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

Important Dates:
October 9, 2020 (23h59m): *ER diagram and API*

1 INTRODUCTION

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.

2 Technical Details

For this assignment, you will have to implement, in teams of three elements, a complete service-based server application with persistent data storage, RESTful API, OpenAPI/Swagger documentation, and a client 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 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
Phase 3: IFMLModel
Phase 4: Final