Simon C. Wong's Portfolio

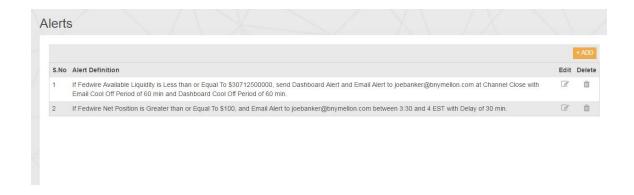
BNY Mellon

One of the most often used applications at BNY's Treasury department is their alerting application. The current Alerts application needed a redesign so it can be more extensible for new features and more user-friendly so new and existing users can easily create alerts to warn them of any monetary change they want to monitor.

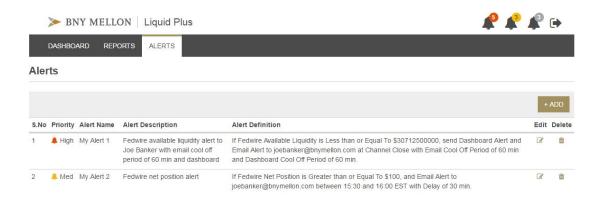
Technologies used: Rapid prototyping (1-2 days turnaround) using HTML5, CSS3, Bootstrap, Font Awesome, ES6, CanvasJS, D3js, jQuery UI, and AngularJS.

Before:

The Alerts Dashboard IA design was unchanged but the company corporate styles were applied.



After:

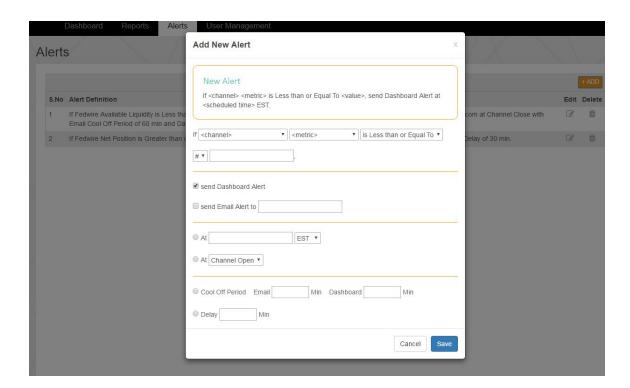


Before:

Most of the change is on the edit/new alert page. The original design is one big form which will only grow bigger as new functionalities are added and can be intimidating to a new user.

Most of the options are hidden in combo boxes and so the user must click on each one to see all his choices before he can select one.

In addition, putting the form in a modal does not really serve any purpose and limits the available space.

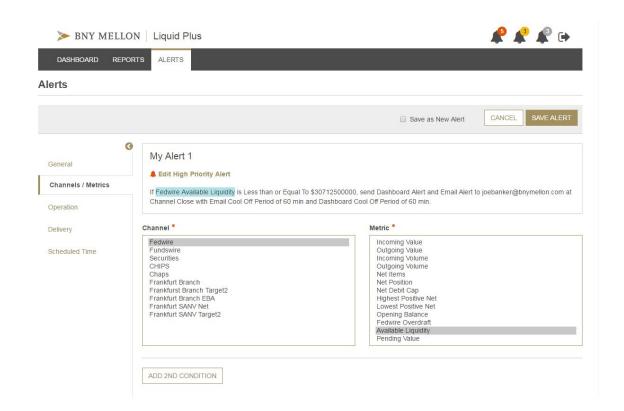


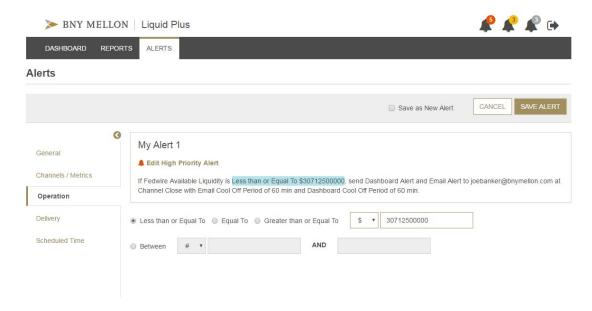
The new design utilizes the corporate styles and divides the alert into several sections thus allowing more room for new features like adding a 2nd alert condition or using a range operation.

Each alert section is accessible from a traditional left navigation or just by clicking on the alert preview at the top. The alert preview in this design now serves both as a quide with it's blue highlighting and as navigation.

With the modal gone and each section having ample of space, all options are now exposed either via exposed combo selector or radio buttons so a user can easily scan all his choices without clicking or hovering on anything.

I also proposed a new functionality, "Save As New Alert" depicted by a simple checkbox on the top control bar. This only appears when editing alerts and will basically allow a user to use an existing alert as template for a new alert, thus reducing the work required when creating a new alert from scratch.





Morgan Stanley

At Morgan Stanley, one of the chief applications their financial advisors use is their portfolio management system to create/edit portfolios for their clients.

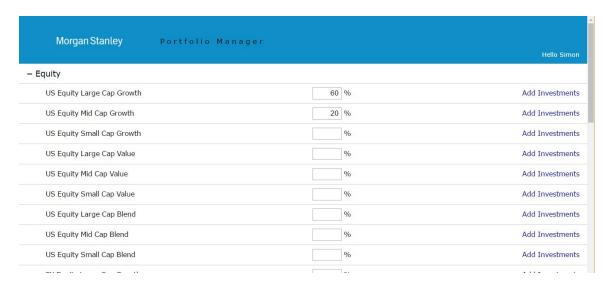
Technologies used: AngularJS, HTML5, CSS3, D3js, jQuery, jQuery UI.

Before:

The original application's create/edit page below suffers from a bad design because it requires the user to fill the allocation of the asset class in order to choose from a list of 93 sub asset classes. For instance, if a user fills the allocations for 2 Equity asset classes and 1 Fixed Income asset class, he cannot see the selected Fixed Income asset without scrolling down (See below "before" screenshot).

Also the selected portfolio is not obvious since the filled allocation is the only sign that it is selected.

Lastly, the original application does not satisfy a use-case of when a user only wants to select the asset classes first and then fill out all the allocations later.



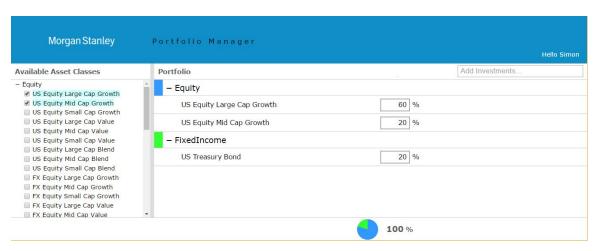
In the much improved new design, the available list of asset classes is now separate as a tree list selection on the left where the user selects from it. The selected assets are then displayed prominently on the right where the user can clearly see his portfolio.

For a small portfolio such as the one below, the user can easily view it without the need to scroll down. The user can actually see the Fixed Income asset class here that he was not able to see on the previous design without scrolling down. Even for a larger portfolio, the user can be assured that his portfolio is clearly displayed on the right and that the asset classes he wants for the portfolio are selected.

The use-case of the FA selecting the asset classes and leaving the allocations blank to be filled out later can now be achieved with the new design.

In addition, industry-standard colors are now associated with each first-level asset class so they can be easily distinguishable.

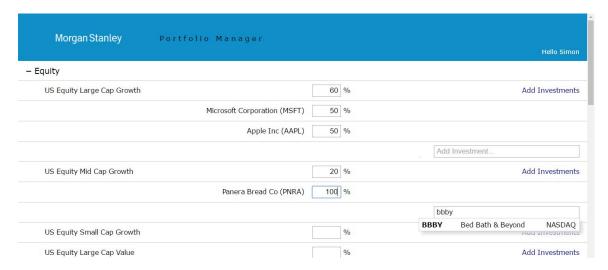
The footer is now fixed so the total is visible at all times. A pie chart of the portfolio using the asset classes' associated colors is also added on the footer and serves as a useful visual cue.



Before:

The second step of creating a portfolio requires the selection of the actual investments under each asset class. In the old design, the user clicks on the Add Investments link to reveal a typeahead search under the asset class. The user then adds the investments under each asset class using this search feature.

Although this may seem harmless at first glance and when creating a simple portfolio such as the one below, the user will soon find this to be quite a chore when dealing with a large portfolio with 20 different asset classes. Now, selecting the investments would mean clicking all 20 Add Investments links and cycling through all 20 typeahead searches just to fill out the investments for all 20 asset classes.



In the new design, it is no longer necessary to click on any link to add investments. There is only one typeahead search on the top which searches through all the investments for all the user's selected asset classes.

And since the typeahead is keyboard-friendly, the user can now easily select all the investments for the portfolio without moving the mouse cursor.

All the results from the typeahead are nicely grouped by the sub level asset classes the user has selected for the portfolio and in the same order.

Checkboxes are also added on the typeahead results to allow multiple selection.

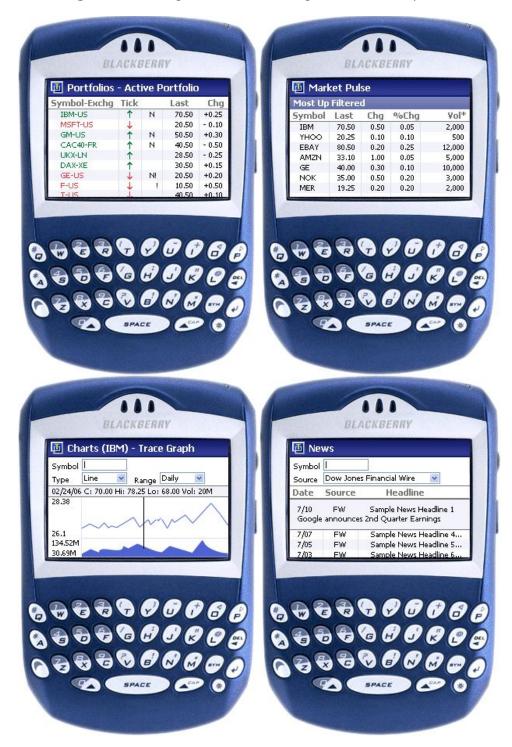
The investment allocation textbox styles are made lighter to differentiate it from the asset class allocation textbox.



Thomson Reuters

As Lead Interaction Designer, one of my most exciting tasks was to design mobile user interfaces based on the company's existing desktop theme.

Technologies used: Designed treatments using Adobe Photoshop.

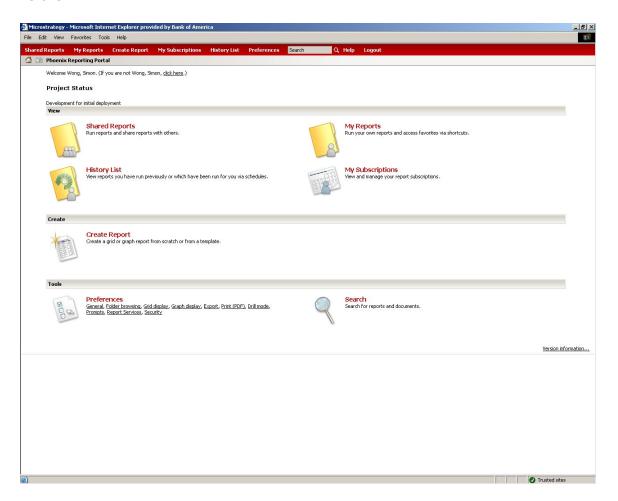


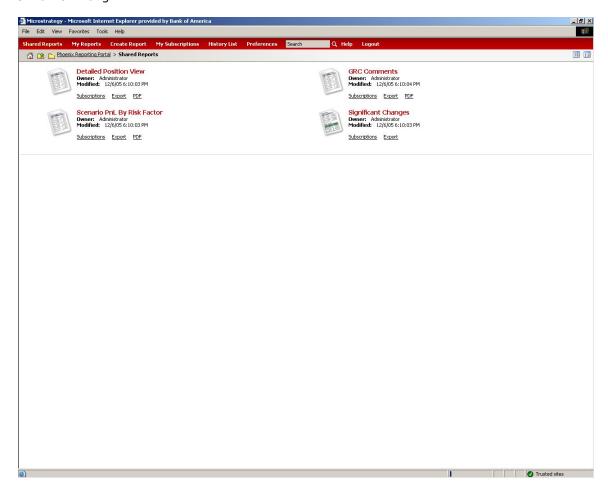
Bank of America

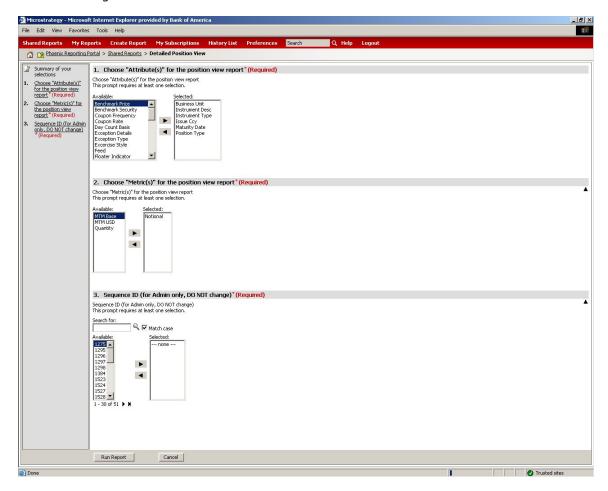
MicroStrategy Web Universal is a powerful on-line reporting tool. I customized this third-party application to adhere to company design guidelines and improve usability of the application primarily by reducing the number of page reloads required to accomplish a particular task. This was accomplished by consolidating key page functionalities onto one page.

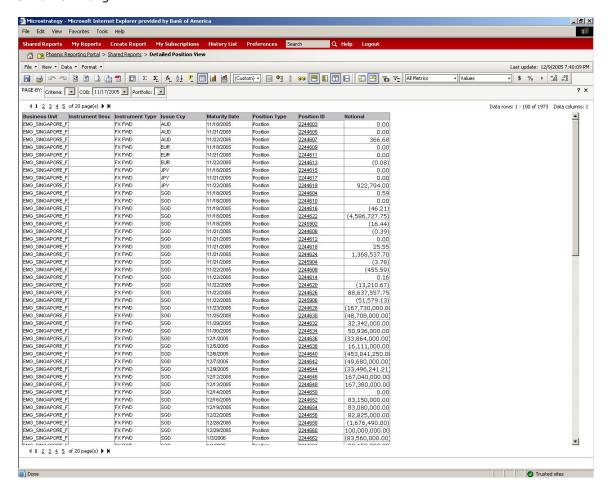
Technologies used: MicroStrategy Web Universal, HTML, CSS, JavaScript, Java. Weblogic Application Server. Wireframes created in Visio. Created and enhanced images using Adobe Photoshop.

Before:

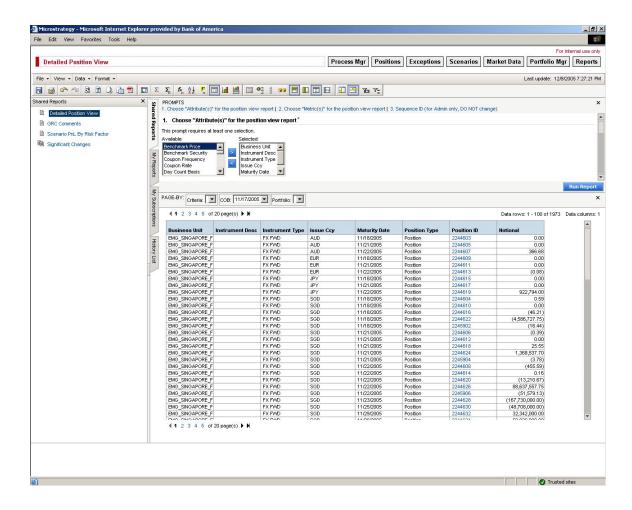








As a result, all the above page functionalities are consolidated onto one single page and are easily identifiable and accessible. Having everything in one page along with the implementation of the AJAX components virtually eliminates all page reloading.

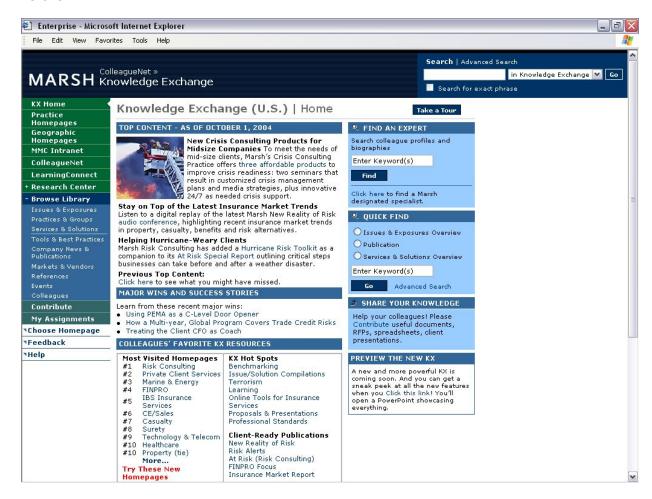


Marsh Inc.

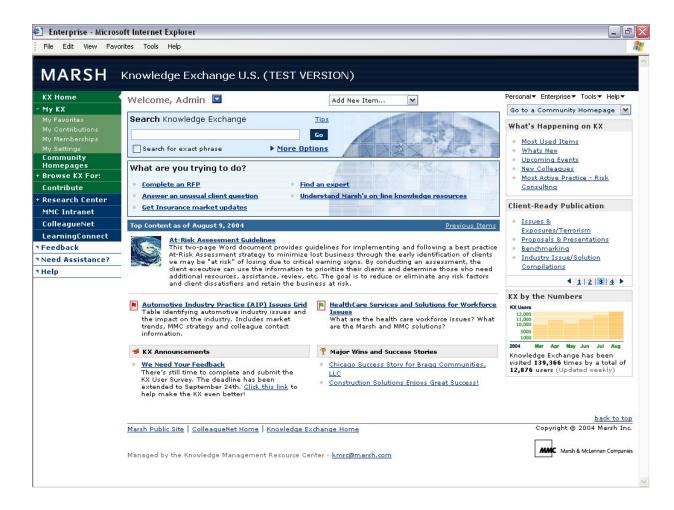
Knowledge Exchange is the central repository where Marsh employees find and exchange information in many different areas of risk and insurance. For this site, I consolidated all search widgets into one central search tool and reorganized the left navigation for better usability.

Technologies used: HTML, CSS, JavaScript, and OScript using Open Text's Livelink Enterprise Server. Wireframes created in Visio. Created and enhanced images using Adobe Photoshop.

Before:

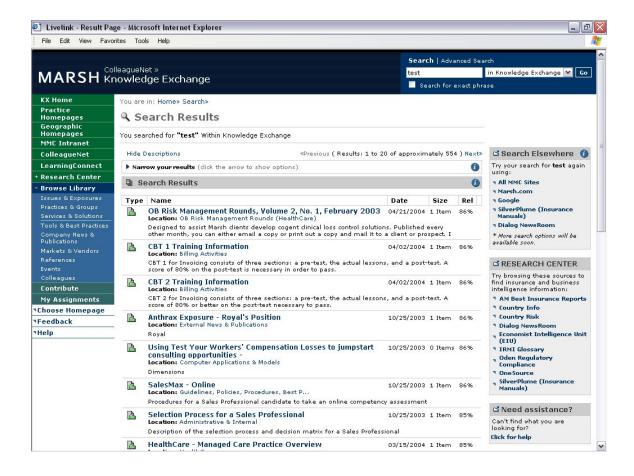


The aftermath is there is only one search tool with the option for more advanced search capabilities. The left navigation was organized so that all menu items in green are pertinent to this site, all blue ones are related to resources external to the site, and all white ones for auxiliary help tools.

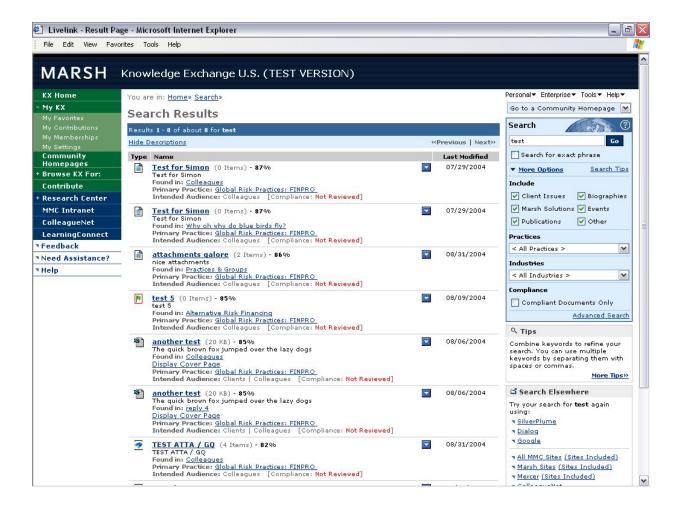


Before:

For the results section, there was too much clutter.



After redesign, the results section contains only 3 columns, having the most important data placed in these columns, while the rest of the data under the main column. The same search tool from home page (miniaturized) was implemented on right column and the restructured left navigation on the left column.

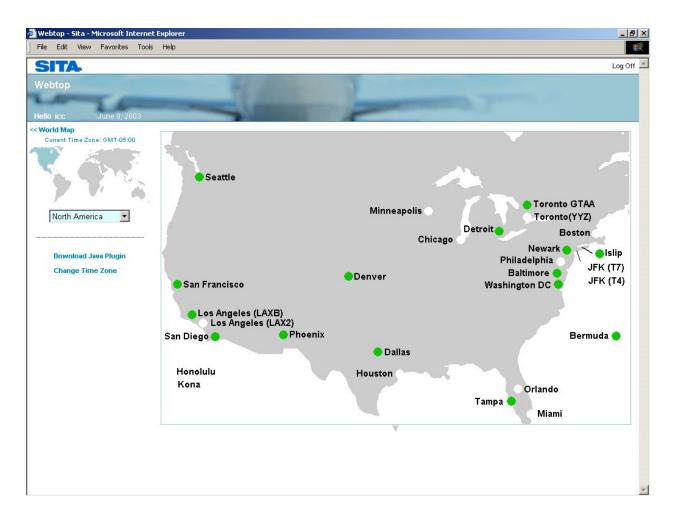


Sita Airport Services

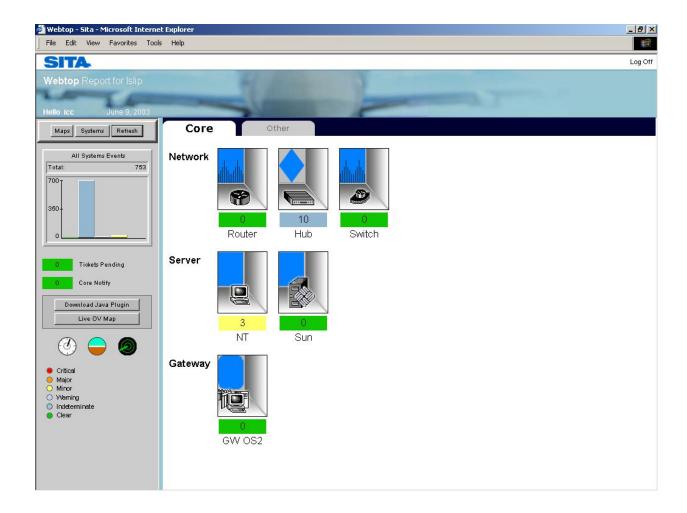
At Sita, I designed, coded, and tested the operations group's centerpiece web-based application that enables its personnel to troubleshoot and monitor the status of hardware components belonging to over 90% of the world's airports and other facilities around the world.

Technologies used: HTML, CSS, JavaScript, and Java Applets using Micromuse's Webtop Application Server. Created and enhanced images using Adobe Photoshop.

This is one of the main screens of the application. From this screen, a user can choose any geographical region in the world. Each dot represents a client's location which is normally an airport. A green dot represents no critical issues with the hardware. A red dot indicates a critical issue. Clicking on the dot will drill down to a more detailed view with all the hardware components belonging to that location (See next page).



Here is a page listing all the hardware components for Islip Airport. Core hardware components are listed under one tab. All other auxiliary components are on another tab. Stats and tools are found on the left column.



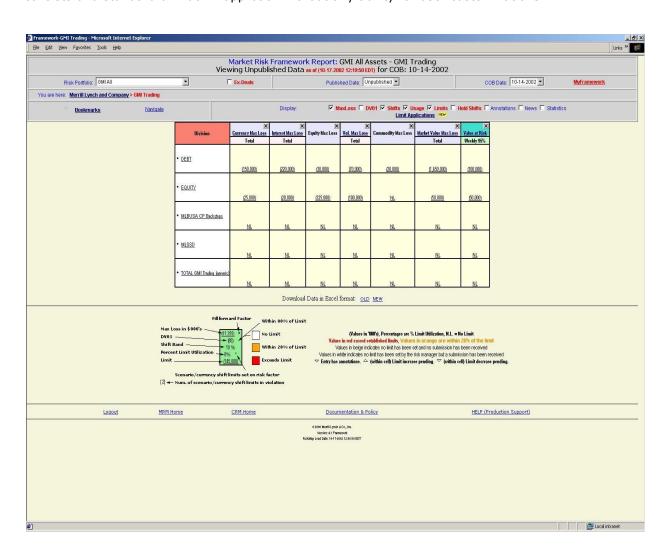
Merrill Lynch

At the Risk Management department of Merrill Lynch, I designed, prototyped, and integrated all risk applications, created a central repository for early design prototypes allowing senior management and users to discover and interact with early working designs, and redesigned the department's web site.

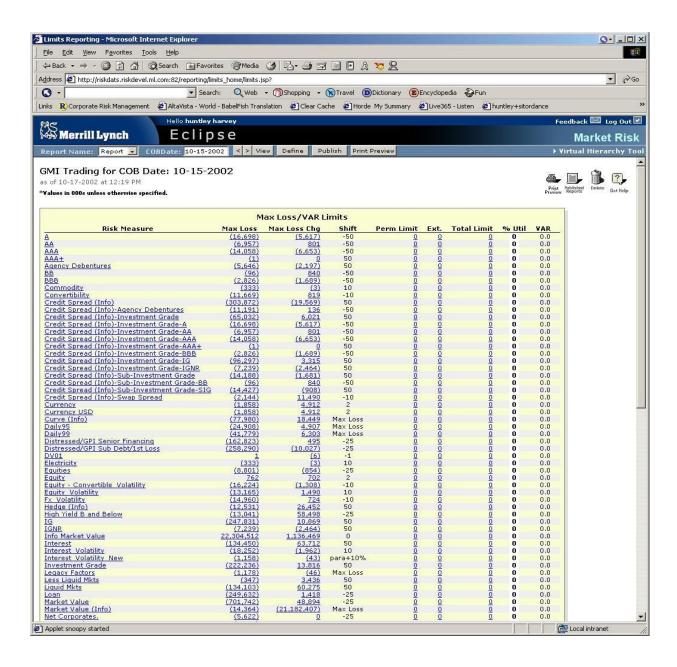
Technologies used: Java Server Pages, Java Servlets, Java Beans, JDBC, DHTML, CSS, JavaScript, ServletExec, Apache Web Server, and Sybase Database. Created and enhanced graphics using Adobe Illustrator and Photoshop.

Before:

One of the key projects that I oversaw was the redesign of their central risk application, Framework. The original "Before" design was terribly outdated and using OTB 1st generation html controls. The IA design consists of a standard drill down approach without any ability for user customizations.



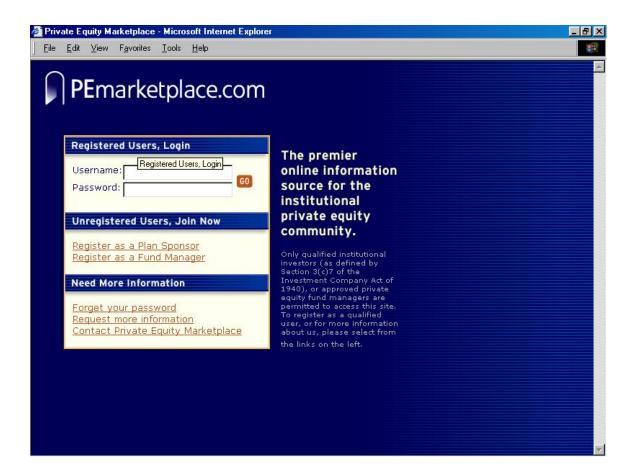
The result is not only a better and more stunning visual design, but one which greatly improves usability by allowing the user greater freedom in viewing, customizing, and saving reports pertinent to his business unit. The new application also allows a way to exchange and receive reports via a built-in email, scheduler, and address book feature.



2Bridge

At 2Bridge, I designed, prototyped, and implemented a financial research application for one of their top clients. This application is much like Yahoo! Finance where the user can find information on stocks and other financial data and can also customize it based on his personal settings.

Technologies used: Weblogic EJBs, Servlets, JSPs, JDBC, HTML, JavaScript, and Oracle stored procedures running on Weblogic Web Server.

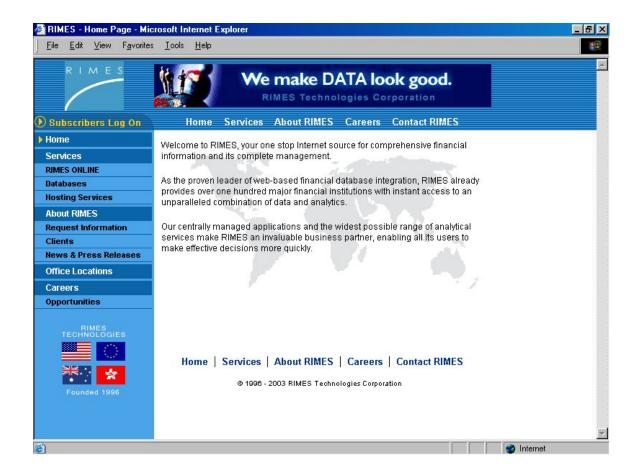




Rimes

As one of the leading providers of financial software and data, the Rimes official web site needed a simple, yet sophisticated, and professional feel. Since there is a great deal of content to start with, it is important to keep it simple so as not to intimidate the user, but still maintain a corporate feel.

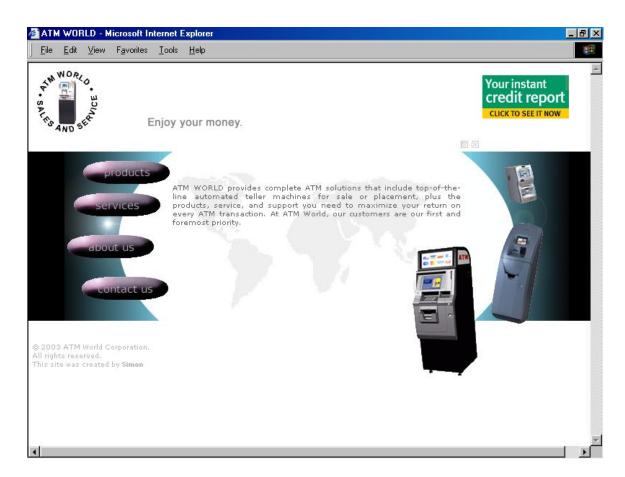
Technologies used: Active Server Pages, JavaScript, VBScript, HTML, and Microsoft IIS Web Server. Created graphics using Adobe Illustrator and Photoshop.



ATM World

At ATM World, I designed a professional web site for them so they can remain competitive with other ATM distributors. An on-line listing of their products offers a convenient way for their customers to stay informed.

Technologies used: DHTML, JavaScript, and CSS. Created and enhanced graphics using Adobe Illustrator and Photoshop.



Atlas ATM is a sister site to ATM World of which I also designed.

