



# **Spaceport Command and Control System Support Software Development**

Leonard Brunotte  
KENNEDY SPACE CENTER  
Major: Software Engineering  
NIFS 2016 Fall Session  
07/11/2016

## Development of Web Applications for Use within SCCS

Leonard J. Brunotte

*Embry-Riddle Aeronautical University – 600 S. Clyde Morris Blvd, Daytona Beach, Florida 32114*

The Spaceport Command and Control System (SCCS) is a project developed and used by NASA at Kennedy Space Center in order to control and monitor the Space Launch System (SLS) at the time of its launch. One integral subteam under SCCS is the one assigned to the development of a data set building application to be used both on the launch pad and in the Launch Control Center (LCC) at the time of launch. This web application was developed in Ruby on Rails, a web framework using the Ruby object-oriented programming language, by a 15 - employee team (approx.). Because this application is such a huge undertaking with many facets and iterations, there were a few areas in which work could be more easily organized and expedited. As an intern working with this team, I was charged with the task of writing web applications that fulfilled this need, creating a virtual and highly customizable whiteboard in order to allow engineers to keep track of build iterations and their status. Additionally, I developed a knowledge capture web application wherein any engineer or contractor within SCCS could ask a question, answer an existing question, or leave a comment on any question or answer, similar to Stack Overflow.

### Nomenclature

|       |  |
|-------|--|
| NASA  | = National Aeronautics and Space Administration              |
| NE-ES | = NASA Engineering Directorate – Software Engineering Branch |
| KSC   | = Kennedy Space Center                                       |
| SCCS  | = Spaceport Command and Control System                       |
| SLS   | = Space Launch System  |
| LCC   | = Launch Control Center                                      |
| STS   | = Space Transportation System – the Space Shuttle program    |
| DA    | = Development Activity                                       |
| IDE   | = Integrated Development Environment                         |

## I. Introduction

Due to the incredible amount of systems involved in the maintenance and launch of a rocket, there is also an equally large amount of data that must be transferred between the LCC and the pad. Because of the technological advancements made since the end of the STS shuttle program, the existing systems are completely incompatible with those used to monitor and control the SLS rocket. Therefore, a small subteam of SCCS engineers from several different branches and contractors was formed in order to create an easily-usable web application that builds the data sets used by equipment both on the launch pad and the LCC.

## II. Detail

### A. Training

Before starting the internship, I was required to complete a slew of training courses on various government codes of conduct and other policies and legal requirements, such as ethics, sensitive document handling, and security. In addition, I was required to complete training courses in a few pieces of software commonly used by NASA employees. Most notably, I completed training on work tracking software, allowing leads and developers to issue work orders and development activities (DA's), and version control software used by SCCS developers to track and manage changes to different versions of the software. However, because I was not working directly on the data set building application and had my own projects that I completed on my own timeline, I was not required to use either of these programs. Additionally, the two other web development interns and I were required to complete an online ebook training course in using Ruby on Rails, in which the author walked the reader through building a "Twitter-esque" web application. After the course was completed, our mentor gave us a Rails tutorial, in which he walked us through building another application, but with a more hands-on approach. This tutorial also taught us how to use an Integrated Development Environment (IDE) program, as well as a few of the more common and useful managed software packages.

### B. Involvement in the Whiteboard Application

One of these previously mentioned separate projects was a whiteboard application to be used by members of the data set building application team, allowing them to better organize their work in building iterations of the web application. This was our first project, allowing myself and two other interns to familiarize ourselves with Ruby on Rails and how to correctly build a Rails application. I would later split off from the group to develop a separate application, but my work on the whiteboard application was an important step in understanding this new framework.

This application is designed to receive information about the status of builds across the SCCS project's domains, and display this data in an easy-to-use and thorough fashion. There can be many of these builds running at one time, and there often are, forcing the team members in charge of this task to keep a temporary list of all of these builds, as well as upcoming iterations. Prior to the implementation of the whiteboard application, this was done on a physical dry erase board, which would become quite cumbersome with the board's limited space and the amount of data that was incorporated into the identity of each build. This new application allows the user to schedule builds and view all running, upcoming, and recent builds on as many domains as the user wishes. This application was deployed in the last week of October, and is currently in use by the data set application team, but is still undergoing fine-tuning and development.

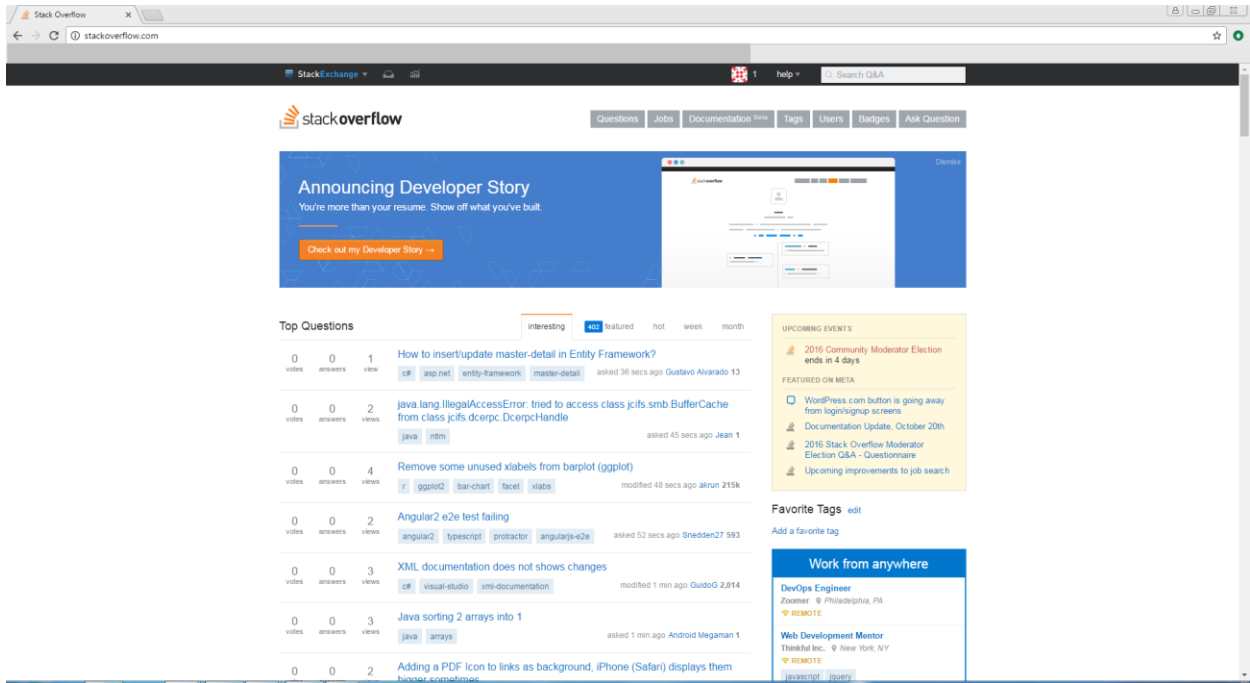
### C. The Knowledge Capture Application

While engineers in NASA's NE-ES branch are generally highly knowledgeable, it is impossible to know the answer to every question one might have. Many of these answers can be found on a knowledge capture site known as Stack Overflow (see [Figure 1](#)), where programmers around the world can post questions and answer other users' questions on a giant wiki forum. However, most SCCS material is too sensitive for NASA engineers to post online, creating the need for a similar web application within NASA's internal network. I learned about this need after approximately a week of developing the whiteboard application, and volunteered to begin developing the new knowledge capture application on my own.

This application features the ability to allow authenticated users to post questions to the main webpage, answer other users' questions, and comment on both questions and answers, as well as edit and delete their own questions, answers, and comments. Additionally, each user has their own page, displaying their recently asked questions and whether or not the questions have been answered, as well as if the answer solved the problem, if applicable. The knowledge capture application also features a tag system, in which the user can enter keywords while creating a question. Upon clicking on one of these keywords, a list will be displayed of all questions tagged with the same keyword. Furthermore, the knowledge capture application also features a list of all tags that have been created, a feature not available in Stack Overflow. Users may also search specific questions or tags through separate search bars

# NASA KSC – Internship Final Report

on their respective pages. However, unlike its spiritual predecessor, the knowledge capture application does not feature a reputation system as requested by my mentor, since there is no need for such a system amongst the SCCS engineers.



**Figure 1. Stack Overflow Home Page**

*The home page of Stack Overflow, which the knowledge capture application is designed after.*

At the time of writing this report, development on the knowledge capture application is not yet finished. CSS styling has been implemented, but is not fully complete as of yet. Additionally, the ability to attach pictures and documents to questions and answers will be added in the future, as well as a user feedback page to allow for the developers to further improve upon the site. The Help page will most likely be replaced later on as well, as a tour will be implemented for any user's first login. Despite the application being technically incomplete, all of its main features have been implemented, as well as thorough automated model and integration testing, and is ready for deployment.

## D. Deployment Methods

Version control software allows for easy access to the latest iteration of each program or application, and is used liberally throughout NASA; SCCS uses a third-party software that fits the needs of the program. However, due to the relatively unofficial nature of my work at NASA, I was not required to use this software, and instead used a much more popular and user-friendly program, using online repositories to store code and commit data. The repositories for both the knowledge capture and whiteboard applications are owned by my mentor, who, once notified that each application was ready to be used, would pull the latest version of the code from the cloud and deploy it to NASA's internal network. This allowed the whiteboard application to be connected to the already-existing data set application team's network, and allows the knowledge capture application to be used by all of SCCS as a standalone application.

## III. Conclusion

## NASA KSC – Internship Final Report

To conclude the final report, this internship has been an incredible experience and an exceedingly important part of my life. I was assigned the task of developing two different web applications to aid NASA's data set building application and SCCS teams using a web framework that I have never seen before. My experience working at Kennedy Space Center allowed me to get a taste of life working for NASA, and helped me form many vital contacts that will certainly be useful as I continue my education and future engineering career. Furthermore, as a student who only recently (within one year) decided to pursue a Software Engineering degree, this experience solidified my decision to stay with this major.

### **Acknowledgements**

I would like to thank Andrew Davis for being an exceedingly helpful and knowledgeable mentor, who was always willing to help with any issue the three of us may have had. On a similar note, I would like to thank Josh Johnson for doing the same, even though he was not technically a mentor. Furthermore, I would like to thank Caylyne Shelton and Jamie Szafran for their excellent supervision and general helpfulness, even to those that they were not assigned to mentor. I would also like to thank Oscar Brooks for his insight and helpfulness, as well as his role in initiating all of us into the NASA culture. Finally, I thank my fellow interns for being excellent coworkers; Jacob Moore, William Gailey, Tim Chinenov, Elijah Spiro, Andrew Hwang, Michael Backus, Mark Rodriguez, and Brandon Echols.

### **References**

"Stack Overflow." *Stack Overflow*. N.p., n.d. Web. 14 Dec. 2016. <<http://stackoverflow.com/>>.