Re-exam for the course Introduction to Scripting, Databases, and system architecture (DSDS-F2011)

48-hour "take-home" exam designed by Thomas Pederson

Start date & time: 13.00 Wednesday the 31st of August 2011, at which time the exam will be published on the course blog (https://blog.itu.dk/DSDS-F2011/).
Hand-in deadline: 13.00 Friday the 2nd of September 2011, exam office in the 2E corridor at ITU.

General information about this exam

Exam exercise solving

The students attending the exam are to follow the same procedure they have used when solving the mandatory assignments on the course, except for the delivery (hand-in) of their solutions (see details below). This means that students are recommended to set up their own database on the ITU MySQL server or use the one they have already set up as part of solving the assignments. Details for setting up a MySQL database on the ITU server can be found in assignment 6 available on the course blog. The exam solution scripts are then to be developed and executed on the ITU PHP server by placing the PHP and HTML files in their personal course directory (W:\f2011\DSDS\username\). Students who want to develop their exam solution using a different server are welcome to do so but need to make sure to make the code accessible to the censor and course responsible by including a web address in the hand-in package to where the files can both be inspected and executed (see exam hand-in below) during four weeks following the exam delivery deadline.

Solving the exam exercises involves both the writing of HTML, PHP, and SQL code as well as describing parts of the design in natural language, similar to how solutions for assignment 6-11 on the course was done. The natural language used can be either Danish or English.

Students are welcome to re-use and include files that they have used themselves throughout the course, e.g. fn_headerfooter.php, fn_input_validation.php, fn_mydb_connect. Such files naturally also need to be included on the CD-ROM and located in the appropriate place on the web server. Information and code examples freely available on the web (e.g. in PHP programmers online forums) are also valid resources if successfully adapted to the exam exercises and suitably commented. Under no circumstances however are students on the DSDS course allowed to copy and/or share information with each other as part of solving the exam (see "Check for Plagiarism" below).

Exam hand-in

The solutions are to be delivered in the shape of .html, .php, .sql, and .txt files all to be burned onto 3 identical CD-ROMs and handed in to the exam office (the 2E corridor at ITU) no later than Friday the 2nd of September 2011 at 13.00. A web address to a folder containing a directly executable version of the designed web service (e.g. http://www.itu.dk/stud/f2011/DSDS/username/exam/) as well as the complementing .sql, .txt, and .jpg files is to be included a) on a paper note accompanying the CD-ROMs, b) as a clickable link from a very simple HTML file called "username.html" included on the CD-ROMs. "username" in both cases mentioned above refers to the specific student's ITU username, e.g. "tped".

Hint: prepare your delivery in good time (e.g. you should ideally have made sure to have a working procedure for burning CD-ROMs *before* the exam starts) because the hand-in deadline is hard. If you miss it, your next chance is not until the next (re-) exam occasion.

Grading

7 point scale. Example of factors that influence the grade:

- **functionality** – do the solutions execute without errors and meet the requirements specified in the exam?
- **system robustness** – do the solutions a) handle bad user input gracefully, b) handle a non- responding mySQL server gracefully?
- **code quality** – is the code well structured and commented so that a fellow system developer (in this case the censor and the course responsible) easily can understand what your code does?
- **overall system design quality** – in the case of a larger exam exercise, is the overall design well thought-through so that the redundancy in the MySQL database is minimized and that the functionality of the web service is logically distributed over a set of HTML and PHP files?
- **(X)HTML validation** – do all HTML pages (including the ones generated by the PHP code) successfully pass XHTML validation (transitional or strict, the student's choice)?

Check for plagiarism

Due to the nature of this exam, extra careful checks for plagiarism will be performed on the handed-in exam solutions. Students are not allowed to collaborate in solving the exam. Thus, students handing in identical or near-identical solutions will be reported as part of the normal handling of misconduct and fraud at ITU. For details, see http://intranet.itu.dk/en/Studiehaandbogen/Eksamen/Regler-og-retningslinjer.

Overall task: Design and implementation of a simple DVD movie subscription service

Introduction

In this exam assignment you will design and implement a subscription service that allow subscribers to regularly receive DVD movies to their home mailbox, one at a time, based on their personal movie preferences. As soon as the subscriber has watched a DVD and mailed it back to the service provider, the subscriber notifies the service provider that the watched DVD is on its way back by for instance clicking on a specific link, at which point your movie web service should send a new DVD to the subscriber. The actual sending is of course never performed as part of this exam but the fact that a DVD has been sent to the subscriber (and which one) should be shown on the web site somehow. Using your web service, the subscriber should be able to select a subset of DVD movies that are of particular interest to the her/him, from a larger set of DVD movies stored in the service provider's database. It is from this subscriber-selected subset of DVDs that the next DVD to be sent is automatically picked by the web service and mailed to the subscriber. The web service is a simplified version of existing similar web services such as www.lovefilm.dk.

General requirements for your system:

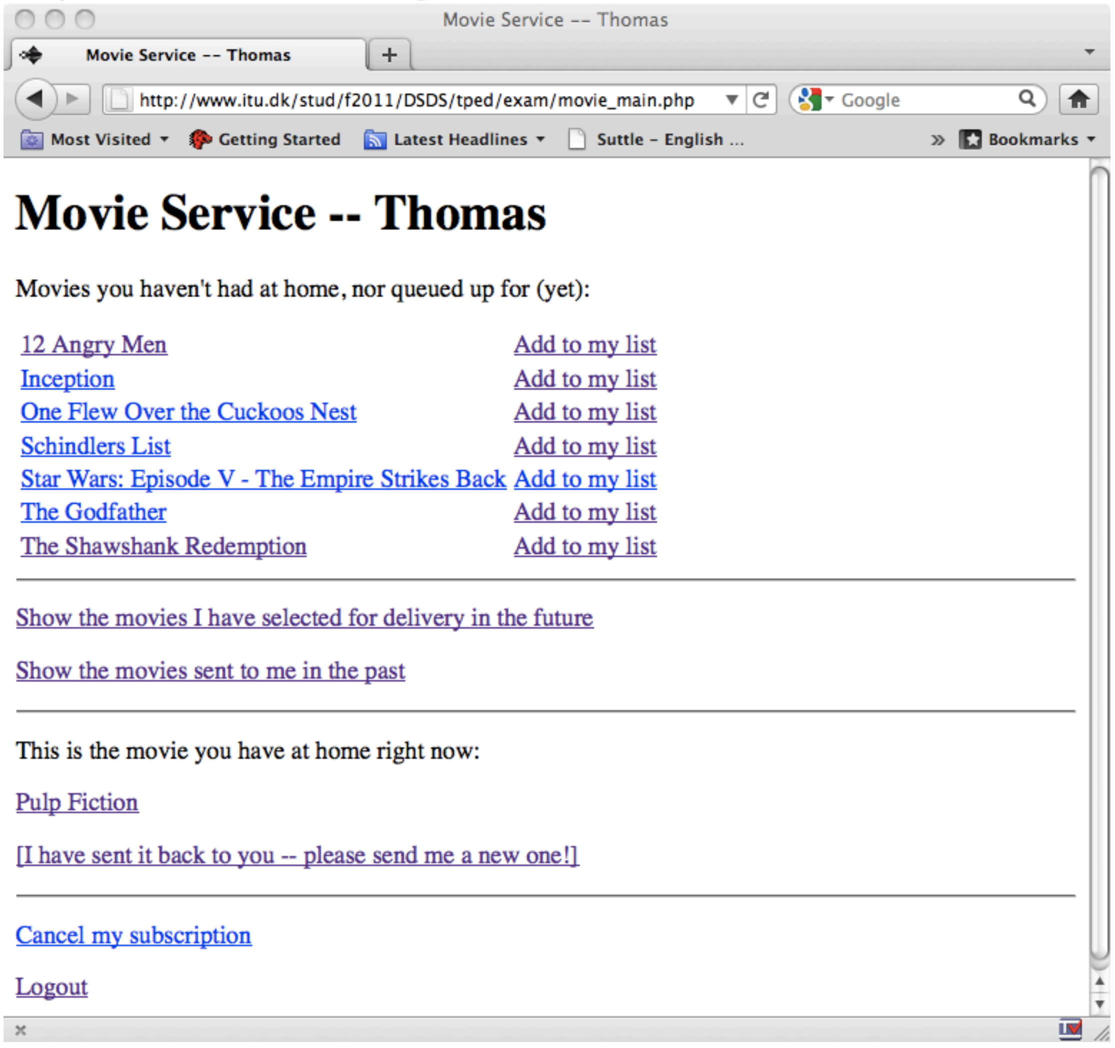
1. The web service should allow new users to subscribe to DVDs by creating a new account. Email and user-chosen password should be used as credentials. After a user (DVD subscriber) has logged in successfully, personal user data (e.g. subscriber id) should be handled jointly by the PHP scripts of the service using the PHP session mechanism. Non-personal user data (e.g. what DVD a subscriber wants to add to the "wanted" list) can be passed using the HTTP POST or GET parameter passing mechanisms.
2. The subscriber should be able to view all available DVDs from the service provider and select, from those, the ones to include on her/his "I want these DVDs to be sent home" list. The list of DVDs should be orderered alphabetically based on the name of each DVD. To avoid confusion and in order to simplify the user interface, the subscriber should a) not be able to add a DVD to the "wanted" list which is already part of that list, b) not be able to add a DVD which has already been delivered home in the past, c) not be able to add the DVD which the subscriber already has at home.
3. The subscriber should be able to view her/his "I want these DVDs to be sent home" list and delete DVDs from that list at wish (i.e. to regret to have chosen a specific movie to be delivered home).
4. The subscriber has to be able to see what DVD that she/he is currently having at home (and/or is on its way from the provider to the subscriver's mailbox) and be able to signal to the service provider, in some way, that the DVD has been watched and sent back.
5. As soon as the subscriber signals that the DVD that the subscriber had in her/his possession has been sent back to the service provider (e.g. by clicking on a specific link or checking a radio button and submitting a form), the web service should automatically send another DVD from the "I want these DVDs to be sent home" list to the subscriber.
6. The actual sending of a physical DVD is simulated by simply updating the "I have this DVD at home" part of the subscribers web interface with the new DVD that the web service chose to send.
7. Whenever the title of a DVD movie is shown on a web page belonging to your service, it should be clickable and linked to an external web resource that shows more information about the move (e.g. www.imdb.com).

Your service does <u>not</u> need to:

- Provide means to the service provider to insert DVDs into the service provider database. Instead, you should manually insert (using appropriate MySQL INSERT commands in the MySQL text client) at least 10 DVD entries into your system database. Here is example code for inserting 10 really good movies:

```
INSERT INTO dvd VALUES
(NULL, 'The Shawshank Redemption', 'http://www.imdb.com/title/tt0111161/'),
(NULL, 'The Godfather', 'http://www.imdb.com/title/tt0068646/'),
(NULL, 'The Good, the Bad and the Ugly', 'http://www.imdb.com/title/
tt0060196/'),
(NULL, 'Pulp Fiction', 'http://www.imdb.com/title/tt0110912/'),
(NULL, '12 Angry Men', 'http://www.imdb.com/title/tt0050083/'),
(NULL, 'Schindlers List ', 'http://www.imdb.com/title/tt0108052/'),
(NULL, 'One Flew Over the Cuckoos Nest', 'http://www.imdb.com/title/
tt0073486/'),
(NULL, 'Inception', 'http://www.imdb.com/title/tt1375666/'),
(NULL, 'The Dark Knight', 'http://www.imdb.com/title/tt0468569/'),
(NULL, 'Star Wars: Episode V - The Empire Strikes Back', 'http://
www.imdb.com/title/tt0080684/');
```

A Screenshot of the main page of a working solution is shown below. Note that this is just one of many possible examples of how the main page of the service could be designed in order to meet the requirements.



Movie Service -- Thomas

Movies you haven't had at home, nor queued up for (yet):

12 Angry Men	Add to my list
Inception	Add to my list
One Flew Over the Cuckoos Nest	Add to my list
Schindlers List	Add to my list
Star Wars: Episode V - The Empire Strikes Back	Add to my list
The Godfather	Add to my list
The Shawshank Redemption	Add to my list

Show the movies I have selected for delivery in the future

Show the movies sent to me in the past

This is the movie you have at home right now:

Pulp Fiction

[I have sent it back to you -- please send me a new one!]

Cancel my subscription

Logout

Note: The exam task is deliberately loosely defined in order to force you to make use of many of the design skills acquired on the DSDS course. Use your common sense to resolve any ambiguities. Whenever you find the requirements specification above unclear, choose an interpretation and state that interpretation clearly in the file where you describe your design (`description.txt`).

Exercise 1: design of data model

Exercise 1a) Entity-Relationship diagram (7%)

Destination files: `exam/1a_ERdiagram.jpg, description.txt`

Construct an Entity-Relationship diagram that shows which MySQL tables you intend to have in your service solution and how they relate to each other. Remember to do your design so that data redundancy is kept at a minimum. Write a couple of sentences describing your design under the heading “Exercise 1a E-R diagram” in `description.txt`.

Exercise 1b) MySQL table definitions (10%)

Destination files: `exam/1bc_MySQL_tables.sql, description.txt`

Define the MySQL tables shown in your ER-diagram in the previous exercise using CREATE TABLE statements. Remember to make use of for instance primary keys, foreign keys, innoDB, etc. Write a couple of sentences describing important aspects of your tables under the heading “Exercise 1bc MySQL table definitions” in `description.txt`.

Exercise 1c) Inserting some data (3%)

Destination file: `exam/1bc_MySQL_tables.sql`

In the end of the .sql file you created for the previous exercise, create table entries in your database using MySQL INSERT commands to the extent specified in the requirements specification so that your service can be properly tested out.

Exercise 2: definition of database transactions (20%)

Destination file: `exam/2_MySQLqueries.sql`

Construct a set of SQL queries that you believe your PHP scripts will need to make use of in order for your service to provide the functionalities specified in the requirements specification earlier. This includes both retrieval of selected data from your tables and insertion of new data into them. For each of the MySQL queries you present, write a sentence or two explaining what it actually does.

Exercise 3: design of the web structure (site map) (20%)

Destination file: `exam/3_sitemap.jpg, description.txt`

Now the time has come to design the overall sitemap of the web service. What web pages and PHP scripts are needed and what variables need to be sent between them? Draw a diagram of your suggested web site structure that includes information about the name of each script/page/file and information about the passing of form variables from one script/page/file to another. Name the files and variables as you seem fit with the constraint that the file generating the front page of the service (the file which subscribers of the service will access first in order to log in) should be called “movieservice.php”. Describe the general flow of interaction and the role of the different files in your web service in a couple of sentences under the heading “Exercise 3 site map” in `description.txt`.

Exercise 4: construction of PHP scripts and HTML pages (40%)

Destination files: `exam/movieservice.php`, `exam/[yourfilenames.php, yourfilenames.html]`, `exam/includes/[yourfilenames.php]`

In this exercise, you are asked to implement (construct) the PHP and HTML files defined in exercise 3 of which some will make use of the database transactions (MySQL queries) defined in exercise 2. Keep in mind the code quality factors listed in the “Grading” section earlier in this exam.

Checklist

If you have done all four exercises of this exam, you should have the following files in your online folder and on your CD-ROMs:

- `exam/1a_ERdiagram.jpg`
- `exam/1bc_MySQL_tables.sql`
- `exam/2_MySQLqueries.sql`
- `exam/3_sitemap.jpg`
- `exam/description.txt`
- `exam/index.html`
- `exam/includes/[yourfilenames.php]` (optional)
- `exam/movieservice.php`
- `exam/[yourfilenames.php]` (the PHP scripts of your web service)
- `exam/[yourfilenames.html]` (optional, in case you used HTML)