Endicia Interactive Style Guides for Future Web Development

A Major Qualifying Project Report

Submitted to the Faculty of

Worcester Polytechnic Institute

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science

By

Lucas McLaughlin

Zeng Liu

Xi Wen

Sponsored by

Endicia

Approved by

Professor David Finkel, Advisor

May 5, 2015

Abstract

We developed a set of style guidelines by creating page templates, components, and widgets for Endicia developers as they create a new CRM called Sputnik. By interviewing users and examining usage logs of Supervisory, Endicia's current, outdated CRM, we also designed the layout for the Sputnik home page. All of these deliverables were created using ASP.NET, AngularJS, jQuery, HTML, and CSS. Over time, our project will greatly speed up the development process and ensure a consistent look and feel for Sputnik.

Executive Summary

The Computer Science curriculum at Worcester Polytechnic Institute (WPI) provides students with the theoretical knowledge and skills to succeed. Through their MQP program, Endicia presented our team with the unique opportunity to synthesize our theoretical education by practically applying our skills to solve a real-world business problem.

Based on the proposed areas of need, our primary objective for this project was to develop website components as well as style guidelines that will assist in and streamline the process of migrating Endicia's old administrative site, Supervisory, to their new administration website, Sputnik. We followed an agile development methodology consisting of gathering requirements, analyzing the current system, and implementing functionality.

Beginning with requirements gathering, we conducted meetings, interviews, and surveys with major stakeholders and potential users of Sputnik. Utilizing these requirements, we determined the key tasks to be implemented. We further analyzed the data in combination with usage logs from Supervisory and created default links for different user groups to provide a more convenient navigation panel. We then segued into the implementation phase. Comprising two major parts, back and front-end, we separated into two small teams working concurrently on the project. The process was highly iterative, as we frequently refined the user interface, database architecture, and other components in response to helpful feedback from Endicia employees.

The deliverables include the Sputnik Developer Hub, in which developers can find all the components they will need to port pages from Supervisory to Sputnik. We also designed a new home page for Sputnik as well as the universal top navigation bar and left favorites panel. Both of these additions will greatly increase customizability and ease of navigation through Sputnik. As a whole, the home page design and Sputnik Developer Hub will cut down development time and unify the look and feel of Sputnik.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

In any form of website or software design, it's extremely important to consider users and how they will be utilizing an application. Without accessibility and ease of use, users get confused, usability plummets, and ultimately a site or program can become useless. In recent years, UI and UX design has become a big concern for companies because of this. Endicia, an online shipping label, stamp, and postage vendor is no exception. Towards the beginning of their company, they built a Customer Relationship Management (CRM) system called Supervisory. Through the years, new functionality was added to this CRM with no concern for design or structure of the site. It quickly became cluttered, unorganized, and complicated for users.

In 2014, Endicia decided to counteract this issue by beginning construction on a new, updated version of the CRM called Sputnik. It was to both better utilize newer backend technologies as well as have an organized, structured UI. In a year's time, however, it too quickly grew out of control. While they were all using Bootstrap, there was still no standard way of using this tool. Developers used it any way they saw fit, leading to major design issues.

To fix this issue before construction progressed any further, Endicia started our project. We were charged with creating a set of components and templates for future Sputnik developers to use. For components, we started with the Bootstrap theme template and then removed unnecessary elements and modified the color scheme to match the Endicia's brand color. For templates, we clicked through every page in Supervisory and classified repetitive patterns in design. We then developed template for each of the patterns we saw. Furthermore, we were tasked with creating a home page and site structure for Sputnik.

In addition to the components and page templates, we also designed a page shell and a home page. In order to get a good understanding for the usage of Supervisory, we interviewed users from different departments, analyzed usage logs, and presented our work to users, managers, and developers for continuous feedback and revisions. All together, the templates, components and home page design would help unify the UI of newly designed pages as well as cut down their development time.

2 Background

2.1 The Bootstrap Framework

2.1.1 Introduction

Bootstrap is a front-end framework for website development that greatly reduces the effort necessary to create a beautiful looking webpage (Bootstrap, n.d.). It is well documented, completely free, and makes development much simpler and efficient. Following its release in 2011, the open source project exploded in popularity and quickly became recognized for its accessible and straightforward nature. In the following sections, we take a look at just what Bootstrap is, some pros and cons, and how it applies to our project.

2.1.2 What is the Bootstrap Framework?

To better understand why Bootstrap is useful for a developer, let's look at a quick example scenario. Imagine you are a web developer trying to design a website that has pretty buttons, animated progress bars, stylish fonts, etc., but you have hardly any experience designing a front-end. Now imagine that all the necessary code was already written for you and explained in detail, and all that was left to do was to write a bit of HTML to make it functional. While it may sound too good to be true, this is exactly what Bootstrap provides (Rahman, 2014).

A front-end framework is defined as a "standardized set of concepts, practices, and criteria for dealing with a common type of problem" (awwwards-team, 2013). These frameworks are often a collection of pre-written HTML, CSS, and JS code. Bootstrap is no exception, and contains some very powerful CSS documents, HTML design templates, and optional JavaScript add-ons. With these tools and the thorough documentation, one can take a bland, un-styled page to beautiful, responsive interface in a matter of hours regardless of experience level. To add to the magic, installation is completely painless, and is as easy as referencing the Bootstrap source file in a style tag.

The framework was originally created in mid-2010 by Mark Otto and Jacob Thornton and released on Friday, August 19, 2011 as an open source project. While it was originally developed for and at Twitter, the project grew so large and is no longer associated with the company (Bootstrap, n.d.).

Because of its ease of use, the tool is commonly used to prototype new sites. However, it can be fully customized to fit the developers' needs well enough to use on a fully functional site. As with anything, there are pros and cons to using the Bootstrap framework, and it's important to assess whether or not it is the right thing to use for a specific project.

2.1.3 Why Use Bootstrap?

2.1.3.1 Save Time

An obvious advantage to Bootstrap is the time saved by using it. The CSS source file is a massive compilation of pre-written, well-thought-out code and understanding how to use it is made easy by the documentation available on the Bootstrap website. As previously stated, a very common use for the framework is prototyping a new site. This is because the developer is able to worry strictly about content and functionality and put very little effort into styling.

2.1.3.2 Consistency

In a large project, there will often be multiple developers working on and writing CSS classes and utilizing them before committing their code changes. Because everyone has a unique way of coding, this can often cause inconsistencies in code structure and styling. With Bootstrap, this is no longer an issue since all coding is based off of the structure of the pre-written CSS.

2.1.3.3 Customizability

It may seem as if Bootstrap locks a developer into one website look and feel. However, the framework is very easily customizable through the CSS pre-processors called Less and Sass (Rahman, 2014). Without going too much into these technologies, there is page on the Bootstrap site where one can change the default values for attributes of each included component. By doing so, it is simple to choose the look and feel of Bootstrap with hardly any hassle.

2.1.3.4 Responsivity

Cisco's Global Mobile Data Traffic Forecast Update (2015) predicts that in the coming years, mobile cloud traffic will increase 11-fold between 2014 and 2019. As such, a major issue in web development is getting a site to both function and look great on multiple platforms. Because Bootstrap uses something called the "mobile first grid system", developers get amazing control over how a page looks on different platforms without putting in too much effort (Tutorial Republic, n.d.).

2.1.4 Downsides to Bootstrap

2.1.4.1 It Clutters the DOM

When writing HTML, elements on a page (i.e. 's, <p>'s, etc.) get added to an application programming interface (API) called the Document Object Model (DOM). This helps maintain a structure and allows for access and manipulation of HTML page elements (Wood, Le Hegaret, & Robie, 2000). It is generally considered good practice to keep the content and functionality of a webpage separate from the styling (Walton, 2013). This is so that modifying the styling does not affect functionality and vice versa. When using Bootstrap, classes control the styling of each element. This ends up filling the DOM with class-modified elements, effectively combining the content with the page styling (Zing Design Blog, 2013).

2.1.4.2 Bootstrap is a Large, Heavy Resource

Bootstrap's CSS and JavaScript files take 126KB and 29KB respectively. As such, the load time of a Bootstrap site may be larger than others. This will always be a problem if one uses the framework as is (Zing Design Blog, 2013). However, by taking the time to customize the CSS and only including the code one needs, this can be remedied to some degree.

2.1.5 How Bootstrap Applies to Our Project

Endicia would like a unified, easily replicated look and feel to both their current and future web based endeavors. We believe that Bootstrap could provide a great base for this. By utilizing the prewritten CSS and JS, we will better be able to focus on providing a great user experience and a beautiful appearance rather than using our time to write code. Furthermore, we can use the customization features it has and easily and efficiently tailor the elements to our needs. Any additional

functionality or customization we need to add could be implemented on top of the base Bootstrap framework.

2.2 AngularJS

AngularJS is a JavaScript framework for creating dynamic web applications. It is developed and maintained by Google (AngularJS -Superheroic Javascript MVW Framework, n.d.). Its goal is to make single-page applications run with ease and success.

2.2.1 Advantages of Using AngularJS

2.2.1.1 *Model-View-Controller (MVC) Compatibility*

AngularJS is a MVC based JavaScript framework. Traditionally, data models and view-dependent controllers are built on the server side. AngularJS brings those services to the client side, which greatly relieves the burden on servers. It also automates the process of component management and simplifies the code HTML view with software components, saving developers a lot of time and effort.

2.2.1.2 *Directives*

AngularJS can extend HTML attributes using directives to allow for more functionality. Directives enable developers to customize and reuse HTML tags that change the behavior of some elements. Examples of some directives are ng-app, ng-controller, ng-view, etc.

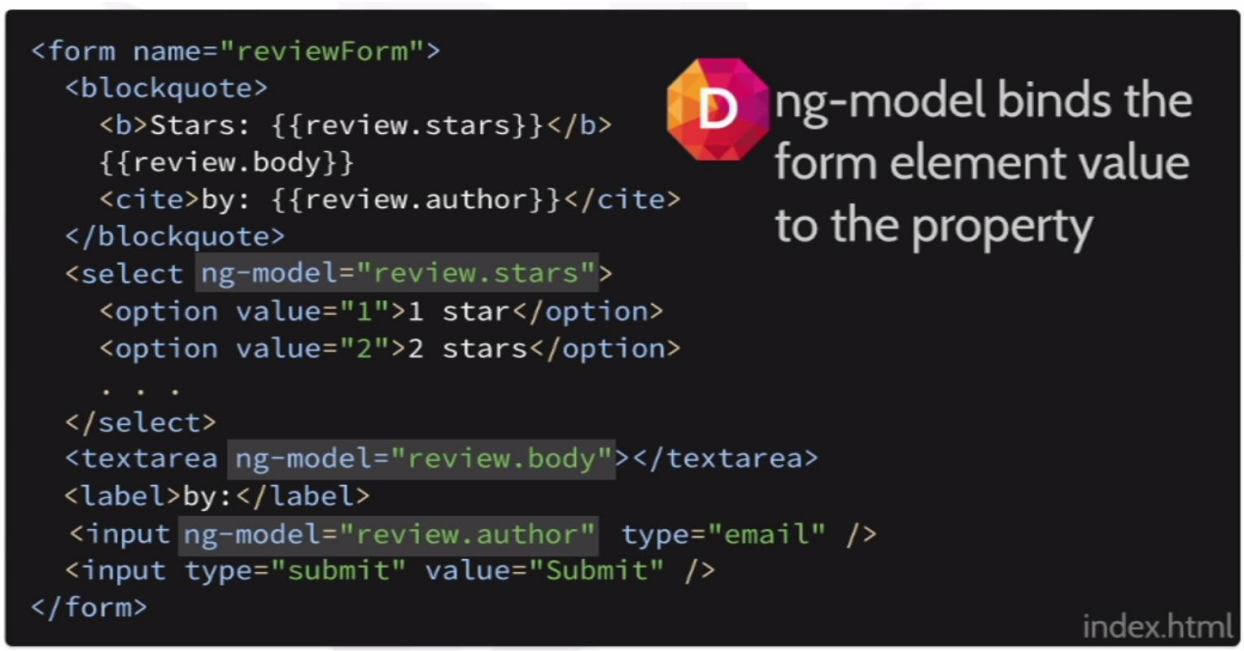
Figure 1 is an example of using directives. In the index.html file, the directive ng-controller was added, pointing to the "StoreController". When the page loads, it will automatically trigger the StoreController function in app.js file. This way calling a function is easy and the functions are highly reusable.

Figure 1 Example of using directives

2.2.1.3 *Two-way data binding*

Data binding is one of the most useful features in AngularJS. It enables automatic data synchronization between the model and view components. Data binding in Angular treats the model as the single-source-of-truth; that is, the view is a projection of the model at all times. Whenever the model changes, the view reflects the change, and vice versa (AngularJS: Developer Guide: Data Binding, n.d.). This would also allow for ease of testing since the controller is completely separated from the view.

Figure 2 shows a simple example that demonstrates the two-way binding several input elements: a multiple choice selection, a textbox and a email input.




```
<form name="reviewForm">
  <blockquote>
    <b>Stars: {{review.stars}}</b>
    {{review.body}}
    <cite>by: {{review.author}}</cite>
  </blockquote>
  <select ng-model="review.stars">
    <option value="1">1 star</option>
    <option value="2">2 stars</option>
    . . .
  </select>
  <textarea ng-model="review.body"></textarea>
  <label>by:</label>
  <input ng-model="review.author" type="email" />
  <input type="submit" value="Submit" />
</form>
```

Figure 2 Example of implementing two-way data binding

This piece of code is creating a form, where users can post reviews to products. The <blockquote> is where users can view all past reviews and add their own reviews to. Below <blockquote> are the input elements. The binding is done using just one simple directive, ng-model. Now there is two-way binding between the view, <blockquote> and the model, the input elements, every time a user types in the form, the content will be live updated in the review history, as demonstrated in Figure 3.

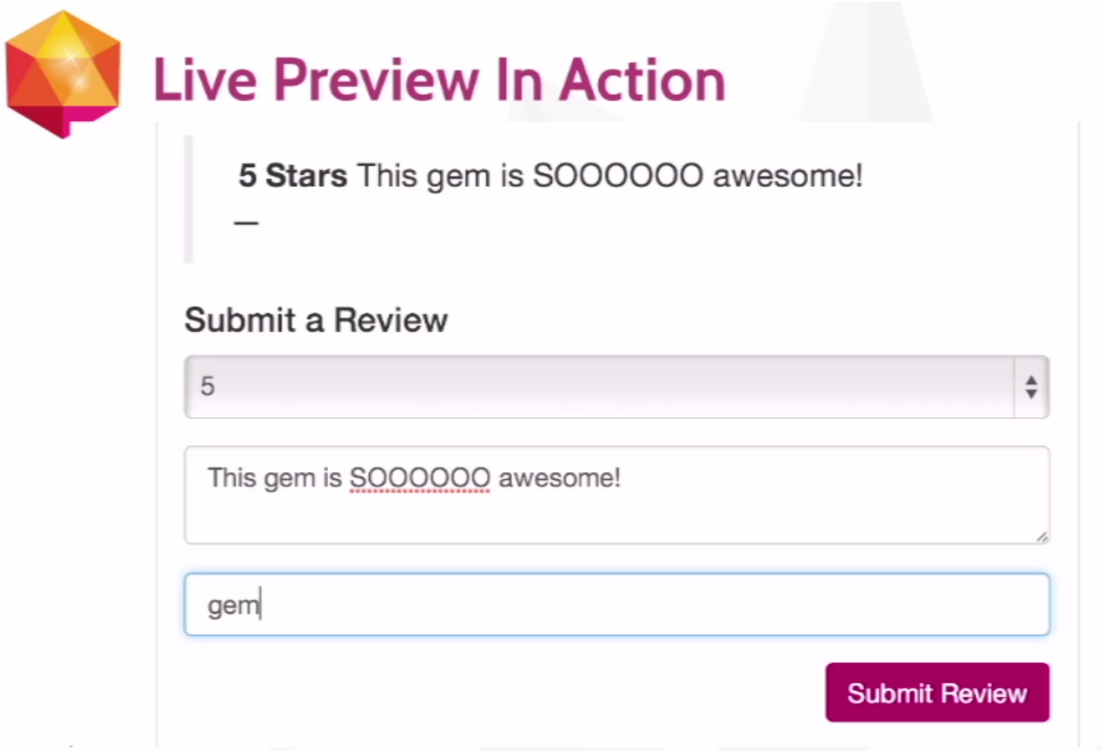


Figure 3 Output of the code

Figure 4 and Figure 5 shows how to bind data with checkboxes and radio buttons.

With a Checkbox

```
<input ng-model="review.terms" type="checkbox" /> I agree to the terms
```

Figure 4 Example of binding data in a checkbox

With Radio Buttons

```
What color would you like?

<input ng-model="review.color" type="radio" value="red" /> Red
<input ng-model="review.color" type="radio" value="blue" /> Blue
<input ng-model="review.color" type="radio" value="green" /> Green
```

Figure 5 Example of binding data in radio buttons

2.2.1.4 Dependency Injection (DI)

AngularJS has a built in DI dependency injection subsystem that is in charge of creating components, resolving dependencies, and providing them to other components as requested. DI can

also increase the testability of code by injecting mock data into controller and measuring the output and behavior.

2.2.2 Disadvantages of Using AngularJS

2.2.2.1 Difficulty of modifying existing code

A good portion of AngularJS code resides in the html files. If the project is large enough, it could be difficult to adapt existing code since the developers would have to re-implement the entire transformation code.

2.2.2.2 Heaviness

AngularJS is a very comprehensive and sometimes bulky framework. It might be good for building large applications, but in case of smaller applications with simple requirements on data binding, it might be a little too much and could confuse developers with excessive functionality that might not be very useful. Other lighter frameworks might be a better option in such situations.

2.2.2.3 Insufficient Security

AngularJS is a JavaScript only framework. Applications that solely rely on AngularJS are not safe. To keep and application secure, server side authentication and authorization are must-haves.

2.3 Razor

Razor is an ASP.NET view engine—it renders the HTML in your view to the browser (ASP.NET MVC - Basic overview of different view engines, 2012). It was first released with the ASP.NET MVC 3 framework in 2010, and supports C# and VB.NET with HTML (Introducing "Razor"--a New View Engine for ASP.NET, 2010). Razor made integration of server code with HTML templates much simpler. Before then, the only way for developers to integrate server code with HTML templates was to use the ASPX syntax—enclosing server code in <%= %> blocks. ASPX syntax can look quiet redundant when writing multiple lines of server code. Consider the example from Figure 6:

```
<ul id="products">

    <% foreach(var p in products) { %>
        <li><%=p.Name%> ($<%=p.Price%>)</li>
    <% } %>

</ul>
```

Figure 6 ASPX Syntax Example

In this simple example for creating a list of product names and prices, it was necessary to use the <%= %> block four times. Razor supports better integration of server code in HTML templates. With Razor, the code from Figure 6 can be simplified to what is shown in Figure 7:

```
<ul id="products">

    @foreach(var p in products) {
        <li>@p.Name ($@p.Price)</li>
    }

</ul>
```

Figure 7 Razor Syntax Example

Looking at Figure 7, the @ sign is used to indicate the start of server code. There is no need to explicitly close the code block since Razor is smart enough to know when your server code ends.

In addition to the simplification on server code integration, Razor also has some other advantages over common APSX syntax. Among these is the ability to add “layouts”. With these, Razor enables you to create a common layout (similar to the ASP.NET master pages) that can be used by all pages in your project. This fosters a consistent look throughout a web page (ASP.NET MVC 3, n.d.). Another advantage is that Microsoft has provided full IntelliSense for Razor based files in Visual Studio, which helps to minimize typing and improve productivity in development.

2.4 Information Architecture

2.4.1 Introduction

Information architecture is the design of content structure and organization in hopes that information can resonate with a target audience (Information Architecture Basics, n.d.). It is an extremely important part of the web design process, yet is often overlooked (Chapman, 2010). In this section we will provide some fundamental knowledge on information architecture design patterns, techniques, and styles.

2.4.2 Design Patterns

Depending on the amount and structure of information we want to present, different design patterns will be used. Some of the common design patterns include (Hunt, 2006):

All-in-one: all information in a single page;

Flat: all pages are organized as peers of others;

Index: a homepage with several subpages;

Hub-and-spoke/Daisy: a main page connecting several branches for different workflows;

Strict hierarchy: design where lower-level page is only accessible through its parent;

Multi-dimensional hierarchy: website has a hierarchical structure, but the same content can be accessed in many different ways;

Search: using search tool to access contents.

Originally, Endicia envisioned the site structured with a Hub-and-spoke design. This is because Supervisory was designed in the same way, with a main page and links to the rest of the site. However, it was quickly seen, as we discuss later, that allowing users to customize the site would be very important. From this, we later turned to designing our site to resemble a multi-dimensional hierarchy. Different users could have different home pages and links to different parts of the site appear in many different places, sometimes redundantly for convenience.

2.4.3 Design Techniques

Design techniques for information architecture design include card sorting (writing the content of each page on an index card before sorting the cards into different groups to help find out how the site should be structured), creating wireframes and prototypes, site mapping, and outlining (Chapman, 2010). These design techniques will be useful throughout our project. For example, we might want to use wireframes and prototypes for the overall design of our site since all pages need to be a uniformed style. We could then use a sitemap to outline the structure of the entire site.

2.4.4 Design Styles

The two basic styles of information architecture design are top-down and bottom-up. With the top-down approach, the overall structure of the main site is decided at the beginning. Details on content relationships are decided over time. With bottom-up design, content relationship details come first. The designer might start by using user stories to find out how the fundamental content blocks should be linked together. The overall structure of the entire site will be decided over time (Chapman, 2010).

In later chapters we will be referring back to this knowledge and discussing the routes we took. Specifically, we'll touch upon which design we used and how it aided us in our design process.

2.5 Web Design and User Experience

2.5.1 Introduction

Web design consists of algorithms, software engineering, and interfaces. A good design will maximize a website's efficiency in delivering its functionality while providing a great user experience. The following sections will discuss various aspects of user experience and how it is related to this project.

2.5.2 User Research

The goal of user research is to understand behaviors, needs, and motivations of the target audience through user analysis and task analysis. Commonly used methodologies include observation, interviews, surveys, heuristic evaluation etc. A good design should always center on the user and their common tasks.

2.5.3 User Interface (UI) Design

UI design focuses on anticipating what users might need to do and providing easy-to-access elements to respond to their needs. The choices of elements and the layout should be consistent and predictable. Maintaining consistency among elements and layout can help with task completion, efficiency and satisfaction.

2.5.4 Interaction Design

Interaction design is creating engaging interfaces with well thought out behaviors. It aims to understand how users and the technology communicate with each other. This knowledge, allows anticipation of how someone might interact with a system.

2.5.5 Visual Design

Visual design is about the aesthetics of a website and its related materials. It is the art of strategically choosing and placing images, colors, fonts and other visual elements. A good visual design will help engage the users and build trust and interest in the Endicia brand.

2.5.6 Usability Evaluation

Usability evaluation assesses how easy an interface is to use. It is a combination of learnability, efficiency, memorability, errors, and satisfaction. Usability is important because if people find it hard to use a website, they will likely leave and switch to other alternatives. Therefore, considering usability is crucial. Among the many methods used to improve usability, the most common is user testing. This will be discussed in detail in the methodology chapter. Some basic metrics to evaluate usability include speed of performance, rate of errors by user, time to learn, retention over time, and subjective satisfaction.

2.5.7 Accessibility

An accessible website is built to cater both the needs of the general population as well as the disabled. It will present information via multiple channels such as sound or sight and will provide additional means of navigation to allow for a higher level of interactivity. By improving accessibility, websites can ensure a pleasant user experience for all potential users.

3 Methodology

3.1 Supervisory and Sputnik

To get started on the project, the first thing we did was familiarizing ourselves with Endicia's two administrative websites—Supervisory, the old website to be migrated, and Sputnik, the hardly implemented new site.

The classic Supervisory site had little to no design or structure, and was essentially a list of links with minimal categorization and hardly any navigation assistance. Some of the frequently used pages were hidden two or even three levels deep and the font was extremely small, limiting the accessibility of the website. Figure 8 shows a screen shot of the classic Supervisory zoomed at 100%.

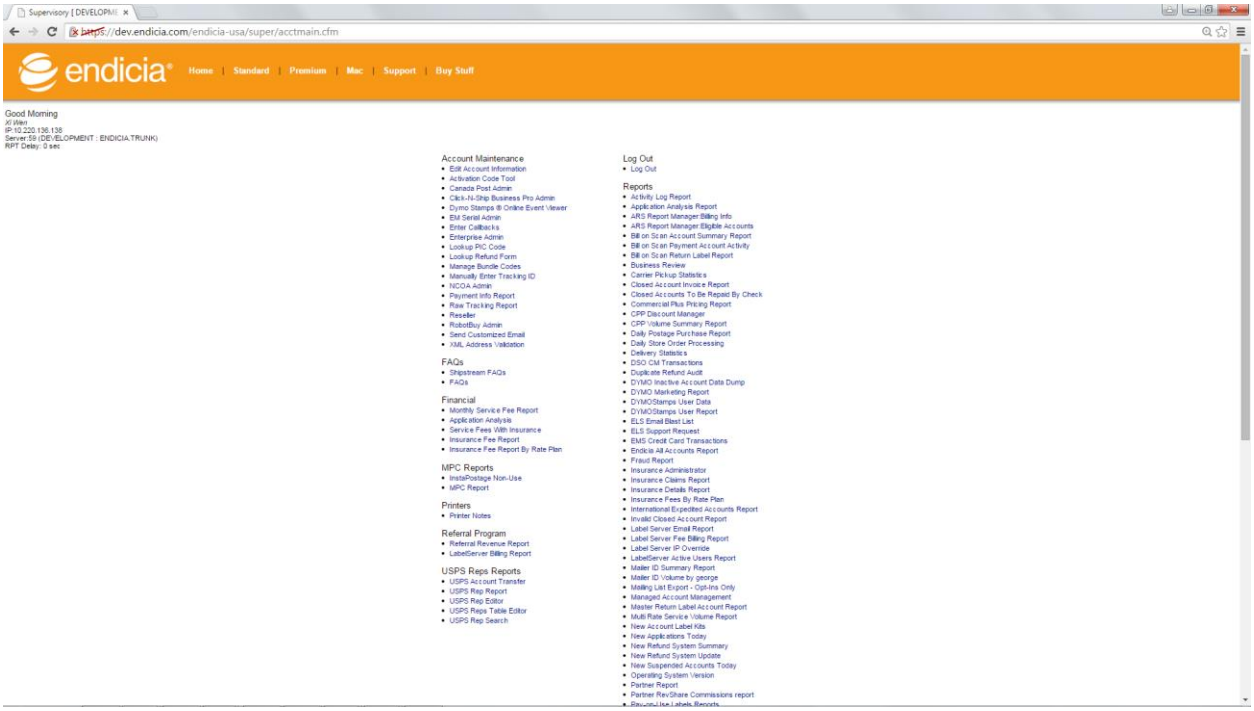


Figure 8 Supervisory Main Page

The Supervisory backend was built with ColdFusion (Adobe, n.d.), a web scripting language developed by Adobe that enables developers to use conditional statements, user defined methods, and database commands in HTML templates. However, as Supervisory grew, Endicia decided to move all non-.NET source code to .NET. From this, they decided Supervisory should follow suit. Furthermore, Supervisory had become much larger than originally planned. Taking these two facts into account, Endicia decided to port Supervisory to a new site in order to utilize newer technologies for the backend and at the same time improve UX and UI. This new site became known as Sputnik.

Construction of Sputnik began approximately a year ago. The frontend was built with Bootstrap and AngularJS with ASP.NET for the back. While it would seem that a year is an ample amount of time for large amounts of progress in the development cycle, the dev team had halted the construction. They knew that if they continued without any templates or styling guides, Sputnik would grow into another Supervisory—a site with bad information architecture, inconsistent styling, and a terrible user

experience. Because of the halt, the implementation was sparse, even lacking a main page. Figure 9 shows a screen shot of the Sputnik main page at the beginning of our project.

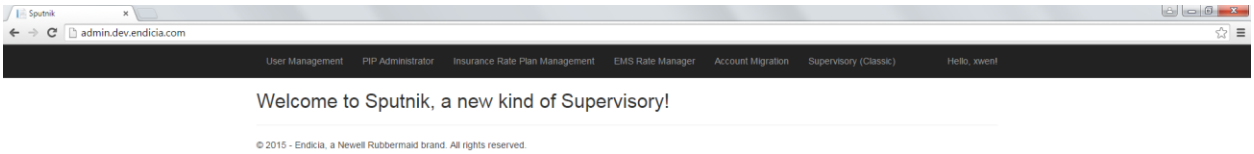


Figure 9 Sputnik Main Page

This main page was the perfect place to start on our project since efficiency of navigation through the site would largely be decided by the relevance of links on the main page. To get this page right, we began gathering requirements from each of the teams at Endicia.

3.2 Requirements Gathering

In order to get a comprehensive understanding on the usage of Supervisory, we began by meeting with employees from a number of different departments and discussing the ways they used it. The most important takeaway from these discussions was the fact that people from different departments generally use different sections of Supervisory. For example, people working in technical support would mostly use the edit client account functionalities; on the other hand, people from the sales department use the reports functionality extensively. To see exhaustive notes taken during these meetings, see Appendix I: User Interview Notes. Because of this wide range of use cases, we saw that a static main page that included links important to all the teams would be impossible. The main design principle quickly became customization. If each employee could tailor his or her main and intermediate pages, we'd get much closer to an ideal solution.

From the meetings, we also learned of common navigation issues, account management inconvenience, and other backend/database related problems that should be addressed. Many pages in the classic Supervisory do not have *Back* buttons or links to the home page. Even if *Back* buttons/links to home page are present, they are often positioned at locations that are hard for to find even for experienced users.

Another feature Supervisory lacks is a centralized account management tool. There is no way for user to view his/her account settings or change password easily. This problem really bothered users who have to change their account password frequently. We also received requests for new functionalities that require significant amount of backend work. Due to the fact that we only had seven weeks and our focus was the UI and site structure of Sputnik, we knew we would likely not get to many of the aforementioned issues. With so much information gathered, we moved to analyzing the overall usage of Supervisory through existing data.

In order to better understand how the Supervisory site was being used across different sets of users, we looked at logs kept by Endicia that track each link click in supervisory and the user who initiated it. We broke down the data by department to see the usage across each of the teams in the company, as shown in Figure 10. The site was most heavily used by Tech Support followed by Sales and Business Development.

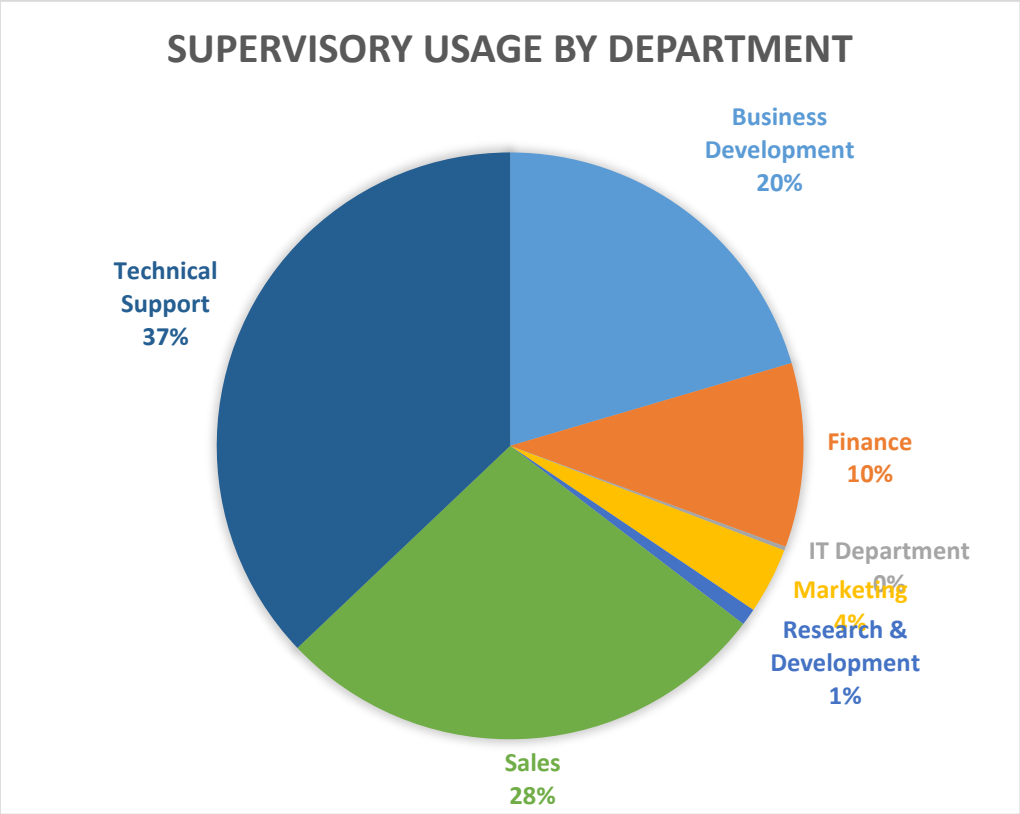


Figure 10 Supervisory usage by department

To see if employees were using links as related to their respective departments, we neglected the user departments and instead clustered them according to their behaviors and the expected behavior of each department. This is visualized in the attached external file supervisory_usage_cluster.xlsx. As can be seen in Table 1, the results show that employees do indeed seem to stick to functionality related to their department. Fourteen users that were identified as Technical Support were actually from the department, and three out of five users who were identified as

Business Development were from the department. We therefore came to a conclusion that the links on the Sputnik website could remain grouped by functionality and usage as they were on Supervisory.

	-	BizDev	PIP	Power	Product	TechSupp	Grand Total
	3						3
Business Development	5	3	2	1			11
Finance	1			5			6
IT Department	1	1					2
Marketing	11	1			2		14
Research & Development	7			1	1		9
Sales	33			17	5		55
Technical Support	19			5		14	38
Grand Total	**80**	**5**	**2**	**29**	**8**	**14**	**138**

Table 1 Relation between user behavior and department

To get a better understanding of the important links on the site, we closely analyzed the number of clicks on each link and derived some interesting results. For instance, we observed that every department uses links from both Account Maintenance and Reports. This can be seen in Table 2 below by checking that each column in the Account Maintenance and Reports rows has values. Sales reps rely heavily on USPS Reps Reports. IT Department mostly uses Reports and some Account Maintenance, but nothing else. Again, all these facts are seen in the Table 2.

	Business Development	Finance	IT Department	Marketing	Research & Development	Sales	Technical Support	Grand Total
Account Maintenance	644	262	5	108	29	721	823	**2619**
Crypto						1		**1**
Dealer Program	1						1	**2**
Financial	14					18	29	**61**
Log Out				2	3	14	43	**62**
Operations		5						**5**
Referral Program	6	1				8	54	**69**
Reports	173	237	42	61	15	400	595	**1531**
USPS Reps Reports	84	30		99	4	556	219	**992**
Grand Total	**922**	**535**	**47**	**270**	**51**	**1718**	**1764**	**5342**

Table 2 Count of click-through of each link group by department

The data has a certain level of inaccuracy since there could be unintended clicks, especially with those links with less than 10 clicks. Therefore, we created a supplementary survey to investigate which of the lesser-used links are actually not useful, as well as what the real users think they use most often to better complement the data from the usage logs. We also asked for each employee's ideas and hopes

for the look and feel of the Sputnik website. The survey can be seen in Appendix II: Supervisory Usage SurveyAppendix II: Supervisory Usage Survey**.**

3.3 Wireframes

3.3.1 Key Elements

Before building a wireframe for the home page, we had several elements in mind, enumerated and explained below.

3.3.1.1 The Home Button

In Supervisory, many pages do not have a “Home” or “Return to Main Menu” button. This problem greatly increased the difficulty for users to navigate through the site.

3.3.1.2 The Search Bar

A powerful search box was built for Sputnik during a hackathon at Endicia. This search box could quickly search for accounts numbers, tracking information, and pages in the site. Figures Figure 11, Figure 12, Figure 13 are screen shots for this search bar. Because of the efficiency and scope of this search box, we wanted to incorporate it into our design.

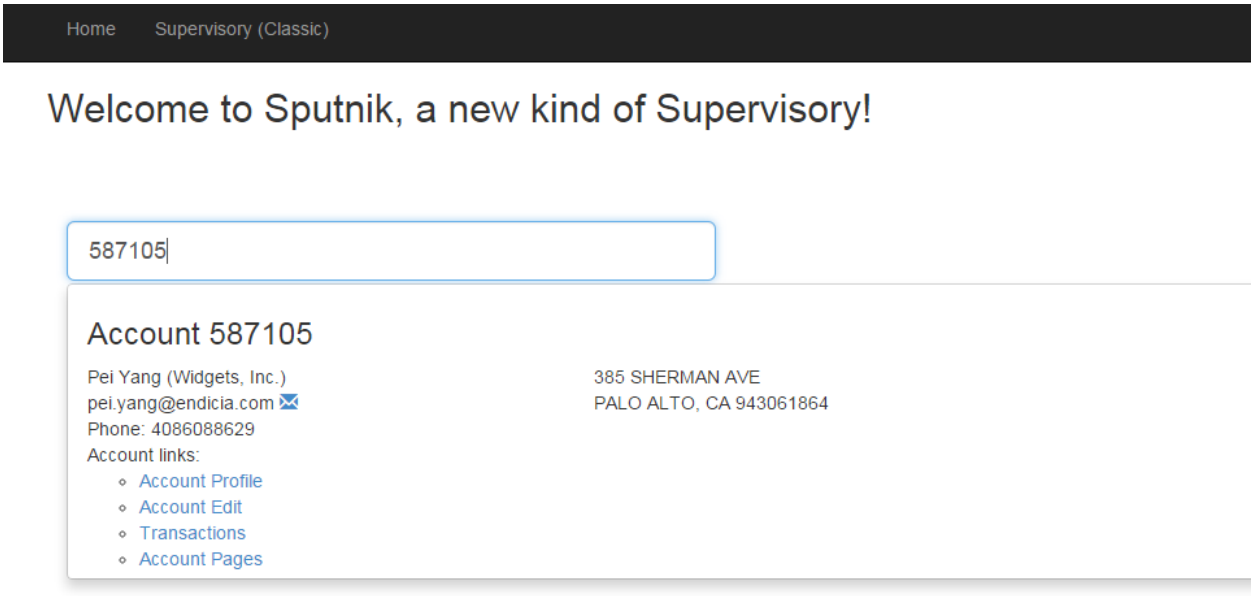


Figure 11 Search Box Screen Shot 1

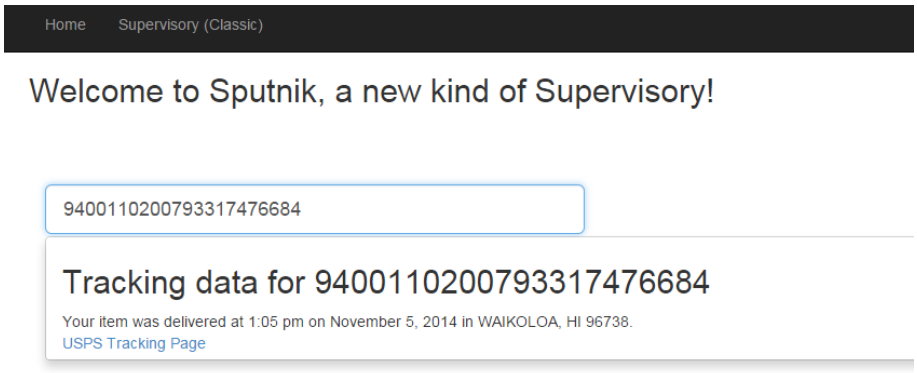


Figure 12 Search Box Screen Shot 1

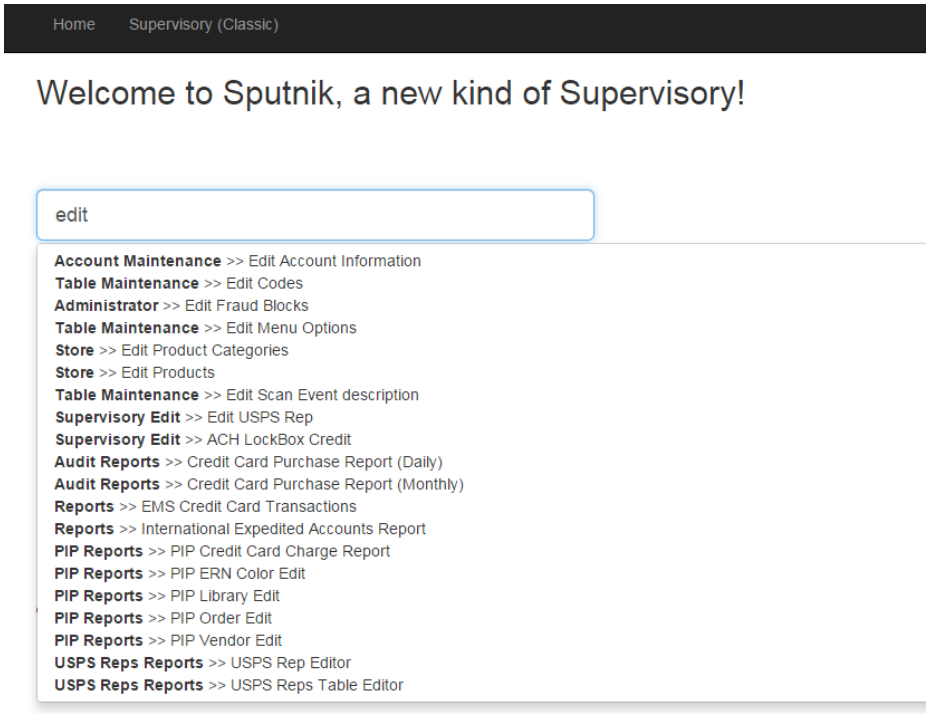


Figure 13 Search Box Screen Shot 2

3.3.1.3 The Account Center

In Supervisory, there was no centralized area for the user to see his/her account information. This left personal account maintenance to be a chore, where even changing one's password was impossible without having to use the "Forgot Password" link. We planned to incorporate an account maintenance page to add transparency and simplicity for employees.

3.3.1.4 The Like Button and the Favorites Bar

As stated in the previous section, the most important concept in our mind while designing the new Sputnik was customization. We wanted each user to have the ability to customize the homepage, and in order to achieve this, we decided add a *Like* button to each page. If the user clicks on that button, that page will be added to the user's *favorites* list.

3.3.2 Home Page Wireframe Iterations

With the home page being such an important piece of the new design, it was one of the most disagreed upon aspects and took many iterations to get right. The team decided that the best place for the aforementioned key elements would be a universal navigation bar since this would make the elements easily accessible from any page. While this navigation bar remained essentially the same throughout each new design (save for one added link shown later), the actual page content was changed drastically.

3.3.2.1 Iteration 1

Our first attempt was heavily focused around using a large accordion menu in place of the massive list of links from the Supervisory home page. This can be seen in the wireframe shown in Figure 14 below.

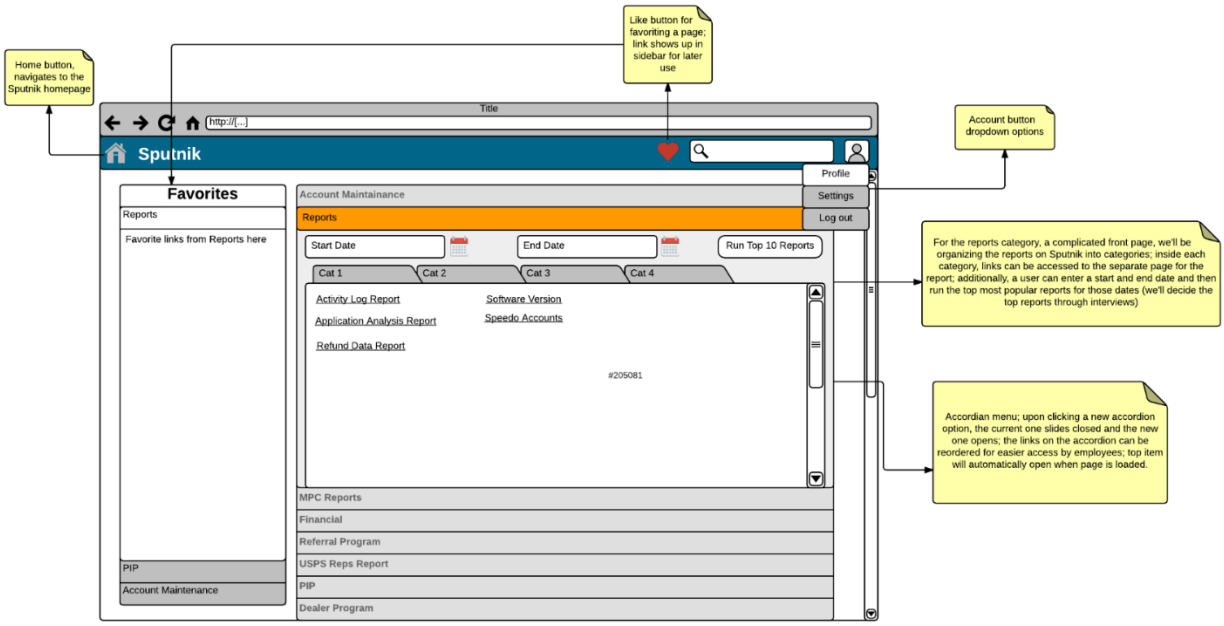


Figure 14 Home Page Wireframe, Accordion Menu

As stated, we decided that every page of Sputnik would have the navigation bar at the top of the page. Down below, we've illustrated the *Favorites Bar* on the left side of the screen, a side panel that could be hidden to make space on the page.

To the right of the favorites, we've shown the accordion menu layout that organizes all links currently on the Supervisory home page. We envisioned the first accordion fold automatically opening once the page was loaded. The best part about this design was that the user would also be able to configure the order in which the accordion options are presented, allowing them to place the more frequently used categories near the top. As a whole, we believed this design would be able to give each user the chance to custom tailor his or her experience and, in the process, improve efficiency. Of course, we soon saw that we could do even better.

3.3.2.2 *Iteration 2*

Our second attempt came in response to the large amount of feedback we received during our discussions with employees. There was concern about the accordion in that so much page real estate seemed wasted by the folds of the accordion. Furthermore, having only one fold open at a time would get annoying for users going back and forth between two different folds. From a technical standpoint, an accordion menu is also tougher to implement since the state must be tracked at all times.

Because of these three issues, it was made clear to us that a change was necessary. We began to brainstorm new designs and came up with the layout shown in Figure 15.

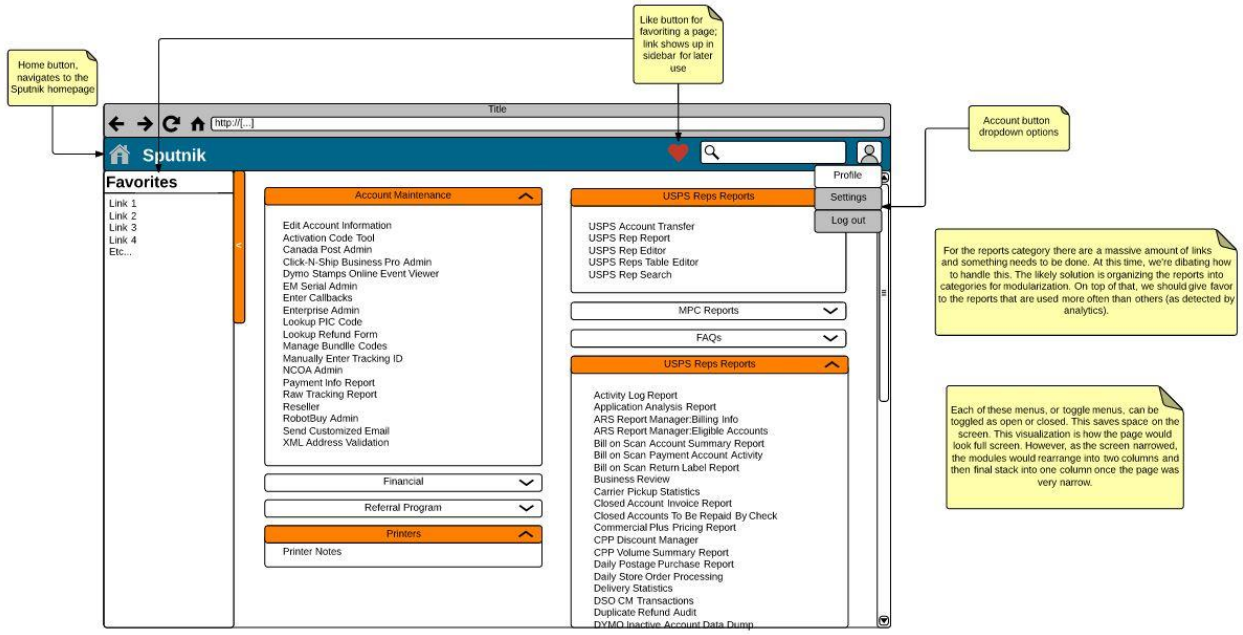


Figure 15 Home Page Wireframe, Toggle Menus

The main advantage we saw with the accordion was the ability to open and close different menu sections, effectively freeing up page real estate for other content or menus. To keep this ability, we thought of using toggle menus that could be opened or closed whenever needed. Also, instead of stacking the menus, we thought having two columns would make better use of space since the links in each menu were not likely to be overly wide. We additionally envisioned the ability to drag each toggle menu to the desired position. This would allow each user to place more important menus in easily accessible locations.

It's important to note that the navigation bar and favorites panel remained virtually unchanged. The only difference in the favorites bar was the removal of a nested accordion menu.

3.3.2.3 *Iteration 3*

After the creation of the wireframe in iteration 2, we began implementing toggle menus with primitive functionality in order to demonstrate the viability of the design. The prototypes were received very well and seen as a large improvement over our accordion menu design. However, there was one issue that was brought to our attention.

It had been made obvious to us from our meetings with employees and our own experience that the main issue with Supervisory was the huge amount of links on the home page. This overload would confuse anyone, and was not useful when employees use only a small subset of the links depending on their department. Our first two iterations had given users the ability to customize their home page to some degree, but the most useful idea we'd come up with was the favorites bar.

The favorites bar would allow users to keep a small subset of links available at all times. This fact would likely cause users to visit the home menu much less once their favorites bar was set up. It was suggested that the main page not even be a menu of links at all, but rather something more useful. Of course, the large page of menus would need to be included somewhere on the site to keep everything accessible without the favorites. We decided that an extra icon would be included in the navigation bar that would link back to the large menu of links (which we began to call the Site Map).

As for the actual content of the main page, we knew that customizability was once again the key. With this in mind, we decided to allow for users to choose a default page from Sputnik that would become their landing page. If no default was chosen, then the user would be presented with a search bar centered in the middle of the page (the smart search shown previously in Figures Figure 11, Figure 12, and Figure 13). The finalized wireframe is shown below in Figure 16.

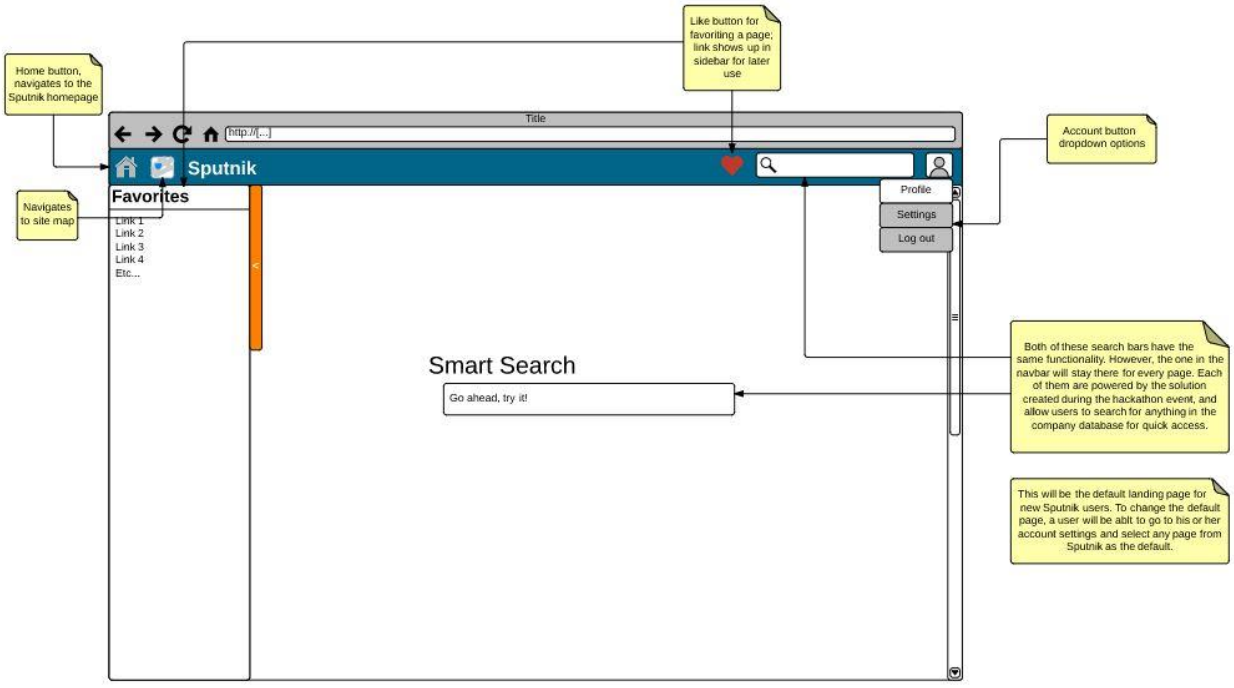


Figure 16 Home Page Wireframe, Smart Search

The favorites remained untouched between this and the last iteration. The top navigation bar hardly changed at all, save for the addition of the site map button. To reiterate, the site map button links to the page of toggle menus and links shown in Figure 15. This design was decided during the fourth week of our project, a little over halfway through, and was finalized only a few days later.

3.4 The Sputnik Developer Hub

Designing and creating the home page of the new site was certainly a large part of our project. However, we also needed to create a site for developers to visit and use as they ported over the hundreds of old Supervisory pages. This site for developers, which we called the Sputnik Developer Hub, would house a large number of resources meant to make the creation process much easier. This section discusses the different facets of this developer site and explains how the final product will make life very easy for developers rewriting old Supervisory pages.

3.4.1 Supervisory Page Types

The first thing we wanted to provide developers was an easy way to classify each page of Supervisory and group it with other pages on the site that were similar. To do this, we needed to create a number of page categories that would define every page from Supervisory. We decided the best criteria for categorizing different page types would be the set of elements needed by a page.

For instance, a page with a large form would need text input boxes, text fields, checkboxes, radio buttons, date pickers, submit buttons, etc. On the other hand, a page displaying retrieved customers from an account ID search might need a data table with pagination buttons, filtering of results, etc. Thus, by looking at the types of elements employed by every page on Supervisory, we would be able to create a uniform set of categories to define the pages regardless of their size or purpose.

To find different categories and then finally classify each page, we went through the entire Supervisory page by page and created a set of page types as we went. The different types we found are enumerated below. The *Contains* section in each describes the elements a page of that type might consist of. The *Assign* section lists the different aspects we had to decide upon when creating elements for these different categories (i.e. we had to decide the width of every text input box and assign it with Bootstrap to ensure similarity between inputs).

Form Page

Contains
Text, text areas, dropdowns, radio buttons, check boxes, date picker

Assign
Width, heights, margins, styles, alignment, icons, and element styles

Data Table Page

Contains
A table with data and possibly editable fields

Assign
Row height, column width, color (alternating)

Toggled Menu Page

Contains
Menu blocks that can be toggled as open or closed. These blocks can contain links, data, forms, anything really. However, they will most often be used for links.

Assign
JS functionality, width, heights, styles, alignment

Left Navigation

Contains
Static left navigation panel on the left side of a screen. This navigation panel should allow access to all pages on the same nested level on Supervisory. For instance, when looking at the Page Type table, it can be seen that Account Features, Account Status, Account Transaction, etc. are all nested within Edit Account. As such, each of these pages should have a left navigation panel on them allowing access to the other pages within Edit Account. Supervisory had this, but was inconsistent at best.

Assign
JS functionality, width, heights, styles, alignment

Tabbed Menu Page

Contains
Menu with tabs horizontally at the top. Pages are displayed inside each tab.

Assign
JS functionality, width, heights, styles, alignment

Java App

Contains
A Java Application with proper heading and description

Assign
Width, heights, styles, alignment, and area for app

It's important to note that pages do not have to have only one type. In fact, they can have any combination of types. For instance, a typical Report page from Supervisory first requires the user to fill out a number of fields such as *Start Date* and *End Date*. By our categories, this would require the elements included in a Form Page. However, once the user has filled out the fields and submitted, the page displays a table with results based on the user entered criteria. This would require the elements included in a Data Table Page, causing the page to have two types. Furthermore, the page might also

contain a left navigation bar, requiring elements from the Left Navigation category. As stated, the combinations are endless and are exhaustive in that they adequately classify every page in Supervisory.

With these categories finalized, we were able to create a page on the Sputnik Developer Hub that listed each of the above definitions. This page is shown below in **Error! Reference source not found.**.

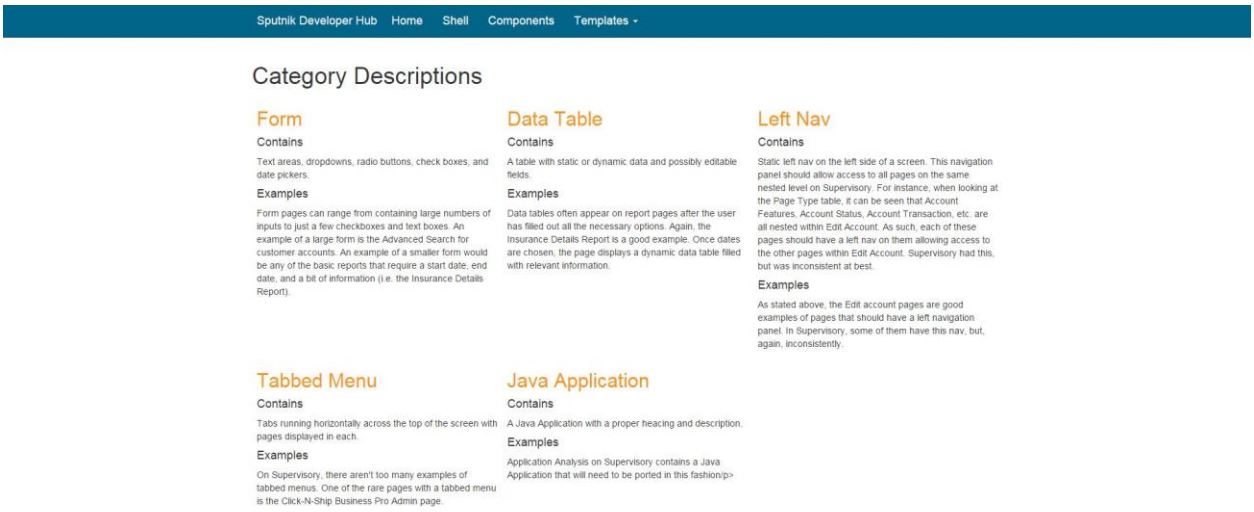


Figure 17 Category Descriptions Page on the Sputnik Developer Hub

3.4.2 Page Classifications Spreadsheet

As we went through all of Supervisory creating page categories, we kept a spreadsheet that kept track of the type of every page we saw. We also recorded issues that we found with pages along the way. A snippet of this sheet is shown below in **Error! Reference source not found.**.

Page	Issues	Form	Data Table	Left Nav	Tabbed Menu	Java App	
Process LabelServer Billing		X	X				
Quick Print Survey	ERROR						
Raw Data Reports				X			
› Raw Data for Accounts Report		X		X			
› Raw Data for Account Feature Report		X		X			
› Raw Data for BML Charges		X		X			
› Raw Data for CAPS Charges		X		X			
› Raw Data for FDMS EMS Charges		X		X			
› Raw Data for FDMS PSI Charges		X		X			
› Raw Data for FDMS USPS Charges		X		X			
› Raw Data for IMPB Codes		X		X			
› Raw Data for Mailer Sequence Number		X		X			
› Raw Data for PicNumber		X		X			
› Raw Data for Postage Discount Report		X		X			
› Raw Data for PTS PicNumber		X		X			
› Raw Data for Tracking		X		X			
› Raw Data for Transaction		X		X			
Reconciliation Report		X	X				
Refund Data Report		X	X				
Refund Electronic Audit		X	X				
Refund PCP-X Form Generator		X					
Resource Software Billing Report		X	X				
Revenue Potential		X	X				
SCAN Form Limit Override		X					
SCAN Statistics		X	X				
Service Fees Bank Reconciliation		X	X				
Shared PES Exception Report		X	X				
Software Version		X	X				
Speedo Accounts			X				
Store Report		X	X				
Subscription Report		X	X				
Survey Tool Report		X					
Suspended Accounts Report		X	X				
Top Mail Classes By Month		X	X				
Top Users By Mail Class		X	X				
Transaction Count		X	X				
Transaction Count Beta Report		X	X				
USPS Sales Report	BROKEN	X					
Verify Locked Account Report		X	X				
Yesterdays Suspensions			X				
ZIP Code Revenue Report		X	X				
Crypto Functions	BROKEN						
Manual Account Adjustment	BROKEN						
Endicia All Users Report		X	X				
Locked Account Report (not working)							
TOTALS	35	209	148	123	7	3	525

Figure 18 Page Type Spreadsheet Snippet

By examining the Totals in the Excel sheet, it can be seen that Forms were the most common types of pages. In second came the data table type. This result was expected since, as previously stated, many of the Reports pages had a form that would produce a data table once filled out.

We created a page on the Sputnik Developer Hub that displayed this spreadsheet and allowed it to be downloaded. We see this page being extremely useful to developers. Once they have a page they are looking to rewrite for Sputnik, they simply come to this sheet and find the category of the page. This coupled with the templates described in the next section will greatly improve efficiency and uniformity in the creation process of new Sputnik pages.

3.4.3 Page Templates

In our opinion, the most important part of the Sputnik Developer Hub is the set of Templates we created. With categories for each type of page and the elements contained within said categories, we designed page templates corresponding to each category. These page templates included all the elements and styles needed to create a new page of a certain type. They also had code snippets so that a developer could copy and paste code straight from the template into their project and immediately have an html shell for their page. This would allow them to focus more on functionality rather than layout and styling consistency.

Overall, we created five different template pages corresponding to the categories described in the previous section. To reiterate, the five templates were Form, Tabbed Menu, Data Table, Left Nav, and Java Applet.

3.4.3.1 Form Template

The form was the most common template across Supervisory pages. Nearly every page required elements that appear in the form template. These elements include different sized text inputs, different sized description titles, definitions corresponding to said descriptions, date picker elements, text areas, checkboxes, radio buttons, and dropdown lists. Finally, code snippets for each of these elements were included. Since there was a rather large amount of HTML to display, we decided to include a left navigation bar for easily sifting through the snippets. The Form template page can be seen below in Figure 19.

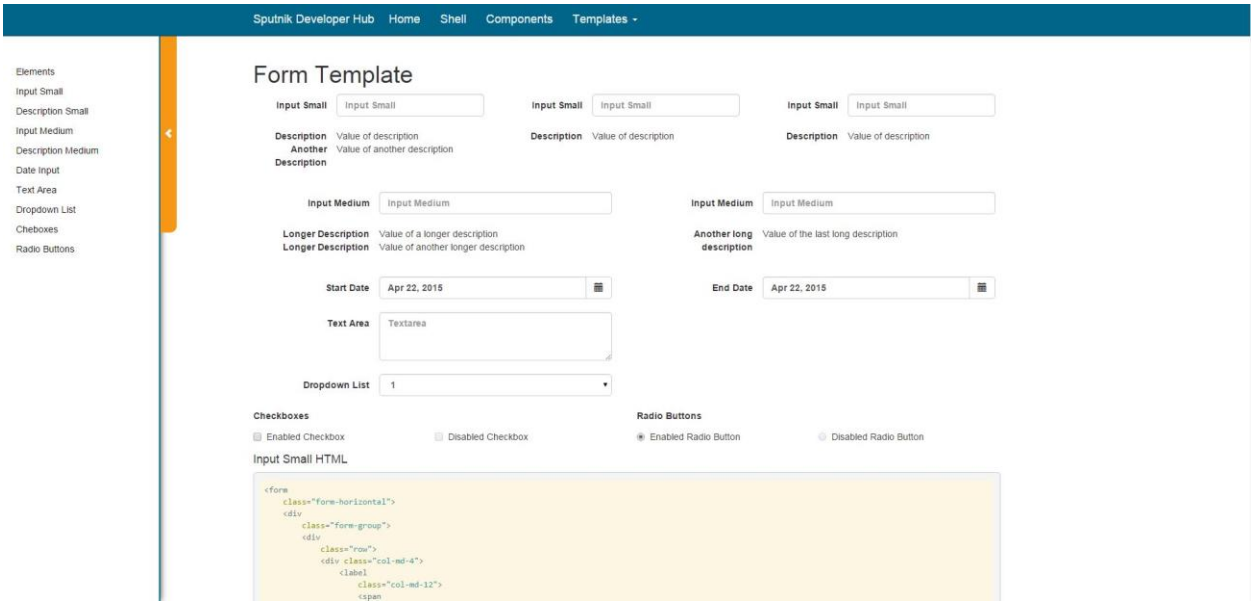


Figure 19 Form Template

3.4.3.2 Tabbed Menu Template

The next template was the tabbed menu, a simple looking page with an easy implementation. Because there were rather few tabbed menus throughout Supervisory, we decided to make the HTML code static instead of supplied by Angular. In other words, the different tabs are specified directly in the

HTML rather than by some backend source. We make this distinction because, as we discuss later, the left navigation does require an outside service to be supplied with the different options. The Tabbed Menu template is shown in Figure 20.

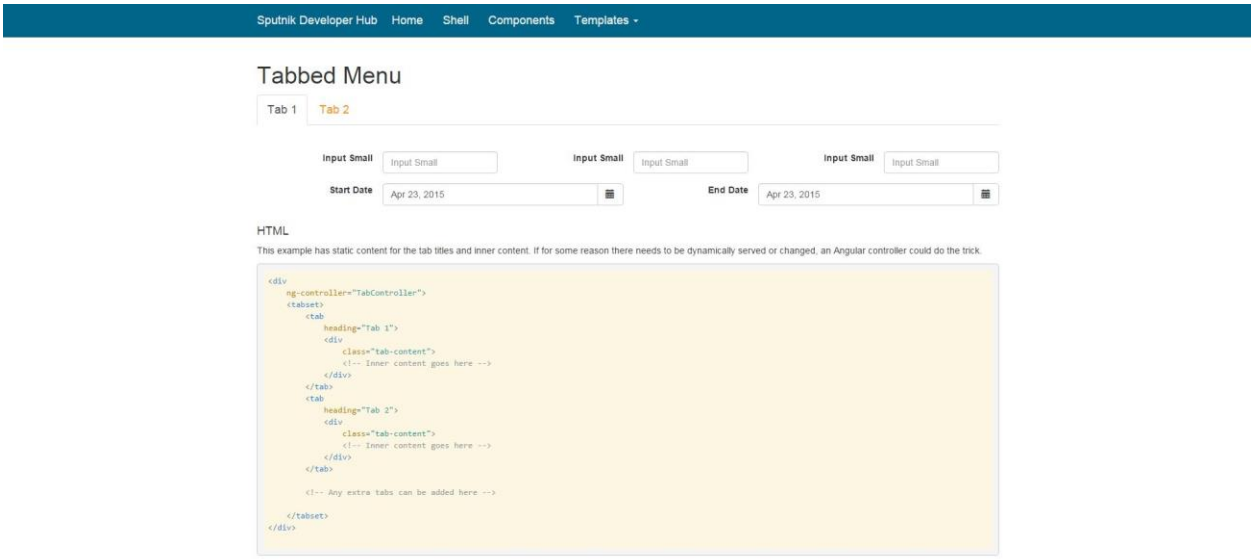


Figure 20 Tabbed Menu Template

3.4.3.3 Data Table

Data Tables were another very popular occurrence in Supervisory, and as such they needed to very versatile. To ensure tables were easily implemented and could be populated by the backend, we used an AngularJS plugin called “angular-datatables”. Because it was written using Angular, we were able to pass configuration of the table through a service which in turn could provide an URL for querying the server for data from the backend.

Angular-datatables also came with an extra set of functionality, such as result filtering, pagination, sorting, server-side processing, etc. These were necessary to have since many of the pages on Supervisory returned massive amounts of data to be displayed in tables. We needed a way to display smaller subsets of data instead of all the results at once. Pagination and filtering were vital for this, and with server-side processing, the amount of data transferred upon each draw is greatly reduced while the speed of processing (filtering, sorting, etc.) is increased. The Data Table template is displayed in Figure 21.

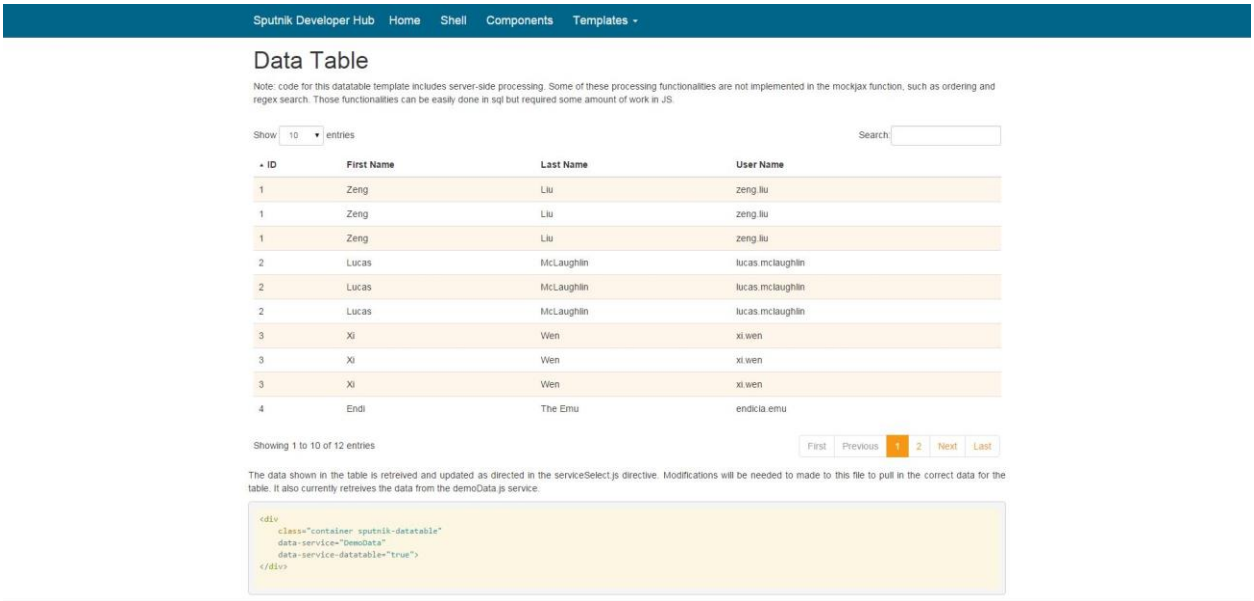


Figure 21 Data Table Template

3.4.3.4 Left Nav

The Left Navigation was yet another common occurrence throughout Supervisory, so once again we knew it needed to be versatile. A large reason for this was the fact that users might have different sets of permissions for seeing certain links. As such, we needed a way to allow developers to filter the links that appeared in each left nav section. The best way of doing this was creating a service that supplied the links to a left nav section. By modifying the service, developers could retrieve links from the backend and then filter them based on permissions. Below in Figure 22 is a screenshot of the Left Nav template.

Figure 22 Left Nav Template

3.4.3.5 Java Applet

The Java Applet did not occur very often on Supervisory, so the template was rather simple. The only issue was that embedded Java apps have a set size and width. As such, when a page is made smaller, the layout of the page can be messed up by the app. To remedy this, we ensured to only display

a Java app when the page was a certain minimum width. Whenever the page was below that minimum width, a title would appear informing the user that they must enlarge the page a bit. Figure 23 shows the final Java Applet template.

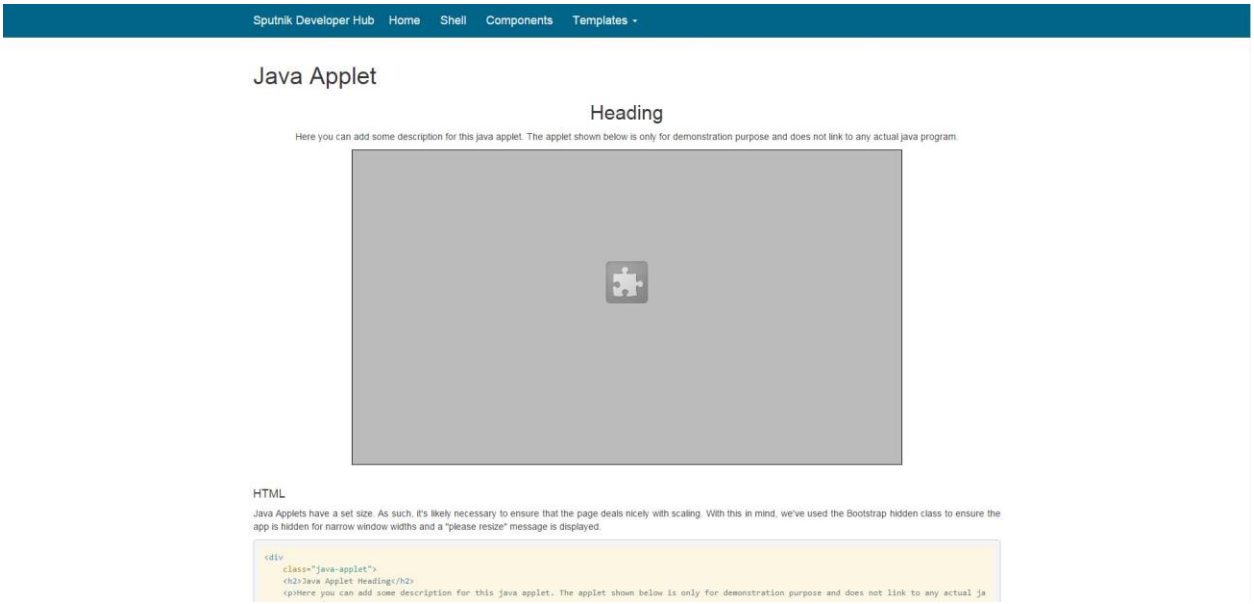


Figure 23 Java Applet Template

3.4.4 Components Page

While we believe that our categories and templates were exhaustive and will cover the recreation of every page from Supervisory, we considered the fact that developers may need more than what we provided if an entirely new page for Sputnik was created. Furthermore, we did not receive access to a small number of pages from Supervisory. With these facts in mind, we knew we would need to include a page with the entire set of elements developers should be allowed to use. This page, called the components page, would include elements that had not yet appeared in our template pages (i.e. loading progress bars).

The final product was massive with a similar layout to the Components page employed by Bootstrap. Just like our template pages, it also included code snippets for each of the components, allowing developers to copy and paste code straight into their projects. The Components page is shown below in Figure 24.

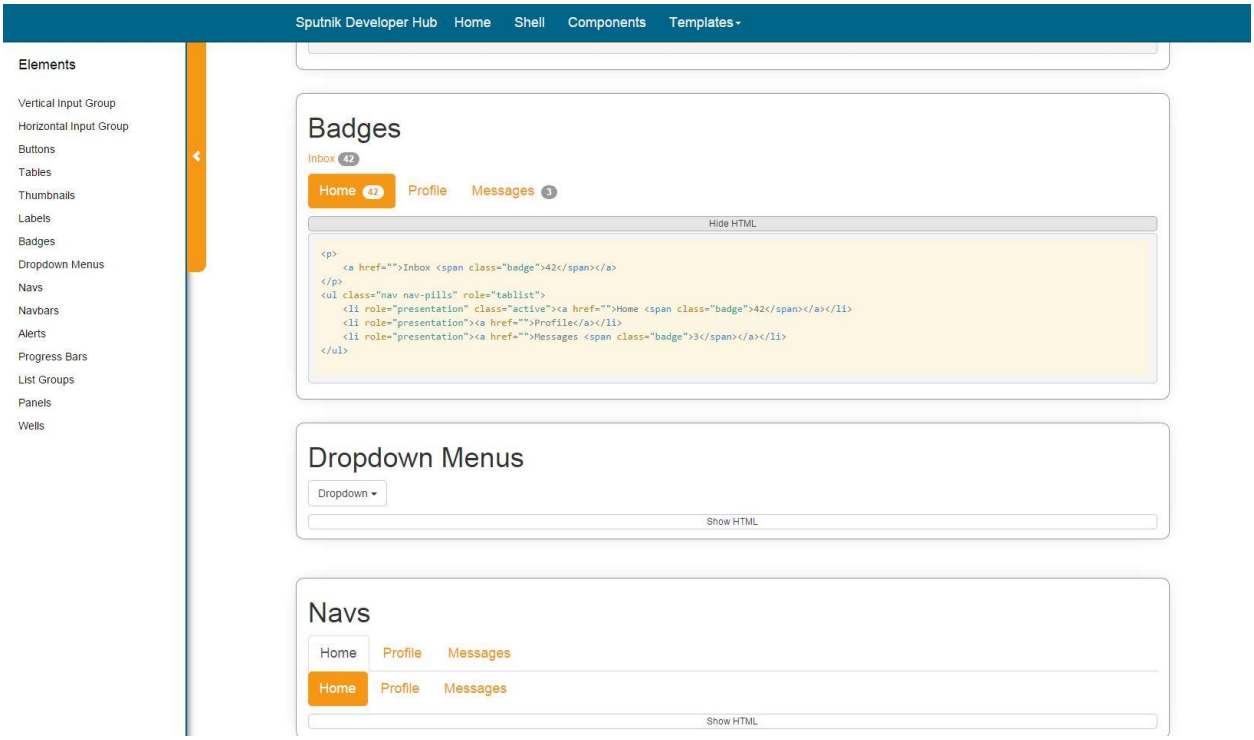


Figure 24 Components Page

3.4.5 Expected Workflow

To recap, we created the Sputnik Developer Hub as a site for developers to visit during creation of pages for Sputnik. On it, we included types to categorize every page from Supervisory, a spreadsheet to determine the type of a Supervisory page, and templates for each category with code snippets for easy copy and paste application. Additionally, we created a large page of all the components a developer should be allowed to use. The reason we believe these pages could be so useful for developers is because they will improve efficiency during the creation process as well uniformity across new Sputnik pages.

Without the templates housed in the Sputnik Developer Hub, a developer would have to blindly redesign a page from scratch. This means that they would need to spend a considerable amount of time contemplating the design of the page. Once a design was decided upon, they would actually need to write the HTML and CSS for the design taking up more valuable time. Finally, they would be able to move on to working on the page functionality. Once finished with the page, they would have no guarantee whether or not it would fit the layout or style of the other pages on Sputnik.

However, with the templates, we envision a much simpler and efficient workflow. Given a page to port over from Supervisory, a developer would first make their way to the Page Classifications spreadsheet and would find the templates needed to create said page. If the page was not found in the spreadsheet, they would go to the Category Descriptions page and decide which templates the page would need.

Once the developer knows which templates to use, they would then navigate each template's page, copy the necessary code snippets, and paste them into their project. If they needed different components than were provided on the page template, they would navigate to the Components Page, find the necessary elements, and paste the code of each into their project. The developer would then be able to move on to coding in page functionality. This entire workflow can be seen below in Figure 25. A larger version can also be viewed in Appendix III: Expected Workflow for Future Developers.

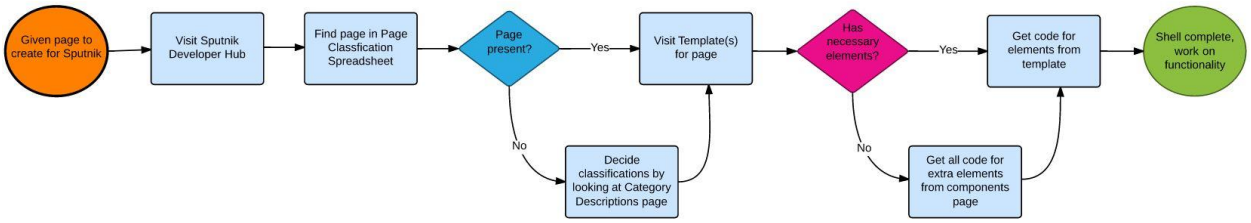


Figure 25 Expected Sputnik Developer Hub Workflow

It is the simplicity and linearity of this workflow which convinced us that the Sputnik Developer Hub will be an amazing tool for developers.

3.5 The Page Shell

Alongside the Sputnik Developer Hub, we were charged with designing the home page of Sputnik as well as a common shell with elements that would appear on every subsequent page. Wireframes and design choices behind the final home page can be found in Section 3.3.2.3. To reiterate, we decided to make the default home page very primitive by solely including a search bar as content. If the user wanted a different home page, they could select a different one in their Profile settings.

However, the large, centered search bar on the home page is not considered to be part of the page shell. The page shell includes the top navigation bar and the favorites bar. These should appear on every page as a user navigates through Sputnik. The final home page implemented in code is shown below in Figure 26.

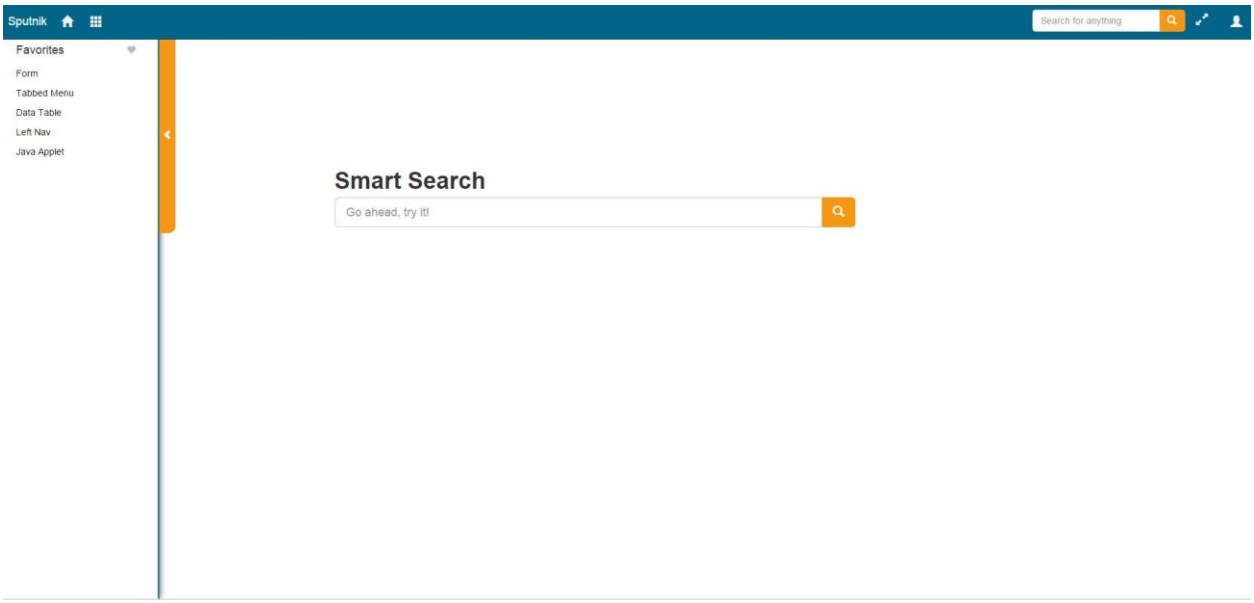


Figure 26 Page Shell and Home Page Content

3.5.1 The Top Navigation Bar

It was imperative that Sputnik have a top navigation bar since the old Supervisory site had no such static bar to navigate around the site. Each element of the nav bar is described in the following sections.

3.5.1.1 Home Button

The home button is the most intuitive among the rest of the nav bar elements and is denoted by a small icon in the shape of a house. When this button is pressed, the user is sent to the Sputnik home page. For each user, the home page could mean something different since it can be changed in the Profile settings. If it is not modified, however, the user will return to the page in Figure 26. Since Supervisory had no home button, we saw this as an important addition

3.5.1.2 Site Map

The site map, the small icon with nine boxes in a grid, was meant to allow the user to access every link on Sputnik quickly and easily. A big issue with the original Supervisory was the massive set of links on the home page. Rather than put these links onto the home page, we decided to abstract them into their own separate page. The decision behind this was discussed in Section 3.3.2.3.

In the actual code implementation, the site map button toggles a dropdown. Whenever it is pressed, the full site map covers over the current page content as shown in Figure 27. It can then be pressed again to make the site map disappear and reveal the content beneath. We wanted this toggle behavior rather than requesting a new page since it saves overhead and allows for quick access to the navigation.

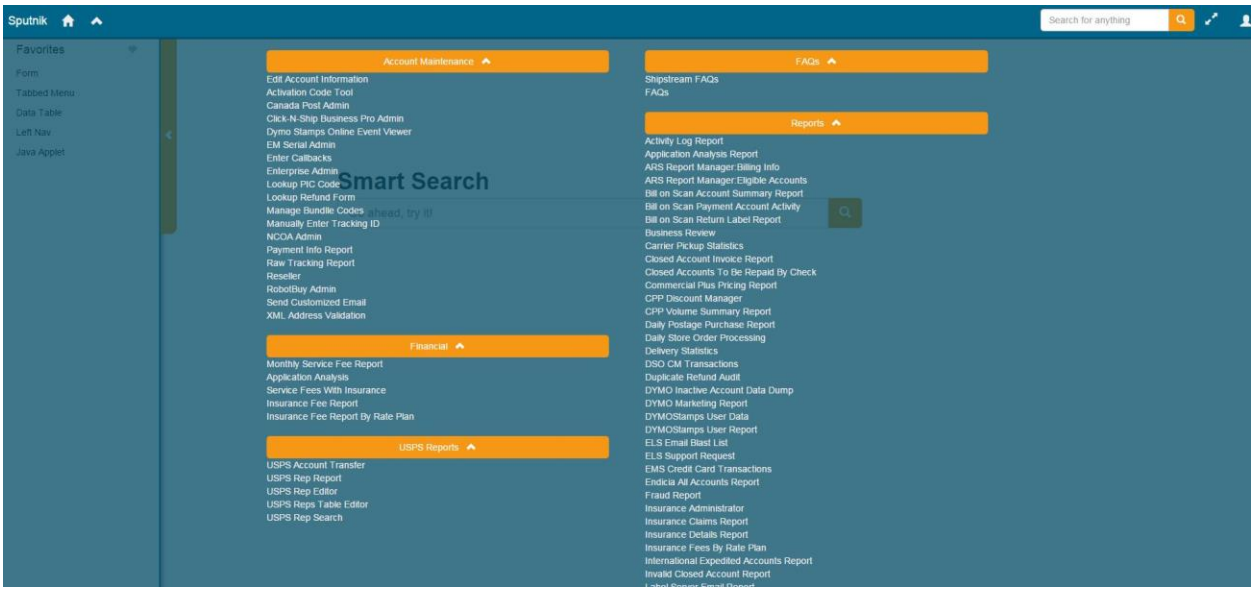


Figure 27 Site Map Open

3.5.1.3 Search Bar

The search bar provides the user with a way to search for virtually anything related to Sputnik. The details behind the functionality were discussed in Section 3.3.1.2. Because of the power of this addition, we decided to include it in the top nav bar, always readily available for users.

3.5.1.4 Advanced Account Search

While the search bar allowed for generalized searches based on one string of criteria, this did not allow users to drill down and use a precise set of information to search for customer accounts. Because of this, we decided to add the advanced account search pane to the nav bar. Much like the site map, this button toggles a dropdown which covers the current content instead of requesting a new page. This is illustrated below in Figure 28.

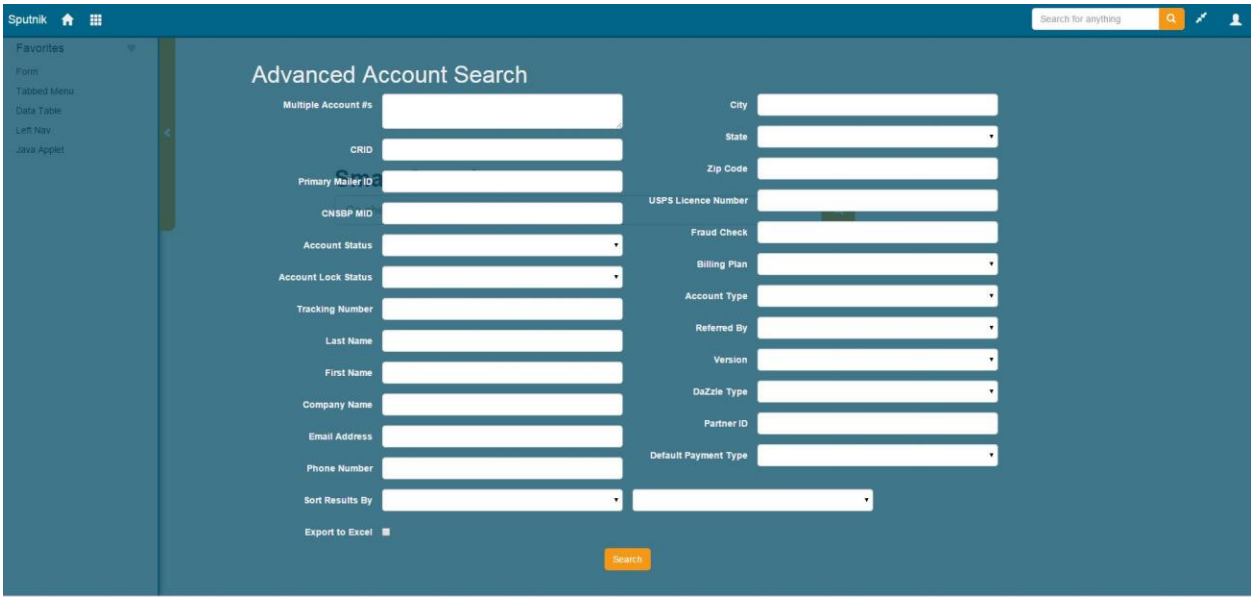


Figure 28 Advanced Account Search Pane Open

3.5.1.5 Profile

Supervisory did not include any information about a user's profile and as such included no customization options. We saw the addition of user profile settings and information as a must, and as such we included a profile button in the nav bar. Unfortunately, we were unable to fully implement the profile functionality in the time allotted to us. Nonetheless, we began creating backend functionality for easy integration at a later date.

3.5.2 The Favorites Bar

3.5.2.1 Default Favorites Lists

The favorites bar was the most important addition for giving users a custom tailored experience with Sputnik. It allowed easy access to any part of the site by allowing users to add Sputnik pages to their own, personal favorites list. Once we made the decision to move away from a home page full of links, we knew the favorites bar would have to be populated from a user's very first visit to Sputnik. However, because of the small, narrow size of the bar, it was obvious that it could not contain a massive set of links to the entire site as the original page did. The set of links would need to be intelligently picked for each user depending on their team in the company.

To choose appropriate lists of links for each department in the company, we combined the data from the usage logs and survey results (discussed in Section 3.2, Requirements Gathering). Because the survey results were more subjective than the data collected from the usage logs, we decided to weight the importance of the results of each differently. Table 3 shows how we determined the weights assigned to survey results.

Because we had small sample sizes relative to the populations, we assigned relatively lower weights to the survey results. If a sample size for a department was somewhat large in comparison to the population, we assigned the survey results a weight of 30%. If it was average, then we assigned the results a weight of 20%. If small, a weight of 10% was assigned. For some of the departments, i.e. the Finance department, the full population would not even be enough to produce a valid normal distribution. In those cases, it was obvious we needed to assign the lowest weight of 10%. Table 3 shows the departments, their populations and sample sizes, the ratio of sample size to population, and the final assigned weight.

Once we had the survey weights, we subtracted them from 100% for each department to get the usage log weights. For example, the sales department had a weight of 30% for survey results and as such were assigned a weight of 70% for the usage logs.

Department	Population	Sample Size	Sample Size/Population	Weight
Technical Support	64	13	20.31%	30.00%
Sales	82	18	21.95%	30.00%
Business Development	16	5	31.25%	20.00%
Finance	7	1	14.28%	10.00%

Marketing	31	9	29.03%	20.00%
Research & Development	91	10	10.99%	20.00%
IT Department	10	1	10.00%	10.00%

Table 3 Assigning weights to survey results by department.

With the data from the usage logs and surveys as well as weights for each, we moved to actually ranking links by order of importance for each department. We calculated the importance of each link through a series of mathematical steps. First, we took the ranking of the link according to the survey results and multiplied it by its survey weight. We then took the ranking of the link according to the usage logs and multiplied it by the corresponding weight. Finally, we added the two computed values together. This process is illustrated in the following example:

> *If for Marketing Department, Link A ranked 1st in their survey results, and 5th according to the usage logs, the final ordering of the link would be:*
>
> $$1 * 0.2 + 5 * 0.8 = 4.2$$

By using this method, we created the final ordered lists of links for three link categories: Account Maintenance, Reports, and USPS Reports. Because of their small number of links and unimportance, we did not order any of the other categories using this complex method. Instead, we calculated their ordered lists solely based on the usage logs. A completed list of default links by category and department can be found in Appendix IV: Default Favorite Lists by Department.

3.5.2.2 Customizability

Once we had finished analyzing the data and had come up with default favorites lists for each department, we moved on to adding the core functionality of the favorites bar. The default favorites list would ensure that new users had pre-populated favorites. However, we needed to add the ability to add, delete, and reorder them as desired. This would require an intuitive interface for users to edit their favorites as well as a way to store the state of their updates in the backend database. In this section, we discuss the interface for the favorites panel. We discuss the backend implementation in section 3.6.

The interface for the favorites was designed to be simple and easy for new users to comprehend. Location-wise, we decided to place the panel on the left side of the page and allow it to be toggled as open or closed. Below in Figure 29 is the basic look of the home page with the favorites panel open, and in Figure 30, the home page with the favorites panel closed.

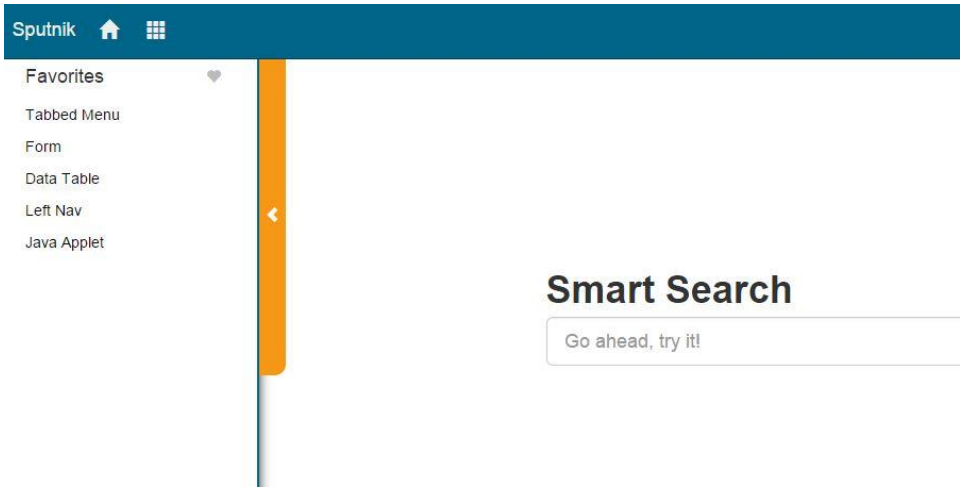


Figure 29 Favorites Panel Open

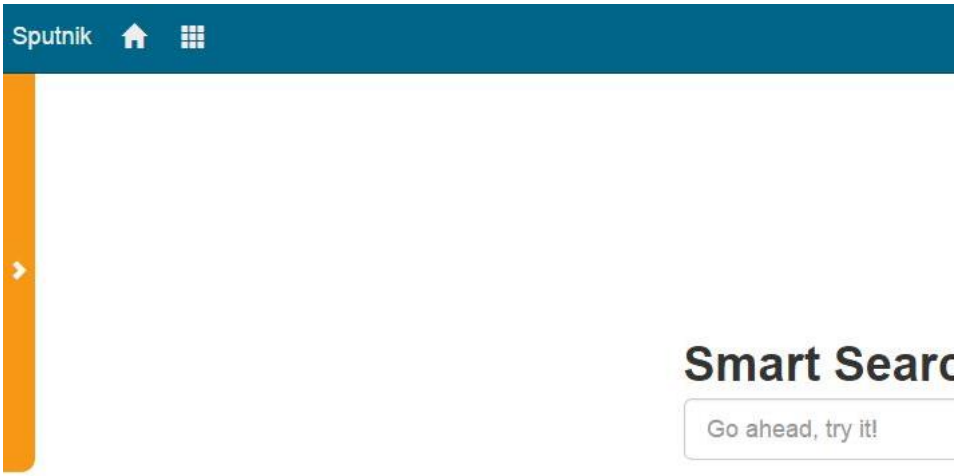


Figure 30 Favorites Panel Closed

If a user clicks the heart inside the panel, a new favorite is added for the page they are currently on. Each of the links can be removed at any time and can also be dragged for reordering. The buttons for making these edits are revealed when a user hovers over the panel, as shown in Figure 31.

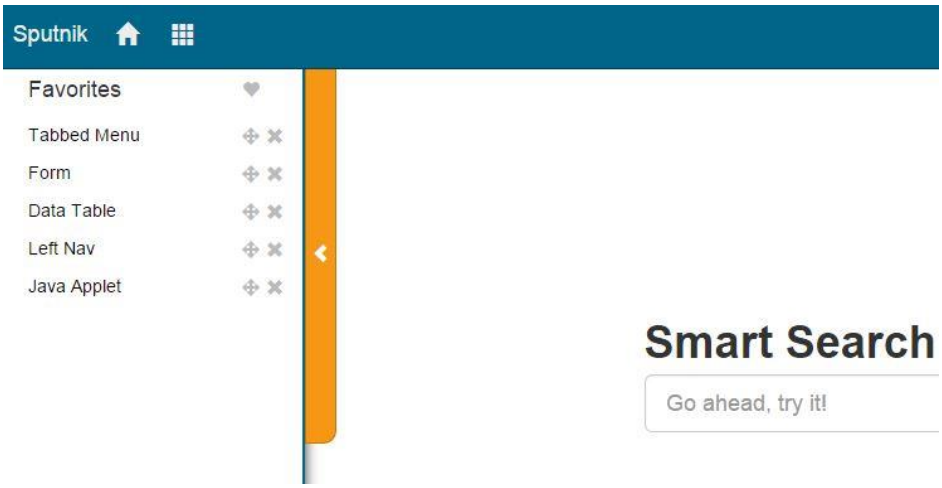


Figure 31 Favorites Panel With Edit Buttons Visible

The rightmost button, shaped like an X, allows a user to delete the favorite. The leftmost button with four arrows allows a user to click and drag the favorite up and down and reorder the list. This dragging is shown below in Figure 32.

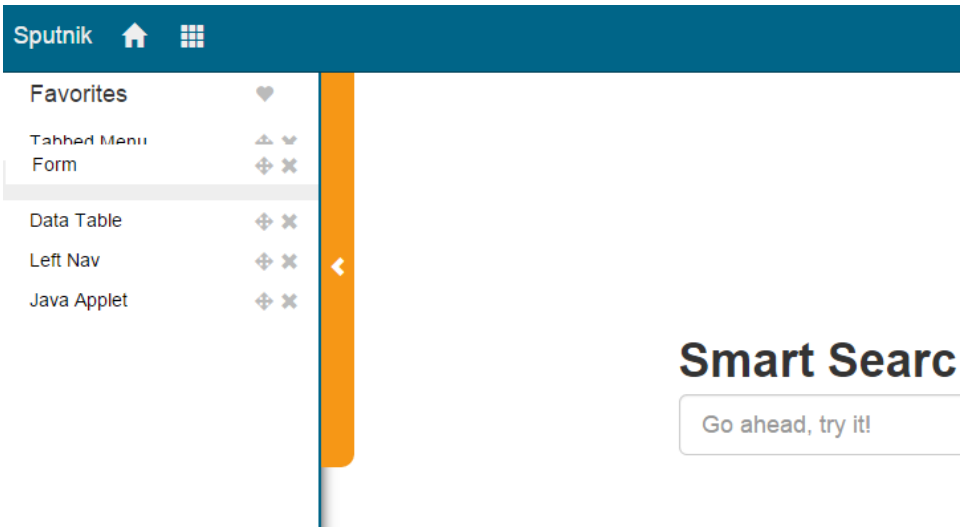


Figure 32 Favorites Drag to Reorder

This drag-to-reorder interface was chosen because of its simple nature and pervasive use throughout computing. For instance, file browsers such as Finder on OS X or Windows Explorer on Windows allow users to drag files between folders. The idea of picking something up and moving it around is inherently intuitive because it is what we do with objects in the real world on a daily basis. As such, we saw it as the best way to implement reordering from a UX standpoint.

3.6 Backend Development

Throughout the project, the biggest focus was customizability of Sputnik for each and every user. The great thing about customizability is that it allows everyone to have their own experience on a site, and from this find just as much convenience with it as the next person. However, implementing a

site based around customization requires a way to store the changes and modifications made by a user. If these changes disappear for later visits to the site, the site would not truly be customizable.

From this, it was clear that a secure and effective backend service was required. We decided to use ASP.NET with C# for our backend since, as discussed earlier in section 3.1 of the paper, Endicia used this for Sputnik. Because our site would be making calls to add to, delete from, and modify tables in our database (such as the favorites list), we also needed methods for exchanging data between the front end and the backend. For this we decided to create an Application Programming Interface (API) using a RESTful architecture (Fielding & Taylor, 2000).

3.6.1 ReSTful APIs

An API (Application Programming Interface) is, put simply, a collection of code that allows two applications to communicate with one another. In our case, we wrote a central API for the backend that allowed any client computer to modify the database depending on what action the user has taken. Because every client uses the same API, any necessary database changes can all be made in one place. This follows the "Don't Repeat Yourself" (DRY) principle of software development. The clients could then be seen as separate, lightweight UI layers. Figure 33 is a graphical explanation of the role that API plays in a system.

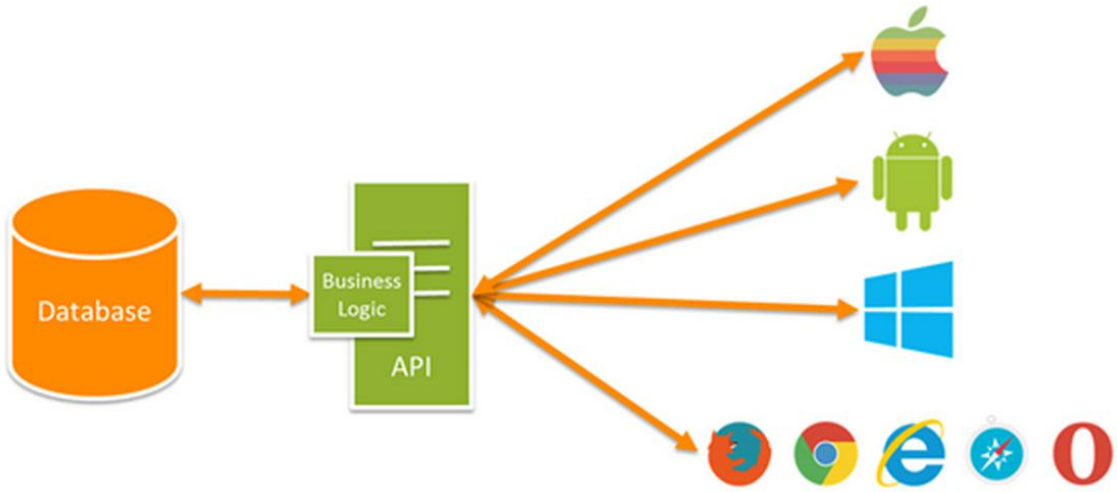


Figure 33 Using API as part of the framework.

ReST stands for “Representational State Transfer”, and is the most popular architectural pattern for creating an API. In a ReST architecture, every resource in the server is accessed by a Uniform Resource Identifier (URI). This meaning entities and collections have their unique representations, and there are no verbs/actions associated with the URL. A URL can only represent the state of an entity. For example, /Customers/John represents the state of John, a customer. This allows the HTTP verbs GET, POST, PUT, and DELETE to be used on this URL to perform CRUD (Create, Read, Update, and Delete) operations.

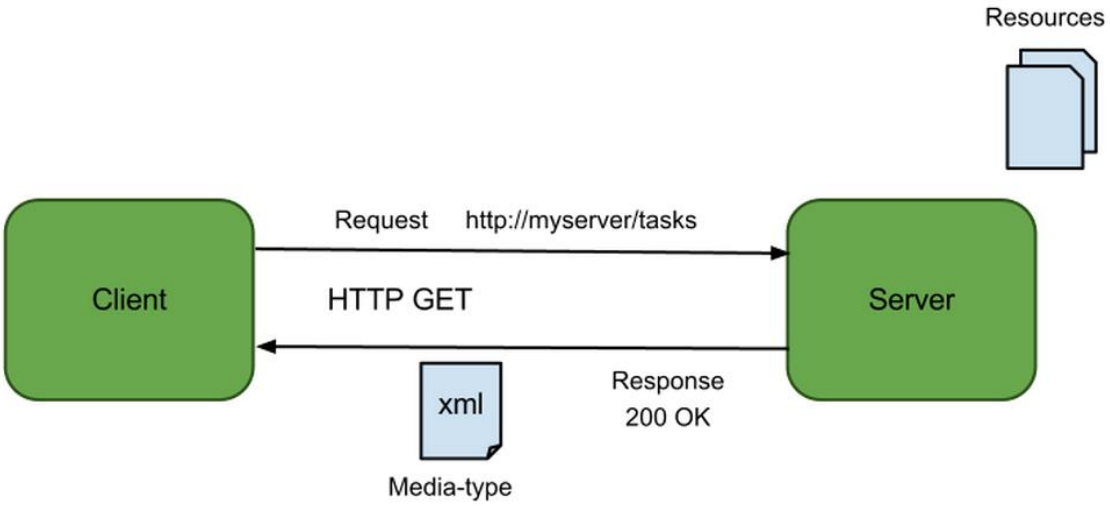


Figure 34 ReST Architecture

Figure 34 shows how a ReSTful architecture would work between a client and a server. The client sends an HTTP GET request to the server with the URI “http://myserver/tasks” which here is an XML file. The server receives the request and responds to the client with both the requested resource and the status code 200, indicating success.

Below are the API methods we created with .NET’s Web API package. Table 4 details each API method with the parameters to be passed in as well as the data to be returned.

API Name	API Description	Parameter(s)
DefaultListingController	Get a list of default links	Link category and user department
FavoritesController	Get a list of favorite links	Username
FavoritesEditController	Edit a list of links	Username
FavoritesAddController	Create a list of links	Username
FavoritesDeleteController	Delete a list of links	Username, list id

Table 4 API method names, descriptions, and parameters

3.6.2 Database Implementation

Given the fact that we only had seven weeks to complete our project, we decided to build the database in a separate environment from Endicia's instead of integrating our work into theirs from the start. Nevertheless, we knew the code would be ported over to their environment in the future by Endicia employees. To make this porting process easier, we implemented the database using a Repository Pattern.

A Repository Pattern separates data access logic and maps it to entities in the business logic. With this, it works with domain entities to perform data access. Put simply, domain entities, the data access logic and the business logic use interfaces to communicate. The benefit of using a Repository Pattern is that it hides the details of data access from the business logic. Business logic can access the data object without any knowledge of the underlying data access architecture. This is particularly useful in our case since it allows underlying data sources and architecture to be changed without affecting the business logic. Figure 35 visualizes this pattern.

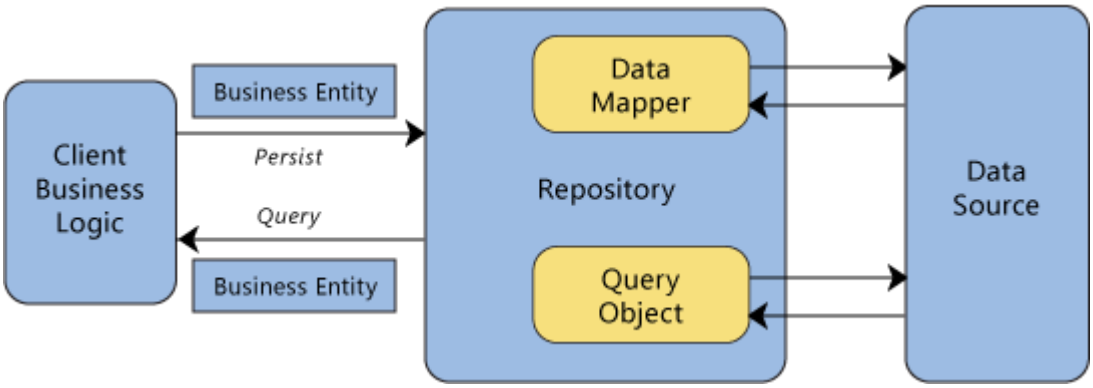


Figure 35. Repository Pattern

4 Results

During our time at Endicia, we interviewed users to find out their uses for Supervisory, analyzed log data tracking the clicks they were making, classified pages on Supervisory to find repetitive patterns that could be used as templates for future migration and development, built template pages to ease this migration process, and redesigned the Sputnik home page. Through all of this, we were getting constant feedback from numerous employees at Endicia from multiple departments, all of which was very constructive and often positive.

Towards the second half of our project work, we began periodically presenting our design to company executives, Supervisory users, and developers, many of whom found our work to be thorough and very positive. Through revisions and redesigns, we were able to create a set of quality deliverables that will hopefully be useful to developers and Endicia employees alike.

We see numerous benefits that will come from our project. The main benefit we see is for Endicia developers. We greatly reduced design confusion by creating a complete set of page templates and style guides for Sputnik, the new administrative site at Endicia. This included deciding the colors, fonts, layouts, etc. of each element to be used by developers.

We then designed a Developer Hub website to house all of our templates and display pseudo code to be easily copied and pasted for use. This will greatly speed up the development process as many of the developers at Endicia specialize in backend development. Instead of spending copious amounts of time writing HTML and CSS, these developers will now have a quick and easy way to get a page shell up and running. This will allow them to focus on the backend instead of the front.

All of these deliverables will ensure a more consistent look and feel for future web pages. Furthermore, our set of components could be abstracted in the future and used on customer facing websites. This was suggested by a number of Endicia executives and developers.

Another very important deliverable was our redesign of the Sputnik homepage. Coming in to Endicia, the homepage was nothing but a top nav bar and a welcome title. With more and more pages added to Sputnik each month, we needed to design a homepage that would allow scalability for the site and provide ease of navigation for users.

To do this, we looked back to Supervisory. We studied it and found what we did and did not like about its structure. We then had discussions with Endicia employees that used it on a daily basis to find out what they would improve. Next, we analyzed log data to find out how users from different departments were using it. Finally, we sent a survey out to employees to gauge the importance of links from a personal standpoint rather than through data.

From all of this research, we gleaned that customizability would be extremely important. We decided that there would need to be a favorites bar that would appear on every page and offer employees the chance to compile a list of easily accessible and personally important links. We also knew there would need to be a navigation bar on every page to provide links to different functionalities and pages. The final homepage had both of these elements as well as a very simple look and feel for the main content.

Our final design provides a simple, customizable homepage as well as an easily included shell for future Sputnik pages. This homepage sets the tone for the rest of the page design wise and can also be used as an example to get employees on board with the new look and feel.

5 Future Work

5.1 Migration to Sputnik

As is with any new technology, it's a challenge to transition from the old way of doing things to the new. Supervisory is widely used at Endicia, and employees have become acclimated to it despite its quirks and shortcomings. There will need to be a large push in the near future to shift over to using Sputnik. While this shift will surely take time, there are good ways and bad ways to promote it.

We believe the best approach would be to embed Supervisory pages into Sputnik by using iframes. This way, users would have the familiarity of Supervisory and at the same time become familiar with Sputnik. Currently, there are a large number of Supervisory pages that have not been ported Sputnik. All such pages would be iframed in from Supervisory. On the other hand, any pages that have already been ported would be included as is. Users would be able to enjoy the benefit of the favorites menu and the smart search bar before all of the Sputnik pages are even finished. This approach is much more favorable over a flag day in which users are forced to begin using Sputnik without any prior familiarity.

5.2 Dashboards

A nice feature to have for future development on Sputnik would be the addition of a dashboard to the user's home screen. The idea of dashboard was brought up multiple times by employees from different departments. However, to make a dashboard actually useful, customization is still the key. The user should be able to choose what and how information is displayed on their dashboard. Otherwise, any data put into the dashboard would only be meaningful to a small subset of people.

During our project, we realized there would not be enough time for us to implement customizable dashboards. Without a deep knowledge of the Supervisory reports and an understanding of how they were used, we would have had no way of knowing what would and wouldn't have been useful in a dashboard. That said, with more time dashboards are a very real possibility for Sputnik, and with so many Endicia developers backing the idea, we believe they'll eventually be implemented.

Works Cited

About Bootstrap. (n.d.). Retrieved from Get Bootstrap: http://getbootstrap.com/about/

Adobe. (n.d.). Retrieved from Adobe ColdFusion 11 Family: http://www.adobe.com/products/coldfusion-family.html

AngularJS -Superheroic Javascript MVW Framework. (n.d.). Retrieved from AngularJS: https://angularjs.org/

AngularJS: Developer Guide: Data Binding. (n.d.). Retrieved from AgularJS: https://docs.angularjs.org/guide/databinding

ASP.NET MVC - Basic overview of different view engines. (2012, October 1). Retrieved from Code Project: http://www.codeproject.com/Articles/467850/ASP-NET-MVC-view-engines

ASP.NET MVC 3. (n.d.). Retrieved from The Official Microsoft ASP.NET Site: http://www.asp.net/mvc/videos/mvc-3

awwwards-team. (2013, February 20). *What are Frameworks*. Retrieved from Awwwards: http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html

Bootstrap. (n.d.). *About Bootstrap*. Retrieved from Get Bootstrap: http://getbootstrap.com/about/

Bootstrap. (n.d.). *Bootstrap - The world's most popular mobile-first and responsive front-end framework*. Retrieved from Get Bootstrap: http://getbootstrap.com/

Chapman, C. (2010, October 18). *Information Architecture 101: Techniques and Best Practices*. Retrieved from Six Revisions: http://sixrevisions.com/usabilityaccessibility/information-architecture-101-techniques-and-best-practices/

Cisco. (2015). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019.* Cisco.

Fielding, R. T., & Taylor, R. N. (2000). Principled design of the modern Web architecture. New York, NY, USA. doi:337180.337228

Hunt, B. (2006, December 3). *Web Site Information Architecture Models*. Retrieved from Web Design from Scratch: http://webdesignfromscratch.com/website-architecture/ia-models/

Information Architecture Basics. (n.d.). Retrieved from usability.gov: http://www.usability.gov/what-and-why/information-architecture.html

Introducing "Razor"--a New View Engine for ASP.NET. (2010, July 3). Retrieved from ScottGu's Blog: http://weblogs.asp.net/scottgu/introducing-razor

Rahman, S. F. (2014). *Jump Start Bootstrap.* SitePoint.

Tutorial Republic. (n.d.). *Working with Twitter Bootstrap 3 Responsive Layout*. Retrieved from Tutorial Republic: http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-responsive-layout.php

Walton, P. (2013, August 17). *Decoupling Your HTML, CSS, and JavaScript*. Retrieved from PhilipWalton.com.

Wood, L., Le Hegaret, P., & Robie, J. (2000, November 13). *What is the Document Object Model?* Retrieved from W3.org: http://www.w3.org/TR/DOM-Level-2-Core/introduction.html

Zing Design Blog. (2013, June 18). *5 reasons NOT to use Twitter Bootstrap*. Retrieved from Zing Design Blog: http://www.zingdesign.com/5-reasons-not-to-use-twitter-bootstrap/

Appendix I: User Interview Notes

Jonathan Jecker

General

Need back to main menu button

Account Management

- Track users and what they do
    - Few different sections
    - Most important is edit account information (search for acct by any criteria)
        - Gives account status', such as past fraud
        - Clicking Profile will give account details
    - Account details view
        - It's really important to see how they signed up, who referred them, and importance of dedication
        - Postage purchases help determine importance of accounts
    - Editing accounts
        - Huge number of editing options
        - Heavily used option is "Account Features"
        - "ACH" means circumventing credit card fields and fees
        - Speedo: a call to an ELS server
        - Some elements that don't even need to be there

Administrator

- Client Release Manager
    - Latest and greatest versions and updates of software
    - Most are optional, while some are required due to postage price changes
    - Release list
        - Large list of versions of DAZzle
        - Comes with update message
        - Release percentage of 100% that means all accounts have to update, 0% means optional, in between sends out a message with YES or NO option for updateing
- Client Account Version Limits
    - Send out a message/force update for specific set of accounts

Reports

- Heavily used is software version

    - "Between this date and this date, how many times has a version of a software been opened"
- Speedo
    - Type of call to ELS, a lot of people on one system, trying to port them to an ELS system called Speedo
    - Sadly doesn't mean much to anyone else
    - Speedo Accounts Report
- It's very important to have this configurable for each person in the case that we do want a dashboard view

PIP

- Picture In Postage
- Printing of customized stamps
- Allows tracking of orders, shipments, partners with large orders
- Very important to those who deal with it, but invisible to a large portion of the company

Software Overview

Plans

- No Monthly Fee
    - Largest portion of customers
    - DYMO Stamps/DYMO Stamps online
    - Has a desktop version
    - Free to use, but stamps are paid for
- Standard
    - DAZzle
    - Workhorse, printing shipping labels
    - Main piece of software, used by larger warehouses
- Endicia Professional
    - High volume shipping focus
    - Order management systems
    - Automated processes
    - Most sales guys will send large shippers straight to this
    - Either uses DAZzle or ELS in the background
- Platinum Shippers
    - Run alongside FedEx or UPS
    - Scans information and reads them off the page; compares rates to see if things can be printed cheaper through other options
- Label Server (ELS)
    - APIs are written for this that call ELS and print labels (XML calls)

Groups in the Company

- Marketing
    - Regular marketing
        - Eileen Mc…
        - Lauren Vance
    - Product marketing
        - Jessica Foth
    - Product management
- Business development
    - Partner management
        - Build partner solutions
        - Wing
        - Ben Oken
        - Laurie Goldman
    - Consolidators
        - Anthony Roy
        - Rich Garcia
- Tech support
    - Melinda Hughes
    - Phillip
- Sales
    - Ron Hui
        - Relatively new, inside sales
    - Nancy Friedland
    - Paul Fredericks
        - Important, could be first point of contact
    - Find more people, sales uses this heavily
    - Email coming
- Devs
    - Patrick
    - Emil
- QA
    - Not as important
    - Alan Glendinning

Paul Fredericks

Uses

- Edit Account Info
- Manage Bundle Codes
    - When clients purchase rolls, bundles allow discount codes to be applied to their account
- Business Review
    - Managed Accounts
    - Consolidator's Transaction Report
    - Pay on use business report
- Commercial Plus Pricing Report
    - Shows when account was last touched to identify the contracts and discounts for each account
- CPP Discount Manager
    - Checkboxes based on the types of discounts provided
    - Sensitive information, maybe have protection against mistakes
    - Contract overview is important
    - Bulk Flagger is very useful
    - CPP Reports link to previous report
    - Upload GEPS and Upload NSA both important
- Multi-rate Service Volume Report
- Revenue Potential, almost the same as Multi-rate Service Volume Report
    - Submit to Road Runner Edition should not be there, doesn't work
    - Running from 0 to 0 range crashes the program
    - Run for one that started this year would be a cool tool
- USPS Rep Editor
    - Designed tool/portal for USPS sales reps
- Account Transfer
- USPS Rep Service
- George Hill
- Report reoccurrences

Ideas

Have an option where a report can be run automatically at a set recurring time. This way, when he comes in the morning, all processing will be complete and work can get done right away.

Philip Loesch

Uses

- Everything under Account Maintenance
- USPS Reps Reports
- Commercial Plus Pricing Report
- CPP Discount Manager
- Edit Account Information
    - The most important link for Tech Support

Suggestions

- Have navigation at top and bottom of page
- Left navigation should be consistent
- List of favorite/frequently used links
- Challenge Q&A should be easily seen when a customer would like their account reset
- Choose which page opens in edit account
- Add a change password button for supervisory account so it's easier
- In account advanced search, be able to filter out based on account status (i.e. don't show closed)
- Allow account info modification in account profile for:
    - Name
    - Company
    - Discounts (link to CPP discount manager report)
    - Paid through
- Add company name in contact info and physical address
- Add email preferences as part of contact info
- Turn off email notification for mailing status
- Allow adding multiple emails to an account
- Disable links to which access is denied
- Add referred by to more than just the advanced account search results

Appendix II: Supervisory Usage Survey

Supervisory Usage Survey

Hi! We are working on a successor to Supervisory. We have an intern team from WPI working on the new design, together with Patrick, Long, Emil and Nandita and assisted by Jon J.

To help us really make it really work we need some help and suggestions. Please take a minute to fill out the survey and give us your thoughts.

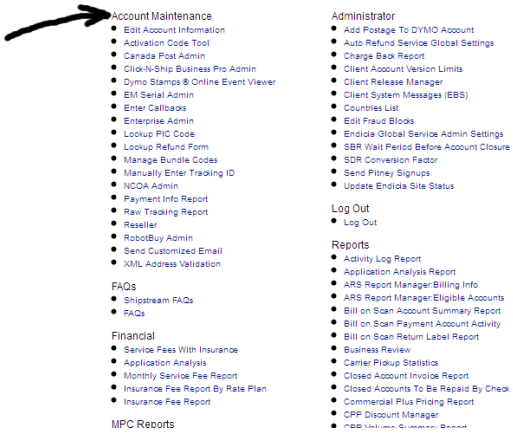
Also, we are holding a campaign for a new site name. If you want to enter, click here: nandita.nadig@endicia.com. Prize is $25 Starbucks card.

WPI Intern team and Patrick.

* Required

1. **Which department do you work for?** *
   Mark only one oval.

   - ◯ Technical Support
   - ◯ Sales
   - ◯ Business Development
   - ◯ Finance
   - ◯ Marketing
   - ◯ Research & Development
   - ◯ IT Department



2. **Which categories of links do you use on the supervisory main page? (Check all that apply)** *
   Please refer to the picture above.
   Check all that apply.

   - [ ] Account Maintenance
   - [ ] Reports
   - [ ] USPS Reps Reports
   - [ ] Referral Program
   - [ ] Financial
   - [ ] Operations
   - [ ] Dealer Program
   - [ ] Crypto
   - [ ] MPC Reports
   - [ ] Printers
   - [ ] FAQs

3. **What applications would you suggest us to take a look as the inspiration for Sputnik?**
   If you select "other", please provide the link to the application.
   Mark only one oval.

   - ◯ Jira
   - ◯ Salesforce.com
   - ◯ Other: .....

Supervisory Usage Survey - Account Maintenance & USPS Reps Reports

If you use any links in the Account Maintenance column or in the USPS Reps Reports column, please answer the following questions. (Optional)

4. **In the Account Maintenance column, what links would you like to see?**

   Select up to 10.
   Check all that apply.

   - [ ] Edit Account Information
   - [ ] Activation Code Tool
   - [ ] Canada Post Admin
   - [ ] Click-N-Ship Business Pro Admin
   - [ ] Dymo Stamps ® Online Event Viewer
   - [ ] Enterprise Admin
   - [ ] EM Serial Admin
   - [ ] Lookup Refund Form
   - [ ] Manage Bundle Codes
   - [ ] NCOA Admin
   - [ ] Payment Info Report
   - [ ] Raw Tracking Report
   - [ ] Reseller
   - [ ] RobotBuy Admin
   - [ ] Send Customized Email
   - [ ] XML Address Validation

5. **Below is a list of links we identified as not commonly used, and we are planning to exclude them from Sputnik. If you would like to see a specific one, please check the relevant checkbox and explain how you plan to use it.**

   Check all that apply.

   - [ ] Enter Callbacks
   - [ ] Lookup PIC Code
   - [ ] Manually Enter Tracking ID

6. **(Following the previous question) How do you plan to use the links?**

   .....

   .....

   .....

   .....

   .....

7. **In the USPS Reps Reports column, what links would you like to see?**
   Check all that apply.

   - [ ] USPS Account Transfer
   - [ ] USPS Rep Report
   - [ ] USPS Rep Editor
   - [ ] USPS Rep Search

8. **We found that the USPS Reps Table Editor link is not commonly used, and we are planning to exclude it from Sputnik. Please let us know if you would like to keep it and explain how you plan to use it.**

Supervisory Usage Survey - Reports

If you use any links in the Reports group, please answer the following two questions. (Optional)

9. **In the Reports column, what links would you like to see?**
   Select up to 15.
   Check all that apply.

   - [ ] Activity Log Report
   - [ ] ARS Report Manager:Billing Info
   - [ ] Bill on Scan Account Summary Report
   - [ ] Bill on Scan Payment Account Activity
   - [ ] Bill on Scan Return Label Report
   - [ ] Business Review
   - [ ] Commercial Plus Pricing Report
   - [ ] CPP Discount Manager
   - [ ] CPP Volume Summary Report
   - [ ] DYMOStamps User Data
   - [ ] DYMOStamps User Report
   - [ ] EMS Credit Card Transactions
   - [ ] Endicia All Accounts Report
   - [ ] Fraud Report
   - [ ] Insurance Administrator
   - [ ] Insurance Claims Report
   - [ ] Insurance Details Report
   - [ ] Label Server Fee Billing Report

- [ ] Label Server IP Override
- [ ] Mailer ID Summary Report
- [ ] Managed Account Management
- [ ] Master Return Label Account Report
- [ ] Multi Rate Service Volume Report
- [ ] New Account Label Kits
- [ ] New Applications Today
- [ ] Partner RevShare Commissions report
- [ ] Pay-on-Use Labels report
- [ ] Process LabelServer Billing
- [ ] Raw Data Reports
- [ ] Revenue Potential
- [ ] SCAN Form Limit Override
- [ ] Software Version
- [ ] Store Report
- [ ] Top Mail Classes By Month
- [ ] Transaction Count
- [ ] USPS Sales Report

10. **Below is a list of reports we identified as not commonly used, and we are planning to exclude them from Sputnik. If you would like to see a specific one, please check the relevant checkbox and explain how you plan to use it.**
    Check all that apply.

- [ ] Application Analysis Report
- [ ] ARS Report Manager:Eligible Accounts
- [ ] Carrier Pickup Statistics
- [ ] Closed Account Invoice Report
- [ ] Closed Accounts To Be Repaid By Check
- [ ] Daily Postage Purchase Report
- [ ] Daily Store Order Processing
- [ ] Delivery Statistics
- [ ] DSO CM Transactions
- [ ] Duplicate Refund Audit
- [ ] DYMO Inactive Account Data Dump
- [ ] DYMO Marketing Report
- [ ] ELS Email Blast List
- [ ] ELS Support Request
- [ ] Insurance Fees By Rate Plan

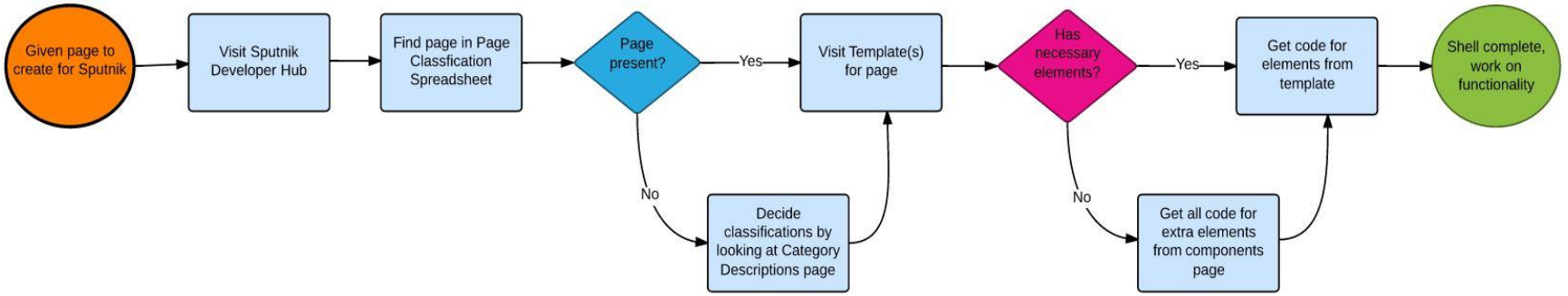
- [ ] International Expedited Accounts Report
- [ ] Invalid Closed Account Report
- [ ] Label Server Email Report
- [ ] LabelServer Active Users Report
- [ ] Mailer ID Volume by george
- [ ] Mailing List Export - Opt-Ins Only
- [ ] New Refund System Summary
- [ ] New Refund System Update
- [ ] New Suspended Accounts Today
- [ ] Operating System Version
- [ ] Partner Report
- [ ] Pitney Bowes Closed Accts to Refund by Check
- [ ] Planet Code Users
- [ ] PlatinumShipper Report
- [ ] Premium User Report
- [ ] Process Invoices
- [ ] Quick Print Survey
- [ ] Reconciliation Report
- [ ] Refund Data Report
- [ ] Refund Electronic Audit
- [ ] Refund PCP-X Form Generator
- [ ] Resource Software Billing Report
- [ ] SCAN Statistics
- [ ] Service Fees Bank Reconciliation
- [ ] Shared PES Exception Report
- [ ] Speedo Accounts
- [ ] Subscription Report
- [ ] Survey Tool Report
- [ ] Suspended Accounts Report
- [ ] Top Users By Mail Class
- [ ] Transaction Count Beta Report
- [ ] Verify Locked Account Report
- [ ] Yesterdays Suspensions
- [ ] ZIP Code Revenue Report

Supervisory Usage Survey

11. **(Following the previous question) How do you plan to use the reports?**

12. **Do you have any suggestions for the website in general?**

Appendix III: Expected Workflow for Future Developers



Appendix IV: Default Favorite Lists by Department

Business Development

Account Maintenance

Link Ordering

Activation Code Tool

Reseller

Dymo Stamps ® Online Event Viewer

EM Serial Admin

XML Address Validation

Send Customized Email

Lookup Refund Form

Canada Post Admin

RobotBuy Admin

Raw Tracking Report

Manually Enter Tracking ID

Enterprise Admin

Lookup PIC Code

Hidden Links

Manage Bundle Codes

Enter callbacks

Click-N-Ship Business Pro Admin

Payment Info Report

NCOA Admin

Reports

Link Ordering

Revenue Potential

Multi Rate Service Volume Report

Software Version

Activity Log Report

EMS Credit Card Transactions

Store Report

Insurance Fees By Rate Plan

Process LabelServer Billing

ZIP Code Revenue Report

Insurance Administrator

Business Review

Delivery Statistics

DYMOStamps User Report

Daily Postage Purchase Report

DYMOStamps User Data

Endicia All Accounts Report

New Refund System Summary

International Expedited Accounts Report

Partner RevShare Commissions report

Mailer ID Volume by george

PlatinumShipper Report

Insurance Details Report

Insurance Claims Report

Top Mail Classes By Month

Label Server Email Report

Operating System Version

Mailer ID Summary Report

Bill on Scan Payment Account Activity

Top Users By Mail Class

Commercial Plus Pricing Report

CPP Volume Summary Report

Managed Account Management

Bill on Scan Return Label Report

LabelServer Active Users Report

Hidden Links

ARS Report Manager:Billing Info

Bill on Scan Account Summary Report

Fraud Report

Label Server Fee Billing Report

Label Server IP Override

New Account Label Kits

Raw Data Reports

SCAN Form Limit Override

Transaction Count

USPS Sales Report

ARS Report Manager:Eligible Accounts

Carrier Pickup Statistics

Closed Account Invoice Report

Closed Accounts To Be Repaid By Check

DYMO Inactive Account Data Dump

Invalid Closed Account Report

Planet Code Users

Process Invoices

Refund Data Report

Refund Electronic Audit

Resource Software Billing Report

SCAN Statistics

Service Fees Bank Reconciliation

Speedo Accounts

Transaction Count Beta Report

Application Analysis Report

CPP Discount Manager

Daily Store Order Processing

DSO CM Transactions

Duplicate Refund Audit

DYMO Marketing Report

ELS Email Blast List

ELS Support Request

Mailing List Export - Opt-Ins Only

Master Return Label Account Report

New Applications Today

New Refund System Update

New Suspended Accounts Today

Partner Report

Pay-on-Use Labels Reports

Pitney Bowes Closed Accts to Refund by Check

Premium User Report

Quick Print Survey

Reconciliation Report

Refund PCP-X Form Generator

Shared PES Exception Report

Subscription Report

Survey Tool Report

Suspended Accounts Report

Verify Locked Account Report

Yesterdays Suspensions

USPS Reps Reports

Link Ordering

USPS Rep Editor

USPS Rep Search

USPS Account Transfer

USPS Rep Report

Hidden Links

USPS Reps Table Editor

Finance

Account Maintenance

Link Ordering

Activation Code Tool

Send Customized Email

EM Serial Admin

Payment Info Report

Dymo Stamps ® Online Event Viewer

Hidden Links

Canada Post Admin

NCOA Admin

Raw Tracking Report

Click-N-Ship Business Pro Admin

Reseller

RobotBuy Admin

Lookup Refund Form

XML Address Validation

Manage Bundle Codes

Manually Enter Tracking ID

Enter Callbacks

Enterprise Admin

Lookup PIC Code

Reports

Link Ordering

EMS Credit Card Transactions

Commercial Plus Pricing Report

Business Review

Managed Account Management

Endicia All Accounts Report

Multi Rate Service Volume Report

Insurance Administrator

Process LabelServer Billing

DYMOStamps User Report

Top Users By Mail Class

New Refund System Summary

ARS Report Manager:Billing Info

Closed Accounts To Be Repaid By Check

Insurance Details Report

Service Fees Bank Reconciliation

Label Server Fee Billing Report

Hidden Links

Raw Data Reports

New Account Label Kits

Software Version

Daily Postage Purchase Report

Planet Code Users

Delivery Statistics

Revenue Potential

DYMO Inactive Account Data Dump

Carrier Pickup Statistics

DYMOStamps User Data

Operating System Version

Bill on Scan Account Summary Report

Process Invoices

Bill on Scan Payment Account Activity

Refund Electronic Audit

Bill on Scan Return Label Report

SCAN Statistics

Fraud Report

Store Report

Closed Account Invoice Report

Insurance Claims Report

Partner RevShare Commissions report

Insurance Fees By Rate Plan

PlatinumShipper Report

International Expedited Accounts Report

ARS Report Manager:Eligible Accounts

Invalid Closed Account Report

Refund Data Report

Top Mail Classes By Month

Resource Software Billing Report

Transaction Count

SCAN Form Limit Override

USPS Sales Report

Activity Log Report

Speedo Accounts

Mailer ID Summary Report

CPP Volume Summary Report

Mailer ID Volume by george

Label Server Email Report

Transaction Count Beta Report

ZIP Code Revenue Report

Label Server IP Override

LabelServer Active Users Report

Application Analysis Report

CPP Discount Manager

Daily Store Order Processing

DSO CM Transactions

Duplicate Refund Audit

DYMO Marketing Report

ELS Email Blast List

ELS Support Request

Mailing List Export - Opt-Ins Only

Master Return Label Account Report

New Applications Today

New Refund System Update

New Suspended Accounts Today

Partner Report

Pay-on-Use Labels Reports

Pitney Bowes Closed Accts to Refund by Check

Premium User Report

Quick Print Survey

Reconciliation Report

Refund PCP-X Form Generator

Shared PES Exception Report

Subscription Report

Survey Tool Report

Suspended Accounts Report

Verify Locked Account Report

Yesterdays Suspensions

USPS Reps Reports

Link Ordering

USPS Rep Editor

USPS Rep Report

USPS Account Transfer

USPS Rep Search

Hidden Links

USPS Account Transfer

Marketing

Account Maintenance

Link Ordering

Activation Code Tool

Reseller

EM Serial Admin

Send Customized Email

Lookup Refund Form

Raw Tracking Report

Payment Info Report

Canada Post Admin

Dymo Stamps ® Online Event Viewer

Manually Enter Tracking ID

Hidden Links

RobotBuy Admin

Click-N-Ship Business Pro Admin

NCOA Admin

XML Address Validation

Manage Bundle Codes

Enter Callbacks

Enterprise Admin

Lookup PIC Code

Reports

Link Ordering

Business Review

Commercial Plus Pricing Report

Revenue Potential

Multi Rate Service Volume Report

Raw Data Reports

Insurance Administrator

EMS Credit Card Transactions

Software Version

Label Server IP Override

Label Server Email Report

Operating System Version

LabelServer Active Users Report

Top Users By Mail Class

ARS Report Manager:Eligible Accounts

DYMOStamps User Data

Bill on Scan Account Summary Report

Endicia All Accounts Report

DYMOStamps User Report

Bill on Scan Payment Account Activity

Store Report

Insurance Details Report

Fraud Report

SCAN Statistics

Insurance Claims Report

New Refund System Summary

Insurance Fees By Rate Plan

PlatinumShipper Report

Bill on Scan Return Label Report

Process LabelServer Billing

Label Server Fee Billing Report

ARS Report Manager:Billing Info

Refund Data Report

Service Fees Bank Reconciliation

Activity Log Report

Top Mail Classes By Month

CPP Volume Summary Report

Hidden Links

Closed Accounts To Be Repaid By Check

ZIP Code Revenue Report

Daily Postage Purchase Report

Planet Code Users

Delivery Statistics

DYMO Inactive Account Data Dump

Closed Account Invoice Report

Process Invoices

Refund Electronic Audit

Transaction Count Beta Report

Partner RevShare Commissions report

International Expedited Accounts Report

Invalid Closed Account Report

Resource Software Billing Report

SCAN Form Limit Override

Speedo Accounts

Mailer ID Summary Report

Mailer ID Volume by george

Transaction Count

Managed Account Management

USPS Sales Report

Carrier Pickup Statistics

New Account Label Kits

Application Analysis Report

CPP Discount Manager

Daily Store Order Processing

DSO CM Transactions

Duplicate Refund Audit

DYMO Marketing Report

ELS Email Blast List

ELS Support Request

Mailing List Export - Opt-Ins Only

Master Return Label Account Report

New Applications Today

New Refund System Update

New Suspended Accounts Today

Partner Report

Pay-on-Use Labels Reports

Pitney Bowes Closed Accts to Refund by Check

Premium User Report

Quick Print Survey

Reconciliation Report

Refund PCP-X Form Generator

Shared PES Exception Report

Subscription Report

Survey Tool Report

Suspended Accounts Report

Verify Locked Account Report

Yesterdays Suspensions

USPS Reps Reports

Link Ordering

USPS Rep Search

USPS Rep Editor

USPS Account Transfer

Hidden Links

USPS Reps Table Editor

USPS Rep Report

Research & Development

Account Maintenance

Link Ordering

EM Serial Admin

Activation Code Tool

Click-N-Ship Business Pro Admin

Canada Post Admin

Send Customized Email

Raw Tracking Report

Reseller

RobotBuy Admin

Manually Enter Tracking ID

Hidden Links

NCOA Admin

Dymo Stamps ® Online Event Viewer

Payment Info Report

Lookup Refund Form

XML Address Validation

Manage Bundle Codes

Description

Enter Callbacks

Enterprise Admin

Lookup PIC Code

Reports

Link Ordering

Business Review

Software Version

Transaction Count

Insurance Administrator

ARS Report Manager:Billing Info

Multi Rate Service Volume Report

Commercial Plus Pricing Report

CPP Volume Summary Report

Refund Electronic Audit

DYMOStamps User Data

DYMOStamps User Report

Endicia All Accounts Report

Raw Data Reports

Revenue Potential

Speedo Accounts

Label Server Fee Billing Report

Hidden Links

New Refund System Summary

Bill on Scan Payment Account Activity

Bill on Scan Return Label Report

Mailer ID Volume by george

PlatinumShipper Report

Carrier Pickup Statistics

ARS Report Manager:Eligible Accounts

Closed Account Invoice Report

Activity Log Report

Closed Accounts To Be Repaid By Check

Partner RevShare Commissions report

Process LabelServer Billing

Daily Postage Purchase Report

SCAN Form Limit Override

Delivery Statistics

Top Mail Classes By Month

DYMO Inactive Account Data Dump

USPS Sales Report

Mailer ID Summary Report

Managed Account Management

EMS Credit Card Transactions

New Account Label Kits

Operating System Version

Fraud Report

Planet Code Users

Bill on Scan Account Summary Report

Process Invoices

Insurance Claims Report

Refund Data Report

Resource Software Billing Report

Insurance Details Report

Insurance Fees By Rate Plan

SCAN Statistics

Service Fees Bank Reconciliation

International Expedited Accounts Report

Store Report

Invalid Closed Account Report

Top Users By Mail Class

Label Server Email Report

Transaction Count Beta Report

ZIP Code Revenue Report

Label Server IP Override

LabelServer Active Users Report

Application Analysis Report

CPP Discount Manager

Daily Store Order Processing

DSO CM Transactions

Duplicate Refund Audit

DYMO Marketing Report

ELS Email Blast List

ELS Support Request

Mailing List Export - Opt-Ins Only

Master Return Label Account Report

New Applications Today

New Refund System Update

New Suspended Accounts Today

Partner Report

Pay-on-Use Labels Reports

Pitney Bowes Closed Accts to Refund by Check

Premium User Report

Quick Print Survey

Reconciliation Report

Refund PCP-X Form Generator

Shared PES Exception Report

Subscription Report

Survey Tool Report

Suspended Accounts Report

Verify Locked Account Report

Yesterdays Suspensions

USPS Reps Reports

Link Ordering

USPS Rep Search

Hidden Links

USPS Account Transfer

USPS Reps Table Editor

USPS Rep Editor

USPS Rep Report

IT Department

Account Maintenance

Link Ordering

Hidden Links

Reports

Link Ordering

Revenue Potential

Commercial Plus Pricing Report

Bill on Scan Account Summary Report

Bill on Scan Payment Account Activity

Bill on Scan Return Label Report

Closed Account Invoice Report

Closed Accounts To Be Repaid By Check

Raw Data Reports

Hidden Links

New Account Label Kits

Top Mail Classes By Month

ARS Report Manager:Eligible Accounts

Mailer ID Summary Report

Planet Code Users

Service Fees Bank Reconciliation

Business Review

USPS Sales Report

Carrier Pickup Statistics

Managed Account Management

Operating System Version

Process Invoices

Refund Electronic Audit

SCAN Form Limit Override

CPP Volume Summary Report

Speedo Accounts

Daily Postage Purchase Report

Transaction Count

Delivery Statistics

Activity Log Report

DYMO Inactive Account Data Dump

Mailer ID Volume by george

DYMOStamps User Data

Multi Rate Service Volume Report

DYMOStamps User Report

New Refund System Summary

EMS Credit Card Transactions

Partner RevShare Commissions report

Endicia All Accounts Report

PlatinumShipper Report

Fraud Report

Process LabelServer Billing

Insurance Administrator

Refund Data Report

Insurance Claims Report

ARS Report Manager:Billing Info

Resource Software Billing Report

Insurance Details Report

SCAN Statistics

Insurance Fees By Rate Plan

Software Version

International Expedited Accounts Report

Store Report

Invalid Closed Account Report

Top Users By Mail Class

Label Server Email Report

Transaction Count Beta Report

Label Server Fee Billing Report

ZIP Code Revenue Report

Label Server IP Override

LabelServer Active Users Report

Application Analysis Report

CPP Discount Manager

Daily Store Order Processing

DSO CM Transactions

Duplicate Refund Audit

DYMO Marketing Report

ELS Email Blast List

ELS Support Request

Mailing List Export - Opt-Ins Only

Master Return Label Account Report

New Applications Today

New Refund System Update

New Suspended Accounts Today

Partner Report

Pay-on-Use Labels Reports

Pitney Bowes Closed Accts to Refund by Check

Premium User Report

Quick Print Survey

Reconciliation Report

Refund PCP-X Form Generator

Shared PES Exception Report

Subscription Report

Survey Tool Report

Suspended Accounts Report

Verify Locked Account Report

Yesterdays Suspensions

USPS Reps Reports

Link Ordering

None

Hidden Links

USPS Rep Search

USPS Account Transfer

USPS Reps Table Editor

USPS Rep Editor

USPS Rep Report

Sales

Account Maintenance

Link Ordering

Reseller

Activation Code Tool

Lookup Refund Form

Raw Tracking Report

Send Customized Email

Canada Post Admin

EM Serial Admin

RobotBuy Admin

Dymo Stamps ® Online Event Viewer

Manage Bundle Codes

Payment Info Report

NCOA Admin

XML Address Validation

Manually Enter Tracking ID

Hidden Links

Click-N-Ship Business Pro Admin

Edit Account Information

Enter Callbacks

Enterprise Admin

Lookup PIC Code

Reports

Link Ordering

Business Review

Commercial Plus Pricing Report

Revenue Potential

ARS Report Manager:Billing Info

SCAN Form Limit Override

Multi Rate Service Volume Report

Store Report

Managed Account Management

Software Version

Insurance Administrator

Raw Data Reports

Bill on Scan Payment Account Activity

Process LabelServer Billing

DYMOStamps User Report

Bill on Scan Account Summary Report

Fraud Report

USPS Sales Report

EMS Credit Card Transactions

Activity Log Report

Insurance Details Report

CPP Volume Summary Report

Daily Postage Purchase Report

Top Mail Classes By Month

Speedo Accounts

ZIP Code Revenue Report

Delivery Statistics

Closed Account Invoice Report

Mailer ID Summary Report

Label Server Fee Billing Report

New Account Label Kits

Endicia All Accounts Report

DYMO Inactive Account Data Dump

Resource Software Billing Report

Service Fees Bank Reconciliation

Mailer ID Volume by george

ARS Report Manager:Eligible Accounts

Invalid Closed Account Report

Carrier Pickup Statistics

New Refund System Summary

Insurance Fees By Rate Plan

Operating System Version

International Expedited Accounts Report

Partner RevShare Commissions report

SCAN Statistics

Planet Code Users

Label Server Email Report

PlatinumShipper Report

DYMOStamps User Data

Top Users By Mail Class

Insurance Claims Report

Transaction Count Beta Report

Closed Accounts To Be Repaid By Check

Refund Data Report

Label Server IP Override

LabelServer Active Users ReportTransaction Count

Bill on Scan Return Label Report

Process Invoices

Refund Electronic Audit

Hidden Links

Resource Software Billing Report

Service Fees Bank Reconciliation

Mailer ID Volume by george

ARS Report Manager:Eligible Accounts

Invalid Closed Account Report

Carrier Pickup Statistics

New Refund System Summary

Insurance Fees By Rate Plan

Operating System Version

International Expedited Accounts Report

Partner RevShare Commissions report

SCAN Statistics

Planet Code Users

Label Server Email Report

PlatinumShipper Report

DYMOStamps User Data

Top Users By Mail Class

Insurance Claims Report

Transaction Count Beta Report

Closed Accounts To Be Repaid By Check

Refund Data Report

Label Server IP Override

LabelServer Active Users Report

USPS Reps Reports

Link Ordering

USPS Rep Search

USPS Rep Editor

USPS Rep Report

USPS Account Transfer

USPS Reps Table Editor

Hidden Links

None

Technical Support

Account Maintenance

Link Ordering

Lookup Refund Form

Send Customized Email

Payment Info Report

Activation Code Tool

Reseller

Raw Tracking Report

EM Serial Admin

Canada Post Admin

RobotBuy Admin

Dymo Stamps ® Online Event Viewer

NCOA Admin

XML Address Validation

Manage Bundle Codes

Click-N-Ship Business Pro Admin

Manually Enter Tracking ID

Hidden Links

None

Reports

Link Ordering

Commercial Plus Pricing Report

Revenue Potential

Business Review

Raw Data Reports

Multi Rate Service Volume Report

SCAN Form Limit Override

DYMOStamps User Report

ARS Report Manager:Billing Info

Store Report

Label Server Fee Billing Report

Insurance Claims Report

Insurance Administrator

Mailer ID Volume by george

Insurance Details Report

Process LabelServer Billing

Bill on Scan Return Label Report

Activity Log Report

CPP Volume Summary Report

Transaction Count

Closed Accounts To Be Repaid By Check

Refund Data Report

Bill on Scan Payment Account Activity

Insurance Fees By Rate Plan

SCAN Statistics

Managed Account Management

ARS Report Manager:Eligible Accounts

Partner RevShare Commissions report

EMS Credit Card Transactions

Mailer ID Summary Report

Carrier Pickup Statistics

Fraud Report

Invalid Closed Account Report

PlatinumShipper Report

Software Version

Resource Software Billing Report

Service Fees Bank Reconciliation

Closed Account Invoice Report

Top Mail Classes By Month

Endicia All Accounts Report

Operating System Version

DYMOStamps User Data

Transaction Count Beta Report

New Account Label Kits

Process Invoices

Planet Code Users

Hidden Links

Daily Postage Purchase Report

Delivery Statistics

New Refund System Summary

Speedo Accounts

Label Server Email Report

Top Users By Mail Class

International Expedited Accounts Report

Bill on Scan Account Summary Report

Refund Electronic Audit

USPS Sales Report

DYMO Inactive Account Data Dump

ZIP Code Revenue Report

Label Server IP Override

LabelServer Active Users Report

Application Analysis Report

CPP Discount Manager

Daily Store Order Processing

DSO CM Transactions

Duplicate Refund Audit

DYMO Marketing Report

ELS Email Blast List

ELS Support Request

Mailing List Export - Opt-Ins Only

Master Return Label Account Report

New Applications Today

New Refund System Update

New Suspended Accounts Today

Partner Report

Pay-on-Use Labels Reports

Pitney Bowes Closed Accts to Refund by Check

Premium User Report

Quick Print Survey

Reconciliation Report

Refund PCP-X Form Generator

Shared PES Exception Report

Subscription Report

Survey Tool Report

Suspended Accounts Report

Verify Locked Account Report

Yesterdays Suspensions

USPS Reps Reports

Link Ordering

USPS Rep Search

USPS Rep Editor

USPS Account Transfer

USPS Reps Table Editor

USPS Rep Report

Hidden Links

None