

Template code review

Overview

This piece of code is a frontend for our project's login page. It adds maximum minimalism to the appearance of the first page.

Goals

- 1) To make the whole web site to become a part of a big family of Innopolis-style portals.
- 2) To make the login system maximally clear and user-friendly
- 3) To make it satisfy all modern standart login pages
- 4) To create a unique design which can be recognizable among Innopolis users

Usage Scenarios

Improving the appearance of the website by changing separate modules

Class Diagrams

Do not make any sense actually, but still it is possible to add them

Code

Categories for review

1. Design decisions

Example:

[1.1] NameTTT: You should use HASH_TABLE for 'yyy' instead of ARRAY.

[1.1.1] Owner: the decision of using ARRAY is based on the fact that ARRAYs are lighter than HASH_TABLES

[1.1.2] NameTTT: That might be true, but you are compromising execution time, 'yyy' is used to perform several searches. Having a HASH_TABLE one can have constant complexity in searching

[1.1.3] Owner: Ok, I will change the design and implementation. [CLOSED]

2. *API design*
3. Architecture (including inheritance hierarchy, if any)
4. Implementation techniques, in particular choice of data structures and algorithms
5. Contracts
6. Programming style, names
7. Comments and documentation

1. Design decisions

2. API design

3. Architecture (including inheritance hierarchy, if any)

4. Implementation techniques, in particular choice of data structures and algorithms

5. Exceptions handling - Contracts

6. Programming style, names

7. Comments and documentation