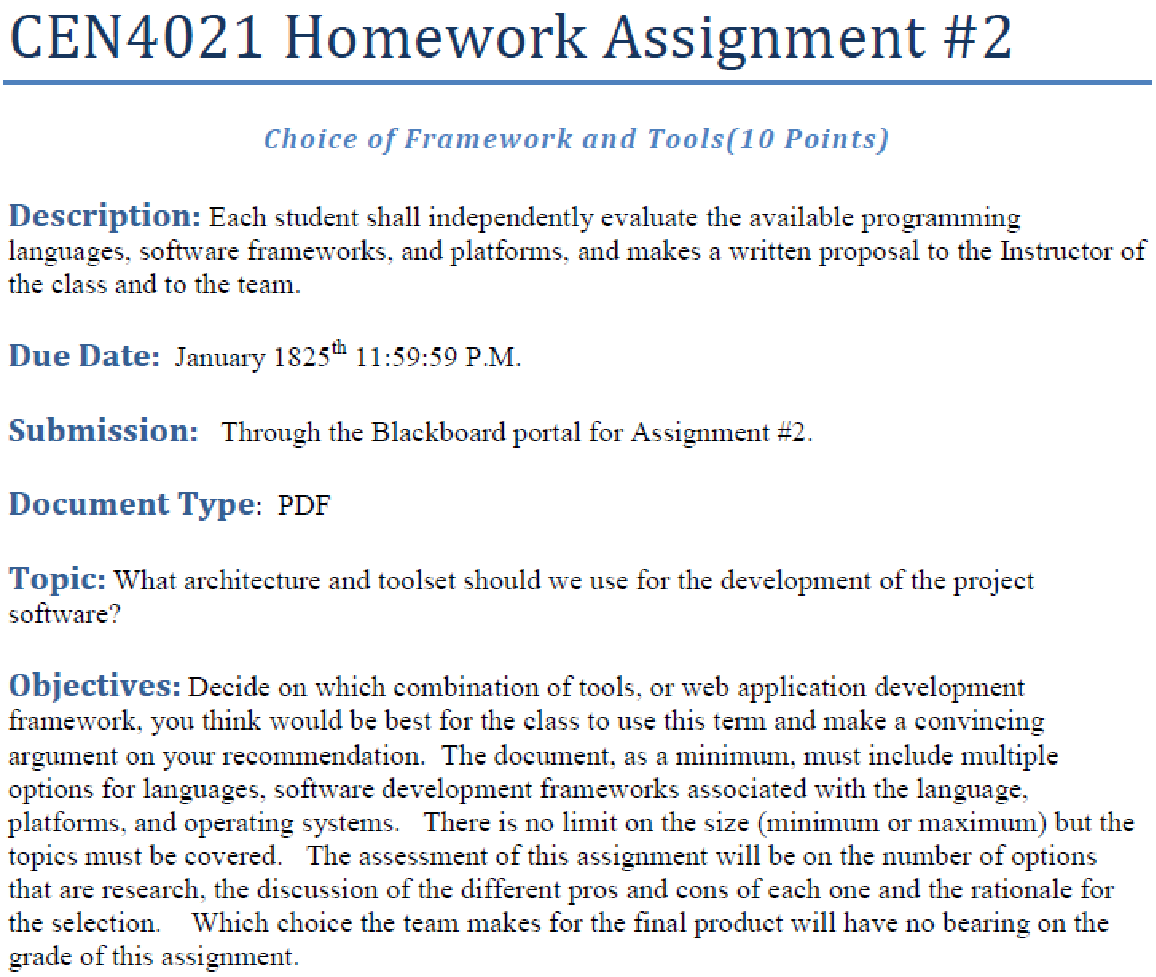
Angela Reaves
CEN-4021 – Spring 2013
Homework Assignment #2
Due Date: January 27, 2013



CEN4021 Homework Assignment #2

Choice of Framework and Tools(10 Points)

Description: Each student shall independently evaluate the available programming languages, software frameworks, and platforms, and makes a written proposal to the Instructor of the class and to the team.

Due Date: January 1825th 11:59:59 P.M.

Submission: Through the Blackboard portal for Assignment #2.

Document Type: PDF

Topic: What architecture and toolset should we use for the development of the project software?

Objectives: Decide on which combination of tools, or web application development framework, you think would be best for the class to use this term and make a convincing argument on your recommendation. The document, as a minimum, must include multiple options for languages, software development frameworks associated with the language, platforms, and operating systems. There is no limit on the size (minimum or maximum) but the topics must be covered. The assessment of this assignment will be on the number of options that are research, the discussion of the different pros and cons of each one and the rationale for the selection. Which choice the team makes for the final product will have no bearing on the grade of this assignment.

There are many tools, languages and frameworks available for the class to use this term in developing our software projects. Our team has decided to build a web application using the MVC architecture. Model-View-Controller (MVC) is a software architecture pattern that separates the business logic (model) from the user interface (view). The controller takes the user input and converts it into commands for the model or view, depending on the type of input. The advantages of the MVC pattern are code reusability and separation of functionality. A disadvantage of the MVC pattern is that it introduces new levels of indirection and therefore increases the complexity of the solution. I think the increased complexity is worth the extra effort as the separation of business logic from the view is easier to maintain. Another software architecture pattern that is similar to MVC is the MVVM (Model-View-ViewModel) pattern. MVVM is based on MVC and MVP (Model-View-Presenter) and attempts to more clearly separate the development of the UI from the business logic and behavior in an application.

Many implementations of this pattern make use of declarative data bindings to allow a separation of work on Views from other layers. This pattern was developed by Microsoft and its main difference between MVC is that the controller is replaced with a view model. The view model sits below the UI layer and exposes the data and command objects that the view needs. Although there are some advantages to this architecture over MVC, it clearly would be overkill for our small application and would add another layer of complexity that is not necessary.

In implementing the MVC pattern in our project, the Model will be implemented as Java classes, the View will be implemented as JSP (Java Server Pages) and the Controller will be implemented as Java Servlets. I will discuss each one in turn with their pros and cons along with alternate implementations.

In developing the business logic (the Model) for our application, we have decided to use the Java language. Java was designed to be easy to use and therefore it is easier to write, compile and debug in comparison to other languages such as C or C++. Java is object-oriented which allows modular programming and reusable code. Java is platform-independent, which means it is easy to move a Java program from one system to another. This is crucial when developing web applications as we should not need to worry about what operating system that a user has, only that the user has browser software such as Internet Explorer or Firefox. Java is one of the first programming languages to consider security as part of its design, so it is much more secure than other languages such as C or C++. Because of Java's robustness, ease of use, cross-platform capabilities and security features, it has become a popular choice for providing worldwide Internet solutions.

Some disadvantages of the Java language is that it can be perceived as slower and more memory-consuming than other natively compiled languages such as C or C++. Java is an interpreted language which means that it needs an interpreter in order to run Java programs. The programs are compiled into Java Virtual Machine (JVM) code called byte code. This has an advantage of portability but a disadvantage of decreased performance in comparison to C or C++. Another disadvantage of Java is that it doesn't support multiple inheritance like C++. It does provide interfaces which can be used to simulate multiple inheritance.

In developing the presentation layer (the View) for our application, we have decided to use Java Server Pages (JSP). JSP is a technology that allows developers to create dynamically generated HTML, XML or other document types. For our application, we will use JSP to create dynamically generated

HTML pages only. It is similar to PHP or other CGI languages but it uses the Java programming language instead. JSPs are translated into servlets at runtime which means that each JSP's servlet is cached and re-used until the original JSP is modified, which increases the application performance. JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver the generated HTML page. Like any other Java program, they must be executed within a Java Virtual Machine (JVM) that integrates with the server's host operating system to provide a platform-neutral environment. With the use of the Java programming language, we have the same advantages and disadvantages as described above.

Another option that could be used for the View layer of our web application is PHP. PHP has many advantages in that it is relatively quick to develop in PHP, a non-programmer can pick it up quickly as it is not a strongly typed language like Java. This is also a disadvantage of PHP in that because it is loosely typed, unexpected behavior can occur due to programming errors that languages like Java would not permit. It also lacks inherent security features like Java.

We will also be using a certain amount of JavaScript to do client-side validation on the JSP/HTML pages. These same validations could occur on the server side through JSP / Servlet code, but this would decrease the application performance if we had to go to the server for each validation request. JavaScript has certain disadvantages as well, such as exploitation by hackers. Because of this, care must be taken when writing JavaScript so that these security flaws are not exploited. JavaScript is also a loosely typed language so it is easy for programmers to create issues that are hard to debug. JavaScript can also behave differently on different browsers. Because of this, the pages that do utilize JavaScript will need to be tested on many different browser types.

In developing the Controller portion for our application, we will use Java Servlets. Java Servlets have many advantages over traditional CGI programs, such as PHP programs. They are more efficient, easier to use, more powerful, more portable and safer than traditional CGI programs. With traditional CGI, a new process is started for each HTTP request. If the CGI program is relatively short, the overhead of starting a process can dominate the execution time. With servlets, the JVM stays running and handles each request using a lightweight Java thread, not a heavyweight operating system process. Servlets also have an extensive infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, tracking sessions and many other high-level utilities. Servlets support several capabilities that are difficult or impossible to accomplish with regular CGI, such as talking directly to the Web server whereas regular CGI programs cannot without using a server-specific API. Multiple servlets can also share data, making it easy to implement database connection pooling and similar resource sharing optimizations. Another advantage of servlets over CGI programs are that they can run virtually unchanged on many different kinds of servers such as Apache, Microsoft IIS, IBM Web Sphere, etc. This is again due to Java's portability.