

# A10: Open Web Service Assignment

[ Hand-in deadline: Friday, Nov 30, 2012 at 08:29:59 ]

## Goal

In this assignment you will design and implement a web service that makes use of PHP and SQL. However, you decide yourself which web service to build! For your inspiration, you will find a list of possible web service ideas (to use "as is", or as inspiration for your own idea) later in this document.

---

## The 4-Step Design Process for (database-driven) Web Services

For your assignment project to be regarded as valid, it needs to force you to go through all four design steps we have discussed in class, and you need to hand in all relevant files (put them in your A10/ directory) for each step of your design process:

### STEP 1: Design of data model:

- What info should be stored?
- How should it be represented?

Note: be aware of redundancy (you might need several tables).

### STEP 2: Development of data transactions:

- How is data to be inserted into the database?
- How is data retrieved?

### STEP 3: Development of site map and forms:

- Draw a site map diagram of screens the user will see (incl. arrows between them)
- Figure out which are PHP scripts and which can be HTML pages and

how information needs to be transferred (submitted) from page to page (write this info on the arrows of the above site map diagram).

- Use HTML for designing the user interfaces (UIs)

## **STEP 4: Program the System:**

- Program the database transactions in SQL
  - Program the rest of the Web Service in PHP
- 

## **Web Service Examples:**

Choose one of the following Web Service alternatives; **A)**, **B)**, **C)**, or **D)**:

### **A): An Online Quiz Service**

Destination file: A10/ \*.sql, \*.html, \*.php, \*.jpg, \*.txt

This exercise alternative consists of making a quiz service where people can register a new question along with one right answer and two so-called “distractors” (ideally plausible, but wrong answers). The possibility of registering new questions might be password protected – it’s up to you. Users should then be able to “take the quiz” which means being presented with questions (either at random or in sequence – up to you). The service could also count the number of correct and incorrect questions and display this information during the quiz. The assignment is deliberately left open.

Remember to go through all four design steps and hand in files for each step as described in the beginning of this assignment. In addition, hand in a file named `description.txt` where you describe, in 10-20 lines of text, your service in your own words and reflect on what was tricky and what was fun with the exercise.

### **B): Polling Service**

Destination file: A10/ \*.sql, \*.html, \*.php, \*.jpg, \*.txt

This exercise alternative consists of constructing a service that enables anyone to insert a polling service (visitors answer yes or no to a given question) to their homepage and follow the progress on how people answer the poll. The user who wants to use the polling service on her/his

web page signs up as a user of the service (e.g. username, password). After that, the user can insert direct links on her/his page to PHP scripts which updates tables in a database that stores the number of yes and no answers of the poll. Examples:

```
<a href='http://www.mypoll.com/poll.php?poll_id=3'>Will SAS  
survive until New Year 2012?</a>
```

```
<a href='http://www.mypoll.com/poll.php?poll_id=7'>Will  
Peter and Amalie still be together in a week?</a>
```

Remember to go through all 4 design steps and hand in files for each step as described in the beginning of this assignment. In addition, hand in a file named `description.txt` where you describe, in 10-20 lines of text, your service in your own words and reflect on what was tricky and what was fun with the exercise.

### **C): Admin Interface for a Previous Web Service**

Destination file: A10/ \*.sql, \*.html, \*.php, \*.jpg, \*.txt

This exercise alternative consists of constructing an administration interface to one of the services you have already constructed as part of a previous assignment. The administration extension should enable the administrator of a service to fully control the service from a web interface instead of accessing the database using the MySQL text client. Suggestion for contents: login, delete/add/update entries in the database (e.g. user names, event names, event descriptions, etc.).

Remember to go through all 4 design steps (as discussed at the lectures, this is necessary not only when you design a completely new service, but also as in this case, when extending/changing an existing one) and hand in files for each step as described in the beginning of this assignment. In addition, hand in a file named `description.txt` where you describe, in 10-20 lines of text, your service in your own words and reflect on what was tricky and what was fun with the exercise.

### **D): Your own (Web Service) idea!**

Destination files: A10/ \*.sql, \*.html, \*.php, \*.jpg, \*.txt, `description.txt`

Implement your own idea for a web service! No matter if you get inspired by the above ideas or if you make your very own service (like aunt Ursula's

Photo Album or uncle Benny's Giraffe Rental Service), you should use as much of what you have learned in this course as possible (e.g. checking user input using regular expressions, including functions from external files, passing hidden variables between forms and scripts, interacting with the database).

Remember to go through all 4 design steps and hand in files for each step as described in the beginning of this assignment. In addition, hand in a file named `description.txt` where you describe, in 10-20 lines, your service in your own words and reflect on what was tricky and what was fun with the exercise.

---

## Checklist

If you have done all the exercises of this assignment, you should have the following files in your personal DSDS directory:

- A10/ \*.sql, \*.html, \*.php, \*.jpg, \*.txt (the exact file names depend on what web service exercise you choose to do.)
  - A10/description.txt (describes your service in 10-20 lines)
-