Tasks to Solve During Seminar 4

Internet Applications, ID1354

Write a document with your answers to the following tasks, and upload it to Canvas at the end of the seminar. The format of the document is not important. The problems shall be solved by all group members together. You may also write the document together, but everyone must add their personal section describing what was learned during the seminar.

Mandatory Task

- All members of the group shall, in turn, explain and motivate their program, and also their report, to the other group members. Write a brief summary of differences, and comment on advantages and disadvantages of the different solutions. Do not write about minor layout differences.
- Go through all requirements specified in the lab tasks, and evaluate that your web sites meet them.
- Evaluate that your reports meet the requirements in the report template and in the lab tasks.
- Evaluate whether the JavaScript and PHP code is well designed and has an appropriate layered architecture. In particular, what have you done to maintain high cohesion and low coupling in the JavaScript code?
- What is the content of your AJAX requests and responses? Verify that no HTML code is sent in response to an AJAX request.
- Try to explain the main difference(s) between an application using Knockout (or another framework introducing a view model), and an application not using such a framework. You may skip this if no group member used a JavaScript framework.
- How much extra work is required to implement long polling (optional task 2)? Even if no one in the group has implemented long polling, you can still try to understand what would have to be done to use long polling.

- A good way to evaluate if the code is flexible and easy to understand is to actually try to change something. Select one or more of your web sites and try to change it *without the help of the author*. For example, ask the user for confirmation when a comment is deleted, by showing the text *Do you really want to delete this comment?*. Another suggestion is to change the way a user is informed of successful or failed login.

Optional Tasks

Solve as many of these tasks as time allows, and write your solutions in the same document as the mandatory task. If the solution is a piece of code, just paste the code in the document, without bothering about figures or formatting. The tasks may be solved in any order, you do not have to start from task one.

Task 1

Callback functions are used very frequently in JavaScript. Here are some exercises to give you a better understanding of callbacks.

a) Try to understand the JavaScript `map` function, which uses a callback to map each element in one array to another element in another array, see for example `http://www.w3schools.com/jsref/jsref_map.asp` Write a function that takes an array as parameter and uses `map` to add the index number to each element. It should change the array `["a", "b", "c"]` to `["0:a", "1:b","2:c"]`

b) Now write you own implementation of the `map` function itself. Try it with the program you wrote in bullet a) above.

c) Explain exactly the meaning of each line in listing 1 below. Also explain when each line is executed.

```
$(document).ready(function () {
    $("#someElem").click(function () {
        $("p").css({display: "none"});
    });
});
```

Listing 1: JavaScript code with callbacks.

Task 2

```
function Person(name) {
    this.name = name;
```

```
 3     this.handleName = function(callback) {
 4         callback(this);
 5     }
 6 }
 7
 8 function nameHandler(obj) {
 9     alert("name: " + obj.name);
10 }
11
12 var person = new Person('Stina');
13 person.handleName(nameHandler);
```

Listing 2: Callback and objects

Listing 2 is an attempt to use objectoriented design best practices, by keeping objects intact.

a) What happens on line 4, and is there any reason to pass `this` instead of `name`? What does `this` contain?

b) Can you see any reason why the function `handleName` (lines 3-5) should *not* be replaced by the function `nameHandler` (lines 8-10)? This would make the code shorter and easier to understand, and the outcome of running the program would be exactly the same.