FACULDADE DE CIÊNCIAS E TECNOLOGIA DA UNIVERSIDADE NOVA DE LISBOA
Mestrado Integrado em Engenharia Informática
Internet Applications Design and Implementation (2020/21)

Grant Management System

Project Report

Henrique Raposo (57059) ; César Nero (58659); Ihor Yushchak - 53176

1. Introduction

In this project we were tasked to develop a server-side application with Spring Boot framework using Kotlin as code base and the client-side web application with React that belongs to a JavaScript library.

This assignment has been divided into 4 phases, that would be server-side specification using OpenAPI/Swagger, server-side application with persistent data storage and a RESTful API, client-side specification, that would consist of IFML diagrams and the corresponding mockups diagrams to implement and for final, client-side, that would be full complete system, corresponding to full web application with frontend in JS with React librerie.

All implemented security was based on stateless authentication using JSON Web Token (JWT) and role based access control by server-side.

Our application consisted to implement a grant management system. This environment would consist of several entities and relationships between them. There would be sponsor entities to create grant calls, for students to submit applications to those grant calls, that would be evaluated based on a set of reviews made by reviewers entities, forming an evaluation panel for the one grant.

2. First Phase

2.1. User Stories

The following list is the list of user stories that we considered for the development of the project:

1. As a student, I want to access the home page and see the list of available grant calls, so that I can select one open grant call.
2. As a student, I want to see all open grant calls, so that I can create a new grant application, and I see the fields required to fill their new application.
3. As a student, I want to access the list of my grant applications, and see the newly created application in the list of grant applications.
4. As a student, I want to fill all information required to submit to a grant call, so that it can be considered for funding, and I see my application in the list of submitted applications.
5. As a student, I want to list my current submissions, so that I can submit them before the deadline.
6. As a student, I want to list my evaluated submissions, so that I can read the reviews and classification.
7. As a reviewer, I want to list all the grant applications assigned to the panels I belong to so that I read the submission's details and the details of the corresponding students.
8. As a reviewer, I want to list all the grant applications assigned to the panels I belong to so that I can read all the available reviews.

9. As a reviewer, I want to list all the applications assigned to the panels I belong to so that I can classify an application and write a review.
10. As the chair of a panel, I want to see the list of all grant applications assigned to panels I lead to so that I can read the details, classifications and reviews of one application.
11. As the chair of a panel, I want to see the list of all applications assigned to panels I lead to so that I can write the final evaluation and assign the final classification.
12. As a anonymous user, I want to see the homepage, so that I can see the list of open grant calls, and the total number of submitted applications.
13. As a anonymous user, I want to see the homepage, so that I can see the list of grant calls, and their status and their opening and closing dates.
14. As a anonymous user, I want to see the list of grant calls, so that I can select a closed call and see the list of funded applications.
15. As an anonymous user, I want to see the homepage, so that I can sign in as a student.
16. As an anonymous user, I want to see the homepage, so that I can sign in as a reviewer.

2.2. Data Model

According to these specifications the following data model was drawn:

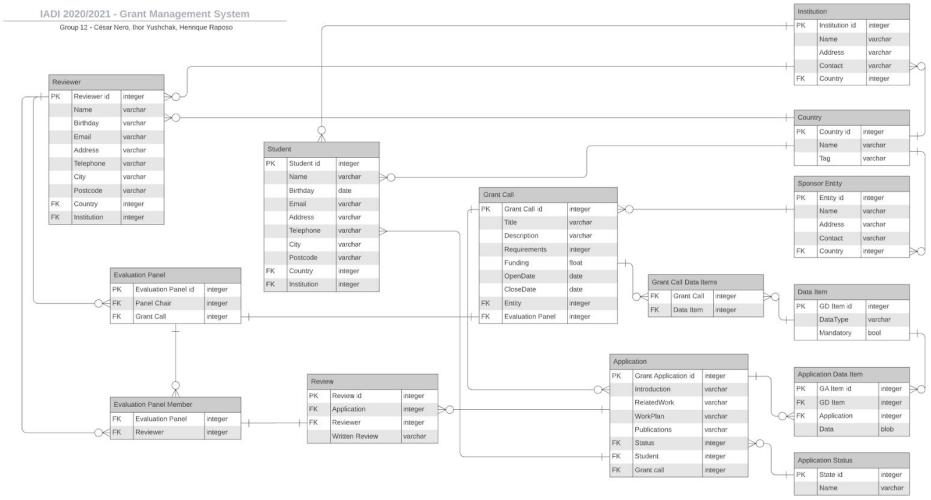

Figure 1: Data Model

2.3. Swagger

Swagger was implemented according to the previous data model and specification. It can be found in the pdf format in the folder.

3. Second Phase

3.1. Server Side

3.1.1 Overview

The project went accordingly with smooth implementation, almost all of the endpoints were implemented at least at the best of our understanding without having the full inclosure of what the front-end needs. Controllers are responsible for receiving DTO (Data Transfer Objects), executing specific service methods and passing a DAO (Data Access Object) that was constructed from the DTO received. Some of the endpoints are secured using both roles and principal methods.

The service layer is then responsible for making some crucial validations and throw errors when needed. They execute methods on the repositories. The repository layer consists of the repositories that are responsible to execute queries and retrieve or save data to the database layer. Bitbucket pipeline was used so our JUnit tests run and get verified. More tests and examples were made using Postman.

3.1.2 Focus features

- Endpoints needed with full discrimination on API Operation and Response;
- Security using roles and principal authority validation;
- Multiple tests on controllers, services and repositories. On execution and security;
- Custom JPQL queries for data retrieve;
- Postman tests and examples.

3.1.3 Not implemented but desired

- Service level security (instead of controller);
- Full integration test.

4. Third Phase

4.1. Concept Models

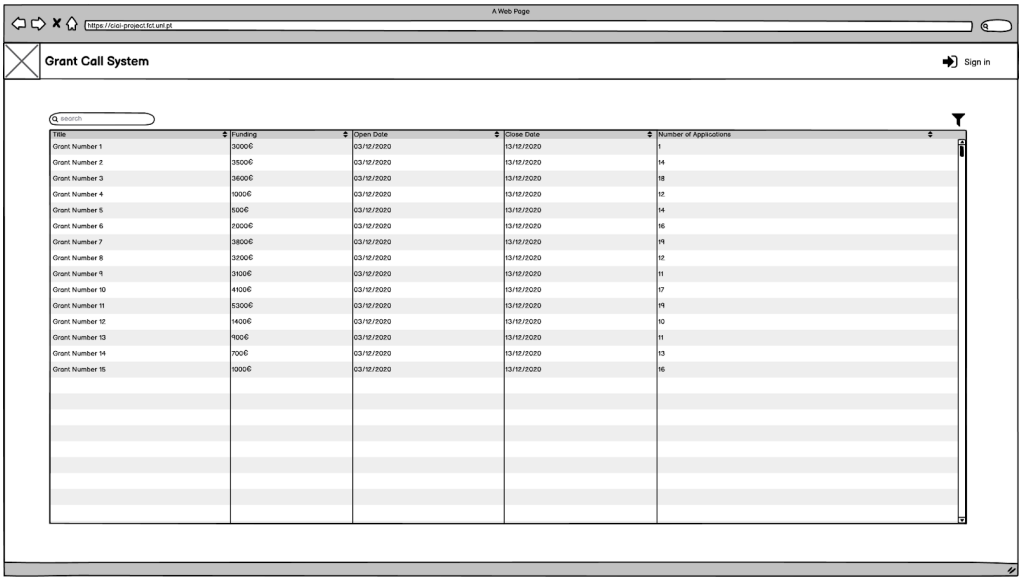

Figure 2: Homepage Mockup

In the list of grant calls is also possible to see right away the number of applications for each grant call. The process of selection is simplified using the filter component

shown in Figure 2. This allows the user to quickly find the grant he's looking for like open/closed grants or funding value. The search bar will also allow the user to quickly find something he's looking for in any columns. After selecting a grant call, the details of that grant call is displayed like shown in Figure 2.

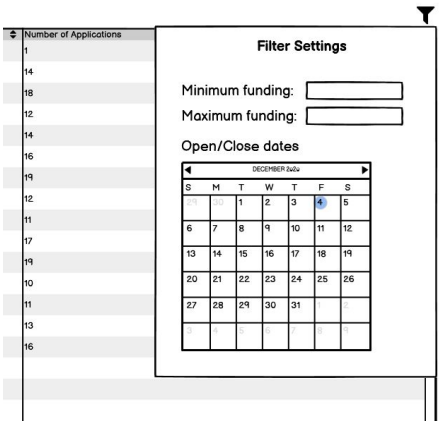

Figure 3: Filter component Mockup

After the user found the desired grant, it is possible to select it and check the full details like shown in Figure 3.

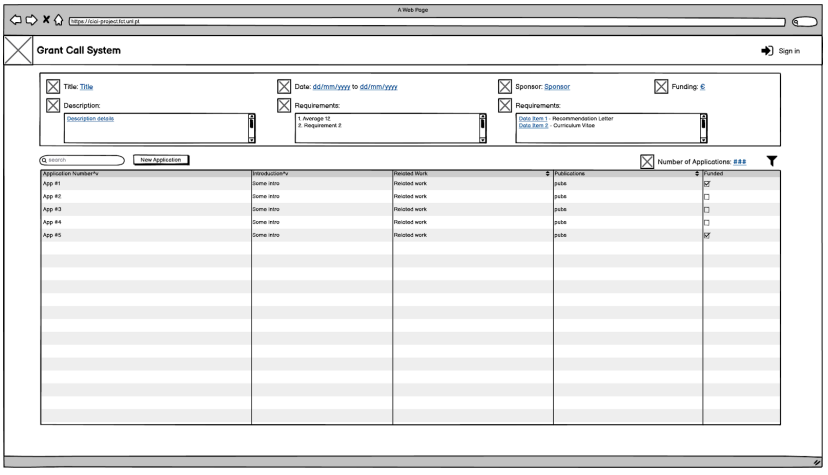

Figure 4: Grant Details Mockup

At Figure 3 is possible to see the full details of a grant call (such as description, open and close dates, funding value and the requirements for application). Here is also possible to see the applications for such grant.

The procedure described in the figures 1,2 and 3 is represented in IFML as shown in Figure 4. Once the user selects a grant from GrantList a navigation event is produced, and as a result, the full details of the selected grant are displayed.

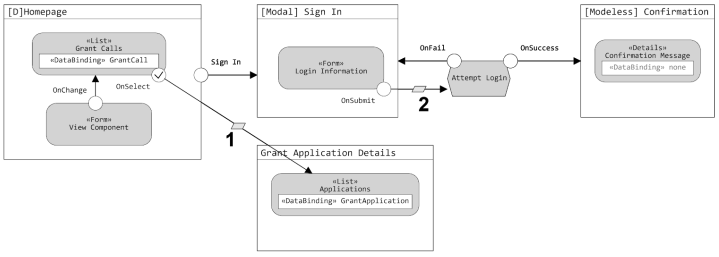

Figure 5: IFML model corresponding to the display of the details of grant TODO:REFERENCE

Is also possible to see some user interaction with Login Form. This is a modal window where the user can write his credentials shown in the Figure 6.

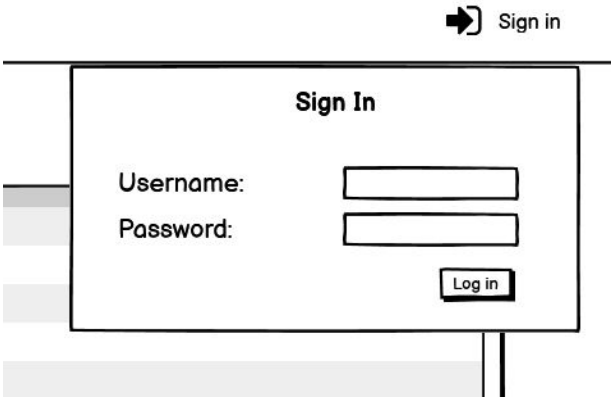

Figure 6: Login Modal Window

After submit the form the system tries to authenticate the user and assign a role for that credentials received. In case of success a Modeless message, shown in the Figure 7, is displayed informing the user that hes now logged in.

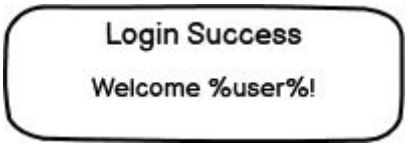

Figure 7: Modeless logged in message

After the user is authenticated, some pages adapt to the role displaying more information or options than normally is shown, for example, the majority of the pages shown on the right top corner the user role and name like shown in Figure 8.

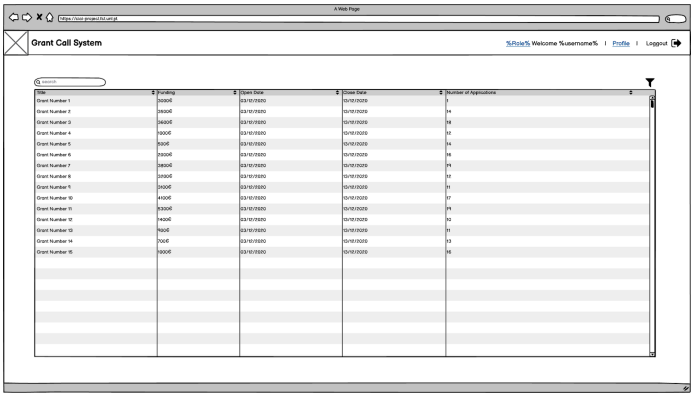

Figure 8: Homepage with authenticated user

In case of the authenticated user being a student, after select a open grant call and accessing the details, is possible to see a create button for a new Grant Application like shown in Figure 9.

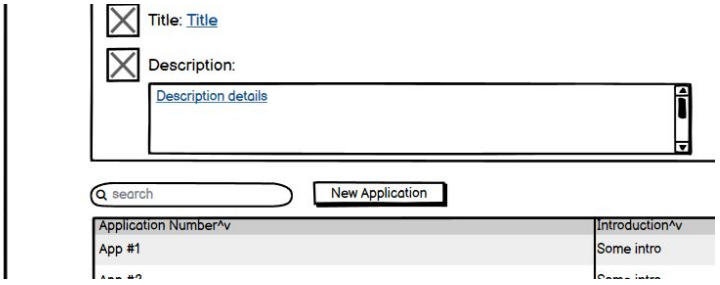

Figure 9: New Application button at Grant Details with authenticated role of Student

After the New Application button is pressed the user is presented with a modal with a form containing the required fields.

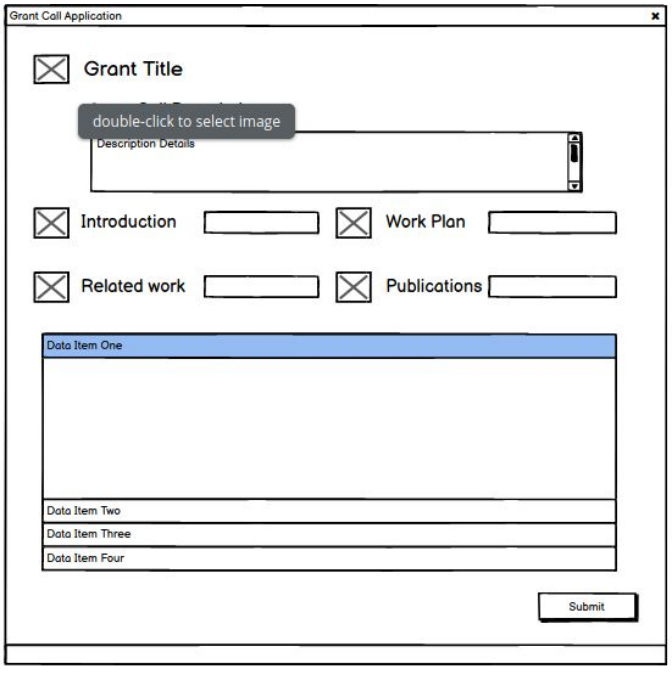

Figure 10: Modal to create Grant Call Application

when the user has filled the required fields he can submit the Submit button to submit the application , this will then show a confirmation or error message, if it was not possible to submit. This message will show in a modal form as described in the next images.

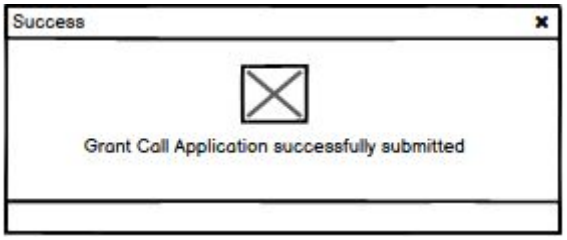

Figure 11: Success message displayed when the application is successfully submitted

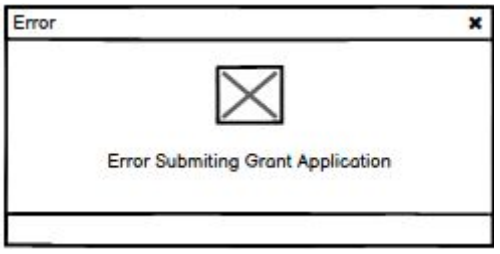

Figure 12: Error message displayed when an error occurs in the submission

If the user does not submit and instead closes this form , it will later be accessible through the Student Applications page, available under the 'My profile' dropdown menu on the header.

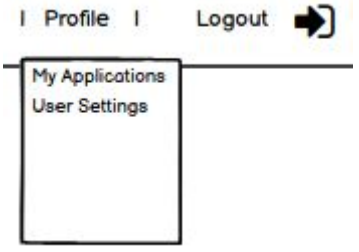

Figure 13: my profile dropdown menu for a student account

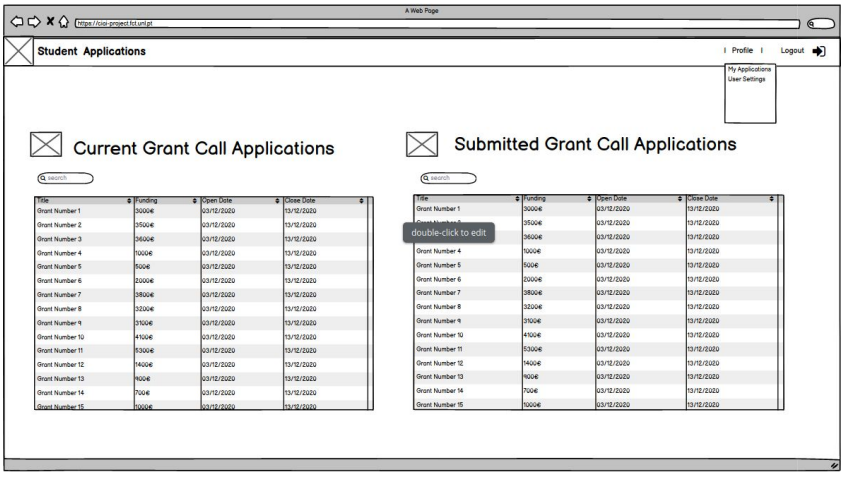

Figure 14: student applications page

In this page the Student can see his current applications and by selecting one of them , the Grant Application form will appear, in the form of a modal as before, and the student can now resume filling the required items and submit.
Once the application is submitted it will appear on the 'Submitted Grant Call Applications' list . If the user selects one of these a new modal with appear with the grant call details , the submitted items and the evaluations divided by reviewer name. The bottom will also show the status of this application

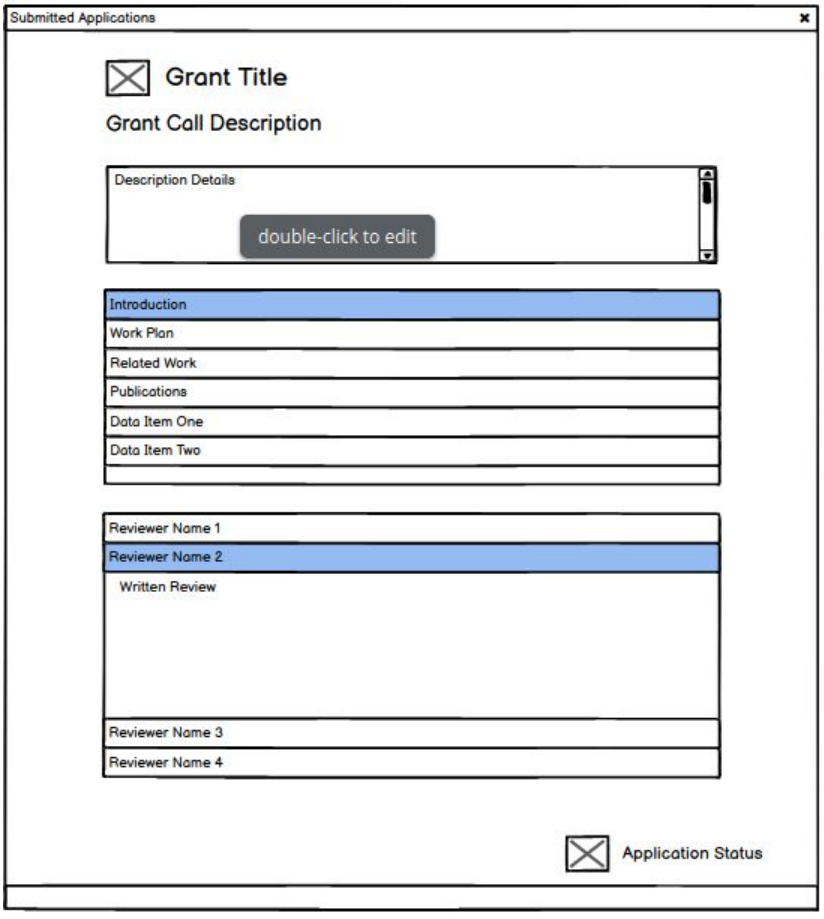

Figure 15: submitted grant applications modal

These procedures corresponding to the user as a student can also be described with the following IFML:

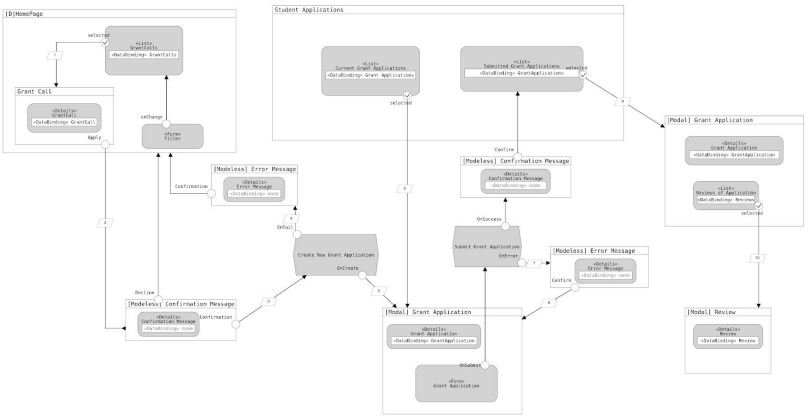

Figure 16: IFML model representing student application progress

TODO REFERENCE

After the user is logged in and authenticated as a reviewer (process shown in Figure 4), is possible to access new options located at the top navigation bar like shown in Figure 16.

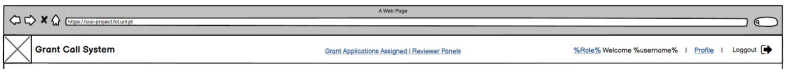

Figure 17: Mockup Topbar Reviewer

Using this menu, the user can access Grant Applications Assigned where can be seen the list of grant applications assigned to the panels that the reviewer belongs to allowing the user to easily search for a specific grant, clicking on one grant will update the list of the right with the corresponding applications of the clicked grant. This page will dynamically update when the user clicks in the row of any data table. Again clicking in one of the available applications will display the corresponding data like application details, data items, reviews and the student information that submitted such application like shown in Figure 17.

Figure 18: Application details mockup

The reviewer can then write a review using the button "Write New Review" where a modal dialog will ask for details about the review shown in Figure 19.

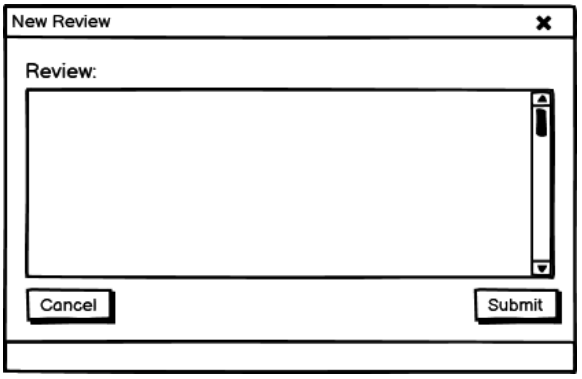

Figure 19: New review modal dialog mockup

Being a chair of a panel the user can access the option in the homepage called 'Chair Panel Review', this leads to a similar page of the written reviews shown in Figure 20.

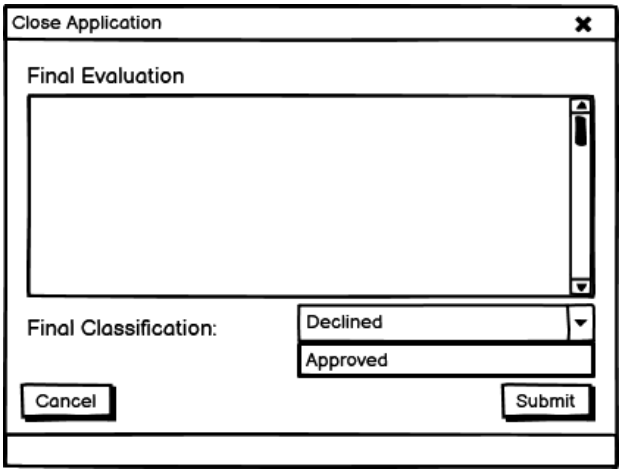

Figure 20: Panel chair write final review and evaluation mockup

This whole process is displayed in the Figure 21, where using XOR operand allows you to define if in-page the reviewer wants to see the grants where he is a panel member or a panel chair. Using some in-page lists the user navigates to the application desired, read the reviews of the other panel members and finally close it leaving a review and final evaluation.

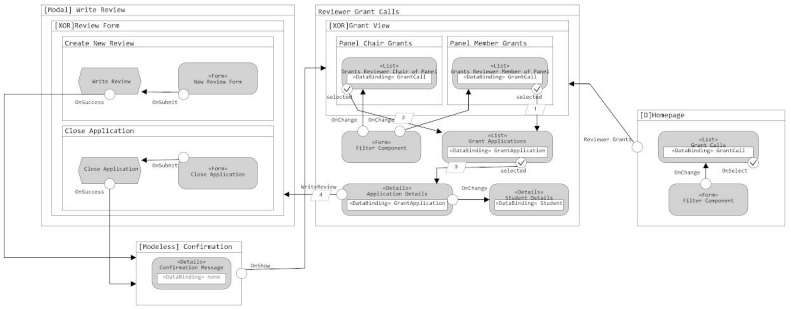

Figure 21: IFML model of reviewer accessing Panel Chair Grants or Panel

5. Fourth Phase

5.1. Final

5.1.1. Student User Stories

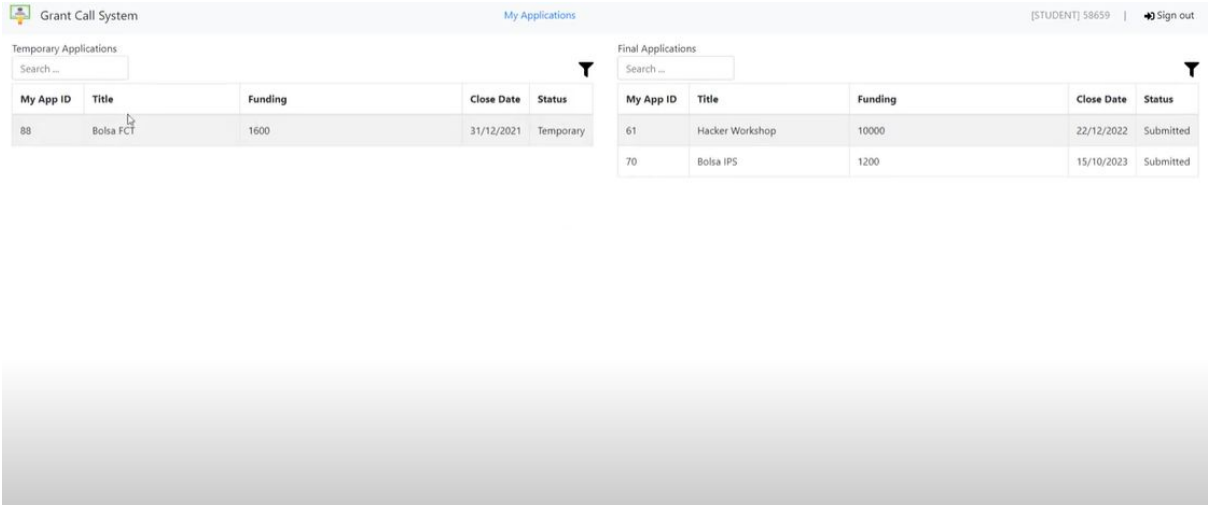

Figure 22: Student Applications page with according applications

In Fig. 22 the student can see 2 tables, first on the left corresponds to all current grant call applications, that student can submit still and on the right table we have all submitted applications, this table would start as empty as the student does not have any submitted applications at start. By searching on the search bar of any of the tables, students can filter applications by title.

Choosing application from current grant call applications on the left table, the following page is presented to student Fig. 23

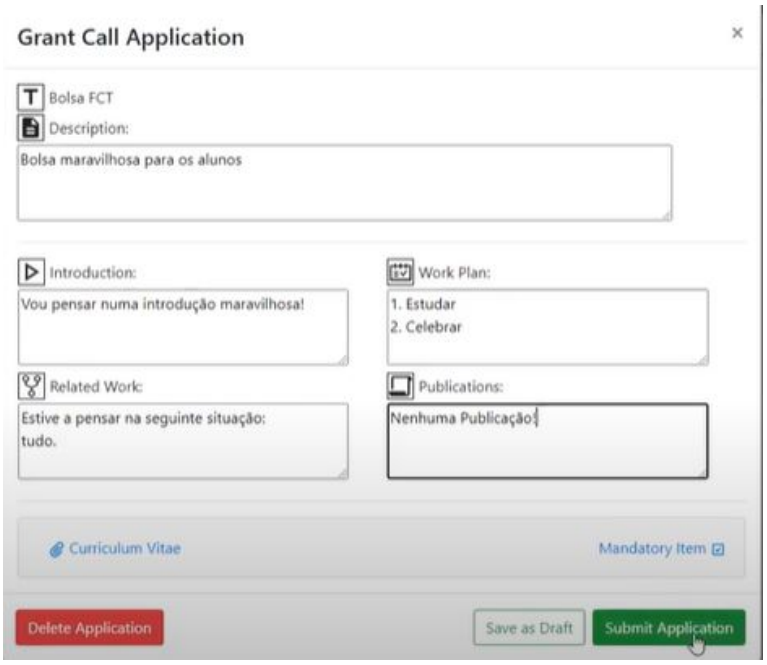

Fig 23: Modal for submitting grant application

This modal presents a grant call application submission with all mandatory items, that after clicking on “Submit Application”, the student submits an application. Submitted application is after presented on the table of submitted applications of the student that is logged in.

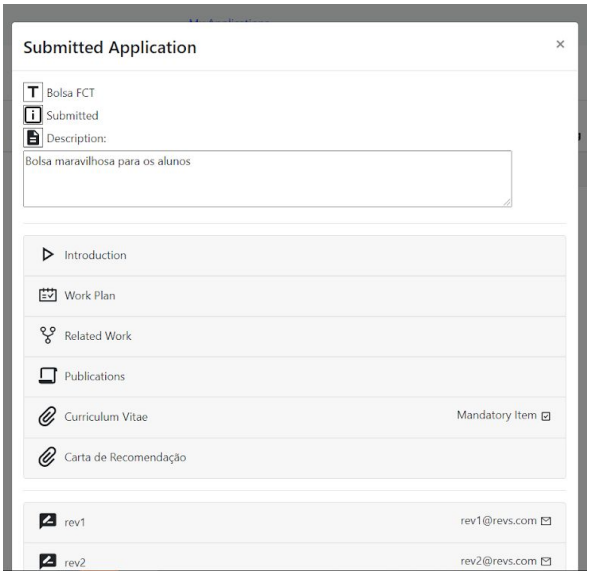

Figure 24: Grant Application Modal Submitted

Choosing one of the applications from the table of submitted applications, students obtain full information about that grant call application, as Fig. 24 shows.

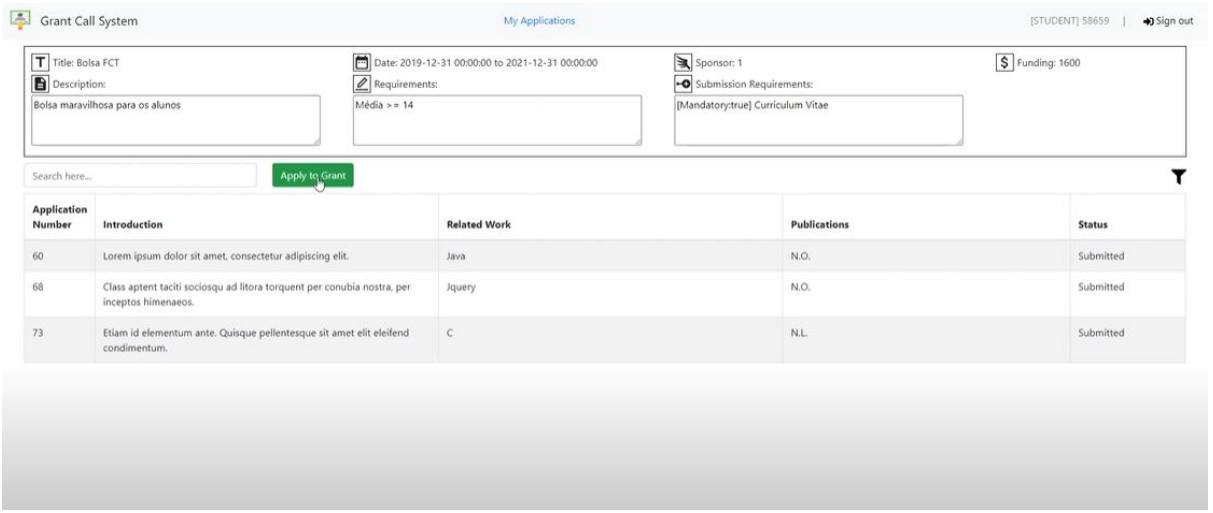

Figure 25: Student Homepage

This page is presented to users with the role “student” after login and the same as on My Application page for students, students can submit their application by clicking on “Apply to Grant”, after which will appear the same modal as on Fig. 23.

5.1.2. Reviewer User Stories

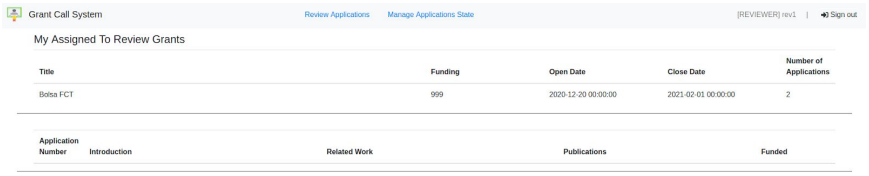

Figure 24: Review Applications page with according grant calls

In this Fig. 24 we can see the Review Applications page for the reviewer , this page will start mostly empty and only display the grants where the reviewer is a member of the evaluation panel.

On pressing in the desired grant call , a list of applications for that grant will be displayed. If the user then clicks in one of the applications the details of that

application will appear (or update) below , these details include the reviews and data items , the student details.

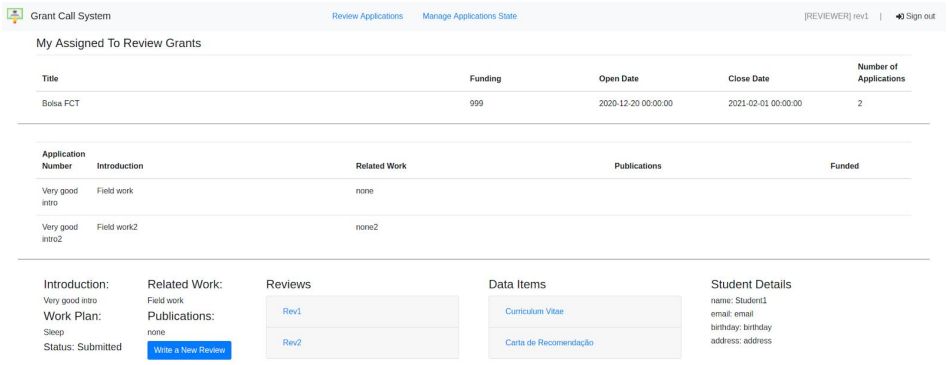

Figure 25: Review Applications page with details

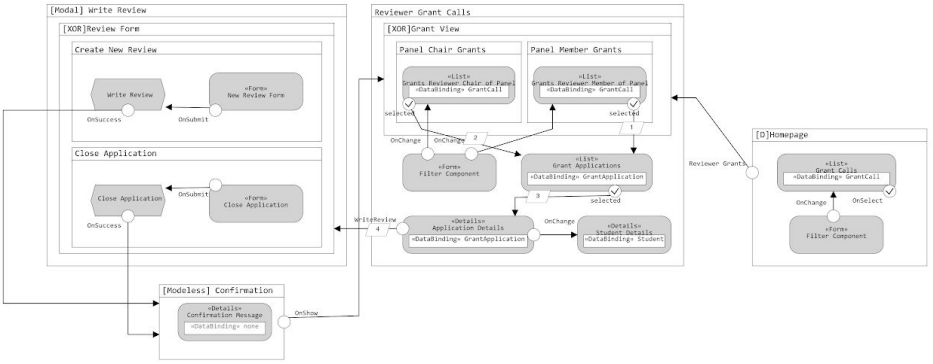

Figure 26: corresponding IFML model

The Reviews and Data items will be displayed with bootstrap's accordion as defined in the mockups and pressing on the name of the reviewer (or in the name of the data item) will display the rest of the information.

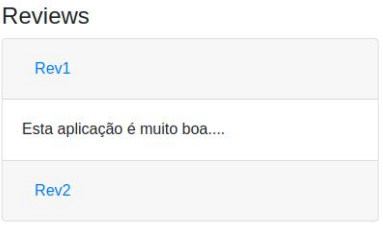

Figure 27: Reviews Accordion

Among the details also appears a button for the reviewer to create a new review. Pressing on this button will show a modal with a form where the user can write and submit a new review.

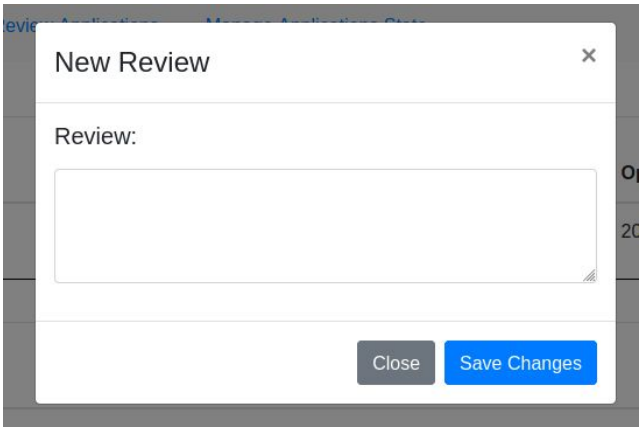

Figure 28: Create Review Modal

5.1.3. Chair User Stories

Pressing on the Manage Application State button in navigation will give us a similar page to the Review Applications page for the reviewer in fact most of the components are recycled from this page , changing only the information displayed.

Here the grant call list shown will only contain the grant calls where the user is set as chair of an evaluation panel.

One of the changes from the reviewer page is the button that gives the ability to write a final review and as chair of the evaluation panel change the status of the application with a dropdown menu.

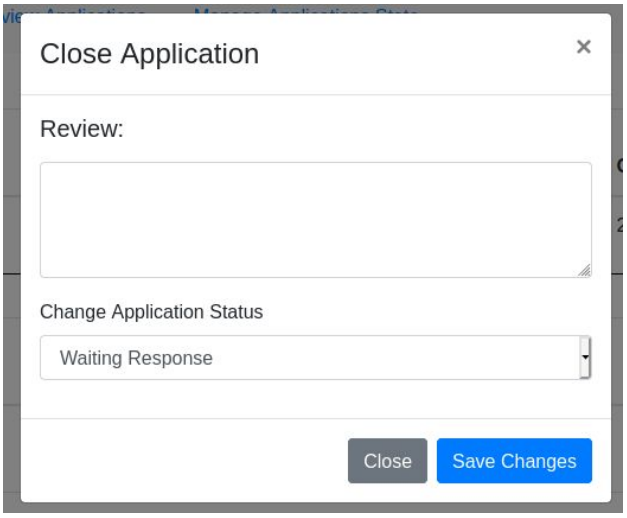

Figure 29: Close review Modal

5.1.4. Anonymous User Stories

Figure 30: Homepage in React

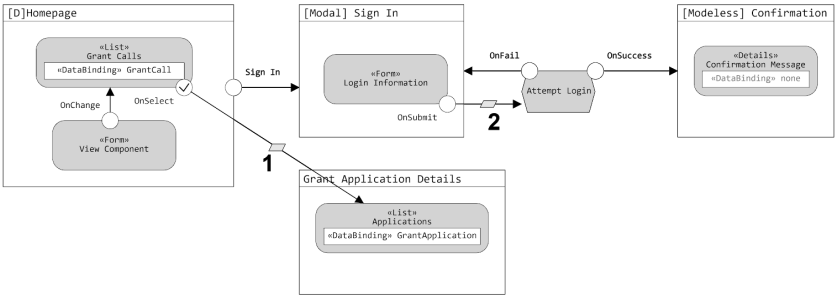

Figure 31: IFML model of the homepage

In Fig. 31 we can see what the homepage looks like, this page will be available to all users and the information displayed doesn't depend on the logged in user.
Here we can see the list grants calls and all the information required. The user can also use the filter to restrict the list of grant calls that show on this list.

You can also access the login modal through the sign in and login to the account type you want , if the login details correspond to it.

5.1.5. Updates

In order to best conclude the assignment more back end methods had to be added to retrieve data in ways that we did not foreseen in the first phase, for this reason the updated swagger documentation can also be found in this folder

There were also some modals shown on the mockups that we didn't implement due to time constraints.

6. Conclusion

In conclusion, our team had a fully working modern Web Api. All user stories were accomplished, as on server-side, as on client-side.

Furthermore, we would like to change web design that because of limited time, didn’t use the best style for pages. We also would have liked to implement a better security and verification on the controller methods and complement that with more verifications in the services.

In the end, we would like to refer, that this assignment was very challenging and interesting, with a lot of modern technologies.