Drew Faber

CIS 263 – Research Paper

Prof. Mick

11/30/14

Vaadin Framework

Recently I have become very interested in finding an internship or a job as a Java developer. Throughout this search I have noticed some of the most common requirements to become a developer. Most job ads recommend or require previous experience, open source contributions, and knowledge of frameworks. Since I have no previous experience or open source contributions I concluded that I should take this opportunity to learn more about frameworks. I have no previous experience and little to no knowledge on the subject of Java frameworks. The version control system Github recently released a list of most commonly used Java frameworks. While going through these top used frameworks I found Vaadin, Vaadin is a web application framework for Java. This framework intrigued me because it can produce apps for any operating system and every device mobile, tables, and desktops. It produces beautiful interfaces. It is also an open source project that could lead to my goals of contributing to an open source project in the future. Since there is plethora of Java frameworks I will use Vaadin as my main focal point in discovering the purpose, history, usage, and importance of frameworks. I will analyze the functions, features, and current projects that use Vaadin to understand more clearly what an open source Java framework can help produce and maintain in terms of software development.

First let’s find a general explanation of a framework. A framework is composed of libraries and APIs to create a complete system to develop software and applications in a systematic way. Frameworks take a complex set of libraries and APIs and make them easier to apply while enforcing consistent coding principles that result in fewer bugs. Furthermore, frameworks also incorporate design patterns and methodologies to create robust applications. Libraries are the toolbox of an application or piece of software, they contain methods and objects for the application to utilize and handle the data. Libraries provide consolidated code that can be modified, support inheritance and polymorphism. During our class we have used many of these Java libraries such as .lang, .io, and .swing. APIs are interfaces or a means of communication from one system to another system. These interfaces provide standardized requests to data within the system so other applications can use it. There are many businesses that operate strictly from using the APIs of other programs such as Mint and websites for games. If you have used the Java website while coding in the class then you are most likely familiar with some API. Design patterns are commonly used functions or solutions in object oriented development, they solve a problem with-in a system that can be reused as needed. If a problem occurs several times you would implement a design pattern to continually handle it in an efficient manner. Methodologies for example SDLC (Software Development Life Cycle) and the like, help maintain large projects to help obtain goals and continue progression. Methodologies do this through tying together components, architecture, and objects to create a well-defined system. They help keep systems tied together by having a set of rules and ideologies to create consist code format and consistent style to developing the code as a business. Keeping everything tied together is essential in larger projects with multiple employees working on the same project. Now that we have a general idea of what a framework is and what it implores let’s move on to the need for frameworks.

Java frameworks were developed because programmers are lazy. More so, they help deal with the complexity of creating a large web application or application which is extremely difficult and time consuming. Developing applications for public or internal use can be a very big project. These applications need to have top notch security measures, appealing graphics, ease of use, fast response times, serve up to millions of users, and testing features. Also, to develop solid and reliable code you will need multiple people to revise and critique your code. If you have the time and resources you can build all of these functions and layers yourself to develop your application. If you don’t have these resources to use on developing all of these layers and code you can use a framework to develop your application. Frameworks are developed to help reduce extended investment in a piece of software. They help you develop large projects faster and cheaper than starting from scratch. Frameworks do this by having a predefined set of code and resources that you can alter to design your application. You can alter the code of frameworks by extending the classes or over-riding the methods. They reduce overhead by not having to start from scratch they provide the system or skeleton and you add to them. Using frameworks help you maintain and upgrade your code this can reduce the need for more employees by empowering the efficiency of the programmers you already have. Frameworks are designed by large groups of programmers that are all skilled at different areas and levels in the realm of programming. This helps ensure that the code is optimized and technologically firm. Programmers that contribute to frameworks often observe or test other parts of the framework to ensure it is programmed properly as a team. This creates a very secure and optimized code. Overall frameworks provide a system to help make every programmer faster and more efficient. Which has a trickle-down effect that helps businesses stay profitable and lower development costs.

Vaadin is used in a variety of different areas with its biggest use being in intranet related areas. While there isn’t much information on the specific uses there are showcases of companies that use Vaadin. Vaadin usage varies from aerospace companies like SAIC that use the framework to let researchers view experiments and data in real-time. All the way to a merchandise review system for PUMA clothing. Over one billion users have used Vaadin since its beginning in 2001 and is growing rapidly.

Now that we understand what a framework is with all of the perks and reasons for frameworks we can look more specifically at a framework to see how it works and how to use it. Java has a handful of very popular web frameworks that use the JVM like Spring MVC, Struts, and Grails. I have chosen to look further into Vaadin a web application framework. Vaadin was started in 2001 and is ran by a company in the EU and United States. Vaadin also provides training courses and a certificate. They strive for modern style web applications with focal point on developer productivity and scalability. While this is an open source project I have noticed that some contributors require a special license to use their contributions after a trial period. There are over 400 components that they have in the Vaadin directory and they have everything documented very well. They have multiple tools and websites dedicated to showing examples of components and the code to go with them. Information is plenty full, I viewed their forums and they seem very fast to respond and usually have large amount of replies. Vaadin is a server-side web application framework with client side features, it uses pure Java libraries and browser-plugins. It is based on GWT (Google Web Toolkit) and HTML5 to build robust applications. Vaadin allows you to build web applications code faster; you can use any IDE you prefer and it has built in testing capabilities. While you can develop on any IDE it is highly recommend you use Eclipse with the Vaadin plugin. You can install this plugin from the Eclipse marketplace. You also need to setup Tomcat apache server to test and run the code or find a web hosting service that supports Tomcat. Vaadin supports server side and client side development.

I’m going to start with server-side which is making a servlet. Here I will delve into how user interfaces are created with Vaadin. Vaadin uses AJAX to provide real-time updates to the servlet without having to reload the page. AJAX uses HTTP requests that can send serialized messages to generate HTML and content. This makes the user experience seamless and gives the application the feeling of being a desktop application. When creating a servlet one important, if not most important features is the user interface. Vaadin servlets are required to extend the abstract class UI. While not enforced by the abstract class you need to define the theme that you want to use, this theme tells the servlet what to look like using CSS or SASS. Sass can either be pre-compiled or compiled per request from the browser. You can use pre-defined themes or develop your own for a unique style and look. The abstract class requires the implementation of the function init(). The init() function is much like the main function. Inside the init() you define your components and add them to the servlet layout manager. These components are ultimately the HTML parts of the code that are given style by the CSS. The user interface layout is divided mostly into two separate parts the input/output and the component layouts exactly like pure Java. You start with the base layout manager that gives you the main layout for all areas on the web application. Inside the various layout spaces you can add either input/output components or further customize your layout with more layout managers. There are commonly used layout managers like grid, horizontal, and vertical; there are also customizable layouts such as CSS, custom, and form. The components placed in the layout used in Vaadin are obviously exactly the same as swing. Vaadin adds to the components of swing by making them easier syntactically. One component I can’t find in swing is graphs, Vaadin has built in graph components. They have standard graph such as pie, bar and plot. The real difference I notice with the components between swing and Vaadin is the ease of use. Vaadin syntax and implementation make it very easy to quickly develop a complex component. It makes it very easy to just start developing your servlet Vaadin has a website sampler that lets you view every component and customizations you can make to them. The client-side of Vaadin is modeled after GWT (Google Web Toolkit). This makes it easy to develop user interfaces for the browser with pure Java that compiles into JavaScript components. These components are then rendered by using the GWT and Vaadin-specific widgets. Similarly how server-side builds on swing client-side builds on top of GWT. One of the best uses but not limited to of client-side applications is the stand alone feature for offline usage. Vaadin recommends looking up the Google Web Toolkit documentation to get a full understanding of client-side applications.

Frameworks provide many benefits to a developer. They support strong code on a consistent platform with enriched features. Vaadin has been an eye opening experience for me. It has made me realize the significance of frameworks and what they have to provide. The documentation and examples that Vaadin provides is shocking to me. They empower me into believing that I can start writing a web application that is something important and effective. Vaadins community is also very strong which will enable me to network with programmers to develop my personal programming portfolio. In the simplest terms Vaadin builds on top of what swing has to offer and turns it into a complete package. Framework knowledge is something that every developer needs if they want to be the best and most efficient programmer they can be. Frameworks like Vaadin are the systems that make businesses work and produce large projects. In the near future I plan on making a web application with Vaadin to put on my resume when I look for a new job in the spring. While there are many topics in the Vaadin framework I don’t fully understand the discovery of frameworks has left me eager to tackle then and create something I can show to employers.

References

https://vaadin.com/download/book-of-vaadin/vaadin-7/pdf/book-of-vaadin.pdf

http://www.javacamp.org/designPattern/

<http://www.fredosaurus.com/notes-java/oodesign/frameworks.html>

<http://www.codeproject.com/Articles/5381/What-Is-A-Framework>

<http://msdn.microsoft.com/en-us/library/kews042w(v=vs.71).aspx>

<http://zeroturnaround.com/rebellabs/the-2014-decision-makers-guide-to-java-web-frameworks/>

https://vaadin.com/who-is-using-vaadin