Introduction to Play! framework

Jonathan Pastor

October 1, 2013

1 Introduction

The objective of this course is to introduce the basic features of a modern web framework, through the study of a java compatible Model-View-Controller [1] (MVC) web framework: Play! [2].

During this course, students will learn how to develop a small web application. This application will use a database for storing objects, and users will interact with it by using web forms.

Students will be asked to work in pairs: each pair will develop a dynamic website. Specifications are given in the section 3 : it is a set of constraints where each satisfied constraint gives points. The satisfaction of the constraints will be evaluated during an oral defense.

2 The Subject of the project

The pairs will develop an application that will be the response to the following needs:

> Professors of the Computer Science (CS) department would like to be able to create meetings involving several participants. The involved people would review the meeting and let a comment about the proposed meeting. It would be also possible to handle conflicts between meetings. People may be notified when they take part to a meeting.
>
> Project subject for year 2013-2014.

> Meetings are created by a user. A meeting involve one or several people. A meeting can contains comments from any member of the meeting. A member can receive notifications about some of his meetings.
>
> Example of data structure for year 2013-2014.

[1] cf. http://en.wikipedia.org/wiki/Model-view-controller
[2] cf. http://www.playframework.com/documentation/1.2.5/home

3 The Specifications

3.1 Fonctional Specification (10pts)

Constraint	Description	Points
Fonctional-1	An user can create, delete, modify and view a meeting.	2 pts
Fonctional-2	Participants of a meeting can view their meetings.	2 pts
Fonctional-3	Participants of a meeting can leave a comment on the meeting.	2 pts
Fonctional-4	New meetings appears automagically without reloading the page (AJAX).	2 pts
Fonctional-5	Implement a feature that improves the application	2 pts
Total		10 pts
Bonus	The application is usable by members of the C.S. department.	3 pts

3.2 Technical Specification (20pts)

Constraint	Description	Points
Controller-1	Develop a controller for the frontend	1 pts
Controller-2	Develop a controller for the backend	1 pts
Controller-3	At least one controller asks data with a custom SQL query	1 pts
View-1	Display data (a list) with HTML	2 pts
View-2	Display data (an element) with details	1 pts
View-3	Create a transition between 2 views	1 pts
Database-1	Database contains at least one complex entity (that contains a set of other entities, cf appendix A)	3 pts
Database-1	The database is in third normal form [3]	1 pts
CRUD	At least one complex data entity have create, read, update and delete actions. These actions are located in the backend.	3 pts
AJAX-1	At least one controller have a method that produce JSON or XML.	1 pts
AJAX-2	At least one view gets data dynamically with an AJAX request.	1 pts
Design-1	Use a CSS framework (Bootstrap[4], Zurb Foundation [5], PureCSS[6], ...).	1 pts
Design-2	The application is simple and friendly to use	2 pts
Design-3	The application has been developed around prototyping methodology. [7]	1 pts
Total		20 pts
Bonus-1	Integrate the application with members given by LDAP.	2 pts

[3] cf. http://en.wikipedia.org/wiki/Third_normal_form
[4] cf. http://getbootstrap.com/
[5] cf. http://foundation.zurb.com/
[6] cf. http://purecss.io/
[7] cf. http://en.wikipedia.org/wiki/Software_prototyping

4 Evaluation

Evaluation will be made during an oral defense : a jury will check if the students have a good comprehension of the technologies seen during this course; each constraint will be evaluated. The jury will then decide of a mark for the oral defense (10 pts).

$$\text{finalMark} = \frac{\text{technicalMark} + \text{functionalMark} + \text{oralMark}}{2}$$

Appendices

A Complex Data Entity

Here is an example of complex data:

```
class A {
    List<B> listOfBs;
}

[...]

class B {
}
```

In this example, entity A can contains several B entities.