Annotated Bibliography

Carri Martin

Regis University

**MSSE 692 Software Practicum I**

**Current product research**

Rapla . (n.d.). Rapla - Introduction. Retrieved from <https://http://rapla.org/>

Rapla is a resource and event planning system that features a graphical calendar interface. It is a fully customizable and configurable calendar that includes common features and functions.

I found this during my product search to see what kinds of open source calendar options were currently available. This product seems to be a nice option that will work with or without database support and has a nice graphical interface.

Berger, M. (n.d.). BORG Calendar. Retrieved from <https://mikeberger.github.io/borg_calendar/>

The BORG Calendar is a combination calendar and task tracking system with functionality that is similar to Microsoft Outlook. It was designed to be a small self-contained Java application that can run locally from a flash drive.

This appears to be a good product with nice graphical interfaces for its calendar functions as well as it's task tracking functions that utilizes Java Swing.

DHTMLX. (n.d.). How to Start. Retrieved from <https://docs.dhtmlx.com/scheduler/how_to_start.html>

This article is one of the support documents posted on the DHTMLX website. It discusses how to get started using their Javascript calendar component. This component provides a very nice graphical event calendar that supports a wide range of views and features.

I examined this article to determine if I could utilize a free version for our practicum project. Unfortunately, utilizing this component turned out to be well beyond my current java skill set.

**Database research**

Nadal, J. (2015, January). Connect to Mysql database with java (Netbeans). Retrieved from <https://www.youtube.com/watch?v=e3gnhsGqNmI>

This is a short tutorial on how to connect a MySQL database into a java application using the Netbeans IDE. It provides a one class example of how to call on that database connection and how to insert new values into it.

I found this video very helpful for figuring out how to set up a very basic database connection within Netbeans and how to test it.

Giunipero, T. (n.d.). Creating a Simple Web Application Using a MySQL Database. Retrieved from <https://netbeans.org/kb/docs/web/mysql-webapp.html>

This is article provides a short tutorial on how to create a simple web application that uses a MySQL database for storage and data retrieval. It covers everything from the initial project creation to setting up JavaServer Pages to displaying dynamic logic and information from the database.

I found this tutorial to be a very useful source of information on how to not only start a web application project but also how to set up JSPs to display the information my project needs. It opened with a very interesting discussion of two-tier architecture where a client communicates directly with a server which applied directly to the way my own project is being developed.

Stackoverflow. (2015, October). Glassfish Admin Console throws java.lang.IllegalStateException when creating JDBC Pool. Retrieved from <https://stackoverflow.com/questions/33048435/glassfish-admin-console-throws-java-lang-illegalstateexception-when-creating-jdb>

This forum question was originally posted almost three years ago on a known bug in the Glassfish Admin Console that persists to this day. This bug prevents the creation of a JDBC connection pool which, in turn, prevents the project from reaching the database.

I found this source to be invaluable in my efforts to figure out why my project couldn't reach my database. It provided me with the workaround that effectively saved the day and allowed me to progress with my project development. After looking through the various solutions in this post I finally went with the option to modify an existing sample pool in the Glassfish Admin Console.

Tutorialspoint. (n.d.). JDBC - Statements, PreparedStatement and CallableStatement. Retrieved from <https://www.tutorialspoint.com/jdbc/jdbc-statements.htm>

This tutorial discusses the concept behind JDBC prepared statements and how they are used to define the properties that enable an application to send SQL commands to the database as well as receive data back. Prepared statements are generally used in situations where SQL statements are needed multiple times and when they must accept varying input parameters at runtime.

I found this source to be an excellent reference for how prepared statements are used as well as how to implement them. Because my project works so closely with the database I needed to master this skill very quickly in order to proceed smoothly with the development of various use cases.

Stackoverflow. (2009, August). jQuery UI DatePicker - Change Date Format. Retrieved from <https://stackoverflow.com/questions/1328025/jquery-ui-datepicker-change-date-format>

This particular source is a forum question that addresses an issue I encountered using the jQuery Datepicker widget. Without modification this widget standardizes date entry in the format of MM/DD/YYYY.

I found the solutions provided in this source to be extremely helpful in my quest to figure out how I could change the date format produced by the datepicker. The MySQL database required dates to be formatted YYYY-MM-DD for successful entry insertion.

**Design research**

Oracle. (n.d.). Class Calendar. Retrieved from <https://docs.oracle.com/javase/7/docs/api/java/util/Calendar.html>

This is an Oracle document describing the Java Calendar class which is an abstract class. It is useful for such things as determining the date one week in advance or for determining which day of the week a specific date falls on.

I found this useful for understanding the capabilities of Java's calendar class as it might pertain to my practicum project.

Sparky GlassFish. (2011, January). Java EE 6 Development with NetBeans and GlassFish. Retrieved from <https://www.youtube.com/watch?v=UBNaiVWwAZw>

This tutorial starts out with a very simple Java EE 6 web application example that uses servlets and discusses how to set up Enterprise Java Beans and JavaServer Faces pages.

I found this video to be an excellent refresher for how to start Java EE web applications and use various components. It was also very helpful in determining what basis to use for my practicum project by swaying me towards Enterprise Java and the use of a browser-based user interface presentation.

Stackoverflow. (2011, February). Show JDBC ResultSet in HTML in JSP page using MVC and DAO pattern. Retrieved from <https://stackoverflow.com/questions/5003142/show-jdbc-resultset-in-html-in-jsp-page-using-mvc-and-dao-pattern>

Although this was a forum question the responses it received from the Stackoverflow community discussed how to implement a Model, View, Controller pattern using JavaServer Pages and data provided from a database.

The discussion in this post provided an excellent overview of how the MVC design pattern is being used by professionals in field. I found the example code they provided very helpful for the development of my project.

jQuery. (n.d.). Datepicker. Retrieved from <http://jqueryui.com/datepicker/>

This page provided the source code to utilize JQuery's popup datepicker widget. It also provided API documentation.

I needed this functionality for my project because I wanted a way to standardize the way dates were entered into form fields. This enables me to bypass logic on the back end that I would have needed to handle all of the various ways dates are entered when manually typed.

CodeJava. (2018, April). How to create dynamic drop down list in JSP from database. Retrieved from <http://www.codejava.net/java-ee/jsp/how-to-create-dynamic-drop-down-list-in-jsp-from-database>

This article provided an excellent tutorial on how create a dynamic dropdown list in a JSP using tag library c:ForEach tags on the front end and Servlet/DAO classes on the back end. It provided an explanation at every stage of the process discussing why each step was taken and how it worked.

This tutorial was instrumental in the development of my own dynamic dropdowns not only for its discussion of how to utilize c:ForEach tag but also how to handle the servlet and DAO coding as well.

Stackoverflow. (2011, March). How to get the current date/time in Java. Retrieved from <https://stackoverflow.com/questions/5175728/how-to-get-the-current-date-time-in-java>

This was a Stack overflow forum question that discussed how to get the current date/time for a Java application. In this thread, forum members debated various options ranging from using the Calendar class to Joda time objects.

I found this forum posting not only useful for providing a way that I could obtain the current date for my project but also interesting for the discussion it provided on the options. How to code this simple function appears to matter of preference depending on whether one wishes to take time zones into account.

Tittel, E., & Minnick , C. (n.d.). How to Use Password Fields and Hidden Fields in Your HTML5 Form. Retrieved from <https://www.dummies.com/web-design-development/html5/how-to-use-password-fields-and-hidden-fields-in-your-html5-form/>

This article provided a tutorial on how to use password fields and hidden fields in an HTML5 form. The authors also discussed how a password field conceals a user's password so that it may not be read as the user types it in.

I found this article to be very useful while working with my HTML login page. As a simple form of security, concealing a user's password from casual onlookers is a basic necessity for modern web-based applications.

Stackoverflow. (2016, August). Adding bootstrap template to jsp. Retrieved from <https://stackoverflow.com/questions/38859248/adding-bootstrap-template-to-jsp>

This forum question discussed how to add a bootstrap template to a JSP. This allows the JSP to draw on the template for the overall style displayed on the page.

I found the answers to this forum question very useful. They helped me to figure out the correct way to direct my JSPs to a bootstrap template so that I could utilize its style formatting to beautify otherwise plain pages.

w3schools.com. (n.d.). Bootstrap 3 Tutorial. Retrieved from <https://www.w3schools.com/bootstrap/default.asp>

This tutorial from w3schools.com provides instruction on a popular HTML, CSS, and Javascript framework known as bootstrap. This framework provides formatting and color themes/templates for HTML web pages.

This tutorial provided an excellent resource to help me learn more about bootstrap as a template for beautifying my JSP pages. I found it's examples very easy to follow and utilize within my project.

Bibeault, B. (n.d.). Scriptless JSP Pages: The Front Man. Retrieved from <https://javaranch.com/journal/200603/frontman.html>

This article discusses the highly debated subject of scriptless JSP pages. It begins with a historical review outlining the script filled beginnings of JSP development before moving on to the discussion revolving around why they should be kept scriptless. The author also offers a discussion on various design patterns before concluding with the Front Controller pattern.

I found this article to be an interesting argument for scriptless JSP pages. It provided a very compelling stance for why scripts must not be allowed within JSP page code.

**MSSE 696 Software Practicum II**

**Cloud research**

OpenShift. (2015, May). OpenShift Commons Briefing #3: Java EE Workflows on OpenShift with Arun Gupta. Retrieved from <https://www.youtube.com/watch?v=cgCVXnQz_iU>

This video provided a basic overview of Red Hat's public cloud offering known as OpenShift. In this discussion the speaker, Arun Gupta, talked about the various options available with this Platform as a Service for a Java EE developer.

This video is a great resource for Java EE developers new to the OpenShift platform. I found it to be very informative regarding what options are available to those of us seeking to learn more about cloud-based development.

NetBeansVideos. (2014, December). Deploying Java Apps to the Cloud with OpenShift and NetBeans IDE. Retrieved from <https://www.youtube.com/watch?v=VtzThoMscAg>

This is a short tutorial video that shows how quickly Java apps may be created and deployed on a Tomcat server hosted on the OpenShift platform. It also provides a step by step tutorial on how to use OpenShift from the Netbeans IDE.

I found this video to be quite interesting as a tutorial resource for just how quickly one may deploy an app into a cloud environment.

Vaughan-Nichols, S. J. (2017, July). Microsoft Azure. Retrieved from <https://www.pcmag.com/article2/0,2817,2496295,00.asp>

In this article, published by PC Magazine, the author provides a very well-balanced review of Microsoft Azure in terms of its place in the Infrastructure-as-a-Service market as well as its overall cloud performance. In his conclusion, Vaughan-Nichols recommended Microsoft Azure not only for Windows-centric companies but also to those more willing to embrace new open-source options as Microsoft continues to integrate opensource and Linux functionality.

This article was interesting for its balanced approach and hands on examples that provided a very well-illustrated stance on the author's findings.

Questpond. (2015, November). Azure Tutorial for Beginners. Retrieved from <https://www.youtube.com/watch?v=wdUK7bCMXqs>

This video tutorial begins with a discussion of the basic ideas and differences between IaaS, SaaS, and PaaS cloud technologies. From there the presenter proceeds to discuss Microsoft Azure, outlining how to sign up for a free trial and proceeding through various ways to use this cloud-based service.

I found this video very informative and helpful in my efforts to better understand Azure as it is the platform upon which our team will be building our project in this class.

**Spring Boot research**

Katamreddy, S.P. R. (2016, May). Why Spring Boot? . Retrieved from <https://dzone.com/articles/why-springboot>

In this article the author provides an outline of the Spring framework, discussing how its dependency injection approach encourages the development of testable code in addition to providing simplified yet powerful databases transaction management capabilities. During the course of the article a simple web application is provided and later contrasted by a version using Spring Boot to illustrate its ability to simplify the development process.

The author of this article presented a very nicely explained example of the power of using Spring Boot for web application development. I was impressed by how easily it simplified the application dependencies as well as its ability to automate a number of things behind the scenes.

Spring. (n.d.). Spring Boot Reference Guide. Retrieved from <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#boot-documentation>

As the title indicates, this document is the official documentation for Spring Boot. Based on the Spring Framework, Spring Boot may be used to create Java applications in a more simplified manner compared to other alternatives.

As official documentation for Spring Boot, upon which my project is based, this source is invaluable for my development efforts.

The Coding Interview. (2016, August). Creating a Restful web service with Spring Boot. Retrieved from <https://www.youtube.com/watch?v=EDgCHCFKo8c>

This tutorial begins with a short basic introduction on REST before launching into the steps necessary for creating Spring Boot RESTful services. The presenter gave examples on how to create both GET and POST methods as well as how the test them in a browser and using an application known as Postman.

I found this video to be very informative regarding the basics of REST as well as the basic configuration for a Spring Boot application. Postman was also a new tool that I had never encountered before and have found to be quite helpful.

WebSystique. (2017, September). Spring Boot Rest API Example. Retrieved from <http://websystique.com/spring-boot/spring-boot-rest-api-example/>

This source is a tutorial that provides step by step instruction on how to build a Spring Boot REST API that utilizes a Model, View, Controller pattern. As the tutorial walks through the steps it also provides an explanation of the Spring Boot annotations being used as well as methods to test the finished product.

This tutorial was an excellent example that helped me to conceptualize how I might create my own RESTful services using the MVC pattern.

**Design research**

Kelly, J. (2017, April). RESTful Web Service In Netbeans using GlassFish. Retrieved from <https://www.youtube.com/watch?v=IvqakwjjHhg>

In this video, the presenter provided a step by step tutorial on how to create very simple RESTful services using Netbeans and Glassfish. He used Netbeans' wizard to create RESTful web services from a database which automatically generated several classes.

Although this method was not one I learned in my Java class, this video was still an interesting refresher for how RESTful services work. A built-in testing method that I had not known of previously was also discussed.

Apache Maven Project. (2018, July). What is Maven?. Retrieved from <https://maven.apache.org/what-is-maven.html>

In this article the author(s) provide a short outline describing what Maven is. As a tool to facilitate the building and management of Java-based projects, Maven's objectives are quite simple. It simplifies the build process, provides a uniform build system that produces quality project information while also establishing guidelines for development best practices.

This article provided a very succinct overview describing what maven is and how it was intended to be used. I found this information very helpful in the beginning stages of project development.