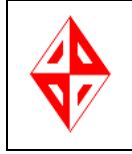


MIDDLE EAST TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT



CEYLAN®

AJAX SOFTWARE DEVELOPMENT STUDIO

ANALYSIS REPORT



Group Members:

Saliha ALTUNSOY

Candan CEYLAN

Canan ESKİ

Yavuz GÖKIRMAK

Duygu GÖRGÜN

ABSTRACT

The Internet has become an indispensable part of our life in conjunction with the rapid growth in technology. The growth in technology made computers reachable for people from every section of the population and as a result, information sharing became a more important concept. The need for information sharing was satisfied by the web sites and pages on the internet. People of all age groups and from all sectors use the web for different purposes; information gathering, communication, financial operations, etc. These users were too eager to reach the information and the late arrival of the results for their requests was making them unsatisfied. However, on the server side things were getting more and more complex. It wasn't easy for them to serve information to a large number of clients. This extensive usage and impatient users made the speed a critical issue for web applications and the web sites with a smaller response time became more popular. To make web applications respond faster, new approaches were developed. AJAX came to solve the speed problem and become the most popular approach. Most of the developers started to construct web applications using AJAX. And our project takes the root from here; to assist the software developers with a robust development environment.

1.0 INTRODUCTION	5
1.1Project Title.....	5
1.2 Problem Definition.....	5
1.3 Statement of scope.....	6
1.4 Application Areas of Our Software.....	7
2. MARKET RESEARCH.....	7
2.1 Reasons Why Developers Chose AJAX.....	7
2.2 Tools on the Market for AJAX Development.....	9
2.2.1 EchoStudio2 :.....	9
2.2.2 Eclipse ATF :.....	9
2.2.3 MyEclipse AJAX Support :.....	10
2.2.4 Google Web Toolkit :.....	12
2.2.5 TIBCO General Interface :	13
2.3 Expectations of Developers from an AJAX Development Studio.....	14
3. PROJECT REQUIREMENTS.....	15
3.1 Functional Requirements.....	15
3.1.1 Widgets such as tree view, toolbars and tabs.....	15
3.1.2 Server connectivity.....	15
3.1.3 Query Execution.....	15
3.1.4 Editing, and Debugging.....	16
3.1.5 Work Space.....	16
3.1.8 Menu Components.....	17
3.1.8.2 Edit.....	18
3.1.9 Integrated browser.....	18
3.1.10 Code and design view.....	18
3.2 Non-Functional Requirements.....	19
3.2.1 User-friendliness.....	19
3.2.2 Modularity.....	19
3.2.3 Platform Independency.....	19
3.2.4 Consistency.....	19
3.3 System Requirements.....	20
3.3.1 Software Requirements:	20
3.3.2 Hardware Requirements:	20
4. PROJECT SCHEDULE.....	21
5. TEAM ORGANIZATION & PROCESS MODEL.....	21
6. RISK MONITORING AND MANAGEMENT PLAN.....	22
7. USAGE SCENARIO.....	25
Use-cases.....	25
8. MODELLING.....	28
8.1 Functional Modeling.....	28
8.1.1 Data Flow Diagrams (DFD).....	28

8.1.2 Process Specifications (PSPEC).....	30
8.1.3 Data Dictionary.....	32
8.2 Behavioral Modeling.....	37
8.2.1 State Transition Diagram (STD).....	37
APPENDICES.....	38
Appendix A.....	38
Query.....	38
Appendix B.....	41
Gantt Chart.....	41

1.0 INTRODUCTION

This document aims to provide information about the solution process of the problem. Firstly, problem definition and the scope of the project can be found in this document. Then, results of our market research is included and discussed. Moreover, specifications about requirements and the features to be developed according to the analysis results are told in details. Data Flow Diagrams are used to show interactions between modules. A risk management plan is also included. And finally, a Gantt Chart which shows our planned progress is included.

1.1 Project Title

Our development studio is called CEYLAN.

1.2 Problem Definition

These days, web applications are being used in a wide range of areas; from banking applications, search engines, e-government applications, online-library applications, etc. to non-institutional web pages. In most of these areas, response time is a very important issue. To illustrate, for a search engine the application must retrieve the result and display it in a short time period. Or in a banking application, the purpose is to allow the customer to transact instead of going to the bank branch and waiting for other people to complete their operations. If the user waits too long for response in front of the screen, using the web application will be meaningless. AJAX offers a different approach to make these web applications respond faster.

In the traditional approach application model works like this: most user actions in the interface trigger an HTTP request sent to the web server, the server does the processing – retrieving data, crunching numbers, etc. – and then returns an HTML page to the client. Even if there is a very little change in the interface, the whole page is reloaded again. And the user waits until the server responds. Instead, AJAX offers an approach which eliminates the start-stop-start-stop nature of interaction on the web. That is; instead of loading the entire page again and again, only the

changed parts are loaded and the user can interact with the application asynchronously, independent of communication with the server. Actually, **AJAX** is shorthand for **A**synchronous **J**avaScript and **X**ML.

Due to the advantages offered by Ajax, most of the developers use this approach. During development stage, the development environment is a very important issue and some utilities are expected from IDEs to make the job of developers easier. The developers of Ajax need a user-friendly and platform independent IDE and in the market, there is not many products working as desktop applications to satisfy the developers' needs.

1.3 Statement of scope

Our project, CEYLAN, is an integrated development environment for developing web pages with Ajax. Our aim is to offer a user friendly and functional IDE which will additionally include database connection and configuration wizard for the developers from different backgrounds; professionals or amateurs. We will develop a desktop application which doesn't need an internet connection to run. Our development environment will be developed regarding the following four basic design principles;

- User friendliness
- Modularity
- Platform Independency
- Consistency

Our IDE's interface will be graphical and user friendly. It will work regardless of operating system and the code developed using it will run in different browsers. It will be consistent; will not surprise the user presenting unexpected behaviors. Concerning these design parameters, our product will include the following features;

- Widgets such as tree view, tool bars and tabs
- Server connectivity

- Extensive tools for development and debugging
- Syntax highlighting
- Syntax checking
- Auto completion
- Error handling
- Code indentation
- Help menu
- Integrated browser
- Code and design view
- Predefined code generation
- Customizable user interface
- Database connection wizard
- Views of database tables
- Displays of SQL queries

1.4 Application Areas of Our Software

Our software will be used in most of the areas where web application development is the issue; by companies, individual developers, students in educational concepts, etc. Our target user profile consists of the whole software developers using Ajax technologies. Our product will be able to serve more than enough functionality for users having different amount of knowledge and experience.

2. MARKET RESEARCH

2.1 Reasons Why Developers Chose AJAX

The intent of applications developed by AJAX is to make web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire web page

does not have to be reloaded each time the user makes a change. This is meant to increase the web page's interactivity, speed, and usability. Because of this, more and more people started to use AJAX for developing web applications.

Inspecting further, we went into details of the reasons stated above so that we will decide on properties of the features accordingly. Below, we have outlined that reasons(*):

1. Using ASP (Active Server Pages), the cost of switching is reduced. The raise in the performance of the web page can be easily observed. GMail and Yahoo Mail are the well known pages including ASP.
2. Drag-and-drop portlets eliminated the need for awkward layout and content pages.
3. Form driven activities are faster with AJAX. Click on the edit link to instantly start changing tags, click on the submit button to asynchronously send off changes to the tags and quickly see in place what changed, no reloading the entire page.
4. In applications with deep hierarchical tree navigation Ajax lessen the burden on the server by lazy loading deep hierarchy data. The whole web page need not to be loaded to see a one line response.
5. In a message posting application, user has to reload the page over and over to see the recently posted threads in classic approach. With AJAX, responses are more instant.
6. Voting, Yes/No boxes, Ratings submissions are places where AJAX is used. By reducing the time and impact of clicking on things, Ajax applications become a lot more interactive - if it takes a 40 seconds to register a vote, most people would probably pass unless they really care. If it takes 1 second to vote, a much larger percentage of people are likely to vote.
7. Interactive Errors are quickly sent to user. A user does not wait a long time just to receive the error message. Ajax can speed up this workflow by quickly letting the user know of an error condition before they try to submit.
8. [Ajax](#) can be used to load pages more efficiently. The load on the servers in bandwidth and resources are reduced.

* http://www.sourcelabs.com/blogs/ajb/2005/12/10_places_you_must_use_ajax.html and “10 Business Reasons to Use AJAX” article from blogs.pathf.com/agileajax/2006/05/10_business_rea.html are referred.

2.2 Tools on the Market for AJAX Development

There are some development tools in the market right now for AJAX. Some are standalone tools and some are plugins for main development environment (mostly Eclipse).

2.2.1 EchoStudio2 :

The EchoStudio2 Eclipse plug-in enables point-and-click AJAX web application development using the Echo2 framework which is a platform for developing web-based applications that approach the capabilities of rich clients. With EchoStudio2, forms can be created visually using a WYSIWYG (What You See Is What You Get) interface. An application's look-and-feel can be defined using the integrated style editor.

EchoStudio2 Features include:

- A visual form editor enables the creation of complete user interfaces as well as reusable Echo2 components using a point-and-click interface.
- A visual style sheet editor creates reusable themes to define the look-and-feel of applications.
- Wizard-driven development tools automate the creation of projects, forms, and style sheets.
- Applications can be run and debugged entirely within the EchoStudio2 environment.
- Highly integrated support for localizing web applications, including the ability to instantly preview forms with different locales as they are being edited.

For more information <http://www.nextapp.com/platform/echo2/echostudio/>

2.2.2 Eclipse ATF :

The AJAX Toolkit Framework (ATF) provides an extensible framework and exemplary tools for building IDEs for the many different AJAX runtime offerings (Dojo, Zimbra, Rico, etc) in the

market. Tools built upon these frameworks will initially include: enhanced JavaScript editing features such as edit-time syntax checking; an embedded Mozilla web browser; an embedded DOM browser; and an embedded JavaScript debugger. An additional and unique aspect of the framework is the Personality Builder function, which assists in the construction of arbitrary AJAX runtime frameworks, thus allowing those runtimes to be used with ATF tools.

Features:

- Enhanced JavaScript editor, syntax highlighting, syntax checking/error display, code completion etc.
- Browser tooling editor, browser editor, DOM inspector, CSS view.
- Embedded browser.
- JavaScript debugging

For more information <http://www.eclipse.org/atf/>

2.2.3 MyEclipse AJAX Support :

MyEclipse is enterprise j2ee development workbench. It has got a variety of development tools for modelling and code generation, web development, etc. And MyEclipse AJAX is one of them.

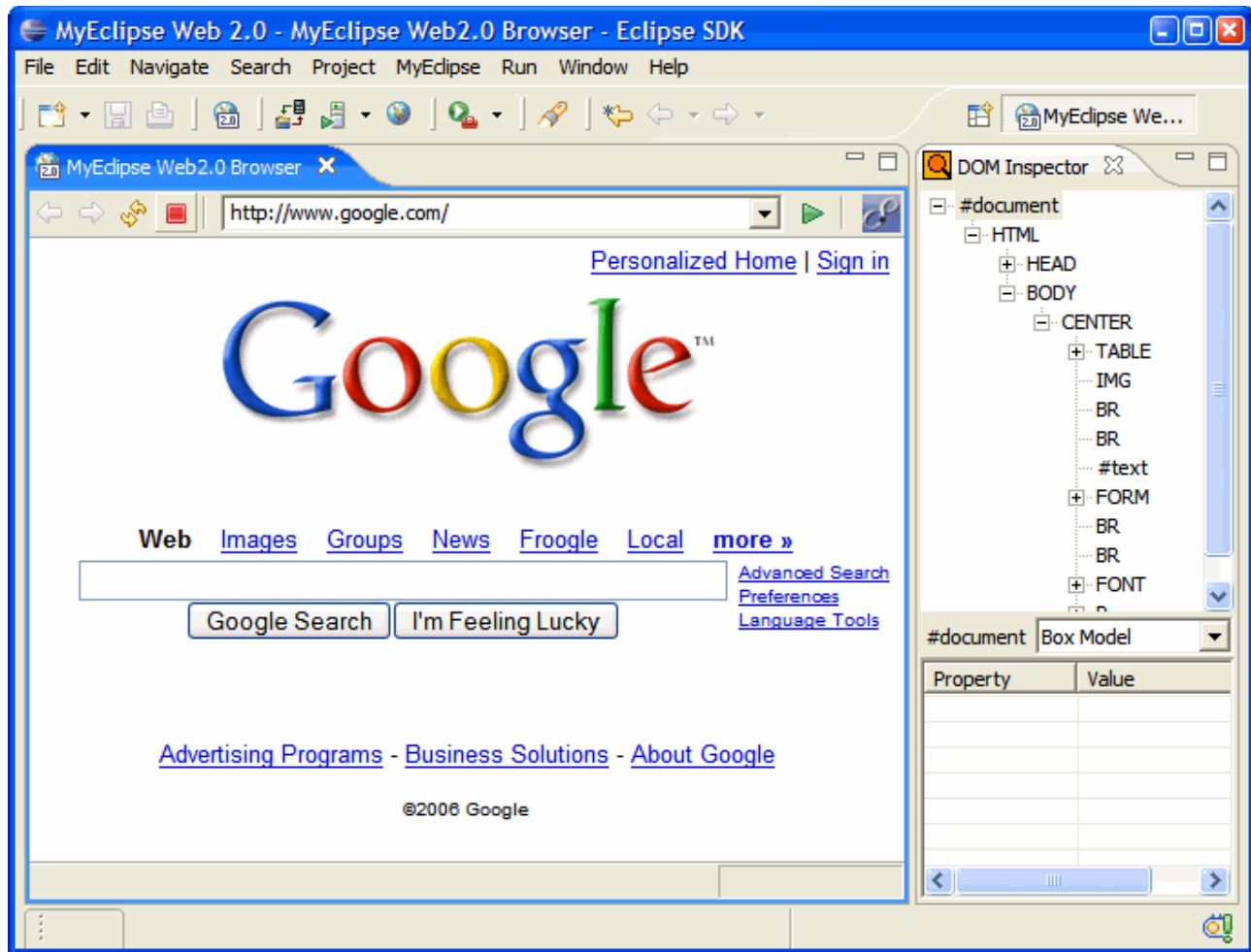
It includes the following features:

- Integrated Web 2.0 Browser : provides the foundation for features such as the DOM Inspector and the JavaScript Debugger.
- DOM Inspector : displays the current browser's document object model (DOM) in real-time. Using the DOM Inspector a developer is able to inspect node attributes and navigate to other nodes in the tree.
- Integrated JavaScript Debugger : debugs any web page containing JavaScript source or included JavaScript files, or standalone JavaScript files. User can set breakpoints in JavaScript files.
- Advanced JavaScript Editor : syntax highlighting, validation and syntax checking, code

assist etc.

MyEclipse AJAX is working under only Windows operating systems right now.

For more information <http://www.myeclipseide.com/ContentExpress-display-ceid-70.html>



2.2.4 Google Web Toolkit :

The Google Web Toolkit (GWT) is a free toolkit to develop AJAX applications in the Java programming language. GWT supports rapid client/server development and debugging in any Java IDE. Different from other tools, GWT compiler translates a working Java application into equivalent JavaScript that programmatically manipulates a web browser's HTML DOM using DHTML techniques.

Google Web Toolkit Features

- Browser Compatible : GWT supports IE, Firefox, Mozilla, Safari and Opera.

- Debugging : GWT runs in the Java virtual machine at development time. This means that user gets Java debugging different from other tools.
- Dynamic, reusable user interface components : easy usage of widgets.
- JUnit integration : lets user unit test both in a debugger and in a browser.

For more information <http://code.google.com/webtoolkit/>

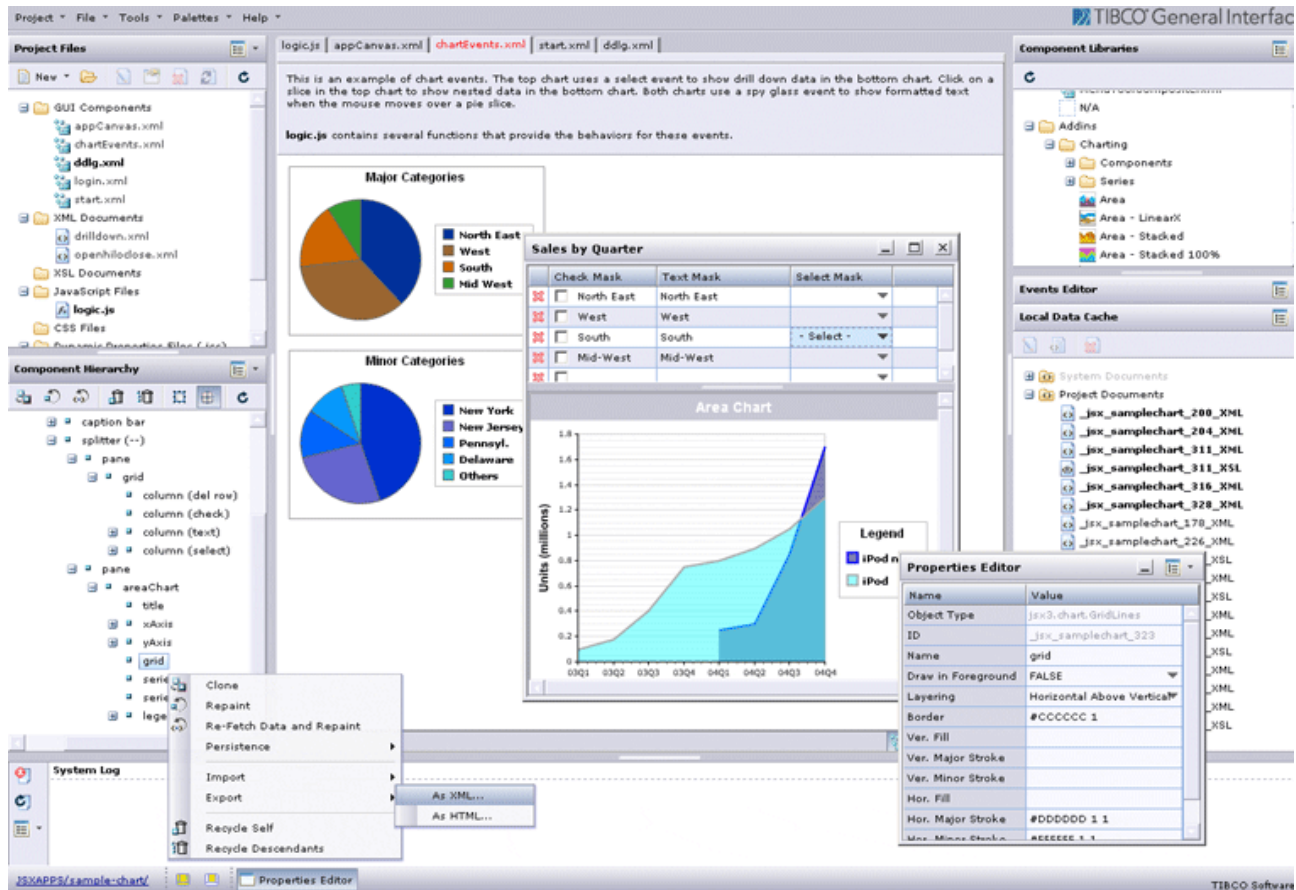
2.2.5 TIBCO General Interface :

TIBCO General Interface is AJAX Rich Internet Application (RIA) toolkit that lets users capitalize on the lower costs of Web applications while delivering the rich graphical look and feel of desktop-install software and components. TIBCO General Interface lets users quickly create sophisticated web-based applications that run in a standard web browser without plugins, Active-X controls, Java applets or client side software installation.

Features:

- Customizable GUI
- Drag and drop property
- Reusable components
- Wizard-driven connectivity to data sources
- Step-through debugging
- Code generation and completion

For more information <http://www.tibco.com/devnet/gi/>



2.3 Expectations of Developers from an AJAX Development Studio

Apart from examining existing tools we have prepared a query (provided in Appendix A) and we have asked questions in the forums about what a web developer can expect from an AJAX IDE. We have made analysis on what our customers expect from our development studio according to those results and consequently we have figured out the following characteristics that are expected from a development studio.

- ✓ It should not re-format the code without intend and user should not be restricted to some predefined formats.
- ✓ Javascript files should be passed to Just in Time Syntax Check.
- ✓ There should be an integrated editor with ease of use properties such as highlighting, syntax checking and error validation.
- ✓ There should be an embedded browser.

- ✓ There should be an integrated debugger having debug commands such as breakpoint, step in, step out etc.
- ✓ There should be post syntax checking, compression and obfuscation for JS side.
- ✓ It is preferable that it makes a good memory utilization.
- ✓ It should not have Memory Leak problems.
- ✓ It should be platform independent.
- ✓ It is preferable that it is a desktop application rather than a web based one.

3. PROJECT REQUIREMENTS

3.1 Functional Requirements

3.1.1 Widgets such as tree view, toolbars and tabs

User will be provided with widgets such as tree view, toolbars and tabs. With these tools it'll be more easy for the developer to follow and manage her/his work.

3.1.2 Server connectivity

Connection Wizard: Connection operation is done via a wizard. User chooses appropriate options step by step. Options are database address, connection engine, connection name, etc

3.1.3 Query Execution

- SQL Command Screen: Executes Sql queries and returns results in dos like command screen .
- SQL Executer Interface: Similar Sql operations “Sql Command Screen” supplied in more userfriendly interface. And some suboperations are provided which are as follows:
 - Save SQL
 - Load SQL
 - Execute SQL
 - Prepared SQL

3.1.4 Editing, and Debugging

There are some features provided with for editing and debugging. These are;

- Syntax Highlighting: The code will be highlighted according to specified types, keywords and functions.
- Code indentation: User can indent the file or some part of the file.
- Auto-completion: While the developer is writing the code, upon her/his request s/he will be provided with auto-completion. This completion can be done using libraries or user code.
- Error-handling: When the user interpretes the code errors are detected and the user will be provided with the appropriate error message.
- Run : Invokes interpretation via GUI.
- Debug : Starts debug mode. User can set breakpoints or do step in step out etc.

3.1.5 Work Space

The user is provided a workspace with the features below:

- Hotkeys : Hotkeys of editor functions is provided. Simply as : ctrl+c copy ctrl+v paste.
- Auto Completion : Auto completion of code. Following code is generated according to the code part we have.
- Split View: Partitioned design screen is available. Design screen is divided as Code&Design.
- File Tree: Organization of files represented as tree view.
- Drag&Drop: By simply dragging and dropping an item into design screen, user can generate code.

3.1.6 Language support

The user will be provided with both English and Turkish Language support not in error handling for instance but in other GUI components.

3.1.7 Help menu

Help menu will contain:

- Tutorial : Usage of program is explained step by step.
- FAQ : Some frequently asked questions and answers.
- Online Help : Link to internet site.

3.1.8 Menu Components

Probable basic menu items are as follows:

3.1.8.1 File

- New: Creates a new file or project. Type of new file can be chosen among possible types. Possible types are: CSS, HTML, XML, **JS**. While opening a project, user can determine the project properties such as workspace, name, etc.
- Open: Opens previously created file or project.
- Close: Close project or file.
- Close all : Close all open jobs.
- Save: Save file/project with previous properties as save location, name, extension, etc.
- Save as: Save file/project with desired properties as save location, name, extension
- Print : Prints Code. Print options can be determined through interface.
- **External View : User can see the result of his/her code in a browser, Internet Explorer, Mozilla Firefox.**
- Import Project : Imports an old project

3.1.8.2 Edit

Basic editor operations can be used which are

- Undo,
- Redo,
- Cut,
- Copy,
- Paste,
- Delete,
- Find,
- Find Next,
- Find Previous
- Find & Replace.

3.1.9 Integrated browser

There will be an integrated browser which pops up after the code is interpreted and ready for display. So the user can see what s/he does on a web page view.

3.1.10 Code and design view

While the user is editing her/his codes/he will be able to see her/his code as code view, design view or split view.

3.1.11 Predefined code generation

When the user is in design or split view and butts a figure on design part, the corresponding predefined code will be automatically put exact place in code file.

3.2 Non-Functional Requirements

3.2.1 User-friendliness

Our aim is to provide a user-friendly program to user. In our project we can accomplish this goal with two components; an User Friendly Interface and Easy Coding Features. GUI provides easy-to-understand and easy-to-use interface. User won't lose within menus. Functions which are similar are grouped. Our program provides Drag&Drop operations. With the help of this user can create applications very easy.

3.2.2 Modularity

Modularity is important in our design because with the help of it we can add and remove components easily. We designed system as modules, this modules work together but one modules don't need to know internal design and internal processes of another module.

3.2.3 Platform Independency

We chose Java as programming language. Java choice is actually related Java Virtual Machine (JVM). We need a machine independent software and JVM provides us a virtual layer between software and machine. Thus we can write machine independent code. Our software will be used on every system which have JVM.

3.2.4 Consistency

Our system will behave in a predetermined manner. We will lessen the unexpected states or behaviors of system as much as possible. To supply this, we will put the system behavior on a deterministic finite automata in design phase.

3.3 System Requirements

3.3.1 Software Requirements:

Development Phase

Project CEYLAN will be developed in Java, since **Java** allows us to develop a platform independent code. Moreover, Java offers a wide range of functions to develop a high-quality graphical user interface. Thus **Java Runtime Environment** and **Java SDK** are the essential tools to be used in this project. Moreover, for reporting and preparing help menus we will use design tools such as **MS Visio, Borland Together**.

To develop our application, we will work in **Windows XP** and **Linux** platforms.

For the End User

Since our application will be developed in **Java** and will be platform independent, the end user will only need a compatible operating system and a Java **Runtime Environment**. And since we are using web service in the development phase, the user can choose whichever database he/she wants.

3.3.2 Hardware Requirements:

- 512 MB disk space
- 512 MB RAM (1024 preferable)
- Intel 500 MHz Processor (or above)

4. PROJECT SCHEDULE

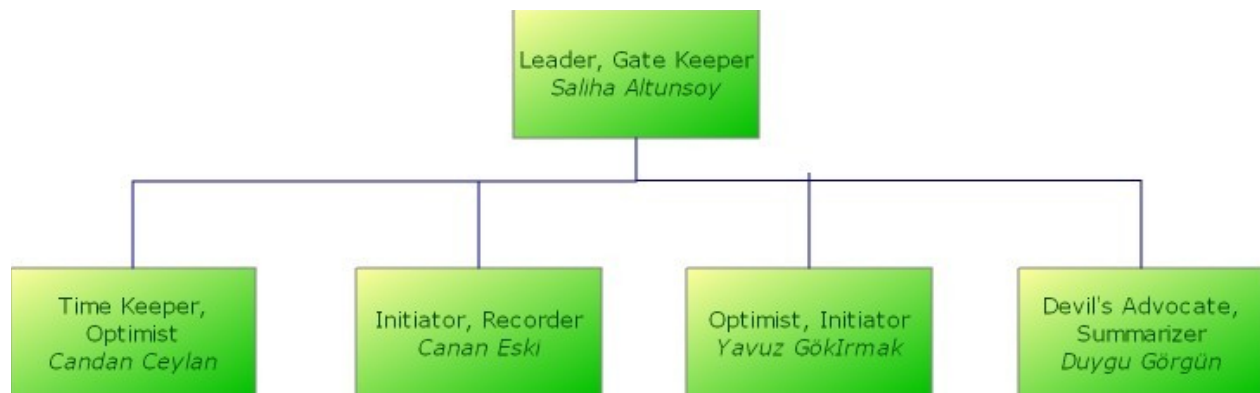
We are going to write design and final reports in coming two months. In 2 weeks time we will provide the teaching assistant with a user interface prototype. By the end of the semester we will have been provided the instructor and TA with a demo of our project. Details of the project can be seen in Gantt Chart provided in Appendix B.

5. TEAM ORGANIZATION & PROCESS MODEL

5.1 Team Structure

We have decided our team structure to be Controlled Decentralized (CD). Because we have a leader to keep all the work on the road. But we all have equal contributions and offerings to the project.

5.2 Role of Team Members



5.3 Process Model

We have chosen Evolutionary (Spiral) Model as process model. Because our project scope and requirements are wide. As new modules are added it has to be revised again and again.

6. RISK MONITORING AND MANAGEMENT PLAN

In order not to come up with an error, we must define some potential risks and devise methods to overcome them. Hence we should have a risk management plan to overcome the possible risks that may have different impacts on the project.

6.1 Possible Risks

The main goal should be identifying and understanding possible risks, monitoring them, managing them and dealing with them with them. To be able to achieve this effectively, some methodology should be defined and used.

Process Definition: Due to the lack of experience of the staff, there may be some defects in the definition of the project and this will reveal some unexpected problems. Possible changes and mistakes in the design and analysis part or in the coding part can cause some unexpected problems. To avoid this effect, we try to do researches as much as we can.

Lack of Knowledge and Experience: Since none of our team members have a background on business world, it is highly possible that our project will be affected. To cope with this risk, we have decided to get into touch with some firms and some employees, and we will try to attend related seminars, courses, consult to the experienced people to learn about their methodologies and go through several books.

Development Environment: Project team is not experienced in development environment. The tools, programming language, servers and web service are new to the group members. Therefore there is a risk of inexperienced staff. Besides this, learning these requirements may require much more time than we planned. To avoid this, we will not be too strict about the methodologies, follow the ways our project requires us to do. In addition, we shared topics and in our meetings, the responsible member is giving information to others about his/ her part.

Technology to be Built: Since the technology changes rapidly there may occurs some consistency problems after release. In order to avoid this, we try to be more careful about new technologies.

Misunderstanding Requirements of Customers: Misunderstanding of customer requirements is also a common failure in software engineering. To avoid this risk, we are meeting with people from as many companies as possible, and trying to lower the risk that specifications may escape unnoticed. These companies are both developer companies and end users, so that we are exposed to different points of view. We are trying to be as clear as we can in terms of language and jargon, and repeat the information we get, asking for confirmative feedback. As the project progresses, we will develop prototypes incrementally and ask for feedback at regular intervals, so that any misleading can be noticed as early as possible.

Members' failure in performing due responsibilities : At this semester, being senior students, all of us have different technical elective courses, so it may be difficult for us to that we may fall behind our personal deadlines. To cope with this problem, we are trying to plan personal tasks in terms of weeks so that the members are able to plan their weekly schedule with respect to project progress. When one of us is very busy, s/he will inform others about his/ her work and the others try to help him/ her.

Staff Size: These risks depend on the technical ability, experience and enthusiasm of the team members. Besides this, team members have a chance to withdraw the course which means

increasing the work weight for each member. We hope anybody withdraws but if it occurs, we would work much more.

6.2 Risk Table

The risk table shows the risks that may causes a negative effect on the project. The risks are categorized according to the risk types defined before; also, their impacts and probabilities are stated. The risks are sorted according to their probabilities and their impacts. The Risk Mitigation, Monitoring and Management Plan (RMMM) will be prepared and the risks will be considered in this plan. For the risks, some additional plans (although not complicated as the RMMM) will also be developed so that their effects can be decreased.

Risks	Category	Probability	Impact	RMMM
Temporary unavailability of a member	ST	20%	3	Working harder for a while in order to handle
Problems with the development tool	TE	40%	3	Search for usage of development tools during development
Poor time management	ST	70%	2	Careful about time management and try to make changes in schedule
One of the team members quits course	ST	5%	21	Sharing responsibilities between members
Low attendance to group work	ST	10%	2	Try to make strict rules and warning group members
Effects of other courses	ST	70%	2	Try to do the time management efficiently and develop the project through the plan

Impact values:

1-catastrophic

2-critical

3-marginal

4-negligible

ST= Staff size and experience

TE= Technology to be built

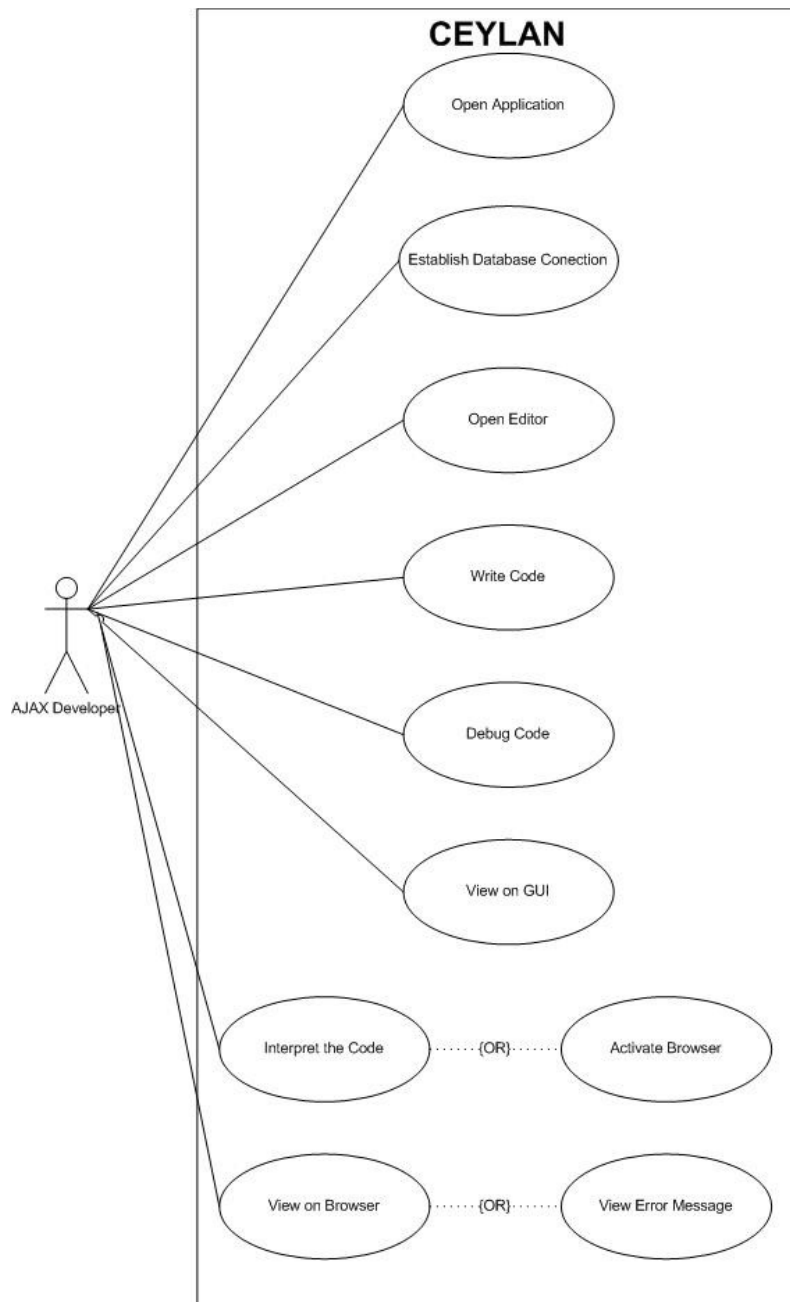
7. USAGE SCENARIO

Use-cases

Use Case: Writing AJAX code and invoking it on the browser.

Actor: The AJAX Developer

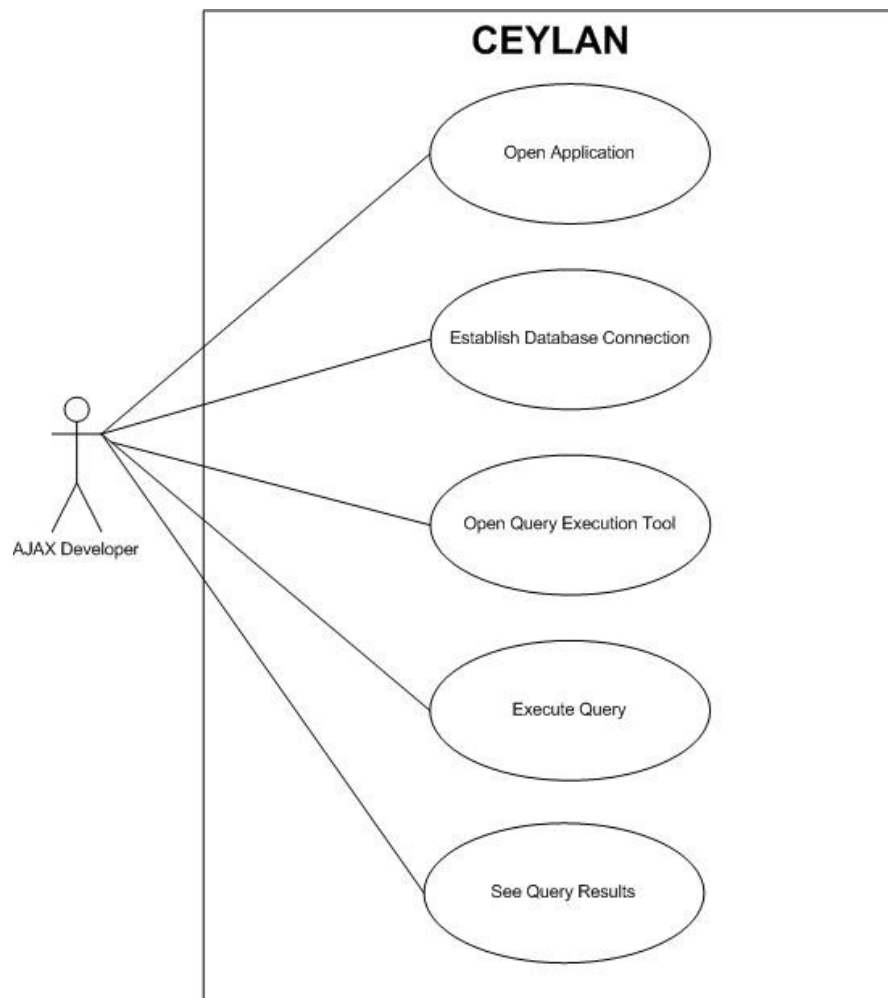
1. The user opens the development studio.
2. If the database connection is not established the user establishes the connection.
3. S/he opens the editor.
4. S/he writes the code.
5. S/he invokes the debugger to debug his/her code.
6. S/he views the outcome of her/his code as code, design or split.
7. S/he interprets her/his code or s/he directly types on browser to invoke interpreter..
8. S/he views the result on browser if the code works properly else s/he receives an error message.



Use Case: Executing query and viewing the results.

Actor: The AJAX Developer

1. The user opens the development studio.
2. If the database connection is not established the user establishes the connection.
3. S/he opens the query execution tool.
4. S/he executes a query.
5. S/he views the result the result of the query on query execution tool.

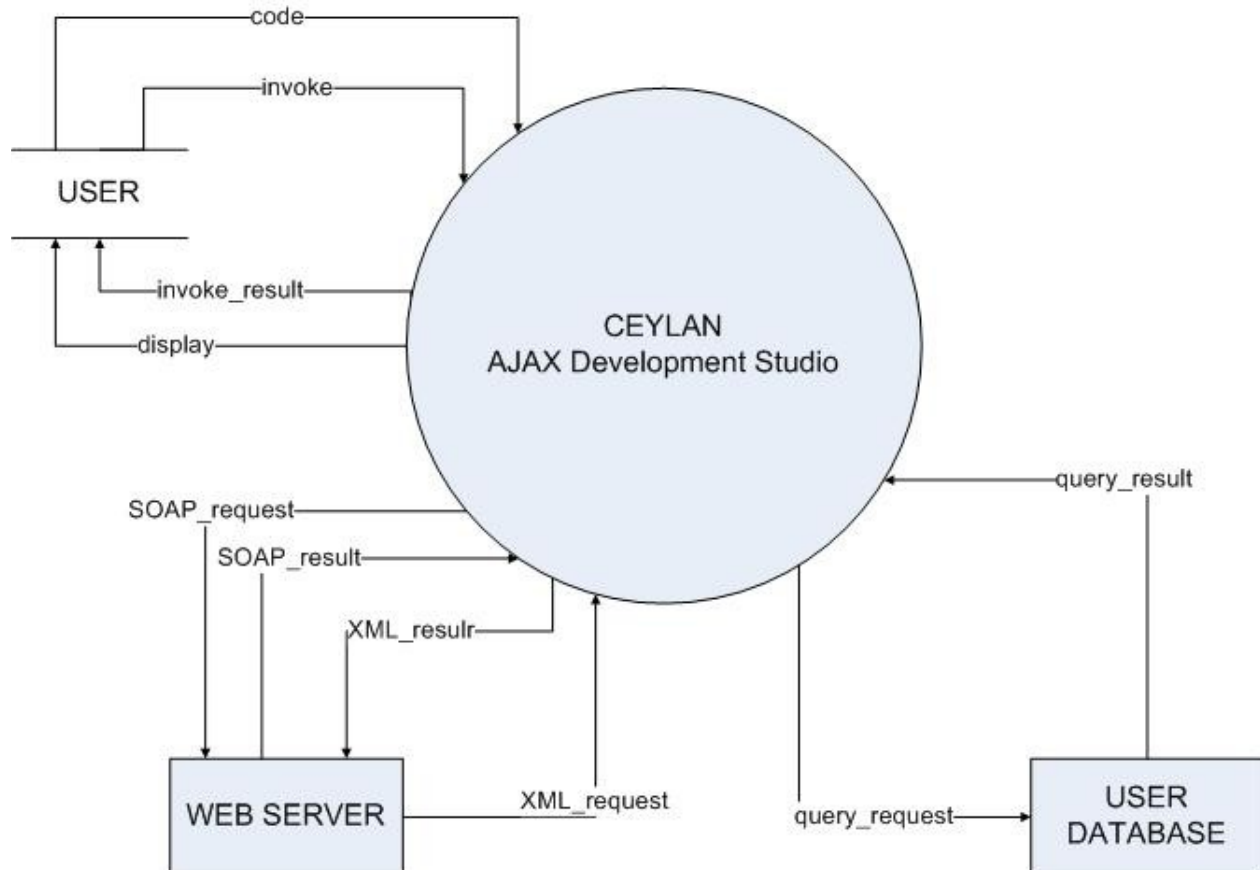


8. MODELLING

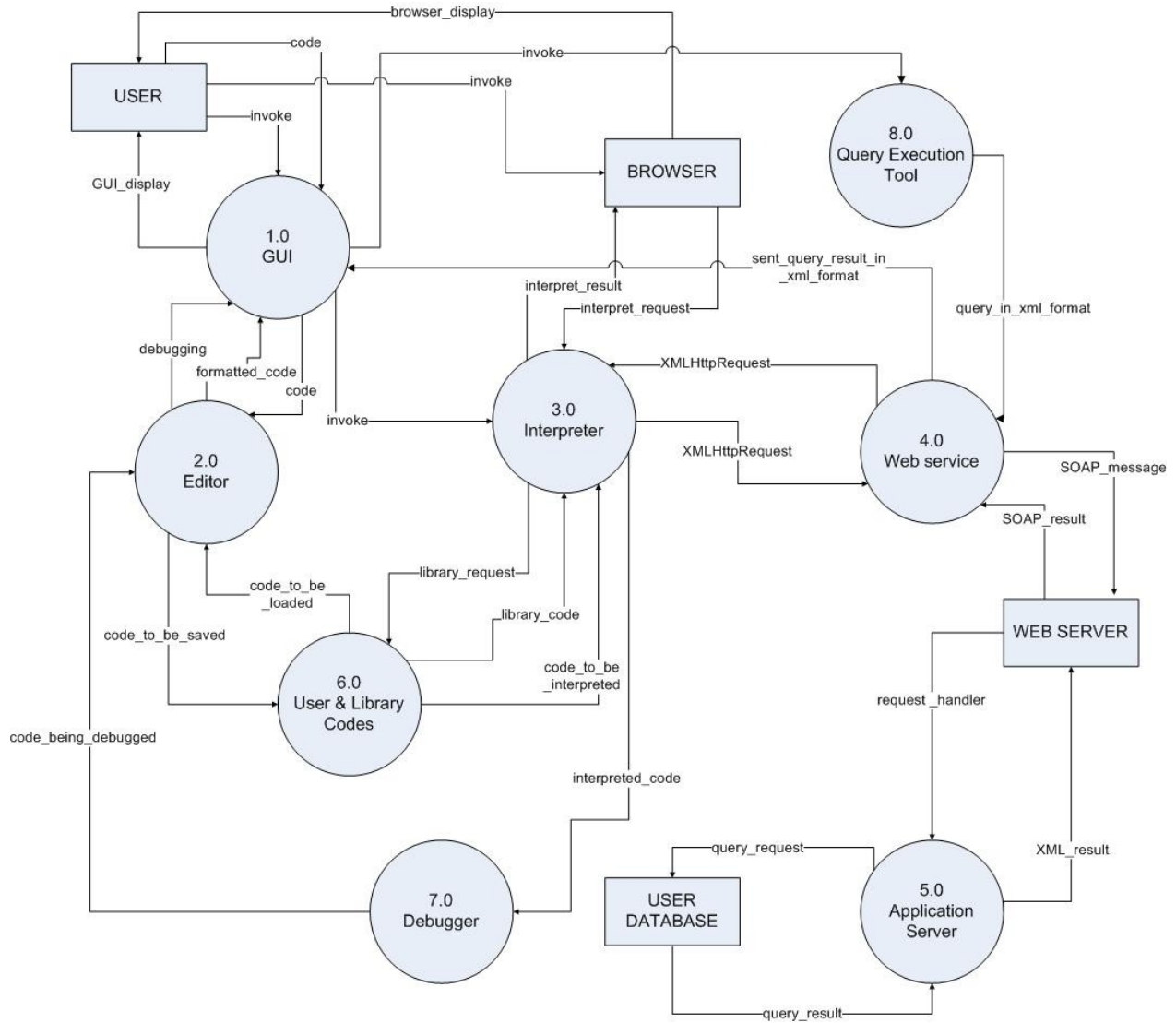
8.1 Functional Modeling

8.1.1 Data Flow Diagrams (DFD)

8.1.1.1 DFD Level 0



8.1.1.2 DFD Level 1



8.1.2 Process Specifications (PSPEC)

CEYLAN

This is our AJAX Development Studio consisting of components GUI, Editor, Interpreter, Web Service, Application Server, User & Library Code, Debugger and Query Execution Tool.

1.0 GUI

This process includes all the graphics view provided by our development studio. Menu, tree view, query execution and view, code-design-split view, etc. are all passed to user via GUI. It takes invocation from user which is passed to Interpreter or Query Execution Tool depending on the request. Code is also passed by user to Editor via GUI. Formatted code sent by Editor, Debugging Process and XML query sent by web service are all viewed by user via GUI.

2.0 Editor

Editor is provided with code from GUI. This coded is formatted(highlighted, indented) by editor and passed back to GUI. If user wants to save the code, this code is then passed to User & Library Codes process. Saved code is loaded back upon user request. The code user wants to debug is processed on editor.

3.0 Interpreter

This process is invoked via browser or GUI. After being invoked it can make a request for library code to User & Library Codes process and provided with the necessary code. This process includes both interpreter for javascript code and parser for XML and HTML code. The result can be in display format and passed to Browser. Interpreted code can also be passed to Debugger for debugging. It also sends XMLHttpRequest and gets the result to/from Web Service.

4.0 Web Service

Web service coverts all the data it is provided to XML format and puts into SOAP “Envelope”. Also deformats the messages sent via SOAP to XML format

5.0 Application Server

“They take information from the user, create SQL statements, query the database, and then format and retune dynamic content based on the queries”*. There will be CGI in application server to make conversion between XML requests and SQL queries.

6.0 User & Library Codes

Includes system specific library codes and user provided library codes. It takes code to be saved from Editor and passes it pack the saved code. It also takes a request from Interpreter and provides it with the requested code if available.

7.0 Debugger

Debugger takes/passes interpreted code/code to be debugged from/to Interpreter/Editor. It provides user with breakpoint, step in, step out etc. features.

8.0 Query Execution Tool

Without providing any code, just to see what a query returns, user enters a query to Query Execution Tool. Tool makes a request from database via web service and the result is passed user via GUI.

*Taken from <http://www.counterpane.com/application-security-assessment.pdf>.

8.1.3 Data Dictionary

Name	Component Invoke
Alias	invoke
Description	This data includes information about which component is going to be invoked. Given by user to the GUI and passed to other components accordingly. If it is an invoke for Query Execution Tool it is an SQL query. invoke = [button invoke method invoke]
Output from	User GUI 1.0
Input to	Browser GUI 1.0 Editor 2.0 Query Execution Tool 8.0
Format	[An activation method SQL query]

Name	User code
Alias	code
Description	This data includes all the source code written by user.
Output from	User GUI 1.0
Input to	GUI 1.0 Editor 2.0
Format	[XML HTML Javascript]

Name	Display
Alias	[GUI_display browser_display]
Description	This is the graphical user interface provided to user by GUI and browser. All the interaction between the user and the system is via the display.
Output from	Browser GUI 1.0
Input to	User
Format	Graphics****

<i>Name</i>	Formatted code
<i>Alias</i>	formatted_code
<i>Description</i>	This is the code formatted according to the features supported by editor (eg. highlighting).
<i>Output from</i>	Editor 2.0
<i>Input to</i>	GUI 1.0
<i>Format</i>	Formatted [XML HTML Javascript]

<i>Name</i>	Interpretation Request
<i>Alias</i>	interpret_request
<i>Description</i>	This information includes request passed to interpreter for the specified file.
<i>Output from</i>	Browser
<i>Input to</i>	Interpreter 3.0
<i>Format</i>	An activation method for interpretation

<i>Name</i>	Code to Be Interpreted
<i>Alias</i>	code_to_be_interpreted
<i>Description</i>	This is the code passed from library to interpreter.
<i>Output from</i>	User & Library Codes 6.0
<i>Input to</i>	Interpreter 3.0
<i>Format</i>	[XML HTML Javascript] *****

<i>Name</i>	Interpretation Result
<i>Alias</i>	interpret_result
<i>Description</i>	This data includes the executed format of the code. This format appears in browser
<i>Output from</i>	Interpreter 3.0
<i>Input to</i>	Browser
<i>Format</i>	Parsed [XML HTML]

<i>Name</i>	Interpreted Code
<i>Alias</i>	interpreted_code
<i>Description</i>	This data includes the interpreted form of the code. This form is a kind of pre_display form(its difference from interpretation result).
<i>Output from</i>	Interpreter 3.0
<i>Input to</i>	Debugger 7.0

Format	[XML HTML]
---------------	------------

Name	Code to Be Saved
Alias	code_to_be_saved
Description	This data includes the source code written by user and is passed 6.0 for save.
Output from	Editor 2.0
Input to	User & Library Codes 6.0
Format	[XML HTML Javascript]

Name	Code to Be Loaded
Alias	code_to_be_loaded
Description	This data includes the source code written by user and saved before request for load.
Output from	User & Library Codes 6.0
Input to	Editor 2.0
Format	[XML HTML Javascript]

Name	Library Request
Alias	library_request
Description	This information includes request passed to 6.0 for necessary library codes.
Output from	Interpreter 3.0
Input to	User & Library Codes 6.0
Format	A request method for specified library

Name	Library Code
Alias	library_code
Description	This data includes the library code requested by 3.0
Output from	User & Library Codes 6.0
Input to	Interpreter 3.0
Format	[*****]

Name	Code Being Debugged
Alias	code_being_debugged
Description	This data contains to code that is being debugged. It has been interpreted.

<i>Output from</i>	Debugger 7.0
<i>Input to</i>	Editor 2.0
<i>Format</i>	[HTML XML Javascript]

<i>Name</i>	XMLHttpRequest
<i>Alias</i>	XMLHttpRequest
<i>Description</i>	An API to transfer and manipulate <u>XML</u> , text data or other formats.
<i>Output from</i>	Interpreter 3.0 Web Service 4.0
<i>Input to</i>	Interpreter 3.0 Web Service 4.0
<i>Format</i>	Application programming interface

<i>Name</i>	SOAP Message
<i>Alias</i>	SOAP_message
<i>Description</i>	An XML message covered by SOAP.
<i>Output from</i>	Web Service 4.0
<i>Input to</i>	Web Server
<i>Format</i>	SOAP formatted message

<i>Name</i>	SOAP Result
<i>Alias</i>	SOAP_result
<i>Description</i>	A protocol to exchange XML-based messages over network using HTTP.
<i>Output from</i>	Web Server
<i>Input to</i>	Web Service 4.0
<i>Format</i>	An XML message covered by SOAP.

<i>Name</i>	Query in XML format
<i>Alias</i>	query_in_xml_format
<i>Description</i>	This is the XML format of the typed by user.
<i>Output from</i>	Query Execution Tool 8.0
<i>Input to</i>	Web Service 4.0
<i>Format</i>	XML

<i>Name</i>	Sent Query Result in XML Format
<i>Alias</i>	sent_query_result_in_xml_format

Description	This is a query in XML format sent from database as a result of query executed by user.
Output from	Web Service 4.0
Input to	GUI 1.0
Format	XML

Name	Request Handler
Alias	request_handler
Description	This is the request for the specified query..
Output from	Web Server
Input to	Application Server 5.0
Format	A request method

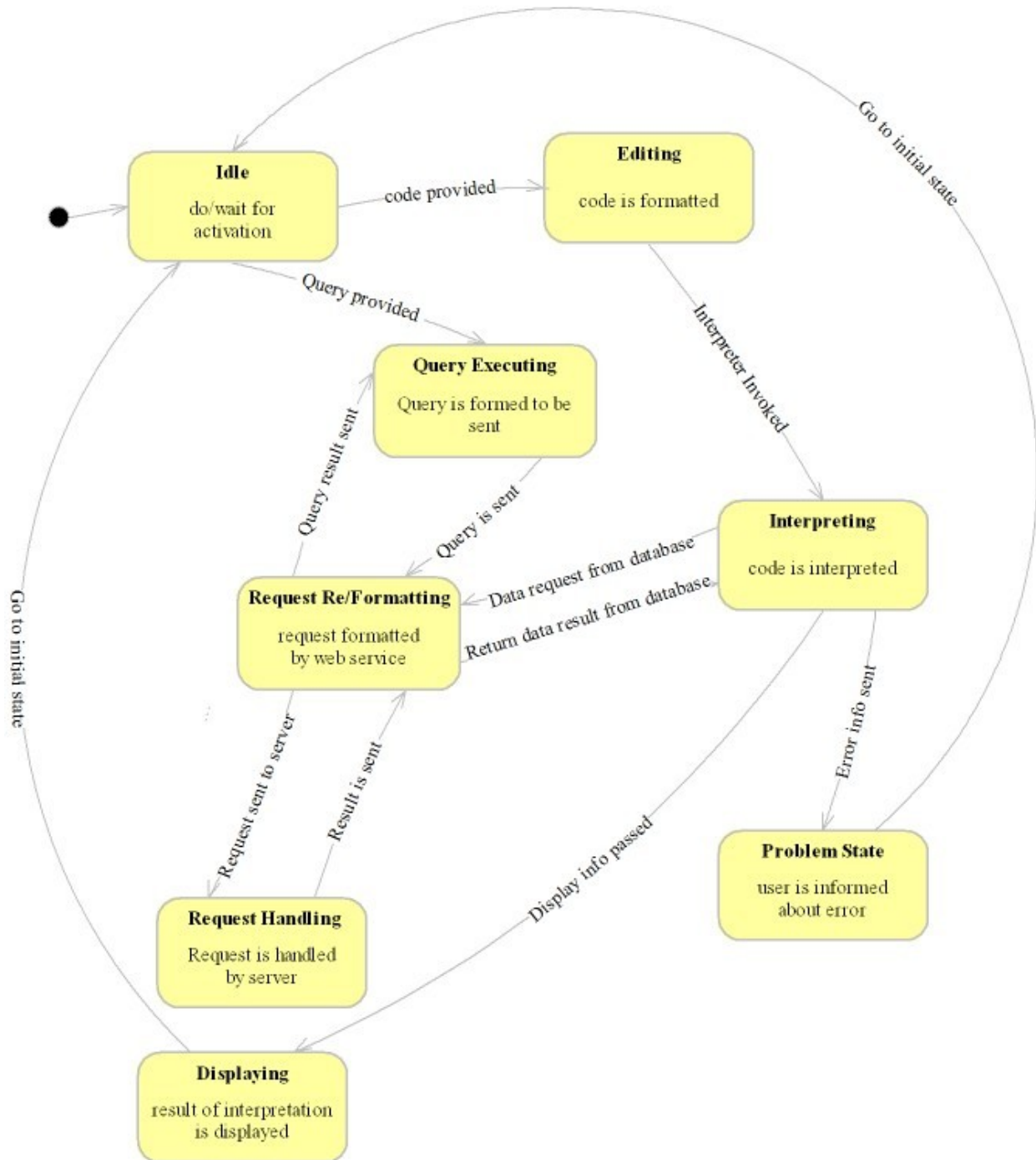
Name	XML Result
Alias	xml_result
Description	This is the XML format of the data requested from database.
Output from	Application Server 5.0
Input to	Web Server
Format	XML

Name	Query Request
Alias	query_request
Description	This is the query passed by Web Server.
Output from	Application Server 5.0
Input to	User Database
Format	SQL query

Name	Query Result
Alias	query_result
Description	This is the result of query passed by application server.
Output from	User Database
Input to	Application Server 5.0
Format	SQL query result

8.2 Behavioral Modeling

8.2.1 State Transition Diagram (STD)



APPENDICES

Appendix A

Query

Web geliştirme sürecinde hangi teknolojileri kullanıyorsunuz?

Masaüstü uygulaması olarak çalışan web geliştirme ortamlarının verimi ve performansı hakkında ne düşünüyorsunuz(Eclipse, Dreamweaver gibi araçların sistem kaynaklarını tüketmesi)?

Ajax kullanıyor musunuz? (Bu soruya cevabınız evet ise 4. soruya geçiniz.)

3.a Ajax teknolojisi hakkında bilgi almak ve/veya kullanmak ister misiniz? (Bu soruyu cevapladıktan sonra sona gidiniz.)

AJAX geliştirmek için kullandığınız bir araç var mı?

Kullandığınız bir araç varsa;

4.a Kullandığınız araçta geliştirilmesini/eklenmesini istediğiniz özellikler var mı?

4.b Kullandığınız araçta en çok neyin eksikliğinden dolayı sıkıntı çekiyorsunuz?

5. AJAX geliştirmek amaçlı bir yazılımın geliştirilmesine katkıda bulunmak ve sonrasında bu aracı kullanmak ister misiniz?

5.a Geliştirilecek bir aracın şu an kullanmakta olduğunuz bir geliştirme ortamına eklenti olarak konulmasını mı tek başına çalışan bir uygulama olmasını mı istersiniz?

5.b Geliştirilecek aracın masaüstü uygulaması mı web üzerinden çalıştırılacak bir uygulama mı olmasını istersiniz?

5.c Geliştirilecek araçta olmasını istediğiniz özellikler nelerdir? Hangi ekstra özellikler geliştirilecek aracı seçmenize sebep olur?

5.d Geliştirilecek aracı hangi platformlarda ve hangi araçlarla birlikte çalıştırmak istersiniz(Mozilla Firefox, Internet Explorer, Apache, Linux, Windows, MacOS vb.)?

—Eklemek istediğiniz fikirler/öneriler var mı?

Gantt Chart

