CO2103 Resit 2019/20

Deadline: Thursday 20 August 2020, 17:00

Contribution to the grade: if you pass this resit coursework you will get a pass mark in CO2103

To pass: You must get >=40% in total.

How to submit: You must submit the following parts to the Blackboard assignment "*Assessment and Feedback > CO2103 Resit Coursework*":

- Your Spring MVC project exported as **a ZIP file** including the PDF of your class diagram
- A **video recording** of your testing and demo.

Multiple submissions: You may submit as many times as you want before the deadline. Only the last attempt will be marked.

Anonymous marking: Marking will be done anonymously. Please do not include your name in any part of your submission.

Contact: For any questions, contact the module convenors Jan (jor4@le.ac.uk) and José (j.rojas@le.ac.uk).

Collusion and Plagiarism: This is an individual assignment. We will follow University guidelines to penalise any collusion or plagiarism https://www2.le.ac.uk/offices/sas2/assessments/plagiarism/.

Task: Placements Management App Suite

Your task is to develop a Spring MWC web application to support the management of industrial placements.

Main Features [40%]

This task requires the creation of a new Spring MVC project, controllers, and JSP views.

- **[15%]** A student should be able to submit an authorisation request by completing a form providing all the necessary data about the placement (see the database bullet point below for details). There is a button "Submit", that allows the student to submit their authorisation request. Once this is done, the student cannot make any changes to the placement data.
- **[15%]** Upon login, the placement tutor should be able to see two lists, one for *ongoing* placements (those that have been authorised already) and another one for *pending* placements (those that still need authorisation).
    - There should be a "View" button for each placement.
    - There should be a button labelled "Authorise" that allows the placement tutor to approve a pending placement (i.e., set the *authorised* field to TRUE).
- **[10%]** When viewing a placement, the placement tutor should be able to add comments for that placement (ongoing or pending).

Spring Security and Validation [20%]

This task requires the security configuration of your app and the creation of controllers, validators, and JSP views for signup.

- **[10%]** The app must implement a Spring Security mechanism. Only logged in Student users can request placements and only logged in Placement Tutors can view, approve, or comment placements.
    - Implement two roles of users and their access restrictions:
        - *Students*, who can submit placement authorisation requests; and
        - *Placement Tutors*, who can authorise placements for their school
- **[10%]** The app must provide a mechanism for users to *sign up* with an *email*, *first name*, *last name* and *password* (all fields should be validated using a Spring Validator, e.g., first name cannot be empty).

Spring Data and Class Diagram [20%]

This task requires the creation of domain classes, repositories, and their JPA annotations. Controllers must be adapted to use the database.

- **[10%]** A database must be designed with the following tables:
    - *School*: The school to which each placement belongs (contains only *school id* and the *school name*)
    - *User*: Including email, first name, last name, password, and the type of user (*userType*: *TUTOR* or *STUDENT*). Each user must be associated to the school they belong to.
    - *Placement*: Must be associated to exactly one user of type *STUDENT*. Each placement must include the *placement role* (e.g., "Software developer")*, company name* (where the placement will take place), *address*, *post code*, *contact email, starting date*, *end date*, *salary*, and an *authorised* field of boolean type that is set to FALSE by default.
    - *Comment*: A date and time and the comment itself as string. Must be associated to a placement.
- **[10%]** Document your entity classes in a class diagram. For each association write one sentence how it is implemented in the DB, i.e., what fields and Spring Data annotations you use.
    - Place the class diagram as a PDF at the root of your Spring project **with filename cd.pdf**.
    - You may use this or any other editor https://app.diagrams.net/ (formerly draw.io).

Manual Testing and Demonstration Video [20%]

This task requires the design of tests scenarios for your Spring MVC application (you are **not** required to write executable test scripts).

- **[10%]** Design 5 test scenarios for your application.
    - These tests must include a negative and a positive outcome of at least one Spring Validator for the placement form.
- **[10%]** Record a 5 to 10-minute demonstration of your software where you walk through your implementation and execute your 5 test scenarios.

Video recording/upload instructions from Blackboard:

Step 1: Record or upload your video in Panopto. To start creating your video, click here to open your personal folder.
Step 2: Open the assignment in Blackboard and select **Write Submission**.
Step 3: In the text editor, expand **Mashups** and select **Panopto Student Video Submission.**
Step 4: A window will open to show the videos in your personal folder. If your video is in a different folder, select the correct folder from the drop-down at the top.
Step 5: Select the video you wish to submit and click **Insert.**
Step 6: Your video will be added to the submission. Add any extra information and **Submit**.

Further Remarks

Dependencies of Tasks

Note that most tasks can be attempted independent of the others, e.g., you can attempt the Spring Data task without the Main Features task and vice versa. However, errors in the Spring Security task might break your Main Features task. Make sure that your code works to receive the marks for each task.

Use of Technologies and Patterns Taught in CO2103

There are many ways to develop Spring MVC applications. If you do not follow the patterns we taught in CO2103, e.g., if you use different database or repository implementations or if you configure Spring Security differently, you must provide a full list of resources and links you have used. This can include code samples, YouTube videos, examples, etc. and you must list all of them if you use them.
If you use implementations patterns that we did not teach and you do not declare their sources, we will start plagiarism investigations.

Video Recording

The Panopto video recorder (see the link above and inside the assignment on Blackboard) is available for Windows and Mac OS. If it is not available for your system you can use any screen recording tool and upload the video to Panopto. On Ubuntu the tool Kazam is a good choice (sudo apt install kazam).