Project Number: MQP-CEW-0602

Web Desktop Environment

A Major Qualifying Project Report:

submitted to the Facility

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Matthew Harrison

Date: May 2, 2006

Approved:

Professor Craig Wills, Advisor

Abstract

This project creates a working prototype of a web desktop environment. With this project I am exploring the idea of merging web applications into a single customizable web environment. The goal is for a user to be able to run multiple web applications in one web browser window. A sample of users was surveyed to evaluate the prototype and determine the potential use of this system. Based on the surveyed users, this system has a good potential for future use.

Table of Contents

Chapter 1: Introduction

1.1 Overview

The web is moving towards a new age. It is moving away from static web pages displaying content to interactive pages where users are able to post and manage content to their needs. This can be seen in many places, such as message boards and wikis. Many popular sites such as Google and Yahoo are allowing users to customize their pages to display the content the user wants and not rely on a specified set of data that the company wishes to display.

In the emergence of the new web technologies that allow these dynamic pages to be created, there has been an emergence of a second branch of web pages. These branches are web applications. Web applications allow a user to go to the specific website and use the tools on the site to accomplish a certain task. Wikipedia provides a good definition of a web application:

> In software engineering, a web application is an application delivered to users from a web server over a network such as the Internet or an intranet. Web applications are popular due to the ubiquity of the web browser as a client, sometimes called a thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. Web applications are used to implement webmail, online retail sales, online auctions, wikis, discussion boards, weblogs, MMORPGs, and perform many other functions[1].

As new web applications are developed, the world of computing is moving away from having applications installed locally on the user's computer to the mobile age, where the applications will be centralized on servers where the user is able to use a thin client to connect to these applications. The web browser is an ideal client since it allows for a large amount of graphics and text to be rendered, and it has become standard to have one installed on every computer. Many companies are taking advantage of web applications so that they do not have to distribute software to the thousands of employees that they have.

One step in the ability to develop new applications is the scheme of AJAX (Asynchronous JavaScript and XML). This new technology allows for background requests to the server to be made asynchronously from the application interface. When AJAX is used, the application is not interrupted when new data needs to be sent or received from the server. There are many methods for doing this as long as data can be transmitted in the background without interrupting the current process. The most popular method is using the JavaScript[2] function XMLHttpRequest[3] which makes a request to a URL and is able to receive and parse the returned data in Extensible Markup Language (XML)[4]. This method is where the AJAX name was coined. With the advent of AJAX, many web application ideas can now be done which would have almost been impossible in the past. With the technology of today, a giant leap can be taken into new regions. With this project I am exploring the idea of merging web applications into a single customizable web environment.

This goal of this MQP (Major Qualifying Project) is to develop a web application system. I believe that the next step of web applications is to move towards a single environment in which a user's web applications that are commonly used can be accessed. The second step of merging these web applications is allowing data saved in one application to be used in other applications, creating a complete merger of these applications. In my project, I would like to create what I believe would be a good way to incorporate web applications and the framework system where the web applications can be made to work inside the system. This system will have a full virtual window management system, which allows for all aspects relating to the system to be seen on one single web browser window. I plan to initially create support for the FireFox web browser. Support for other browsers would be a future project.

The system takes full advantage of newer web technologies, using JavaScript to allow the browser to display dynamic content inside of the web browser. AJAX allows for behind the scenes access to the web server to update the browser with new data as needed. The server uses PHP to manage the content that is sent to the browser as well as connecting to the database. A MySQL database is used to store all data required. This system is able to be installed on any web-server running PHP and MySQL.

For this project, I am creating a prototype of what I believe to be a good way to run a variety of web applications that a user will be able to use daily. These applications will most commonly be applications such as word processing, spreadsheet, calculator, and email. I would like it to be expandable to fulfill any need a user would have for an application in an office.

1.2 Potential for Use

Since individual users' habits differ, some may like having their web applications together while others may prefer to use them separately. This MQP attempts to find out if an integrated web application environment would be of use to users. Some users may feel an integrated interface is awkward and clumsy while others may find that it is smooth and streamlined.

1.3 Goal

The goal of this project is to create a prototype that will provide basic functionality of what a web desktop environment could be. The features of this include incorporating all aspects of the system into one web browser window while not having a need to load multiple web pages to use the system. It will support a minimal amount of functionality in order to collect user feedback on what potential they see in a system similar to this one. A few applications will be included in order to show the potential of what type of usability the system could have. The environment is different from any that have been created before, so in doing this, I will be exploring what potential users would think of an environment to access multiple web applications at once. This environment would potentially replace the applications used on a current operating system.

1.4 Roadmap

In Chapter 2 I discuss background information and work related to this project, along with the languages and tools I used. Chapter 3 contains design information, which includes the overall design of the system and the specification for the application framework. Chapter 4 describes the implementation of the system, covering how it was implemented and problems I faced along the way. Chapter 5 gives an overview of the system in action, detailing features of the system. Chapter 6 provides an analysis of the surveyed users along with the results of the project. Chapter 7 concludes the report, including potential future work that could be done along with results of the project. Appendix A contains the survey that was given to the users. Appendix B contains the individual responses received from the users tested.

Chapter 2: Background

2.1 Previous Work

A layout that uses a simplified window management system of what I would like to create can be seen at http://protopage.com/v2[5]. The difference with Protopage is that each virtual window on the screen only contains a small widget, not a complete web application. The Protopage uses the movable windows only for setting up a user's page; once the user sets the windows, they stay in one place. With this system, a user would be able to switch applications, move them around, minimize, and maximize them. It also would allow a user to launch new applications with the click of a button. This site shows me that the tasks that I want to be able to do in window management are possible, so I know I will be able to create my own system to do what I would like.

Writely is a web word processor. It can be found at http://www.writely.com/[6]. It allows a user to create and save word processing files from the site, saving the files to their server. This is a similar feature to what I would like to incorporate into my web system. The word processor part will be opened up in a virtual window to be able to be managed along with other virtual windows. When saving the document, it will be saved to the system in a similar way to Writely, except that it will be saved along with documents from all of a user's web applications, allowing a user to manage files from all the web applications together.

Other work has already been done at WPI to explore the concept of a using a web browser as the user interface for web applications. This project was done in 1995 by Jon Hanson and Stephen Rubin as an MQP[7]. Their project used Xmosaic as the browser and provided an e-mail interface, a shell interface to execute UNIX commands, and a simple file manager that allows copying, moving, and deleting files. Now 11 years later in 2006, there are many web applications available to be used. My project goes further to develop a common interface to support multiple web applications with a powerful interface to provide an enhanced user experience.

Javeline® DeskRun™ is a software application which allows for web applications written in their own Javeline® FrameWork™ to be compiled into Windows applications[8]. Applications can look and feel exactly like a native Windows application. The framework itself is quite complex as it allows for web applications to seem like desktop applications. Compiling it into a Windows application allows for it to be run offline on the user's desktop. The software is currently in beta stages and does not appear to be feature complete currently. A future possibility would be implementing the Javeline® framework into an interface system like the one I am creating in this project would allow for a large array of applications to be created

2.2 Languages and Tools

PHP is a scripted programming language that can be used to create websites. It is an open-source programming language used mainly for developing server-side applications and dynamic web content[9]. PHP was chosen as the server side programming language for its open-source nature, power of functionality, and easy interaction with databases. I also chose it because I have previous experience with the language.

MySQL is a free implementation of a relational database which uses the SQL (Structured Query Language) syntax for its commands. Along with being free, it is a robust language which is able to access data from the database quickly and efficiently[10].

AJAX is not a technology in itself, but is the method of retrieving data from a web server in the background. XMLHttpRequest can be used to send and receive data from a web server using a client side scripting language such as JavaScript, and the client side language can then interpret the data received from the web server (usually in XML in this case) to make modifications to the page[11].

HyperText Markup Language (HTML) is a markup language designed for the creation of web pages. HTML is used to structure information, denoting certain text as headings, paragraphs, lists and so on; and can be used to describe the appearance and semantics of a document. For this project, HTML is used for the graphical presentation[12].

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in a markup language. Its most common application is to style web pages written in HTML. CSS is used by both the authors and readers of web pages to define colors, fonts, layout, and other aspects of document presentation. It is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation (written in CSS). CSS is used in this project to define size and colors of objects on the screen[13].

2.3 Summary

This chapter covers various related projects that are similar to this one. Protopage contains window moving techniques that are similar to ones that I am implementing into my design. Writely is a sample web application that could be implemented into my system. The previous MQP project gave an example of an interface for basic web applications on the web. The Javeline® FrameWork™ shows the interest in making web applications more usable and complex. The languages and tools section gives a summary of all of the languages and tools that are used in this project.

Chapter 3: Design

3.1 Problem

The objective of this project is to design and implement a prototype of a web desktop environment. The user will be able to open and run applications all on the same window system. A web browser, which acts as the client, should be able to load and display the applications on the screen and allow for them to interact with the application. Data that are input by the user needs to be sent to the server in order for information to be saved and loaded. The goal of this project is to create a usable environment, which will demonstrate the ability to run multiple web applications from a single window. The interface must be fluid and intuitive. The user should not be interrupted with having to wait for pages to load. Also he or she should be able to click on any object on the display or drag any window or icon, and the action would give results without the appearance of a page load being processed.

3.2 Design of System

The interface of the web desktop environment has to be fluid and intuitive. To make the interface intuitive, I chose to model it after an interface that almost every computer user would have experience with. This interface would be the one that the Microsoft Windows operating system uses. The interface consists of a desktop area, which allows windows to be displayed on top of it, and on top of each other. The windows can be moved around and arranged however the user wishes. They are also able to resize the windows to whatever size they wish. It also has a taskbar, which allows the user to view which applications are currently running or to bring one of them into focus.

Finally, it has a popup menu, which allows a user to start a new application by clicking on it.

In the background the system maintains a connection with the server in order to load new applications and save what the user has done. AJAX is used for this connection making requests and sending data to the server without the user being aware of these transfers. This behind the scenes work is done to maintain a fluid interaction with the user.

Every application that is loaded is a separate entity that uses its own code to run. The application is loaded into a window object to be displayed to the user. AJAX is used both for loading the code to start the application along with to transfer data that the application may need. The application remains open, unless the user specifically closes it, and other applications may be used along side it. Multiple instances of the same application can be loaded at the same time incase the user wants to use more then one instance of it at the same time.

3.3 Application Framework

In order for multiple applications to be used, they must have some similar properties associated with them. This association is so that they can all be managed by the system in the same way and be able to run smoothly together. To make loading applications as easy as possible, I created a simple structure that all programs must follow when being written. This structure, which I call an application framework, allows all applications to be developed separately. When the code for the application is dropped into the proper directory it will work without changing the environment to run the specific application. Using XML fields, the application is able to split itself up into sections that the environment is able to read. This framework is general and any web application could be created. There is a specific place where the HTML and JavaScript code is placed that is used for the application. There is also a place to include additional files which the application needs to function.

Here is the basic XML layout for an application:

```
<application>
        <width></width>
        <height></height>
        <left></left>
        <top></top>
        <maximize></maximize>
        <htmlcode>
        </htmlcode>
        <toolbar>
        </toolbar>
        <title></title>
        <toolbarheight></toolbarheight>
        <javascript>
                <code></code>
                <file></file>
        </javascript>
        <rightclick>
        </rightclick>
</application>
```

An overview of the tags used follows:

- Application: overall wrapper tag for the application
- Width: default width (in pixels) of the window
- Height: default height (in pixels) of the window
- Left: default x coordinate of the window
- Top: default y coordinate of the window
- Maximize: contains true or false if the application should start maximized or not
- Htmlcode: contains the html code required for the application
- Toolbar: contains the html code required for the toolbar
- Title: the title of the application
- Toolbarheight: the height (in pixels) of the toolbar
- Javascript: wrapper for the section of javascript code for the application
- Code: contains javascript code to be used in the application
- File: the name of a javascript file that needs to be loaded for the application (this tag may be used more than once inside the javascript wrapper
- Rightclick: contains the items to be used in a rightclick menu

3.4 Summary

This chapter describes the goal of the project to create and test a web environment on an array of users. The overall design is to mimic the usability of Microsoft Windows going over the features to be used. AJAX is used to do the transmission of data between the client and server while the system is running. An application framework has been designed to implement multiple applications that share common features in order for them to operate together within the system.

Chapter 4: Implementation

4.1 Server

Now that I had a set of features that are to be implemented, I began implementing these features. First, I need a server where the system will reside. The web server that is required for this project is any standard web server that supports both PHP and MySQL. I originally thought that I could do this on my WPI UNIX account (that every student is provided with), but WPI has now disabled PHP on user accounts, so I was not able to develop my system on my UNIX account. An alternate server to develop on was not hard to find since I currently own an account with 1&1 Internet Inc.[14] along with domain names through them. I decided to host this project in a subfolder on the domain sathallrin.com giving the URL to the demo http://www.sathallrin.com/MQP/.

I setup a new MySQL database on the server with the following database tables shown in Figure 4.1 for storing settings.

The 'desktop' table stores the information about the shortcut icons on the desktop: 'id' is the unique id of the shortcut; 'userid' is the user id the shortcut is associated with; 'name' is the text that is displayed below the icon; 'shortcut' is the name of the application that is opened; and 'locationx' and 'locationy' stores the x and y coordinates of the icon on the desktop.

The 'users' table stores information about each user: 'id' is the unique id of the user; 'username' is the login name for the user; 'password' stores an MD5 hash of the password (for security); and 'showwindowcontents' changes the option of showing outlines of windows, or the whole window, when dragging.

The 'windows' table stores the previous locations of windows (for the applications): 'id' is the unique id of the window; 'userid' is the user id the window is associated with; ‘leftval’ and ‘top’ store the x and y coordinates of the window; ‘width’ and ‘height’ store the width and height of the window; and ‘max’ stores if the application is maximized (it is a varchar(10) even though it only stores true or false - it stores them in strings so the value can be returned to the JavaScript code to maximize or not).

TABLE desktop			
	id	int	primary key
	userid	int	
	name	varchar(255)	
	shortcut	varchar(255)	
	locationx	int	
	locationy	int	
TABLE users			
	id	int	primary key
	username	varchar(50)	
	password	varchar(255)	
	showwindowcontents	bool	
TABLE windows			
	id	int	primary key
	userid	int	
	applicationname	varchar(255)	
	leftval	int	
	top	int	
	width	int	
	height	int	
	max	varchar(10)	

Figure 4.1 MySQL Database Tables

4.2 Detailed Implementation

The first step of the project was to create the desktop that poses as a background. The layout of the page had to be done in CSS and HTML. I created the basic desktop which expands to 100 percent of the screen and does not allow the window to scroll. The next step was to create the windows; the calls to create the windows were written in JavaScript, so they can be created on demand. After getting them to be the right shape and colors, the next step was to make them usable. The resizing was simple as it only required changing the width and height attributes on the window. Moving it was not too challenging as it only required adjusting the location, but I did have to make sure it could not be dragged off the screen and lost. One problem I encountered was that whenever a window was moved; any object around it jittered as the window was dragged over it. I fixed the jittering by creating a huge transparent box that moves over the entire screen as it is dragged, and then goes away once it stops (so it does not get in the way of clicking on objects).

The next task after the windows was to create a taskbar. I accomplished creating a taskbar in multiple steps. First was to create the area for the taskbar to be displayed; this was a simple HTML object which was fixed in size and position using CSS. I created a location for the application launching menu, and also a location for open applications to be seen. I tied into the application launching code a way to create a taskbar item on creation of the window and then link it to the window by the window id number. Once these were created, there needed to be a way to switch between which of the windows is active. Z-index positioning in CSS allowed me to change the z-index of the window and bring certain ones to the front when active. Along with the positioning, changing the color of the title bar of inactive windows and indenting the taskbar item gave the effect of showing which window was actually active.

With windows being able to operate properly, I now needed a way to launch applications. I came up with two solutions. One was an application launcher, which I modeled after the Microsoft Windows Start menu. This is basically a menu where the user would click the icon which would open the menu with a list of applications to launch. The second is desktop shortcuts. I did not want the desktop shortcuts to be static objects on the desktop, so borrowing code from the window system I was able to make the shortcuts able to be dragged around the screen to where the user wants them.

At this point having completed a good portion of the work for the client, I needed to connect it to the server to save and load data. In order to organize the data, I decided I needed to have user accounts which would be used to log into the system and access data pertaining to that account. I created the server side PHP scripts, which allowed for a user to register and then log into the system and store a session id to keep track of which user is logged in. This is one part of the project where I strayed from the goal of creating a prototype to be tested. I spent a good deal of time on the login system implementing security measures to prevent fake logins or unauthorized access to the system.

Now that I had a way to store settings on the server, I needed to interact with the client to load and save these settings. A simple setup script to load the necessary data for it to run on startup was easy to make. The difficult part was getting the settings to save to the server once changed. This required the use of AJAX to communicate with the server. I used the XMLHttpRequest function to make asynchronous calls to the server whenever new data needed to be retrieved or saved. This allows the client to flow smoothly and any data that needed to be sent or received to be managed in the background. The big problem I encountered with this method was forcing the client to wait for data to load as the data was being fetched from the server. This has been causing problems, and I have put some checks in to test to make sure the data is loaded before loading applications. At this point, I am still having problems where the application will begin loading in the window before all the code is loaded, causing JavaScript errors in the browser.

With communication to the server mostly working at this point, I needed a way to start creating applications with a common method to load them. I created an application framework in XML which allows the code for the application to be written into a XML file. Then when the system loads the application, it can treat all applications the same and not need to know what application is what. This framework separates applications from the system; the applications do need to be specifically coded for the system, but it allows new applications to be created without changing the system. My first tests of the application framework involved a simple “test application,” which was a small amount of text inside of a white box, which provided no function, but allowed for testing of the framework’s ability to be loaded and processed by the system.

Another method for interaction is to create right click context menus to give a user more options when accessing the applications. This method involved overriding the web browser’s default context menu and replacing it with one of my own. These menus were relatively easy to create and allowed for the system to launch anything when clicking on it. Upon completion, I realized there was one important feature missing from the system. This was a dialog box which would make a small window pop up to prompt the user for information and which can be clicked for verification. Then the box would disappear and whatever information is entered is processed to do some task. I was able to take my current window system and change it slightly to support dialog boxes which ended up being very similar in style to regular windows. I modified the application framework into a dialog framework, which was able to use the data received from the user. The first one of these I created was one that allowed the user to add new shortcuts to the desktop. The dialog box prompts the user for the name of the shortcut and then gives choices for application links. This dialog would be an option when right-clicking anywhere on the desktop in the new context menu.

After finishing creating the desktop and windows, it was time to test the project. Since the goal of this project is to test the use of the interface, I only needed to make dummy applications, not actual working ones. I did end up creating a working calculator, and the notepad application has some functionality to it. Using the application framework I created previously, I made a few simple dummy that which only load and display the initial screen. Actually developing these applications to do much more would be another project in itself. I created a calculator, email client, file manager, notepad, and a word processor in order to give a small array of potential applications for testing.

One of the final steps to do was to store the window locations and the icon locations to be used again upon next login. This step was done by creating PHP handlers which would save the information about the window or shortcut to the MySQL database when requested on creation of the shortcut or closing of a window.

4.3 Summary

This chapter covers the steps taken to implement the system from start to finish. It goes over the troubles I encountered and how I accomplished them along with the order in which the features were implemented. The first section talks about the web server where the prototype was tested on. The MySQL server was setup with the tables that are required to store the settings for each user.

Chapter 5: System in Action

5.1 System

This chapter starts out explaining what a user would do to access the system. It goes on to explain the various features implemented in the system.

The first step to start the system is to go to http://www.sathallrin.com/MQP/. The user is presented the login screen which can be seen in Figure 5.1. From here he or she can either login to the system if an account already exists, or click on the register button and register a new account. After the system logs the user in, the first screen seen is the desktop. The empty desktop can be seen in Figure 5.2. The desktop contains the menu button and the taskbar. From this screen the user may perform any action that the system allows.

One approach may be to open the menu to launch an application. The menu can be seen in part A of Figure 5.3. With the menu open, the user may open any of the applications or log off of the system. Figure 5.4 shows the desktop with one application being displayed showing what would happen if the user selected the email application. The purpose of this system is that it is open-ended with what a user is able to do. The features exist to completely customize the appearance of what applications are open and where they appear on the screen. A display with multiple applications open at the same time can be seen in Figure 5.5.

What the user sees when he or she moves and resizes windows can be seen in Figures 5.6 and 5.7 respectively. The move feature requires the user to click and hold the mouse button down on the titlebar (seen in Figure 5.3 Part A) and drag the mouse so the outline of the box appears in the correct position. The resize feature works similarly where the user clicks on the resize icon (seen in Figure 5.3 Part D) and drags the mouse so the outline of the box appears in the size the user desires.

5.2 Summary

This chapter shows what the system is capable of and what a user would see when using it. It gives a demonstration of how a user would use features of the system in order to run the applications that they wish.

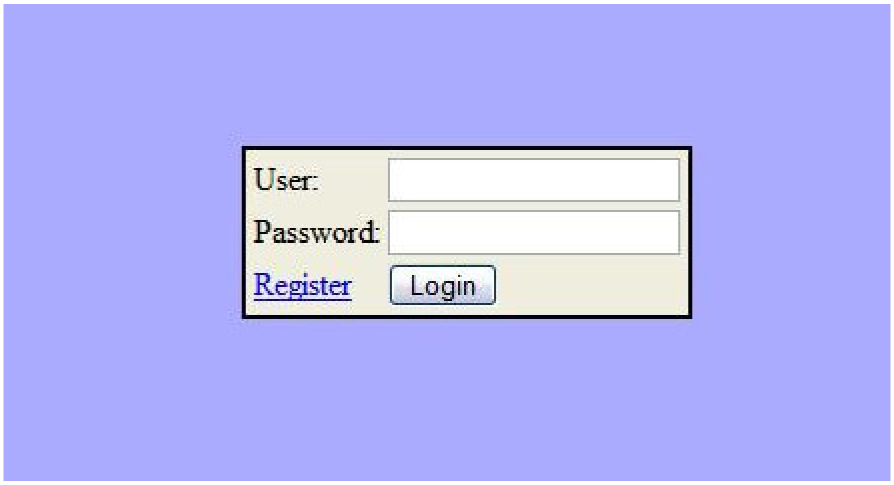

Figure 5.1 The Login Screen

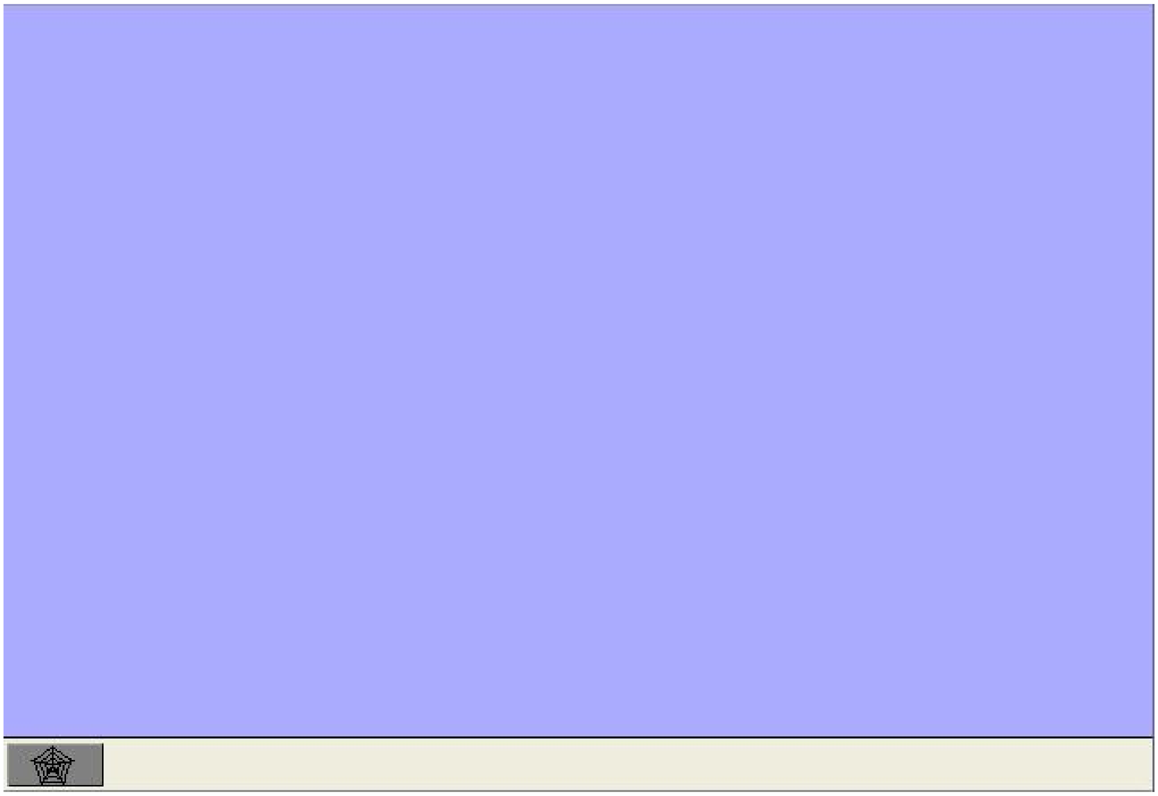

Figure 5.2 An Empty Desktop

A: Menu
B: Titlebar
C: Window controls
D: Resize control
E: Taskbar item

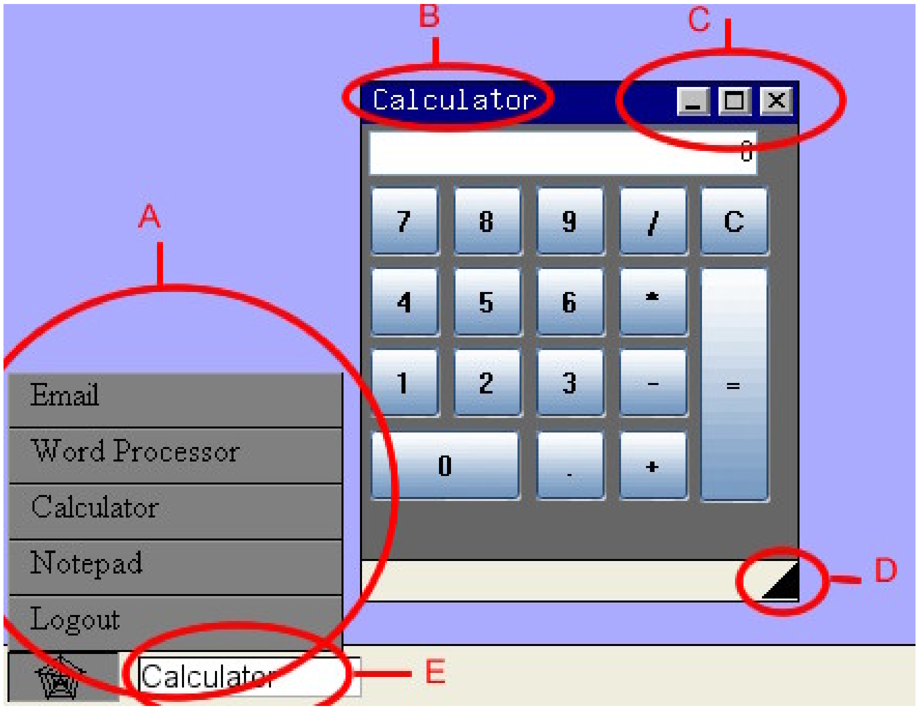

Figure 5.3 Various Parts of the Interface.

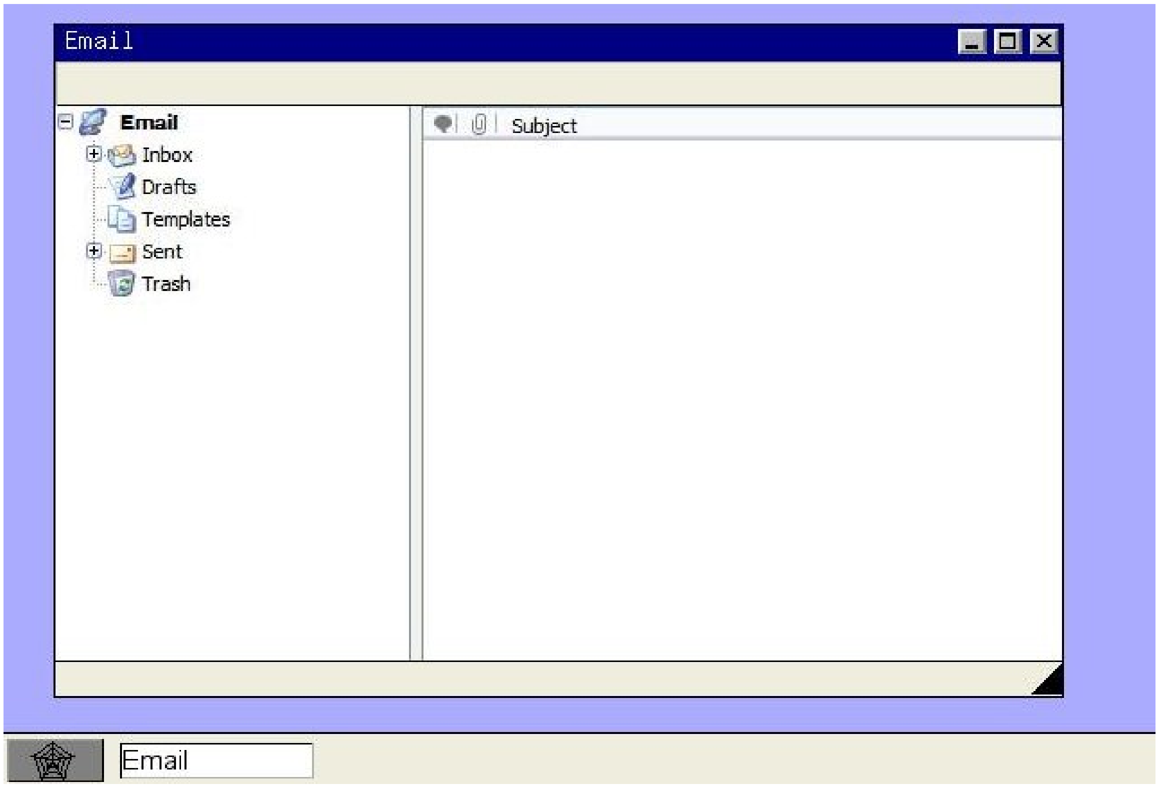

Figure 5.4 A Single Application Open on the Desktop.

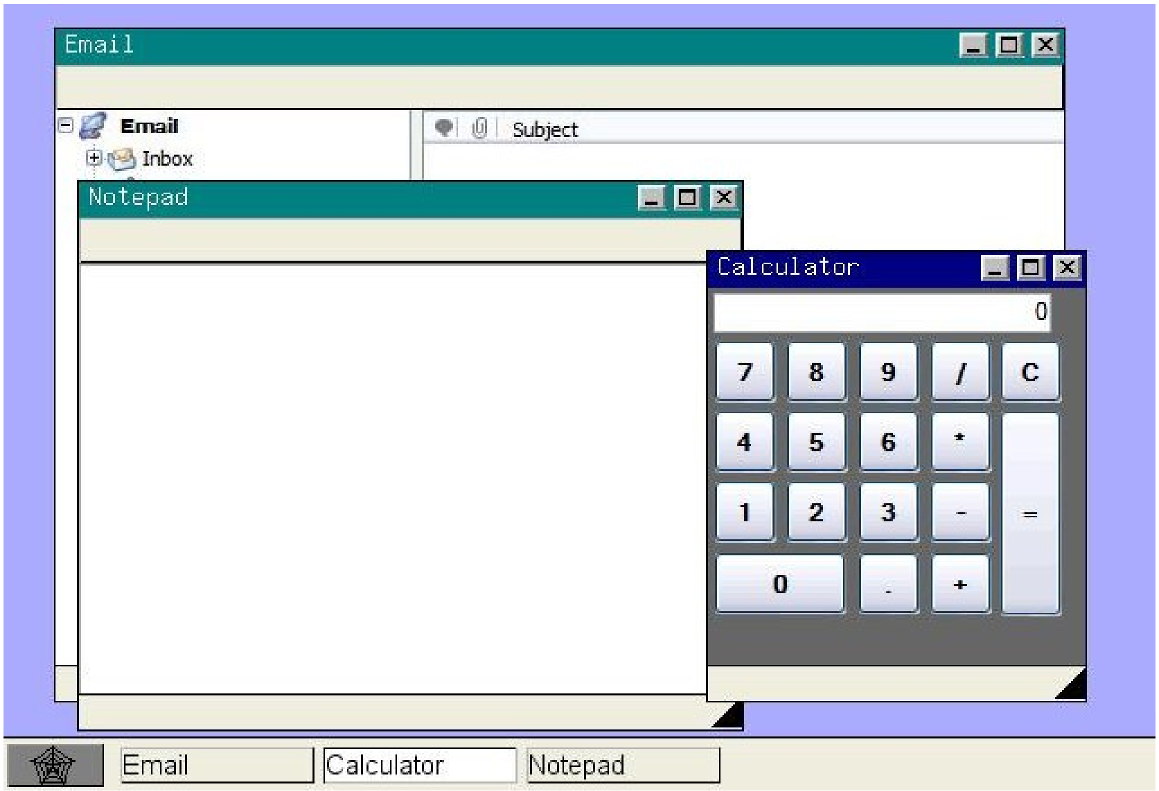

Figure 5.5 Multiple Applications Shown on the Desktop.

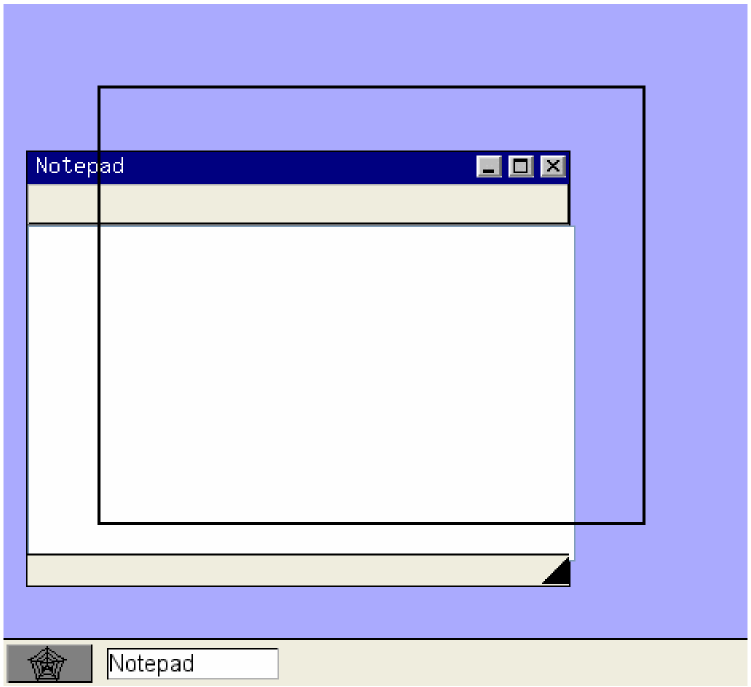

Figure 5.6 Moving a Window

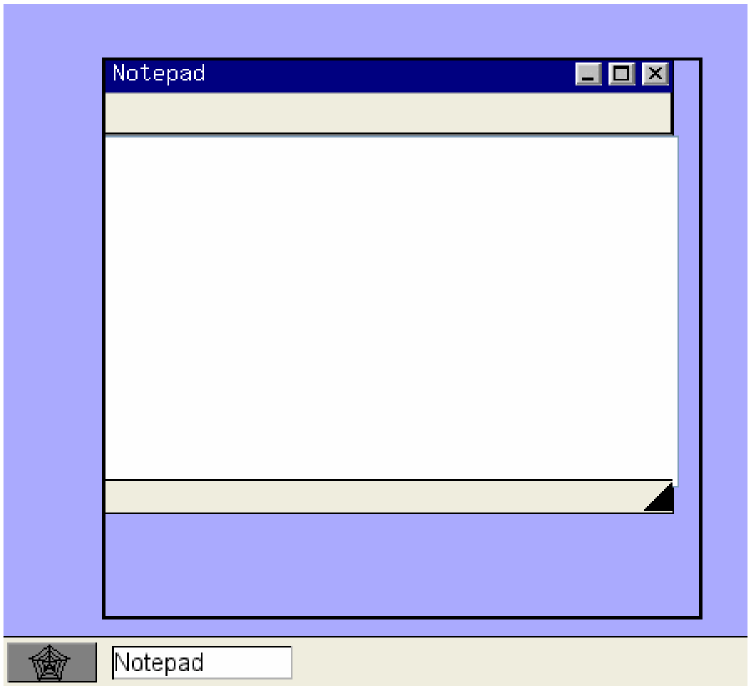

Figure 5.7 Resizing a Window

Chapter 6: Evaluation and Results

6.1 Evaluation

I created a form which I gave to several people to test the system. The form consists of a summary of what the system is along with instructions for running it. There are also 10 questions on the form for the users to answer relating to the system. The form I gave to the users can be found in Appendix A.

6.2 Results

Responses were received from five individual users, the individual responses from each user can be found in Appendix B. Based on the answers to the evaluation form, a majority of the users found that they liked using my system as an interface. Most of the users tested found that the system was intuitive to use as they were used to using Microsoft Windows previously. They found that they were able to find the application and manage the windows easily. Some of the applications that were mentioned that the users would like to see in this type of system are; an email client, a calculator, map software, a text editor, an instant messenger client, RSS feeds, a clock, weather information, games, graphics software, and music players.

When using web applications in general, the users I tested said that they liked the fact that they could access them from anywhere, but did not like the fact that they can be slow and intermittent. The biggest complaints that I had with my system was that it did not look nice enough, but this was expected as I focused on functionality and not the graphics portion of the project.

The feature that stuck out for users was that they were able to control the windows in a way that they were already used to. One of my main focuses for this project was to use an interface that many users would be familiar with, but not necessarily familiar with for using on the web. Based on the responses, it seems that the users were happy with adopting the same interface onto a new system.

When asked if the users would like to use a system like this in the future, most of them said that they would. They stated that it would be convenient to be able to use from anywhere, but more then one user was concerned with the security of keeping private files on the web.

6.3 Summary of Results

Overall the results were promising that this type of system could be used in the future. The major drawback would be performance issues where the system may be slow in some cases. The system would have potential of being implemented in the real world, but with a major concern being in the security of the applications.

Chapter 7: Conclusions

7.1 Future Work

Work on this system could be continued indefinitely. The number of features that could be added to the product is limitless. Since this system is simply a prototype, future work in this area may require starting from scratch to make it more robust. One of the drawbacks of my system is the large amount of memory that it consumes. As applications are loaded, the code for them is loaded into the browser, and when the application is closed the code is not removed. So, when more applications are launched, more memory is used. A garbage collector would have to be created that would free up the memory of applications that are no longer in use.

Exploring the Javeline® FrameWork™ as mentioned in Chapter 2 could be one step in integrating the windowing environment presented here with a potentially robust web application framework. As browser technology improves over time the potential for web application to provide more functionality also increases.

Another option for web applications besides the web browser would be to explore the use of a specific operating system targeted at web applications. This approach would function similarly to my project but the underlying operating system would provide the user interface and the applications would be run from specific web servers.

7.2 Summary

In order to determine if this project is successful it is necessary to determine whether the project completed its goals. The goal of this project was to create a prototype that would provide basic functionality of what a web desktop environment could be and then test this prototype on real users. According to that goal, this project was successful. A working prototype was created and users provided feedback as to what they thought about its usability.

The next question that could be asked is if the web desktop environment is a viable solution. Based on the results of the evaluation of the users, I believe that it is. Most of them said that they use at least some web application regularly. The users found that this system was easy to use and would potentially use a system like this if it contained the applications that they needed. Overall, the surveyed users believed that the system performed to their expectations.

Based on this project, I believe that further work on a system similar to this would have successful results. Even from the small sample of users I tested, I believe that there would be a large demand to use this type of environment to use multiple web applications and there may be similar projects appearing in the near future.

Appendix A: Evaluation Form

This is the survey I used to get feedback.

Web Desktop Environment

This is a prototype of a web desktop environment. The goal of this is to demonstrate what could potentially be a new age of web applications where a user's entire system is located on the web. With this MQP project, I have created a prototype of what I believe to be a solution to a working web environment. Included is the ability to control the windows for the applications. There are a few sample applications to show what the interface could look like.

Please answer these questions before trying the system:

Do you use any web applications currently (webmail, online maps, or other), if so which ones?

What do you like about the web applications that you use, and if not, why not?

Now please go to http://www.sathallrin.com/MQP/ and test out the system. You may either log in with the username/password "guest/guest" or register your own. Now, try out opening multiple applications and test the ability to control the windows. The applications available are dummy applications meaning they do not provide functionality, but are meant to allow you to see the potential for what applications could be present. One exception to this is that the calculator is functional. This isn't meant to be pretty, and the icons are currently only black boxes, so when evaluating, pretend that the graphics are more attractive. Please test and experiment with features of the system and provide your feedback below. *Note: right-clicking on the "desktop" portion of the system will bring up a context menu which allows you to create shortcuts

Please answer these questions after you have tested the system:

Were you able to access the applications? Did you run into any problems running applications?

Did the interface seem intuitive (did you get lost looking for things)?

What did you like about the interface?

What did you dislike about the interface?

Did you like being able to customize (move around and resize) the way you view windows? Please explain.

What type of applications would you like to see in a web environment like this?

Do you feel you may like to use a system like this if it had the applications that you wanted?

Were you able to create Icons on the desktop to launch applications? Did they function as you would expect?

Appendix B: Answers to Evaluation form

Respondent 1

Do you use any web applications currently (webmail, online maps, or other), if so which ones?

Yes, school mail and Yahoo mail

What do you like about the web applications that you use, and if not, why not?

I like them both.

Were you able to access the applications? Did you run into any problems running applications?

Yes, I was able to access them. No, I didn't have any problems.

Did the interface seem intuitive (did you get lost looking for things)?

It was easy to find things.

What did you like about the interface?

It was simple and everything worked.

What did you dislike about the interface?

It was too plain. Also, the word processor and Notebook are basically the same thing, so you probably didn't need both.

Did you like being able to customize (move around and resize) the way you view windows? Please explain.

Yes, it would be annoying to not be able to do that.

What type of applications would you like to see in a web environment like this?

Pictures and music players (like Winamp).

Do you feel you may like to use a system like this if it had the applications that you wanted?

It would be convenient, and most people would probably like it, but I'm perfectly happy with my laptop and I wouldn't trust saving personal or important stuff on the web.

Were you able to create Icons on the desktop to launch applications? Did they function as you would expect?

Yes I was, and yes they did

Respondent 2

Were you able to access the applications? Did you run into any problems running applications?

I was able to access all of the applications and had no problem running any of them.

Did the interface seem intuitive (did you get lost looking for things)?

The interface is very intuitive.

What did you like about the interface?

I liked the ability to overlap and minimize windows.

What did you dislike about the interface?

I would have liked to be able to right click on options in the task bar to close, minimize, or maximize windows

Did you like being able to customize (move around and resize) the way you view windows? Please explain.

Being able to resize windows would be very helpful in a real application. The system used for moving and resizing windows was very intuitive.

What type of applications would you like to see in a web environment like this?

E-mail client, a calculator, map software, a text editor, an instant messenger client, RSS feeds, clock, weather would all be very nice to use with an environment like this

Do you feel you may like to use a system like this if it had the applications that you wanted?

Yes I would.

Were you able to create Icons on the desktop to launch applications? Did they function as you would expect?

The icons I created on the desktop worked as I expected.

Respondent 3

Do you use any web applications currently (webmail, online maps, or other), if so which ones?

Webmail.

What do you like about the web applications that you use, and if not, why not?

I don't like the webmail I use. It's too bland, not enough options, and hard to navigate.

Were you able to access the applications? Did you run into any problems running applications?

Yes, I was able to access it, no problems at all.

Did the interface seem intuitive (did you get lost looking for things)?

No, although the spiderweb image on the "start button" was not exactly clear.

What did you like about the interface?

Clean and easy to use

What did you dislike about the interface?

Nothing besides lack of graphics, but it displays the potential of such a design

Did you like being able to customize (move around and resize) the way you view windows? Please explain.

Yes, absolutely. In terms of navigation potential on a website, it would allow you to have different parts ("applications") of the site open at once in one browser window.

What type of applications would you like to see in a web environment like this?

Email, chat, teamspeak, and a content browser

Do you feel you may like to use a system like this if it had the applications that you wanted?

Yes

Were you able to create Icons on the desktop to launch applications? Did they function as you would expect?

Yes, yes.

Respondent 4

Do you use any web applications currently (webmail, online maps, or other), if so which ones?

Webmail, mapquest, games.

What do you like about the web applications that you use, and if not, why not?

They seem to work reliably, and it's nice to access them anywhere at any time. At times they are slow.

Were you able to access the applications? Did you run into any problems running applications?

yes

Did the interface seem intuitive (did you get lost looking for things)?

At first, yes, but when I figured out the button it was very easy

What did you like about the interface?

It was simple. No crazy graphics!

What did you dislike about the interface?

I did not know when it was done loading, screen was blank.

Did you like being able to customize (move around and resize) the way you view windows? Please explain.

Yes, that is something I would expect to be able to do with any program.

What type of applications would you like to see in a web environment like this?

Everything with the exception of things that require personal information, i.e. accessing bank records, credit car information etc.

Do you feel you may like to use a system like this if it had the applications that you wanted?

yes

Were you able to create Icons on the desktop to launch applications? Did they function as you would expect?

Yes, except I could not delete them.

Respondent 5

Were you able to access the applications? Did you run into any problems running applications?

Was able to access and run the application without difficulty.

Did the interface seem intuitive (did you get lost looking for things)?

It seemed intuitive once I reverted back to a "Windows" mindset and looked for the equivalent of a "Start" button in the lower left hand corner of the interface.

What did you like about the interface?

Very simple to use, despite not being eye-catchy.

What did you dislike about the interface?

I was also able to close the calculator.

Did you like being able to customize (move around and resize) the way you view windows? Please explain.

The manipulation of windows was good. Another thing that I think could change was that resizing could only occur from using the symbol at the bottom right hand corner of the window, and not all four sides and corners (as in Windows). It made the process simpler, yes, but at some cost to initial convenience (as people would get used to it fairly quickly).

What type of applications would you like to see in a web environment like this?

A graphing application would be amazing, as would be a unit and currency converter. An e-mail client as pictured would be of great use, as would be the ability to transfer files.

Do you feel you may like to use a system like this if it had the applications that you wanted?

Yes, I would use such a system if it had the applications I desired.

Were you able to create Icons on the desktop to launch applications? Did they function as you would expect?

Creating shortcuts that way would be good for a system with a fairly limited amount of applications/files on it. Should it have more than say 20 applications/files, a new way to create shortcuts may need to be investigated.

References

[1] Web application, From Wikipedia, the free encyclopedia
< http://en.wikipedia.org/wiki/Web_application >

[2] JavaScript, From Wikipedia, the free encyclopedia < http://en.wikipedia.org/wiki/Javascript >

[3] The XMLHttpRequest Object, W3C Working Draft 05 April 2006
< http://www.w3.org/TR/2006/WD-XMLHttpRequest-20060405/ >

[4] Extensible Markup Language (XML), W3C Architecture Domain < http://www.w3.org/XML/ >

[5] Protopage, free personal start pages < http://protopage.com/v2 >

[6] Writely, The Web Word Processor < http://www.writely.com/ >

[7] "Xmosaic as a Computing Environment" by Jon M. Hanson and Stephen M. Rubin.
A Major Qualifying Project. 1 May 1995.

[8] Javeline® FrameWork™ < http://www.javeline.org/modules/products/framework.php >
Javeline® FrameWork™ and DeskRun™ are Trademarks of Javeline BV

[9] PHP, From Wikipedia, the free encyclopedia < http://en.wikipedia.org/wiki/PHP >

[10] MySQL, From Wikipedia, the free encyclopedia < http://en.wikipedia.org/wiki/MySQL >

[11] Ajax (programming), From Wikipedia, the free encyclopedia
< http://en.wikipedia.org/wiki/AJAX >

[12] HTML, From Wikipedia, the free encyclopedia < http://en.wikipedia.org/wiki/HTML >

[13] Cascading Style Sheets, From Wikipedia, the free encyclopedia
< http://en.wikipedia.org/wiki/Cascading_Style_Sheets >

[14] 1&1 Internet Inc. < http://www.1and1.com/ >