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

This is the project assignment for the Internet Applications Design and Implementation course in 2020/21 edition. Project fulfillment unrolls in two stages. In the first stage of the project, you will develop a service-based server-side application; in the second stage, you will develop the client-side web application. This document will be incrementally updated throughout the semester to address continuing logistic needs.

Version log:
September 30, 2020: Initial version
September 30, 2020: Details of the midterm submission
November 23, 2020: Text refactoring and details of the third and fourth submissions

Important Dates:
October 9, 2020 (23h59m): *ER diagram and API*
October, 30, 2020 (23h59): *Serverside application*
December 4, 2020 (23h59): *IFML specification and Mockups*
December 22, 2020 (23h59): *Full system*

1 Project Assignment

For this semester, you'll have to implement a grant management system. The system allows for sponsor entities to create grant calls, for students to submit applications to those grant calls. The grant of a student may or may not be awarded based on a set of reviews.

When creating grant calls, sponsors provide the basic grant call information such as the grant title, a brief description, the requirements that must be fulfilled by students, the funding being distributed, and a set of data items which all grant applications will have to have. Such attributes of a grant application must be described at least by a datatype and a basic boolean property stating if the attribute is mandatory. The datatype and mandatory attributes can then be used by the client application to correctly shape the user interface to obtain a dynamic amount of information. Examples of a grant applications attributes to be assigned on grant call creation can include some of the following: "Introduction", "Related work", "Work plan and methods", "Publications". Grant calls also have opening and closing dates.

Upon grant creation, sponsors must also define an evaluation panel for the grant call. The evaluation panel comprises a set of reviewers, one of which acts as the panel chair (the leader). A reviewer, in an evaluation panel, is a member of an institution that has access to all applications of the corresponding grant call. A reviewer can only evaluate applications from students of other institutions.

An evaluation panel is in charge of a single grant call, by reviewing and deciding on whether to accept an application or not. Each reviewer writes a review for a given set of assigned applications, and then, based on those reviews, the panel decides to accept or deny an application. Take note that an application's acceptance is decided by the panel chair on behalf of the whole panel. The panel chair may also have written an individual review for some applications.

Students are the ones responsible for the submission of grant applications. In the grant management system, students have access to the list of open grant calls and the details of each call, the information needed to submit an application. Students can only apply to open grant calls, that is, grant calls whose opening date is in the past and the closing date is in the future. When applying to a grant call, students must supply, in their application, all the fields required by the grant's data items. Regarding operations other than grant creation, students can, as mentioned above, list all open grant calls. They can also list their applications, check the panel's decision on the acceptance of their applications, and check the reviews of their applications once the evaluation panel had reached a decision.

As for the details of students and reviewers, both belong to an institution, and it is necessary to keep basic information about them, such as name, email, and address. About institutions and sponsors, the system stores their name and contact.

Students must supply a CV comprising a number of dynamically determined items. These items are also tagged as mandatory or note, have a data type, and can be modified via a specific webservice API.

1.1 User stories

The list of user stories that you should consider for the user centred development phase are the following:

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 a 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 a anonymous user, I want to see the homepage, so that I can sign in as a student.
16. As a anonymous user, I want to see the homepage, so that I can sign in as a reviewer.

2 Technical Details

For this assignment, you will have to implement, in teams of three elements, a complete service-based server-side application with persistent data storage, RESTful API, OpenAPI/Swagger documentation, and a client application, a SPA web application. To accomplish this goal, you should use the technological stack Kotlin/Spring/React/TypeScript.

Note: This document will be continuously updated to include more information, such as submission details, on each of the assignment stages.

2.1 Server-side Specification

The first stage of the project consists in the complete openAPI specification of the API for the project described in section 1, dynamically generated from the corresponding code in Kotlin.

2.2 Server-Side

The second stage of the project consists in a complete service-based server application with persistent data storage, and a RESTful API. Your system should comply with architectural principles studied in the course. You should implement the necessary model-based security policies that you think are necessary in this scenario.

2.3 Client-Side Specification

The third phase of the project consists in the specification of the client-side of the grant application system. Your specification consists of an IFML diagram and the corresponding mockup diagrams to implement.

Use the Bookstore example available in the official IFML website as reference[1], and use the language reference as support [2].

Your specification and implementation should consider the user stories described in section 1. Do not diverge from those user stories, and focus on their correct specification and implementation. You can add more artifacts to your user interface solely for visual purposes.

2.4 Client Side – Complete system

Your final submission consists of the client-side component and the complete working system, covering the user stories listed above, but should implement correctly the security policies related to the resources involved.

[1] `https://www.ifml.org/wp-content/uploads/IFML-Bookstore-Example.pdf`
[2] `https://www.omg.org/spec/IFML/1.0/PDF`

3 Submission Details

The submission is performed via a bitbucket repository, each phase tagged in a different git tag. Your commit should have a date before the deadline.

Phase 1: DataModel, 9th of October, 23h59m
Phase 2: MidTerm, 30th October, 23h59m
Phase 3: IFMLModel, 4th December, 23h59m
Phase 4: Final, 22nd December, 23h59m