AJAX DEVELOPMENT ENVIRONMENT

REQUIREMENT ANALYSIS REPORT

1. INTRODUCTION 2
1.1 PROBLEM DEFINITION 2
1.2 PROJECT SCOPE AND GOALS 2
2. MARKET RESEARCH 3
2.1 APTANA[1] 3
2.2 TIBCO GENERAL INTERFACE (TIBCO GI) [2] 5
2.3 COMPARING APTANA WITH TIBCO 7
3. USER SURVEY 9
4. REQUIREMENTS 10
4.1 FUNCTIONAL REQUIREMENTS 10
4.1.1 Database Connectivity 10
4.1.2 Editors 10
4.1.3 Debugger 11
4.1.4 Ftp Functionality 12
4.1.5 CVS Functionality 12
4.1.6 Panel Functionality 13
4.2 NON-FUNCTIONAL REQUIREMENTS 15
4.2.1 User Friendliness 15
4.2.2 Portability 15
4.2.3 Reliability 15
4.2.4 Security 16
4.2.5 Performance 16
4.2.6 Maintenance 16
5. USAGE SCENARIO 17
5.1 EDITOR MODULE 17
5.2. JAVASCRIPT DEBUGGER MODULE 17
5.3 PANEL MODULE 17
5.4 DATABASE MODULE 18
5.5 REMOTE CONNECTION MODULE 18
5.6 CVS MODULE 18
6. USE CASE DIAGRAMS 19
6.1 EDITOR MODULE 19
6.2 JAVASCRIPT DEBUGGER MODULE 20
6.3 PANEL MODULE 21
6.4 DATABASE MODULE 21
6.5 REMOTE CONNECTION MODULE 22
6.6 CVS MODULE 22
7. REFERENCES 22
7.1 MARKET RESEARCH REFERENCE 22
7.2 GANNT CHART 23

1. INTRODUCTION

1.1 PROBLEM DEFINITION

Today Internet and Web-Applications have entered all part of daily life. Since, these technologies are used highly by people, speed and interactivity must be indispensable features of these applications. Because of the insufficiency of these features in traditional web applications, developers have developed new technologies. One of these is AJAX (Asynchronous JavaScript and XML).

AJAX refers to a set of techniques centered on background browser to server communication. Instead of always submitting a full page of data to the server and receiving a full page back, by using AJAX techniques, an application can send an individual field value and receive information to update a portion of the page. The result is that, with AJAX, web applications can be much more responsive and intuitive.

Since AJAX is a new technology, new development environments should be implemented to support this technology. As a result of our researches, the number of development environment is not enough and most of available ones are integrated to the general development environment such as Eclipse, Net Beans etc. In other words they are not a stand alone product for the web developers. To meet the needs, we have decided to develop an AJAX IDE (Integrated Development Environment) named as **kajax**.

1.2 PROJECT SCOPE AND GOALS

Our product **kajax** is simply an Integrated Development Environment for Ajax. **kajax** is designed for developers to develop web applications faster and easier with in a user friendly environment with the following goals:

- To develop an independent desktop application/IDE with a setup, supporting operating systems Windows and Linux.
- To develop a modular product using a component based approach.

- To provide a Database Module with database connection, sql query, viewing tables features.
- To supply an Editor Module with HTML, CSS, XML and JavaScript text editors.
- To provide a Debugger Module with a JavaScript Debugger.
- To supply Remote Connection Module with FTP.
- To provide Panel Module with File Explorer, Solution Explorer, Toolbox, Properties and Ajax Action panels.
- To supply Versioning Manager Module with CVS.

2. MARKET RESEARCH

The Market Research helps us to specify the features of our product, prevents us from unnecessary repetition of previous mistakes and unwanted reinvention of available designs, algorithms, techniques, code libraries and applications. There are lots of applications providing development environment for web projects however we have chosen the two most famous standalone AJAX IDEs to examine:

The Market Research help us to specify the features of our project, prevents us from unnecessary repetition of previous mistakes and unwanted reinvention of available designs, algorithms, techniques, code libraries and applications. There are lots of applications providing development environment for web projects however we have chosen the two most famous standalone AJAX IDEs to examine.

- Aptana
- Tibco

2.1 APTANA[1]

- Aptana consists of three editors which are Javascript, HTML, CSS(Cascading Style Shee) editors and also you can use all these editors in a common displaying view.
  - Javascript editor contains code assist, colorization, open declaration(feature to jump a function in the code) properties to make Javascript coding easier.

- HTML editor includes colorization, assisted tag completion, incorporating with Javascript editor for the Javascript code in HTML properties.
- CSS editor also has code assist which provides user hints and suggestions during development.

- Aptana JavaScript debugger provides us with numerous tools to help us closely examine and troubleshoot our JavaScript code. The JavaScript debugger uses its own Debug perspective so that debugging tasks are separated from development tasks. Debug perspective is composed of from the following parts.
  - Debug View displays all processes that are currently running and debugging targets for the Aptana Debug perspective.
  - Variables View to display current stack frame variables and their values.
  - Breakpoints View allows you to work with breakpoints that you have added to your JavaScript code.
  - The Expressions View lists the current value for the expressions that you add to the view. This view lets you inspect your JavaScript code in closer detail.
  - Console View to find out the value of a variable or object at a given time. The value will be written out to the Console.
  - AJAX Monitor View logs all AJAX traffic(XMLHttpRequest and asynchronous HTTP requests) in real time, and allows you to inspect the requests and responses.
- Aptana's FTP tool transfers files between hard drive and a remote web site. You can also download the current files from your remote site, or synchronize both your hard drive and remote site simultaneously. It has the following properties:
  - Adding a FTP site connection defines the site that you want to transfer files to so that you can reuse that connection any time that you want to synchronize that site with your hard drive.
  - Adding a site connection is used to set up a FTP site and make a connection with it.
  - Other properties are editing a file from a remote web site and uploading or downloading files.
- Aptana enables developers to access other available libraries on the web to develop more dynamic web applications such as AFLAX, Dojo, Mochikit, Prototype, Rico etc.

- Aptana`s Snippet feature quickly inserts small pieces of code into your HTML, CSS, JavaScript and XML files so you can save same time by simply clicking the name of Snippet.
- Aptana has a local feature that is used for tracking versions of files in your Aptana project. It has the following properties: setting local history preferences, comparing different versions of a file, recovering a deleted file.
- Other basic functionalities of Aptana can be listed as the followings: adding a bookmark to jump specific places in your code, searching a string in your workspace, configuring Aptana to use an external web server, using Aptana with non-HTML files such as PHP,ASP, adding filters to the error messages to make debugging easily.

Aptana has lots of basic functionalities to help the developers however we don`t want to put all of them here. In our opinion, general features of Aptana above is sufficient to understand its functionality and user friendly features of it. For more information its web site is in references part.

2.2 TIBCO GENERAL INTERFACE (TIBCO GI) [2]

When we start the examination of TIPCO IDE, we see a feature of it, which is different from not also Aptana but also most of the development environments. This feature can be called as web based development environment. When we click the excutable file of program, Tibco General Interface Builder opens it Tibco GIB within an Internet Explorer browser window.

TIBCO General Interface uses a container-based model for visual components, similar to that of Java Swing and other object-based GUIs. Rather than treating a web application as a series of hyperlinked pages, this architecture provides true object modeling when building an application. Through this object modeling perspective, software products are intelligent data refresh, incremental functionality, modular design paradigm. Below, we will tell the components which were implemented in TIBCO using this object modeling view and supplied as pallets at the product.

- Component libraries palette provides a wide array of GUI prototype objects for adding functionality to your application.

- The Component Hierarchy palette lets us view and manage the hierarchical model of objects in our application. When object has taken place in another object an hierarchical situation is realized.
- Properties Editor Palette provides an abstraction layer for the complexities of browser events models and HTML syntax. The objects that make up prototype components are preconfigured with default property values, which can be modified.
- Because of objects having event property, Events Editor Palette detect user actions and react in a defined way, allowing your application to respond to user input.

These all object modeling techniques above are similar to the MS Visual Studio Toolbox. These techniques also have drag and drop feature.

- Since TIBCO IDE is a web based development environment, it should have a mechanism updating the file that we have changed in our computer file system. To realize this TIBCO IDE has data and cache management which does reloading the standard XML, XSL and CSS files into the browser cache, when they have been modified.

TIBCO also has Presentation Logic Development and Debugging Tools which includes several techniques that makes coding and debugging easier. To realize this TIBCO has the following palettes:

- The Project Files palette is a list of all files referenced by an application, including GUI components, XSL files, and included JavaScript files. Files are grouped according to type. Functions and other JavaScript constructs defined in an included file are loaded into browser cache, and can be used by all components of an application. So, moving back and forth between writing code and implementing the logic in GUI components is quick and seamless.
- Dynamic properties can be used to customize look and feel, to localize the application with language resource files, and to create dynamic layouts. In all these scenarios, properties can be specified as static values or JavaScript code.
- The JavaScript Test Utility is a useful tool for debugging any issues with JavaScript code before adding it to your application. The utility provides a

convenient, standalone test environment for your code. It has the property of jumping to a method in our code.

- The JavaScript Step-Through Debugger provides line-by-line access, object inspection, and insight into code execution. Supported operations include step into, step out of, execute all, and cancel execution. Only code within the context of a function can be debugged.

Above are the general functionalities of the TIBCO IDE. It has different modules from the Aptana because of being a web based development environment. Lets see what are the main differences between each AJAX IDE and drawbacks and advantages of these main differences.

2.3 COMPARING APTANA WITH TIBCO

	Aptana	TIBCO	Drawbacks and Advantages
Web Based Development Environment	**-**	**+**	**TIBCO IDE can be used where you have an internet connection and a browser. You don't have to install it.**
Javascript Editor	**+**	**+**	**Both have Javascript Editor however, Aptana has more features to make coding easily such as code assist, open declaration which TIBCO does not have.**
CSS&HTML Editor	**+**	**+**	**Both have but again Aptana has more features.**

Debugger	**+**	**+**	**Both have Javascript debugger and they have similar properties. Aptana has variable view, AJAX view features extra when we consider TIBCO which provides jumping to a method, breakpoint, step by step debugging which Aptana also has.**
FTP Support	**+**	**-**	**Aptana has a FTP support from an added website to our hard drive however in TIBCO we have not such a mechanism supporting FTP.**
Access to other Developing Libraries	**+**	**-**	**Aptana allows developers to use available libraries to make more dynamic web applications however in TIBCO we could not find such a property.**
Enabling Code Snippets	**+**	**-**	**Aptana has snippet property supplying small code parts to make developing easiliy however there is not in TIBCO.**
Component Library	**-**	**+**	**TIBCO has a component library which has drag and drop supports to use these components in development as a result of this property it also has event and properties editor for components however we have no such library in Aptana.**

3. USER SURVEY

Since our project is so straightforward we do not need to survey with specialists. From our market search we have got the necessary feedbacks to decide on the requirements. But meanwhile, we have talked to with our Computer Engineering Department students especially who are related with web development and our friend Mustafa ZENGIN who is a computer engineer in MILSOFT Company via msn. In our conservations we mainly asked the below questions:

- What do you expect from a development environment tool in general?
- Have ever used web development tools such as Microsoft Office FrontPage or Macromedia Dreamviewer? If so, what were the main problems you have faced with? And which features did you like?
- What do you know about AJAX technology? Do you know an IDE for AJAX?
- If you consider a web-based application and desktop application for a specific task, which one of that you prefer and why?
- Do you think a FTP connection and a CVS support is necessary for a development tool?

The feedbacks of these questions were as we expected. Most of them have been used FrontPage, Dreamviewer or another similar tool. As a result, users want a development tool with a user friendly Graphical User Interface, a simple GUI but with necessary main requirements. Moreover, tools interactivity with the developer is very important. Users do not want to write all of the straightforward codes. So, they wants from the tool to help them with auto completion, help comments etc. A debugger is also expected by users. They want to find and solve the problem in the codes by the help of a debugger. Mustafa ZENGIN as a graduate from our computer engineering department, has mentioned the above points and in addition to that he said that a development tool fasters the development process in the companies and developers.

Since AJAX is a new technology most of the students have heart about it. But most of them have not known what its main features. As a result of that they have not heart about an

AJAX IDE. This gave motivation to us to develop a development environment for AJAX which is needed. Mustafa ZENGIN has mentioned that AJAX technology may dominate the web development area in the future. He added that the investigations of the Google to that technology and examples of Gmail and Yahoo mail will increase the popularity of AJAX.

Considering the web based application vs. desktop applications users prefers the desktop applications since it is faster, reliable and easier from the web-based ones. Especially for a development environment the desktop application is indispensable since the other one is a great headache for the users most of the time. Mustafa ZENGIN has added that to make a web-based application may add some other extra responsibilities for us.

Students were familiar with FTP, but a few of them was familiar with CVS. But the ones who used CVS in their summer practices have marked the importance of the CVS. Also, Mustafa ZENGIN has dictated that FTP and CVS are indispensable part of the today current development environments.

4. REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

4.1.1 Database Connectivity

Developers often need to know how the database tables are structured and how the relationships are occurring. To meet this need a database utility should be supplied. Its main properties will be:

- Connection Wizard
- Viewing current tables
- Viewing relationships
- Querying statements

4.1.2 Editors

Since we are building an integrated development environment, providing a text editor is very essential, so in some sense it is the core of the IDE. An Ajax IDE must fundamentally include HTML, JavaScript and XML editors. Besides this presenting a CSS editor will be crucial.

General features of the editors are:

- Tab view
- Show line numbers
- Bookmark lines
- Undo – Redo
- Select all
- Cut – Copy – Paste
- Delete
- Find – Replace
- Set Encoding
- Goto Line
- Line wrapping
- Shortcuts for navigation
- Asterisks in window tabs (on changes in file)
- Increase/Decrease Indentation
- Font Format – size, color, type.
- Tool tips

Specific features will be:

- Intellisense/Auto-completion for tags, functions, variables.
- Syntax highlighting
- Auto pair matching – braces, quotes, etc.
- Pair highlighting
- Code folding/unfolding
- Auto validation check
- Comment/Uncomment
- Tree view
- Auto pair matching for tags

4.1.3 Debugger

We know that there is no bug free code. So, all programmers need debuggers. For an AJAX application your need is mainly a JavaScript debugger. Thus, the JS Debugger will be able to:

- Allow breakpoints
- Continue
- Step into/out/over
- Run to cursor
- Show variables
- Add watch

- Log the outputs
- Show error line
- Debug selected part of code

4.1.4 Ftp Functionality

FTP (file transfer protocol) is used to connect two computers over the Internet so that the user of one computer can transfer files and perform file commands on the other computer. In our program **kajax** we will provide this functionality to the users. So that user can transfer files from his/her hard drive to a remote web site or vice versa. By that way synchronization between user's hard drive and remote site will be accomplished. Features of our Ftp are:

- Setting up a connection with remote host
- Making ftp site configurations
- Creating a connection between the Ftp site and local folder
- Transferring files between Ftp site and local folder
- Editing files in the remote site
- Synchronizing between different Ftp sites
- Transferring files with a simple drag and drop

4.1.5 CVS Functionality

The Concurrent Versions System (CVS) or the Concurrent Versioning System performs a version control system. Mainly, it keeps track of all work and all changes in a set of files, especially the implementation of a software project, and allows several developers to collaborate in the project. CVS has become popular in the free software and open-source worlds. CVS is released under the GNU General Public License. So, our program **kajax** will implement CVS functionality for that aim. Today most of the projects are developed by a group of developers. For that reason, we have decided to provide this functionality to the program. The features of our program are:

- Setting up a connection with a remote CVS host

- Making the CVS server configurations
- Uploading and downloading files between our program and the CVS server

4.1.6 Panel Functionality

Indispensable part of a development environment is its panels. Panels contribute the usability of the program by grouping the functionalities. Panels also provide a simple and efficient user interface to the developer. For this aim or program will provide File Explorer Panel, Solution Explorer Panel, Toolbox Panel, Properties Panel and Ajax Actions Panel to the developer.

File Explorer Panel

File Explorer is necessary for our program **kajax.** Developer can deal with all kinds of files. Our File Explorer has below features:

- Creating CSS, HTML, JavaScript, XML and Text files
- Refreshing files
- Classifying files according to their names
- Showing all files in a tree structure which is collapsible and extendible
- Showing files at Computer (Hard drives), Local Network Connections and FTP

Solution Explorer Panel

Solution Explorer is also an important part of a development tool. Developer must be provided with a solution explorer with the following features:

- Showing all solutions in a tree structure with their files
- Opening selected solution and the file in the editor area

Toolbox Panel

Our aim with the toolbox panel is to provide user basic tools for user interface design of their project. User will choose a tool, and use it by simply dragging and dropping it. The necessary tools are listed below. This list may change later.

- **Block Tools:** Block, Image, Label, Text
- **Containers:** Dialog, Layout, Splitter, Stack, Tab, Tabbed Pane
- **Form Element:** Button, Checkbox, Date Picker, Radio Button, Select, Combo, Text Area, Text Box, Time Picker
- **Matrix:** Grid, List, Multi Select, Tree
- **Menus and Toolbars:** Menu, Menu Bar, Task bar, Tool bar, Tool bar Button

Properties Panel

User will be provided with a properties panel in preview mode to enter the properties of a selected tool. This panel will not be active in source code mode of the program. So, development will be faster and the user will get lost in the code. According the selected tool properties panel will be updated with selected tool's properties. The properties of the selected tool will be categorized into groups.

Ajax Action Panel

Ajax action is the indispensable part of interactive web pages. Otherwise the developer will be bothered up with the implementation details and massive code. User can use predefined ajax actions or can add new ajax actions with specified template. Newly created ajax actions will be saved in **kajax** developer repository. User can remove the actions in Ajax Action Panel.

- Use Predefined Ajax actions
- Add Ajax Action
- Remove Ajax Action

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 User Friendliness

One of our main goals in our product **kajax** is to create the best user interface among the other products. Our research in the current product range showed us that the user interfaces of these programs expect TIBCO and Aptana do not satisfy the users' needs. TIBCO and Aptana simple and user friendly interfaces but the other ones were too crowded. Since interface is seen side of the program to the user, it must be robust. So, we will create a powerful interface that can be modified according to user preferences. This way, the user can create the optimum workspace for his/her project. The interface items will be well-defined that the user will easily access to everything without searching it deeply. The menus and toolbars will be clear, and the user will be able to access almost every menu item with either hotkeys or panels.

4.2.2 Portability

We can safely say that our product will be portable, since during the development process we will be using the platform independent tools. For instance Java Net Beans is open source platform independent technology. The application shall run in almost all computers having a Java Runtime Environment.

4.2.3 Reliability

We are planning to program our product reliable so that any minor program will not stop the program or corrupt the program. For that aim our testers will make white or black box tests to minimize the bugs of our program. So that our program will be as bug free as possible. All modules should work asynchronously so that any delay of one of our modules should not block others work flow.

4.2.4 Security

Since our program is a desktop application there is not a security check for our program. But considering the FTP Connections and CVS features of our program, security check may be done.

4.2.5 Performance

The performance of our program is very important. A slow development environment will not satisfy the needs of the user. Since we are using Java technologies in our program, we have to be more careful about the performance issue. The system resource usage will be minimized to increase the performance on slower computers. The user should run other applications while using our program.

4.2.6 Maintenance

Since we will implement our program with modules using component based approach, maintenance will be our big advantage. Since maintenance is required to be able to solve problems encountered before and after installing the system, we can add new modules after releasing our product easily. As a result of the fact that it may be requested to add new functionality or enhance the program, we will write maintainable source code as possible.

5. USAGE SCENARIO

5.1 EDITOR MODULE

There are 5 kinds text editing in our software: HTML editor, CSS editor, XML editor, JAVASCRIPT editor and generic text editor. After opening a new project, user can add one of these files from File-> Add menu. After adding a new file user can edit this file user can use the functionalities of editor. For example, user can see the line numbers, bookmark lines, do undo-redo, select all text, delete text, cut-copy-paste the text, increase/decrease indentation, format the font (color, size). The editor will be in tab view, namely user can edit project files in different tabs and also there will be asterisk in files to indicate that file has changed after the last save. User can use Find-Replace utility that nearly common to all text editors. Addition to these features line wrapping and tool tips can be used by user. There will be some language specific features that user can use. These are: Intellisense/Auto-completion for tags, functions, variables, Syntax highlighting, Auto pair matching – braces, quotes, etc, Pair highlighting, Code folding/unfolding, Auto validation check, Comment/Uncomment, Tree view, Auto pair matching for tags.

After editing the file user saves his file using File->Save or File->Save As menu.

5.2. JAVASCRIPT DEBUGGER MODULE

While coding JavaScript files user will debug his/her code with the help of our JavaScript debugger. User will set breakpoints to inspect the code carefully. Debugger allows continuing, stepping into/out/over run to cursor functions. User can inspect variables at the instance of breakpoint. Also debugger logs the outputs and shows the error line in the code to the user.

5.3 PANEL MODULE

User can select the panels that required in that project. From the Panel menu he/she can selects which he/she wants to see in development environment. The software has 4 panels: File Explorer Panel, Solution Explorer Panel, Toolbox Panel, and Properties Panel.

File Explorer Panel, helps to user navigate over the files and folders in his/her computer and connected ftp site. Tree representation will help user to see the levels and places of files.

Using Solution Explorer Panel, user can switch one file to another in that project. Tree representation will help user to see the levels and places of files.

In Toolbox Panel, user can find useful components for especially .html files. Using drag&drop he/she will add elements to the .html pages. And Properties Panel will show the properties of the element that user selected. Properties of that element can be changed.

5.4 DATABASE MODULE

User can connect database using connection wizard of our database module. Connection wizard asks user some information about database, like database name, place, user name and password to access database. After connection user can see the tables in database and relationships of those tables. Our database module lets user to query the database and the results can be used easily in project.

5.5 REMOTE CONNECTION MODULE

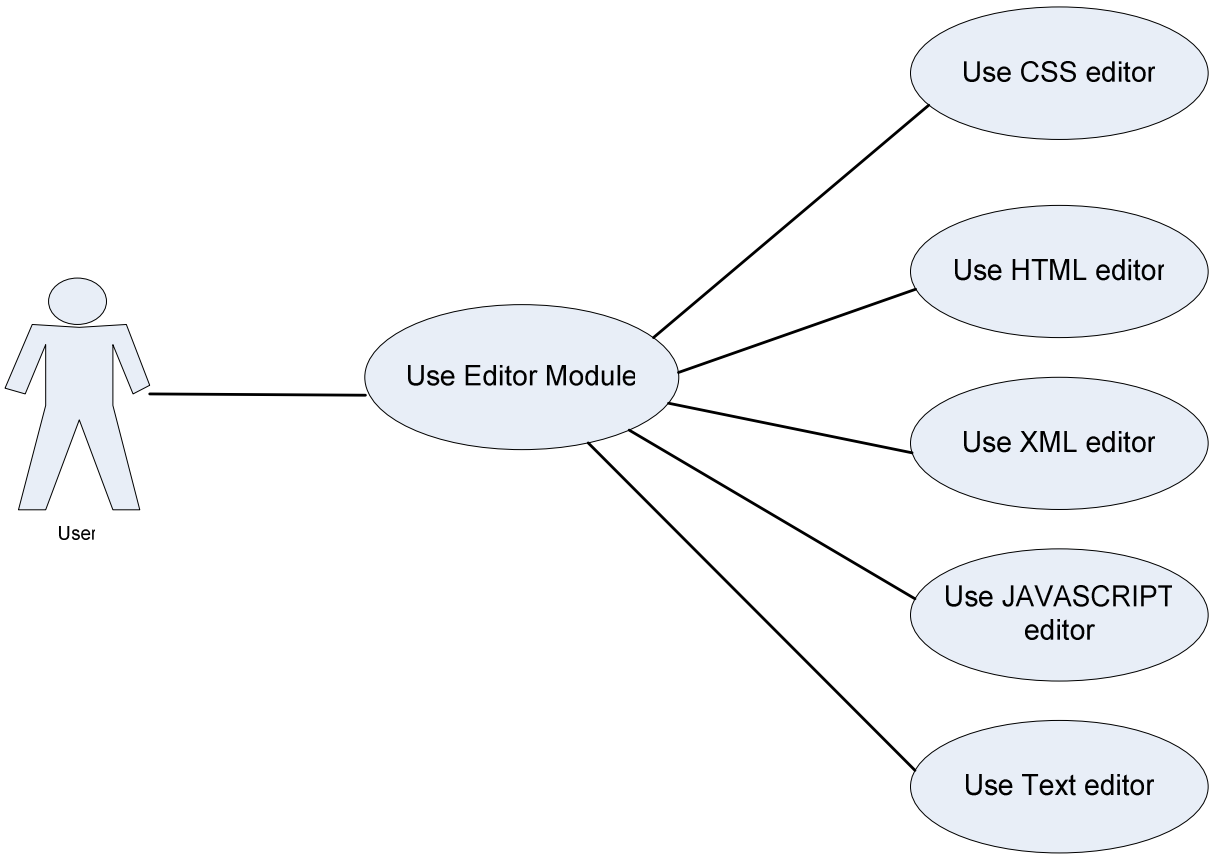
In deployment process user can set ftp connections to the web server using our Remote Connection Module easily. Like Database Connection Module the connection wizard asks user some information like, server host address, port number, username and password to connect the server. After setting the connection using File Explorer Panel user can access the files in the server. Adding new files, deleting files and editing files can be done in our software. There is synchronization utility in our software which synchronizes the files between local project folder and server side folder.

5.6 CVS MODULE

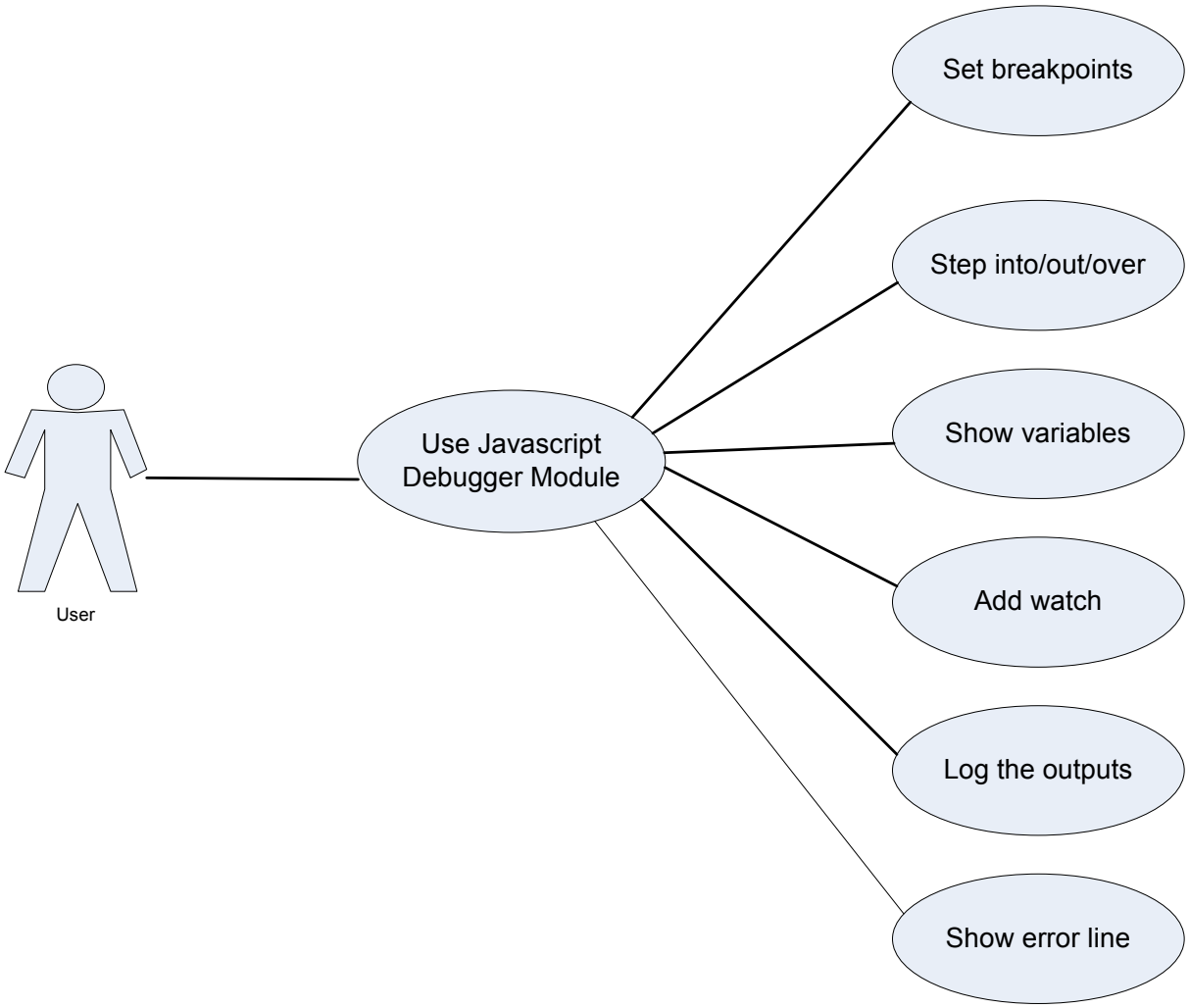
CVS module is much likes to Remote Connection Module. Connection wizard helps again to connect the CVS host. After connecting the host the files can be upload or downloaded using CVS Module.

6. USE CASE DIAGRAMS

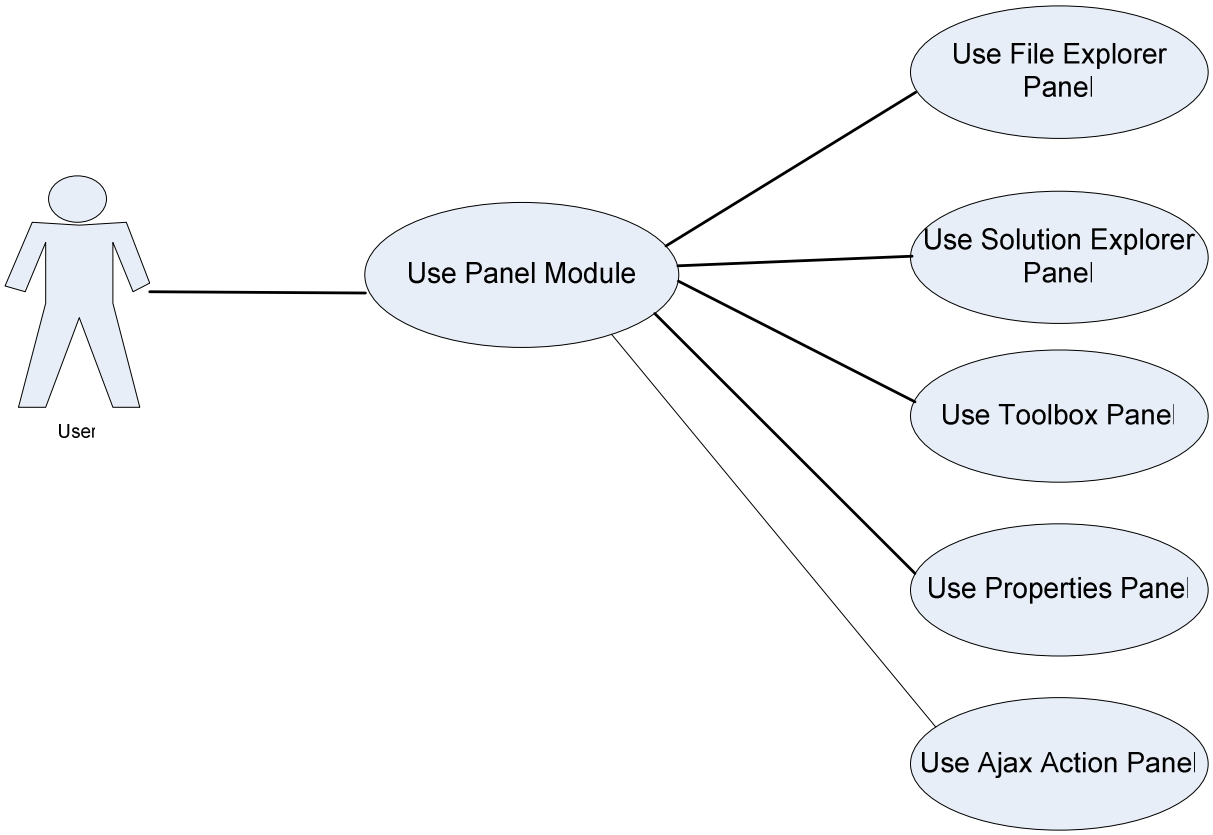
6.1 EDITOR MODULE



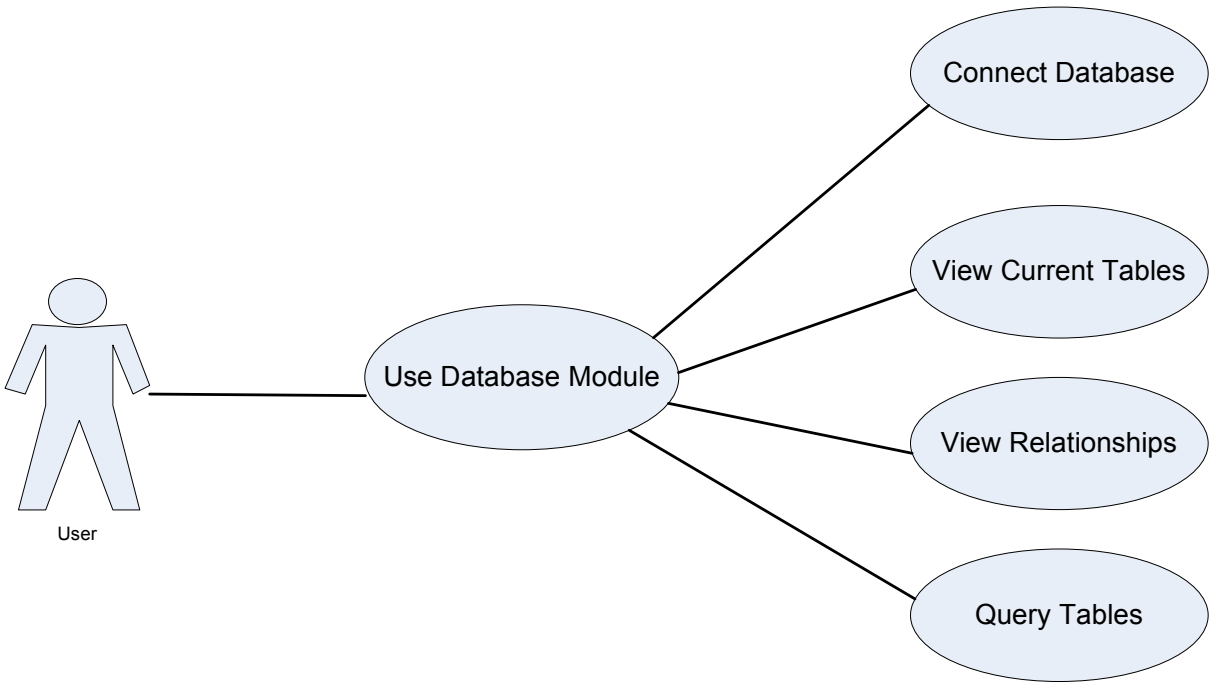
6.2 JAVASCRIPT DEBUGGER MODULE



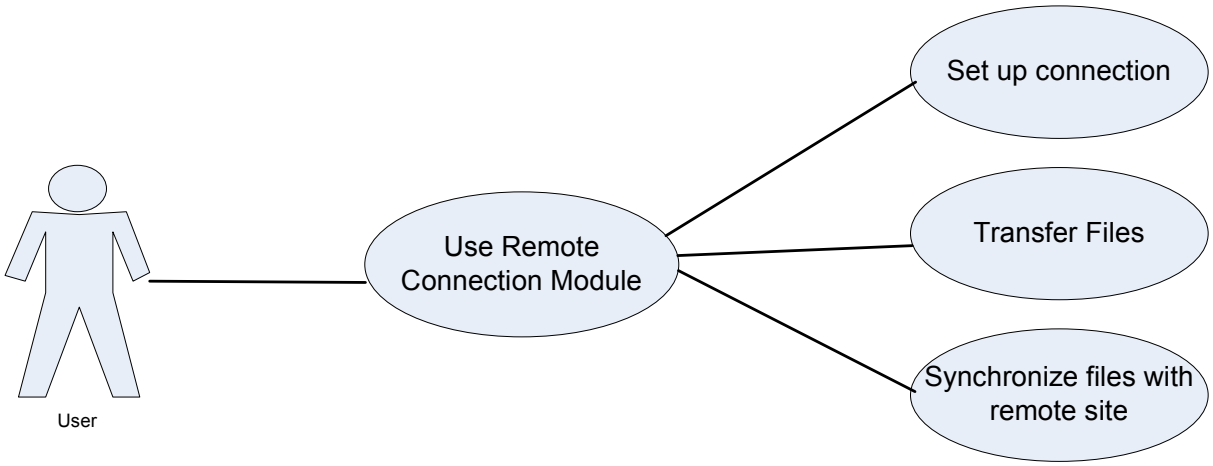
6.3 PANEL MODULE



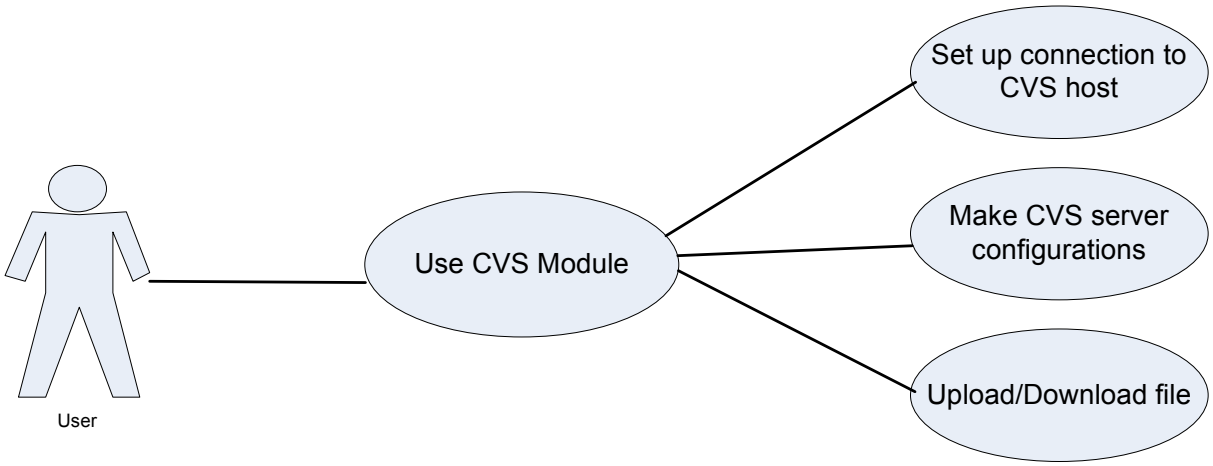
6.4 DATABASE MODULE



6.5 REMOTE CONNECTION MODULE



6.6 CVS MODULE



7. REFERENCES

7.1 MARKET RESEARCH REFERENCE

[1] www.aptana.com
[2] www.tibco.com

7.2 GANNT CHART

ID	Task Name	Start	Finish	Duration
1	Information Gathering	9/26/2006	10/3/2006	8d
2	Proposal Report	10/9/2006	10/9/2006	0d
3	Studying AJAX Tutorials	10/10/2006	10/18/2006	9d
4	Project Web Site Design	10/18/2006	11/4/2006	18d
5	Market Research	10/18/2006	10/29/2006	12d
6	User Survey	10/26/2006	11/1/2006	7d
7	Data Modeling	10/27/2006	10/28/2006	2d
8	Functional Modeling	10/29/2006	10/31/2006	3d
9	Behavioral Modeling	11/1/2006	11/2/2006	2d
10	Use Case Modeling	11/2/2006	11/3/2006	2d
11	Data Dictionary	11/3/2006	11/3/2006	1d
12	Requirement Analysis Report	11/6/2006	11/6/2006	0d
13	Task Assignment	11/7/2006	11/7/2006	1d
14	Architectural Design	11/13/2006	11/17/2006	5d
15	User Interface Design	11/18/2006	11/21/2006	4d
16	Component Level Design	11/22/2006	11/26/2006	5d
17	Initial Design Report	11/28/2006	11/28/2006	0d
18	Prototype Implementation	11/30/2006	1/8/2007	40d
19	Detailed Architectural Design	12/3/2006	12/17/2006	15d
20	Detailed User Interface Design	12/18/2006	12/30/2006	13d
21	Detailed Component Level Design	12/31/2006	1/9/2007	10d
22	Final design report	1/15/2007	1/15/2007	0d
23	Preparing for Demo	1/16/2007	1/20/2007	5d
24	Prototype demo	1/23/2007	1/23/2007	0d