Steve Gubenia

CMIS 485 Final Project Approach Document

Building A Secure WebApplication Using VB.NET

Approaches, Testing, and Lessons learned

[Figure 1 The 2015 Top Ten Programming Languages 3](#_Toc425599463)

For the final project I chose to use the Visual Basic language under the ASP.NET framework for my web application. I chose this language and framework for my project because I already had a fully functioning web application with this framework after Project 2. That fact made it a simple matter of adding the select, insert, and update statements for the other database tables as well as the other necessary code for the required report functions, and security measures to full the requirements of this project. Also, I am very familiar with Java, and want a bit of a challenge to design an application in a language that I was still learning.

In an attempt to mitigate SQL injection I first limited the size of the text that can be inserted by the user in each text box. This won’t do much to reduce the risk as the name and address fields are still quite large so an attacker could still use those fields to still inject malicious commands, but ID fields and other fields are quite limited. I also used data validation. I used asp text boxes to accept all user input, but then converted input from the text boxes to the required data type. If an attacker had tried to insert a string command in an integer type an error would occur and an exception would be thrown. Lastly, I used parameterized queries. This is the main method I used to reduce the risk of SQL injection. To further reduce risk I could have used stored procedures in which I passed parameters to the procedures. Thus if the datatypes of the parameters didn’t match, the procedures wouldn’t have executed. However, I just didn’t have enough time to do this as I had to move on to other projects.

To set up the application you will need to place all files in the ASP folder in the path C:\inetpub\wwwroot\ASP. I have included an SQL file that includes all the DDL and DML to build and populate all of the initial tables, plus create the sequences used for the primary key ID numbers. Double click on the FinalProject.html file to run the application which will launch the project homepage. From here you can click the “View” button to view the current tables, “Insert Data” button to insert new rows, “Edit Data” button to edit tables, “Delete Data” button to delete rows.” The “Sales Report” button displays the total homes sold, total sales amount, and average sale amount per home.

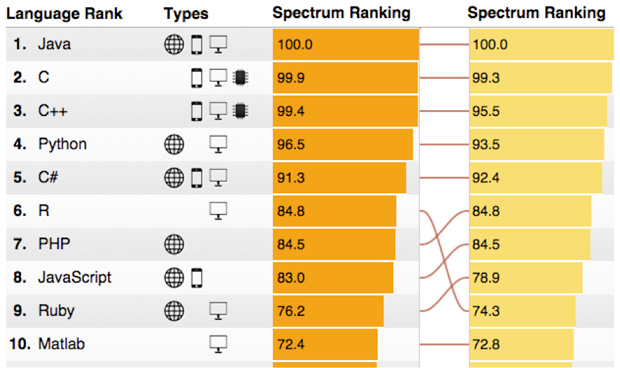
To test my application I used a cyclical method for the insert. I knew that I had a working project from project 2 to start with so I added one table at a time and would comment out lines that I thought gave me a problem. I also used debugging statements to understand the behavior and progression of the program. The only issue I ran into with the insert statement was the oracle error of missing expression. I had to do some research and realized that this was due to the fact that my parameterized queries were using the wrong syntax. I had used the SQL server syntax “@” symbol in my parameters instead of the Oracle “:” colon. Once I corrected this, my insert statement worked. After everything worked for one table I would add the code for the next one and progress from there until I had all four tables. Once I had the insert code working it was a simple matter to change the SQL statement to an UPDATE statement for each table and then I had a fully functioning application for my insert and update. After that it was too, easy to just copy, paste, and update code from project 2 for the delete function.

My approach to this project was a pretty straight forward waterfall method. The lessons I learned from this project are that the syntax for parameterized queries for Oracle and SQL server are different. I had some lost time trying to figure out why my parameterized queries wouldn’t work until it took the right Google search to figure out that SQL server syntax for parameters uses the “@” symbol, while Oracle uses the “:” colon. I spent hours wondering why my insert just wouldn’t work, and that was all it was.

My design strengths are that the application is easy to use, and that it has relatively reduced the risk of SQL injection. The limitations are that after each insert or update the user is required to go back to the homepage before executing another insert or update. One improvement would be allowing for multiple inserts or updates. Another improvement would be controlling which tables the user can enter data into through a radio button event by disabling the text fields of the other tables on button click. I think that the most challenging aspect of developing this application was trying to develop a secure application while working with both an unfamiliar language and framework, especially when mitigating security risks such as SQL injection was not a topic covered in depth during the course of this semester.

I think that VB.NET, as I have seen it commonly referred to, has its strengths and weaknesses. I think that it can still be used to build both relevant and secure web applications that are functional and aseptically pleasing. However, I do think that Java EE has a clear advantage over both PHP and ASP.NET (C#, VB). I think that this can be evidenced by the fact that Java has been the number language for some time now(as shown in Figure1) with C a close second, although PHP has gained some popularity.

Figure The 2015 Top Ten Programming Languages



Note: Ranking system is driven by weighting and combining 12 metrics from 10 data sources such as the IEEE Xplore digital library, GitHub, and CareerBuilder. From Cass, S (2015) “The 2015 Top Ten Programming Languages” IEEE Spectrum

Java’s OO properties make it a clear power house, and Java has its own query language (JPQL), plus its own data base connectivity API (JDBC) . On top of all that it is platform independent so it is completely portable. Plus with Java you have the choice of Servlets, JSP’s or JSFs. There are just so many ways to build web applications. Also, there are several great IDE’s to choose from to build them with such as NetBeans or JDeveloper, were as with ASP.NET you are limited to Visual Studio and I’m not even sure about PHP. I may be partial to Java and Oracle products, but then again, it looks like I’m not the only one.

This project was both challenging and rewarding. In the end I am glad that I chose to go with an unfamiliar language as I think it broadened my horizons and helped to enhance my understanding of the ASP.NET framework and not only the Visual Basic language, but also the C# language as I found many examples of both syntax along the way. Also, I have learned that I can be very proficient at coding without an IDE which is something that I never would have thought I could do. I never would have known this if I would have chosen to stay in my comfort zone with Java and use NetBeans, I am very glad that I didn’t.

# References

Cass, S. (2015). “The 2015 Top Ten Programming Languages.” IEEE Spectrum. Available at <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages/?utm_source=techalert&utm_medium=email&utm_campaign=072315>