

Final project: Timekeeping

DESCRIPTION

Develop a web-app according to the task variant: Timekeeping.

DETAILS

1. There are roles: administrator and system user.
2. The user :
 - a. must have an account;
 - b. can have one or more activities;
 - c. notes the amount of time spent on each activity;
 - d. can send a request to add / remove activity.
3. The administrator manages :
 - a. users;
 - b. categories of activities;
 - c. activities;
 - d. confirms (fixes) a certain activity on users;
 - e. view a list of all available activities, perform sorting:
 - i. by name;
 - ii. by category;
 - iii. by number of users.
4. Perform filtering by activity category. View report for all users, number of activities and time marked by user.

REQUIREMENTS

	Status
Create classes that correspond to essence of the subject area.	Processing ▾
Classes and methods should have names that reflect their functionality and should be spaced in packages.	Processing ▾
The code design must comply with the Java Code Convention.	Processing ▾

	Status
Store subject area information in a relational database (it is recommended to use MySQL or PostgreSQL as a database).	Done ▾ PostgreSQL
<p>The application must support:</p> <ul style="list-style-type: none"> <input type="checkbox"/> the use of Cyrillic (be multilingual), including the storage of information in the database; <input type="checkbox"/> to switch the interface language; <input type="checkbox"/> there should be support for input/output and storage of information (in the database) recorded in different languages; <input type="checkbox"/> choose at least two languages: one Native, the other English; <input type="checkbox"/> dates must be implemented through the java.time. 	Not implem... ▾
Implement protection against re-sending data to the server when refreshing the page (implement PRG).	Not implem... ▾
Authentication and authorization must be implemented in the application, delimitation of access rights of system users to program components. Password encryption is recommended.	Not implem... ▾
Introduce an event log into the project using the log4j library.	Not implem... ▾
The code must contain comments on the documentation (all top-level classes, non-trivial methods and constructors).	Not implem... ▾
The application should be covered by modular tests (minimum coverage percentage 40%). Writing integration tests is recommended. The use of Mockito is recommended.	Not implem... ▾
Implement the mechanism of data pages pagination.	Not implem... ▾
All input fields must be with data validation.	Not implem... ▾
The application must respond correctly to errors and exceptions of various kinds (the end user should not see the stack trace on the client side).	Not implem... ▾

	Status
Independent task functionality expansion is recommended! (Adding captchas, generating reports in various formats, etc.)	Not implem... ▾
The use of HTML, CSS, JS frameworks for the user interface (Bootstrap, Materialize, etc.) is recommended!	Not implem... ▾
Development of projects with the help of Git is required.	Not implem... ▾
The application must be structured according to the MVC and Spring Boot architectures.	Not implem... ▾
Project Lombok use is allowed.	Not implem... ▾
The use of the Spring Resource Bundle is recommended.	Not implem... ▾
Spring Authorization must be used.	Not implem... ▾
JPA must be used for data access (Spring Data and/or Hibernate).	Not implem... ▾
Handling exceptions with help of Exception Handling with Spring for REST API is recommended.	Not implem... ▾
Use of ThymeLeaf is recommended.	Not implem... ▾
Use of additional frameworks (Swagger etc.) is recommended.	Not implem... ▾