System for Managing Academic Information

I. The application

The process of managing information required in an academic environment is a time-consuming activity. In order to produce and manage this information, using a dedicated software application is from far the best solution. Mainly it is about classroom activities engaging students and teachers. The System will keep track of information for each student starting with the first undergraduate year and finishing with the last one or with the last master year.

Each faculty may have different specialisations both for the undergraduate and master degrees; a student is enrolled in one of these.

Each faculty year will have a contract of studies, a curriculum (list of mandatory disciplines + 1 optional discipline).

II. Functional requirements

The system should:

- allow its users (students, teachers and faculty administrative staff) to sign in using a unique username and a password
- allow its users to have a profile and upload their personal information to it
- logout of the system

Students should be able to:

- enrol to a year (or at most 2 years) of study to a faculty
- view the curriculum for the year of study, consult the list of optional courses and specify their preferences in decreasing order for the optional disciplines
- get assigned to an optional discipline
- sign the contract of studies (that contains the curriculum) with the faculty at the beginning of the academic year
- view their grades for each discipline

Teachers should be able to:

- propose a list of maximum 2 optional courses for the following academic year

- 1 teacher per faculty will be the chief of the department
    - view & approve the final list of optional courses
    - specifies the maximum number of students for each optional course
    - view the teacher (discipline) with best or worse results obtained
    - view the disciplines given by a teacher in a semester or in an academic year
- add grades for each student enrolled in their class (both mandatory and optional classes)

Faculty administrative staff should be able to:

- ask and print different documents like:
    - the list of students from each group ordered by their professional results
    - the list of students from each year ordered by their professional results or complying with some criteria like the average mark included in an interval, a.s.o.
    - at the end of each semester, view students classified in decreasing order of obtained results. This classification together with the funding level is used to decide studying grants.

The algorithm for optional courses enrolment is explained in the following steps:

- Each teacher (having at least a lecturer degree) proposes a list of maximum 2 optional courses intending to give students in the next academic year.
- The chief of each department approves the final list of optional courses; selects a maximum number of students to attend each course. Usually there are between 2 or 3 groups. The decision is uploaded on the appropriate page of the faculty site.
- Students consult the list of optional courses and specify their preferences in decreasing order for each group.
- Then, the process of assigning optional courses to students may start. In order to be given to students, a course must have at least 20 potential followers. The courses with less potential followers will not be organised. By consequence, for the students choosing these courses the next preferred course will be assigned. After, each assignment need to be validated by the criteria above mentioned.

III. The implementation

As the SE course proposes an object-oriented approach, it is required that both the domain model and the solution model to be object oriented. There are mandatory requirements regarding the artifacts produced during the software development life cycle like:

- Use Case diagrams and documents, describing the functional model
- Class diagrams for the analysis and design model
- The application will be multilayer structured: User Interface, Business Logic and Database. Object orientation concerns mainly the Business Logic
- By its nature, this is a web application. So, it is recommended that UI provides browser forms specific for each user. The front end functionality will be responsible to gathering data, transmission, and appropriate king of validations. The database and the business logic will be resident on the server.
- A particular attention will be given to the validation process by using assertions (pre & post-conditions, invariants) also to the testing phase, including the techniques of selecting (generating) tests data.
- Also the usage of design patterns in the late design model will be appreciated. The same attitude related to capturing and accessing the rationale of the system.
- Software production is a team activity. So, specifying the different tasks to each team member, analysing and managing the risks of the development will be considered.
- The teams should use Git for version management

The system should:

- be able to display information & perform actions in less than 5 seconds
- generate the documents mentioned in the functional requirements within 20 seconds
- must use validations for usernames/emails to be unique and for password to be securely stored
- ensure the the user's personal information is confidential, and only the user can see it
- ensure the user's personal & authentication information cannot be modified by any other person
- be able to restart after a failure
- be able to back-up data and recover the data using the backups
- generate fault reports & various logs (both for user behaviour & error logging)