MIDDLE EAST TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

Senior Project
Requirement Analysis Report

A Company Manufactures Everything
(ACME)

TABLE OF CONTENTS

1. Introduction

The Internet has become an indispensable part of our life in conjunction with the growth of technology. People of all age groups and companies in all sectors use the Internet extensively. Therefore, current time period is also called as Information Era. Web is the most contributive element to this fame.

In recent years, web technology has made big progress. At 2004, O'Reilly and MediaLive International have organized a conference with contribution of big world-wide web companies such as (Google, Yahoo, Msn, Amazon, eBay…) to discuss future of the web and to give a name to recent developments and trends. At this point a new decision has aroused, Web 2.0. The complex and evolving technology infrastructure of Web 2.0 includes server-software, content-syndication, messaging-protocols, standards-based browsers with plug-ins and extensions, and various client-applications. These differing but complementary approaches provide Web 2.0 with information-storage, creation, and dissemination capabilities that go beyond what the public formerly expected of websites[1]. And AJAX is the one of the technologies coming with this concept. AJAX is not Web 2.0; it is a one of the techniques featured in a Web 2.0 website. Web technology has widely adopted the Web 2.0, every day number of new web 2.0 pages increases, therefore number of pages with AJAX increases. AJAX that stands for Asynchronous JavaScript and XML is a new technology that helps to create interactive web applications. In this technology there is no need to reload the whole page, instead only the necessary and sufficient parts of the web application will be reloaded[2]. So we can say AJAX technology brings the power of internet and the speed of desktop applications together[3]. Therefore, web pages with AJAX technology are mostly preferred by users who are got familiar with AJAX technology by the web pages generated by Google Company such as Gmail, Google Maps.

[1] Web 2.0, Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Web_2

[2] *AJAX (programming)* http://en.wikipedia.org/wiki/AJAX_%28programming%29

[3] *Mastering AJAX, Part 1: Introduction to AJAX* by Brett McLaughlin April 12, 2006, http://www.javascriptsearch.com/guides/Beginner/articles/MasteringAJAX_1.html

Today implementing web pages with AJAX technology saves time of end-users by increasing efficiency of the web-pages by shortening loading time of the page. However, developers are losing time when they are trying to implement this kind of web applications. Nowadays, speed is a priority for developers, and most developers are using environments that help them to create their applications in a more rapid way. So most developers prefer integrated development environments to create their applications.

2. Project Definition

As AJAX is a new, powerful technology, developers need such a powerful environment which simplifies code writing and makes it easy to design web sites using AJAX.

LADES will be an AJAX Development Environment Studio, which will be able create any (AJAX based or not) web application. LADES will be a platform independent and a user friendly development environment. In its design, diversity of users, their different knowledge levels, different devices on which it runs, different ways of interacting, different implementations and environments will be all taken into consideration. In LADES, simple tools that allow rapid development and rapid deployment will be available. Also, it will simplify access to databases from user interfaces as easy as drag and drop. In order words, our project LADES will be a rapid web application development environment that will let you create AJAX web page applications while reducing complexity and deployment times.

2.1. Application Areas

Nowadays we know the importance and frequent use of web applications. As well as web applications AJAX usage frequency is getting higher so that the web applications and AJAX has a wide usage area. Consequently our LADES program has several application areas. Most common of these application areas are as the following:

- Mapping applications
- Real time search engines
- Chat programs
- Real time form validation and processing
- Online shopping applications
- Email applications
- And any kind of web applications

In conclusion, when easiness and efficiency of our LADES program is taken into consideration, AJAX can be used in web any application; both for small and complicated ones. Therefore, we can say our LADES program will be helpful for most of the web application developers. No matter the complexity of the web application, as developers prefer writing less code, and saving time; our project will provide these features to the web application developer.

2.2. Project Features

Some features that are listed through this list will compose the core functionality that is required for basic functional AJAX development software. Other features that will extend the usability of the product will be marked as optional and will be implemented as external components. These components will be added to project if the core functionality works properly and we have enough time left from implementing core functionality.

In addition to this, we would like to implement a modular design in our project. If we can achieve this, we can implement the core functionality and add new modules to the system any time we want. We will work on this modularity issue.

LADES begins like many other applications, by starting a new project by the help of wizards. A project defines the bunch of the web pages which serves one aim or just a single application. In addition, using project templates is another option to get started.

To see the details of project, a navigation tool provides an organized view of the objects in a view, as well as commands for managing a project. Alternatively, the File System view could be selected, which reflects the physical file system of the project on disk.

After that to create your application, there are two options,

- **Drag-and-Drop Visual Components:**

  A simple drag-and-drop user interface design ensures fast refinement from prototype to completed application. In addition to creating and using visual components like text fields and buttons, you can also assign database information, AJAX events to components through drag-and-drop. Component size and placement can be adjusted using the mouse or keyboard. Also by mouse or keyborad interaction views or properties of these visual components can be modified by the help of LADES only filling a text boxes. In addition, component size and placement can be adjusted by dropping components to desired location.

  When you open the application and start a new project, an empty canvas and a visual editor are presented. The canvas provides the work area for laying out the user interface. Applications can be visually built quickly and intuitively by dragging and dropping graphic elements and user interface components onto the canvas.

  At the top of design window, a toolbar is designed to provide access to graphical user interface components, such as custom graphics, HTML elements, database connection or adding AJAX events. Simply drag and drop graphic elements and user interface components from the toolbar onto the canvas constructs the web application by the default property of the components. Each component has properties that can be easily edited through their property fill wizard, which are available for standard components and for custom components. Typical properties might include text size and styles, color, or the size of a component.

  In addition, views can switch from visual to working with code directly. Synchronized editing ensures that changes made to objects in any view are reflected in the canvas.

- **Creating Manually:**
  For those of you who already know the programming language and prefer to hand code, or for those who want to see what's happening behind the scenes, LADES provides a Code Editor that supports HTML pages, PHP pages. In this mode, syntax highlighting and code completion is supplied to the user. Syntax colorization helps the user keep coding issues straight. Also the user can move between development modes, from visual to the Code Editor, with ease. Maybe we could add drag and drop functionality to this component, to supply code template for most used components.

LADES also provides different variety of functionality, such as debug, preview mode or running in browser or deployment of created files. LADES will be more detailly explained in following sections.

General features that LADES will support are as follows.

Design View will support:

- Drag & drop webpage components (core)
- Dragging webpage components to support repositioning and resizing (core)
- Setting webpage component properties and their events using its GUI (core)
- Generating AJAX related codes automatically (core)
- Creating database connections by user friendly wizards (core)

Text Editor will support:

- On the fly syntax highlighting (core)
- Syntax checking (optional)
- Auto indenting (optional)
- Find and replace support (core)
- Class / Function hierarchy browser (optional)

General project features:

- CVS support (optional)
- FTP upload support for remote deployment (core)
- Running the project on the remote machine (core)
- Timer based auto save (core)

3. Market Observation

3.1. Literature Survey

For our project we have done much research about AJAX and similar IDE programs that are created by developers. We have seen that, there are similar AJAX developing studios that aim to help users to create web based applications using AJAX technology. Some of these projects are standalone IDES, while some of them are well know JAVA IDE Eclipse plug-ins. We have investigated these projects and understand their strong parts and weaknesses.

3.2. Tibco General Interface

TIBCO Software Inc. is a leading business integration and process management software company that enables real-time business. Real-time business is about helping companies become more cost-effective, more agile and more efficient. They have a solution for rapid AJAX development called TIBCO General Interface.

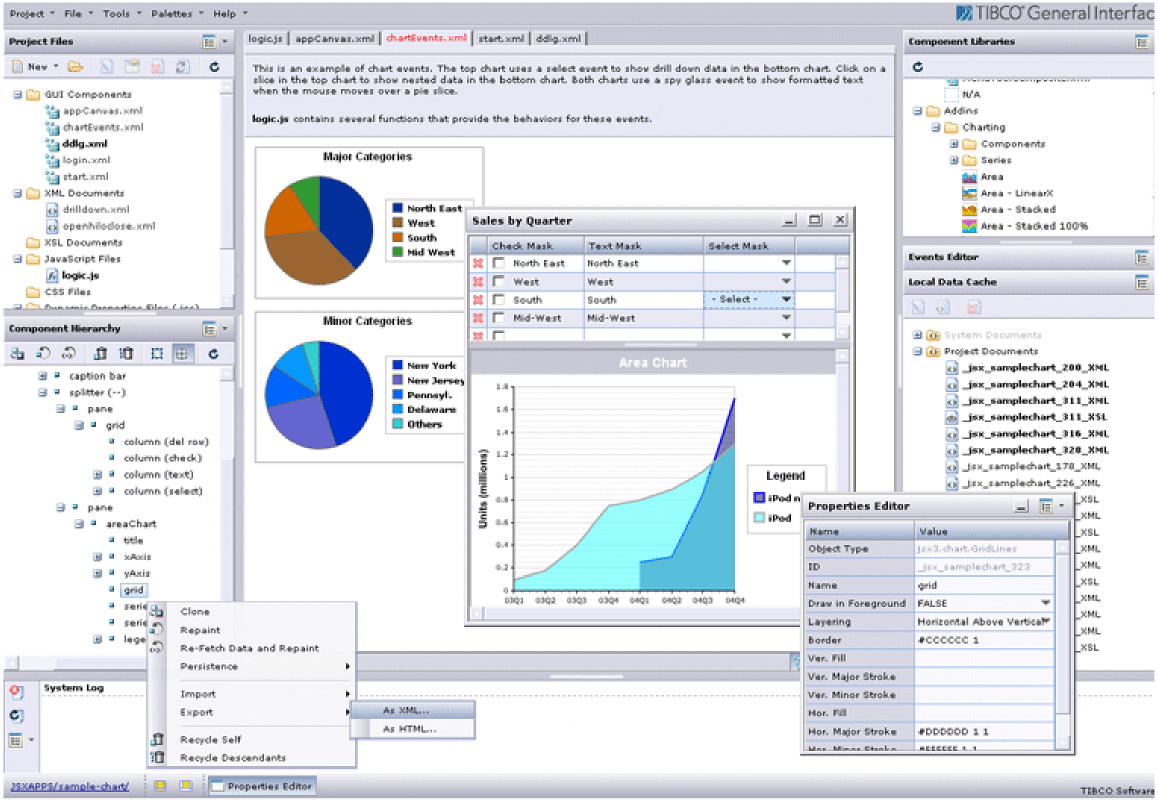
TIBCO General Interface is a Rich Internet Application (RIA) development solution that uses the asynchronous communications, JavaScript, and XML ("AJAX") capabilities present in common Web browsers to allow developers to easily deploy Rich Internet Applications without dependency on applets, plug-ins or installation of client/server frameworks. Using TIBCO General Interface’s framework and visual development tools, developers can use AJAX techniques to quickly develop more sophisticated, highly responsive Web applications.

The system is a complete set of GUI widgets that will run in Microsoft's Internet Explorer 6.0 or Mozilla Firefox 1.5. The completely browser-based IDE is written in the same set of widgets, itself serving as one of several very good examples of what the toolkit can do. There are panes within panes to hold the source code, and many of these panes come with multiple tabs, multiple views, accordion tabs, and even some graphs. All of this is loaded from an HTML file, not an executable.

General Interface uses XML as an intermediate language: The JavaScript code on the local browser turns this XML into the HTML actually displayed. The richness of the interface is surprising. GUIs are naturally hierarchical, a perfect match for the treelike rigor of XML. Many users of TIBCO's General Interface won't be spending much time with the underlying XML. Although the IDE kindly offers an "expert" window for editing the XML code, most developers will probably stick with the visual drag-and-drop tools.

The TIBCO product is also the most developer-friendly tool in the bunch. The main editing pane offers five ways to look at the current project. The first, of course, is the visual designer that shows the layout. The second displays the underlying XML. Many visual applications hide this information or at least make it difficult to see. General Interface brings it forward and even lets you edit it. Yet another view shows the HTML that will be generated by the JavaScript when it processes the XML layer. This information is invaluable for anyone trying to get the look of the application just right. Even though the application is written in JavaScript, the browser's HTML rendering engine decides on the appearance. Seeing the HTML produced makes it possible to understand what is going on under the hood.

One annoying thing that I experienced using this tool is, its drag and drop mechanism doesn’t support resizing or positioning elements. You can only drag and drop elements to the top left corner. If you want to resize or change its position, you should manually edit its CSS properties through CSS panel. Another badly designed point is, the HTML code comes from an automated algorithm that produces just one long string of tags, without any indenting which makes it hard to read.



This is an example of chart events. The top chart uses a select event to show drill down data in the bottom chart. Click on a slice in the top chart to show nested data in the bottom chart. Both charts use a spy glass event to show formatted text when the mouse moves over a pie slice.

logic.js contains several functions that provide the behaviors for these events.

Sales by Quarter

	Check Mask	Text Mask	Select Mask
☒	□ North East	North East	
☒	□ West	West	
☒	□ South	South	- Select -
☒	□ Mid-West	Mid-West	
☒	□		

Properties Editor

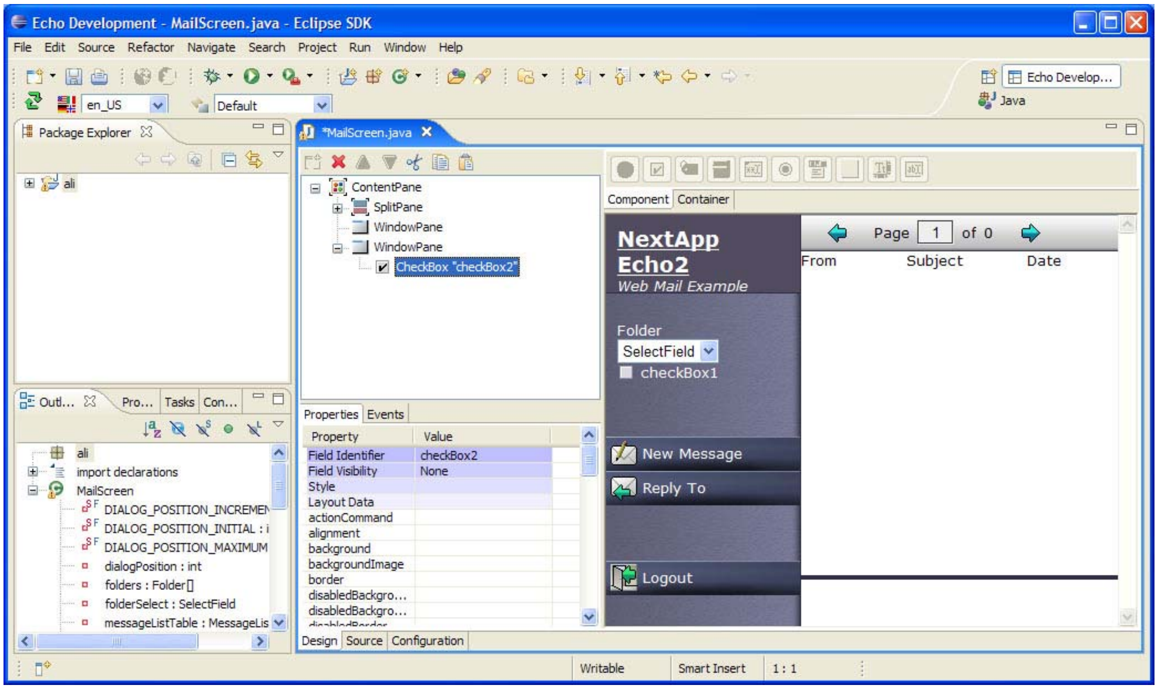
Name	Value
Object Type	jsx3.chart.GridLines
ID	_jsx_samplechart_323
Name	grid
Draw in Foreground	FALSE
Layering	Horizontal Above Vertical
Border	#CCCCCC 1
Ver. Fill	
Ver. Major Stroke	
Ver. Minor Stroke	
Hor. Fill	
Hor. Major Stroke	#DDDDDD 1 1

Figure 1 - TIBCO General Interface GUI

3.3. EchoStudio 2.1

EchoStudio 2.1 is an Eclipse 3.1-based rapid application development tool capable of creating web-based applications that work and act like rich clients. Building upon the power of the open-source Echo2 Framework, EchoStudio offers easy-to-use tools that enable developers to design more compelling user interfaces in less time.

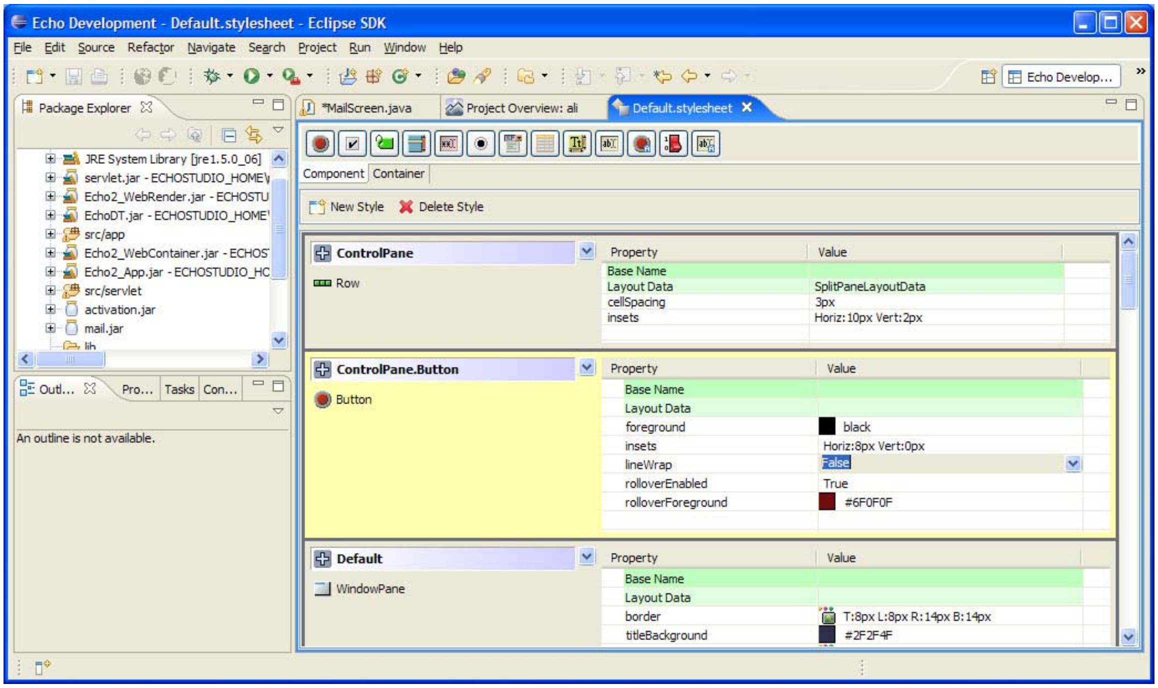
It can not be called as an AJAX IDE, since it is designed as a plug-in for Eclipse platform as shown in Figure 2. It provides certain functionalities such as property and events editing, hierarchical component views and built in browser view. On the other aspect, it is not WYSIWYG, so it doesn't support drag and drop.



Property	Value
Field Identifier	checkBox2
Field Visibility	None
Style	
Layout Data	
actionCommand	
alignment	
background	
backgroundImage	
border	
disabledBackgro...	
disabledBackgro...	

Figure 2 - EchoStudio 2.1 GUI

Its CSS editor is very handy to set different styles to several form elements. Figure 3 shows this CSS editor.


Echo Development - Default.stylesheet - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Echo Develop...
Package Explorer
JRE System Library [jre1.5.0_06]
servlet.jar - ECHOSTUDIO_HOME\
Echo2_WebRender.jar - ECHOSTU
EchoDT.jar - ECHOSTUDIO_HOME'
src/app
Echo2_WebContainer.jar - ECHOS
Echo2_App.jar - ECHOSTUDIO_HC
src/servlet
activation.jar
mail.jar
lib
Outl... Pro... Tasks Con...
An outline is not available.
*MailScreen.java
Project Overview: ali
Default.stylesheet
Component Container
New Style Delete Style
ControlPane
Row
ControlPane.Button
Button
Default
WindowPane

ControlPane

Property	Value
Base Name	
Layout Data	SplitPaneLayoutData
cellSpacing	3px
insets	Horiz:10px Vert:2px

ControlPane.Button

Property	Value
Base Name	
Layout Data	
foreground	■ black
insets	Horiz:8px Vert:0px
lineWrap	False
rolloverEnabled	True
rolloverForeground	■ #6F0F0F

Default

Property	Value
Base Name	
Layout Data	
border	T:8px L:8px R:14px B:14px
titleBackground	■ #2F2F4F

Figure 3 - EchoStudio 2.1 CSS Editor

3.4. GWT

Google Web Toolkit is just newly born software development framework. It is firstly released on May 2006. However, nowadays its 1.1.10 version is available on the project web site. The project is developing very rapidly with the help of an active user forum in which technical discussions and bug reports are held. Therefore, it seems that this project will get more effective in near future in web programming.

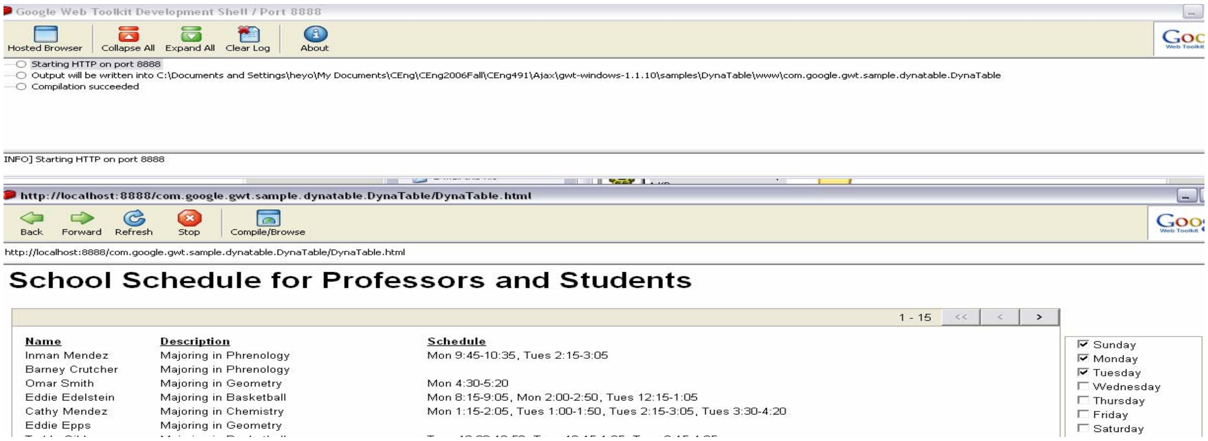
GWT is not a commercial product. As for IDE integration, there's mostly just an Eclipse project generator and a "hosted mode" runtime. The project helps people to develop featured dynamic web pages using AJAX with only basic knowledge on HTML and java. The server and the client application are embedded into just one application written in

Java. And the creating web application through this toolkit is just like a creating swing application in java like creating visual components, setting up event handlers, debugging, and so forth which are all within a familiar IDE.

The GWT provides very different and useful features from other AJAX development environments. First of it supplies reusable and dynamic user interface components, and then it has really simple remote protocol call to establish communication between web application and web server, the needed the request serialization and response deserializations are automatically done via serializable Java classes for request and response in production. Moreover, it overcomes one of the most known mistakes of AJAX which is breaking back button functionality which is based on not to having a state. For this purpose, the state is added to the history of back button. Furthermore there is a feature which makes it noticeable around other AJAX development tools. This feature is real debugging since it is java coded, in virtual java machine it runs, therefore you have advanced debugging features of java environments like Eclipse. Finally the other important feature is browser compatibility of the GWT, it is stated in the GWT homepage as 90% of the development time is spent when being worked around subtle incompatibilities between web browsers and platforms, and also, the java script codes are very problematic because of JavaScript's lack of modularity which makes sharing, testing, and reusing AJAX components difficult and fragile.

To sum up, the history and its advantages are mentioned so far, however its advantages and features are not restricted with these, above only the important and different features are listed, since this is still on construction, following updates, forums and new version will be very useful to have idea and discuss the technology.

Running in hosted mode:



```
○ Starting HTTP on port 8888
○ Output will be written into C:\Documents and Settings\heyo\My Documents\CEng\CEng2006Fall\CEng491\Ajax\gwt-windows-1.1.10\samples\DynaTable\www\com.google.gwt.sample.dynatable.DynaTable
○ Compilation succeeded

INFO] Starting HTTP on port 8888
```

http://localhost:8888/com.google.gwt.sample.dynatable.DynaTable/DynaTable.html

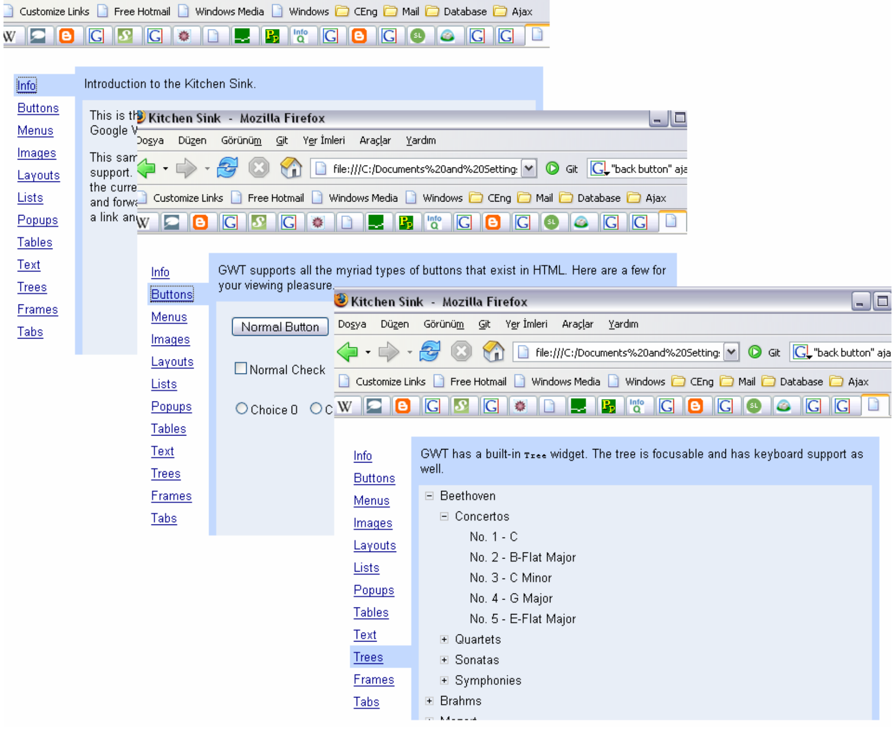
School Schedule for Professors and Students

Name	Description	Schedule
Inman Mendez	Majoring in Phrenology	Mon 9:45-10:35, Tues 2:15-3:05
Barney Crutcher	Majoring in Phrenology	
Omar Smith	Majoring in Geometry	Mon 4:30-5:20
Eddie Edelstein	Majoring in Basketball	Mon 8:15-9:05, Mon 2:00-2:50, Tues 12:15-1:05
Cathy Mendez	Majoring in Chemistry	Mon 1:15-2:05, Tues 1:00-1:50, Tues 2:15-3:05, Tues 3:30-4:20
Eddie Epps	Majoring in Geometry	

Figure 4 - GWT Hosted Mode

It opens the GWT browser with the application inside. Since hosted mode is run, the application is running in the Java Virtual Machine (JVM). This is typically the mode which is used to debug applications.

Running in web mode:



Introduction to the Kitchen Sink.

This is th
Google V

This sam
support.
the curre
and forw
a link an

GWT supports all the myriad types of buttons that exist in HTML. Here are a few for your viewing pleasure.

GWT has a built-in Tree widget. The tree is focusable and has keyboard support as well.

- Beethoven
  - Concertos
    - No. 1 - C
    - No. 2 - B-Flat Major
    - No. 3 - C Minor
    - No. 4 - G Major
    - No. 5 - E-Flat Major
  - Quartets
  - Sonatas
  - Symphonies
- Brahms

Figure 5 - GWT Web Mode

The GWT compiler will generate a number of JavaScript and HTML files from the Java source code. To see the application, the generated HTML file should be opened in any web browser. Since the project have been compiled, now it is running pure JavaScript and HTML that should work in IE, Firefox, or Safari.

3.5. JoyiStar WebShop

Another investigated AJAX IDE is WebShop. The JoyiStar WebShop is the AJAX web development tools released by JoyiStar Corporation. It provides an efficient OOP (object oriented programming) developing method to construct stable and convenient rich web applications based AJAX component.

The JoyiStar WebShop is a visualized modeling IDE which can be seen in Figure 6. It follows the OOP methodology and is constructed by many components which are useful to accelerate application development process.

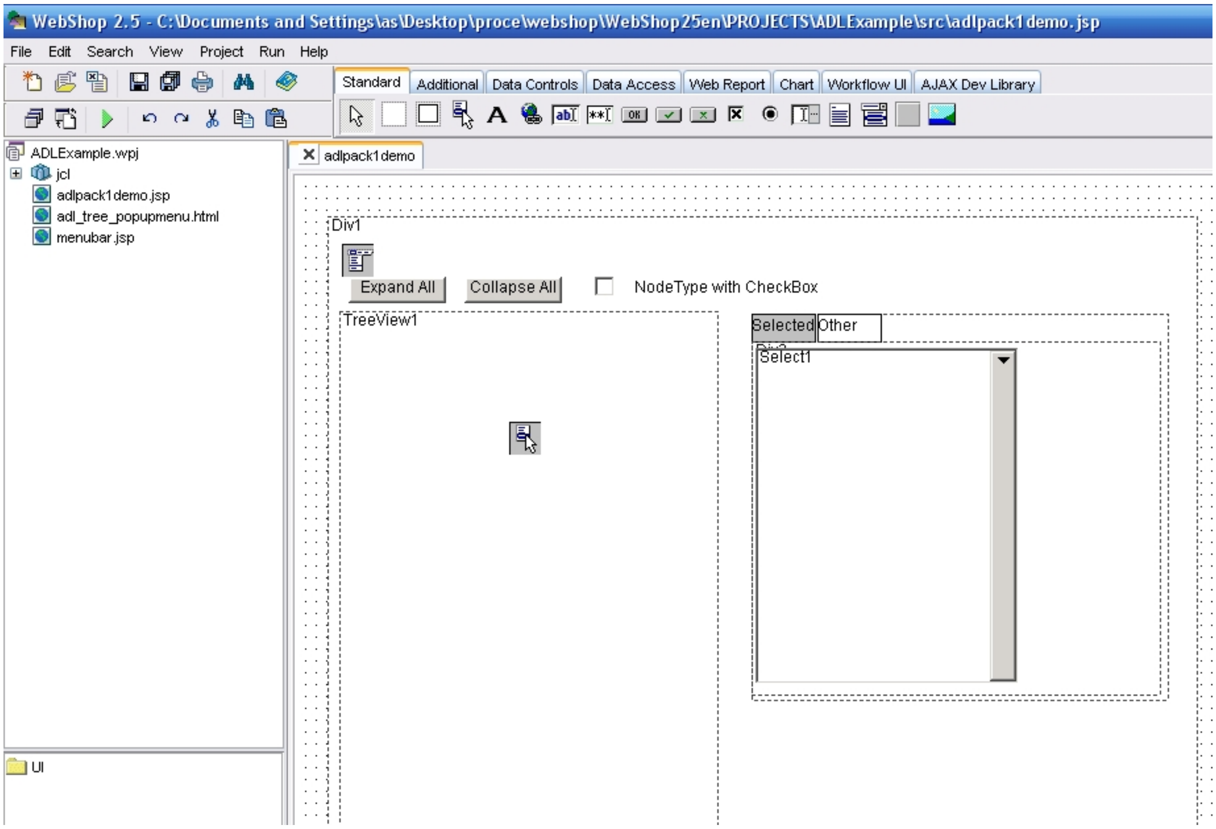
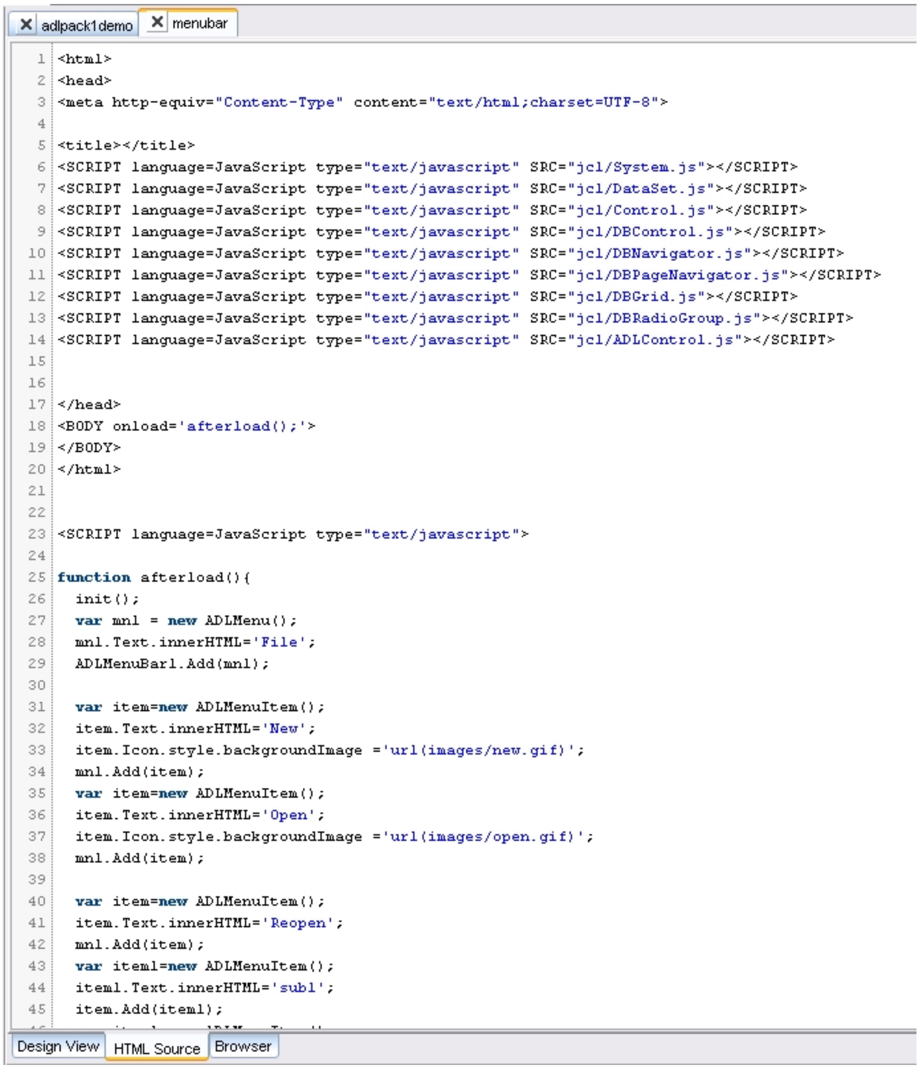


Figure 6 - JoyiStar WebShop GUI

WebShop IDE has three main window modes: Design view, HTML source, Browser. Each view has its own properties.

Design view lets the user to use drag and drop to add elements. Some of these elements are HTML elements, elements for database requests (linked to an event triggered by JavaScript) client side scripting elements. Those elements' properties (such as events or visual properties) can be set within this view. This view also allows the users to see what the finished page will look like roughly. The design view can be seen in Figure 6. Elements that can be added are visible in the upper portion of the image.

In HTML source view one can see the underlying source code of the current application. This view is similar to a standard text editor. It has syntax highlighting but doesn't have debugging and doesn't detect syntax errors. This view can be seen in Figure 7.



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">

<title></title>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/System.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/DataSet.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/Control.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/DBControl.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/DBNavigator.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/DBPageNavigator.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/DBGrid.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/DBRadioGroup.js"></SCRIPT>
<SCRIPT language=JavaScript type="text/javascript" SRC="jcl/ADLControl.js"></SCRIPT>


</head>
<BODY onload='afterload();'>
</BODY>
</html>


<SCRIPT language=JavaScript type="text/javascript">

function afterload(){
  init();
  var mnl = new ADLMenu();
  mnl.Text.innerHTML='File';
  ADLMenuBarl.Add(mnl);

  var item=new ADLMenuItem();
  item.Text.innerHTML='New';
  item.Icon.style.backgroundImage ='url(images/new.gif)';
  mnl.Add(item);
  var item=new ADLMenuItem();
  item.Text.innerHTML='Open';
  item.Icon.style.backgroundImage ='url(images/open.gif)';
  mnl.Add(item);

  var item=new ADLMenuItem();
  item.Text.innerHTML='Reopen';
  mnl.Add(item);
  var iteml=new ADLMenuItem();
  iteml.Text.innerHTML='subl';
  item.Add(iteml);
```

Figure 7 - JoyiStar WebShop HTML View

The last viewing mode is browser. This mode simply loads the current web page into an embedded Internet Explorer.

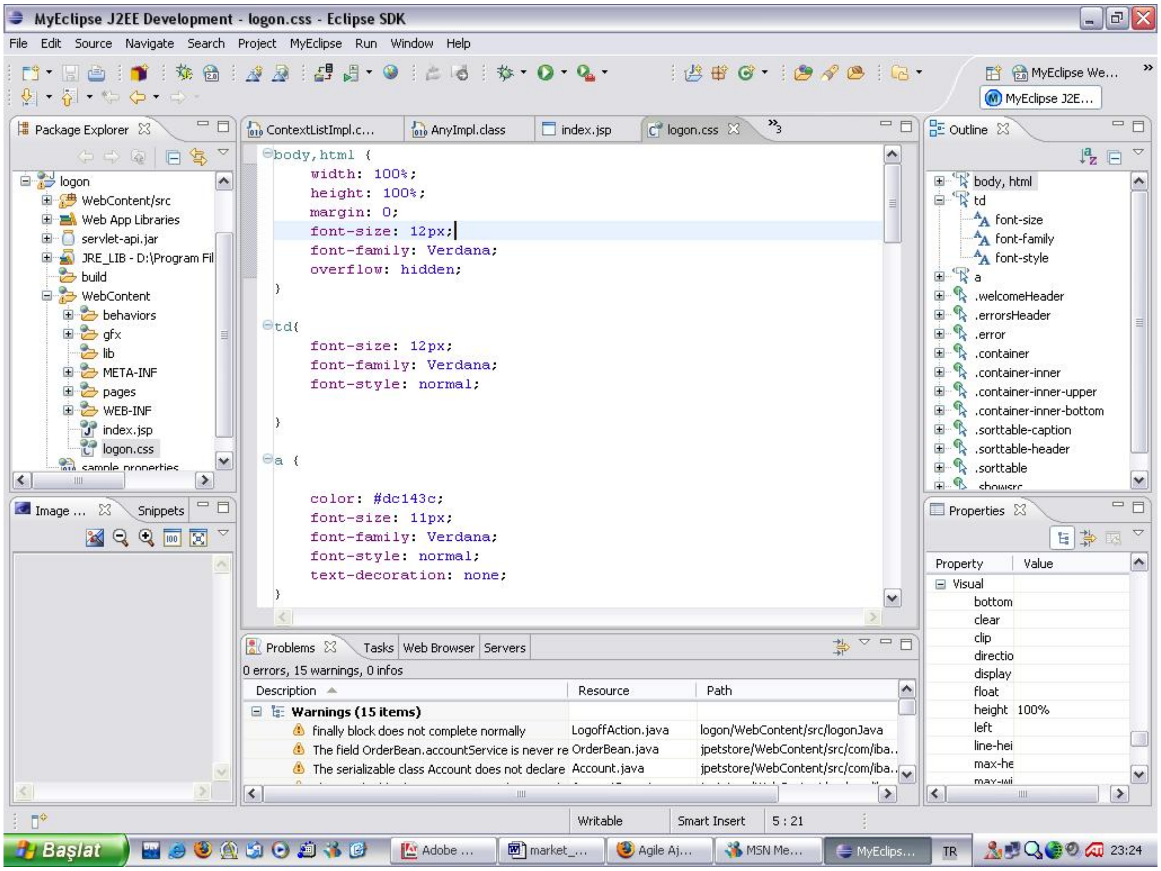
In WebShop you can create events related to HTML elements that use AJAX to connect to a web server to get data. The system generates JSP code to handle those events at server side. So in order to see the projects run in an external browser one must have tomcat (or another JSP server) installed in server. This way is the way we will use in our project but in our case we will try to broaden the supported code variety. Our main objective is to generate PHP code but we will try to add asp and JSP support too.

3.6. MyEclipse AJAX

Main features that “MyEclipse AJAX” supports are:

- UML diagram and modeling
- HTML, JSF, JavaScript, CSS deployment editors with code completion and syntax highlighting
- JSP source editor with syntax validation and debug support
- AJAX web tool with JavaScript editor, syntax highlighting, code assistance and client-side JavaScript debugger.
- XML and XSD editors with smart code completion, DTD caching for offline support and outline navigation viewer.
- Web Service support including project wizard, web services explorer and WSDL editor
- Database Explorer with SQL editor, Oracle Connector, MS SQL Server support.

Along other 8 products the most professional and detailed one is MyEclipse AJAX. It has executable files for three major Operating Systems (MS Windows, Linux and MAC OS) and many useful packages exist for all these operating systems. First of all it gives users all the tools needed at web application design. It has a powerful editor which can detect many programming and query languages and can perform syntax highlighting on codes of these languages. Users can create many different kinds of projects and there is project explorer tab which enables users to view project files in a tree hierarchy. But among all its useful and powerful features, MyEclipse AJAX does not support tools for high-level abstraction. In our opinion this product is only for professionals, it is too complex for a new web application designer. Figure 8 shows the HTML view and Figure 9 shows the Design View.



```
body,html {
     width: 100%;
     height: 100%;
     margin: 0;
     font-size: 12px;
     font-family: Verdana;
     overflow: hidden;
}

td{

     font-size: 12px;
     font-family: Verdana;
     font-style: normal;

}

a {

     color: #dc143c;
     font-size: 11px;
     font-family: Verdana;
     font-style: normal;
     text-decoration: none;
}
```

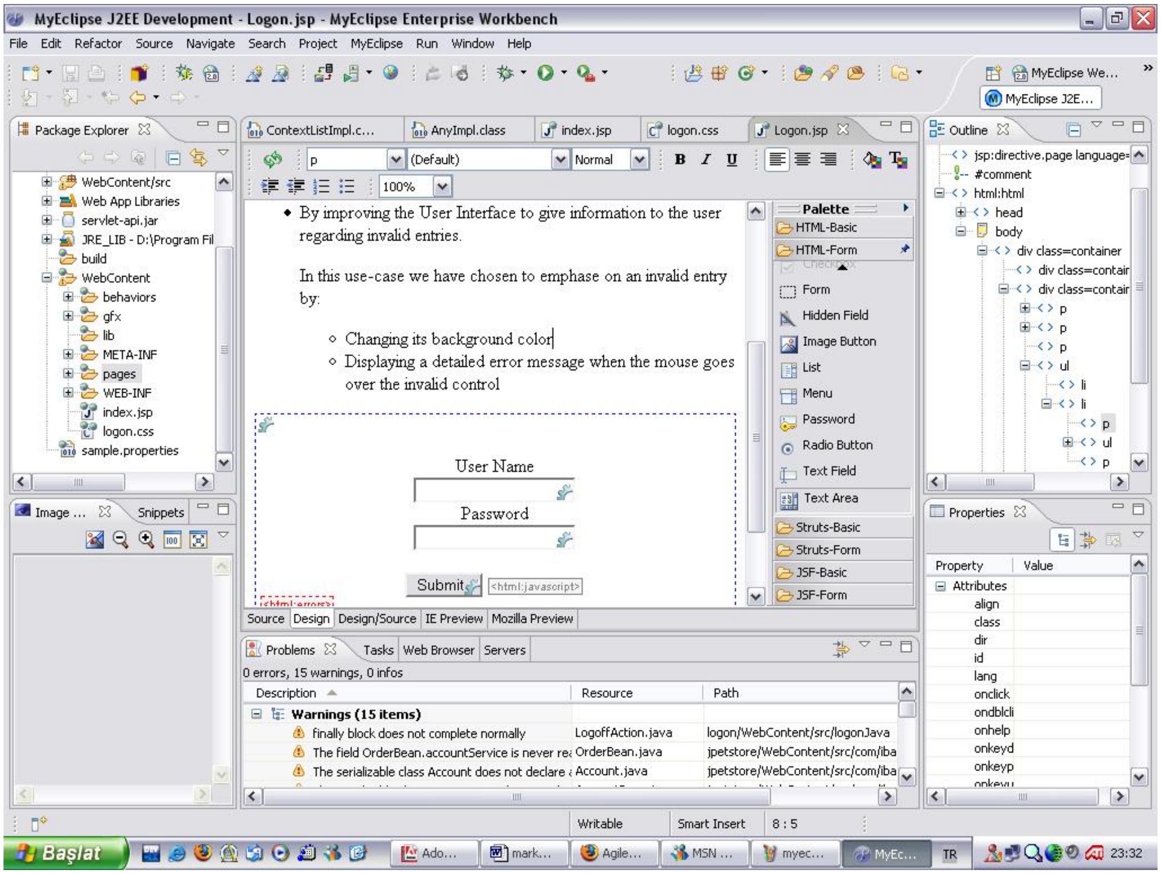
0 errors, 15 warnings, 0 infos

Description	Resource	Path
Warnings (15 items)		
finally block does not complete normally	LogoffAction.java	logon/WebContent/src/logonJava
The field OrderBean.accountService is never re	OrderBean.java	jpetstore/WebContent/src/com/iba..
The serializable class Account does not declare	Account.java	jpetstore/WebContent/src/com/iba..

Properties

Property	Value
Visual	
bottom	
clear	
clip	
directio	
display	
float	
height	100%
left	
line-hei	
max-he	
max-wi	

Figure 8 - MyEclipse AJAX HTML View



- By improving the User Interface to give information to the user regarding invalid entries.

In this use-case we have chosen to emphase on an invalid entry by:

- Changing its background color
- Displaying a detailed error message when the mouse goes over the invalid control

User Name

Password

Submit <html:javascript>

<html:errors>

0 errors, 15 warnings, 0 infos

Description	Resource	Path
Warnings (15 items)		
finally block does not complete normally	LogoffAction.java	logon/WebContent/src/logonJava
The field OrderBean.accountService is never re:	OrderBean.java	jpetstore/WebContent/src/com/iba
The serializable class Account does not declare a	Account.java	jpetstore/WebContent/src/com/iba

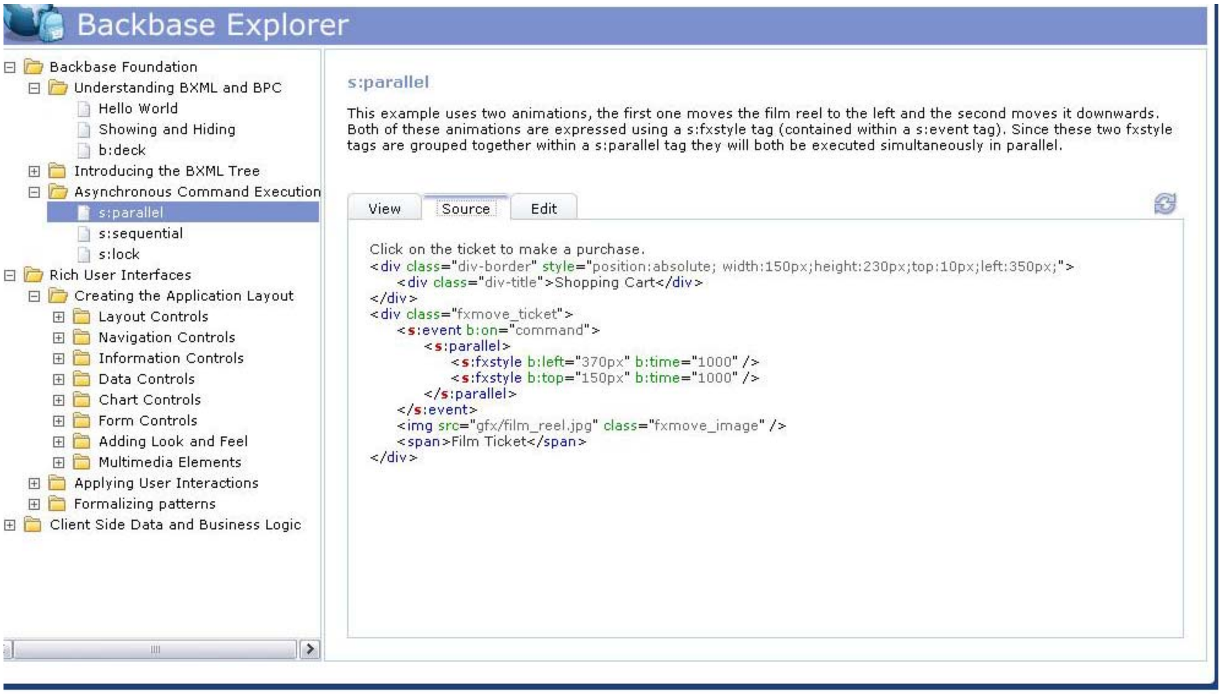
Figure 9 - MyEclipse AJAX Design View

3.7. Backbase Struts

Main features that "Backbase Struts" supports are:

- AJAX Engine
- Development Tools
- Server-side and client-side components
- 50 out-of-the-box AJAX widgets with their source codes
- Drag and drop, animations, back-forward button support, bookmarks, focus model, state management, data binding and template support for development and modeling process.
- AJAX Struts Editor provides a complete set of development tools within Eclipse and debugging tools within the web browser itself.

"Backbase Struts" is a well designed and powerful tool for AJAX developers. On the contrary of "MyEclipse AJAX", this plug-in is prepared for only AJAX development. Backbase Struts not only gives powerful features to users it also accommodates built-in widgets and their source codes. Another advantage over "MyEclipse AJAX" is user-friendliness. Simple and well-designed GUI makes users life easier and class hierarchies enables users to access information fast and easily. Figure 10 shows the HTML View and Figure 11 shows the Design View.

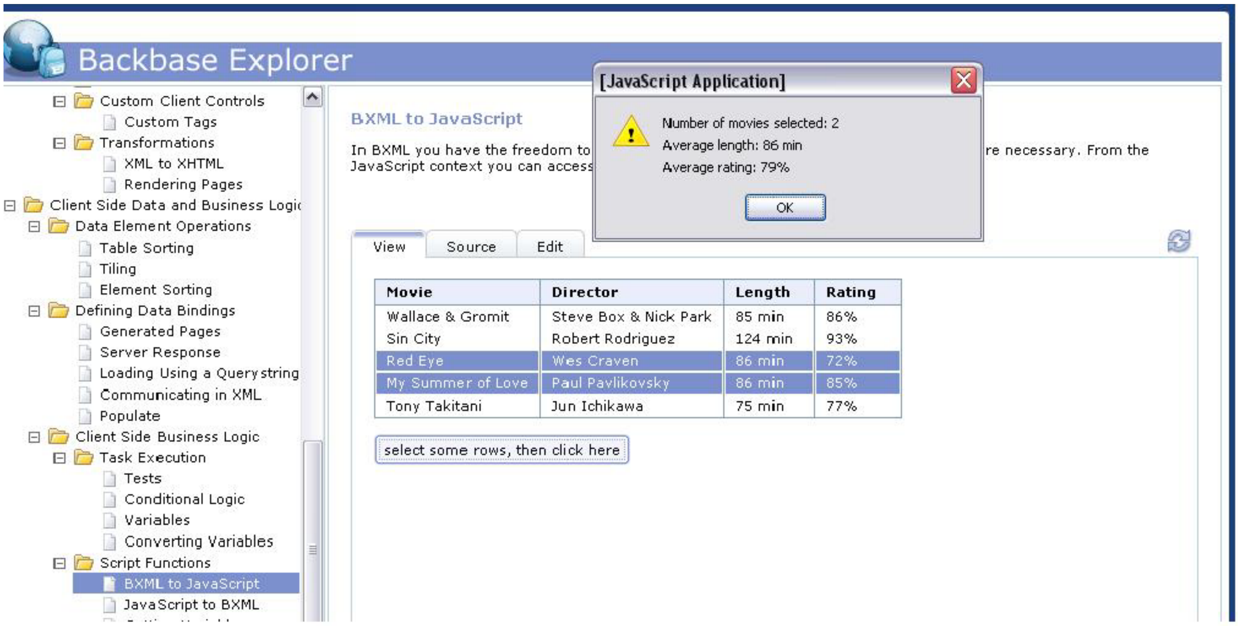


s:parallel

This example uses two animations, the first one moves the film reel to the left and the second moves it downwards. Both of these animations are expressed using a s:fxstyle tag (contained within a s:event tag). Since these two fxstyle tags are grouped together within a s:parallel tag they will both be executed simultaneously in parallel.

```
Click on the ticket to make a purchase.
<div class="div-border" style="position:absolute; width:150px;height:230px;top:10px;left:350px;">
    <div class="div-title">Shopping Cart</div>
</div>
<div class="fxmove_ticket">
    <s:event b:on="command">
        <s:parallel>
            <s:fxstyle b:left="370px" b:time="1000" />
            <s:fxstyle b:top="150px" b:time="1000" />
        </s:parallel>
    </s:event>
    <img src="gfx/film_reel.jpg" class="fxmove_image" />
    <span>Film Ticket</span>
</div>
```

Figure 10 - Backbase Struts HTML View



BXML to JavaScript

In BXML you have the freedom to re necessary. From the JavaScript context you can access

[JavaScript Application]

Number of movies selected: 2
Average length: 86 min
Average rating: 79%

Movie	Director	Length	Rating
Wallace & Gromit	Steve Box & Nick Park	85 min	86%
Sin City	Robert Rodriguez	124 min	93%
Red Eye	Wes Craven	86 min	72%
My Summer of Love	Paul Pavlikovsky	86 min	85%
Tony Takitani	Jun Ichikawa	75 min	77%

Figure 11 - Backbase Struts Design View

3.8. Comparison Chart

Features	Tibco	EchoStudio	GWT	WebShop	MyEclipse	Backbase	Lades
Standalone IDE	√			√			√
Eclipse Plug-in		√			√	√	
AJAX Framework			√				
Drag & Drop	√			√			√
JavaScript Debugging	√		√		√	√	√
Runtime Syntax Checking					√		
Syntax Highlighting	√	√		√	√	√	√
Built-in browser view		√		√		√	√
Built-in FTP Support				√			√
Database Support	√	√	√	√	√	√	√
Server Side Components					√		
Platform Independent		√	√		√	√	√
CVS Support							√

3.9. Questionnaire for Web Programmers

1. What are the main features you expect from an AJAX IDE?
2. If you have used an AJAX IDE, have you been satisfied by it? Why or why not?
3. Is WYSIWYG (what you see is what you get) property important for you?
4. What programming languages do you use when creating your AJAX code? Is wide variety of programming languages essential for you?
5. Do you want web service support from an AJAX IDE?
6. For debugging your source code, what kind of debugging utilities or features do you need?
7. Do you prefer more graphical but less flexible IDE or vice a versa? In other words "easiness and simplicity" or "powerfulness and complexness"?
8. Do you find tip boxes useful or annoying as Microsoft Office's Clippie?
9. Do you need commonly used JavaScript or html components built-in or prefer creating them in your way?
10. What features do you expect from the text editor you use while creating your AJAX code? Text - highlighting, find-and-replace, class hierarchy or etc?

3.10. Meeting with Web Application Developers

Our main customers are web application developers. We tried to get contact with both individual web developers and web development companies. We have prepared a questionnaire – which is in Appendix I – in order to find out current market status and market expectations. Our goal is to learn their needs and expectations from a web development IDE and design our project according to these specifications.

AJAX is a newly founded technology so it was difficult to find experienced web developers using AJAX. We tried to get touch with companies, we have sent e-mails indicating our purpose, and unfortunately we cannot take any response from them. We also tried to make an interview with Ahmet Saçan – who is the person prepares COW system for METU Computer Engineering Department –, but he told us he was not experienced in AJAX applications. Fortunately he told his expectations from a web development IDE and directed us to more experienced developers using AJAX.

At the interviews, we asked developers the role of AJAX in their applications. General answer was they have not used AJAX at all. They told us that they heard AJAX recently and liked the concept and thought that it would add a lot of functionalities to their applications, but they also told that they could not find the opportunity to learn much about AJAX. We concluded that web developers are eager to use AJAX and there is a need at market for graphical AJAX development environments.

We asked them what the main features they expect from a graphical web application development environment and from an AJAX IDE. Answers were generally similar to each other. Expectations can be classified into four main groups:

- Graphical Design: Nearly all participants of our questionnaire expect graphical tools such as drag and drop, built-in objects for frequently used stuffs like HTML forms, tables and etc.
- Textural Design: Seven of the interviewers, out of ten, expect a textural editor for design process. Expectations from text editors are auto indentation, text-highlighting, syntax checking and edit options like search, replace.
- Debugging: All of the participants expect debug utilities. They want an application that find out syntax errors and make suggestions to correct the errors. They also expect some more utilities like variable watch and partial run property.
- Simplicity: Most of the interviewers complained about lack of simplicity of current graphical development environments. They expect a simple GUI and simple ways to add event and options like creating AJAX event or establishing a database connection.

We asked them if they have used an AJAX IDE previously. Only two of ten participants have used an AJAX IDE when creating their web applications. We asked these two users why or why not they satisfied from these IDEs. They stated that they were generally satisfied with the IDEs, but they complained about bugs in these programs. They said these programs were beta versions generally and there were a lot of bugs in automatically generated codes.

We asked them is WYSIWYG (what you see is what you get) property important for them. They said that it would be a plus but it is not essential for them. So we decided to implement this property if we have time after essential properties as a package.

We asked the interviewers that what programming or scripting languages do they use at their applications and asked them wide variety of these languages are essential for them. Commonly used languages were PHP, Java, JavaScript, HTML and ASP. Interviewers stated that HTML, Java and JavaScript are essential for them. They said that at least one of PHP and ASP is must. So we decided to implement HTML, Java, JavaScript and PHP.

We asked them if they want web service support from an AJAX IDE. Three of ten interviewers asked for this property. We decided to not implement this feature because market need for this feature is much less when compared with hardness to implement it.

We asked the participants what they want from debug utility. Common answers were syntax error detection and correction help messages. They also asked for typical features like variable watch and partial execution.

We asked users to choose one of the followings:

- less flexible but easy to use a more graphical IDE
- more flexible but hard to use a more textural IDE
- balanced graphical and textural supported IDE

Interviewers' choices are not focused one of the options, number of peoples asked for each option were nearly equal. So we have to make a choice from them and balanced option makes more sense to us.

We asked the interviewers that if they find tip boxes useful or annoying as Microsoft Office's Clippie. All of the participants agreed that tip boxes are meaningless, they prefer reading help files to tip boxes. This common need encourages us to skip this feature.

We asked the web application developers that what features that they expect from text editor. Common needs were syntax-highlighting, powerful search and replace options, class hierarchy according to programming language, crash recovery, and variety of fonts and redo undo property.

4. Team Organization

4.1. Team Structure

Our team structure is democratic decentralized (DD). Democratic Decentralized model will provide us with an efficient working environment because of the flexibility provided by the model. This model ensures a high rate of cooperation since it forces us to communicate in decision making. In decision making coming to a total agreement will be the most important part. If any of the members opposes an idea and the decision is made against his/her opposition, he/she might take an aggressive stance against other members of the team because of his/her feel of humiliation. This feeling may grow into a large and unmanageable problem. To prevent these types of problems from arising total agreement is a must but to take precautions against extreme situations (such as permanent disagreement on a topic) we have chosen a leader on the contrary to DD. If the situation requires he will say the last word on a topic. Our chosen leader is Ali Demir.

4.2. Ground Rules

To work with high efficiency and motivation depends on agreeing some rules to provide this kind of environment. We have come to an agreement to some rules; these rules are as follows:

- Members will be well prepared before coming to project meetings.
- Late arrival to the meetings is restricted.
- Start and end times of meetings will be decided before and it will be obeyed strictly.

5. Process Model

In our project we will develop our work by analyzing, designing, making implementation, and testing. So it is obvious that we will develop our project as a step by step manner. Then as a team we have come together and have thought about the possible process models that we may use in our project. We know that the project phases are distinct. The requirements are well defined and understood. But there is a possibility that in some phases we may do mistakes and sometimes we may need to return to a specific phase and make changes. So during the project a feedback mechanism is needed. By the help of this information as a team we decided to apply “waterfall model with feedback” in our project.

“Waterfall model with feedback” suits our goal best, because the steps are distinct as we needed. In addition to this between each phase feedback loops exist so it allows us to make modifications easily.

6. Project Requirements

6.1. System Requirements

6.1.1. Hardware Requirements

Our hardware requirement is composed of three main groups; first one is development side requirements, second one is user side and the last one is server side requirements.

Developer Side

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

- P4 2.0 GHz or equivalent processor
- 512 MB RAM
- 40 GB HDD
- 32 MB Video Card
- Internet Connection (for testing purposes)

Computers of the project members meet the minimal requirements, so we don’t expect to face any difficulty in this area.

User Side

The user of our project “Lades” should a have computer capable of running several service programs if he/she wants to deploy his/her project locally (to his/her computer). Internet connection is required if the project will be deployed to a web server using ftp connection. Approximate minimal requirements, based on an assumption that the user will use local deployment, are as follows:

- P4 1.6 GHz or equivalent processor
- 512 MB RAM
- 40 GB HDD
- Internet Connection (for external deployment)

Server Side

Our server component is a typical web server. It will run an Apache web service with PHP support. PHP will have embedded MySQL libraries. According to users’ choice the server running Apache may run MySQL server component too. This will increase the strain on the server thus will increase minimal requirements. The minimal requirements for full web service support are as follows:

- P4 3.2 GHz or equivalent processor
- 1024 MB RAM
- 100 GB HDD
- Internet Connection (broadband connection preferred)

6.1.2. Software Requirements

Our software requirement is composed of three main groups; first one is development side requirements, second one is user side and the last one is server side requirements.

Developer Side

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

Documentation Tools

During the first term, we will be using several documentation tools extensively. Our choices for word processing are Microsoft Office Word combined with Adobe Acrobat Professional (for converting to PDF), or OpenOffice 2.0 Writer. We will usually use MS Word.

SmartDraw will be our primary drawing tool for diagrams because of its ease of use. In requirement analysis report we have drawn data flow, state transition, use case diagrams with SmartDraw. Gantt chart will be drawn with Microsoft Project.

Development Tools

For our project to be platform independent we have chosen Java as our programming language. In order to program in Java we need Java Development Kit (JDK). JDK is composed of two main parts; Java Runtime Environment (JRE) and command line development tools (such as compilers, debuggers, applets... etc). We will mainly use some Integrated Development Environments (IDE) to develop Lades in Java. These IDEs will be NetBeans and Eclipse. These IDEs are mostly interchangeable.

To test our project we need the web services that will be used in the web project that was created with Lades. These are Apache with PHP support and MySQL server.

User Side

For the user to run our project he/she will need JRE preinstalled in his/her computer Apache, PHP and MySQL must be present in his/her system if he/she chooses to deploy his/her project locally.

Server Side

Required software for the server side to operate correctly will be Apache, PHP and MySQL.

6.2. User Requirements

6.2.1. Use Case Diagram

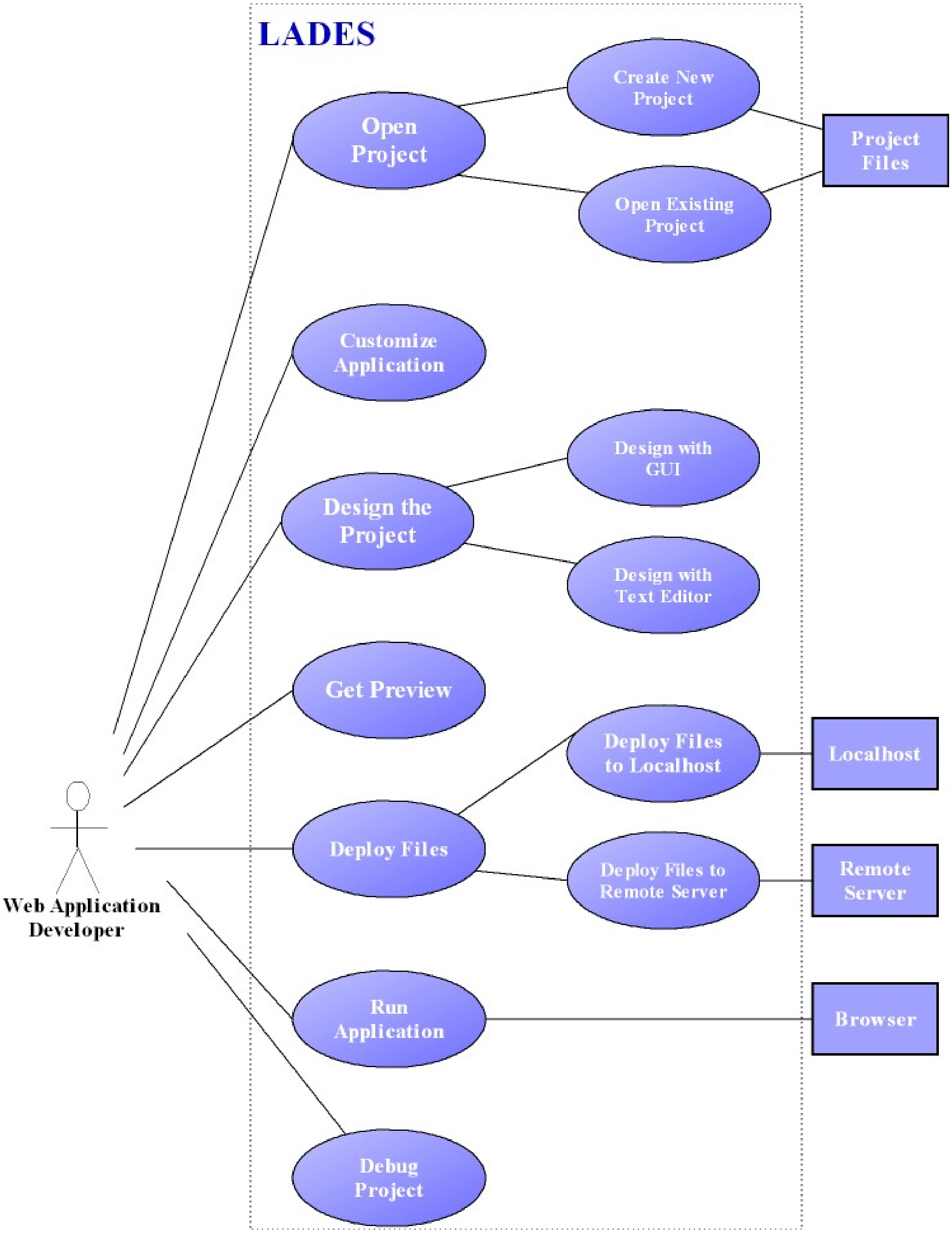


Figure 12 - Use Case Diagram

6.2.2. Functional Requirements

6.2.2.1. User Functionality

For LADES Project we have only one user. The user is the web application developer. This user has several abilities to develop the application. The general functionalities are as the following.

Use Case: Open Project

Primary Actor: Web application developer.

Goal in context: To develop a web application with LADES IDE, the first step is to open a project.

Preconditions: To open a project in LADES IDE the program should be installed completely and correctly, and the specified requirements should be satisfied.

Scenario:

1. By using icons or menus the user is able to open a project.
2. Related to the type of choice (the user opens an existing process or create new project), the system displays related file dialog box.
3. The web application developer makes necessary arrangements.
4. The web application developer clicks ok button at the file dialog box.
5. Then project is formed.

Exception: Unrelated type of files with the LADES IDE causes exception.

In open project the user have two choices. One is to create a new project; the second one is to open an existing project.

Use Case: Create New Project

Primary Actor: Web application developer

Goal in Context: To create a new project for LADES program for web applications.

Preconditions: Open the LADES IDE program.

Scenario:

1. After open the LADES program the user clicks the menu tab from the menu bar.
2. The web application programmer clicks “New” item from this tab.
3. Then a file dialog box is appeared.
4. The web application developer selects the location to save the new project.
5. The web application developer writes the name of the project.
6. The web application programmer clicks “Ok” button on the file dialog box.
7. Then the system creates an empty project for the user.

Use Case: Open Existing Project

Primary Actor: Web application developer

Goal in Context: To open an existing project with the LADES program.

Preconditions: Open the LADES IDE program.

Scenario:

1. After open the LADES program the user clicks the menu tab from the menu bar.
2. The web application programmer clicks “Open” item from this tab.
3. Then a file dialog box is appeared.
4. The web application developer moves through the location of the existing project.
5. The web application developer selects the project to open.

6. The web application programmer clicks "Ok" button on the file dialog box.
7. Then the system opens the project for the user.

Exceptions:

1. If the user tries to open a file which is unrelated with LADES IDE then an error message is displayed to the screen to warn the user.

<u>Use Case: Customize Application</u>

Primary Actor: Web Application Developer

Goal in Context: To customize LADES application including text editor options like background and text-highlight color, button arrangement, font and page preferences, and keyboard shortcuts and etc.

Preconditions: Necessary packages must be installed a priori.

Trigger: In the first execution following the installation, system automatically starts "Customize Application" process.

Scenario:

1. When Web Application Developer executes "Customize Application" process by clicking buttons or pressing keyboard shortcuts.
2. "Customize Application" process displays all possible functionalities of selected feature.
3. Web Application Developer selects or changes preferences displayed in the menu.
4. Web Application Developer confirms its selection by clicking confirm button.
5. System changes its configurations and change display mode.

Exceptions:

1. Changes are not possible values for the system – system displays appropriate error message and cancels the user request.

2. Web Application Developer selects "change to graphical design environment or textural design environment" – see use case: "Design with GUI" or "Design with Text Editor".

Use Case: Design the Project

Primary Actor: Web application developer

Goal in Context: To develop the web application, dragging dropping web components, by adding AJAX events, arranging properties of web components, creating code with text editor.

Preconditions: A new project should be created or an existing project should be opened.

Trigger: During the development process, most of the time the web application developer will be in this functionality such as the project id opened with this and when the user presses visual design tab or text editor tab.

Scenario:

1. The web application developer adds web components.
2. The web application developer adds AJAX events.
3. The web application developer adds code to the project.

Use Case: Design with GUI

Primary Actor: Web application developer

Goal in Context: To develop the web application, to add web page components to the web application, to specify the properties of selected items, to add AJAX event to the application, to add database connection to the application.

Preconditions: A new project should be created or an existing project should be opened.

Triggers: The web application developer decides to develop web application with the GUI and presses the graphical design tab on the screen.

Scenario:

1. The web application developer presses the graphical design tab.
2. The web application developer drags items from the items toolbar.
3. The web application developer drops the dragged item on the graphical design tab.
4. The system displays the dropped items on the design tab.
5. The web application developer adds database connection to a specified database.
6. The web application developer changes the location of the web page components.
7. The web application developer changes the properties of the selected web page components.
8. The web application developer adds AJAX events from event editor.

Use Case: Design with Text Editor

Primary Actor: Web application developer

Goal in Context: To change the generated code by the program, or to add extra functionalities to the application.

Preconditions: A new project should be created or an existing project should be opened.

Trigger: The web application developer decides to generate code with text editor and clicks the text editor tab.

Scenario:

1. The web application developer presses the text editor tab.
2. The web application developer adds functions to this area.

Use Case: Get Preview

Primary Actor: Web Application Developer

Goal in Context: To enable Web Application Developer test his/her application in integrated web browser: Mozilla, Firefox

Scenario:

1. Web Application Developer creates web application by using graphical or textural development environment.
2. Web Application Developer change view mode to preview.
3. System changes display mode to preview mode.
4. System displays integrated Mozilla Firefox preview of supplied web application.

Exceptions:

1. Supplied code can be incompatible to view in web browser – system display parts it can distinguish and throws an appropriate error message.
2. Only static parts of the web application (html objects, not JavaScript, flash or database connections) are displayed.

Use Case: Deploy Files

Primary Actor: Web Application Developer

Goal in Context: To enable Web Application Developer to put and save project files in appropriate and desired locations at local or remote machine. Remote machine uploads will be done via ftp.

Preconditions: Ftp connection package must be installed a priori.

Trigger: Web Application Developer clicks “save file” or “upload file” button in menu bar or pressing keyboard shortcut to trigger “Deploy Files” process.

Scenario:

1. Web Application Developer creates web application by using graphical or textural development environment.
2. Web Application Developer selects save or upload file property by clicking

appropriate button or pressing keyboard functions.

3. System displays default location to save or upload file. Web Application Developer can change the location to save or upload file and confirms the location.
4. If save to local machine option is selected, system saves the file to local machine's given location.
5. If user select upload to remote machine option, system creates a ftp connection to that remote machine. After establishing successful ftp connection to remote machine, system uploads file to selected location.
6. System displays successful or error message to Web Application Developer.

Exceptions:

1. User does not have necessary rights for saving file – file cannot be saved to given location and an error message is displayed.
2. Ftp connection cannot be established with remote machine – an error message indicating connection problem is displayed.
3. During file transfer with via ftp connection, connection can be lost due to external problems like Ethernet cable can be unplugged. In this case file upload cancels and an error message is displayed.
4. Web Application Developer can enter wrong file names – like containing '”' character or prep reserved file names like “con” in MS Windows –. In this situation system asks Web Application Developer to enter a correct file name.

<u>Use Case: Run Application</u>

Primary Actor: Web application developer

Goal in Context: To run the created web application program

Preconditions: All the necessary files should be deployed to the local server or the remote server. And one of the Mozilla or Internet Explorer browsers should be installed on the local computer.

Trigger: The web application developer decides to run the developed application.

Scenario:

1. The web application developer presses the run button to run the process.
2. The system does the syntax checking.
3. The system establishes connection with the default browser on the local machine.
4. The deployed files are run on the local browser.

Exceptions:

1. When there is a syntax error on the user generated code, the application will not be displayed on the default browser else an error message will be displayed on the screen and the system asks the user if he/she wants to debug the code (see use case debug project).

Use Case: Debug Project

Primary Actor: Web application developer

Goal in Context: To debug the files of an existing project.

Preconditions: There should be an open project to debug.

Trigger: If an error occurs just before run process and the user wants to debug the files. Or the user clicks the debug button on the menu bar.

Scenario:

1. The user clicks the debug button on the menu bar.
2. The syntax errors on the text editor part are displayed to the user by the system.
3. The user corrects the parsing errors.
4. The user again presses the debug button.
5. The system informs the user about successful debugging process.

6.3. Non-Functional Requirements

Usability:

Usability is one of the important concepts in our LADES program. For the reason that web application developers always prefer to use IDE programs that are easy to use we will provide this issue in our project. User friendliness is especially is a significant concept in designing IDEs. Providing users comfortable and easy usable IDE is our goal. Hence, the graphical user interfaces that we will develop in our application will be very easy to understand and use.

Portability:

Being portable is a significant concept for files that are generated with an IDE. As our LADES program will support platform independence the files that are generated with LADES program will work both on Windows and Linux operating systems. Beside this, as our program will be JAVA based, we will provide portability for the LADES users.

Reliability:

Reliability of a program is a significant issue, since users want the program that they use be reliable. For a system it is important to perform its features in a stable manner. Mainly, after implementation we will try to make our LADES program bug free. To provide this we will test our application carefully. Therefore we will prevent the program to fail and destroy the existing files of LADES users because of a simple bug.

7. Modeling

7.1. Functional Modeling

7.1.1. Data Flow Diagram

7.1.1.1. DFD Level0

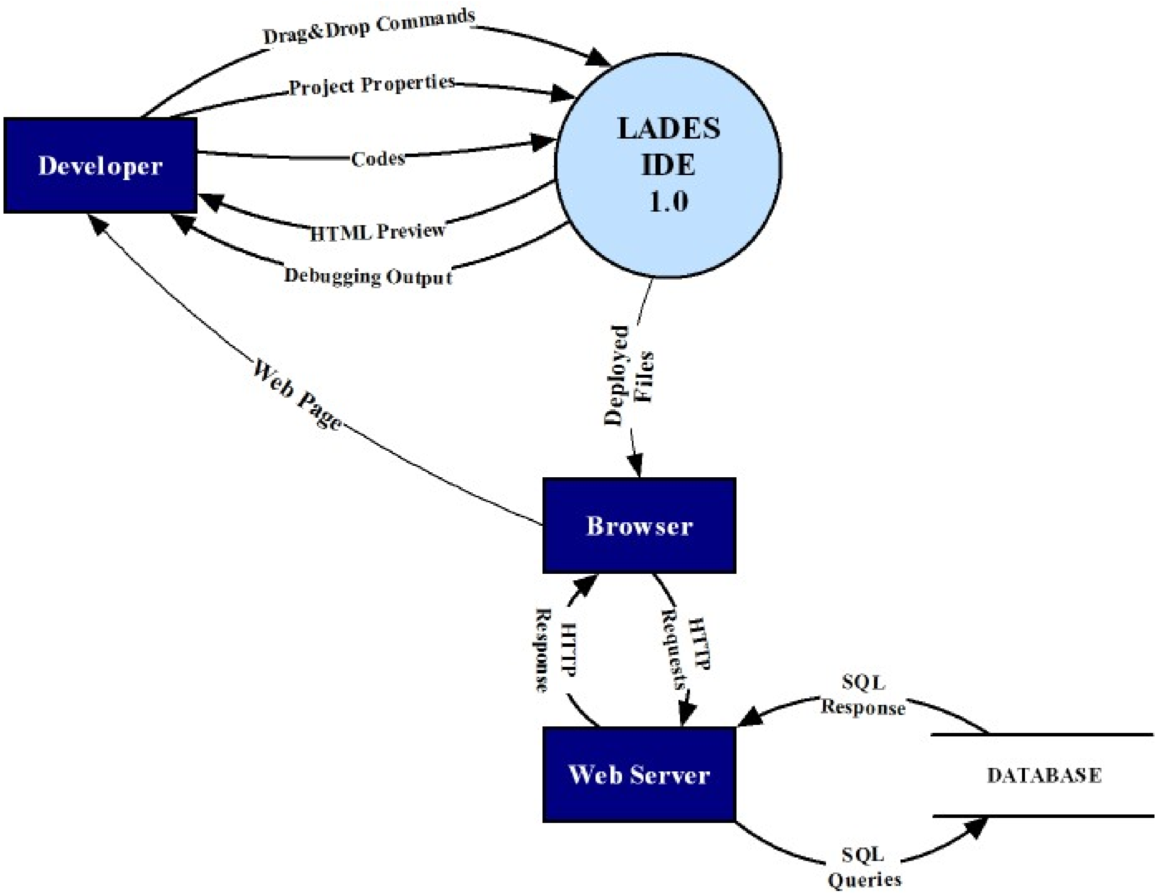


Figure 13 - DFD Level0

7.1.1.2. DFD Level1

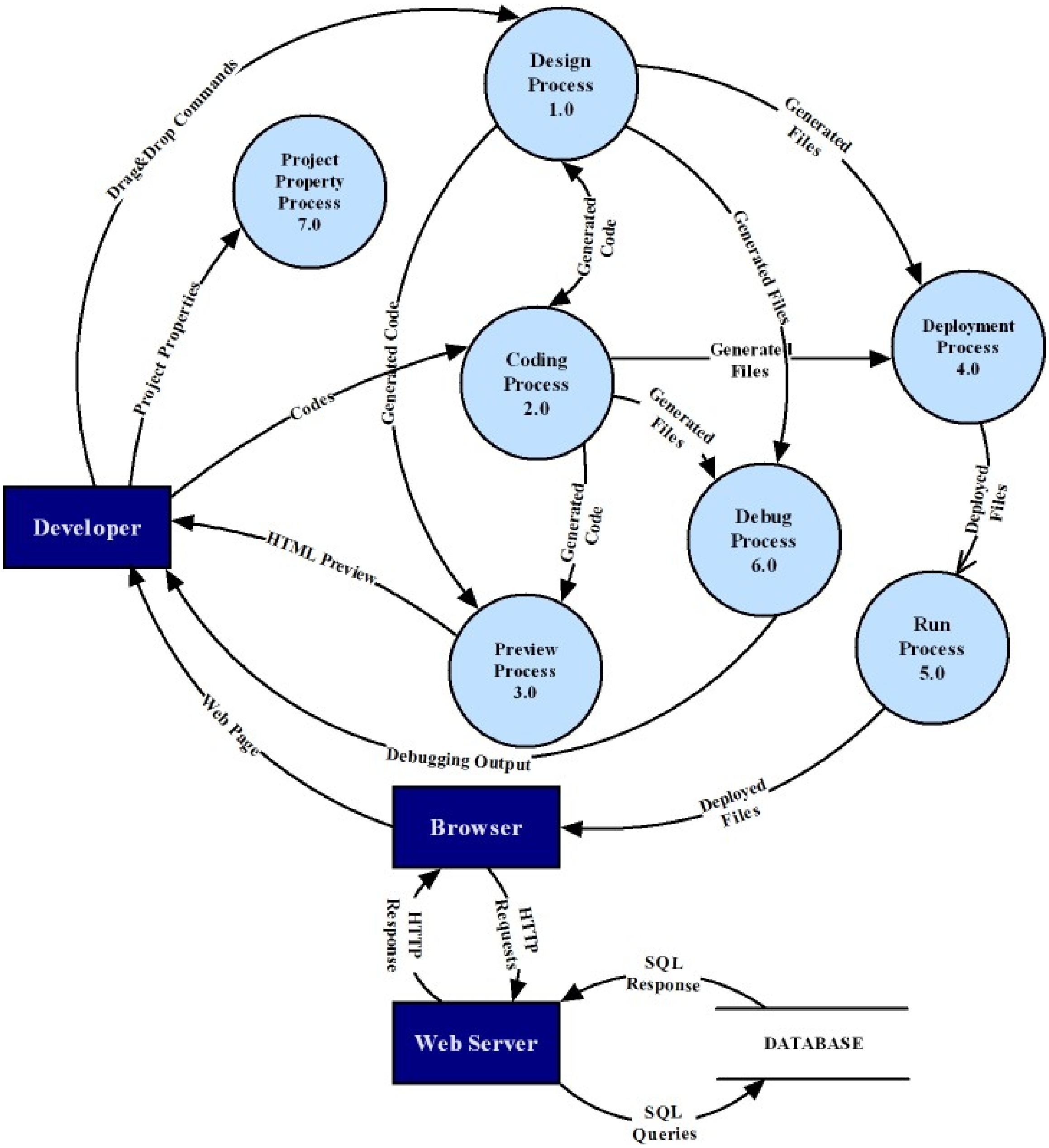


Figure 14 - DFD Level1

7.1.1.3. DFD Level2

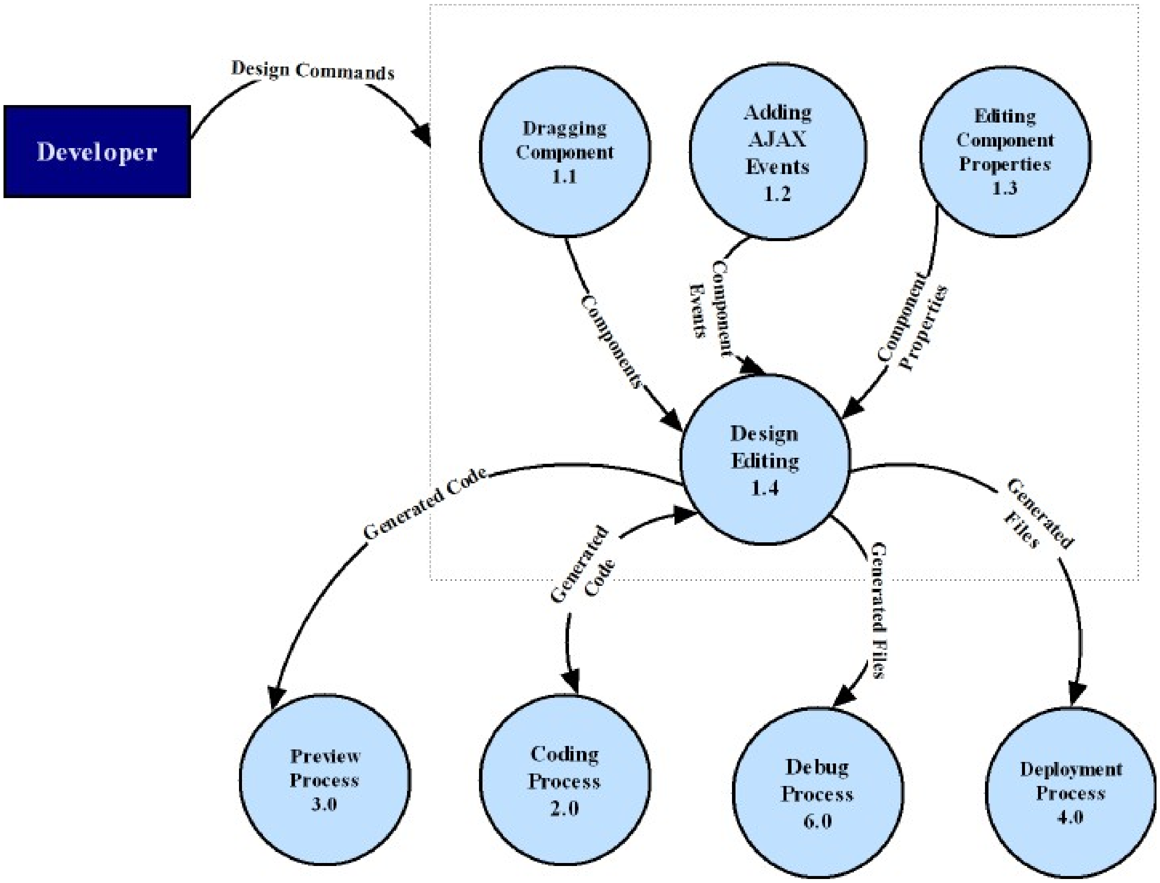


Figure 15 - DFD Level2 of Design Process 1.0

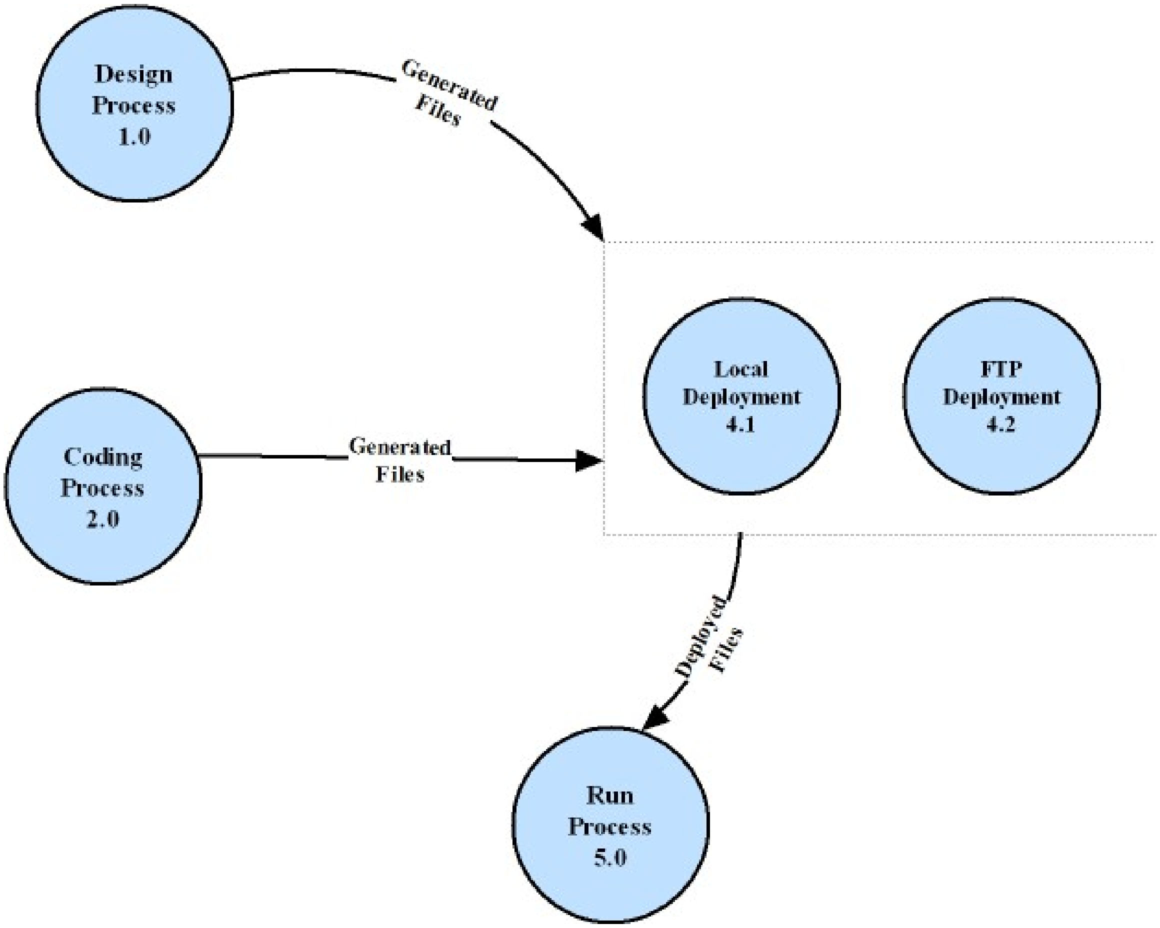


Figure 16 - DFD Level2 of Deployment Process 4.0

7.1.2. Data Dictionary

name:	Codes
where used / how used:	Codes are *input* to Coding Process and typed by User This data are validated and processed in coding Process such as syntax hightlighting and shows available options such as when a tag as opened a closing tag will be added and cursor will be moved to between tags.
Description:	codes = string

name:	Component

Where used / how used:	Component is *input* to Design Editting Process (1.4) coming from Dragging Component(1.1). General name of web page components which are used to create form with buttons, checkboxes or to add images or to connect to database or to add table.
Description:	Codes = string

name:	Component Event
Where used / how used:	Component Event is *input* to Design Editting Process(1.4) coming fromAdding AJAX Events(1.2). This Events are used to provide asynchronization which are used in onChange or onSelect JavaScript events.
Description:	Code

name:	Component Properties
where used / how used:	Component Properties are formed in Editting Component Properties Process(1.3) and given to the Design Editting Process(1.4). This information varies according to the component for example if component is used for database connection the info will includes properties such as an endpoint address of the database connection, username and password .If it is HTML form element, it could include aligment, size, color etc.
Description:	Form Element Property = size+alignment+color Database = database address + username + password +schema name

Name:	Deployed Files
where used / how used:	Input to Run Process and Browser. Output from Run Process and Deployment . It is a ready to run file which is either on localhost or remote web server
Description:	HTML or PHP File

Name:	Debug Output
where used / how used:	Output from Debug Process shown to the user. In case of the error, debug output shows syntax error information.
Description:	Error message | Well Done

name:	Drag & Drop Commands
where used / how used:	Design Process(1.0) *input* given by User It includes drag and drop functionality output by the help of these commands design window and generated code can be formed.
Description:	DragCommand = Name of web page component + Name of the part where it is dragged Drop Command= Name of the web page component + Name of the part where it is dropped

Name:	Generated Codes
where used / how used:	Input to Design Process and Coding Process and Preview Process. Output from Desing Process and Coding Process It is used to show any output the user, such as preview, code or design of web page
Description:	This very generalized data name.It consists of variety of elements.

Name:	Generated Files
where used / how used:	Input to Debug Process and Deployment Process. Output from Desing Process and Coding Process It is different than the generated code by the way that the it is not part of any code, the completed file are deployed or debugged
description:	HTML or PHP File

Name:	HTTP Request
where used / how used:	Input to Web server Output from Browser. It is automatically generated by browser when the user want to run the web page.
description:	HTTPRequest

Name:	HTTP Response
where used / how used:	Input to Browser. Output from Web server It is response to requester via webserver.
description:	The HTTP response typically consist of an HTML document, but can also be a row-text file, an image or some other type of document; if something bad is found in client request or while trying to serve the request, a web server has to send an error response which may include some custom HTML or text messages to better explain the problem to end users

Name:	Project Properties
where used / how used:	Project Property Process (7.0) *input* given by User This data is used as to adjust the IDE prefererences and view to customize the user requirements and habits such as auto-indent and syntax highlight, show class hiearchy.
description:	This very generalized data name.It consists of variety of elements.

Name:	SQL Queries
where used / how used:	Input to database(7.0) Output from web server
description:	SQL query

Name:	SQL Response
where used / how used:	Input toweb server Output from database(7.0)
description:	SQL query

Name:	Web page
where used / how used:	Result of the project shown in browser
description:	HTML or PHP file

7.1.3. Process Specification

Design Process 1.0

Users of LADES have two options to generate HTML code:

- By using the design window,
- By using the text editor.

Design process provides several options the user in order to generate HTML code. These are as follows:

- Design window has the capability of displaying current active components on the webpage. User will be able to drag these components through the design window in order to change their positions.
- Webpage components like form elements (checkboxes, text fields, buttons…), database connection elements (database connections, SQL queries), charts etc can be added from the "Webpage Components Tree" by using drag and drop.
- Several component events can be selected from the "Events Editor" section. These events include JavaScript events and AJAX related events.
- Component properties can be edited through the "Properties Editor". This editor also includes built-in CSS editor.

Although design process generates the required code, user will always be able to this code through the text editor part of LADES which is explained in next section.

Coding Process 2.0

Users have another option to generate the code, by using the text editor LADES provides. This process and design process works in collaboration with each other. Users may write some code in text editor and switch to design process. Design process processes these codes and displayed the webpage components in its window. Then users may use the functionality design process provides. Vice a versa users can switch from design process to code process.

Preview Process 3.0

The generate code from design process and code process can be viewed by preview process. This process uses an embedded browser (Mozilla Firefox 2.0) to display the created webpage. Users may preview their web pages without deploying the files and opening and external browser.

Deployment Process 4.0

Generated files can be deployed on a local computer or on a remote web server. This process provides this functionality. Remote deployment uses the FTP protocol.

Run Process 5.0

Created webpage can be seen on an external browser at any time. This external browser can be selected through LADES preferences. By this process, user can see the output on any browser unlike the embedded browser support. It will have two different modes to display the webpage.

- Displaying on an external browser using the absolute file path
- Displaying on an external browser by using local deployment and running on the local web server.

Debug Process 6.0

LADES project supports debugging of JavaScript and PHP. Debugging support may detect syntax errors and watching variables.

Project Property Process 7.0

Several IDE options can be changed through this process. Hotkeys, default project settings and text editor settings like syntax highlighting, viewing line number, etc can be changed by the Preferences window.

Dragging Component Process 1.1

There is the list of draggable items. Dragging components process takes the design commands from the developer, forms the web components, after that this process sends related components to “Design Editing Process 1.4”. By the help of this process developer can be able to add drag and drop web page components to web application easily.

Adding AJAX Event Process 1.2

As the web developer will be able to add AJAX events to strength the functionality and efficiency of program, the Adding AJAX Events process takes the design commands from the user, forms related events and sends the component events to the “Design Editing Process 1.4”.

Editing Component Properties Process 1.3

Design commands are taken from the developer. Developer can edit the properties of the web page components, and when the editing component properties process takes the design commands from the user, send the component properties to the “Design Editing Process 1.4”.

Design Editing Process 1.4

Design Editing Process takes drag and drop components from dragging component, component events from “Adding AJAX Events Process 1.2”, and component properties from “Editing Component Properties Process 1.4”. Design Editing Process displays the formed web components on the screen. This process also supports to change the location of the web form components. Codes and files are generated according to the gathered components from the related processes and the design editing process sends generated codes to “Preview Process 3.0”, sends generated codes to “Coding Process 2.0”, sends generated files to “Debug Process 6.0”, and again sends generated files to “Deployment Process 4.0”.

Local Deployment Process 4.1

If the developer prefers to run the process on local computer, “Local Deployment Process 4.1” takes place. This process takes the generated files from the “Design Process 1.0” as well as from the “Coding Process 2.0”, and then deploys the files to local web server root.

FTP Deployment Process 4.2

If the user wants to deploy the generated files to a server, “FTP Deployment Process 4.2” starts to perform. This process gets the generated files from the “Design Process 1.0” and from the “Coding Process 2.0”, and then deploys these files to remote web server root by using the FTP protocol.

7.2. Behavioral Modeling

7.2.1. State Transition Diagram

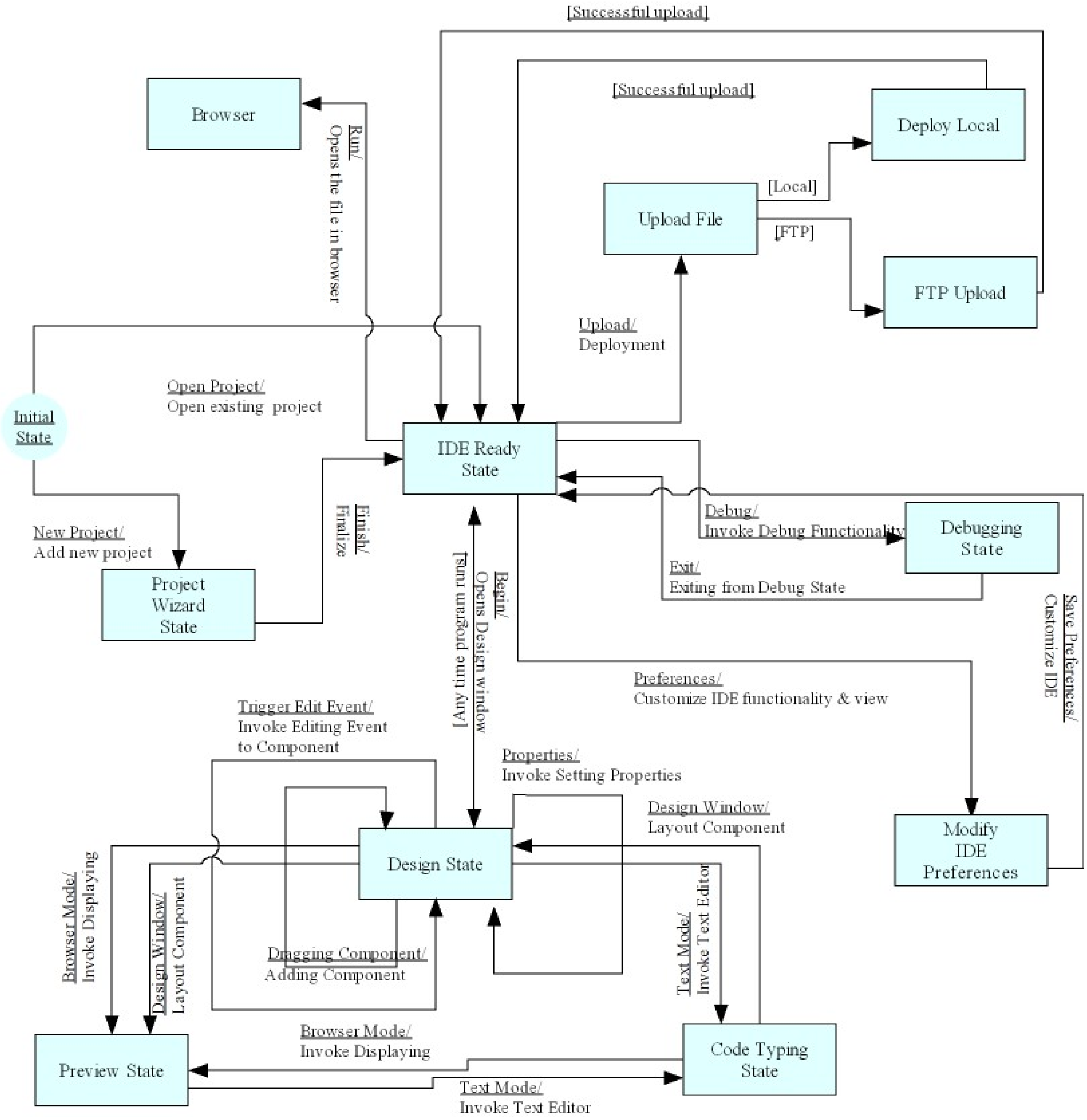
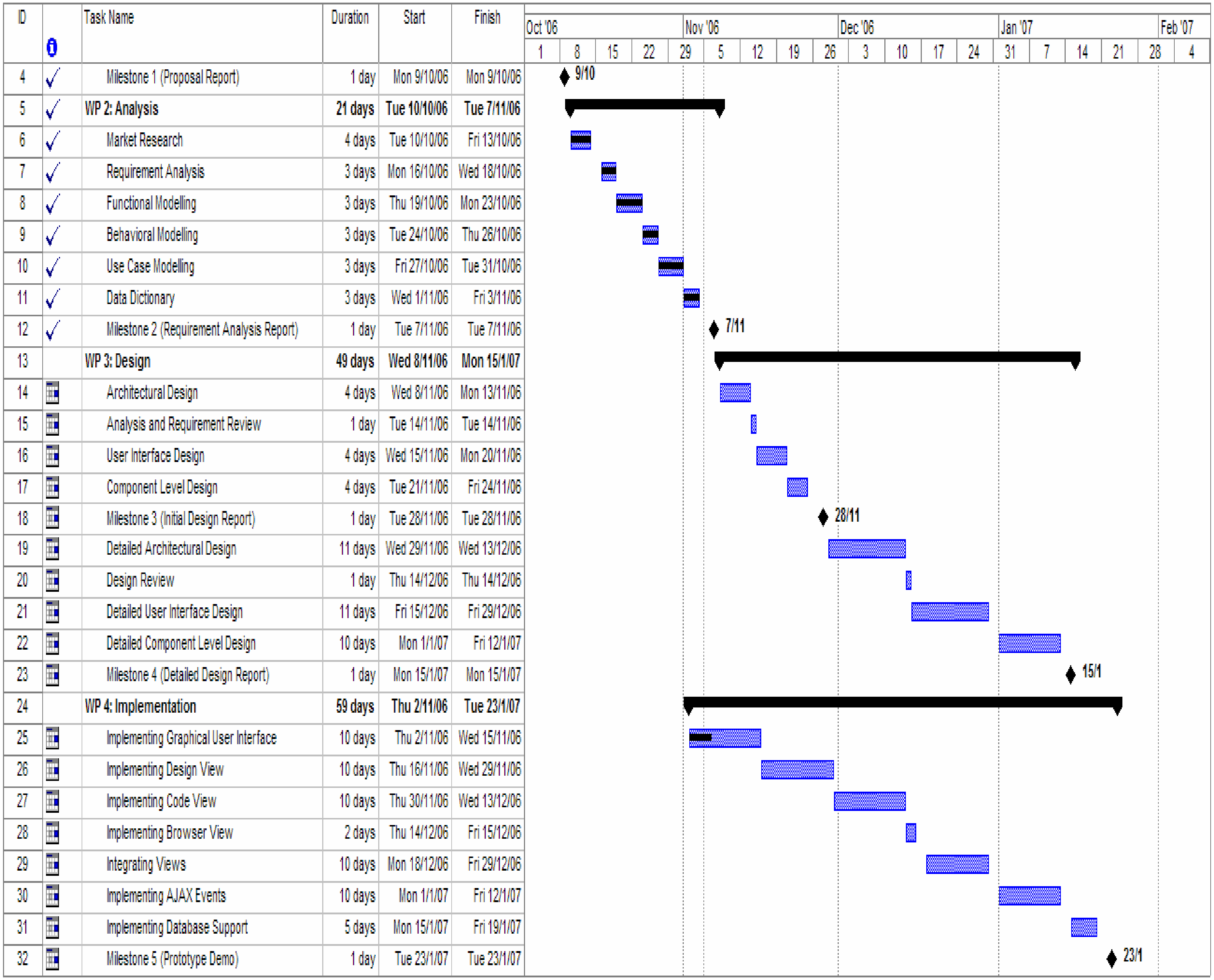


Figure 17 - State Transition Diagram

8. Project Schedule



ID	ⓘ	Task Name	Duration	Start	Finish
4	✓	Milestone 1 (Proposal Report)	1 day	Mon 9/10/06	Mon 9/10/06
5	✓	**WP 2: Analysis**	**21 days**	**Tue 10/10/06**	**Tue 7/11/06**
6	✓	Market Research	4 days	Tue 10/10/06	Fri 13/10/06
7	✓	Requirement Analysis	3 days	Mon 16/10/06	Wed 18/10/06
8	✓	Functional Modelling	3 days	Thu 19/10/06	Mon 23/10/06
9	✓	Behavioral Modelling	3 days	Tue 24/10/06	Thu 26/10/06
10	✓	Use Case Modelling	3 days	Fri 27/10/06	Tue 31/10/06
11	✓	Data Dictionary	3 days	Wed 1/11/06	Fri 3/11/06
12	✓	Milestone 2 (Requirement Analysis Report)	1 day	Tue 7/11/06	Tue 7/11/06
13		**WP 3: Design**	**49 days**	**Wed 8/11/06**	**Mon 15/1/07**
14	▦	Architectural Design	4 days	Wed 8/11/06	Mon 13/11/06
15	▦	Analysis and Requirement Review	1 day	Tue 14/11/06	Tue 14/11/06
16	▦	User Interface Design	4 days	Wed 15/11/06	Mon 20/11/06
17	▦	Component Level Design	4 days	Tue 21/11/06	Fri 24/11/06
18	▦	Milestone 3 (Initial Design Report)	1 day	Tue 28/11/06	Tue 28/11/06
19	▦	Detailed Architectural Design	11 days	Wed 29/11/06	Wed 13/12/06
20	▦	Design Review	1 day	Thu 14/12/06	Thu 14/12/06
21	▦	Detailed User Interface Design	11 days	Fri 15/12/06	Fri 29/12/06
22	▦	Detailed Component Level Design	10 days	Mon 1/1/07	Fri 12/1/07
23	▦	Milestone 4 (Detailed Design Report)	1 day	Mon 15/1/07	Mon 15/1/07
24		**WP 4: Implementation**	**59 days**	**Thu 2/11/06**	**Tue 23/1/07**
25	▦	Implementing Graphical User Interface	10 days	Thu 2/11/06	Wed 15/11/06
26	▦	Implementing Design View	10 days	Thu 16/11/06	Wed 29/11/06
27	▦	Implementing Code View	10 days	Thu 30/11/06	Wed 13/12/06
28	▦	Implementing Browser View	2 days	Thu 14/12/06	Fri 15/12/06
29	▦	Integrating Views	10 days	Mon 18/12/06	Fri 29/12/06
30	▦	Implementing AJAX Events	10 days	Mon 1/1/07	Fri 12/1/07
31	▦	Implementing Database Support	5 days	Mon 15/1/07	Fri 19/1/07
32	▦	Milestone 5 (Prototype Demo)	1 day	Tue 23/1/07	Tue 23/1/07

9. Conclusion

In conclusion, we can say that our project will be one of the featured IDEs to develop AJAX applications. By the help of market analysis the borders of our project scope is mainly created. We recognize the most necessary features that should be stated in an IDE as well as necessary properties to add AJAX functionality to web projects. In our project we will cover all important properties as similar projects have. Beside existing features we will add some features unique to our project to make it more users friendly and functional.

We believe the importance of sharing our ideas with web application programmers and getting feedback from them. In order to achieve this purpose, we have prepared a questionnaire which can be seen at 3.9. The aim of preparing this questionnaire is to gain knowledge about what the web programmers need and their expectations and their knowledge about AJAX.