Gamified Portfolio (2020-2A)
Account
Dashboard
Courses
Calendar
Inbox
History
Help
Home
Syllabus
Grades
People
Collaborations
Interactive Gamified Student Portfolio
Overview
Search Badges...
BEST DESIGN
BEST PROGRESS
BEST PLANNING
BEST PRESENTATION
BEST DESIGN
Best Design
Rank 3
Level 3
You are currently logged in to student view
Resetting the test student will clear all history for this student and allow you to view the course as a brand new student.
Reset student
Leave student view

5 Test Plan and Test Results

To ensure the UT's student's privacy the developers key we had received had to be kept secret. Testing during the development phase was therefore difficult. Also because the program was hard to set up so the average student and module coordinator would not be able to easily test the program out and give us their input. Instead, we decided to do weekly testing together with our team's supervisor, besides the prototype testing during week 2.

5.1 Weekly Meetings

Every week, on Monday morning, a meeting was held together with our supervisor Yeray Barrios and later on with Manon Hulsbeek as well. During this meeting, ideas were discussed and the developments were tested. The system was shown using the screen share function on Microsoft Teams.

5.2 Week 2

At the end of the second week the hi-fi prototype was finished on Framer (Framer, 2021). Four prototypes were made for the different user groups:

Students:
https://framer.com/share/Gamified-Portfolio--wSJOfTWneImooFQo6J7A/Pofy3Kmor
Teachers:
https://framer.com/share/Gamified-Portfolio--wSJOfTWneImooFQo6J7A/PMj_DwoYz
Managers:
https://framer.com/share/Gamified-Portfolio--wSJOfTWneImooFQo6J7A/jBK6R225V
Admins:
https://framer.com/share/Gamified-Portfolio--wSJOfTWneImooFQo6J7A/YrFtH7KvP

Four people were invited to interact with the prototypes and be interviewed one-on-one. The subjects were asked to share their screen and show their interaction with the prototype. The test subjects included one module coordinator and three students. Each subject gave criticism and compliments that will be described in detail below.

5.2.1 Subject 1 (module coordinator)

Subject 1 believed that there should be guidelines or criteria on when someone should get a badge. Different teachers have different mindsets. To ensure the badges system works properly and titles are not distributed unfairly, a page should be dedicated towards teaching the teachers on how to use the badge system.

In addition, in the badge assignment page, distribution percentages of each rank should be added. Then, a little note ‘recommended percentages’ should be added so all teachers distribute about the same amount of badges. This is to ensure proper distribution of badges among students in all modules.

During the creation of a badge, the module coordinator should be able to add criteria to a badge. For instance for “Best planning”, there should be criteria that the module coordinator can enter such as: “Always on time”, “Always handed in assignments on time”, “Ask questions”, and “Sends emails”.

Subject 1 gave sightful information about why such a system would be useful for module coordinators. It could for instance be used for the TA planning system. As a teacher you would want to have access to your students portfolio. See their badges, their progress, etc. This system can be used to portrait a good representation of a student’s strengths and discipline, simple grades do not always suffice.

Discussion

Subject 1, being a module coordinator, gave many useful insights in the project which we, as students, did not notice ourselves. A module coordinator has a significantly different view on such a system than students do. All of the ideas mentioned by Subject 1 were good valid points and were taken into consideration during the remainder of the project.

In the User Manual, guidelines for handing out badges were given to ensure each teacher had a similar mindset. For a future project, this could be added in a separate page on the Canvas plugin for teachers. Guidelines are important since different teachers will have different views on this system. This can be because of personal reasons but also cultural differences. To ensure a similar number of badges are handed out for each badge, guidelines will be added. If not, some badges will become extremely rare while others will be common. This is not the intended system. Furthermore, teacher access to students’ portfolios is one of the first future advancements we would make.

5.2.2 Subject 2 (student)

Subject 2 believed that badges should be smaller. The student should be able to have an overview of all badges. Seeing three badges as a carousel simply increases the effort required to navigate.

In addition, the teacher should have more information on how to use the badges. Subject 2 did not know how to get the process started of giving a badge to a student. Badges could have default pictures loaded in when creating a new badge. This would make it clearer to the teacher what the purpose is. Subject 2 did not know what the meaning and purpose of the "default picture" was. Everything could use some extra text with explanation.

The logs page should be expanded. For instance, add the possibility to see the logs per person or per module. The admin should be able to filter the logs.

Discussion

Subject 2 noted that the student should be able to have an overview of all the badges, rather than just three. We decided that this was indeed true and would result in quick and easy navigation between the badges.

As Subject 1 had mentioned, the teachers should have a similar mindset when handing out badges so a separate screen for teachers with information on how to use the badges was a great idea by Subject 2.

The logs page was not finished at the end of the project and thus can be expanded as much as administrators deem necessary during future expansions of the project.

5.2.3 Subject 3 (student)

Subject 3 was interested in the innovative idea of badges representing a student's progress. However, the purpose of the summary page was not entirely clear. In addition, the Subject mentioned that clicking on titles to see the description would be a nice feature.

For the teacher screen, Subject 3 gave a recurring suggestion, the ability to see students their badges.

The badge creation and title creation for managers was self-explanatory and the design was user friendly. However, a badge's badge levels should each have its personal description. This could give the student more information about their progress or how well they did in that particular subject and the reward can be more personal. Subject 3 also mentioned that titles do not necessarily need to contain descriptions, as that is too much information.

Administrators should not be the only ones to have the ability to see logs. Subject 3 believes that teachers should also have access to this feature. He also mentions that teachers should be able to decide if TA's can assign badges or not. The roles of TA's and teachers should be split.

Discussion

Subject 3 gave lots of positive feedback, which is amazing to hear. Subject 3 also mentioned, similarly to Subject 1, that teachers should be able to view their students' badges.

Furthermore, Subject 3 talked a lot about the separation of roles between TA's and teachers. This would be ideal, however it is difficult to realise in the LTI framework we use in our project. A TA and a teacher have the same roles inside a module so it is not possible to differentiate the two. To do so, direct API calls to Canvas should be made.

We have added that each badge level has its own description as it increases the ability for teachers to more precisely hand out badges. Logs in the end did not get added to our final product and thus will be added to future possible advancements.

5.2.4 Subject 4 (student)

Subject 4 gave good remarks but ones we cannot have much impact on. For instance, Subject 4 explained their fear that a coordinator has to plan around the badge system. They describe that some modules might just not fit into the whole idea of a badge system in the first place. E.g. when a module is heavily lecture based. Furthermore, there may be bias/conflict between people due to the number of badges. Teachers may choose people with a greater number of badges to be TAs and so on even if a person with less badges is academically better. In addition, Subject 4 wondered if teachers would put in the effort to use such a system. Perhaps many teachers and module coordinators did not have the time to create/assign badges.Some work multiple jobs and teach multiple courses.

Subject 4 afterwards gave good feedback on each section. Firstly for students, Subject 4 would have liked to see an option to dispute a rank or badge. Also, clear instructions on how to collect an uncollected badge would be nice. In addition, in the current rank system, the highest rank is that with the highest number. Subject 4 believed that Rank 1 should be a higher rank than Rank 3.

For teachers, Subject 4 believed that heterogeneity between badges/ranks dilutes meaning and does not preserve coherence between modules. In addition, distributing the badges would cost a lot of manual labour. Subject 4 was worried that it would take too much time and teachers would not find this time spent worth it.

Subject 4 was wondering if, for module coordinators, the fields on the right on the title creation page are already filled in when the page is opened. If so, what information do they display? Is there a title selected by default or do they appear once a title is clicked?

Lastly, Subject 4 was wondering why there was a function to add/remove badge assign permissions. What is the purpose of assigning roles if both TAs and teachers start with the ability to assign/remove badges?

Discussion

Many of the arguments given by Subject 4 were strong arguments and well constructed. It is true that some modules could take better advantage of this badge system than others. Furthermore, Subject 4 is worried that teachers might be biased to those with more badges. However, this is a possible upside of our system. No longer will teachers only look at grades but also other immeasurable qualities such as effort and planning. A combination of badges and grades would be a more ideal way of representing a student.

Disputing a badge will not be an option. It is also not an option to dispute a grade given to you on Canvas. If necessary, the student can approach the teacher. However, since badges are not required to pass, this is mostly a non-issue.

To ensure handing out badges would not take too much effort for module coordinators, the module coordinators can assign teachers to assign badges for them. To ensure handing out badges would not take too much effort for teachers, an import and export function has been added. This results in a .csv file being created so the teachers can assign badges in Excel. Many are used to working with this software and have an easier time assigning badges in it. This was added as a request from other teachers and module coordinators questioned/interviewed.

Badge assign permission will stay an implementation to stop biased TAs from giving their friends badges. We can imagine the Module Coordinator hiring lots of TAs. Not each TA needs to have this functionality. Only those who the module coordinator trusts should be allowed to assign badges.

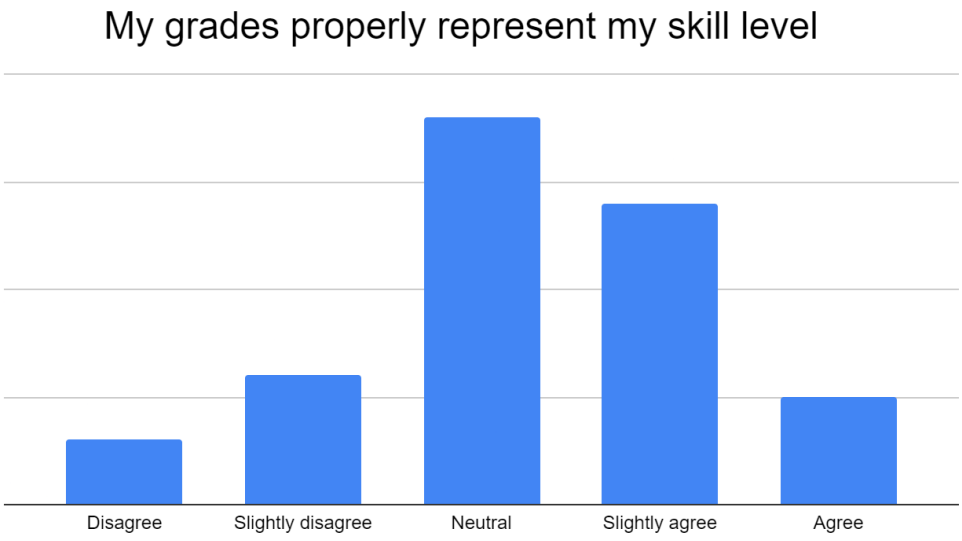
5.3 Student Survey

To challenge the assumptions we have made at the start of the project, we have made a survey and shared it amongst our fellow students. The project is intended to be able to give rewards to those students who put in a lot of effort but do not see results through their grades. This can be because of a number of reasons. Students could perform worse at an exam than during class. This can be caused by personal reasons.

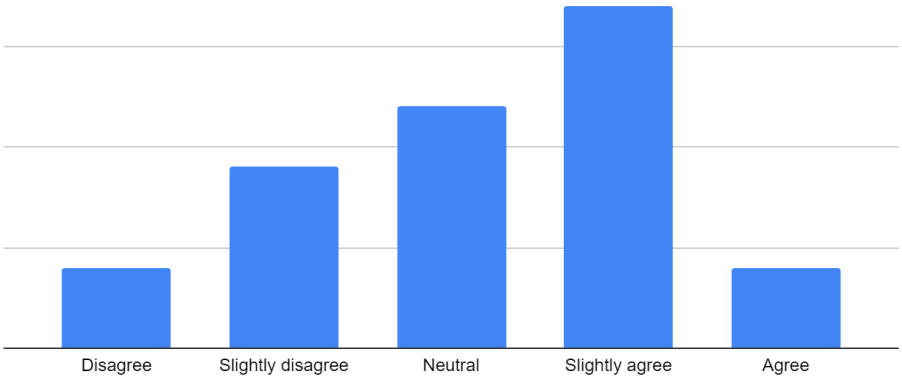
The survey consisted of several statements and each respondent had to give their level of agreement towards the statement from a scale of 1 to 5. The following statements were made in the survey:

- My grades properly represent my skill level.
- My grades reflect my strengths and weaknesses.
- The effort I put into my study properly reflects on my grades.
- I am satisfied with the current grade system.

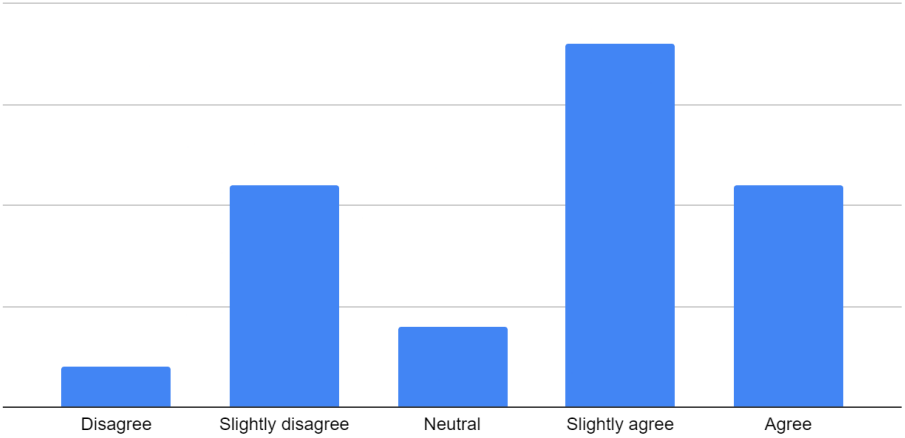
In addition, we asked for the respondents current level of education, to ensure we only took results of students. The results of the survey can be found when clicking on the following link: https://docs.google.com/spreadsheets/d/1-eMUAuelBwIa35mIeM0CxhLIBnRNNs1X-sxa0LdD04M/edit?usp=sharing
The response can be found in the graphs below.



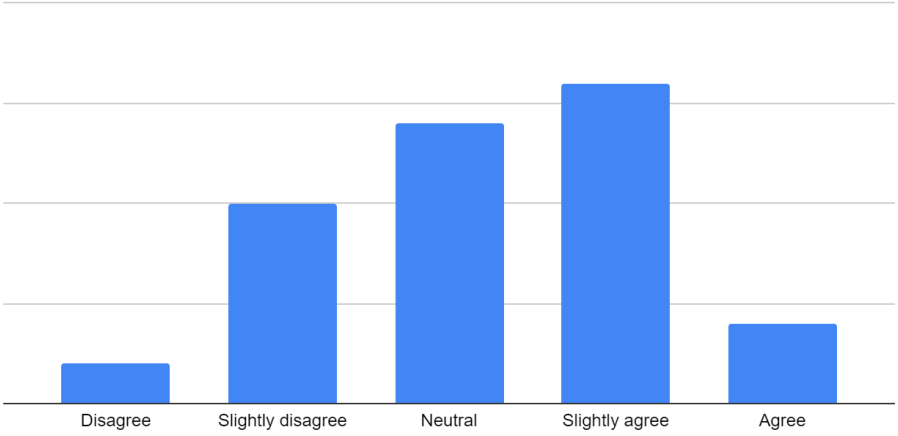
My grades reflect my strengths and weaknesses



The effort I put into my study properly reflects on my grades



I am satisfied with the current grade system



6 Discussion & Conclusion

The design and development of the canvas plugin was discussed in this report. The goals of this project was to develop a gamified portfolio plugin for canvas. This requirement was met.

This project has introduced some of us to the web framework Angular for frontend and Spring for backend development. It allowed us to get more experience in the languages these frameworks use, as well as using the tools that these frameworks provide. By developing in collaboration with a client, in addition to testing with potential users, we were more confident about the usability of the plugin and its ability to solve the issues that it was designed to solve.

The project was very enjoyable to work on. Communication with Yeray was good, and while the project challenged us, it was never an insurmountable challenge. The project itself was interesting for us in our role as students, and we enjoyed working with our teammates. We all understood our responsibilities, and tasks were mostly clear and do-able. Each of our strengths were applied, but we also made sure to do tasks that were outside our comfort zone as a way to learn from the expertise of other teammates. For example, Stijn often worked on the frontend, but worked together with Frans who helped him by explaining how our backend technologies work and how to use them effectively.

6.1 Individual Contribution

Below is a description of the contributions of each team member in general. All members have contributed in every aspect of the project, these are just their main roles in the project.

Stijn Broekhuis : **Frontend Designer and Frontend Programmer.**

As a frontend designer I did most of the design of the frontend.

As a frontend programmer I made the HTML and SCSS for most of the pages, and functioned as support for group members with questions about these technologies. Besides this, I also worked on most of the functionality of many pages in the form of buttons, pop-ups, and information icons.

Frans de Boer : **Backend Programmer, Database Designer.**

As a backend programmer and database designer my main responsibilities were at the setup of the project. Here I made sure everyone had the developer keys, and helped people with setting up the frontend, backend and database. After the setup phase my main tasks were creating the database representation on the backend, and general backend work like creating endpoints, and creating services to make other tasks easier.

Matthias Sleurink : **Backend Programmer, Group Lead.**
As a backend programmer I mostly worked on backend programming, where I created the system to upload badge assignments as a spreadsheet, as well as the system that handles title assignment, and a lot of other minor things.
As the group lead I made sure that tasks were being done. I planned meetings within the group, as well as meetings for the group with our project manager Yeray. I also kept track of deadlines during the project.
Geronimo Wolf : **Designer, Programmer, Presenter.**
As a prototype designer I worked on designing the hi-fi prototype.
As a frontend programmer I created the Teacher page and worked on the implementation of teacher privilege.
As a backend programmer I mostly worked on the REST API.
Lastly, I created the poster for the group and prepared and presented the several presentations.
Adarsh Denga :**Prototype Designer, Frontend Programmer**
As a prototype designer I worked on designing both the rudimentary frontend design as well as the hi-fi prototype.
As a frontend designer I helped with the design and development of the student page as well as minor contributions to the other pages in the project.

6.2 Possible Improvements

6.2.1 Planning of Tasks

At the start of the project a trello board was made to track tasks, after only a couple weeks this was found to be too stiff, and a more ad-hoc system of editable messages in the chat environment “Discord” was used. We tracked tasks per week in a message that was then either edited or replied to to mark progress in tasks. This was easier in the short term, but in the long term this causes issues like: Who did what? Was this task done already? When is this task I’m supposed to do done? Longer term task planning was also very difficult to do correctly with this ad-hoc system.

6.2.2 More Rigorous Requirements

During the project a host of requirements were made, but only very little had been set in stone. This made it difficult to know when a feature was really done and what features were really important. For future projects and assignments it is key for our group to focus more on this essential element of project management, so that the project can be managed more efficiently and effectively.

6.2.3 Documentation of Meetings

Just like the planning of tasks most of the weekly meetings we had during this project were documented only in short small messages on our communication platform. Messages like with the bare minimum in information were shared as “good enough” documentation at that day, but often provided too small or not precise enough to provide enough information in the longer term. More effective documentation of information shared during these meetings would have provided the team with better grips to hold onto regarding decisions taken and reflection.

6.2.4 Rigorous Automated Testing

Most software projects that intend to run for the long term, as we do, need some level of automated testing, ranging from testing small methods in code to testing the integration and building of the code into a working binary. The project does not include any of this. We have done significant manual testing, but there are no automated unit tests, integration tests, or other tests to check the validity of the code.

7 User Manual

7.1 General Information

This project serves as a personal portfolio for students to track their progress throughout their study. Module coordinators, together with an artist, create badges and titles. These can be handed out by teachers and selected TAs to students. Students are enthused by getting a badge and title, feel congratulated, and are influenced to work hard and to do their best.

7.1.1 System Overview

The Interactive Gamified Portfolio is a Canvas plugin that can be activated in every module at the University of Twente. Its operational status is still under development. Interactive Gamified Portfolio is meant to be used on a computer.

7.1.2 Organisation of the Manual

The user's manual consists of six sections; General Information, Getting Started, Students, Teachers, Module Coordinators, and Administrators.

The General Information section explains in general terms the system and the purpose for which it is intended. It also explains the concepts of 'Badges' and 'Titles' which should be known for all system users.

The Getting Started section explains how to get the Interactive Gamified Portfolio plugin and activate it for a module.

The Students section provides a general overview of the system from a student perspective. This section outlines the functionalities available for students in the system's current state.

The Teachers section provides a general overview of the system from a teacher perspective. This section outlines the functionalities available for teachers in the system's current state.

The Module Coordinator provides a general overview of the system from a module coordinator perspective. This section outlines the functionalities available for module coordinators in the system's current state. A module coordinator also has the functionalities of a teacher available.

The Administrators section provides a general overview of the system from an administrator perspective. This section outlines the functionalities available for administrators in the system's current state. An administrator also has the functionalities of a module coordinator available.

7.1.3 Badges

Badges are created by the module coordinator or an administrator in collaboration with an artist to create the badge images.

A list of uncollected badges per module is available for the students. The badge contains an uncollected image that students will see when they have not collected the badge yet. After collecting a badge at a specific rank, the student will see the collected rank's image.

A badge can have as many ranks as desired. Each rank has its own specific image to differentiate itself from other ranks. Each rank also has a description that the teachers can see when handing out a badge and the students can see after obtaining the badge. Ranks are of ascending order; a higher rank number means it is a higher rank.

7.1.4 Titles

Titles are created by the module coordinator or an administrator. Each title has its own colour.

Specific badges are required to obtain a title, including the required rank. A title can be obtained by the student when the student received all the badges at the rank required for that title.

7.2 Getting Started

This section explains the steps that have to be taken to get the files running. A high level of developer expertise is required to take these steps.

7.2.1 Files

Step one is to obtain the project files, which can be done in the following two ways. The first way is to use the .rar file that was handed in at the end of the Design Project. If this .rar file is no longer available, all files can be pulled from git:
https://git.snt.utwente.nl/s2191202/interactive-gamified-portfolio (backend)
https://git.snt.utwente.nl/s2191202/portfolio-frontend (frontend)
For permission to access these repositories, email one of the developers.

If the unchanged files from the .rar file are being used, only two steps of the section backend parameters have to be followed (setting up the database, and setting up the developer keys). From here the application can be run, or built and then run.

7.2.2 Backend Parameters

After all files have been obtained, the backend will have to be supplied with the correct parameters to be run. In the root directory of the project create a directory called **config**. In this directory create a file called `application.properties`. This properties file will have to be filled with parameters for the backend to run. The following parameters will be the same for everyone, and can be copied and pasted into the file

```
spring.security.oauth2.client.registration.canvas.authorization-grant-type=implicit
spring.security.oauth2.client.registration.canvas.scope=openid
spring.security.oauth2.client.registration.canvas.redirect-uri={baseUrl}/lti/login
spring.security.oauth2.client.provider.canvas.authorization-uri=https://utwente-dev.in
structure.com/api/lti/authorize_redirect
spring.security.oauth2.client.provider.canvas.token-uri=https://utwente-dev.instructur
e.com/login/oauth2/token
spring.security.oauth2.client.provider.canvas.jwk-set-uri=https://utwente-dev.instruct
ure.com/api/lti/security/jwks
spring.security.oauth2.client.provider.canvas.user-name-attribute=sub
spring.profiles.active=ssl
```

Up next the parameters for the database connection have to be added. Here the URL, USERNAME, and PASSWORD are placeholders, and have to be filled in depending on the database used. As an example we used the following for a database hosted locally

```
spring.datasource.url=URL
spring.datasource.username=NAME
spring.datasource.password=PASSWORD
```

As an example, we used the following url for a database hosted locally:

```
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/portfolio
```

And the following for a database hosted on a machine on a different network (ip address has been removed):

```
spring.datasource.url=jdbc:mysql://{IP_ADDRESS}:3306/portfolio
```

For our development testing we also added in the following three parameters

```
spring.servlet.multipart.max-file-size=-1
spring.datasource.initialization-mode=always
spring.jpa.hibernate.ddl-auto=update
```

In a production environment these values should be changed. `Max-file-size` -1 removes any limit on file size, and can make the server an easy target for attacks. Initialization-mode is the mode that the database gets initialized in on project startup. On always the database is purged on every startup, and data from the data.sql file (located in `src/main/resources`) is loaded into it. Finally ddl-auto is what gets done to the database structure on every startup, and on update this

will update the structure depending on any changes made (other options are `create`, `create-drop`, and `validate`. The parameter can also be removed).

Finally the canvas client id and secret keys will have to be added to the parameters.

```
spring.security.oauth2.client.registration.canvas.client-id=CLIENT_ID
spring.security.oauth2.client.registration.canvas.client-secret=CLIENT_SECRET
```

Either request these from one of the University of Twente Canvas maintainers/admins, or send us an email for our developer keys.

7.2.3 Building Frontend

Because the frontend and backend are two different projects. To make the backend return the correct frontend pages on a request, the frontend will have to be built into the backend. When no changes have been made to the frontend this has already been done, and no work is required here. If the frontend has been changed, the following command will have to be run in the frontend: `ng build --output-path /path/to/project/src/main/resources/public` (linux/mac). On windows the command is as follows: `npm run ng build ----output-path=\path\to\project\src\main\resources\public --watch` If the `ng` command is unavailable, Angular will have to be installed. Follow the instructions from Angular on how to do this: https://angular.io/guide/setup-local (npm will be needed as well).

7.2.4 Running

After all these steps have been completed the project is ready to be run. This can be done in two ways. Either open the project in IntelliJ (Ultimate edition might be required), go into the **.run** directory and open the `Lti13Application.run.xml` file. Intellij should display a message at the top of the file asking the user to open it in the run/debug configurations. Clicking on this opens the configuration in the start menu and allows the user to start.
The second way to run the project is to open a terminal, go to the location of the backend, and run the command `mvn spring-boot:run`**.** Maven will have to be installed for this

7.2.5 Building the Backend

To obtain an executable `.jar` file of the backend, the following steps have to be completed. First, copy all the parameters from `config/application.properties` into `src/main/resources/application.properties` (make sure the parameters that are already in there do not get overwritten). Next execute the following command in the root directory `mvn clean package`, this will clean up the currently `build.jar`, and create a new one. Finally copy the entire `config` directory into the newly created `target` directory. This makes sure the ssl keys get used in the .jar, and the plugin can use https.

7.3 Students

Students can track their progress in their Gamified Portfolio throughout their whole study. Each module for which the plugin is active, their portfolio will be the same.

7.3.1 Modules

When opening the plugin on Canvas, the student will see an overview of the modules he has attended with the plugin active. Modules for which the plugin is not active will not show up. Clicking on a module results in an overview of all the badges that can be (or already have been) acquired.

7.4 Teachers

Teachers have the ability to assign badges to students, and by virtue of assigning badges can also assign titles. The teacher role is shared by teachers and TA's.

7.4.1 Badge Ranks

Each badge has multiple ranks. Each rank has its own specific image to differentiate itself from other ranks. Each rank also contains a description. Ranks are ordered the same way every time they are shown, this can be used for an internal grading, but does not have to be used this way.

7.4.2 Assigning Badges

When assigning a badge on a computer, hovering over a specific rank with your cursor results in that rank's description being shown on the cursor. A teacher can only assign badges with the permission of a module coordinator.

7.4.3 Import and Export

Teachers can import and export files in csv format. Clicking the export button results in a template for the teachers that can be filled in and afterwards imported. The system will then process the file and automatically assign the badges depending on the content of the csv file.

In the csv file the user sees rows of students, where the columns are badges. To add or remove a badge rank from a student, the user can add, replace, or remove a number in the corresponding cell. When a user adds a faulty badge level the system ignores it. The number entered in a field corresponds to the badge rank that the system automatically processes.

7.4.4 Guidelines

All teachers should have a similar mindset when assigning badges. This is to ensure a proper distribution of badges.

Badges are not meant to be a replacement for the grading system, and are in no way required to pass. They are simply an addition to the current grading system. Therefore, a sufficient mark should not be a requirement for receiving badges.

Instead, those students who the teachers or teaching assistants deem worthy of a reward should receive a badge. For instance, students who communicate well with their teammates or students who put in extra effort. These factors are not always directly realised in their grades.

7.5 Module Coordinators

A module coordinator is allowed to add and edit titles and badges. They are also able to do everything a teacher can.

7.5.1 Creating and Editing Badges

New badges can be added in the “Badges” screen. In this screen, it is also possible to alter existing badges. New badges can be added by clicking on the circular + on the left hand side.

Selecting a badge on the left hand side of the screen allows for module coordinators to edit the current title, badge ranks descriptions, and badge rank images.

Clicking on the “Save” button saves the changes made to the badge currently selected.
Clicking on the “Delete” button deletes the badge currently selected.

7.5.2 Assigning Badges

Module coordinators have the ability to assign badges, similar to teachers. A description of badge assignment can be found in section *7.4.2* up to *7.4.4*.

7.5.3 Creating and Editing Titles

A new title can be created on the Titles page using the ‘+’ icon at the bottom of the left-hand side of the screen. An existing title can be edited as well by selecting it and editing the fields on the right-hand side of the screen. The fields that can be edited include the name of the title, the description, the color of the title, and lastly the list of badges that are required to obtain that title.

When a new title is made, it can be saved using the ‘Save’ button at the bottom of the right-hand side of the screen. Similarly, an existing title can be deleted or saved using the ‘Delete’ or ‘Save’ buttons at the same location.

7.5.4 Linking Badges to Titles

Badges are linked to titles by use of the required badges section at the bottom of the right-hand side of the Titles page. A requisite badge can be added using the small ‘+’ icon located in this area, which will then show a modal where a badge can be selected.

A badge can be selected with a rank that is required and then saved using the same save button mentioned in the section above.

7.5.5 Allowing a Teacher to Assign Badges

The Teachers page can allow module coordinators to decide which teachers have the ability to assign badges to students. All the people with the teacher role are displayed in a table-like format along with a button that can be toggled to allow them to assign badges to students or prohibit them from doing so.

7.6 Administrators

An administrator is allowed to assign the administrator role to teachers, as well as a teacher role to students (like TA’s and the like). They are also able to do everything module coordinators and teachers can.

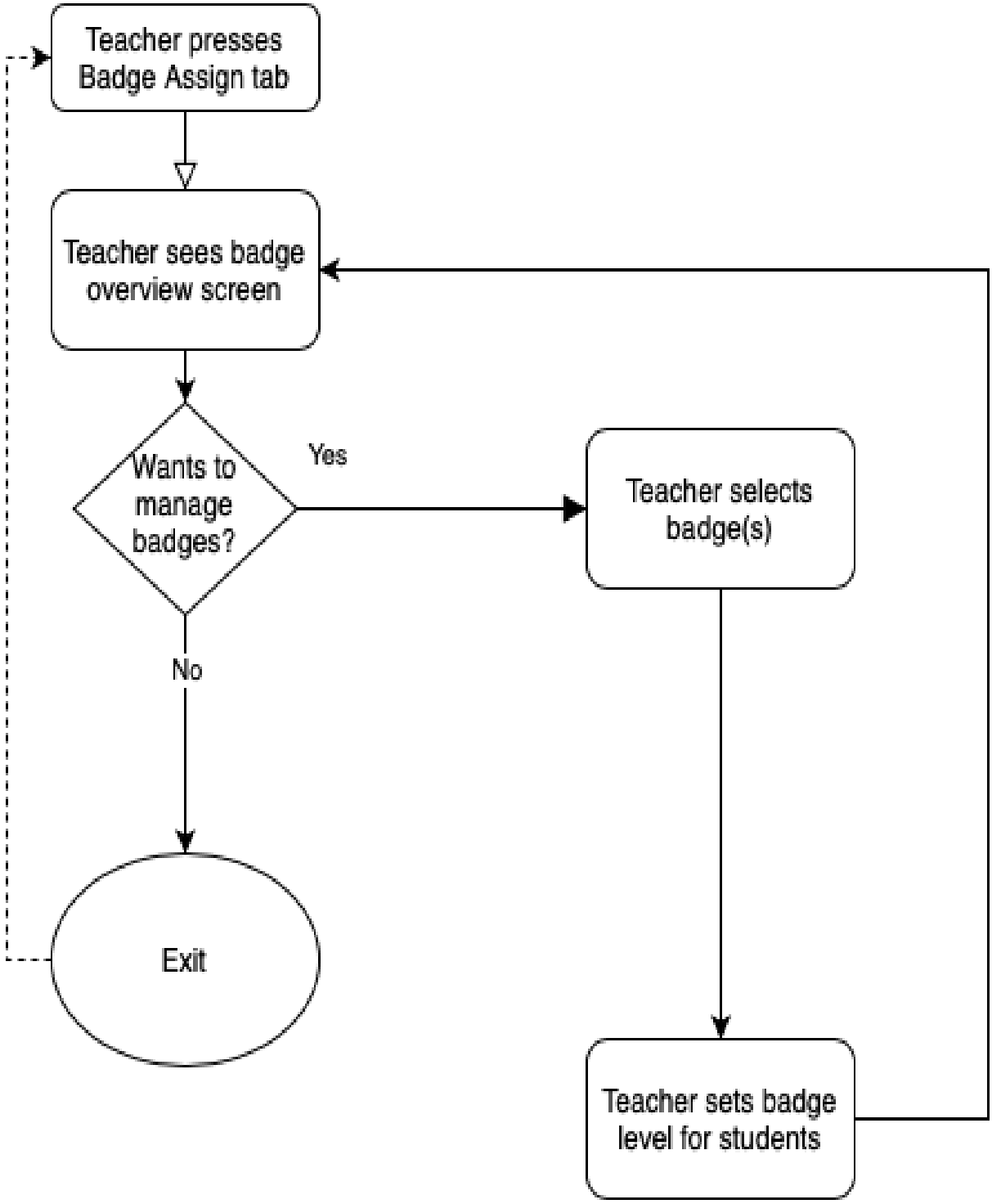
8 Bibliography

Framer. (2021, 01 01). *Framer*. Framer. https://www.framer.com/

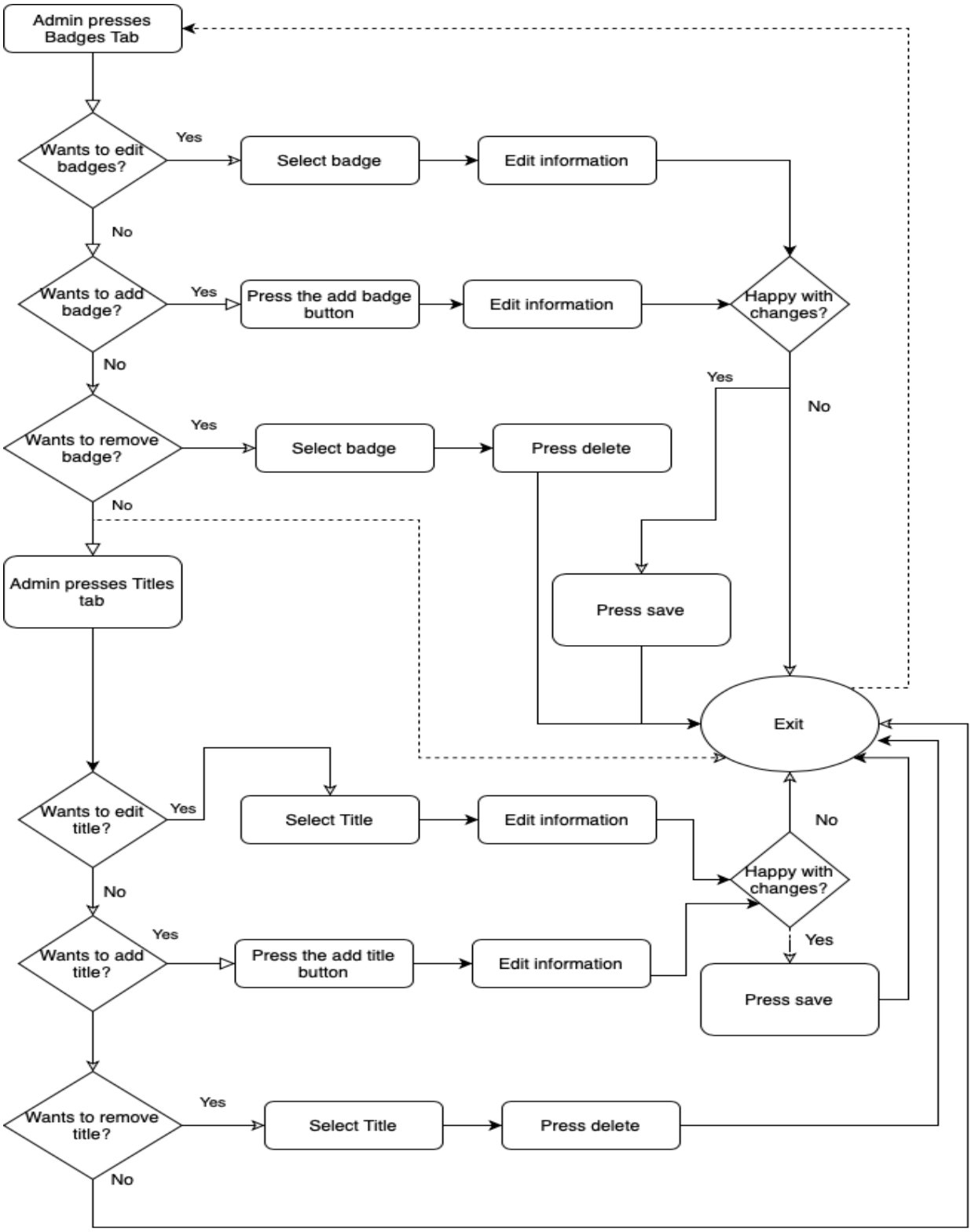
Instructure. (n.d.). *Styleguide*. https://canvas.instructure.com/styleguide#

9 Appendix A

Teacher Action Diagram



Admin Action diagram



Student Action Diagram