CONDITIONS OF COMPLETING THE COURSE PROGRAMMING WORKSHOP V

Faculty: Applied Informatics
Full-time first degree studies
Winter term 2013/2014

Lecturer of the course: **Katarzyna Wójcik**

Cracow, 30 January 2015

1 REQUIREMENTS

The assessment of the course is done on the basis of final test and final project. Project can be created in maximum 3-students groups while test is taken individually

1.1 FINAL TEST.

Final test will be organized in examination session (initial proposition – together with Data Mining. Concepts, Model and Methods exam). It will have the form of open questions with short answers.

To be well prepared for the test student have to know answers to the following questions:

1. Version control system:
    - What is the Version Control System?
    - What is the difference between collaboration branch forking?
2. Bootstrap
    - What is Bootstrap?
    - How Bootstrap is used in application coding?
3. Object-oriented programming in PHP:
    - Between what tags should the PHP code be included?
    - According to PHP convention in what kind of files the closing PHP code tag should be skipped?
    - How to run a PHP script from the command line?
    - What is the difference between require and require_once?
    - How do we write class names according to PHP convention?
    - How do we write class members names according to PHP convention?
    - From what character the name of each variable in PHP begins?
    - What do we use the „this" keyword for?
    - On the basis of OOP, the class and the object are the same?
    - On the basis of OOP, objects are passed by value or reference?
    - In the absence of a label before member function, what is its access modyfier?
    - What is an abstract class, what is it used for?
    - What is the interface, what is it used for?
    - How overloading of methods works?
    - How overwriting of methods works?
    - What are the static member functions?
    - Write at least one commend displaying text in PHP.
4. Composer:
    - What is Composer?
    - What kind of information is included in composer.json file?

5. Symfony2:
    - What is Symfony2?
    - Which command allows you to list the commands in Symfony2?
    - How to identify the root directory of the project in Symfony2?
    - What is included in the src directory in Symfony2 project?
    - What are bundles in the Symfony2 project?
    - What is MVC model?
    - How does the MVC model is implemented in Symfony2?
    - What is Twig?
    - How does the template inheritance in Twig work?
    - What is Doctrine?
6. Other
    - What is the primary use of the methods GET and POST?
    - Is it possible to transfer data to the server using GET and POST at the same time?
    - What is SQL Injection?
    - What is CURL?

1.2 Final Project

The topic of the final project is an application used to borrow movies. The application must implement all the scenarios listed below. In addition, it must meet the following requirements:

1. All members of the team chould have comparable contribution to the project.
2. The project has a full history kept in the repository:
    a. All members of the group use a single repository, or iech of them has his/her own fork of the project.
    b. The consecutive functionalities of built system are added to the repository in a "feature/branch" schema.
    c. It can be directly verified who is responsible for certain part of the project.
3. Each team member has the knowledge allowing him to present (sell to the customer) created system.
4. Each member of the team is able to explain the solutions used in personally created parts of the application.
5. Final project is written entirely by group members or with the use of open source components under the terms of their licenses.
6. To obtain 5.5/cel (additional requirements) all requirements for a 5.0/very good must be met

1.3 USER STORIES

1. User story 0
    - As User I can log in
    - As a logged in user I can view my orders,
    - As a logged in user I can log out,
2. User story 1
    - As a user I visit home page, (/)
    - I see the list of available movies
    - I see a list of popular movies (most often borrowed)
    - I see a list of the most-reviewed films

    a. I see a list of genres from which I can choose movies
3. User story 2
    - As a user I visit home page, (/),
    - The film's Title is a link to the movie
    - The film's Cover is a link to the movie
    - I visit first video on the list (/movie/{slug})
    - I see the Description, Cover art, a List of actors,
    - I see the price for borrowing a movie,
    - I see Borrow button,
    - As a logged in user I see Add a review button
4. User story 3
    - As a logged in user I display the movie
    - I click the button Add a review
    - I fill the text box with the movie review
    - I am sending a review
    - I am redirected back to the movie page
    - I see the information about adding a review
    - I don’t see newly added review.
5. User story 4
    - As the unknown user I visit the movie page
    - I click the Borrow button
    - I am asked to sign in
    - I see a link to the registration,
    - If I don’t have an account I register
    - If I have an account I log in

6. User story 5
    - As a logged in user I visit the movie page
    - I see Borrow button,
    - I click the Borrow button
    - I am redirected to the page with the information about the form, terms and payment date
    - I get an email with the information about the form, terms and payment date
7. User story 6
    - As a logged in user I go to the list of orders
    - I see the orders along with the statuses (in progress, paid)
8. User story 7
    - As a logged in user, I display a list of borrowed movies
    - I see the button Watch
    - I click the Watch button (message is displayed)
9. Additional requirements (for 5.5):
    - Shopping Cart - the film is added to the shopping cart and the service of the order is carried out from the shopping cart, where you can manage (add items, delete items, realize orders, etc.)
    - Implementation of payments in the test channel
    - When you chose Watch the movie option it turns on movie, downloaded/streamed from the server, using html5 player
10. List of necessary models:
    - Movie
    - Order
    - User.

1.4 DOCUMENTATION

Part of the project is the documentation. You need to prepare both the technical documentation and user manuals. Technical documentation can be generated automatically using widely available tools, however, must meet the requirements set.

1. The technical documentation shall contain:
    a. The names and versions of the technology used
    b. Physical model of database used in the project
    c. Information on the environment required to run the program
    d. Links to software that make up this environment
    e. (if object-oriented) class names, table with a list and description of the attributes, table with a list and description of (the return value, taken parameters, how does it work) member functions (including constructors and destructors);

f. (if procedurally): a table with a list and description of the variables, the table with the list and a description of (the return value, taken parameters, how does it work) the functions in division into files

User Documentation - user's manual showing step by step (preferably with screenshots) how the application works from the way it is run up to display data from a database collected during the application performance; manual should more or less present the realization of user stories.

2 EVALUATION SYSTEM

2.1 FINAL TEST

Final test will be organized in examination session (together with Data Mining. Concepts, Model and Methods exam). It takes about 15 minutes. Test will have the form of open questions with short answers. Evaluation of the test will be given to students within the two weeks after the date of test. Points obtained from the test will be converted to the assessment scale:

Percentages	Grade
0-49	niedostateczny (2.0)
50-59	dostateczny (3.0)
60-69	plus dostateczny (3.5)
70-79	dobry (4.0)
80-89	plus dobry (4.5)
90-100	bardzo dobry (5.0)

2.2 PROJECT

Final project must be hosted on university v-ie.uek.krakow.pl server or on any hosting server so that it can be accessed from a web browser. By the appropriate slot on the e-learning platform a link to the homepage and a link to the public repository on the GitHub (or another based on the git) should be sent. Through the next slot the technical documentation and instructions for the user must be sent. Each in a separate file. In the next slot only the database schema as an image file must be sent. Slots for sending files will have labels corresponding to part of the project, which should be sent through the slot.

All components of the project must be submitted by the end of the session, ie. until January 15th 2015. Then the whole project group should report on duty hours in order to verify the project. All components of the project will then be graded (application, database, files). General criteria for each part of project are presented in the table in section 2.5.

ATTENTION! If the presentation will be sent on time, but the team is subjected to verification in correctional session (sesja poprawkowa) the punctuality grade will be reduced by half a degree.

2.3 PENALTIES / REWARDS

The lack of preparation for classes or lack of work in the classroom may be punished by subtracting up to 5 points at once.
The activity at classes can be awarded by adding up to 5 points at once.
Penalties/rewards will be subtracted/added after student will complete the course with positive grade. Points of punishments and rewards will be added to the result of a test (1 point - 1%).

2.4 FINAL GRADE

After the verification of projects the final grade will be determined. Every criterion will be graded individually and then weighted average will be counted. On this basis final grade will be determined. Every part of the project (connected to particular criteria) must be graded positively to be the basis of final grade.

	Partial evaluation	**Weight**	**Grade**
Application	at least 3.0	33,5%	**Weighted average**
Database	at least 3.0	11%	
Documentation	at least 3.0	11%	
Theory	at least 3.0	33,5%	
Punctuality	at least 3.0	11%	

Grade	2.0	3.0	4.0	5.0	5.5
Application	Student is not able to create project of simple internet application. Student is not able to code simple application on the basis of its project.	Student is able to create project of simple internet application. Student is able to code simple application on the basis of its project.	All required for 3.0 and additionally: Student is able to code simple internet application in object-oriented PHP. Student is able to use simple visualization in his/her application.	All required for 4.0 and additionally: Student can write the application using Symfony2 framework. Student can enrich his/her application with complex visualization effects.	All required for 5.0 and additionally: Student can use different authorization methods in created application.
Database	Student is not able to create database logical model.	Student can create database logical model according to specified requirements.	All required for 3.0 and additionally: Student is able to create relational database on the basis of its logical model using creator or similar tool for database creation. Student is able to write simple queries to database.	All required for 4.0 and additionally: Student can create physical database model. Student is able to write complex database queries. Student knows how to use forms to insert data into database.	All required for 5.0 and additionally: Student can specify database requirements for particular application.
Documentation	Student is not able to write neither user guide nor documentation to the application. Student is not able to present results of his/her application during classes.	Student can write simple user guide to created application.	All required for 3.0 and additionally: Student is able to prepare technical documentation to his/her application.	All required for 4.0 and additionally: Student can create professional user guide and technical specification of created application.	All required for 5.0 and additionally: Student can prepare multimedia presentation containing instructions for final users and technical description of the application.

Punctuality	Student haven't submitted working application within the deadline.	Student can prepare timetable for the project . Student submits working application on time.	All required for 3.0 and additionally: Student submits full working application on time. Student submits user guide and documentation on time.	All required for 4.0 and additionally: Student is able to, within the prescribed period of time, work out graphical side of application. Student submits multimedia presentation on time.	All required for 5.0 and additionally: Student presents multimedia presentation and working application during classes.
Theory	Student doesn't know what is framework. Student doesn't know object-oriented programming paradigms	Student know few framework definitions. Student can mention most important object-oriented programming paradigms	All required for 3.0 and additionally: Student is able to describe framework in his/her own words and knows what is it used for. Student can give some examples of using object-oriented programming paradigms in practice.	All required for 4.0 and additionally: Student can give few examples of PHP frameworks. Student can specify most important PHP programming conventions. Student can use in practice object-oriented programming paradigms.	All required for 5.0 and additionally: Student can specify main differences between particular PHP frameworks.

CONTENT