

**CS 353 Spring 2019**  
**Homework 4**  
**Programming Assignment**  
**Due: April 10, till 23:59.**

In this assignment, you will first write a simple Java application to connect to a MySQL database and modify it. Next, you are required to design a basic internet application using PHP on top of this database. In what follows, we explain each of these stages in detail.

**Part 1: Connecting to the database with a Java application**

For this part of the assignment, you will first create an empty database using MySQL console, and then write a simple Java application to connect and interact with this database.

- *Connecting to the database:* Connect to the `dijkstra.ug.bcc.bilkent.edu.tr` machine, which runs MySQL server (Use **SSH** or **PUTTY** as dijkstra allows only “secure” shell connections but not telnet!). Next, in the shell prompt, execute the command

```
shell> mysql --user=USERNAME --password=PASSWORD
```

To connect to the dijkstra server and MySQL, you need a username and password, which should be e-mailed to you by now. This MySQL account only allows you to create a database exactly named as your “username”. For instance, if your username is “ahmetc” then you are only allowed to create a database with name “ahmetc”. A database names as your username is already created. Check your databases with ‘show databases;’ command in mysql. For the user “ahmetc” the output is as follows:

```
mysql> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| ahmetc             |
| test               |
+-----+
```

If you don’t have the database same as your name create it with the command:  
Create database your\_user\_name; (create database ahmetc; for the above example)  
Contact to the TA immediately, if you encounter any problem creating your database.

Next, write a Java program that does the following:

- *Connection:* Successfully connect to your MySQL database using Java and your MySQL account information. You can easily find the JDBC driver for MySQL on the Web.
- *Table creation:* Create the following relations in your database. **REMARK:** We want to set the foreign key relationships, and in MySQL this is supported with **InnoDB** tables. Do NOT forget to set table type to InnoDB and set foreign keys for each table appropriately. (Again, see MySQL reference manual for details).

**customer**(cid: CHAR(12), name: VARCHAR(50), bdate: DATE, address: VARCHAR(50), city: VARCHAR(20), nationality: VARCHAR(20))

**account**(aid: CHAR(8), branch: VARCHAR(20), balance: FLOAT, openDate: DATE)

**owns**(cid: CHAR(12), aid: CHAR(8))

- *Table population:* Insert into the newly created relations the following records.

customer					
cid	name	bdate	address	city	nationality
20000001	Cem	10.10.1980	Tunali	Ankara	TC
20000002	Asli	08.09.1985	Nisantasi	İstanbul	TC
20000003	Ahmet	11.02.1995	Karsiyaka	İzmir	TC
20000004	John	16.04.1990	Kızılay	Ankara	ABD

account			
aid	branch	balance	openDate
A0000001	Kızılay	2,000.00 TL	01.01.2009
A0000002	Bilkent	8,000.00 TL	01.01.2011
A0000003	Çankaya	4,000.00 TL	01.01.2012
A0000004	Sincan	1,000.00 TL	01.01.2012
A0000005	Tandogan	3,000.00 TL	01.01.2013
A0000006	Eryaman	5,000.00 TL	01.01.2015
A0000007	Ümitkoy	6,000.00 TL	01.01.2017

owns	
cid	aid
20000001	A0000001
20000001	A0000002
20000001	A0000003
20000001	A0000004
20000002	A0000002
20000002	A0000003
20000002	A0000005
20000003	A0000006
20000003	A0000007
20000004	A0000006

- Print the contents of each table on screen.
- While doing the above tasks, give meaningful error messages if necessary.
- Recall that your program may be executed several times during your code development and during the grading. So, you should check whether the database and/or tables actually exist before trying to create them to prevent MySQL errors. If they exist, first drop them and then re-create them.

## Part 2: A simple Web based application using PHP

For this part of the assignment, you are required to design an internet application on top of the simple bank database you created above. In particular, you will design and implement a website that involves the middle and presentation tiers of a web application and makes use of the bank database at the data management tier. The website should include the following pages:

1. **Index page:** All users start at a common log on page, where each user is expected to enter his/her login and password. We assume that customer names serve as logins and customer id's serve as passwords (for instance, customer "Asli" can login by entering "asli" and "20000002" as her login and password). Note that, login is case-insensitive. Give an appropriate error message if the log-on operation fails. In addition, give an error if one or both of the input fields left blank and "login" button is clicked (i.e., use a simple Java Script code to check).

2. **Customer welcome page:** Display the welcome page which shows all the accounts (show info for all (aid, branch, balance and openDate)) of that customer. Next to each of these accounts, display a link, namely "close", so that the customer can close this account. When the student clicks on the "close" link, display either an "error message" or a "successful deletion" message, and allow the customer to return to customer welcome page, again (which, of course, doesn't display the tuple(s) deleted from the owns table).

At the bottom of this page, also provide a link called "money transfer". When this link is clicked, open a new page. At this page, show the accounts of this customer again and also all other accounts in your bank database so that the customer can choose which account to make transfer (you can use combo boxes and/or radio buttons for a more sophisticated GUI). Note that you should allow customer to make a transfer to one of his/her accounts. Provide three input fields (fromAccount, toAccount and transferAmount) and a "submit" button, so that the customer can specify the accounts for the transfer (e.g., by typing the account ids) and the transfer amount. Money transfer operation is initiated by clicking the "submit" button. You should first check that the "fromAccount" belongs to the current customer and if it does not give an appropriate error message. Also give a warning if any of these three input fields is blank (you can use JavaScript here). Then, check that the fromAccount balance is enough for the transfer amount given. Otherwise we give an appropriate error message. If all of these conditions are satisfied, allow the transfer by making account balance updates of the corresponding accounts and give an appropriate message indicating the successful completion of the money transfer. Allow user to return to the previous page that displays money transfer screen, to allow him/her make more money transfers. At every page, remember to put a link to an appropriate previous page, so that the customer can go back without doing any modifications. Also, you may need to keep track of the current customer id through pages, as well.

Finally, a logout link in all appropriate pages will be useful.

**IMPLEMENTATION** You should use PHP to implement the application logic (except the Java Script codes used to check blank form fields at the client side). Your application should connect to your MySQL database prepared in the first part of this assignment.

Test your web pages under your public html directory (at *dijkstra*) in a folder called *surname\_name* (if the public\_html directory does not exist, create it at your home directory and set its r-w-x permissions appropriately). You will copy and submit this folder as described below, and we will copy it under our own public\_html and execute there. To allow this, please use *relative links* in your web pages (i.e., if you include links to an absolute page `http://.../~ahmet/can_ahmet /index.php`, it would not work under my public html folder. If this happens, **you won't get any grade** from this part of the assignment).

### What to submit?

You will submit a WinRAR file including two folders, part1 and part2. Folders must be **exactly** named as *surname\_name\_p1* and *surname\_name\_p2* (*please do not use Turkish characters*). The RAR file must be named as *surname\_name*. Each folder should include the following:

- part1 folder: In this folder, provide the Java source code file(s).
- part2 folder: This folder should include all your PHP files. We will copy them to our public\_html and then execute there.

For both parts, grading will be based on the **execution** of your code.

### Where to submit?

You will use the Moodle course page for submission of this assignment. If you encounter a problem during submission, please send your assignment by email to a TA.