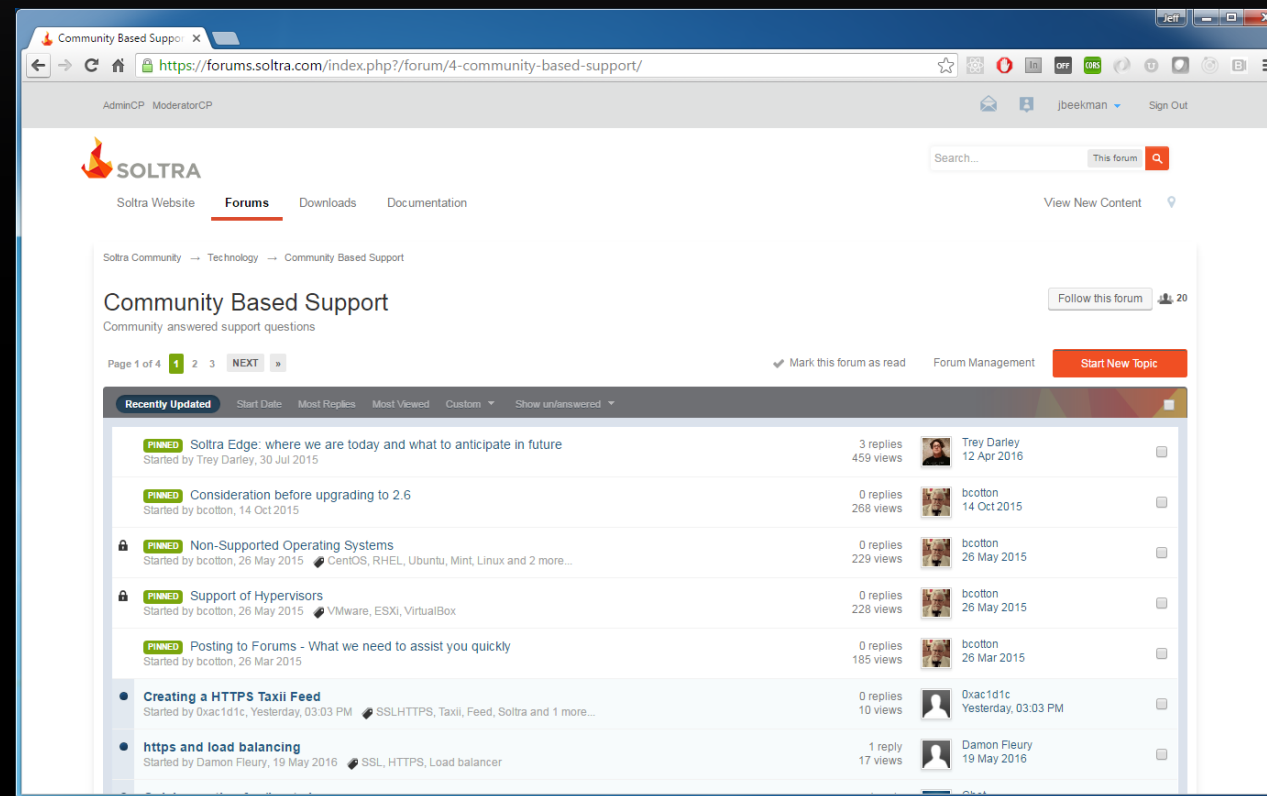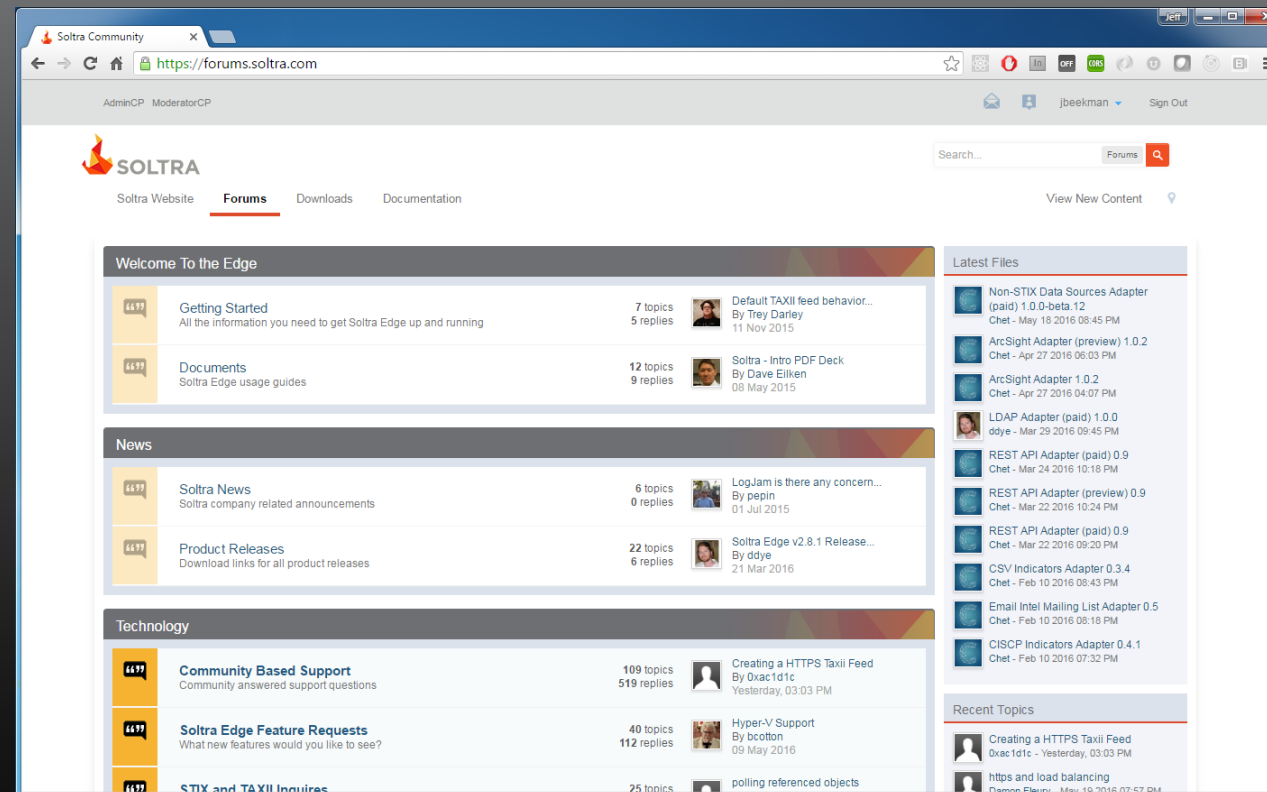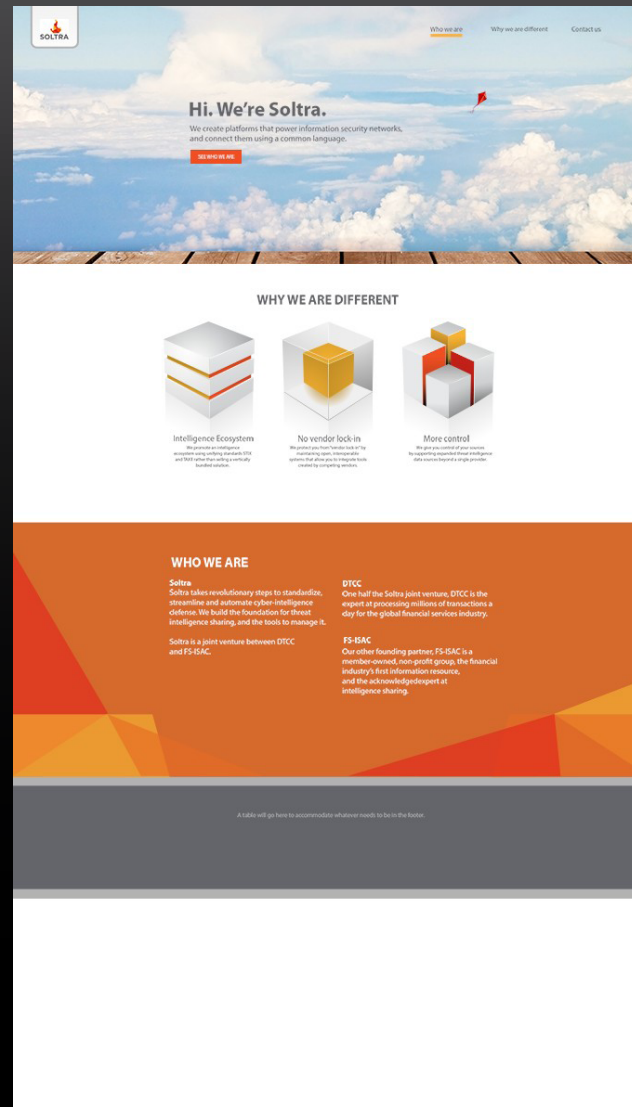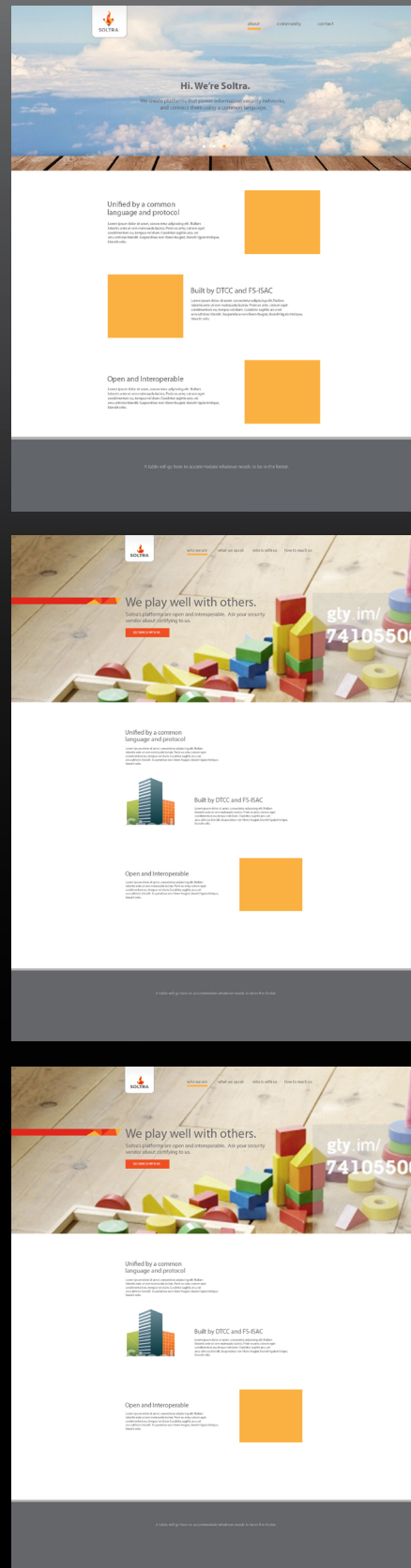# soltra.com Online Identity

brief

Created a branded collateral look and feel among the various Soltra online venues to include the soltra.com website, the support forums at http://forums.soltra.com and the developer wiki at http://wiki.soltra.com (special access required).

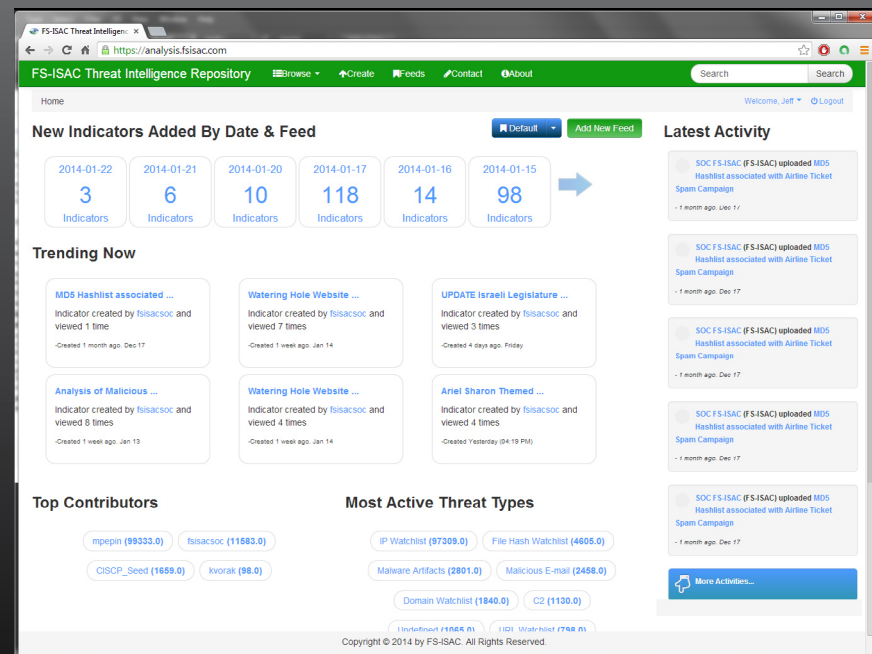Worked with DTCC UX Design Center of Excellence as I produced several mockups in accordance with corporate guidelines.
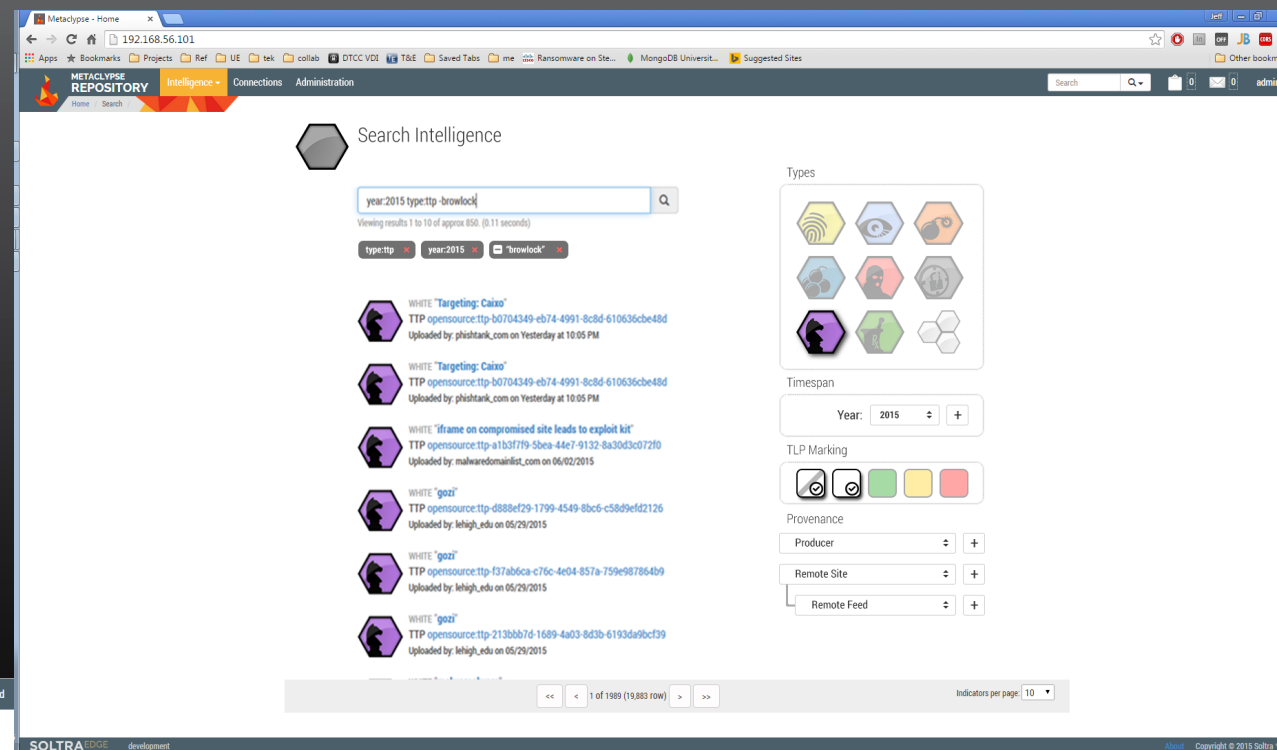
tools/tech

Bootstrap, PHP5, HTML/CSS3

platform

All

Soltra Edge v2.0

v1.x

brief

Edge is the flagship product of Soltra, a joint partnership funded by DTCC and the FS-ISAC, and is powered by STIX, an intelligence data format created at MITRE and used by DHS. When I arrived on the scene, the 1.x UI was the one created by the developers, so there was little forethought to navigational taxonomy and much refactoring was required on the Bootstrap/KnockoutJS application.

I redesigned the STIX icons, developed a strong navigational approach based upon card sorting exercise results, and implemented some data visualization of STIX constructs created using D3js.

The STIX graph/tree visualization utilized a Google Maps-like UX, with panning and zooming via scrollwheel, with expandable nodes, to facilitate viewing of extremely complex threat intelligence that potentially contained tens of thousands of connected nodes.
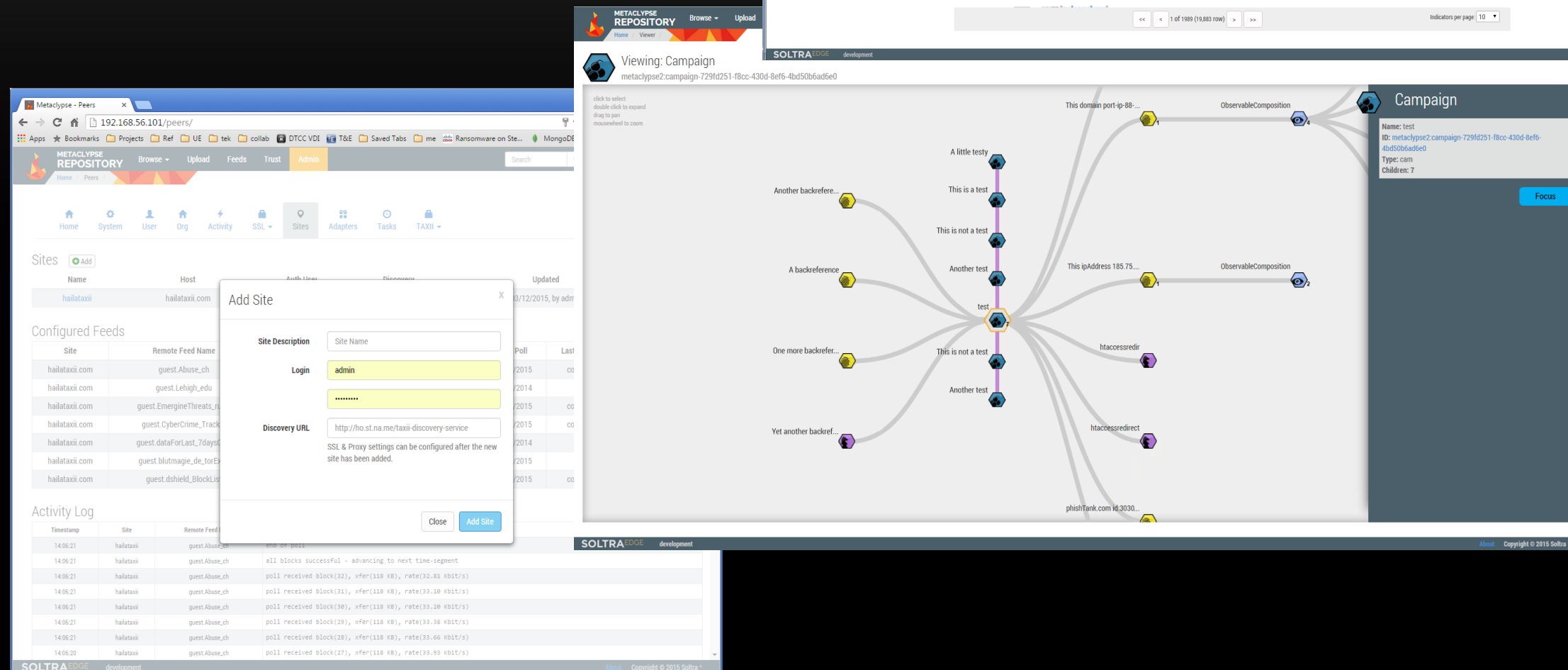
The next iteration of this interface was to make the graph full CRUD, with insertion and deletion of nodes, and modifying the data of selected nodes.

tools/tech

KnockoutJS, Bootstrap 3
D3, REST SOA

platform

IE8+

Soltra Edge v3.0

brief

The upcoming 3.0 release of Edge presented an opportunity to right the wrongs of previous versions.

I discovered that the vast majority of our users, analysts and engineers in security operations centers, had workstations with from 2 to 6 monitors attached, and unanimously wanted to utilize the full real estate of their expensive monitors. So I shifted the responsive breakpoints upward, supporting medium form factor devices (tablets) up to a custom XL form factor that would fill the screen where appropriate. The response was very positive.

I also drove a stack change to React/Redux and Elasticsearch in order to streamline development and support designed features geared toward big data.
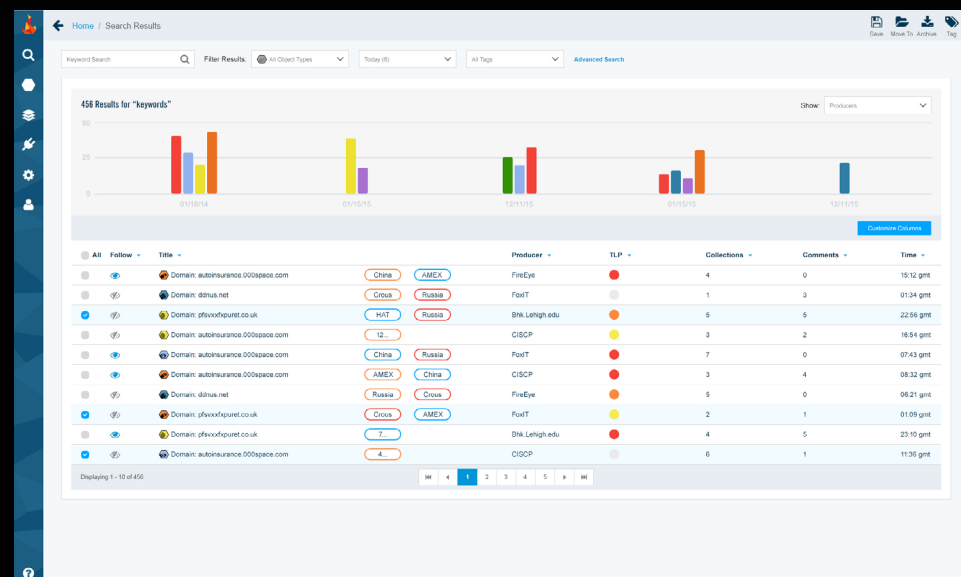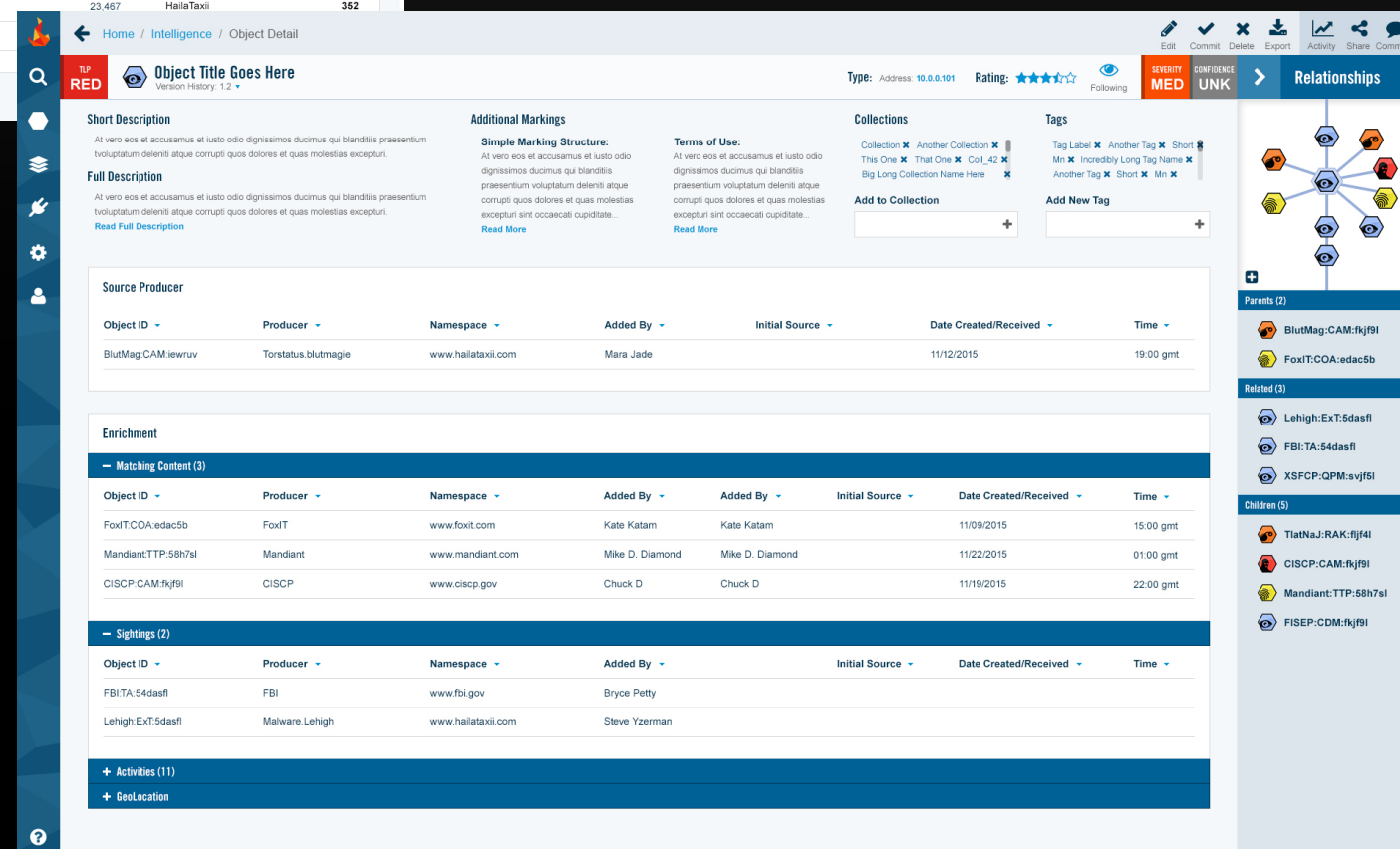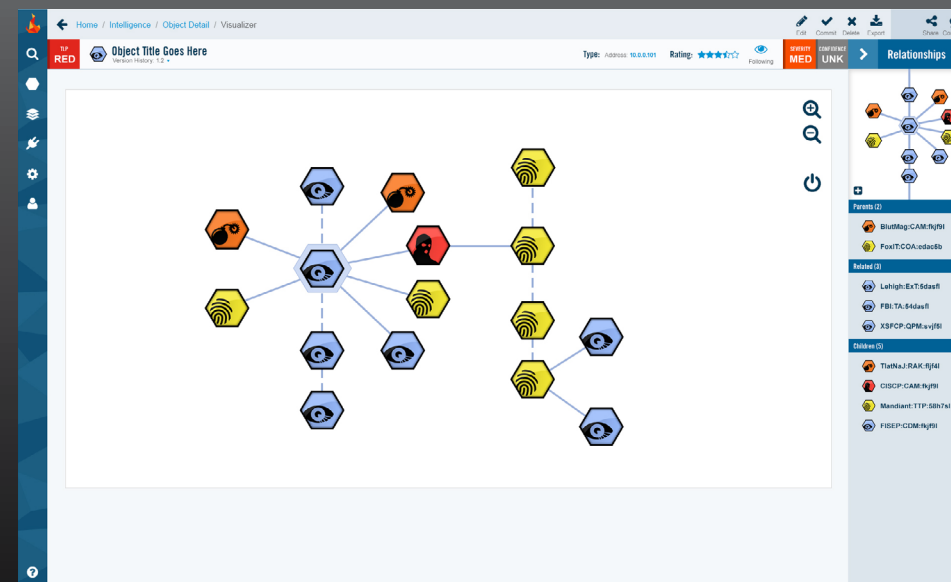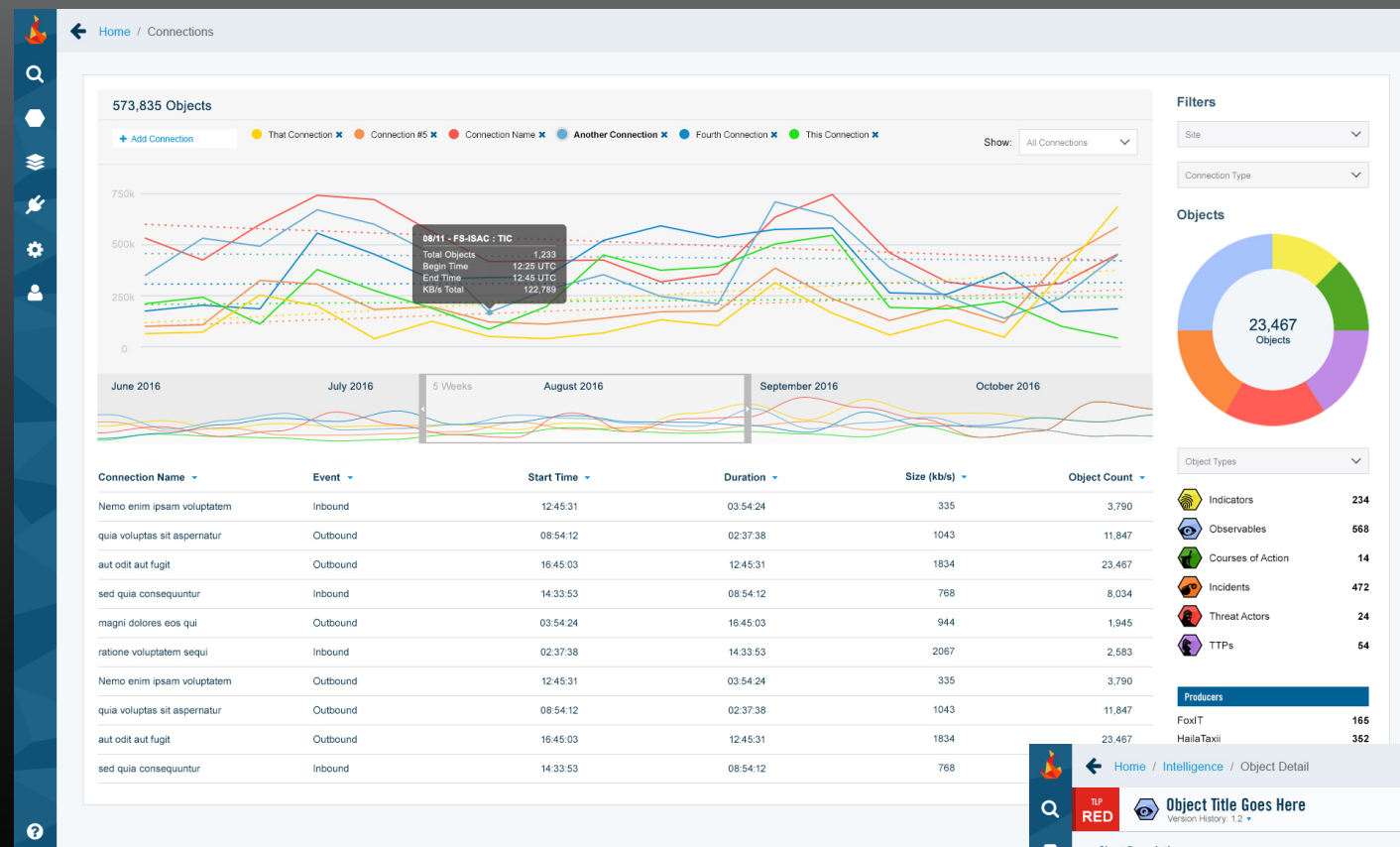
Managed engagements with Perficient XD and Keylines developers to create a full-screen responsive application with a drag and drop STIX graph builder, robust data visualizations and a clean, flat, usable design that our working group users love.

tools/tech

ReactJS, Redux, Elasticsearch, Keylines

platform

IE10+

Disney Passport

The WDW Passport interface is PHP5, HTML5, standards-compliant code, one of the first applications in the GBTS Composite UI to make that claim.

My involvement was in the pre-development phases, solely in providing look and feel, prototyping and usability services.

The challenge was to create an interface that was able to navigate hundreds of diverse and faceted workflows without resorting to the page-based navigation that is the legacy of Web application design.
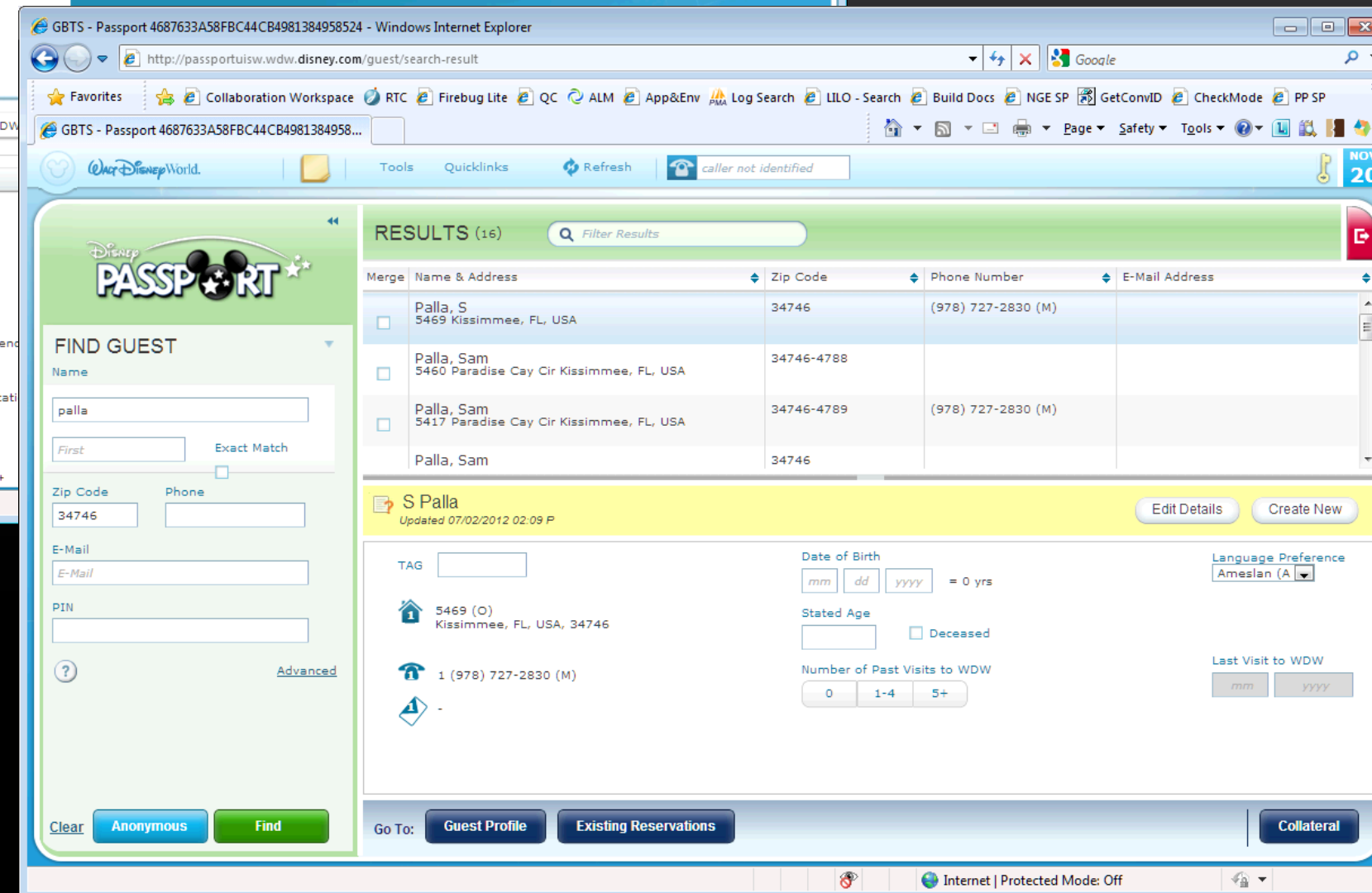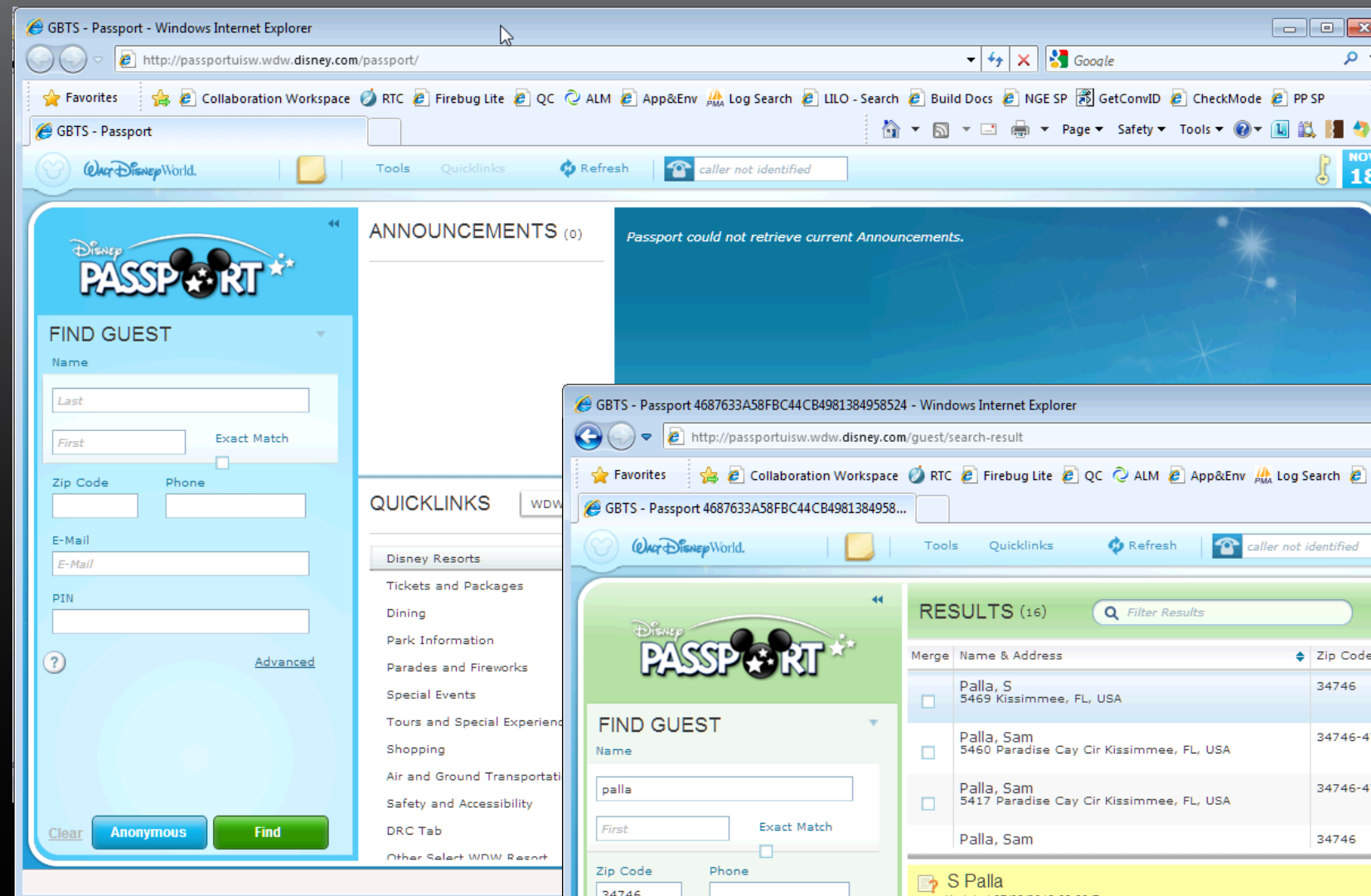
Consistently applying a "search > select > action" metaphor, we structured the application into three discrete regions for each step of that common workflow. The stowable search panel on the left was the entry point for every use case, and doubled as a context display for items being actioned. The upper section serves as resultset display, and the bottom portion for updating those records. This consistent master-list-detail workflow is a clear model that users can assimilate.
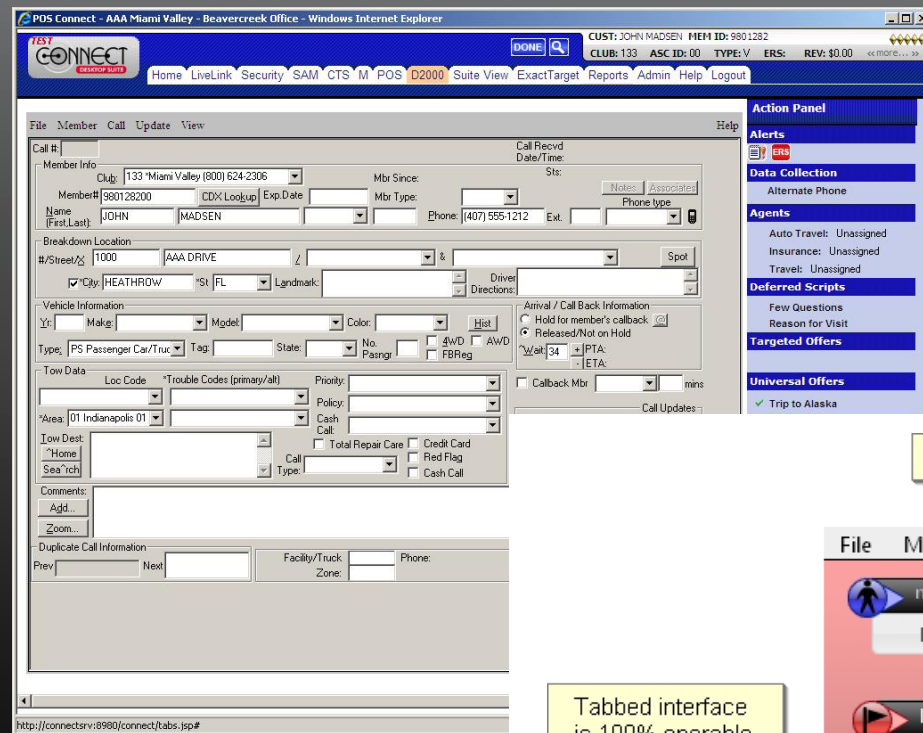
tools/tech

HTML5/CSS3
jQuery

platform

IE8 w/ Chrome Frame

AAA D3 CR

brief

The legacy D-2000 Application Suite is comprised of many applications supporting AAAs logistics, call-centers, vendors and fleets, and I was involved in the development of the next major version, D3, for several of these applications. One of them, Call Receiving, was a legacy C++ desktop application that the AAA operators used for handling incoming breakdown calls, from the first contact through the vehicle drop-off. As convoluted as this application was, the operators were experts at using it, so preserving that level of expert usability was key. The prototype I delivered illustrated mouse-free operation with quick keyboard access to menus, shortcuts and for field navigation, and a widget I invented called "rich tabs". A "rich tabs" is a tabset with more robust navigation, better information density, and data-enabled tabs with alerts/indicators, really geared for multi-part form data entry for advanced users, while remaining easy for users to create a mental model of complex information.

after

before

Member information is pared down to the essentials for common usage patterns

Tabbed interface is 100% operable by keyboard via PgUp/PgDn keys

Tabbing past the last field on a screen automatically advances to the next screen for advanced users' mouse-free operation

Any screen element can be expanded to fill the viewport for unconstrained operation

tools/tech

jQuery
HTML

platform

IE8

AAA D3 CR

Menus function exactly like desktop application-style menus --
click to open/close, full keyboard navigation using
cursor keys, and full support for keyboard shortcuts
and keystroke combinations

One-key access
to specific tabs is
also supported

Visible tab content shows pertinent information
with support for alert and status indicators, reducing the
need for switching tabs after data entry has been completed

The state progression and
current state of a call
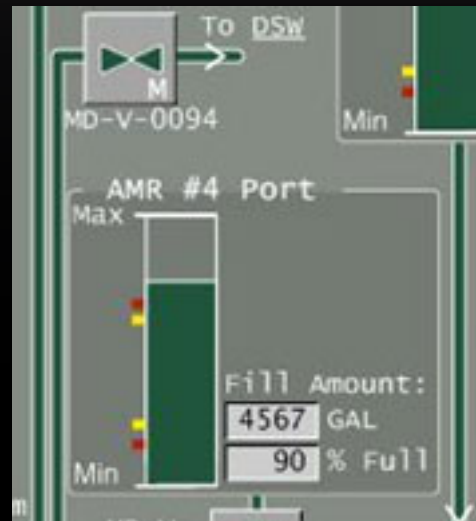is represented visually with
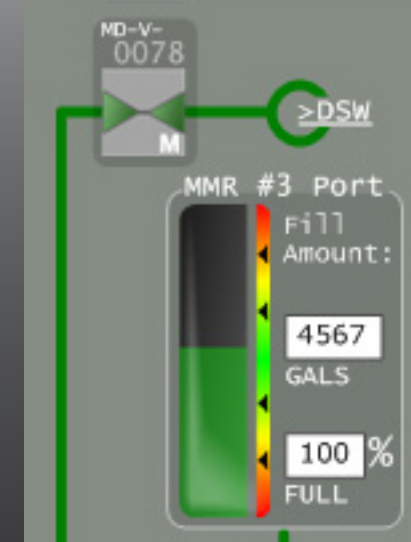the other call information

DDG-1000

after

before

brief

During the course of a 2-month usability analysis and UI cleanup of the ballast, fire and pump control touch-screen interface for the DDG-1000 Stealth Destroyer, I identified an opportunity for enhanced functionality using SVGs instead of the animated GIFs that were in place. With upwards of 100 animated GIFs on some screens, it was also a client performance problem. The nodes and elements of an SVG are programmatically addressable both at the DOM level and through libraries like Batik, so by using vector shapes, tank levels and on/off states of fans and multi-position valves could reflect real-time data with smooth performance.

It was a short effort, and the layouts needed to match schematics and bulkhead compartments that the technicians were already familiar with. In that time, though, usability was improved by reducing errors from confusing label positions, gross misalignment, distinction between control vs. display-only elements and more visible nominal vs. alert states. *Screens blurred for security reasons, actuals can be viewed in person.*
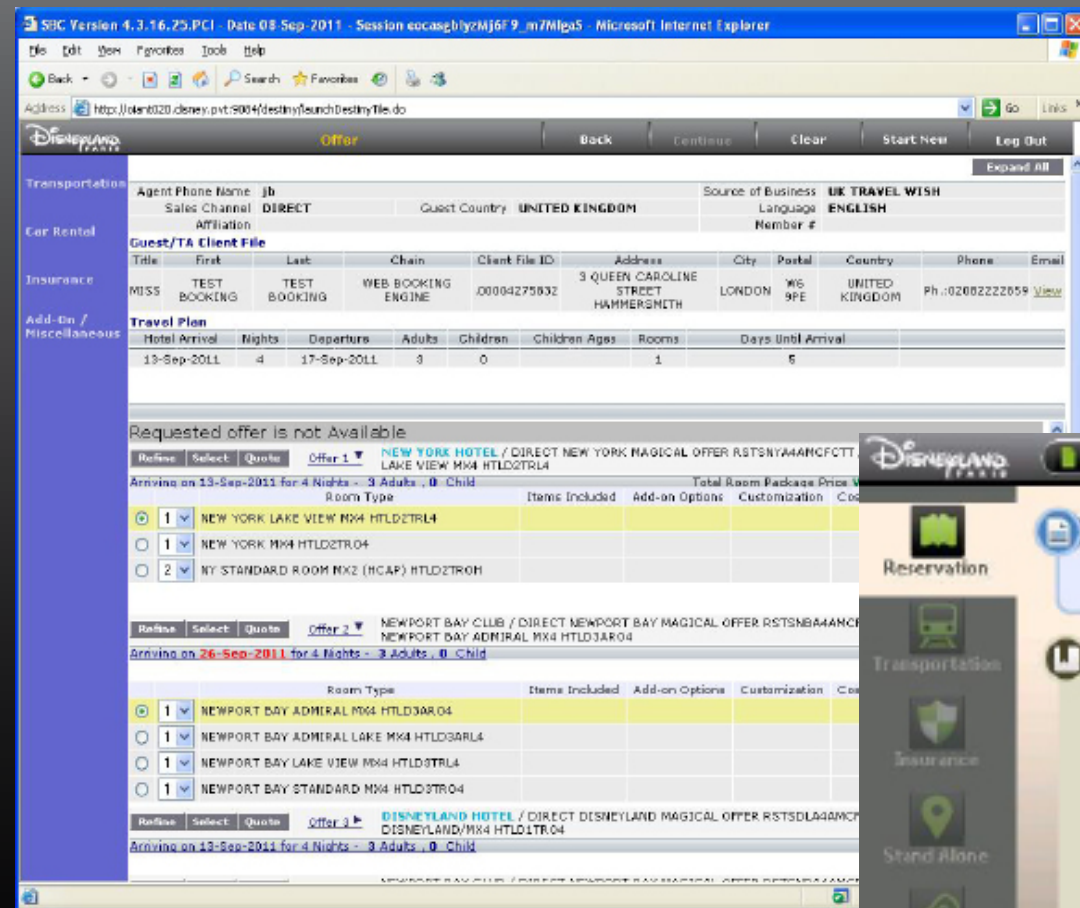
tools/tech

Inkscape
Batik

platform

Java Swing

Disneyland Paris SBC

after

before

TSS ZMB

Leading two other developers at Lockheed Martin whom I had mentored in the arcane world of OO JavaScript, I developed a jQuery-based windowing system dubbed ZMB ("Zombie"), an acronym of our last name initials. It was fairly complete in terms of expected window management: a taskbar showing active and minimized windows, minimize/maximize functionality, and resizable windows. We also built in a stowable sidebar panel and a trashcan to accommodate our existing drag-and-drop, icon-based application framework.

This was done as a proof-of-concept for the Lockheed Martin Training Support System to manage classroom and flight curricula, sortie scheduling and student/instructor scheduling for the Royal Singapore Air Force.

It was shelved temporarily until we were able to drop support for IE7, which suffered performance problems using this interface.

tools/tech

Pure jQuery
HTML

platform

IE8+