

CENG 490

STARSOFT

Requirement Analysis

Report

By;

Fatih YILDIRIM

Özer GÜMÜŞ

Özge YAMASAN

Mehmet ALBAYRAK

Ömer ESER

1. Introduction.....	3
1.1. Scope	3
1.2. Goals And Objectives	3
2. The Process.....	4
2.1. Team Organization	4
2.2. Process Model.....	4
2.3. Schedule	5
2.3.1. Gantt Chart.....	6
3. Literature Survey.....	7
3.1. Research About Ajax.....	7
3.1.1. Brief history	8
3.1.2. Why to use Ajax?	8
3.1.3. What is Ajax, in depth?	8
3.1.4. How does it work?	9
3.1.5. Ajax and DHTML	10
3.1.6. The XMLHttpRequest class	10
3.1.7. Building a request, step by step.....	11
3.1.8. Examples	12
3.1.9. How to build an Ajax website?	15
3.1.10. The Ajax Toolkit Framework.....	15
3.1.11. Drawbacks	15
3.1.12. Specifications	16
3.2. Marketing Research.....	16
3.2.1. Microsoft Visual C++ Analysis.....	16
3.2.2. Eclipse IDE Analysis	19
3.2.3. MORFIK WEBOS AppsBuilder Analysis	21
3.2.4. Google Web Toolkit Analysis	23
3.2.5. The Common Features	24
3.3. Meeting With Cevat Şener	25
3.4. Meeting With Emrah Şener	26
4. Technical Analysis	26
4.1. Functional Requirements	26
4.1.1. Editor Module	26
4.1.2. Project	27
4.1.3. Debugger	28
4.1.4. Database	28
4.2. Non-Functional Requirements.....	28
4.2.1. Usability	28
4.2.2. Security	29
4.2.3. Reliability.....	29
4.2.4. Serviceability.....	29
4.3. Software Requirements	29
5. System Modeling.....	30
5.1. DFD LEVEL0.....	30
5.2. DFD LEVEL 1.....	31
5.3. Use-Case Diagram.....	32
6. References.....	33

1. Introduction

1.1. Scope

Ajax is mainly a web development technique for creating interactive web applications. The components that made up Ajax were being used even before the naming took place. Due to the convenience that Ajax brings to developers it has gained popularity quite quickly. In this project we will design and implement a Graphical Development Environment for webpages with Ajax which helps the developers further by putting many options and features together.

The software is to include an editor to write and edit the necessary codes. The text-highlighting method will be used to increase readability. Predefined scripts and actions will be available. Features such as adding, locating, editing, removing scripts and actions will be put to use. Without using any other software, debugging the code when necessary will also be at hand. The user will be able to see the effect of the changes made and run the code. The advanced graphical interface is to be implemented as a way of easing the processes of the user. Hence, the user of the software is to be satisfied with every need he/she has through one complete package.

1.2. Goals And Objectives

During the analysis of the software we have focused on the following goals and objectives. These will be examined again as non-functional requirements in the following sections of the report as they also constitute an important part of the requirements of our project.

Easy to Understand : The software package is bound to include many help options and documentary to help the user understand how to use the software more efficiently with less effort.

Easy to Use : One of the most important reasons for a developer to use such a software is to make his life easier with the features presented through the software which are not

available in a regular editor. Consequently, a great deal of effort is to be made to make the software satisfactory to these needs : Text-Highlighting, pre-defined scripts, etc...

Performance : Since such software should satisfy complex developing as well as simple ones, the performance issue is rather an important one. The aimed-customer database is also making the performance goals more important since it is clearly more obvious and critical to someone who is a developer than a regular software user.

Update Readiness : The technology that the software will depend on in terms of scripts , actions, debugging and so on, makes the update issue an important one. One of the main goals of the project is to make sure not to overlook this concept as with time and with the high speed of the introduction of new features it should be easy to add new modules, remove old ones which will not be necessary anymore.

2. The Process

2.1. Team Organization

Our team is formed by five people. Therefore, we choose a group leader in order to coordinate and interact between group members. However, topics are discussed by all group members and decisions are taken by all group members and all of them approve it. Also, all group members will work on every phase of project and will make both documentation and coding.

Our team makes a meeting every week and also every day we are connecting each other on the internet and have a conversation about our last topic. Besides, all our documentation is collected at our mail group and also we can use it in order to communicate.

2.2. Process Model

We will not use a specific process model. Our process model is changed according to our project phase. Until prototype of our project, we use linear model. Firstly, we decide to our

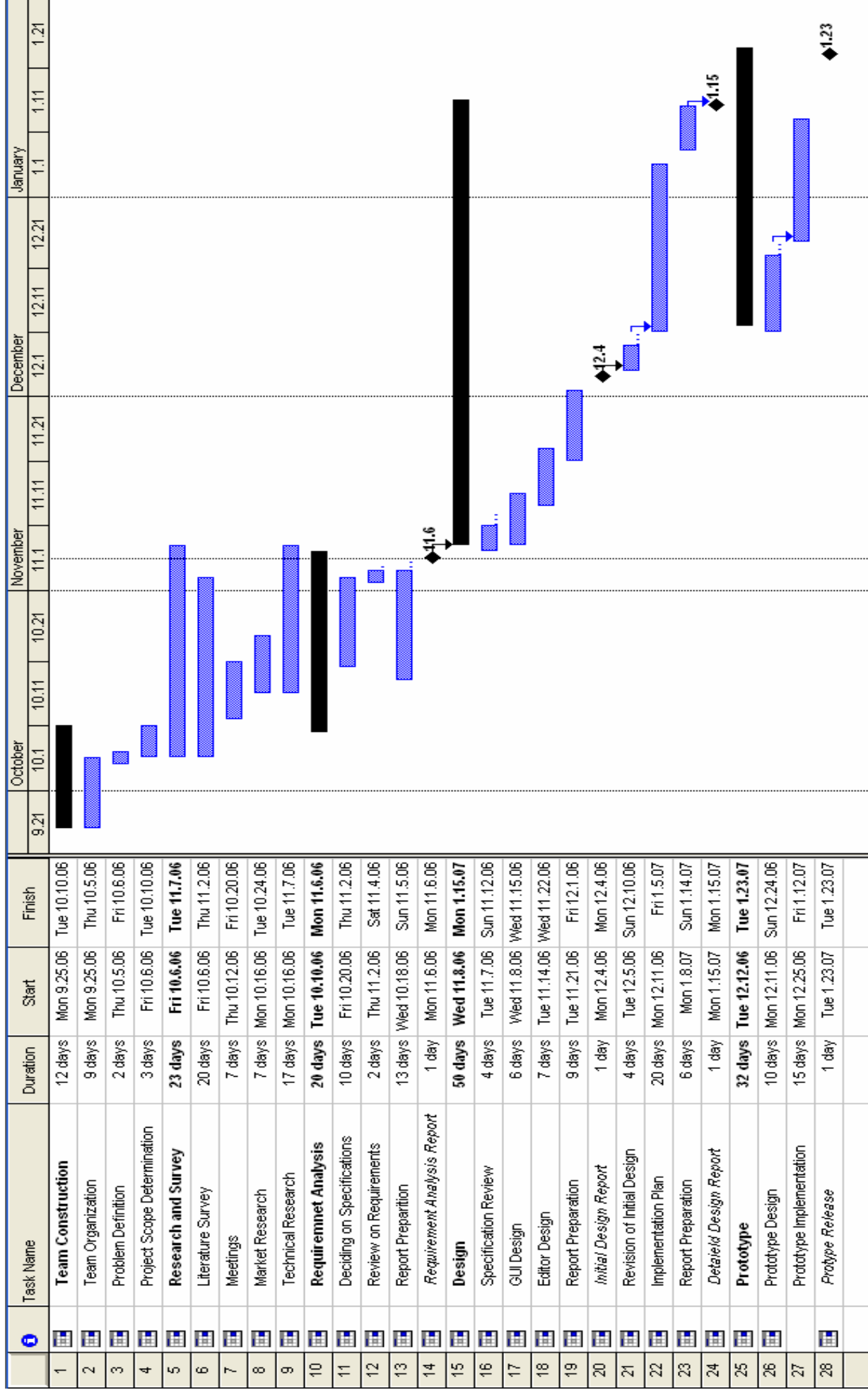
requirements. Then we make design of our project. We make prototype our project which shows basic function of our project. Besides, we will use evolutionary model in the implementation part of our project. This model helps us to show our mistakes and problems with our program.

2.3. Schedule

We have to finish our project at the beginning of June 2007. At the first part of our project we have to make a prototype of the project until the January 23rd and we show it at this day.

In order to be more specific and exact about our schedule we have worked on a Gantt Chart.

2.3.1. Gantt Chart



3. Literature Survey

3.1. Research About Ajax

Ajax (Asynchronous Java script And Xml), as noted from the beginning and will be clearly explained in the below part, is a web development environment for client-side dynamic web pages. Ajax is actually a set of tools that were previously existed. For further information about Ajax, let's first define some topics, and then according to these topics, give some detail.

- 3.1.1. Brief history
- 3.1.2. Why to use Ajax?
- 3.1.3. What is Ajax, in depth?
- 3.1.4. How does it work?
- 3.1.5. Ajax and DHTML
- 3.1.6. The XMLHttpRequest class
- 3.1.7. Building a request, step by step
 - 3.1.7.1. Create an instance
 - 3.1.7.2. Wait
 - 3.1.7.3. Send the request
- 3.1.8. Examples
 - 3.1.8.1. Get a text
 - 3.1.8.2. Get from xml
 - 3.1.8.3. Write to body
 - 3.1.8.4. Post a text
- 3.1.9. How to build an Ajax website?
- 3.1.10. The Ajax Toolkit Framework
- 3.1.11. Drawbacks
- 3.1.12. Specifications

3.1.1. Brief history

Ajax is only a name given to a set of tools that previously existed. The main part is XMLHttpRequest, a class usable in Java Script that was implemented into Internet Explorer since the 4.0 version. The same concept was named XMLHttpRequest some times, before the Ajax name becomes commonly used. The use of XMLHttpRequest in 2005 by Google, in Gmail and Google Maps has contributed to the success of this format. But this is the name Ajax itself that made the technology so popular.

3.1.2. Why to use Ajax?

Main purpose of using Ajax is to build a fast, dynamic website, but also to save resources. For improving sharing of resources, it is better to use the power of all the client computers rather than just a unique server and network. Ajax allows performing processing on client computer (in JavaScript) with data taken from the server. The processing of web page formerly was only server-side, using web services or Php scripts, before the whole page was sent within the network. However Ajax can selectively modify a part of a page displayed by the browser, and update it without the need to reload the whole document with all images, menus, etc. For example, fields of forms, choices of user, may be processed and the result displayed immediately into the same page.

3.1.3. What is Ajax, in depth?

Ajax is a set of technologies, supported by a web browser, including these elements:

- Html and CSS for presenting.
- JavaScript (ECMAScript) for local processing, and DOM (Document Object Model) to access data inside the page or to access elements of Xml file read on the server (with the getElementByTagName method e.g.)
- The XMLHttpRequest class read or sends data on the server asynchronously.

And there are some more used optionally:

- The DomParser class may be used

- PHP or another scripting language may be used on the server.
- XML and XSLT to process the data if returned in Xml form.
- SOAP may be used to dialog with the server.

The "Asynchronous" word, means that the response of the server while be processed when available, without to wait and to freeze the display of the page.

3.1.4. How does it work?

Ajax uses a programming model with display and events. These events are user actions; they call functions associated to elements of the web page. Interactivity is achieved with forms and buttons. DOM allows to link elements of the page with actions and also to extract data from Xml files provided by the server.

To get data on the server, XMLHttpRequest provides two methods:

- Open: create a connection.
- Send: send a request to the server.

Data furnished by the server will be found in the attributes of the XMLHttpRequest object:

- ResponseXml for a Xml file or
- ResponseText for a simple text.

Take note that a new XMLHttpRequest object has to be created for each new file to load.

We have to wait for the data to be available to process it, and in this purpose, the state of availability of data is given by the readyState attribute of XMLHttpRequest.

States of readyState follow (only the last one is really and mostly useful):

- 0: not initialized
- 1: connection established
- 2: request received
- 3: answer in process
- 4: finished

3.1.5. Ajax and DHTML

Dhtml has same purpose and is also, as Ajax, a set of standards:

- Html,
- Css,
- JavaScript.

Dhtml allows changing the display of the page from user commands or from text typed by the user. Ajax allows also sending requests asynchronously and loading data from the server.

3.1.6. The XMLHttpRequest class

The XMLHttpRequest class allows interacting with the servers, thanks to its methods and attributes.

Attributes

ReadyState: the code successively changes value from 0 to 4 that means for "ready".

Status : 200 is ok , 404 if the page is not found.

ResponseText: holds loaded data as a string of characters.

ResponseXml: holds an Xml loaded file, DOM's method allows to extract data.

Onreadystatechange: onreadystatechange

Methods

Open (mode, url, boolean)

Mode: type of request, GET or POST

Url: the location of the file, with a path.

Boolean: true (asynchronous) / false (synchronous).

Optionally, a login and a password may be added to arguments.

Send ("string")

Null for a GET command.

3.1.7. Building a request, step by step

Building a request occur in there steps.

3.1.7.1. Create an instance

This is just a classical instance of class, but two options must be tried, for browser compatibility.

```
if (window.XMLHttpRequest) // Object of the current windows
{
    request = new XMLHttpRequest(); // Firefox, Safari...
}
else
if (window.ActiveXObject) // ActiveX version
{
    request = new ActiveXObject("Microsoft.XMLHTTP"); // Internet Explorer
}
```

A sample code for creating an instance of XMLHttpRequest, the "request" object

3.1.7.2. Wait

The response and further processing are included in a function and the return of the function will be assigned to the onreadystatechange attribute of the object previously created.

```
request.onreadystatechange = function () { // instructions to process the response};

if (request.readyState == 4)
{
    // received, OK
} else
{
    // wait...
}
```

3.1.7.3. Send the request

Two methods of XMLHttpRequest are used:

open: command GET or POST, URL of the document, true for asynchronous.

send: with POST only, the data to send to the server.

The request below read a document on the server.

```
http_request.open('GET', 'http://www.ceng.metu.edu.tr/seniorproject.xml', true);  
http_request.send(null);
```

3.1.8. Examples

3.1.8.1. Get a text

This is an example to get a text and syntax of form using Ajax

```
<html>  
<head>  
<script>  
function submitForm()  
{  
    var req = null;  
    if(window.XMLHttpRequest) req = new XMLHttpRequest();  
    else  
        if (window.ActiveXObject) req = new ActiveXObject(Microsoft.XMLHTTP);  
  
    req.onreadystatechange = function()  
    {  
        if(req.readyState == 4)  
        {  
            if(req.status == 200)  
                document.ajax.dyn="Received:" + req.responseText;  
            else
```

```

        document.ajax.dyn="Error code " + req.status;
    };
    req.open( GET", "data.xml", true);
    req.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    req.send(null);
}
</script>
</head>

<body>
<FORM method="POST" name="ajax" action="">
    <INPUT type="BUTTON" value="Submit" ONCLICK="submitForm()">
    <INPUT type="text" name="dyn" value="">
</FORM>
</body>
</html>

```

If to give some comments on the code above:

```
new ActiveXObject(Microsoft.XMLHTTP)
```

This constructor is for Internet Explorer.

```
new XMLHttpRequest()
```

This constructor is for any other browser including Firefox.

```
http.onreadystatechange
```

An anonymous function is assigned to the event indicator.

```
http.readyState == 4
```

The 4 state means for the response is ready and sent by the server.

```
http.status == 200
```

This status means ok, otherwise some error code is returned, 404 for example.

```
http.open("POST", "data.xml", true);
```

POST or GET

URL of the script to execute.

true for asynchronous (false for synchronous).

```
http.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
```

This is for POST only.

```
http.send(document.getElementById("TYPEDTEXT").value);
```

Send data to the server. Data comes from the "TYPEDTEXT" variable filled through the form by the user.

3.1.8.2. Get from xml

To get data from an xml file we have just to replace this line:

```
document.ajax.dyn=""Received:" + req.responseText;
```

by this code:

```
var doc = req.responseXML; // assign the Xml file to a var
```

```
var element = doc.getElementsByTagName('root').item(0); // read the first element with  
a dom's method
```

```
document.ajax.dyn.value= element.firstChild.data; // assign the content of the element to  
the form
```

3.1.8.3. Write to body

In the code segment given above, the text read is put in the body of the page, and not into a textfield. The code below replaces the textfield form object and the second part replaces the assignment into the JavaScript function.

```
<div id="zone">
```

```
... some text to replace...
```

```
</div>
```

```
document.getElementById("zone").innerHTML = "Received:" + req.responseText;
```

3.1.8.4. Post a text

In this code example, a text is sent to the server and is written into a file. The call to the "open" method changes, the argument is POST, and the "send" method also has now a value for argument.

```
req.open("POST", "ajax-post.xml", true);  
req.send(document.getElementById("dyn").value);
```

3.1.9. How to build an Ajax website?

There is a need for some wrapper. A short list of frameworks is provided below. Your JavaScript program, integrated into a web page, sends request to the server to load files for rebuilding of pages. The received documents are processed with Dom's methods or Xml parsers and the data are used to update the pages.

3.1.10. The Ajax Toolkit Framework

It is an Eclipse add-on that provides tools for building IDE for Ajax runtimes, and testing Ajax applications. This toolkit uses XULRunner, the Xul runtime, for embedding Gecko, the Html displayer in Eclipse.

3.1.11. Drawbacks

- If JavaScript is not activated, Ajax can't works. The user must be asked to set JavaScript from within options of the browser, with the "noscript" tag.
- Since data to display are loaded dynamically, they are not part of the page, and the keywords inside are not used by search engines.
- The asynchronous mode may change the page with delays (when the processing on the server takes some times), this may be disturbing.
- The back button may be deactivated (this is not the case in examples provided here).

3.1.12. Specifications

Ajax is based on these specifications:

- XML 1, HTML 4.0, DOM 2, from W3C
- ECMAScript 1.5 (standard for JavaScript) from ECMA.
- W3C draft specification for XMLHttpRequest.

3.2. Marketing Research

3.2.1. Microsoft Visual C++ Analysis

Microsoft Visual C++ (MSVC) is an Integrated Development Environment (IDE) product of Microsoft for the C, C++, and C++/CLI programming languages. It has tools for developing and debugging C++ code, especially that written for the Microsoft Windows API, the DirectX API, and the Microsoft .NET Framework.

According to Microsoft: "Microsoft Visual C++ is the most productive C++ tool for creating the highest-performance applications for Windows and the Web. Nearly all world-class software, ranging from the leading Web browsers to mission-critical corporate applications, is built using the Microsoft Visual C++ development system. Visual C++ 6.0 brings a new level of productivity to C++, without compromising flexibility, performance, or control."

Significant Features

Microsoft Visual C++ includes a large and elaborate collection of software development tools, all used through a windowed interface. The Microsoft Visual C++ includes the tools listed below.

Compiler

MSVC compiler can process both C source code and C++ source code. Furthermore, the version 4.0 compiles Fortran code as well.

Debugger

MSVC has an advanced debugging functionality. For example, it allows for remote debugging using a separate computer and allows debugging by stepping through code a line at a time. The "edit and continue" functionality allows changing the source code and rebuilding the program during program debugging, without restarting the debugged program.

Editors

The MSVC environment includes an integrated text editor to manage, edit, and print source files. With the text editor, you can:

- o Automatically fill in code syntax by choosing from a generated list of class members, parameters or values. This feature is called IntelliSense.
- o Automate the text editor with macros.
- o Set and customize syntax coloring for source files. It supports C/C++, SQL, and HTML source files.
- o Emulate two popular text editors: BRIEF and Epsilon.
- o Perform advanced find and replace operations in a single file or multiple files.
- o Use virtual spaces for advanced cursor positioning.
- o Navigate through sections of code by matching group delimiters, matching conditional statements, or using the Go To dialog box.
- o Use Bookmarks to mark frequently accessed lines in your source file.
- o Customize the text editor with save preferences, the selection margin, tabs, and indents.
- o Modify the font style, size, and color.
- o Select lines, multiple lines, or columns.
- o Use drag-and-drop editing within one editor window, between editor windows, and between the Text editor and the debugger.
- o Test SQL scripts, triggers, and stored procedures, and insert triggers or stored procedures into a database.
- o Manage the source window.

Projects

MSVC programs are organized into a Visual C++-based projects that contain all the files you use when developing, compiling, building, and debugging your applications. You view your project in a workspace window which has four different tabs for displaying information: ClassView, which shows the classes in your project, FileView, which shows the files in your project, and ResourceView, which shows the resources in your project.. Visual C++ projects are part of a project workspace that allows you to create multiple projects in more than one language in a shared development environment. Files created in one Visual C++ project can be shared with other Visual C++ projects and, where appropriate, with projects created for other programs.

Wizards

Wizards, like the MFC AppWizard, the MFC ActiveX ControlWizard, the ISAPI Extension Wizard, and the ATL COM AppWizard help you generate the basic source files for a variety of different types of Windows programs.

Database Support

MSVC has three options to access databases:

- o ActiveX Data Objects (ADO) is the strategic application programming interface (API) to data and information. ADO provide's consistent, high-performance access to data and supports a variety of development needs, including the creation of front-end database clients and middle-tier business objects that use applications, tools, languages, or Internet browsers. ADO is designed to be the one data interface needed for single and multitier client/server and Web-based data-driven solution development.

- o OLE DB is the Microsoft strategic system-level programming interface to data across the organization. OLE DB is an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data. Whereas ODBC was created to access relational databases, OLE DB is designed for relational and nonrelational information sources, including mainframe ISAM/VSAM and hierarchical databases; e-mail and file system stores; text, graphical, and geographical data; custom business objects; and more.

- o Open Database Connectivity (ODBC) interface is an industry standard and a component of Microsoft® Windows® Open Services Architecture (WOSA). The ODBC

interface makes it possible for applications to access data from a variety of database management systems (DBMSs).

3.2.2. Eclipse IDE Analysis

Eclipse is actually an open source community rather than being only an IDE. Yet our interest in the community consists of the IDE part. The existence of many Eclipse based IDE's that are commercially used and also are very prestigious makes it important to go through the Eclipse IDE. Eclipse, even though being mainly used for development in Java, supports many languages. There are various features included in the IDE and in our analysis the following main features are observed.

Significant Features

Compiler

The Eclipse IDE can compile and show the result of the compilation without any keyboard input from the user. For example in the JAVA case javac.exe is found prior to first use and used as needed. Problems detected by the compiler are classified as either warnings or errors. The existence of a warning does not affect the execution of the program; the code executes as if it were written correctly. Compile-time errors are always reported as errors by the Java compiler. For another case if CDT plugin is installed for example, C++ code can be compiled in a similar way and even a makefile can be produced automatically.

Debugger

The Java Development Toolkit (JDT) includes a debugger that enables you to detect and diagnose errors in the programs written.

The debugger allows the control of the execution of your program. Breakpoints, suspending launched programs, stepping through the code, and examining the contents of variables are included.

The debugger has a client/server design so one can debug programs running remotely on other systems in the network as well as programs running locally the your workstation.

Editor

The following features are available in terms of the editor :

- Syntax highlighting
- Content/code assist
- Code formatting
- Import assistance
- Quick fix
- Integrated debugging features

Templates

In the Eclipse Help section templates are defined as a structured description of coding patterns that reappear in source code. For the Java case as it is our main interest area The Java editor supports the use of templates to fill in commonly used source patterns. Templates are inserted using content assist.

Projects

Projects and to be more specific a Java project contains source code and related files for building a Java program. It has an associated Java builder that can compile Java source files as they are modified.

A Java project also maintains a model of its contents. This model includes information about the type hierarchy, references and declarations of Java elements. This information is constantly updated as the user changes the Java source code. The updating of the internal Java project model is independent of the Java builder; in particular, when performing code modifications, if auto-build is turned off, the model will still reflect the present project contents.

Perspectives

Each Workbench window contains one or more perspectives. A perspective defines the initial set and layout of views in the Workbench window. Within the window, each perspective shares the same set of editors. Each perspective provide's a set of functionality

aimed at accomplishing a specific type of task or works with specific types of resources. For example, the Java perspective combines views that you would commonly use while editing Java source files, while the Debug perspective contains the views that you would use while debugging Java programs. As you work in the Workbench, you will probably switch perspectives frequently.

Plug-in Support

One of the most interesting parts of the Eclipse IDE is the availability of the number of plug-ins. Actually Eclipse encourages the build of those plug-ins and in turn these do provide great flexibility to the program.

In Eclipse.org the The Eclipse Plug-in Model is described as follows : “A plug-in in Eclipse is a component that provides a certain type of service within the context of the Eclipse workbench. The component here means an object that may be configured into a system at system deployment time. The Eclipse runtime provides an infrastructure to support the activation and operation of a set of plug-ins working together to provide a seamless environment for development activities.”

It is obvious that the plug-in environment is a very strong suit for Eclipse IDE and it is not to be overlooked.

Database Access

In Eclipse IDE, database access can be achieved easily through the plug-ins that are easy to acquire even through a quick search on the net. The variety of plugins mentioned above constitute a rich database access options to variety of database models.

3.2.3. MORFIK WEBOS AppsBuilder Analysis

The WebOS AppsBuilder is a specialized IDE for building web-based applications. It brings together a visual designer, a choice of high level languages - C#, Java, Basic and Pascal, database and web servers, and tightly integrates them into a single entity that is both powerful and easy to use. The WebOS AppsBuilder permits to remain focused on client

needs. WebOS AppsBuilder brings extreme simplicity to the design of sophisticated AJAX applications without compromising their functionality.

Significant Features

Visual Development

The Visual Designer enables to create database tables, link these tables, maintain their schemas, and create complex forms and dynamic data-driven PDF reports in a familiar visual environment. There is no need to learn complex SQL syntax, as queries involving multiple tables, sorting, parameters, and programming functions can be built using the intuitive and powerful visual query builder.

High Level Languages

Using WebOS AppsBuilder, you can implement the business logic of your application in a high-level object-oriented language of your choice. Basic, C#, Java and Object Pascal are currently supported. The Morfik Compiler uses patent pending JavaScript Synthesis Technology (JST), to generate the JavaScript AJAX Engine for the browser and the server-side application. If you wish, you can even mix and match languages to meet your needs!

Morfik Compiler

The Morfik Compiler and the integrated JavaScript Synthesis Technology (JST) at WebOS AppsBuilder allow to develop and debug the business logic of your application. The WebOS AppsBuilder compiles the project's code, resources and objects into an AJAX enabled Engine that can run both online and offline. It generates pure HTML and JavaScript for the browser and machine code for the server.

Debugging Views

The Morfik WebOS AppsBuilder IDE offers an extensive range of features and views to help in the application debugging process. Amongst these user finds the Profiler View, Breakpoints View, Watches View and Call Stack View. These views appear at the the main window and popup when the respective tab is clicked.

Rational Database

The WebOS AppsBuilder gives all the power of a sophisticated relational database. The database multi-generational architecture engine provide's efficient performance even with

large numbers of concurrent users. It is designed for business-critical distributed database environments that require both power and flexibility, but requires minimal maintenance and administration.

Web Server

The WebOS AppsBuilder brings user the power of a web server but without the stresses of learning endless acronyms, protocols, settings and scripting languages that are associated with using a web server. All the relevant parameters for the web server are controlled within the design environment - the rest are taken care for you - quietly and automatically. There is no command-line programming, unless user really miss it.

3.2.4. Google Web Toolkit Analysis

Google Web Toolkit (GWT) is a Java software development framework that makes writing AJAX applications like Google Maps and Gmail. With GWT, you can develop and debug AJAX applications in the Java language using the Java development tools of your choice. When you deploy your application to production, the GWT compiler translates your Java application to browser-compliant JavaScript and HTML.

GWT has four major components :

Java-to-JavaScript compiler translates the Java programming language to the JavaScript programming language

"Hosted" web browser lets you run and execute your GWT applications in hosted mode

JRE emulation library includes most widely used classes in the Java standard class library and java.lang and java.util package classes.

Web UI class library is a set of custom interfaces and classes that let your create web browser "widgets," like buttons, text boxes, images, and text.

Significant Features

Remote Procedure Calls: Remote Procedure Calls (RPC) makes it easy for the client and server to pass Java objects back and forth over HTTP.

Building User Interfaces: With using the Document Object Model interface which contains methods for manipulating for elements and events,it is possible to manipulate the browser directly.

Debugging: In GWT, your applications run in two modes: Hosted and Web. You can edit, run, and debug applications from an IDE at anytime in hosted mode. Applications run as JavaScript or HTML in web mode. End users only see the web mode version of your application

Compiler: The GWT compiler translates Java source code into Java Script.

Project: The GWT projects root package contains module XML files and overlaid onto Java packages. Most of configuration can be inferred from classpath and your module definition. Module XML files contains Client-side source files and subpackages and Server-side code and subpackages. So your application interact with server or sent across a network to a user, where it runs as JavaScript inside his or her web browser. GWT applications automatically support IE, Firefox, Mozilla, Safari, and Opera with no browser detection or special-casing within your code in most cases

UI components: UI components called Widgets. They includes including hierachical trees, tab bars, menu bars, and modal dialog boxes. The tree contains a hierarchy of TreeItems that the user can open, close, and select. Tab bars commonly used a part of tab panel which represents a tabbed set of pages, each of which contains another widget. A menu bar can contain any number of menu items, each of which can either fire a Command or open a cascaded menu bar.

3.2.5. The Common Features

In our market research we have tried to analyse some very powerful IDE's to come up with the uses of our project topic in real life. The real goal here is to identify the features that serve the user most and that are the most satisfactory so that in our project we can include those.

As our research is complete we have managed to see the most neccessary and fundamental features and some of them can be summarized as follows :

- Compiler Debugger efficiency is all common in these programs. Actually the user is made to be unaware of the background processes. This is quite relevant to user satisfaction.

- Editors are given extra consideration and effort in the IDE's analyzed. One of the main goals of the user of these programmes is to achieve his/her goals in an easier and more efficient way and the editor serves in many ways to come up with that. Text Highlighting is a must. Auto completing the code with available class structures and predefined code is a big plus. Going through the written code in terms of set conditions like loops, if statements etc. provide a great contribution. Moreover there are features that make the transfer of code from one editor window to another one and this seems to be of great importance. Drag and Drop options are available as well. Customization of the editor is another feature that seems to be given importance including features that let the user identify the editor view in a more personal way the way he/she wishes. Realization of the code structures written in the editor and listing them in a separate place brings great benefits to the user.

- In a big project all the files and the source code may become difficult to maintain manually. In our analysis we have realized that this burden is overcome with the Project features that are presented. This is a very common property and it serves the user in a way that the user can access the files and do the organization in a much easier way.

- Database today is a fundamental part of programming. They are quite often required and the availability of different databases makes it important to be able to access many different database management systems easily. The IDE's analyzed tend to provide these features. In Eclipse for example the plug-in availability for such use is huge, or in Visual C++ the built-in availability of different access methods are provided.

3.3. Meeting With Cevat Şener

In our meeting with Cevat Şener we were at the stage of determining the main programming language to be used for the programming part of our project. The main options were .Net and Java Technologies. Cevat Şener has been of great benefit for our project group regarding this.

During our meeting, Cevat Şener has informed us about the main issues of the choice to be made. In the end the conclusion was the use of Java Technology as the most optimum choice. The main reason behind the choice was that in terms of availability of the complicated

libraries that will be examined and put to use in the project, Java has a richer background. The libraries concerned are also more easily examined and modified according to needs in Java.

The choice of the possible use of NetBeans for Graphical User Interface has also been examined after the input we gained from the meeting.

3.4. Meeting With Emrah Şener

Emrah Şener works as a Senior Analyst in System Analysis Department of Garanti Technology. Since he works and lives in Istanbul first a meeting time and place has been determined and then the meeting was held in Istanbul.

During the meeting many analysis questions have arised and the following conclusions have been reached.

There was the question of the platform our software is supposed the work on. The initial question was, should the software be web-based(i.e. works on an internet browser) or client-based(i.e. works as a exe file). Emrah Sener has strongly advised to design the software to be client based.

During the analysis performed one of the most challenging subjects was the use of web services as the information found was quite abstract. We have taken important ideas about the use of the web services in terms of database communications. The use of XML has been clarified.

The design of the GUI of our project was another topic of our meeting. Some important software relating the subject has been advised(i.e. GUI Design Tool).

4. Technical Analysis

4.1. Functional Requirements

4.1.1. Editor Module

Editor module will serve as the main development enviroment so it can be considered as the heart of our software. It will basicly consist of two elements: **Code Editor** and **Desing Editor**.

Code editor will be an integrated text editor. Code editor will be used to just see the codes generated by the software or developers can modify the code directly from here. It will also show the line numbers on the left side. For comfortable coding, code editor will have some basic features that most editors have like undo, redo, copy, cut, paste, find, find and replace, balancing braces, going to specified line number, and text-highlighting. Since our software is for web developers code editor must support popular web development languages/tools like HTML, Javascript, PHP, etc.

Design editor will be an integrated graphical development environment. Whatever one does in design environment will generate an equivalent code that can be seen from the code editor. For example inserting an image in the design editor -through menus or toolbars- will generate a code with `` tag. Equivalently if one insert an image in the code editor -through menus, toolbars or directly writing code- this will also cause some changes in the design editor.

Design editor will have a helper module called “properties”. All the properties like font, color, border color, etc. of elements; their events; if any database relations of these elements will be controlled from this module. At this point, AJAX worths mentioning. If one wants to add AJAX actions to his web component some necessary steps can be automated but also some coding will be needed. For example assume that a developer wants to have suggestions to names that are written to a text box from a database table. Every time a user types a letter, suggestion should be refreshed accordingly. To do this a Javascript “onKeyUp” event is necessary. User can choose this event from a list of events and also connecting to database and selecting the required table can be automated but comparing the string written to the text box and comparing it with those taken from the database table is the business of the developer. Fortunately this does not decrease the quality of the program because coding issue is mostly related to PHP or ASP.

4.1.2. Project

Project module will provide a comfortable development environment for the developers. By means of this module more than one file can be opened at the same time in a tabbed fashion. Files, related databases and other components of a project can be viewed in a tree-view in a separate window.

4.1.3. Debugger

Debugger module will help the developer to find his errors. Web developers usually suffer from syntactic errors because browsers does not generate errors for these types of errors and they are difficult to find. For example forgetting to close a tag may not cause an abnormal situation where the mistake is made but arise in a different location. Our debugger module will be warning the developer due to these types of errors.

Another aspect of our debugger will function for server side programming languages like PHP or ASP. In fact web browsers give error messages for them but usually they are far from telling the cause of error. For this, our debugger will run line by line and show the values of variables in the code. By this way, the cause of error can be understand better.

4.1.4. Database

Databases are indispensable parts of interactive web programming. Our database module will handle all database related issues. It will be able to connect to remote servers to access databases. These database tables will be able to be viewed and modified inside the software. Due to the purpose of our software the uses of databases gain a special importance. And the existence of the different computer connectivity makes web-service uses a must. Therefore web services will be used for these purposes. Extensive use of XML (even with its drawbacks about performance) will be implemented and the web service will act as a layer that standardizes the input output processes that are generated or generated into the software. With the different XML tables written for different kinds of databases compliance issues will be overridden.

4.2. Non-Functional Requirements

4.2.1. Usability

Usability is one of the major issues to attract costumer and to stand in the market. According to researches; if usability is not good enough for the user, project cannot survive in

the market even if it provides better solutions to the problems. Therefore; during the development of GUI, usability is considered as main issue.

4.2.2. Security

At our program, there is security system and every user has a username and a password in order to access to database. In this way user can access to database and take file which he/she wants. The user can also make changes on this file and send it to back.

4.2.3. Reliability

We will do our program so stable such that there will be not minor problem. Moreover, a lot of tests will be done after implementation of program in order to decrease the errors on program.

4.2.4. Serviceability

We should provide a tutorial for the users, how to use the developer, the possible application areas, security constraints and standard issues. In addition, we should provide a set up tutorial in order to explain how the user set up program.

4.3. Software Requirements

User Requirements

Windows Operating System

J2SE Runtime Environment

A web browser with AJAX support

Developing Requirements

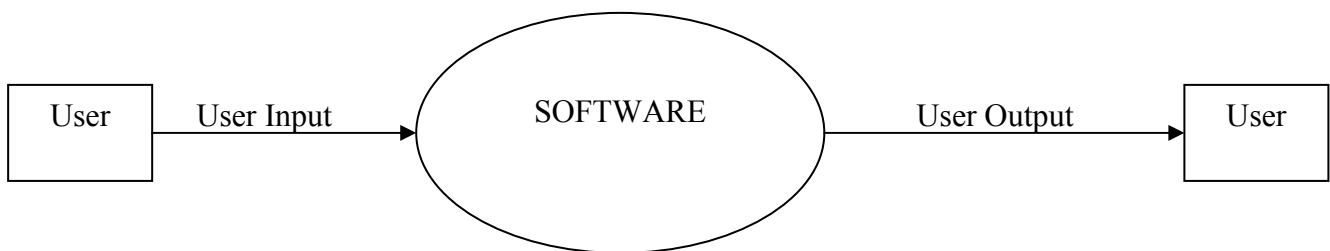
Windows Operating System

J2SE Runtime Environment

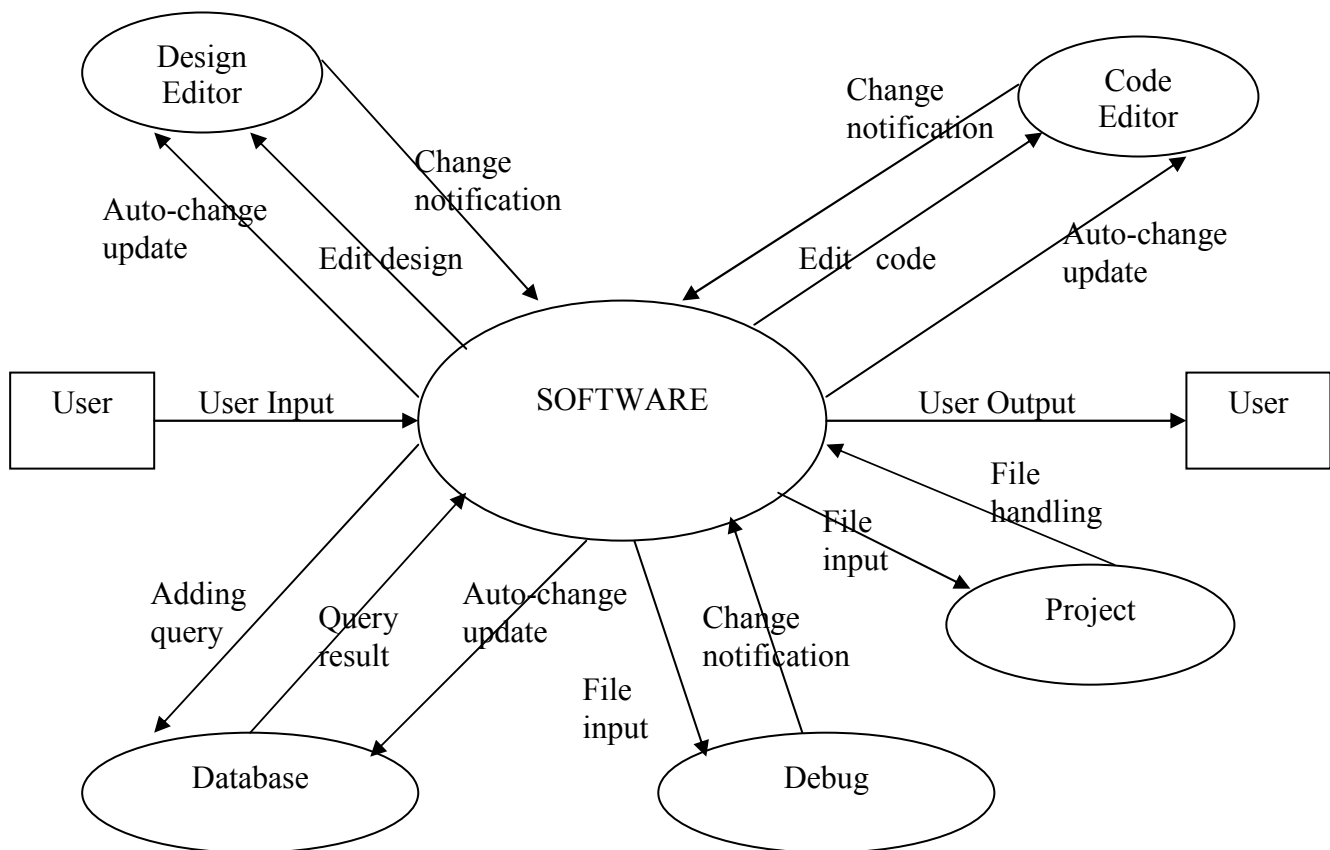
Eclipse with Netbeans

5. System Modeling

5.1. DFD LEVEL0



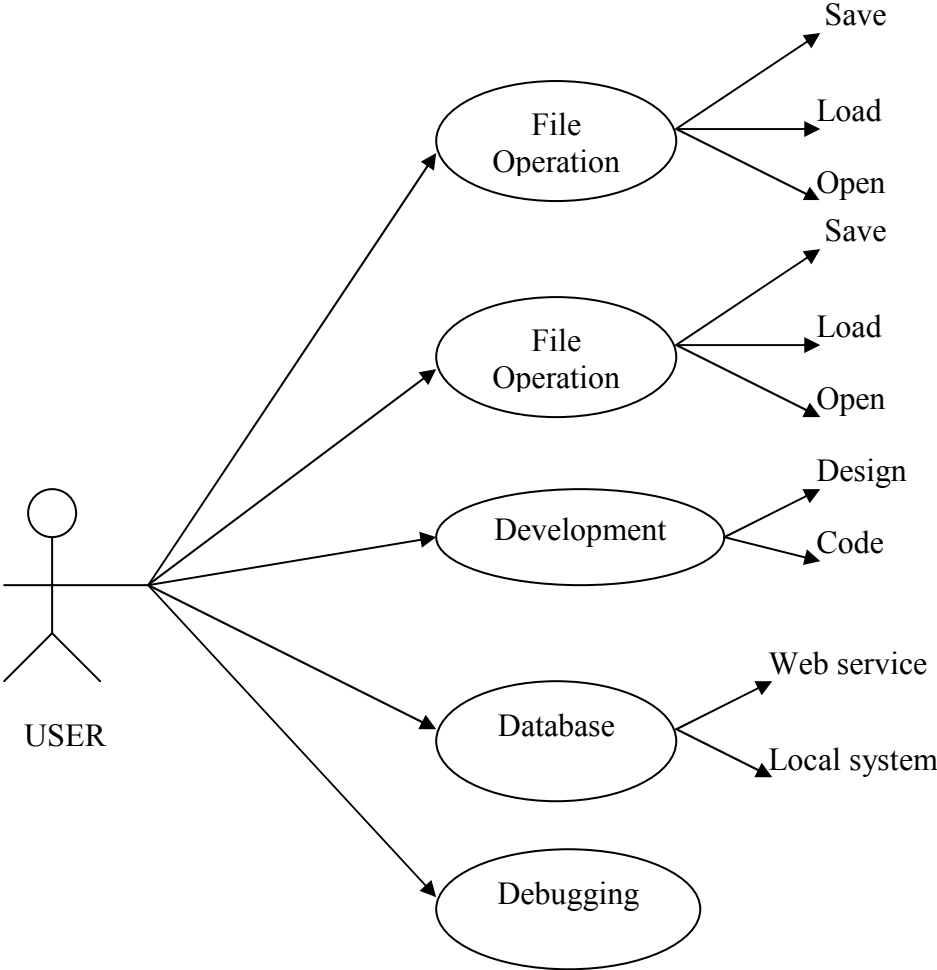
5.2. DFD LEVEL 1



Auto-change Update: A means of updates of change, and works when there is a change. For example there can be a change in code editor that is, developer may possibly add some codes to code editor, or in design editor that is, developer may possibly add some graphical elements to design environment. When there is such a situation, software automatically reflects new patterns to other parts and thus update.

Change Notification: This is a notification to software that there is a change in some parts and there must be some changes in other parts of studio, that is, reflections of previous changes must be applied to others. For example, if there is a change in or adding a graphical element to design editor, this must be reflected to code editor immediately.

5.3. Use-Case Diagram



6. References

[1] Wikipedia, the free encyclopedia : <http://en.wikipedia.org/wiki/Ajax>

[2] Sun Developer Network : <http://java.sun.com/>

[3] Software Engineering: A Practitioner's Approach : Roger S Pressman, Roger Pressman

[4] Eclipse Home : <http://www.eclipse.org/>

[5] NetBeans Home : <http://www.netbeans.org/>

[6] Ajax Matters : <http://www.ajaxmatters.com/>