# MIDDLE EAST TECHNICAL UNIVERSITY



# **DEPARTMENT OF COMPUTER ENGINEERING**



## SENIOR PROJECT FALL 2006

# **REQUIREMENT ANALYSIS REPORT**

# 06.11.2006



| <b>TABLE OF CONTENT</b> | S |
|-------------------------|---|
|-------------------------|---|

| 1 | INTRODUCTION  | .4       |
|---|---|----------|
|   | 1.1 Purpose of the Document                         | 4        |
|   | 1.2 Goal and Objective                              | 4        |
|   | 1.3 Scope of the Project                            | 5        |
|   |   |          |
| 2 | PROCESS PLAN  | .7       |
|   | 2.1 Team Organization                               | 7        |
|   | 2.2 Process Model                                   | 7        |
| 3 | RESEARCH  | .8       |
| Ū | 3 1 Comparison of Programming Languages             | 8        |
|   | 3 2 Literature Survey                               |          |
|   | 3 2 1 More Motion Advance Suite 3                   |          |
|   | 3.2.2 Tibco GI                                      | 9        |
|   | 3 2 3 ATF(Ajax Toolkit Framework)                   | 11       |
|   | 3 2 4 Aptana  | 12       |
|   | 3 2 5 ATLAS   | 13       |
|   | 3 2 6 Conclusion                                    | 14       |
|   | 3.3 Customer Survey                                 | 15       |
|   | 3.4 Meetings  | 16       |
|   |   |          |
| 4 | PROJECT REQUIREMENTS                                | 17       |
|   | 4.1 Functional Requirements                         | 17       |
|   | 4.1.1 GUI Design                                    | 17       |
|   | 4.1.2 Text Editor                                   | 20       |
|   | 4.1.3 WYSIWYG Editor                                | 22       |
|   | 4.1.4 JavaScript Debugger                           | 23       |
|   | 4.1.5 Database Connection                           | 24       |
|   | 4.1.6 DOM Inspection Tool                           | 25       |
|   | 4.1.7 Concurrent Versioning System (CVS)            | 25       |
|   | 4.1.8 Publish built applications on the web via FTP | 25       |
|   | 4.2 Non-Functional Requirements                     | 26       |
|   | 4.3 System Requirements                             | 27       |
|   | 4.3.1 Software Requirements                         | 27       |
|   | 4.3.2 Hardware Requirements                         | 28       |
| 5 | Usage Scenario                                      | 20       |
| J | 5 1 User Drofiles                                   | 2)<br>20 |
|   | 5.2 Use Cases                                       | 29       |
|   | 5.2 Use Cases                                       | 30<br>20 |
|   | 5.2.1 Use Case Diagrams                             | 20<br>26 |
|   | 5.2.2 Use Case Scenarios                            | 30<br>40 |
|   | 5.2 1 CUI   | 40<br>70 |
|   | 5.2.2 Toxt Editor                                   | 40<br>40 |
|   | 5.3.2 IEXTEDITOR                                    | 42<br>12 |
|   | 5.3.5 W I 51W I U EUII01                            | 45<br>11 |
|   | 5.3.4 Debuggilig                                    | 44<br>15 |
|   | J.J.J. Datavase Eulivi                              | 43       |

| 6 DATA MODELING               |    |
|-------------------------------|----|
| 6.1 Data Flow Diagrams        |    |
| 6.1.1 Level 0                 |    |
| 6.1.2 Level 1                 |    |
| 6.2 Data Dictionary           |    |
| 7 CONSTRAINTS AND LIMITATIONS | 51 |
| 8 SCHEDULE                    | 53 |
| 9 APPENDIX                    | 53 |
| 9.1 Gantt Chart               |    |
| 9.2 References                |    |

## **1 INTRODUCTION**

#### 1.1 Purpose of the Document

This is a requirements analysis report for the project "kodadı: AJAXDEV". The purpose of the document is to give a general description of the project and to identify requirements. Firstly, scope of the project is stated in a detailed way. Our objectives and goal are indicated clearly. Secondly, process detailed information is added such as team organization and process model. Next step includes the literature, customer and technical research our team has done until now. Meetings encountered can also be found in this part. After, one of the most important parts of this report which includes the requirements of the project can be seen. Functional, nonfunctional, software and hardware requirements are investigated one by one. Fifth part is believed to explain our product more clearly. It includes the use cases that describe the usage of AJAXDEV. Sequence diagrams are added in order to show the flow of states of the components. Data Flow Diagrams are also added to make the data process clearer. Restrictions, limitations and constraints are the last topic of our report.

#### 1.2 Goal and Objective

Ajax, shorthand for <u>Asynchronous JavaScript</u> and <u>XML</u>, is a web development technique for creating interactive web applications [1]. Ajax came with JavaScript and was enabled by previous versions of JavaScript, but developers have started to use it to create rich internet applications (RIA) recently. The purpose of using Ajax is to develop a website which doesn't need to refresh whole page in order to exchange data from server. This creates a more responsive webpage and saves service time. Before Ajax, to create rich internet applications either Macromedia Flash programs or IFrames which is a property of HTML were used. Flash programs are not browser independent and they are loaded slowly. IFrames are inefficient and difficult to implement. Ajax removes those disadvantages and provides more efficient, user-friendly RIAs.

The objective of project AJAXDEV is to develop an environment in order which developers can create a website having the features we described above. Our team will develop a user-friendly environment which allows non-experienced developers to build useful and attractive web sites. Besides, experienced programmers will be able to edit and debug scripts by using our tool.

According to the market research we have done up to now, we have realized that there aren't enough development kids for AJAX that support all needs of users. Each of them has a distinctive feature but nevertheless there is only one or two development tool that is completely sufficient for experienced web developers who want to use AJAX components. Even those are still being developed. Our aim is to provide a tool that gathers all of these features of existing development tools. We are planning to handle standard features of IDE's, besides we want to present user an interface to use existing AJAX based components.

### 1.3 Scope of the Project

AJAXDEV project consists of mainly 5 components which are HTML Text editor, WYSIWYG (What You See Is What You Get) Editor, parser and debugger, GUI Design and Database process handler. Moreover we will provide a CVS support which will also lead us to make use of a web service. Embedded browser will also be supported to test developed application.

#### HTML Text Editor

An HTML editor is a software application for creating web pages. Although the HTML markup of a web page can be written with any text editor, specialized HTML editors can offer convenience and added functionality. For example, many HTML editors work not only with HTML, but also with related technologies such as CSS, XML and JavaScript. In some cases they also manage version control systems such as CVS or Subversion.

We are planning to write a text editor with extra functionality for manipulating and previewing of typical programming languages used for web development. Standard features such as syntax highlighting and automatic completion will be supported. HTML, XML and Java Script are supported by this editor.

#### WYSIWYG (What You See Is What You Get) Editor

WYSIWYG HTML editors provide an editing interface which resembles how the page will be displayed in a web browser. Most WYSIWYG editors also have a mode to edit HTML directly as described above. Because using a WYSIWYG editor does not require any HTML knowledge, they are easier for an average computer user to get started with.

The WYSIWYG view is achieved by embedding a layout engine based upon that used in a web browser. The layout engine will have been considerably enhanced by the editor's developers to allow for typing, pasting, deleting and moving the content. The goal is that, at all times during editing, the rendered result should represent what will be seen later in a typical web browser.

Our WYSIWYG Editor will support standard HTML features such as buttons, forms etc. that users will be able to drag and drop. In addition to this, some simple AJAX components will be presented in labor of the user. These components are also available with drag and drop option.

#### Parser and Debugger

The parser that we plan to write will support XML, HTML and DOM files. Debugger supports only JavaScript because user will create AJAX components with JavaScript. Since it is impossible to develop a debugger for this project due to time constraints, we are planning to find, adapt and use an open source debugger component.

#### **GUI Design**

We will design a Graphical User Interface which is similar to the existing Development Environments. "Tibco", "Aptana" and "Eclipse" are being used as a layout of our design. We will develop a GUI design which shows our functionalities a user friendly and costless way.

#### **Database Process Handler**

This component is planed to manage a database connection. User will use this functionality to reach his/her database with a user friendly environment. Standard database functions like connection, table operations and SQL query evaluation are provided with this component.

#### **Embedded Browser Support**

Embedded browser will be provided to user to test and see existing file. With the help of design view, user is able to see the HTML view however, since AJAX components are not static, this feature will provide the realistic preview of the application.

#### CVS Support – Ftp Publishing

We will provide a version control system to the user and ftp support for publishing.

## **2 PROCESS PLAN**

### 2.1 Team Organization

We have a **democratic decentralized model** for our team organization. The reason is that none of us has a better experience than the others. Moreover, our project can easily be divided into components, which makes it easy to commission everybody with a separate part.

### 2.2 Process Model

We plan to use a prototyping process model for this project. Our goal is it to achieve a prototype in 2 months to make an understanding of the project. Then we will continue to develop several features of AJAXDEV. This model however does have some restrictions due to the course syllabus. We have to follow proposal- analysis- design phases incrementally do obey deadlines. Namely we combine an incremental and prototyping model for our project.

## **3 RESEARCH**

## 3.1 Comparison of Programming Languages

To decide the programming language that we will use to develop our product: **kodadı: AJAXDEV**, we did some research on the internet and got information from experienced developers. We looked at the common language: C++, JAVA and .NET. All of them are high level language and there are some similarities among them. They have broad built-in libraries and APIs. C++ is the fastest among them but also the hardest to implement. JAVA is the operating system independence language but it has less performance then others because of JVM in it. From the result of the research that we did, we decide on JAVA that we will use top develoop **kodadi:AJAXDEV**, Because we are not familiar with .NET and JAVA offers wide range of libraries and APIs.

#### 3.2 Literature Survey

We did some internet search to do market research for AJAX IDEs and frameworks. We divided our research into two market: Local Market and Foreign Market. We realize that local market is very small because we could find only one company that has a AJAX product. But the foreign market is very big and there are lots of companies and some of are the biggest software companies in the world. Majority of the product licences are commercial but there are some open source projects and products.

We create a table to show information about the products which are the most popular ones. There are nine parameters (Name, Company, Version, URL, Licence, Pricing, Support-Type, Base Concepts, Browser Support) for each AJAX IDEs and frameworks. [2][3]

#### 3.2.1 More Motion Advance Suite 3

#### Local Market:

| Name    | More Motion Advance Suite 3                   |
|---------|---|
| Company | Mor Yazılım                                   |
| Version | 3.2   |
| URL     | http://www.moremotion.com/advsuite/index.html |
| Licence | Commercial                                    |

| Pricing         | \$2.999   |
|-----------------|---|
| Support - Type  | - Custom Ide  |
| Base Concepts   | The provided elements:  |
|                 | -RefreshFields (refresh DOM elements such as input, select)             |
|                 | -RefreshPanel (Refresh the content of a whole panel element)            |
|                 | -SuggestBox (To suggest refined options as the user types in a editbox) |
| Browser Support | IE - Mozilla  |



Figure 3.1

## 3.2.2 Tibco GI

#### Foreign Market:

| Name           | Tibco GI  |
|----------------|---|
| Company        | TIBCO Software Inc.   |
| Version        | 3.1   |
| URL            | http://www.tibco.com/software/ria/default.jsp                       |
| Licence        | Developer, Public, Priviate (Commercial)                            |
| Pricing        | Free(Developer, Public), \$499 (Private - 5 concurrent end users)   |
| Support - Type | Javascript - Browser Based  |
| Base Concepts  | Development: General Interface                                      |
|                | BuilderWith General Interface Builder developers can rapidly create |

| feature-rich applications and reuseable components |   |
|--|---|
|  | • Development environment itself is AJAX-based GUI windowing          |
|  | framework with visual and drag-and-drop tools for layout and          |
|  | authoring.  |
|  | • Robust component libraries, visual tooling, step-through debugging, |
|  | automated memory management and other enterprise-grade services.      |
|  | • Integrated code testing and step-through debugging utilities.       |
|  | Deployment: General Interface Framework                               |
|  | General Interface Framework enables developers to deliver browser-    |
|  | based applications that run completely in a standard web browser.     |
|  | • Deployment-time framework, delivered from any HTTP server,          |
|  | instantly adds RIA capabilities to a standard browser.                |
|  | • Applications can be deployed as embedded components within any      |
|  | HTML page, as robust applications that occupy the entire browser      |
|  | window or portlets with drag-and-drop functionality.                  |
|  | • Client-side XML data cache boosts performance, storing information  |
|  | for use throughout a session and eliminating the need to "click-and-  |
|  | refresh."   |
| Browser Support                                    | IE  |



Figure 3.2

## **3.2.3 ATF(Ajax Toolkit Framework)**

| Name            | Ajax Toolkit Framework  |
|-----------------|---|
| Company         | Eclipse   |
| Version         | 0.1 - 20060912  |
| URL             | http://www.eclipse.org/atf/   |
| Licence         | Open Source   |
| Pricing         | -   |
| Support - Type  | JavaScript - Eclipse Plugin   |
| Base Concepts   | Open Source Eclipse Plugin.   |
|                 | Really more of a toolkit for building other AJAX IDE plugins. Starts  |
|                 | by combining Dojo and Zimbra Toolkit. Very early in it's development. |
|                 | From the project docs: ATF enables support of DOM browsing and        |
|                 | JavaScript debugging by using Mozilla XULrunner to embed the          |
|                 | Mozilla browser component (Gecko) in the Eclipse framework.           |
| Browser Support | Mozilla   |

| Debug - simple.html - Eclipse SDK  |  |   |
|--|--|---|
| File Edit Source Navigate Search Project Run Window Help   |  |   |
| 19 • 11 11 • • • • • • • • • • • • • • •   | 🐤 🔶 • 🗇 - 🔛 🖺 🛱 Debus  | g 🔟 Mozila  |
| 移 Debug 🖄 Servers 🛛 🙀 🔤 🖉 🚴 🗇 📌 🗮 🛒 💆 🗖 🗖  | M- Variables 🖾 Breakpoints Expressions Sc  | cript View 🐑 🐗 🕞 🗢 🗖  |
| III) SimpleDojo-simple.html [Mozila Appication]     III) SimpleDojo-simple.html [Mozila Appication]     III) Appication Thread (debugger keyword)     III) added[) file=dojoAjax/dojo.js line=234 pc=46     III [Daded]) file=dojoAjax/dojo.js line=234 pc=36     III [Daded]) file=dojoAjax/dojo.js line=238 pc=36     IIII [Indulest.oaded]) file=dojoAjax/dojo.js line=243 pc=60     IIII anonymous() file=dojoAjax/dojo.js line=643 pc=60     IIII anonymous() file=dojoAjax/dojo.js line=643 pc=60     IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII  | Name<br>(e) $\Leftrightarrow$ this<br>(f) $\Rightarrow$ _prototype<br>(f) $\Rightarrow$ _porent<br>$\Rightarrow$ num<br>$\Rightarrow$ num2<br>$\Rightarrow$ num3 | Value<br>function () (/n dojo.hostenvprintln_sofe = t<br>[object Window]<br>1<br>2<br>undefined |
| Mozila Browser   | 5  | 21  |
| <pre>     superior of function of the second second</pre> |  | E Dunine IS (Duni Inspector   E ⊂<br>® D html   |
| Console 🔝 Tasks Debug Output Browser Output Browser Console XHR Monitor  |  |   |
|  |  |   |

Figure 3.3

## 3.2.4 Aptana

| Name            | Aptana   |  |
|-----------------|--|--|
| Company         | Aptana Inc.  |  |
| Version         | 1.0  |  |
| URL             | http://www.aptana.com/   |  |
| Licence         | Free and open source licensed under the Eclipse Public License   |  |
| Pricing         | -  |  |
| Support - Type  | Javascript/HTML/CSS - Eclipse Plugin & Custom IDE  |  |
| Base Concepts   | <ul> <li>Javascript/HTML/CSS - Eclipse Plugin &amp; Custom IDE</li> <li>Aptana is a robust, JavaScript-focused IDE for building dynamic web applications. Highlights include the following features:</li> <li>Code Assist on JavaScript, HTML, and CSS languages, including your own JavaScript functions</li> <li>Outliner that gives a snapshot view of your JavaScript, HTML, and CSS code structure</li> <li>NEW: FTP/SFTP uploading, downloading and synchronization</li> <li>Error and warning notification for your code</li> <li>Support for Aptana UI customization and extensions</li> <li>Cross-platform support</li> <li>Works with AFLAX, Dojo, MochiKit, Prototype, Rico, script.aculo.us, Yahoo UI</li> </ul> |  |
| Browser Support | Mozilla, IE  |  |



Figure 3.4

### **3.2.5 ATLAS**

| Name            | ATLAS   |
|-----------------|---|
| Company         | Microsoft   |
| Version         | V1.0 beta - July 2006                                   |
| URL             | atlas.asp.net   |
| Licence         | Technical preview                                       |
| Pricing         | -   |
| Support - Type  | ASP 2.0 – MS Visual Studio                              |
| Base Concepts   | Multiple concepts:                                      |
|                 | - Direct AJAX Programming.                              |
|                 | - Declare / Design AJAX                                 |
|                 | - Add control that provide AJAX features (PanelUpdater) |
| Browser Support | IE, Mozilla   |



Figure 3.5

#### 3.2.6 Conclusion

Big enterprises like Microsoft, Sun, Micosystems, Adobe, Oracle and Google are currently developing and releasing plugins for Ajax support to their IDEs. Also, there are many open source projects going on for either standalone IDEs like Morfik, Aptana or plugins for open source IDEs like Eclipse.

Tibco GI and ATLAS are very good examples from the ones we examined. Tibco GI is a standalone IDE and has very nice user interface capabilities like drag-and-drop (Actually it won 2006 AJAX Product of the Year from InfoWorld). One can insert anything with drag-and-drop in Morfik.

Although ATLAS is a plugin for Microsoft's IDE which is commercial and not very good at performance, comes with many pre-defined Ajax actions like password-checker and provides a simple gui to apply these actions to HTML components like text boxes.

The browser support of IDEs we examined actually dependes on the support of browsers for JavaScript. If an IDE provides an Ajax action done with a later JavaScript and the browser does not support that version of JavaScript.

- Some frameworks offer a manager control (or something similar) to provide designer and runtime control over specific features.
- Turning AJAX on and off at design time is an important feature to help you finding bugs and AJAX problems. You can easily compare the content update with and without AJAX.
- Automatically transferring only changes of the HTML content between two AJAX responses can save a lot of additional traffic but also produces a significant higher CPU load on your web server for the diff-calculation.

#### 3.3 Customer Survey

Since we are creating a web page developing tool that supports AJAX, we determine our customer as web developers. Therefore, we contacted to companies which design web pages, via email. We asked them to guide us through our development process. Below are the e-mails we sent:

**To:** info@adasoft.com.tr, info@uzmandesign.com, info@ctsyazilim.net, bilgi@kaleyazilim.com.tr, bilgi@ly.com.tr, info@mybilet.com, info@meteksan.com.tr, info@bott.com.tr, info@webiletisim.com, info@e-netdizayn.com, bilgi@rtasarim.com, info@ventine.net, sontechyazilim@gmail.com

İyi günler,

Ben ODTÜ Bilgisayar Mühendisliği 4. sınıf öğrencisiyim, 4 arkadaşımla birlikte bitirme projesi yapıyoruz, proje konumuz "Ajax Developers Studio", ayrıntılı bilgiyi ve detayları ekte gönderdiğim "proposal" dosyasında bulabilirsiniz.

Bu projeyi gerçekleştirebilmemiz için öncelikle "Gereksinim Analizi" yapmamız gerekli, bu analizi ancak müşteri olarak nitelendirebileceğimiz, "profesyonel anlamda web uygulamaları geliştiren bir şirket"le görüşerek ve fikir alarak gerçekleştirebileceğiz.

Geliştirme ortamı kullanarak "zengin Web uygulamaları" yaratan çalışanlarınızdan fikir almak, nasıl bir ortamda çalışmak istediklerini öğrenerek projemize yön vermek istiyoruz.

Sizden bu konuda yardım ve destek bekliyoruz...

İlgilenirseniz e-posta yoluyla ya da birebir görüşerek bağlantıya geçmek istiyoruz.

Teşekkür ederim...

**To:** info@webdesignstudio.com, <u>contact@iflexion.com</u>, nquiry@lorentzconsulting.com, info@fullestop.com, support@conkurent.com, info@pedalo.co.uk I am a senior student at Middle East Technical University in Turkey. Together with four of my friends, we are going to implement a graphical development environment for web pages with Ajax. Currently we are doing requirements analysis. We want to contact to and hear suggestions of people and companies who can be customers of that product in order to better identify the requirements. We would be very glad if you share your thoughts and features you would expect from such a product with us. I am sending the proposal of the project as an attachment. If you are interested, we would like to contact you via e-mail.

#### Thank you.

We received two emails from companies which accept helping to us for our project. We interviewed with Hakan Onur, who is a Software Architect, from Adasoft via Internet. He guides us to investigate existing IDEs that support AJAX such as "Tibco" and "Morfik". He also answered some of our questions about requirements of our project. We will be able to contact to him in following phases of development.

We also made an interview with Kerem Önal who is an electrics and electronics engineer from TUBİTAK-Bilten. He is interested in developing web sites that have AJAX applications. He gave us brief information about AJAX and Google Web Tool (GWT) kit. He said that GWT can be very useful for our project and recommended us to consider it as a guide for our development process.

#### 3.4 Meetings

We had a meeting with Dr. Cevat Şener about our project. We asked him to guide us about our requirements and development process. He recommended us to use JAVA for our Development Environment and to investigate existing examples in a detailed way. We asked him that if we can use some external components in our project such as debugger that we are not able to write ourselves. He told us that there are component-type applications except from standalones and recommended us to find out a component type debugger to embed our system. We asked whether a web service is required in our project or not. He answered that if we decide to use a web service we should try to make use of it for every related component of our project.

Hi,

## **4 PROJECT REQUIREMENTS**

#### 4.1 Functional Requirements

#### 4.1.1 GUI Design

"GUI Design" component is one of the most important parts of AJAXDEV project because it provides the permanent interaction of user with Development Environment. We will design a GUI that supports all features of our IDE in a user-friendly way and also view of our IDE should be nice-looking. We have investigated existing Development Environments such as "Aptana"[3] and "Tibco"[4] to be able to identify our design as an applicable combination of these well-designed tools. Consequently, we determine our GUI functional requirements mainly, as they are stated below:

#### Source View / Design View

First of all, transition between two views of project that is being created, is the attractive feature of our Development Environment. This transition will be supplied with a tab view in same window. This will be the largest window in the middle of Environment. User will able to see how his/her project looks like while writing the source code of Internet application. In "Source View", user writes his/her source code with the help of a featured text editor. On the other hand, a WYSIWYG editor will be provided for user while he/she is works on the "Design View". Design View is also useful for inexperienced user who doesn't want to deal with so many coding. When user switches to Design View, an "Action Tool Bar" (a WYSIWYG editor) will appear on the screen which has "ready-to-use components" to add to Design View.

#### Workspace View

Workspace View will show the current workspace of user. User will see all files of current project in a hierarchical way and reach them by clicking on them. If user selects one of his/her project files, this file appears in Source View to be ready for editing. New files can be added, edited or deleted in this view.

#### **Project View**

Project View will be another useful sub window in our Development Environment. It will be used for to navigate and manage all AJAXDEV projects of user. We thought that grouping related files into a "project" makes them easier to work with as a unit. Project View is a tree view of all of the projects and associated files in user's workspace. User can expand and collapse the folders in this view to make it easier to access project files for editing. If user selects one of these projects, that project will be set as a current project and shown in Workspace View. New projects can be added, edited or deleted in this view.

#### **Outline** View

Outline View will show a visual outline of the structure of the currently open document. It will display a hierarchical grouping of the elements of code, such as variables and functions. If the document is an HTML document, tags that are used in document, will be shown in a sorted view. User will use this view to get a high-level overview of his/her code. Also user can jump to that spot in code by double-clicking an element in the Outline View.

#### Palette View

Components View allows user to add and use HTML, JavaScript components and AJAX Actions that are created before for the ease of user. While user works on Design View, he/she can easily select one of these components and add it into his/her application. This view has also "Properties" sub window to show the properties of added component such as name, type, orientation etc.

#### **Debugger** View

Debugger View controls execution of scripts during development. It will appear at the bottom of environment and provides user see his/her process clearly. There will be two views which are Variable view and Call Stack view in debugger view.

#### Database View

Database View provides an interface for user to reach and manipulate his/her existing database. When user connects to database, new window will be open from another tab view. In this view, there will be an SQL query window with which user execute his/her database

queries. There will be also a view for table operations for user to manipulate tables easily. Database view provides Schema View for user to see and reach all his/her schemas.

#### Menu bar

Our Development Environment will have a menu bar that will consists of "File", "Edit", "Project", "Tools", "Window" and "Help" as many Development Environments. Organization of items in our menu bar is listed below;

- File
  - o New File
  - o Open File
  - o Close File
  - o Save File
  - o Save File As
  - o Exit
- Edit
  - o Undo
  - o Redo
  - o Cut
  - o Copy
  - o Paste
  - o Delete
  - o Find
- Project
  - o New Project
  - o Open Project
  - o Run Project
- Tools
  - o Database Connection Manager
  - o Publishing via FTP
  - o Preview in selected browser
- Window
  - o Workspace View
  - o Project View

- o Outline View
- o Palette View
- o Debugger View
- Versioning
  - o CVS Manager
  - o Import
  - o Check-out
- Help
  - o Help Contents
  - o About

#### Toolbar

Toolbar will be a standard one as many Development Environments as well. There will be icons for user's ease to reach mostly used options while developing a project such as "New", "Save" and "Run", etc.

#### Embedded browser

Embedded browser provides user to test his/her application after running project. When user runs project, an embedded browser appears in tab view with the result web application. We also have "Preview in selected browser" option in menu bar to preview the application in different external browsers.

#### 4.1.2 Text Editor

We are planning to write an HTML text editor for our development kid. HTML editors are basic text editors with extra functionality for the manipulation and previewing of code, typically of programming languages used for web development[5]. According to the research we have done, we have specified following functional requirements for the text editor of our IDE:

- It will have the ability of reading and writing large files.
- It will provide syntax highlighting for XML, HTML, JavaScript and CSS files.
- Efficient keyboard shortcuts will be provided.
- Unlimited undo/redo will be provided.

- "Markers" for remembering positions in files to return to later will be supported.
- Any number of editor windows may be opened.
- Rectangular selection is provided.
- We will provide an auto-completion that does the followings[6]:
  - If you are typing the name of an object (e.g. "document"), when you type the period (".") to call either a method or access a property for that object, it pops up a small window displaying the available methods and properties for that object. You can also type ctrl + space to access this help at any time.
  - If you are calling a method on that object, when you type the first open parenthesis ("("), our editor will automatically create the closing parenthesis (")") for you, and it will pop up a small window with the parameters that the method takes.
- There will be a relation with WYSIWYG editor to support code generation while user uses this editor.
- It will provide intelligent bracket matching skips quoted literals and comments.
- It will provide automatic indentation.
- It will provide commands for commenting and commenting out code.
- Both literal and regular expression search and replace supported.
- It supports a large number of character encodings including UTF8[7].
- Custom file system browser component is used in open and save dialog boxes.
- Powerful keyboard navigation in the file system browser is allowed.
- Files can be deleted and renamed, and new directories can be created from the file system browser.
- Fully customizable tool bar and right-click context menu is provided.
- Link-checking tool is provided.
- It can make code checking and validation.
- Code cleanup and formatting is provided.
- Automatic save is provided to prevent user from loosing data.

### 4.1.3 WYSIWYG Editor

- Generates HTML and JavaScript code
- CSS for formatting
- User will be able to insert text in Design view.
- Multiple browser support
- Efficient keyboard shortcuts will be provided.
- Unlimited undo/redo will be provided.
- Create and modify tables and table cells. Set their border, alignment, cellspacing etc.
- Create and modify forms, text boxes, radio buttons, check boxes and buttons.
- We will provide built-in Ajax Actions such as [8]:
  - o AJAX Dynamic Table
  - o AJAX Photo Gallery
  - o Drag and drop
  - o Accordion
  - o Tabset
  - Collapsible region
  - Suggest text field
  - o Dialog box
  - o Rating widget
  - o Edit in place
- A Palette for displaying built-in Ajax actions and HTML elements [9].
- It will display the changes made in text editor instantly.
- Selecting and resizing objects option.
- Drag and Drop option to add Built-in Ajax Actions and HTML elements from Palette.
- Properties Editor for customizing the properties of added Elements.
- Insert, resize and delete images.
- Drag & drop of image files directly into the editor, as well as file browsing [10].
  - Drag images from the desktop into the editor. Browse image files on the local computer or in remote libraries. Set limits on file size and type.
- Permits entire folders to be dragged directly into the editor [10].
  - Uploaded folders are automatically zipped and a hyperlink created to the zipped file.

- Configurable toolbar
  - o Show / hide / move the buttons on toolbar.

### 4.1.4 JavaScript Debugger

We will provide the following facilities for user in the JavaScript debugger in our product to control the execution of scripts that users are debugging:

- Instant-on JavaScript debugger will be provided.
- Debug any web page containing JavaScript source or included JavaScript files, or standalone JavaScript files.
- Pause, Resume, step in/over/out, terminate operations will be provided for debugging.
- Keyboard shortcuts for pause, resume, step in/over/out, terminate will be provided.
- Several views will be shown to user:
  - Call Stack View
  - Variables View
  - Console View
- User will be able to set and clear JavaScript breakpoints in:
  - JavaScript files
  - HTML with embedded JavaScript and linked JavaScript files
- User will be able to set a breakpoint by:
  - Simply single-clicking on the line number of the line at which s/he wants to set a breakpoint.
  - If the selected line contains executable code a red dot will appear next to the line number and a breakpoint will be set at that location.
- User will be able to clear breakpoint by:
  - Place the cursor on the line at which you want to clear a breakpoint
  - Simply single-click on the red dot or the line number of the line at which you want to clear a breakpoint.

### "XMLHttpRequest Tracing" for AJAX debugging:

Actually XMLHttpRequest (XHR) is a JavaScript API that can be used to transfer and manipulate XML data to and from a web server using HTTP, establishing an independent connection channel between a web page's Client-Side and Server-Side.The data returned from

XMLHttpRequest calls will often be provided by back-end databases. XMLHttpRequest is an important part of the Ajax web development technique.

It logs the "open" and "send" calls, as well as the response status code and text. Its purpose is to help peek into AJAX applications, to learning or troubleshooting, without having to run a network sniffer. Each XMLHttpRequest instance is traced ant this allows you to track multiple requests being run in parallel.

#### 4.1.5 Database Connection

We will implement a database editor. It will be a basic editor through which the user will be able to connect to, see and modify a database. According to the research we have done, we have specified following functional requirements for the database editor of our IDE:

- The user will be able to connect to a database server if s/he has access rights on it.
- After connecting to a database a GUI window will be provided to user for database operations.
- User will be able to execute queries on the database.
- The user interface will provide user the ability to execute queries (table, column or row creation, modiffication, deletion) without the need to know the proper syntax by just clicking on the appropriate action.
- Schema selection will be provided.
- All the tables of a selected schema will be shown.
- User will be able to select a table to view or modify.
- Detailed information of the selected table (columns, rows) will be shown.
  - In the detailed table view the columns of the table will be shown and all rows of the table will be listed.
  - User will be able to select to view detailed information about column(s) of a table and modify it.
  - Data types of a table's columns will be shown when selected.
  - User will be able to manipulate rows.
- If the user tries to execute an illegal query or does not have the necessary privillages to execute a query, an error message will be shown.
- When the user makes a change on database, the result will be shown immediately.

- The user will be prompted if s/he looses his/her connection.
- The connection information will be provided as an include file to the user.

### 4.1.6 DOM Inspection Tool

Its main purpose is to inspect the Document Object Model (DOM) tree of HTML and XML-based documents by using dom parser. The initial HTML for an Ajax Application is often minimal, and in any event likely to change over time due to DOM Manipulation. All of this is very useful for checking assumptions and diagnosing problems, since many Ajax bugs arise because the programmer misunderstood the DOM state at a particular time.

- showing the DOM-Tree with nodes.
- drill down the hierarchy, search for keywords.
- current element highlighted in page.
- Node name, type and value are shown.
- javascript variables referenced in current line will be displayed.

### 4.1.7 Concurrent Versioning System (CVS)

- Provides a Versioning Manager wizard that helps:
  - To designate the repository(the current and historical file data is stored, often on a server).
  - Set up your working directory.
- Import
  - Copying a local directory tree into the repository for the first time.
- Check-out
  - Creates a local working copy from the repository. Either a specific revision is specified, or the latest is obtained.

### 4.1.8 Publish built applications on the web via FTP

- Provides a FTP Publish wizard that helps setting up FTP Preferences
  - Set the "FTP Host"
  - Enter Login Name into the "Login" field
  - Enter Password into the "Password" field
  - "Save" option if user would like to remember the password

## 4.2 Non-Functional Requirements

During our research we also examined products besides functionality. Our product not only needs to be functional but also needs to satisfy the customer by other aspects such as usability. We examined other products in this market and determined how our product shall appear.

#### Look and Feel

- When the program is started user will see design view on center, Ajax actions and HTML objects that can be added to his/her project on right, and project view on left of the main window.
- There will be visuals helping the user developing web applications.
- Text editor and design view will get the bigger part of the layout.
- The product shall appear authoritative.
- The product shall give user the feeling that s/he is doing something professional.

#### Usability and Understandability

- The product shall make the users want to use it.
- The product shall be used by users varying from someone who has little experience and wants to build a web page to people with years of experience and having professional aims.
- In order to use all the functions of the product fully, users need to have some experience in databases, HTML and web development.
- The product shall be used without any training.
- Users must know English in order to use the product.
- The product shall hide the details of its construction from the user.
- The product shall use symbols and words that are naturally understandable by the user community.

#### **Productization Requirements**

• The product shall be able to be installed by an untrained user.

#### Adaptability Requirements

- The product will run on both Windows XP and Linux.
- The language of the product will be English.
- The product might be introduced to Turkish market in which case a Turkish version will be provided.

#### Integrity requirements

- When a file or project is opened, the product shall check if that file type is supported, preventing incorrect data from being introduced.
- The product shall protect itself from intentional abuse.

## 4.3 System Requirements

### 4.3.1 Software Requirements

Minimal software requirements of our system are examined in two sides:

## **4.3.1.1 Development Side**

- Windows XP or Recent Linux Distribution
- Installation of JDK (Java Development Kit)
- Installation of J2EE1.4 (Java Platform Enterprise Edition)
- Web Server (Apache)
- CVS Server
- FTP Server
- DBMS(MySql-Oracle)
- Web Browser (Internet Explorer, Mozilla)

## 4.3.1.2 End user Side

- Windows XP or Recent Linux Distribution
- Installation of JRE (Java Runtime Environment)
- CVS Server (optional, for using CVS feature)
- FTP Server (optional, for using publishing feature)
- Web Server
- DBMS
- Web Browser (Internet Explorer, Mozilla)

### 4.3.2 Hardware Requirements

Minimal hardware requirements of our system are examined in two sides:

## 4.3.2.1 Development Side

A PC or compatibles with the following configurations is needed:

- 1GHz processor speed
- 512 MB RAM (1 GB recommended)
- 1024 x 768, 16-bit display (32-bit recommended)
- 1GB available disk space

## 4.3.2.2 End user Side

A PC or compatibles with the following configurations is needed:

- 1GHz processor speed
- 256 MB RAM (512 MB recommended)
- 1024 x 768, 16-bit display (32-bit recommended)

## **5 Usage Scenario**

## 5.1 User Profiles

There are two types of users who will use our final product. We intent to release a product which can be used efficiently by both experienced and inexperienced web developers who want to develop AJAX products. Therefore, we will support a graphical user interface which will allow inexperienced users to make use of AJAX components easily. Besides, there will be additional features for experienced users to build professional web sites supporting AJAX.

#### • Experienced Web Developers:

They can use text editor and JavaScript debugging options, database connection handling, project management tool and CVS support for their projects.

#### • Inexperienced Web Developers:

They can use built-in AJAX components that are applicable by using our WYSIWYG editor. They can use automatic completion, syntax highlighting and database editor for their projects to build AJAX supporting web sites easily.

## 5.2 Use Cases

## 5.2.1 Use Case Diagrams

## 5.2.1.1 GUI





## 5.2.1.2 Text Editor



## 5.2.1.3 WYSIWYG Editor



# 5.2.1.4 Debugging



## 5.2.1.5 Database Editor



## 5.2.2 Use Case Scenarios

## 5.2.2.1 GUI

Source View: User will select Source View from the GUI and see source code.

**Design View:** User will select Design View from the GUI and see the existing design of the page.

**Workspace View:** User will select Workspace View and see all files of currently developing project.

**Project View:** User will select Project View and see all Projects and their hierarchical organization.

**Outline View:** User will select Outline View and see the objects of currently editing source code.

**Debugger View:** User will select Debugger View and see some variables and execution flow.

**Palette View:** User will select Palette View and see built-in components that s/he can add to design view.

**Database View:** User will select Database View and see his/her database information and also tables.

**Embedded Browser:** User will run his/her project to test it and see preview of application in embedded browser.

**Preview Browser:** User will select Preview Browser and an external browser supplies a preview of application.

**Versioning:** User will select a versioning process; import and check out his code and a CVS server will handle required process.

**Publishing via FTP:** User will select a file; system will send it to an FTP server via a FTP protocol.

## 5.2.2.2 Text Editor

**Undo/Redo Code:** User will press undo or redo to disable or enable changes he/she made on his/her file.

**Comment/ Comment out code:** User will select a part from the file and comment in or out this part.

Search & Replace code: User will find an expression, word or sentence and replace it with another.

**Use keyboard shortcuts:** User will use keyboard shortcuts to manage the tasks easily. **Select rectangle:** User will select a part in a rectangle and change it according to his/her needs.

**Bracket Matching:** When user comes to a bracket, cursor will automatically shoe the match of that bracket.

Customize toolbar: User will customize the toolbar according to his/her needs.

**Use palette:** User will use the palette to add the source codes of the built-in components. **Write Code:** User will write source code.

**Syntax Highlighting:** When the user writes his/her code syntax highlighting will automatically highlight the built-in functions or expressions of the related language.

Automatic Completion: When user is typing the name of an object (e.g. "document"), when you type the period (".") to call either a method or access a property for that object, it pops up a small window displaying the available methods and properties for that object. User can also type ctrl + space to access this help at any time. When user is calling a method on that object, when you type the first open parenthesis ("("), our editor will automatically create the closing parenthesis (")") for him/her, and it will pop up a small window with the parameters that the method takes.

**Automatic Indentation:** When the user is writing a code, automatic indentation will indent his/her code according to the related programming language.

**HTML code cleanup/formatting:** After user writes the code, editor will check HTML validity and clean the code to make a correct HTML file.

Link Checking: When the user has entered a link, editor will automatically highlight it as a link.

**HTML Validation:** While user is writing the code, editor will check if he/she is writing HTML code validly.

**Code Generation:** When the user uses the palette, editor will automatically generate the related code of the component.

**Provide Marker:** When the user opens another file, "markers" for remembering positions in files to return to later will be supported.

## 5.2.2.3 WYSIWYG editor

**Undo/Redo operation:** User will press undo or redo to disable or enable changes s/he made on his/her file.

Keyboard Shortcuts: User will use keyboard shortcuts to manage tasks easily.

**Using Palette:** User will use drag & drop option to add built-in component to his/her design view.

Insert Text: User will enter text input to his/her design view.

Modifying Object: User will modify components that are previously added.

**Customizing the properties of element on properties editor:** User will arrange the desired properties of elements.

**File Operations from desktop:** User will add images and files to his/her design view with drag and drop directly from desktop.

Image Operations: User will add, delete, resize etc. images.

**Code Generation:** When the user use palette/insert text/modify objects/customize properties of elements/make file operations /make image operations.

## 5.2.2.4 Debugger

**Keyboard shortcuts:** User will use the keyboard shortcuts to manage the debugger operations which are: Pause/Resume, Step in/over/out.

**Pause/Resume:** User will press the Pause/Resume button. The debugging engine will stop or continue to control the execution of scripts. Call stack view and variables view are updated according to these operations.

**Step in/over/out:** User will press the Step in/over/out button. The debugging engine will go in/over/out the execution step of the scripts its debugging.

**Set/clear breakpoints:** User will click the line number at which he/she wants to set/clear breakpoints on the editor window. Breakpoint set/clear at this line. The debugging engine will stop/continue at breakpoints. Call stack view and variables view are updated according to these operations and the user will see the values of the variables at that breakpoints.

**Call Stack view:** When the debugger is stopped, the Call Stack view displays the list of active functions.

**Variables view:** When the debugger is stopped, the variables view displays values for the current function.

## 5.2.2.5 Database Editor

**Connect to Database:** User will press connect button. Then user interface will bring up connection dialog and waits for the user to enter connection info. After user enters connection info, user interface will send it to DBMS. If the connection info is correct, DBMS will return database info and user interface will show the result to user and also will prepare an include file.

**Select Database Schema:** User will select to view a schema. User interface will generate query and send it to DBMS. DBMS will execute the query and send the result to UI. UI will show the result to user. If the query is invalid, UI will show an error message to user.

**View, Modify Table:** User will select an operation on a table. User interface will generate query and send it to DBMS. DBMS will execute the query and send the result to UI. UI will show the result to user. If the query is invalid, UI will show an error message to user.

**Enter SQL Query:** User will write a query. UI will send it to DBMS. DBMS will execute the query and send the result to UI. UI will show the result to user. If the query is invalid, UI will show an error message to user.

## 5.3 Sequence Diagrams

## 5.3.1 GUI





publishing via FTP (file name)



send (file)

### 5.3.2 Text Editor



### 5.3.3 WYSIWYG Editor



## 5.3.4 Debugging



### 5.3.5 Database Editor



## **6 DATA MODELING**

## 6.1 Data Flow Diagrams

6.1.1 Level 0



6.1.2 Level 1



# 6.2 Data Dictionary

| name:                  | User commands                         |
|------------------------|---------------------------------------|
| where used / how used: | GUI(1.0) input                        |
| description:           | Every external input that user enters |

| name:                  | Displayed Response              |
|------------------------|---------------------------------|
| where used / how used: | GUI(1.0) <i>output</i>          |
| description:           | Every output provided by system |

| name:                  | Database Information                  |
|------------------------|---------------------------------------|
| where used / how used: | Database Editor (5.0) input           |
| description:           | Information Stored in user's database |

| name:                  | Connection Information                             |
|------------------------|--|
| where used / how used: | Database Editor (5.0) <i>output</i>                |
| description:           | Connection information and Queries entered by user |

| name:                  | Request                                  |
|------------------------|--|
| where used / how used: | Main Process (2.0) output                |
| description:           | Signal to publish application in browser |

| name:                  | Check-in Files              |
|------------------------|-----------------------------|
| where used / how used: | Main Process (2.0) output   |
| description:           | Sending files to CVS server |

| name:                  | Import Files                    |
|------------------------|---------------------------------|
| where used / how used: | Main Process (2.0) input        |
| description:           | Receiving Files from CVS server |

| name:                  | User Files                  |
|------------------------|-----------------------------|
| where used / how used: | Main Process (2.0) input    |
| description:           | Sending files to FTP server |

| name:                  | Publishing Files                |
|------------------------|---------------------------------|
| where used / how used: | Main Process (2.0) output       |
| description:           | Receiving files from FTP server |

| name:                  | Debug Operations                |
|------------------------|---------------------------------|
| where used / how used: | Main Process (2.0) output       |
|                        | JavaScript Debugger (6.0) input |
| description:           | Debugger related inputs         |

| name:                  | Debug Result   |
|------------------------|--|
| where used / how used: | Main Process (2.0) <i>input</i><br>JavaScript Debugger (6.0) <i>output</i> |
| description:           | Outputs of debug operation   |

| name:                  | Source Code  |
|------------------------|--|
| where used / how used: | WYSIWYG Editor (3.0) <i>input</i><br>Text Editor (4.0) <i>output</i><br>JavaScript Debugger (6.0) <i>input</i> |
| description:           | Source Code of Application   |

| name:                  | Component info  |
|------------------------|---|
| where used / how used: | WYSIWYG Editor (3.0) output<br>Text Editor (4.0) <i>input</i> |
| description:           | Inputs from design view to generate source code               |

| name:                  | User Request             |
|------------------------|--------------------------|
| where used / how used: | GUI(1.0) output          |
|                        | Main Process (2.0) input |
| description:           | User inputs              |

| name:                  | System Response           |
|------------------------|---------------------------|
| where used / how used: | GUI(1.0) input            |
|                        | Main Process (2.0) output |
| description:           | System output             |

| name:                  | Display Info   |
|------------------------|--|
| where used / how used: | WYSIWYG Editor (3.0) <i>output</i><br>Main Process(2.0) <i>input</i> |
| description:           | Design View output for display                                       |

| name:                  | Visual Operations   |
|------------------------|---|
| where used / how used: | WYSIWYG Editor (3.0) <i>output</i><br>Main Process(2.0) <i>output</i> |
| description:           | User inputs related with WYSIWYG editor                               |

| name:                  | Output  |
|------------------------|---|
| where used / how used: | Main Process(2.0) <i>input</i><br>Text Editor (4.0) <i>output</i> |
| description:           | Output from Text editor to display                                |

| name:                  | Input   |
|------------------------|---|
| where used / how used: | Main Process(2.0) <i>output</i><br>Text Editor (4.0) <i>input</i> |
| description:           | User inputs related with Text editor                              |

| name:                  | Database Operations                      |
|------------------------|--|
| where used / how used: | Database Editor (5.0) input              |
|                        | Main Process (2.0) output                |
| description:           | User requests on database                |
|                        |  |
| name:                  | Desired Information                      |
| where used / how used: | Database Editor (5.0) <i>output</i>      |
|                        | Main Process (2.0) input                 |
| description:           | Information of user database for display |

## **7 CONSTRAINTS AND LIMITATIONS**

We should consider some constrains and limitations while satisfying the requirements and functionalities in order not to face with serious problem during other steps of the project process.

## 7.1 Time Constraints

We have to finish our project by June and also we should provide a prototype at the end of these semester. Therefore, especially for a software project, this is the most important constraints. Being able to use our time efficiently is very important for us to follow our program regularly. In case of schedule problem, to compensate lost time we should focus on the project instead of other responsibilities and spend more time on it. As a result, although we thought lots of features and special properties for development environment, for timing reasons, we may not able to do some exciting features because we should provide expected functionalities and basics firstly.

#### 7.2 Experience & Skills of Members Constraints

As developers, our programming and design skills and experiences is also one of the restrictions. Although we have made software projects before, it was simpler than our current project and we do not have experience about creating development environments. Thus, this restricts our opinions of what we are able to make. In addition, It is very difficult for us to manage unexpected problems about this field but we may consult experienced people to get help about solving problems

### 7.3 Funding Constraints

Since we will not need any additional hardware and software that have a cost for us to implement our project, we do not have a cost for them. In addition our team members are students and we will not pay anyone to during the project. Therefore, there is not any funding constraint.

#### 7.4 Resource Constraints

While we are doing our project we need different hardware and software resources. We generally get easily these resources; as software requirements, we need web server, databases servers and some of development tools. Many of these are freeware, and we can get others in our department freely. We can also deal with hardware requirements for our project by the help of our personal resources temporarily so we do not think that the resources will be a problem for us to complete the project.

## **8 SCHEDULE**

Gantt chart is avaliable at Appendix Part 1.

## 9 APPENDIX

## 9.1 Gantt Chart



## 9.2 References

- [1] http://en.wikipedia.org/wiki/AJAX
- [2] http://www.daniel-zeiss.de/AJAXComparison/Results.htm
- [3] http://blogs.pathf.com/agileajax/2006/08/ides\_with\_ajax\_.html
- [4] <u>http://www.aptana.com/docs/index.php/Displaying\_Aptana\_Views</u>
- [5] http://www.tibco.com/software/ria/gi\_resource\_center.jsp
- [6] http://www.aptana.com/docs/index.php/About\_Code\_Assist

- [7] http://en.wikipedia.org/wiki/Text\_editor
- [8] http://ajaxpatterns.org/wiki/index.php?title=Tools
- [9] http://www.eclipse.org/gef/overview.html

[10] <u>http://www.xstandard.com/page.asp?p=A4372B00-8D7F-4166-977C-64E5C4E3708E&s=E638AEB0-ADC1-448B-9CE5-FB8AAE1FE55B</u>