



**ROLLINS**

**Designing and Developing Web-based Administrative Tools for  
Program Management**

Michael Gutensohn  
Kennedy Space Center  
Major: Computer Science  
NIFS Fall Session  
Date: 07 11 2017

# Designing and Developing Web-based Administrative Tools for Program Management

Michael Gutensohn<sup>1</sup>

*Rollins College, Winter Park, FL, 34772, USA*

## Nomenclature

GSDO	=	Ground Systems Development and Operations
KSC	=	Kennedy Space Center
WBS	=	Work Breakdown Structure
POC	=	Point of Contact
MVC	=	Model View Controller
API	=	Application Program Interface
UX	=	User Experience
UI	=	User Interface
App	=	Application

## I. Introduction

The task assigned for this internship was to develop a new tool for tracking projects, their subsystems, the leads, backups, and other employees assigned to them, as well as all the relevant information related to the employee (WBS (time charge) codes, time distribution, certifications, and assignments). Currently, this data is tracked manually using a number of different spreadsheets and other tools simultaneously by a number of different people; some of these documents are then merged into one large document. This often leads to inconsistencies and loss in data due to human error. By simplifying the process of tracking this data and aggregating it into a single tool, it is possible to significantly decrease the potential for human error and time spent collecting and checking this information.

## II. Objective

The main objective of this internship is to develop a web-based tool using Ruby on Rails to serve as a method of easily tracking projects, subsystems, and points of contact, along with employees, their assignments, time distribution, certifications, and contact information. Additionally, this tool must be capable of generating a number of different reports based on the data collected. It was important that this tool deliver all of this information using a readable and intuitive interface.

---

<sup>1</sup>NIFS Intern, NE-XS, Kennedy Space Center, Rollins College

### III. Approach

#### A. Training

At the beginning of this internship, I had no experience with Ruby on Rails, the framework to be used for the project, thus the first few weeks of the internship were spent becoming familiar with it. This was done by building basic applications emphasizing the key concepts of the framework. Ruby on Rails utilizes the MVC design pattern, in which there is a model handling the data being stored, a view displaying the data, and a controller communicating between the model and view. My mentor and I started by creating a note-taking app, emphasizing model associations and structures as well as what is considered conventional for Ruby on Rails. I then went on to create a to-do-list app using more complex model structures, also implementing user authentication.

#### B. Agile

Given the requirements and an understanding of the current methods of tracking this information, I chose to take an approach loosely based on the Agile Development process. Agile consists of three major phases: Design, Development, and Review. These phases are repeated until the project has met the requirements and has been deemed deliverable.

##### *a. Design*

After familiarizing myself with Ruby on Rails, I began coming up with an initial design of the app. At its core, the app to be built for this project is a method of tracking work assignments and POCs for a given subsystem and delivering this information in a quickly understandable way. As previously stated, this is currently done using a combination of spreadsheets and other table-based tools. While tables are an entirely valid method of displaying information, they are not a universal solution. Multi-line text can become difficult to read, while highly similar pieces of information such as phone numbers, emails, and addresses become redundant. Data such as this cannot be adequately represented in a table, but are too minor to warrant their own page.

*Material Design*<sup>2</sup>, a popular design language developed by Google, presents cards as an alternative method of displaying data as a middle ground between tables and pages. A good representation of this concept is the business card. A business card is used to provide an at-a-glance summary of what a person does, what organization they do it for, and how they can be reached. Expanding on this concept and making it relevant to this project, cards can be used to represent an employee's most summary information such as contact information, branch, and time distribution, as well as an employee WBS association.

Much like tables, cards are not a universal solution. And as the project progressed it became apparent that the best solution was to use both delivery methods when necessary, and occasionally a combination of the two.

---

<sup>2</sup>“Introduction - Material Design,” Material Design Guidelines Available: <https://material.io/guidelines/>.

### *b. Development*

At the beginning of the project, I worked with my mentor to design the initial model structure of the project. Using the spreadsheets provided, we broke down the information in a web of models and their associations, creating a model structure to build upon. After building this structure and a very basic version of the app, it was necessary to ingest the data in the spreadsheets provided. To do this, I created a tool that traverses each row of an Excel document and parses the relevant data, some of which is stored or used to generate an assignment. This tool was used to ingest employee data, certification data, branch functions, and WBS data.

In order to cut down on the need to navigate through multiple pages, the app utilizes AJAX in order to make request from the server without having to load an entirely new page. This allows the user to access the most pertinent aspects of the app from the home page such as POC reports or employee branch functions.

The final portion of the project focused on discerning and defining privileges for types of users. This was relatively straightforward. For the average user, the app is reduced to a contact management index, restricting access to POC data, but not allowing them to access details such as branch functions, WBS, data and detailed employee information. In contrast, supervisors will have the access and the ability to manipulate most aspects of the site relative to their organization within KSC's Engineering directorate.

### *c. Review*

Throughout the project I met with my mentor as well as the supervisor requesting this app be built. During these meetings, both of them provided feedback on what I had created; additionally, we discussed the next steps of the project. These feedback meetings were crucial throughout the development process as they were a way of keeping the app within the scope of its requirements, and also kept an open line of communication between myself and the customer.

## **IV. Results**

The final product of this internship is a modern web app currently titled SpaceDex that tracks relevant administrative data for NASA Engineering. When implemented, the tool should save management time on updating and verifying data. Future development could include integrating an API to ingest updates in certification and employee data, as well as possibly integrating the tool with other internal web apps, creating one central location for weekly notes, assignments, and administration.

## **V. Acknowledgement**

I would like to acknowledge my mentor Andrew Davis as well as the other developers working in in the software branch of NASA Engineering. All of them acted as excellent resources both for this project as well as my own curiosities. I would also like to acknowledge USRA for sponsoring this internship, allowing students such as myself to gain valuable work experience in our chosen fields.