Lab project on Information system and databases discipline

Objectives

As a result of this assignment you will learn basic principles of client-server information systems.

Prerequisites

Courses: Programming, Databases, Web Development.

Instructions

You should create teams of 3 classmates to develop the information system.

The information system to develop should be the organized following the principles of three-tier model. Front-end tier as UI for a user to interact with the system, typically implemented as web-application or web-site. Back-end tier carrying all the domain-specific logic of the system and managing the data, resources, user sessions and accounts. Back-end tier typically runs on a web-server or as a web-service being separated from the front-end host. Data storage tier responsible for data storage and manipulations typically represented as database management system. For example, in the running system, user interacts with front-end using web-browser, which sends requests about different operations about the data to the back-end, which in turn stores the data into the database, queries it to give back to the front-end and demonstrate to the user, and so on.

Steps to accomplish the task

- Stage 1 (10 points)
  - Create the Use Case diagram for your informational system. The diagram should demonstrate the actors and their actions in system.
  - Create UI sketch showing which pages you are going to organize it with and how are you going to navigate between them.
  - Create the diagram demonstrating database scheme you are going to use for data storage.
  - Choose technologies to use for implementation of the project.
    (defaults are: HTML+JS for front-end tier and Spring Framework on Java for back-end tier)
  - Send your diagrams along with explanations of the design in the discipline's gitlab to get approve and comments.
  - If there are any comments, make modifications according to them.
- Stage 2 (10 points)
  - Implement REST-service with CRUD (Create, Read, Update, Delete) operations.
- Stage 3 (10 points)
  - Implement front-end for presenting collection of system's elements, their creation, editing, deletion and make pagination on the front-end and back-end sides.
- Stage 4 (10 points)
  - Add to existing back-end and front-end users, registration, login forms, the log off button. Changing the password.
- Stage 5 (10 points)
  - Add access control to your system.

Optional additional tasks (if you want to improve your score):

- Add search or filters (5 points).
- Add interaction between users of different roles (5 points). You could discuss it with your practice instructor.

Task variants

Variant 1: Educational courses catalog

The system to be designed should represent an information about the training courses for students. Each teacher giving the particular course can add some information about it into the system. It is required to specify course's name, planned amount of hours for learning activities, brief description text, course's syllabus file in any format. When a number of courses was submitted into the system by teachers, any user can browse through the catalog's collection to reach and obtain the information being presented and download syllabus.

The requirements for your project:

1. User should be able to see a catalog of course descriptions.
2. User should see course's name, planned amount of hours for learning activities, brief description text, course's syllabus file in any format in a menu.
3. Teacher should be able to add any number of courses to the catalog.
4. Teacher should be able to modify his course' information.
5. Teacher should be able to delete his course' information.
6. User should be able to use pagination if there are too many courses in the catalog.
7. User should be able to register in a system.
8. User should be able to log in a system.
9. User should not be able to log in while being logged in.
10. User should be able to log out.
11. User should be able to change password.
12. Administrator should be able to switch user operational mode to a *teacher*.
13. Administrator should be able to switch user operational mode back to *student*.

Variant 2: Rental ads board

The system to design intended to be an ads board for a rental offers. It is targeted for users wanting to rent an apartment or any other residential property. If you are an owner of an unoccupied living apartments or some estate, which you want to rent out, you can use the system to publish an ad on the board, so that it would be available for your potential tenants. A rental offer consists of the address, where the property is located, the description text, and a number of photos being attached.

The requirements for your project:

1. User should be able to see a catalog of ads.
2. User should see address, photo and description of the apartment in a catalog.
3. Landlord should be able to create the new ad in a catalog.
4. Landlord should be able to modify the ad in a catalog if he created them.
5. Landlord should be able to delete the ad if he created them.
6. User should be able to use pagination if there are too many positions in a catalog.
7. User should be able to register in a system.

8. User should be able to log in a system.
9. User should not be able to log in while being logged in.
10. User should be able to log out.
11. User should be able to change password.
12. Administrator should be able to turn on property owner functions for a user.
13. Administrator should be able to turn off property owner functions for a user.

Requirements for a system

Front-end

Web application consists of some pages and built using following technologies: html5, css/sass/scss, JavaScript (or TypeScript on demand), JSP. If you want to use any front-end framework you can discuss it with teacher. It's allowable, but not the aim of the course.

Back-end

Spring Java project runs on the web server and communicating with database using Hibernate ORM.

Database

PostgreSQL

Hints

- Stage 1 should be done at the beginning. Other stages could be done in any order, but for the end of the course your whole project should meet all mentioned requirements.
- As you are working on groups you have at least 2 types of strategy: to divide the tasks and responsibility or to write all the code together following the pair programming styles.
- You don't need to zip your source code to submit it to the gitlab. Try to figure out how to use git system control version. It will be useful in the future when working in the company.
- Check that you can build and run your project without compilation or runtime errors before each submission. If your solution can't run, it can't be assessed.