



METU

Computer Engineering

AKAMAI WEB TOOLKIT

ANALYSIS REPORT



Members of the Team:

- Ahmet Emin Tosun → e141801@metu.edu.tr
- Uğur Can Tekin → e134800@metu.edu.tr
- Hasan İşler → e134758@metu.edu.tr
- Vedat Şengül → e129829@metu.edu.tr
- Muhammet Yavuz Aşık → e134713@metu.edu.tr



Contents

1. PROJECT DEFINITION, SCOPE AND GOALS
 - 1.1 Project Definition
 - 1.2 Project Scope and Goals
2. PROCESS
 - 2.1 Team Organization
 - 2.2 Process Model
 - 2.3 Major Constraints
 - 2.3.1 Project Schedule
 - 2.3.1 Language Constraints
 - 2.3.3 User Interface
 - 2.4 Gantt Chart
3. MARKET RESEARCH
 - 3.1 CURRENT PRODUCTS
 - 3.1.1 APTANA
 - 3.1.1.1 What is Aptana?
 - 3.1.1.2 Main Features of Aptana
 - 3.1.1.3 About the Aptana Editors
 - 3.1.1.3.1 JavaScript Editor
 - 3.1.1.3.2 HTML Editor
 - 3.1.1.3.3 CSS Editor
 - 3.1.1.4 Screenshots
 - 3.1.2 AJAX JOYISTAR WEBSHOP
 - 3.1.2.1 What is Ajax Joyistar Webshop?
 - 3.1.2.2 Main Features of Joyistar Webshop
 - 3.1.2.3 Screenshots
 - 3.1.3 ZAPATEC
 - 3.1.3.1 What is Zapatec
 - 3.1.3.2 Main Features of Zapatec
 - 3.1.4 GOOGLE WEB TOOLKIT
 - 3.1.4.1 What is Google Web Toolkit?
 - 3.1.4.2 Main Features of GWT
 - 3.1.4.3 Google Web Toolkit Components
 - 3.1.5 DOJO TOOLKIT
 - 3.1.5.1 What is Dojo?
 - 3.1.5.2 Main Features of Dojo



3.1.6 MORFIK WEBOS APPSBUILDER

3.1.6.1 What is Morfik WebOS AppsBuilder

3.1.6.2 Main Features of Morfik WebOS AppsBuilder

3.1.6.3 About the Morfik Editor

3.1.6.4 Screenshots

3.1.7 Comparison Table

3.2 Questionnaire

3.3 Interview

4. PROJECT REQUIREMENT

4.1 System Requirement

4.1.1 Hardware Requirement

4.1.2 Software Requirement

4.2 User Requirement

4.2.1 Use Case Diagram

4.2.2 User Functionality

4.3 Functional Requirement

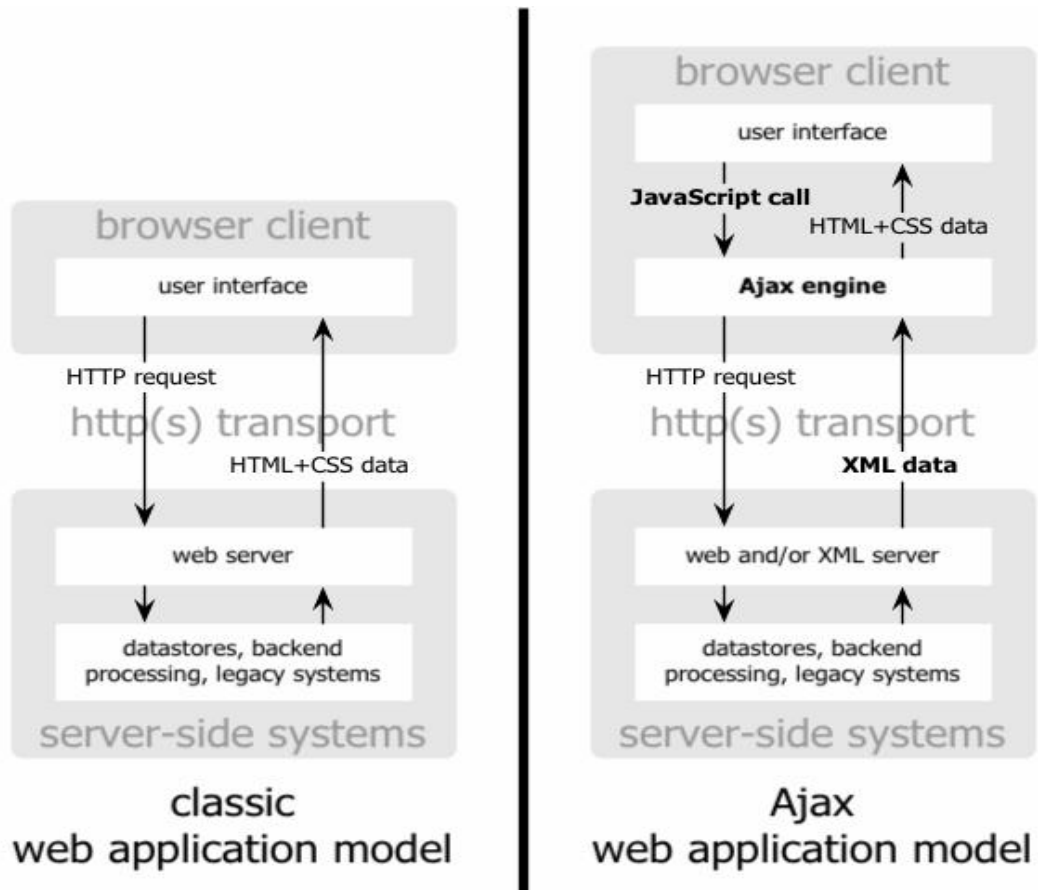
4.3.1 Menu Requirement



1. PROJECT DEFINITION, SCOPE AND GOALS

1.1 Project Definition

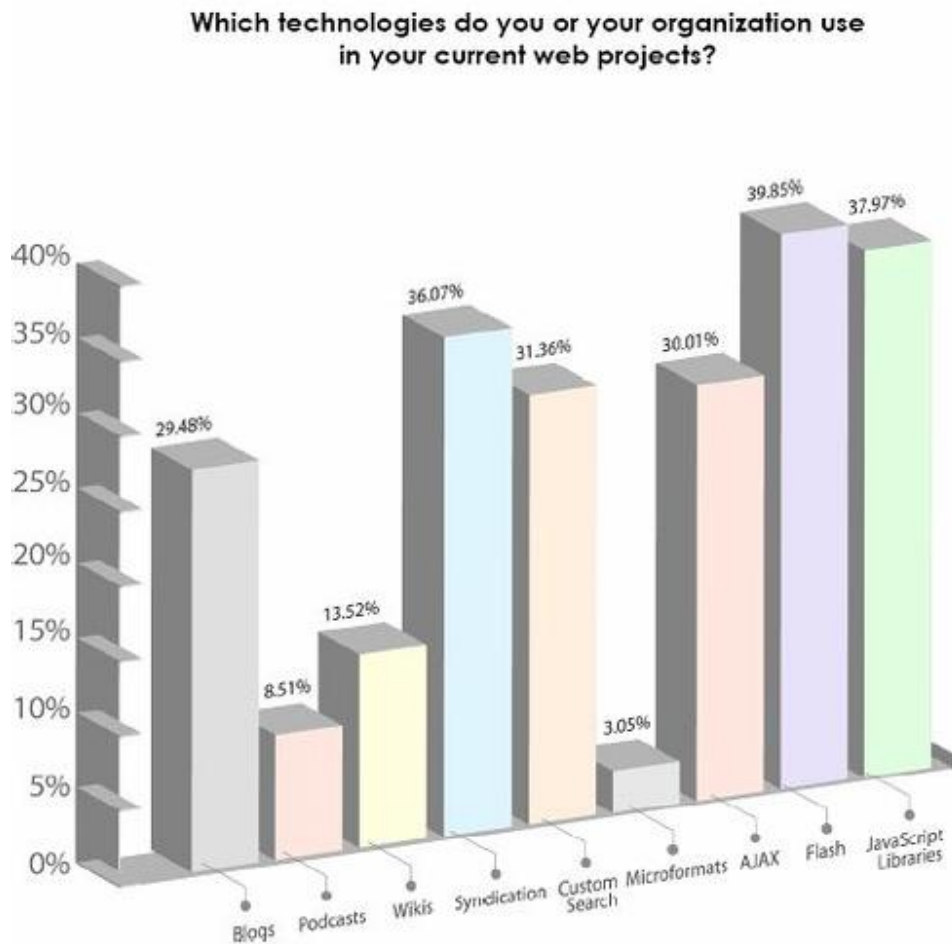
AJAX stands for Asynchronous JavaScript and XML and is a web development technique for creating interactive web applications. Ajax's most appealing characteristic is its asynchronous nature, which means it can do all of this without having to refresh the page. This allows you to update portions of a page based upon user events. With AJAX, we only get the data from the server that we absolutely need, not the whole page. More importantly, data can be posted to and retrieved from the server after the entire page is loaded. This can be leveraged in creative and powerful ways to create a more fluid browsing experience. It can send as well as receive information in a variety of formats, including XML, HTML, and even text files. The technologies that are used to build AJAX web applications encompass a number of different programming domains, so AJAX development is neither as straightforward as regular applications development, nor as easy as old web development.





From above modeling picture we can easily understand the difference between the classic web application model and the Ajax web application model. In the Ajax model the user do not contact directly with the server. The Ajax engine handles it. AJAX development is such an enormous leap forward for web development; instead of having to send everything to the server in a single, huge mass, then wait for the server to send back a new page for rendering, web developers can communicate with the server in smaller chunks, and selectively update specific areas of the page based on the server's responses to those requests.

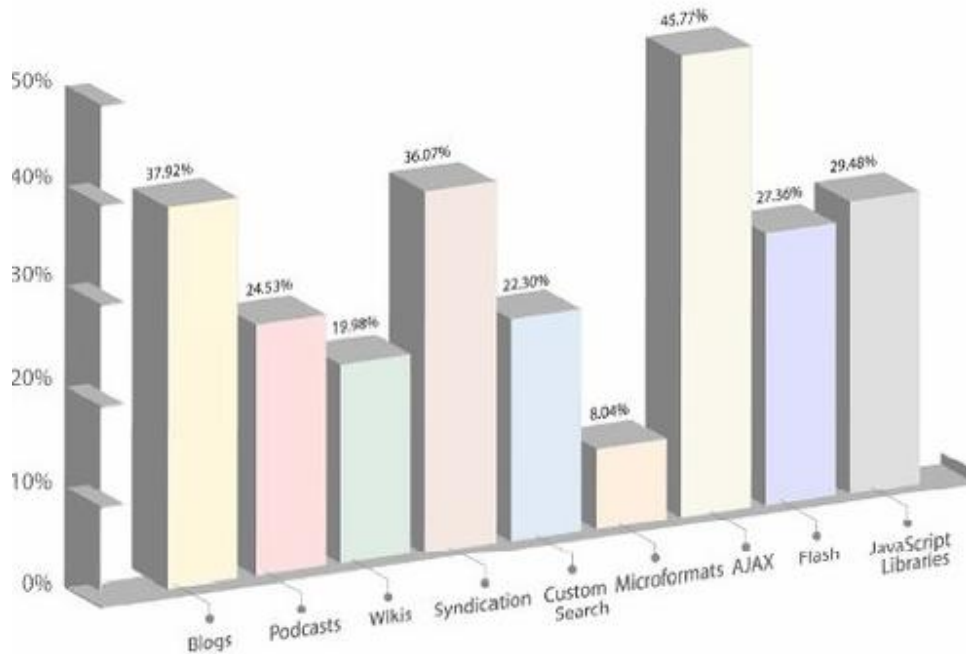
Nowadays Ajax is one of the most popular web development technologies. We can see this from the survey results. As you can see %30 of web projects are done by AJAX in these days.



But Ajax will become the most popular web development technology during the next twelve months. We can see this from the below survey results. As you can see %45 of web projects will be done by AJAX in the next twelve months.



Which technologies are you or your organization planning to use in your future web projects (next 12 months)?



These results are taken from;

http://www.readwriteweb.com/archives/the_state_of_web_development.php and done on 5000 web developers in USA.

And as Akamai Systems we are going to implement a graphical development environment for web pages with Ajax. The users of our product will be able to write new scripts and also edit them within the HTML files. The program will let users to modify web components. Users will be able to reach to a specific database, and retrieve the necessary information for the web page. Also we are planning to add a query wizard that will be used for creating queries without writing any code. We will expand the number of ready to use components as much as we can.

1.2 Project Scope and Goals

In the project we want to serve the Ajax users a user friendly Ajax development environment. We will develop a system that will collect beneficial parts of current Ajax IDEs on the market and other similar developing studios.



Project scope;



- To provide user a graphical development environment for web pages with Ajax
- To provide user a usable interface
- To provide user the opportunity of writing new scripts, editing, deleting them.
- To provide user database connection
- To provide user the opportunity of implementing the server-side actions in order to have the property of executing the written scripts
- To provide user drag and drop ready to use components
- To provide user error checking
- To provide user code editing

Main goals of the Akamai Web Toolkit (AWT) are;

- Usability: AWT will be useful program for all type of web developers. And our program will support all browsers. (Firefox, Internet Explorer, Safari, Opera...)
- Performance: AWT will run and compile fast enough to meet the developer's needs.
- Interface: AWT's interface will be a user friendly one, so users can easily find what they need in the program.
- Developable: We choose component based programming as a process model. Therefore, new features can be added.

2. THE PROCESS

2.1 Team Organization

In software engineering process, team structure has a vital importance. Since we are all relaxed and easy-going persons, we choose a team leader for coordination and interaction between our group members. This structure exactly fits into Democratic Centralized (DC) model. Decisions are taken by all project members and final approval is given by team leader.

2.2 Process Model

While we are choosing our project team organization, we planned to move in a direction called Component-Based Development which is spiral model variation in which applications are built from prepackaged software components called classes. Since our project has many components that can be coded separately and found in open source libraries this model best fits. Object oriented approach is a must in developing our project so we are focusing on creating components and packages.



2.3 Major Constraints

2.3.1 Project Schedule

We have to finish the project until end of May 2007 so this is one of the main major constraints. We have to design, build and test the project during this period. In this report, there is a project schedule for this term that includes first prototype.

2.3.2 Language constraints

We are going to implement our project in Java. This choice helps us while designing interface. Also most of the open source libraries are written in Java. We thought that we can find more sources with this language.

2.3.3 User Interface

User friendly menus will be created. Easy to use and simple menus will be taken into consideration. We will work on interface for making easy to access to all the components. We believe that our program will be one step ahead from similar programs with this feature.

2.4 Gantt Chart



3. MARKET RESEARCH

Market observation is one of the key concepts in order to understand what the system is. With this aim in mind, we made literature survey and arranged meetings with the customers. We have searched the existing programs having similar features with our system in order to improve our knowledge about the current marketplace. With the help of the survey, we have determined the main properties of our application and we have also added some new features on it. In addition, literature survey showed us the missing points in the area. There are wide ranges of features implemented in different software systems and also there are many programs that support most of the features in the market. A comparative way of thinking enabled us to recognize most necessary features. Interview that we made helped us at this point.

3.1 CURRENT PRODUCTS

3.1.1 APTANA

3.1.1.1 What is Aptana:

Aptana is a IDE for building dynamic web applications and designed for web developers who are using (X)HTML, CSS and JavaScript. A brief overview of the Aptana IDE for JavaScript, CSS and HTML and how Aptana can make AJAX development simpler is available from www.apтана.tv.

3.1.1.2 Main Features of Aptana:

Aptana includes the following features:

- Code Assist on JavaScript, HTML, and CSS languages, including your own JavaScript functions.
- Error and warning notification for your code.
- Ability to extend the IDE via JavaScript with new 'Actions' and new 'Views'.
- Cross-platform support
- Ability to create and edit JS, CSS, HTML, and XML files.
- Aptana Editors with other file types, such as PHP, by configuring support for those file types.
- Support for Aptana UI customization and extensions
- Also it can be install Aptana into Eclipse as a plugin but if you want to use Aptana as an Eclipse plugin all Aptana features should be functional.



- Aptana works with AFLAX, Dojo, MochiKit, Prototype, Rico, Script.aculo.us, Yahoo UI.

3.1.1.3 About the Aptana Editors

3.1.1.3.1 JavaScript Editor

The Aptana JavaScript Editor contains many features to make coding JavaScript easier. These include:

- Code Assist helps you code faster and more accurately.
- Colorization provides visual cues for Syntax.
- The Open Declaration feature which allows you to instantly jump to a function in your code.

The image below shows a documented JavaScript file open in the Aptana JavaScript Editor.

```
31 }
32
33 /**
34  * Function that actually attaches the styles to the step elements.
35  * @param (Object) liElement  Element to attach the style to.
36  * @param (Object) isEven    If true, indicates that the step number is even.
37  */
38 function attachStepStyles(liElement, isEven)
39 {
40     liElement.onclick = toggleStep;
41     liElement.className = "wtStepCollapsed";
42     liElement.onmouseover = mouseOn;
43     liElement.onmouseout = mouseOff;
44
45     var textNode = liElement.firstChild;
46
47     if (textNode.nodeType == 3)
48     {
49         var spanNode = document.createElement("span");
50         liElement.insertBefore(spanNode, textNode);
51         spanNode.appendChild(textNode);
52         spanNode.className = "wtStepText";
53         if (isEven)
54             addStyleToElement(spanNode, "wtStepTextEven");
55     }
56
57     else
58     {
59         alert("Wrong node type!");
60     }
61 }
62
63 /**
64  * Gets the next sibling element in a list of walkthrough elements.
65  * @param (Object) element  Element to get the sibling of.
66  * @param (String) tagName  Tag name of the sibling to get.
67  */
68 function getNextSiblingElementByTagName(element, tagName)
69 {
70     alert("nextSibling didn't match: " + element.nodeName);
71     var nextSibling = element.nextSibling;
72 }
```



3.1.1.3.2 HTML Editor

Use Aptana's HTML Editor to code your HTML pages. The Editor includes the following features:

- Colorization provides visual cues.
- Assisted tag completion helps keep your HTML valid.
- A Preview tab allows you to quickly preview the web page currently open in the HTML Editor.
- The HTML Editor also incorporates several JavaScript-focused features of the Aptana JavaScript Editor if you include JavaScript in your web pages.

The image below shows an HTML file open in the Aptana HTML Editor.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/st^
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
5     <title>Untitled Document</title>
6     <link href="style.css" rel="stylesheet" type="text/css" />
7     <script type="text/javascript" src="validation.js"></script>
8
9   </head>
10  <body onload="focusFirst();"
11
12  <h1>Welcome to the House of Nachos!</h1>
13
14  <form action="thankyou.htm" name="orderform" id="orderform" method="get"
15    onsubmit="return validate();"
16    onreset="return confirm('Clear all fields and reset form?');" >
17
18  <h2>Products</h2>
19
20  <p>Select your products and specify order quantities:</p>
21
22  <table cellspacing="2" cellpadding="2" border="0" summary="">
23
24  <tr>
25    <td width="120" class="heading">Product</td>
26    <td width="130" class="heading">Quantity</td>
27    <td width="75" class="heading">Cost</td>
28  </tr>
```



3.1.1.3.3 CSS Editor

Aptana CSS Editor is useful for adding styles for your web sites.

The image below shows a CSS file open in the CSS Editor.

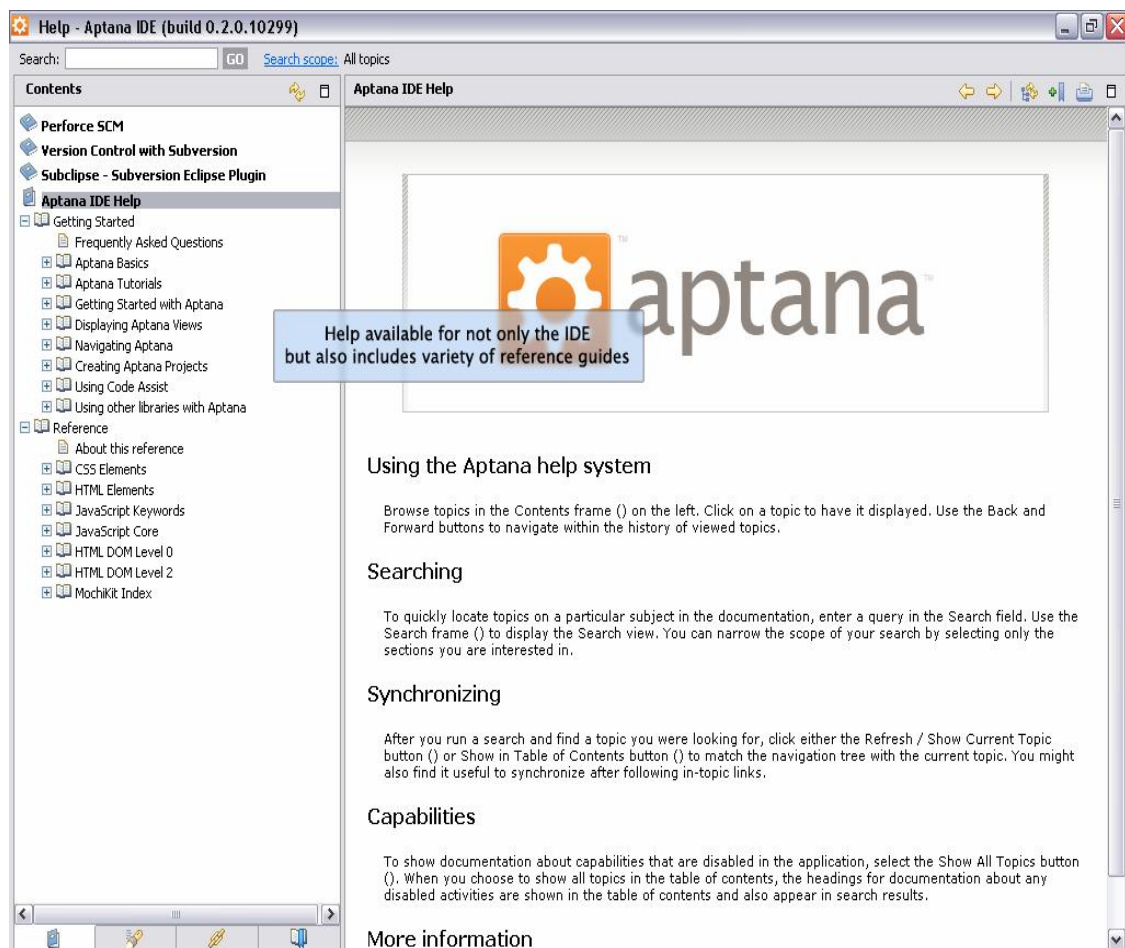


```
23 )
24
25 td {
26     font-family: Arial, Helvetica, sans-serif;
27     font-size: 12px;
28
29 }
30
31 .lastrow {
32     border-bottom-style: solid;
33     border-bottom-color: #000066;
34     border-bottom-width: thin;
35
36 }
37
38 h2 {
39     font-family: Arial, Helvetica, sans-serif;
40     font-size: 14px;
```



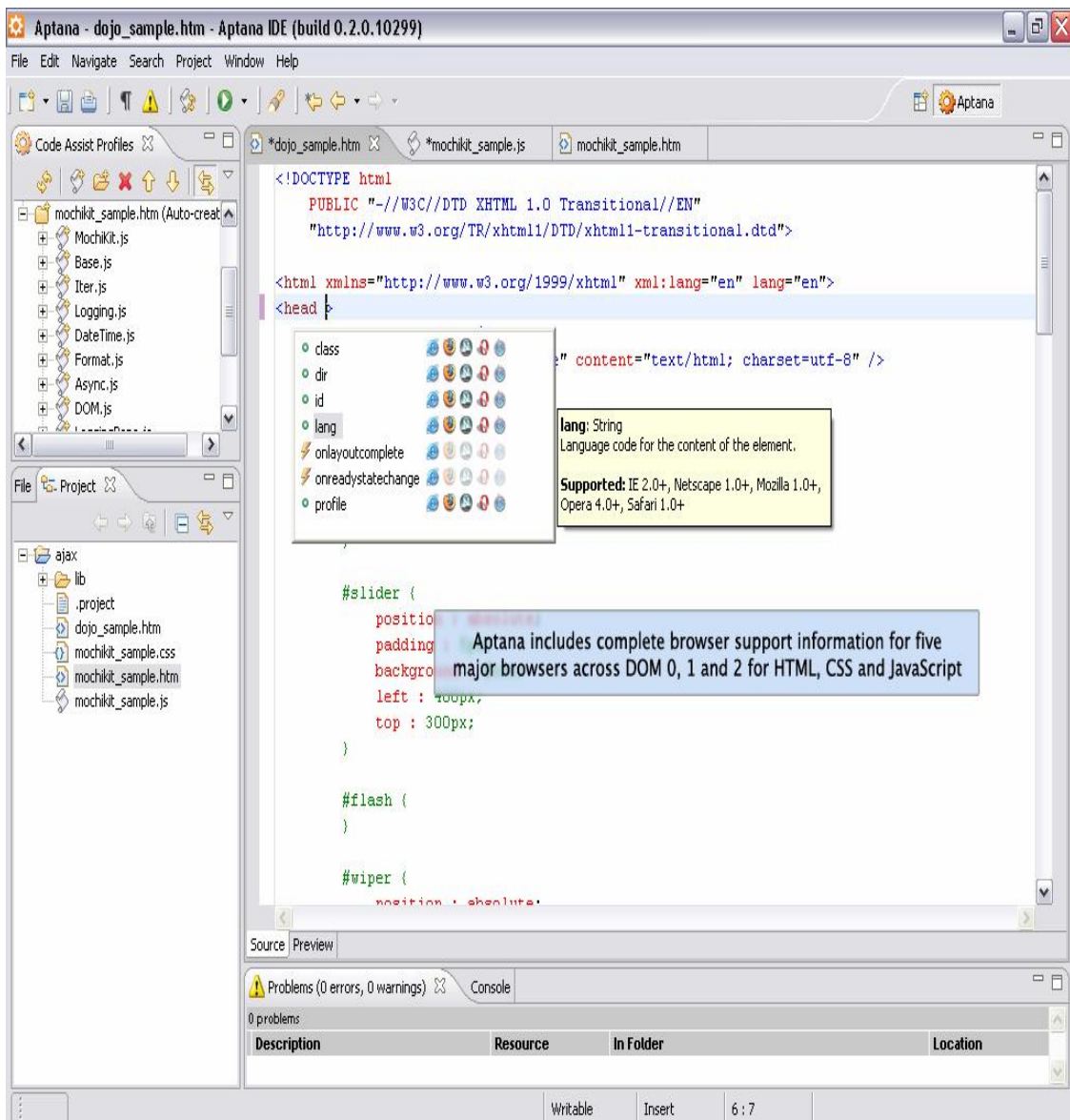
3.1.1.4 Screenshots:

This Help menu is available for IDE of Aptana and also it includes a variety of reference guides.





While writing html code, developer can see the supported browsers for selected tag.





Outline property analyze the code and give an outline for CSS, JavaScript and HTML

The screenshot displays the Aptana IDE interface for a file named 'dojo_sample.htm'. The main editor shows the source code of an HTML document with embedded CSS and JavaScript. Three blue callout boxes are overlaid on the code to highlight specific sections:

- CSS Outline:** Points to the CSS rules for #fadeElem, #slider, #flash, #wiper, and #wiper2.
- JavaScript Outline:** Points to the JavaScript code blocks, including dojo.js and custom scripts like testExplode and wowwi.
- HTML Outline:** Points to the HTML structure, including the head section with title, meta, and style tags, and the body section with h3, a, and div elements.

The right-hand side of the IDE features an 'Outline' panel that provides a hierarchical tree view of the document's structure, corresponding to the HTML Outline callout. The bottom of the IDE shows a 'Problems' panel with 0 errors and 0 warnings, and a 'Console' panel.



3.1.2 AJAX JOYISTAR WEBSHOP

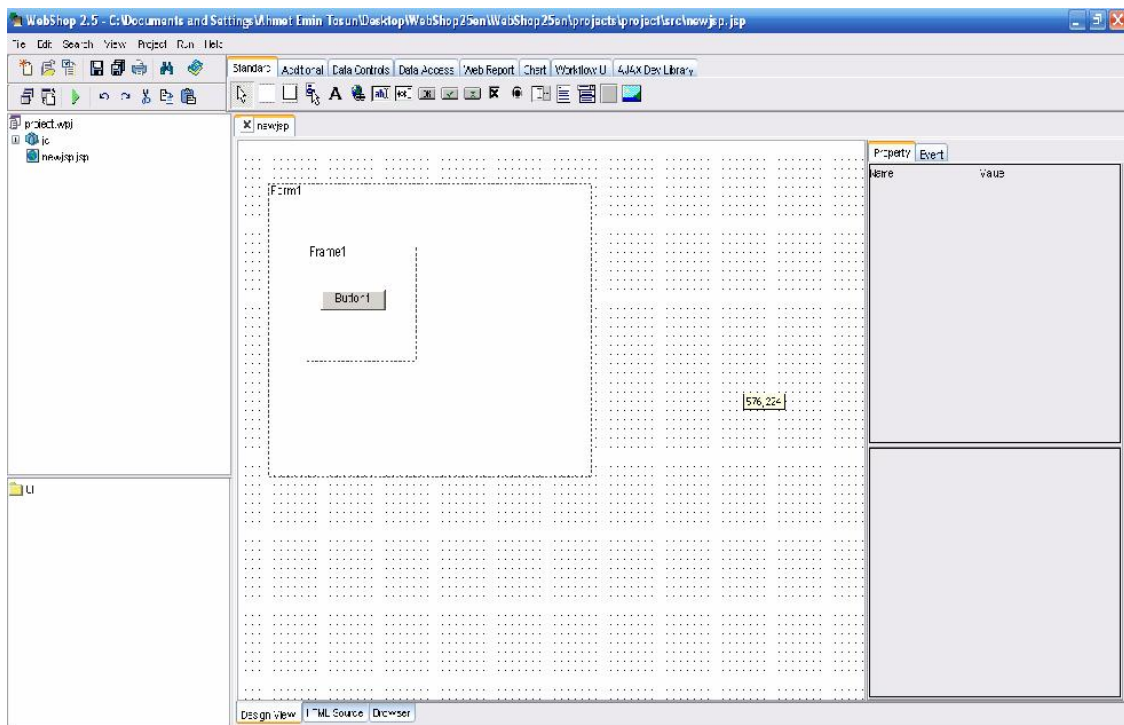
3.1.2.1 What is Ajax Joyistar Webshop:

The JoyiStar WebShop is the Ajax web development. It provides an efficient OOP developing method to construct stable and convenient rich web applications based AJAX component.

3.1.2.2 Main Features of Joyistar Webshop:

- Support Java and JSP
- Cross Platform- support all major browsers at present.
- Standards Based (Standards of HTML, XML etc.)
- User Friendly Interface
- OOP Methodology
- Develops business-critical web applications and it offers optimizing power of AJAX technology.
- Provides many web components to attain powerful functions such as dynamically generate master-details, user interfaces, web reports.
- Integration - integrates with existing server-side codes and services.
- Pull-drag developing - program in the mode of your choice.

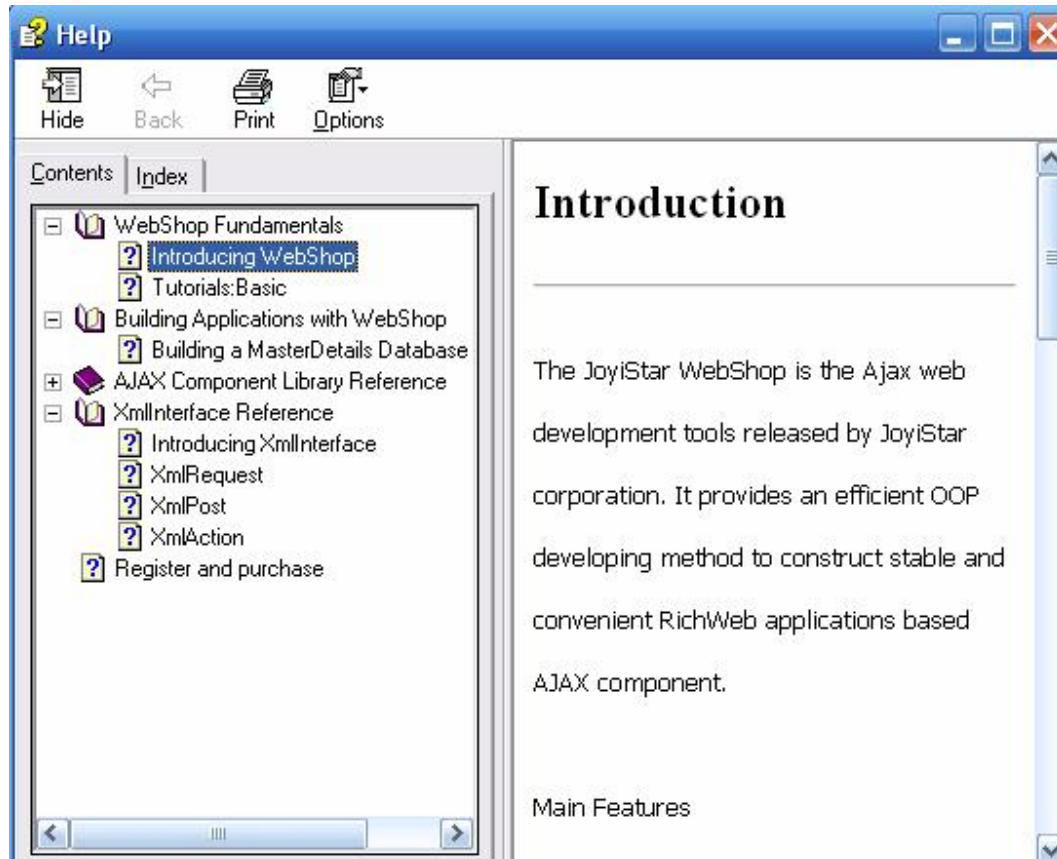
Below, there is a picture shows the main window of Ajax Joyistar WebShop.





3.1.2.3 Screenshots:

This Help menu is available for IDE of Ajax Joyistar Webshop and it includes basic tutorial and Ajax component library references and also XML Interface reference.





3.1.3 ZAPATEC

3.1.3.1 What is Zapatec:

Zapatec is another Ajax web development tool which also can act like ASP, can host a database and web application environment.

3.1.3.2 Main Features of Zapatec:

They are listed below;

- Cross Browser Compatible
- Standards Based
- CSS Based Themes
- Widgets work independently or with each other
- Platform Areas (.NET, JSP, PHP, ASP, Perl, Python, Ruby)

3.1.4 GOOGLE WEB TOOLKIT

3.1.4.1 What is Google Web Toolkit:

Google Web Toolkit (GWT) is a Java software development framework that is for writing AJAX applications. With GWT, you can develop and debug AJAX applications in the Java language using the Java development tools of your choice

3.1.4.2 Main Features of GWT

- Dynamic, reusable UI Components
- Simple Remote Procedure Call (RPC)
- Browser History Management
- Full-featured Java Debugging
- Browser Compatible
- JUnit Integration for testing
- Interoperability

3.1.4.3 Google Web Toolkit Components

- GWT Java-to-JavaScript Compiler
- GWT Hosted Web Browser
- JRE emulation library
- GWT Web UI class library



3.1.5 DOJO TOOLKIT

3.1.5.1 What is Dojo:

The Dojo Toolkit is a set of JavaScript libraries that help you build dynamic web applications.

3.1.5.2 Main Features of Dojo

The Dojo Toolkit includes libraries in the following JavaScript development areas:

- JavaScript core libraries to help manipulate the DOM, HTML, and CSS
 - [dojo.lang](#): Utility routines to make JavaScript easier to use.
 - [dojo.string](#): String manipulation routines.
 - [dojo.dom](#): DOM manipulation routines.
 - [dojo.style](#): CSS Style manipulation routines.
 - [dojo.html](#): HTML specific operations
- Data Structures
- Web I/O
- Visual web development, such as animation and drag-and-drop
- Math and cryptography
- Widgets
 - Core widgets
 - Layout widgets
 - SVG widgets

3.1.6 MORFIK WEBOS APPSBUILDER

3.1.6.1 What is Morfik WebOS AppsBuilder:

The WebOS AppsBuilder is a specialized IDE for building web-based applications. WebOS AppsBuilder brings simplicity to the design of AJAX applications without compromising their functionality. Program makes easier to write HTML, XHTML, XML, JavaScript and CSS. It has ready to use components.

3.1.6.2 Main Features of Morfik WebOS AppsBuilder:

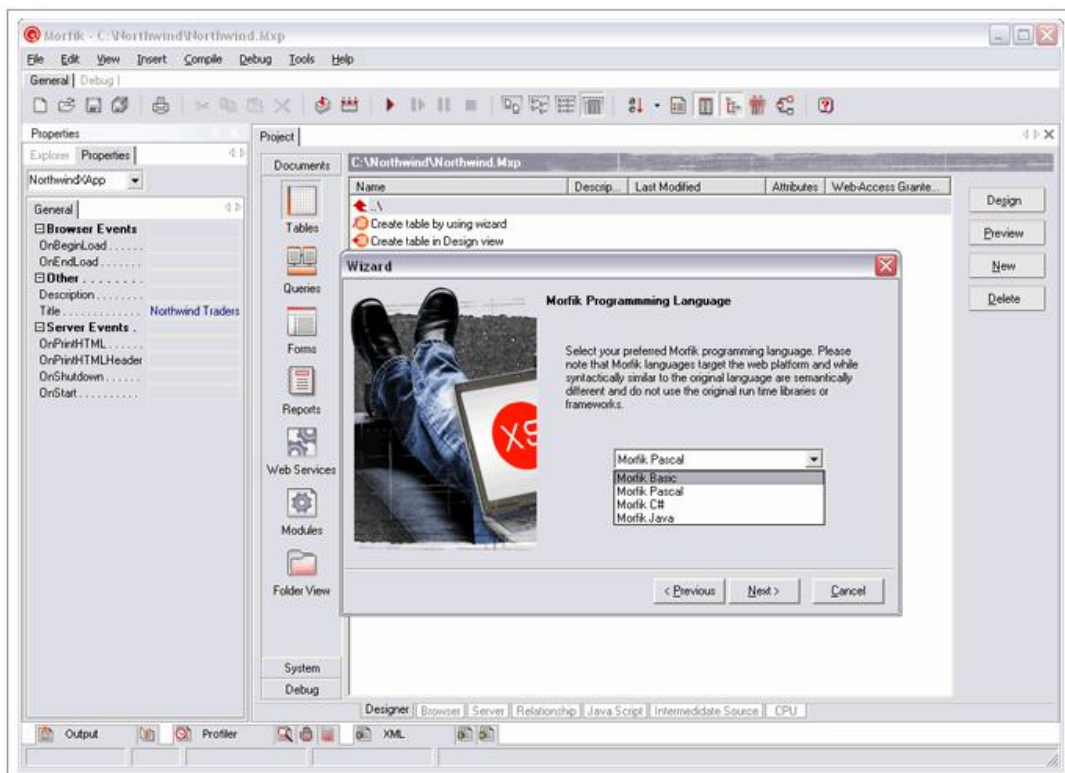
- Visual Designer for creating database stuffs
- Support for high level languages (Basic, C#, Java and Object Pascal)



- Morfik Compiler and JavaScript Synthesis Technology for debugging
- Easy to deal with relational database
- Easy to adjust web server
- User Friendly Interface

3.1.6.3 Screenshots:

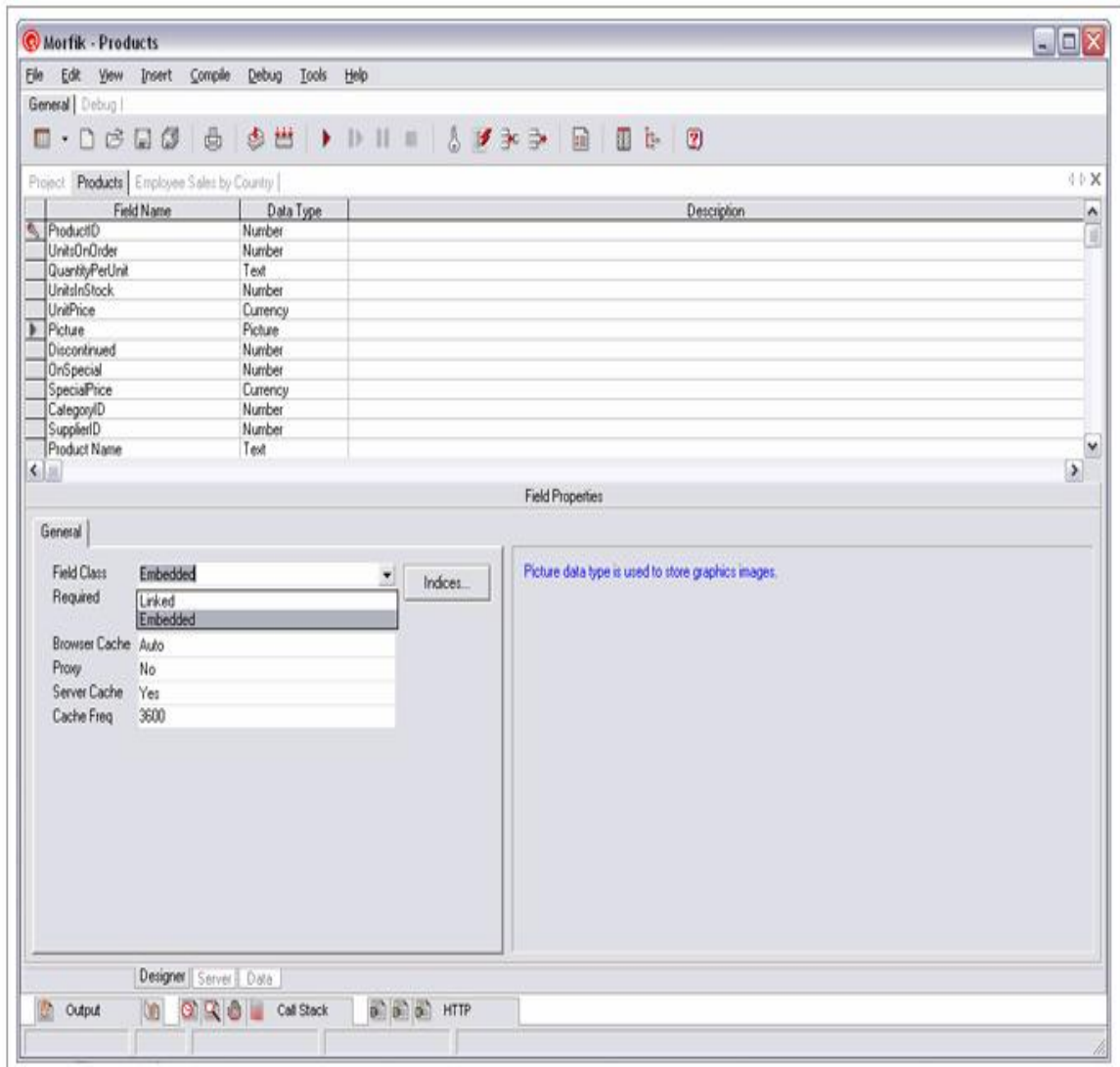
Below, there is a picture shows the main page and creating project page.



EASY TO USE VISUAL DESIGN ENVIRONMENT



Below, there is a picture shows how to design a table in Morfik WebOS AppsBuilder.





Below, there is a picture shows how to view and search data in Morfik WebOS AppsBuilder.

The screenshot displays the Morfik - Products application interface. At the top, there is a menu bar with options: File, Edit, View, Compile, Debug, Tools, Help. Below the menu bar is a toolbar with various icons. The main area shows a data table with columns: Product Name, ProductID, UnitsOnOrder, QuantityPerUnit, UnitsInStock, UnitPrice, SpecialPrice, OnSpecial, Picture, Discontinued, CategoryID, and SupplierID. A Filter dialog box is open over the table, showing a list of fields to search and search options like 'Exact Match', 'Partial Match at Beginning', and 'Partial Match Anywhere'. The table data is as follows:

Product Name	ProductID	UnitsOnOrder	QuantityPerUnit	UnitsInStock	UnitPrice	SpecialPrice	OnSpecial	Picture	Discontinued	CategoryID	SupplierID
Balsamic Vinegar	1	0		23	\$20.00	\$0.00	0	(PICTURE)	0	1	1
Olive Oil	2	0		12	\$19.00	\$0.00	0	(PICTURE)	0	1	1
Corn Oil	3	0		3	\$10.00	\$12.00	1	(PICTURE)	0	1	3
Canola Oil	4	0		4	\$22.00	\$0.00	0	(PICTURE)	0	1	1
Organic Olive Oil	6	0		3	\$25.00	\$0.00	0	(PICTURE)	0	1	2
Terabianca Olive Oil	7	0		23	\$30.00	\$0.00	0	(PICTURE)	0	1	2
Peanut Oil	8	0		56	\$40.00	\$0.00	0	(PICTURE)	0	1	3
White Wine Vinegar	9	0		76	\$97.00	\$20.00	1	(PICTURE)	0	1	3
White Vinegar	10	0							0	1	3
Baba Ganoush	11	0							0	2	4
Bengal Hot Chutney	12	0							0	2	4
Black Pepper	13	0							0	2	5
Horseradish Powder	14	0							0	2	5
German Mustard	15	0							0	2	6
Hummus Tahini	16	0							0	2	6
Pickled Ginger	17	0							0	2	6
Sea Salt	18	0							0	2	7
Yeast Extract	19	0							0	2	7
Vegetable Stock	20	0							0	2	7
Chocolate	21	0							0	3	8
Caramelle Coffee	22	0							0	3	8
Clear Mint Candy	24	0							0	3	9
Mars Bars	25	0							0	3	9
Marzipan	26	0							0	3	9
Musk Sticks	27	0		23	\$43.90	\$6.00	1	(PICTURE)	0	3	10
Snakes	28	0		90	\$45.60	\$0.00	0	(PICTURE)	0	3	11
Rice Candy	29	0		98	\$123.79	\$10.00	1	(PICTURE)	0	3	11
Cheestiks	32	0		2	\$32.00	\$0.00	0	(PICTURE)	0	4	13



3.1.7 Comparison Table

PROPERTIES							
IDE	APTANA	JOYISTAR WEBSHOP	ZAPATEC	GOOGLE WEB TOOLKIT	DOJO TOOLKIT	MORFIK	AWT
TYPE	Eclipse Plug-in & Custom IDE	Custom IDE	IDE	Command Line	IDE	Custom IDE	IDE
FRAMEWORK	Multiple	JoyiStar	Zapatec	GWT	Dojo JavaScript Toolkit	Morfiks	AWT
LANGUAGE	JavaScript/HTML/ CSS	Java/JSP	Java	Java	Java	Pascal, Java, C#, VB	JavaScript
DB Connection	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Drag & Drop	Yes	Yes	No	No	No	Yes	Yes
HTML Edit	Yes	Yes	Yes	Yes	Yes	Yes	Yes
LICENSE	Open Source	Commercial	Commercial	Free	Free	Commercial	Free
COMMENT	User can write and edit code on HTML, CSS and Java Script Editor. Aptana works with AFLAX, DOJO, MochiKit, Prototype, Rico...	Supports Java and JSP. Cross platform. Simple and secure, Pull and drag developing.	The Suite includes the following components: Calendar, Menu, Tree, Grid, Forms, Windows, Effects, Drag and Drop, Slider, Time and Transport.	Dynamic reusable UI components, java debugging, browser history management	proper introspection, autosuggest, code navigation, integrated help, and so on	Support several source languages including Pascal, Java, C# and VB. Drag-and-drop. For example, right click doesn't give you the ability to cut and paste, etc... Easy to adjust web server	Free to use for personal and Commercial purposes. We We want to take the custom good points of the other IDEs.



3.2 Questionnaire

We made a small questionnaire with 24 people in department of computer engineering and web programming business. There are nine questions in the questionnaire. Our commands about the given answers in the following:

1- Which browsers should be supported?

Comment: Users told that browser compatibility is the most important feature that the program should have. They think that a web page developed with the product should run in all browsers. Therefore, our program will have cross browser feature.

2- Which technologies should be supported?

Comment: Users expectations from the program is to support ASP, PHP and JSP. Since there are many such technologies, we are planning to support these three in our program. But at the end of the term, we can add one or two of others if we have time. We give priority to these three technologies.

3- Which databases should be supported?

Comment: Users pay attention on this subject. They said that program should support as many as it can. We are planning to begin with access, then ms sql and finally mysql. We leave oracle as future work. At the end of first term, we will implement the access support for the first release.

4- Do you need HTML editor?

Comment: Users said “yes” to this question. Since the program is a web development tool, HTML editor is a must. We will try to develop a HTML editor and add as many feature as we can. Editor may have autosuggestion option.

5- Do you need JavaScript debugger?

Comment: Users opinion on this subject shows that we don't need a JavaScript debugger. They told that they use the codes that they found over the internet. This means they are working codes. Therefore they don't need a debugger. We are not planning implement a debugger.

6- Do you need CSS Editor?

Comment: Users divided into two in this question. People who said that it will be good if there is a CSS Editor in the program are a little bit more. Others who don't need an editor told that the visual design is another issue, so they use other more useful programs for that. They use TopStyle commonly.

7- Do you need drag and drop ability while adding the components?

Comment: Users told that this property is again a must. According to our market research, all of the similiar programs have this property. Therefore our program will have drag and drop ability. We try to add many components to use with drag & drop.



8- Do you want to use the program as a plug-in for eclipse or as a separate program?

Comment: Users want to use this program as a separate program. The program will be a separate program, but since program consists of components, these parts can be added to eclipse with some modifications.

9- Do you want program to support high level languages ?

Comment: Users told that this will be a future work. In such a limited time, developing this feature is a hard work. Therefore, we planned to add it later.

3.3 Interview

We made an interview with Tolga Özgüç. He is working in CTS Yazılım, a company that commonly works on web projects. He told that if we introduce a good program, he can use it. We ask his expectations from such program. According to his opinion, database connections and browser compatibility are two main features. Supporting mssql and mysql is important. Also he told that, some web sites have small databases and they handle this with access database. Web pages that are developed with the program should run in all browsers. A friendly user interface and viewing the content of the database will be a very good feature. We asked him to analyze our releases and give us feedbacks. We believe that he will continue to help us.

4. PROJECT REQUIREMENTS

4.1. System Requirements

4.1.1. Hardware Requirements

Our hardware requirement decomposes into two main groups; first one is development side requirements, second one is user side requirements.

Development – Side

For fast and comfortable development, approximate minimal requirements are as follows:

- 1.6 GHz CPU
- 512 MB DDRAM
- 30 GB HDD
- 16 MB Video Card
- Ethernet Card
- Internet Connection

Our own computers meet those minimal requirements, so we don't expect to face with a difficulty.



User – Side

In order to not to face with difficulties while using the program, approximate minimal requirements are as follows:

- 1.6 GHz CPU
- 512 MB DDRAM
- 20 GB HDD
- 16 MB Video Card

4.1.2 Software Requirements

During the analysis, design, implementation and testing phases of the project, we will use several tools to carry out the project. These tools can be divided into two main groups; first one is documentation tools and the second is development tools.

Documentation Tools

Especially during the first term, we will be using several documentation tools. We have two opportunities for word processing; either Microsoft Office 2003 Word combined with Adobe Acrobat Professional, or OpenOffice 2.0 Writer. We usually use the first one. We have chosen SmartDraw as the drawing tool because of its ease of use. We have some experiences from the previous terms also. In this phase of the project, we have drawn use case diagrams using SmartDraw. For drawing Gantt chart, we preferred using Visio.

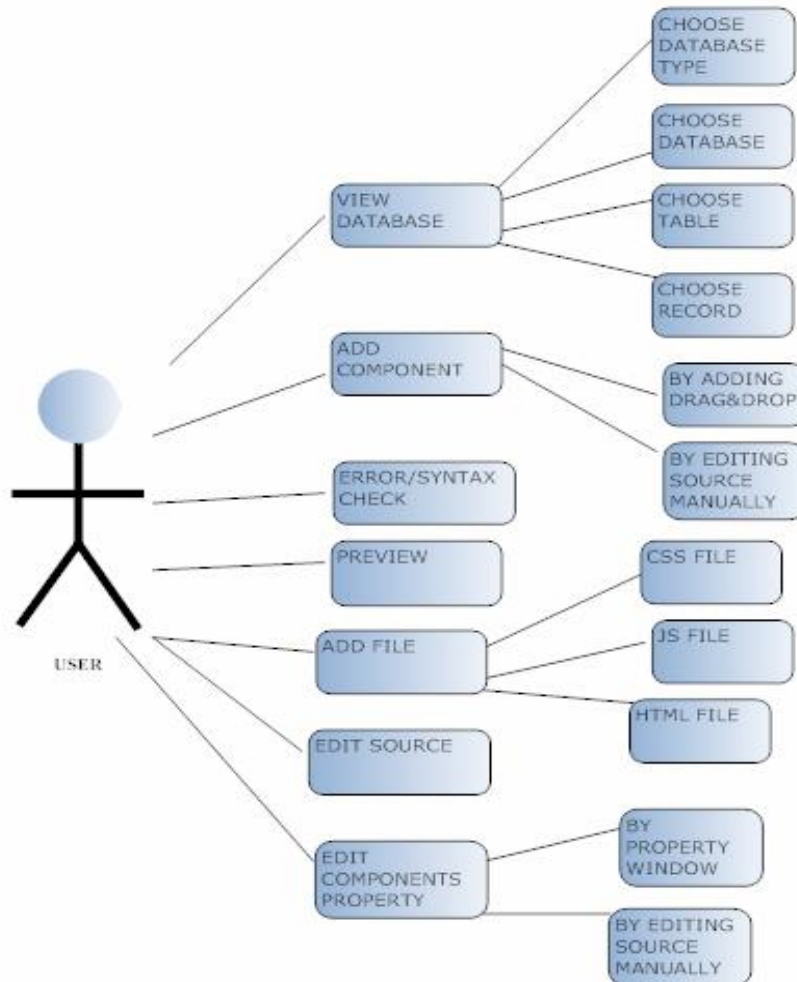
Development Tools

Since we choose Java as development language, we choose a tool for coding. There is very successful Integrated Development Environments (IDE) for Java. Eclipse is our favorite; however there is a NetBeans fan in the team also. Since compatibility is provided by Java, we can use both of the IDEs at the same time. At any time we feel we are in doubt, we are sure that we can find help on the internet in several ways: tutorials, forums, developers' web sites, etc.



4.2 User Requirements

4.2.1 Use Case Diagrams



4.2.2 User Functionality

We have listed the properties that can be most possible used by users. These are also main features of our program. We are planning add more features at the end of the term according to time remaining.

- **View Database**

User can connect to the database by a wizard. First he chooses the database type. Then he gives the path for it. By the help of this information, wizard connects to database and list its contents.



- **Create Query**

User can create a query in order to take the desired information for using in the web site. This feature can be thought as viewing database. After viewing the table, user can use the information.

- **Add File**

User can add any file to the project folder, especially css file, Js file, HTML file. The images that will be published on the web, some files to be downloaded whose link is on the page could be added to the project folder.

- **Add Component**

User can add some ready to use components. These components can be added by drag and drop ability or by editing the source manually. We are planning to add the following components: Menu, Tree view, Grid, Drop down menu, Image

- **Edit Component**

User can edit the added component's properties. He has two options for this; he can edit manually by editing the source or he can choose the component and view its properties in the property window, then he can change the values of properties.

- **Edit Source**

User can edit the documents in the solution. HTML file, Js file, css file can be edited manually. These changes can be seen by preview ability. Other files in the project directory can be edited too.

- **Error Check**

User can make an error check for the code written. Build ability will be added for this purpose. User builds the solution and system shows the errors. These errors are syntax errors.

- **Preview**

User can see the HTML pages in preview window. This window shows the page as what it will look like in the browser. By this ability, user can see the effects of the changes immediately. While in preview mode, user can add components by drag and drop.

4.3 Functional Requirements

4.3.1 Menu Requirements

There will be five main parts in our menu. You can find explanations of them below:



- **File**

This part contains the following features:

1. Open new files and projects
User can open new files to existing project or create new project.
2. Save current or all windows
User can save the current window or all windows opened at that moment.
3. Close current or all windows
User can close current window or all windows opened at that moment.
4. Exit the program
User can exit from program.

- **Edit**

This part contains the following features:

1. Undo – Redo
User can undo or redo while writing on editors.
2. Cut, Copy, Paste
User can cut, copy, and paste the selected text.
3. Find
User can search for a word from the current window.
4. Replace
User can replace the word with another one.

- **View**

This part contains the following features:

1. Toolbox
User can add ready to use components by drag and drop ability.
2. Solution Explorer
User can view the content of the current directory.
3. Server Explorer
User can analyze the database connections and view them.
4. Properties
User can edit the properties of selected component.

- **Project**

This part contains the following features:

1. Build project
User can build the project in order to see the errors.
2. Run project
User can run the project

- **Help**

This part contains the following features:

1. Help content



- User reaches the content of our help document.
2. Search
User can search in the help document for a specific problem.
 3. About AWT
Some information about the product.
 4. Report bugs and errors
A form for reporting bugs and errors.