

## TRIANGULATION APPROACH TO TEACH DATABASE APPLICATION PROGRAMMING

*Thomas L. Ngo-Ye, Alabama State University, tngoye@alasu.edu*  
*Jiin Wang, Alabama State University, jwang@alasu.edu*

### ABSTRACT

*In this paper, we share our first-hand experience of how we prepared and taught a new database application programming course. We proposed and implemented a new teaching approach – Triangulation. Instead of teaching students just one platform such as ASP.NET, we introduced three popular architectures (Java, PHP, and ASP.NET) in a single database application programming course. To assist faculties interested in teaching the database application programming course, we present the key teaching materials, projects, and test questions that we created. We also report tips and lessons learned that could be useful for teaching database application programming.*

**Keywords:** Database Application Programming, Triangulation Teaching Approach, Three-tier Architecture, Java Servlet, PHP, ASP.NET

### INTRODUCTION

Most modern websites are powered by database management systems (DBMS) and database application programs behind scene. Database application programming is a critical technical skill for Information Systems (IS) students entering job market. A database course is offered by all business schools with an IS program and is recommended to be taken by all business majors (Wang & Wang, 2016). However, database application programming is rarely offered as a standalone course. In general learning to write programming code is perceived to be hard by many students (Blayney, 2009; Ramalingam, LaBelle, & Wiedenbeck, 2004; Robins, Rountree, & Rountree, 2003). Learning and teaching database application programming is even more challenging due to the complexity inherent in the n-tier database application architecture as well as the lack of a dedicated textbook on the topic. In this paper, we report our undertaking of teaching a new database application programming course. We hope that the teaching approach, materials, projects, tests, and tips gained from our first-hand experience will be helpful to faculties planning to teach database application programming.

### Two Database Courses

In our undergraduate IS program, we are currently offering two database courses: CIS 404 Database Administration and CIS 461 Advanced Database Systems. CIS 404 can be mapped to IS 2010 model curriculum's core course – "IS 2010.2 Data and Information Management" (Topi, et al., 2010). CIS 404 covers the following topics: basic concept of relational database systems, Structured Query Language (SQL), relational model and normalization, Entity-Relationship model, database design, database administration (concurrency control, transaction, database security, database backup and recovery), and data warehouses and business intelligence. CIS 461, as a second course in database management systems, builds on the fundamentals of relational database management systems covered in CIS 404. The primary focus of CIS 461 is on integrating database systems into the web environment using technologies such as Java, PHP, and ASP.NET. CIS 461 can be related to IS 2010 model curriculum's elective course – "Application Development". CIS 461's student learning objectives include: 1) Demonstrate an understanding of relational database using the SQL Server platform, which includes tables, queries, views, and forms. 2) Demonstrate an understanding of using Java technology for developing data-driven web sites. 3) Demonstrate computer programming skills through the use of Dreamweaver for developing PHP applications for the Internet. 4) Demonstrate computer programming skills through the use of Visual Studio 2015 for developing ASP.Net applications for the Internet.

### **Challenges of Teaching Database Application Programming**

First, a functioning modern database application system involves multiple components – application programs, DBMS and database, application server, and web server. Not only individual components should work, but also the communication and interface among them should work. Therefore, it is challenging for students to piece together all the required elements to have a working database application system. Second, there is no textbook dedicated on this topic from publishers. We searched on Pearson Education online product website – <https://www.pearsonhighered.com/sign-in.html> and could not find a proper textbook. For other IS topics, it is easier to identify a suitable textbook. For example, System Analysis and Design course can use (Kendall & Kendall, 2014); Network and Telecommunication course can use (Panko & Panko, 2013); Project Management course can use (Vaidyanathan, 2013); Introduction to Database Management course can use (Kroenke, Auer, Vandenberg, & Yoder, 2018). A proper textbook and high quality instructor resources (PowerPoint slides, instructors’ manuals, exercises and solutions, and test bank) from publishers are very valuable to teachers. Without them, teachers have to invest large amount of time and effort to develop lecture notes, exercises, projects, and exams. Third, because in our IS program the only prerequisite for CIS 461 is CIS 404, many students have not taken Java or Internet programming (PHP) course yet. Hence, some students are not familiar with programming languages in general and object-oriented programming in particular. Given the importance of database application programming skills and the challenges of teaching and learning database application programming, it is natural to raise the salient and pressing *research question: how to overcome the challenges of teaching database application programming?*

### **PREPARE TEACHING DATABASE APPLICATION PROGRAMMING**

Although we cannot find a perfect textbook for teaching database application programming, we identify three textbooks with relevant materials and list them in our syllabus as required textbooks for students. We gathered teaching materials from these textbooks and adapt them for teaching. (Kroenke & Auer, 2015) is a great text for the Introduction to Database Management course. However, among its eight chapters, only chapter 7 “Database Processing Applications” focuses on database application programming. This chapter gives a good conceptual overview on the topic and presents a few PHP code examples. (Powers, 2011) mainly concerns with advanced topics on Dreamweaver and PHP. Its lesson 3 “A Quick Crash Course in PHP” is valuable for students who are unfamiliar with PHP. Its lesson 5 “Designing and Building Your Own Database” describes how to use Oracle MySQL database. Its lesson 6 “Generating PHP Automatically with Server Behaviors” presents Dreamweaver’s built-in Server Behaviors to generate PHP database code using wizard-like GUI. It minimizes the amount of code programmers have to manually write for connecting to a database, inserting, displaying, updating, and deleting records. However, the power of Server Behaviors is limited to several predefined database functions. Another issue of Server Behaviors is that it uses older technology such as `mysql_pconnect()` and `mysql_query()`, rather than newer technology like `mysqli_connect()` and `mysqli_query()`. (Spaanjaars, 2013) is a nice book to teach novice ASP.NET. Among its 19 chapters, only chapter 12 “Introducing Databases” and chapter 13 “Displaying and Updating Data” are most relevant to database application programming. We went over all the relevant chapters of these three textbooks and practiced all the codes in those chapters. We also wrote down study notes for (Powers, 2011) and (Spaanjaars, 2013), because, unlike (Kroenke & Auer, 2015), no PowerPoint slides are available for them.

In addition to the above three textbooks, we also gathered teaching materials from Internet. For PHP, we refer to <http://www.w3schools.com/php/>, <http://www.php.net/>, and <http://www.tutorialspoint.com/php/>. For Java, we refer to <http://www.oracle.com/technetwork/java/javaee/servlet/index.html>, <http://www.sitepoint.com/print/java-servlets-1> (Yank, 2001), <http://www.herongyang.com/JDBC/JDBC-ODBC-Bridge-Driver.html> (Yang, 2017), and [http://pdhut.freesevers.com/dev\\_env/servletjsp/jswdk/index.htm](http://pdhut.freesevers.com/dev_env/servletjsp/jswdk/index.htm). For ASP.NET, we refer to <https://www.asp.net/>, which has free videos and interactive tutorials for learning ASP.NET.

We spent a lot of time and effort to prepare lecture notes and programming exercises based on the available programs in textbooks and online resources. We combined and adapted these content to form our own teaching materials. We discovered various technologies being used in database application systems. For example, Java Servlet, JSWDK (JavaServer Web Development Kit), and Access database together make a system. PHP, Apache, and MySQL Server make a complete system. ASP.NET, IIS (Internet Information Server), and SQL Server make a complete system. In addition to the observed combinations, we pondered about the possibility of other combinations. This leads us to

explore the theoretical base of three-tier architecture and in consequence to propose the triangulation approach of teaching database application programming.

### THREE-TIER ARCHITECTURE AND TRIANGULATION

#### Three-tier Architecture

Modern database application systems normally adopt a three-tier architecture. The three distinct tiers are presentation tier, business logic tier, and data storage and data access tier. According to (Kroenke & Auer, 2015), the word tier means a separate computer that runs a portion of a web database processing environment. The user's web browser running on the user's own PC is one tier, and a Web Server along with an Application Server running on another computer is another tier. If the DBMS supporting the web application runs on yet another computer, this is a third tier. This environment is referred to as a three-tier architecture. Usually, if the Web Server and the DBMS are running on the different computers, the system has a three-tier architecture. In a real-world production environment, a three-tier architecture is manifested in three different physical computers. However, in teaching and learning environment, normally we have all three tiers physically reside in one single PC, although logically there are still three distinct tiers. Because database application systems can be divided into three distinct tiers, in theory we can mix and match any technology in the three-tier architecture.

**Table 1.** Database Application Systems Components and Technologies

Application Programming Language	Web Server	Application Server Technology	DBMS API Interface Standard	DBMS
Java Servlet or JSP (JavaServer Pages)	JSWDK	JSWDK and Java SE 7 (JDK 1.7)	JDBC-ODBC Bridge	Any database supporting JDBC-ODBC Bridge
	Apache	Tomcat and Java SE 8 (JDK 1.8)	JDBC	Any database supporting JDBC
PHP	IIS or Apache	PHP Parser	ODBC	Any database supporting ODBC
	Apache		MySQLi	MySQL
	Apache		PDO (PHP Data Objects)	Twelve databases supporting PDO
ASP.NET (VB.NET)	IIS	.NET	ADO.NET	Any database supporting ADO.NET

While preparing database application programming course, we discovered three popular application programming languages – Java, PHP, and ASP.NET. In principle, any of these three technologies can be coupled with any web server, application server, and DMBS. Java Servlet or JSP extend and enhance web servers. They are platform-independent method for building web-based applications. Apache Web Server and Tomcat Application Server together is a preferred open-source combination supporting Servlet API and JSP.

For a Java-based application to communicate with a database, the database preferably directly supports JDBC. While JDBC-ODBC Bridge is not recommended for a production system, it is still used in legacy database systems. JDBC-ODBC Bridge is also useful for building a quick prototype for proof-of-concept demonstration. In the context of teaching database application programming, Access is a convenient and easy-to-use DBMS product. JDBC-ODBC Bridge is used to connect Java web application to Access. Instead of using Apache/Tomcat, we chose JSWDK as the Web Server and Application Server for Java application. JSWDK 1.0.1 supports JSP 1.0 and Servlet API 2.1. While Apache/Tomcat is more powerful than JSWDK, JSWDK has the advantages of being lightweight (jswdk1\_0\_1-win.zip is only 746 KB!) and simple to install, run, and use. Thus, JSWDK is still suitable for the purpose of teaching students Java-based database application programming. As we use JDBC-ODBC Bridge in Java applications, any database that supports ODBC should work. In theory, as long as we can setup correct ODBC data source name, not only Access, but also MySQL and SQL Server can also be used in Java applications. In Java SE 8, the JDBC-ODBC Bridge driver is not included any more. However, Java SE 7 still includes the JDBC-ODBC Bridge driver and supports JDBC API 4.1 (Yang, 2017). Therefore, we opt to use Java SE 7, rather than Java SE 8 due to the need of supporting JDBC-ODBC Bridge.

For a PHP program to communicate with a database, there are three options at large. The first option is through ODBC. As long as a database supports ODBC, it can be interfaced by a PHP program. The second option is MySQLi extension (the “i” stands for improved), which only works with MySQL databases. The MySQL extension was used in earlier versions of PHP. However, MySQL extension was already deprecated in 2012, although it is still used in Dreamweaver’s Server Behaviors. MySQLi is object-oriented, but it also offers a procedural API. If it is decided that the DBMS will always be MySQL and never changes to other product, MySQLi is a sound choice. However, if we do change our mind later to switch to a different DBMS such as Access, we need to rewrite the entire PHP code – SQL queries included. The third option is PDO, which works on twelve different database systems, including Access, as long as PDO drivers are installed and activated in the PHP environment. Hence, if for some reason we need to switch our project to use another DBMS, such as Access, PDO will make the database migration process easy. We only need to change the connection string and a few queries. PDO is object-oriented only. While IIS Web Server can be used with PHP, Apache is commonly used with PHP, along with MySQL. Although PHP, Apache, and MySQL can be installed and used separately, to run PHP program, we employ an all-in-one package – WAMPServer (Windows OS, Apache, MySQL, and PHP) from <http://www.wampserver.com/en/>.

For an ASP.NET program to communicate with a database, it uses ADO.NET technology, which can communicate with many types of DBMS. In the production environment, normally we would use a standalone SQL Server at the database tier and a standalone IIS as the Web Server. After an ASP.NET database application is fully developed and tested, it is deployed in the production SQL Server/IIS environment. However, in the teaching environment, we can use Visual Studio 2015’s built-in local IIS Express version and SQL Server Express version as the Web Server and Database Server. Hence, we can create and run an ASP.NET database application in Visual Studio 2015 without the need of additional software. In principle, as long as a database can be configured and connected as a “sqlDataSource” in Visual Studio 2015, it can be used in ASP.NET application. Therefore, in addition to Visual Studio 2015’s built-in local SQL Server Express version, we can definitely use Access as the data source.

### **Triangulation**

Since modern database application systems use the three-tier architecture and we can use different technologies at different tier, it opens many possibilities of combining various technologies to construct a functioning system. Instead of just focusing on one fixed configuration, such as ASP.NET/IIS/ADO.NET/SQL Server, we introduce students several different ways to construct a database application system. More importantly, we encourage students to explore other possibilities. We call this teaching approach “triangulation”, which means that students can take different routes of technology and all arrive at the outcome of building a functioning system that satisfies the requirements. There are at least three potential benefits of this triangulation approach. First, it explicitly enhances students’ theoretical understanding of the three-tier architecture. Students can see that different technologies can be adopted for a particular role of each tier. Second, it exposes students to a variety of technologies and products. This approach better prepares students for future professional career in database application system development. In real-world, a developer might be asked to use a different technology at different projects or companies. Therefore, it is better for students to have at least basic understanding and working knowledge of different technologies. Third, this teaching method has potential pedagogical advantage that students have multiple chances to practice database application programming. Students can apply what they learn from one approach, for example PHP-ODBC, and apply it to a new approach such as PHP-MySQLi. This potential transfer of knowledge from one setting to another will deepen students’ understand of the topic. If students just blindly follow the step-by-step instruction to complete a hands-on project without a chance to apply the knowledge to a new setting, it is very doubtful how much knowledge the students can retain (Hingorani, Beasley, & Bradford, 2015). Our triangulation teaching approach gives students ample opportunities to apply, combine, and transfer knowledge. Moreover, triangulation also helps students to compare and contrast different technologies and appreciate the subtle difference among them.

Throughout the whole CIS 461 course, we emphasized to students that different technologies can all achieve the same goal. We implemented the triangulation teaching approach at different stages of database application system development. At the DBMS tier, we introduced students Access, MySQL Server Database in WAMPServer, and Microsoft SQL Server (local Express version). We used Access, phpmyadmin (a free web-based tool to manage MySQL Server), and Microsoft SQL Server 2016 Management Studio to create a small database and several tables, perform data entry, and construct some SQL queries. We also implemented referential integrity constraint and exported and imported data in all three DBMS. Practicing with three DBMS helped students realize the commonality

among them to perform basic functions of relational DBMS. Moreover, students became aware some differences among the three DBMS. In Access DBMS, we can have only one Access database at a time. On the other hand, in both MySQL Server and SQL Server, they can have multiple databases in a server at the same time. In Access, SQL queries, forms, and reports are all saved as part of the Access database file. In SQL Server, those applications have to be saved outside SQL Server as separate files. To implement “view” in Access, we can create a SQL query and save it. Then we can refer to the query just like a “view” in SQL Server. In Access, we have to execute one SQL statement at time, while in MySQL and SQL Server we can execute many SQL statements in a batch mode with a SQL script. Moreover, the tasks of importing and exporting data are very different in these three DBMS.

At the Application Server and application program tier, we introduced students Java, PHP, and ASP.NET (VB.NET). We use Adobe Dreamweaver CS5.5 as the IDE to write Java and PHP programs. While Dreamweaver supports ASP.NET, we use Visual Studio 2015 to write ASP.NET programs. Next, we report the most important practical tips and lessons we learned from building database application system with these three technologies.

### **PRACTICAL TIPS AND LESSONS LEARNED**

Because there is no textbook or formal document for compiling, deploying, and running a web-based Java and PHP database application, we documented the critical steps of the process and assembled them into a lecture note. Next, we present the most important steps. The general background of the application is that we want to build a web application that fulfills four basic functions – inserting a new record in a database table, displaying all records in a table, updating a record in a table with user provided information (primary key value and necessary column values), and deleting an existing record in a table with user provided primary key value. For each function, in both Java and PHP, we use two files. One is the HTML file providing web interface for users to enter data. The second file is the application program (Java Servlet or PHP) processing the user input data posted by the HTML page via web form. Here, we employ the “CustomerOrderInsert” use case to explain the process.

#### **Detailed Procedure for Configuring and Running Java Servlet Database Applications**

First, we need to create necessary HTML and Java source code files in Dreamweaver. “index.htm” is the homepage for the application, which links to “GeneralTableServlet.html” and “CustomerOrderInsertServlet.html”. “GeneralTableServlet.html” asks user to input ODBC DSN (data source name) and table name. “CustomerOrderInsertServlet.html” asks user to input all the data for a new record. Each HTML file’s form action is linked to a Java Servlet. In “GeneralTableServlet.html” we set <form method="GET" action="servlet/GeneralTableServlet">. In “CustomerOrderInsertServlet.html”, we set <form method="GET" action="servlet/CustomerOrderInsertServlet">. “GeneralTableServlet.java” displays the content of any specified database table. “CustomerOrderInsertServlet.java” inserts the new record in the “CustomerOrder” table.

Next, we need to create a new Access database (CIS461SP2017Final.accdb), which contains a new table – “CustomerOrder”. We place it under “C:\temp”. Next, we need to setup a 32-bit ODBC DSN – “CIS461SP2017Final” pointing to “C:\temp\CIS461SP2017Final.accdb”. Next, at the local PC, in Windows Explorer, navigate to the file “C:\Windows\SysWOW64\odbcad32.exe”. Then we right-click it and choose “Create shortcut” and place the new shortcut on desktop. Next, we right-click the new shortcut to “odbcad32.exe” on desktop and choose “Run As Administrator”. Next, we select “System DSN” Tab and click “Add” button. Next, we need to select the right 32-bit database driver by scrolling down the list and click “Microsoft Access Driver (\*.mdb, \*.accdb)”. Then we click “Finish” button. In “Data Source Name” textbox, we type in “CIS461SP2017Final”. Under “Database:” we click “Select” button and navigate to “C:\temp\CIS461SP2017Final.accdb” and select it. Then we click OK, OK, and OK to finish configuration of 32-bit ODBC DSN for CIS461SP2017Final.accdb.

Next, we need to download and install the 32-bit version of Java SE 7. Since most Access is 32-bit version, we have to match it with 32-bit ODBC Data Source Administrator and 32-bit JDK. The default path for JDK is “C:\Program Files (x86)\Java\jdk1.7.0\_80\bin”. We use “C:\temp” as the working directory. We also need to download and install JSWDK. After downloading “jswdk1\_0\_1-win.zip” to “C:\temp”. We unzip it there using “Extract All” option and accept all the default setting.

Next, we copy all HTML files from “C:\temp” to “C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\webpages\”, which is the root folder of JSWDK Web Server to hold all webpages.

To compile and deploy Java Servlets, we need to perform the following steps:

Start DOS Command Prompt: Start → in the Search programs and files textbox type “cmd”  
Enter into working directory (folder) Type: “cd C:\temp”  
Set path for the DOS Window by typing: “set path=C:\Program Files (x86)\Java\jdk1.7.0\_80\bin;%path%”  
Set classpath for the DOS Window by typing:  
“set classpath=.;C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\lib\servlet.jar;C:\Program Files (x86)\Java\jdk1.7.0\_80\lib\tools.jar;%classpath%”  
Next, compile Java Servlets by typing in DOS Command Prompt:  
javac GeneralTableServlet.java  
javac CustomerOrderInsertServlet.java  
Copy the above two newly compiled Java Servlet class files from C:\temp to “C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\webpages\WEB-INF\servlets”, which holds all compiled Java Servlets.

To start JSWDK server and run Java Servlets, we need to perform the following steps:

In the DOS Command Prompt type:  
“cd C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1”  
“startserver.bat”  
In a web browser type in <http://localhost:8080/index.html> to test JSWDK 1.0.1. If <http://localhost:8080/index.html> works, we can test Java Servlet database applications by running <http://localhost:8080/index.htm>. Then we can test all Java Servlets and HTML pages.

To stop JSWDK server gracefully, we can type “stopserver.bat” in DOS Command Prompt.

After we deploy compiled Java Servlets, if we modify the Java source code, we have to re-compile the Java file and copy the new Java Servlet class file to “C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\webpages\WEB-INF\servlets”. Then we need to stop JSWDK server and start JSWDK server again to make the new Servlet take effect.

Finally, instead of using the default port number of 8080 in JSWDK server, we can re-configure it to use port 80. We can open “C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\webserver.xml” in Notepad, and search for “8080” and locate line 207: port NMTOKEN “8080”. Next, we replace it with: port NMTOKEN “80”. Then we save “C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\webserver.xml”. Next, we can restart JSWDK server by stop it and start it again. Then we can use <http://localhost/index.htm> rather than <http://localhost:8080/index.htm>.

### **Tip for Configuring and Running PHP Database Applications**

To setup a 32-bit ODBC DSN to connect to a MySQL Database (CIS461SP2017Final), we need to perform the following steps:

First, we need to download the 32-bit WAMPServer from <http://www.wampserver.com/en/> and the 32-bit MySQL Connector/ODBC from [http://dev.mysql.com/downloads/connector/odbc/Windows \(x86, 32-bit\), MSI Installer \(mysql-connector-odbc-5.3.7-win32.msi\)](http://dev.mysql.com/downloads/connector/odbc/Windows%20(x86,%2032-bit).msi). Then we install the 32-bit MySQL Connector/ODBC on a PC. Next, we perform the procedure to setup a DSN to link to a MySQL Server, which is very similar to that of setting up a 32-bit ODBC DSN linking to an Access database. The difference is that instead of selecting “Microsoft Access Driver (\*.mdb, \*.accdB)” as the database driver, we choose “MySQL ODBC 5.3 Unicode Driver”. Moreover, we need to setup Connection Parameters as following:

Data Source Name: CIS461SP2017Final  
TCP/IP Server: localhost Port: 3306  
User: root  
Password:  
Database: CIS461SP2017Final  
Then click “Test” button to make sure that Test Connection is successful.  
Finally, click OK and OK to finish configuration of 32-bit ODBC DSN for MySQL database.

With WAMPServer, we put all HTML and PHP files under “C:\wamp\www\”, the default Web Server’s document root for their proper execution. We highlighted the similarity of WAMPServer and JSWDK in terms of a centralized root folder to hold HTML files to students. We also pointed out to students that JSWDK has a dedicated subfolder “C:\temp\jswdk1\_0\_1-win\jswdk-1.0.1\webpages\WEB-INF\servlets” to hold servlets. Moreover, unlike JSWDK and

Java Servlets, after putting a modified PHP program in “C:\wamp\www\”, we do not need to restart WAMP Server to make the new PHP program take effect.

### *Tip for Connecting to an Access Database with PDO*

The following PHP code (as part of PDO\_MS\_Access.php) creates a new connection to an Access database.

```
$DB_Username = ""; //username
$DB_Password = ""; //password
$Database_Path = "C:\Users\home\Documents\CIS461SP2017Final.accdb"; // path to the database file
if (!file_exists($Database_Path)) {
    die("Access database file is not found!");
}
$Connection = new PDO("odbc:Driver={Microsoft Access Driver (*.mdb, *.accdb)};Dbq=$Database_Path;Uid=$DB_Username");
```

When execute `http://localhost/PDO_MS_Access.php`, we encounter an error saying that it could not find PDO ODBC driver. To enable PDO ODBC driver, we uncommented/enabled “`extension=php_pdo_odbc.dll`” in the “`C:\wamp\bin\php\php5.5.12\php.ini`” file. However, even after restarting WAMP Server, it still does not work. We finally found the solution. From Windows GUI WAMP --> PHP --> PHP extensions --> check `php_pdo_odbc`. Then from `http://localhost/?phpinfo=1`, we found that within PDO support, PDO drivers enabled: `mysql, odbc, sqlite`. PDO Driver for ODBC (Win32) is enabled. Then we tested `http://localhost/PDO_MS_Access.php` and it works! Moreover, somehow if we have set `$Database_Path = "C:\temp\CIS461SP2017Final.accdb"`; we get error “Access database file is not found!” Thus we have to put the Access file under the “Documents” folder.

### *Securing PHP Program against SQL Injection Attacks*

Web Application Security is becoming a pressing issue for web database systems. SQL Injection Attacks are attempts to issue SQL commands through a Web interface such as a web form (Kroenke & Auer, 2015). To mitigate the threat from SQL Injection Attacks and Cross-site Scripting Attacks launched against web forms, a basic defense method is form input validation. In (Spaanjaars, 2013)’s chapter 9 “Validating User Input”, it discusses ASP.NET’s technique for input validation. Here, we present a simple method for validating user input from a web form with the following PHP code.

```
function cleaning_inputData($inputData) {
    $data = trim($inputData);
    $data = stripslashes($inputData);
    $data = htmlspecialchars($inputData);
    return $inputData;
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $CustomerName = cleaning_inputData($_POST['CustomerName']);
}
```

PHP `trim()` function removes unnecessary characters (extra space, tab, newline) from the user input data. PHP `stripslashes()` function removes backslashes (`\`) from the user input data. PHP `htmlspecialchars()` function transforms special characters to HTML entities by replacing HTML characters like `<` and `>` with `&lt;` and `&gt;`. We need to pass all variables posted by HTML forms, such as `($_POST['CustomerName'])`, through the PHP’s `htmlspecialchars()` function. Then the code is relatively safe. By passing user input data through `htmlspecialchars()` function, `$_SERVER["PHP_SELF"]` exploits can be circumvented. For example, we can use the following PHP code to reduce security risk:

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

### *Tip for Cleaning up MySQL Server*

On a PC, if we install a 32-bit version of MySQL Server, remove it, and then install a 64-bit version of MySQL Server (or vice versa), it will mess up the system in a way that we cannot create a new database with the same name as used in the other version of MySQL Server. If for some reason, `http://localhost/phpmyadmin` does not work, we can perform following steps:

```
Exit WAMP Server
Navigate to “C:\wamp\bin\mysql\mysql5.6.17\data” in Windows Explorer
Remove all the content of “C:\wamp\bin\mysql\mysql5.6.17\data\mysql-bin.index”
Delete files: “C:\wamp\bin\mysql\mysql5.6.17\data\ib_logfile0” and “C:\wamp\bin\mysql\mysql5.6.17\data\ib_logfile1”
Restart WAMP Server and MySQL Server and http://localhost/phpmyadmin will work again.
```

## Lessons Learned about ASP.NET

By default, ASP.NET web form always use POST method. Moreover, an entire ASP.NET webpage always contains exactly one web form by default. There is a big difference between ASP.NET and classic ASP/PHP in terms of web form processing. By default, ASP.NET web form always submits back to itself. However in classic ASP and PHP, we normally set the action attribute of the web form to a second page (\*.asp or \*.php), which processes the form data users have entered and submitted. In ASP.NET, even if we explicitly set a different second page for the action attribute, at ASP.NET run time, it will be revert back to the name of the current ASP.NET page.

One issue with Visual Studio 2015 is that, unlike JSWDK or WAMPServer that uses a fixed port number (8080 or 80), its built-in local IIS express server runs on a random port number at run time. We cannot type URL such as `http://localhost:55786/Management/Reviews.aspx` in a web browser, because the port number (55786) is not known up front. So we have to run the ASP.NET application from within Visual Studio 2015 by right-click the Reviews.aspx in Solution Explorer and choose View in Browser.

Visual Studio 2015, as an advanced IDE, has many advantages. It provides a powerful visual design mode, with which users can drag and drop various built-in controls and configure the application with context-relevant Smart Tasks Panel. The visual design approach minimizes the amount of manual coding by automatically generating most HTML, ASP.NET Server Controls, and SQL codes. Overall, it makes programming work more efficient. Moreover, Visual Studio 2015 is not only a development tool, but also includes a built-in lightweight local Web Server, Application Server, and Database Server. It is convenient to use such all-in-one package for end-to-end database application system development. While Visual Studio 2015 simplifies the development work, the clear theoretical function of each component (web server, application server, and database server) of a database application system is hidden. It blurs the boundary of the different conceptual components of the three-tier architecture. Therefore, Visual Studio 2015 may not be the best tool to teach students the three-tier architecture concept. By using a triangulation approach, we compensate Visual Studio 2015's limitation with Java Servlet-JSWDK and PHP-WAMPServer, which explicitly demonstrate the structure of the three-tier architecture and role of each component. Next, we present a sample of the PHP test programming questions we created.

## PHP TEST PROGRAMMING QUESTIONS

Question 1: Write PHP code to connect to a MySQL Database Server running on "localhost" using MySQLi extension and PHP Procedural version. The user name is "root" and password is NULL. Name the MySQLi connection variable "\$Connection".

Answer:

```
$servername = "localhost";
$username = "root";
$password = "";
$Connection = mysqli_connect($servername, $username, $password);
```

Question 2: Assume the MySQLi and PHP Procedural connection to the MySQL database server running on "localhost" was already created and stored in variable \$Connection. Write PHP code to Close the MySQLi Database connection.

Answer:

```
mysqli_close($Connection);
```

Question 3: Assume the MySQLi and PHP Procedural connection to the MySQL database server running on "localhost" was already created and stored in variable \$Connection. Write PHP code to Create a new MySQL database called "phpExam".

Answer:

```
SQLCREATEDATABASE = "CREATE DATABASE phpExam";
mysqli_query($Connection, $SQLCREATEDATABASE);
```

Question 4: Assume the MySQLi and PHP Procedural connection to the MySQL database server running on "localhost" was already created and stored in variable \$Connection. Assume MySQL database "phpExam" already exists on "localhost". Write PHP code to Create a new Table "customer", which has the following five columns: customer\_id, first\_name, family\_name, phone, email. customer\_id is integer (10), primary key, Auto Number, and UNSIGNED. The other columns are VARCHAR(50) and required.

Answer:

```
$servername = "localhost";
$username = "root";
```



```
$password = "";
$dbname = "phpExam";
$Connection = mysqli_connect($servername, $username, $password, $dbname);
$SQLCREATETABLE = "CREATE TABLE customer (
    customer_id INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    family_name VARCHAR(50) NOT NULL,
    phone VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL
)";
mysqli_query($Connection, $SQLCREATETABLE);
```

Question 5: Write PHP code to Create and Execute a SQL statement that retrieve all the columns and rows of table "customer", using MySQLi and PHP Procedural. Assume the MySQLi connection to the MySQL database "phpExam" was already created and stored in variable \$Connection. Put the results of Executing the SQL statement into \$RecordSet. Assume \$RecordSet holds the content of table "customer" (all rows and columns). Write PHP code to print out the table "customer" in HTML page using table element. Assume table "customer" has the following five columns: customer\_id, first\_name, family\_name, phone, email.

Answer:

```
$SQLSELECT = "SELECT * FROM customer";
$RecordSet = mysqli_query($Connection, $SQLSELECT);
echo "<table border='1'>
    <tr>
        <th>customer_id</th>
        <th>first_name</th>
        <th>family_name</th>
        <th>phone</th>
        <th>email</th>
    </tr>";
while($RecordSetRow = mysqli_fetch_assoc($RecordSet))
{
    echo "<tr>";
    echo "<td>" . $RecordSetRow["customer_id"] . "</td>";
    echo "<td>" . $RecordSetRow["first_name"] . "</td>";
    echo "<td>" . $RecordSetRow["family_name"] . "</td>";
    echo "<td>" . $RecordSetRow["phone"] . "</td>";
    echo "<td>" . $RecordSetRow["email"] . "</td>";
    echo "</tr>";
}
echo "</table>";
```

Question 6: Assume the MySQLi and PHP Procedural connection to the MySQL database "phpExam" was already created and stored in variable \$Connection. Assume table "customer" has the following five columns: customer\_id, first\_name, family\_name, phone, email. Assume \$first\_name, \$family\_name, \$phone, \$email hold the data that an user already entered via InsertNewCustomer.html and posted by clicking the submit button. Write PHP code to Create and Execute a SQL statement that INSERTs a new record into table "customer".

Answer:

```
$SQLINSERT = "INSERT INTO customer ";
$SQLINSERT .= "(first_name, family_name, phone, email) ";
$SQLINSERT .= "VALUES(";
$SQLINSERT .= "'$first_name', '$family_name', '$phone', ";
$SQLINSERT .= "'$email')";
mysqli_query($Connection, $SQLINSERT);
```

Question 7: Assume the MySQLi and PHP Procedural connection to the MySQL database "phpExam" was already created and stored in variable \$Connection. Assume table "customer" has the following five columns: customer\_id, first\_name, family\_name, phone, email. Assume \$customer\_id, \$first\_name, \$family\_name, \$phone, \$email hold the data that an user already entered via UpdateExistingCustomer.html and posted by clicking the submit button. Write PHP code to Create and Execute a SQL statement that UPDATEs an Existing record in table "customer".

Answer:

```
$SQLUPDATE = "UPDATE customer SET ";
$SQLUPDATE .= "first_name = " . $first_name . " , ";
$SQLUPDATE .= "family_name = " . $family_name . " , ";
$SQLUPDATE .= "phone = " . $phone . " , ";
$SQLUPDATE .= "email = " . $email . " ";
$SQLUPDATE .= "WHERE customer_id = " . $customer_id;
mysqli_query($Connection, $SQLUPDATE);
```

Question 8: Assume the MySQLi and PHP Procedural connection to the MySQL database "phpExam" was already created and stored in variable \$Connection. Assume table "customer" has the following five columns: customer\_id, first\_name, family\_name, phone, email. Assume \$customer\_id holds the data that an user already entered via DeleteExistingCustomer.html and posted by clicking the submit button. Write PHP code to Create and Execute a SQL statement that DELETES an Existing record in table "customer".

Answer:

```
SQLDELETE = "DELETE FROM customer WHERE customer_id = " . $customer_id;  
mysqli_query($Connection, $SQLDELETE);
```

### SUMMARY

Database and database application programming are important skills for IS students. The best way to learn these skills is through practicing. In this paper, we present the implementation of a new IT artifact – our new teaching model (triangulation). We believe that by introducing students various technologies and products for achieving the same goal of building a functioning database application system, it helps students gain a deeper understanding of the three-tier architecture concept. It also gives students multiple chances for practicing database application programming by transferring and adapting what they learned in one environment to another. Moreover, this triangulation approach familiarizes students with several important technologies and products, which builds a sound foundation for students to work with those platforms in the future career. The contributions of this study to the literature include not only the proposed triangulation teaching approach, but also the practical teaching materials, tips, lessons learned, and test questions that we refined. We hope that the triangulation teaching method as well as the teaching materials would be helpful to faculties planning to teach database application system development.

Adopting the design-science paradigm as the research methodology, this study starts with identifying a relevant problem (how to overcome the challenges of teaching database application programming) and developing a design artifact (triangulation teaching method). We plan to further develop this research by following the seven guidelines for design science in information systems research (Hevner, March, Park, & Ram, 2004). We will enhance research rigor by conducting a systematic evaluation of design artifact. To validate our design artifact and demonstrate its effectiveness and utility, we intend to conduct a controlled experiment, where one class session is taught in a normal way without triangulation (control group), while another session is taught with triangulation (experiment group). We will report the result of the experiment and new insights in a future study.

### REFERENCES

- Blayney, P. J. (2009). Knowledge gap? Accounting practitioners lacking computer programming concepts as essential knowledge. In G. Siemens, & C. Fulford (Ed.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 151-159). Chesapeake, VA: AACE. Retrieved December 30, 2013, from <http://editlib.org/d/31495>
- Hevner, A., March, S. T., Park, J., & Ram, S. (2004, March). Design Science Research in Information Systems. *MIS Quarterly*, 28(1), 75-105.
- Hingorani, K., Beasley, B., & Bradford, J. (2015). Enhancing student learning of ERP configuration through a Quickbooks tutorial. *Issues in Information Systems*, 16(1), 132-141. Retrieved May 28, 2016, from [http://www.iacis.org/iis/2015/1\\_iis\\_2015\\_132-141.pdf](http://www.iacis.org/iis/2015/1_iis_2015_132-141.pdf)
- Kendall, K. E., & Kendall, J. E. (2014). *Systems Analysis and Design* (9th ed.). Upper Saddle River, New Jersey, U.S.A.: Pearson.
- Kroenke, D. M., & Auer, D. J. (2015). *Database Concepts* (7th ed.). Hoboken, New Jersey, U.S.A.: Pearson.
- Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2018). *Database Concepts* (8th ed.). Hoboken, New Jersey, U.S.A.: Pearson.

- Panko, R. R., & Panko, J. L. (2013). *Business Data Networks & Security* (9th ed.). Upper Saddle River, New Jersey, U.S.A.: Pearson Education.
- Powers, D. (2011). *Adobe Dreamweaver CS5 with PHP: Training from the Source* (1st ed.). Berkeley, CA, U.S.A.: Adobe Press.
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004, September). Self-efficacy and mental models in learning to program. *Association for Computing Machinery SIGCSE Bulletin*, 36(3), 171-175.
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- Spaanjaars, I. (2013). *Beginning ASP.NET 4.5: in C# and VB*. Indianapolis, IN, U.S.A.: John Wiley & Sons, Inc. Retrieved from <http://www.wrox.com/WileyCDA/WroxTitle/Beginning-ASP-NET-4-5-in-C-and-VB.productCd-1118311809.html>
- Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F., Sipior, J. C., & Vreede, G.-J. d. (2010, April). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, 26, 359-428. Retrieved May 13, 2016, from <http://aisel.aisnet.org/cais/vol26/iss1/18>
- Vaidyanathan, G. (2013). *Project Management: Process, Technology and Practice*. Upper Saddle River, New Jersey, U.S.A.: Prentice Hall.
- Wang, S., & Wang, H. (2016, Spring). Renewal of classics: Database technology for all business majors. *Journal of Computer Information Systems*, 56(3), 211-217.
- Yang, H. (2017). *JDBC Tutorials - Herong's Tutorial Examples*. Retrieved May 11, 2017, from herongyang: <http://www.herongyang.com/JDBC/JDBC-ODBC-Bridge-Driver.html>
- Yank, K. (2001, August 4). *Java Servlets – Part 1 Article*. Retrieved May 11, 2017, from sitepoint: <http://www.sitepoint.com/print/java-servlets-1>