

MIDDLE EAST TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

REQUIREMENT ANALYSIS REPORT
FALL 2007

AQUT
Anatolian QUalified Technology

Table of Contents

1. INTRODUCTION.....	3
2. PROBLEM DEFINITION & PROJECT SCOPE	3
2.1. PROBLEM DEFINITON.....	3
2.2. PROJECT SCOPE.....	5
3. RESEARCH.....	6
3.1. LITERATURE SURVEY	6
3.1.1. AJAX.....	6
3.1.2. JABBER SERVER	7
3.2. MARKET RESEARCH.....	8
3.3. CONTACTS	12
4. PROCESS.....	13
4.1. TEAM STRUCTURE.....	13
4.2. PROCESS MODEL.....	15
4.3. MAJOR CONSTRAINTS.....	15
4.3.1. HARDWARE CONSTRAINTS	15
4.3.2. SOFTWARE CONSTRAINTS	16
4.3.3. TIME CONSTRAINTS	16
4.3.4. EXPERIENCE CONSTARAINTS.....	16
5. REQUIREMENTS	17
5.1. FUNCTIONAL REQUIREMENTS	17
5.1.1. SYSTEM FUNCTIONALITY.....	17
5.1.2. CORE FUNCTIONALITY.....	18
5.2. NON-FUNCTIONAL REQUIREMENTS	19
5.2.1. USER FRIENDLINESS	19
5.2.2. PORTABILITY.....	19
5.2.3. PERFORMANCE	19
5.3. SYSTEM REQUIREMENTS.....	20
5.3.1. HARDWARE REQUIREMENTS	20
5.3.2. SOFTWARE REQUIREMENTS	20

6. USAGE SCENARIO	21
6.1. USER PROFILE.....	21
6.2. USE CASE DIAGRAMS	21
6.2.1. USER USE CASE DIAGRAM.....	21
6.2.2. ADMIN USE CASE DIAGRAM	22
6.2.3. QA ASSISTANT USE CASE DIAGRAM.....	23
6.2.4. QA USE CASE DIAGRAM	23
7. MODELING	24
7.1. DATA MODELING	24
7.1.1. ENTITY RELATIONSHIP DIAGRAM.....	24
7.1.2. ER EXPLANATION	25
7.2. FUNCTIONAL MODELING	26
7.2.1. DATA FLOW DIAGRAM (LEVEL 0)	26
7.2.2. DATA FLOW DIAGRAM (LEVEL 1)	27
7.2.3. DFD (LEVEL 1) EXPLANATION	28
8. SIMPLE PROTOTYPE	29
9. GANTT CHART	30
10. REFERENCES	32

1. INTRODUCTION

As the web technology evolves, people start to spend most of their times on the internet. Internet is being used as communication and information gathering environment at any time. In other words, people now socialize on the internet. Like in the real world, it becomes an important issue to bring people together on the web. On the other side, giving people information about what they need as quick as possible becomes another important issue. According to these needs, one of the most important necessities of the internet is now an instant messaging platform that is independent from the website with question answering agent. This is the project that we are planning to complete at the end of the next semester.

2. PROBLEM DEFINITION & PROJECT SCOPE

Understanding the problem and clarifying the purpose and goals of the project is the first important step that we need to take. After collecting enough background information and getting help from our assistant Çağatay Çallı and AGMLAB Information Technologies R&D director Güven Fidan, we have obtained the required information to be able to define the problem clearly and determine the project scope.

2.1. PROBLEM DEFINITION

Our project aims to develop a collaborative messaging environment based on Ajax and XMPP open platforms in the form of a Web 2.0 application. On the server side, JABBER will be used and on the client side, AJAX will be used. While JABBER server deals with instant messaging protocols, AJAX based user interfaces will make things work faster on the client side. By the help of these technologies we are going to develop Firefox/Internet Explorer plug-in that provides

users, on-page messaging environment. With these plug-in you can chat with anybody that is at the same webpage with you. Moreover this project will have question answering (QA) side. But instant messaging is the first issue that we need to handle. At this point JABBER really makes our job easier as we don't need to care too much about instant messaging mechanism because JABBER will do it for us most of the time. We only have to account for creating some plug-in to help JABBER server to overcome these messaging protocols.

On the other hand, we will develop an AJAX base extension. A toolbar will appear at a side of our browser and this application provides users on-page messaging environment. But this part has extra properties. One of them is, this extension has the ability to take many information from the webpage such as which webpage we are visiting at that moment, whether the webpage includes any tagged information or not and so on. This part will be achieved by the appropriate Java Scripts that we are going to develop. The other one is this part will also have a simple question answering part in it that provides the users certain information about visited web page. That is, if a user is interested in something on that side, then the agent should answer his question about the issue interested on that page. QA agent may also use user tagged information on the webpage to achieve this goal. The thing that we should keep in our minds is that QA is requiring more complex natural language processing (NLP) techniques than other information retrieval techniques.

In addition to these core requirements some more developments can be integrated into the project. For instance a user can leave notes on that webpage for himself/herself or for his/her friends. This can help a user to remember to look something on that site later. If note is left for a friend, this friend can see the note when he/she visit that page or the note can reach that friend immediately. Also the user can tag any information at the webpage to use it later, to share it with friends. Measuring the rating of web pages by looking which pages are visited by users can be added to the project. This shows users the web pages mostly visited in a day. This feature may help users to indicate web pages that are popular. Moreover, a user can vote for a web page. This feature gives a chance to users for evaluating web pages. All users can see the average note of that web page. Also users can see the web pages that are being visited by his/her friends at that time. Therefore he/she can visit that page and establish a contact with his/her friend on that page. These are some of the extensions that we are planning to develop, but all can be found in the system functionalities part of our analysis report.

In this project our aim is to satisfy web users' needs by performing the things that we mentioned above. This project will be named as "XILENT".

2.2. PROJECT SCOPE

Our project, XILENT, is a collaborative messaging environment based on Ajax and XMPP open platforms in the form of a Web 2.0 application. Our aim is to offer web users, user friendly instant messaging toolbar together with question answering agent with rich user interface. We will develop a Web 2.0 application which is independent from the webpage that is visited right now. Our project will be developed regarding the following basic principles:

- User friendliness
- Security
- Browser independency

By user friendliness, we mean Xilent will be an easy application to use and its features will be both clear enough and understandable with user interface. By browser independency, we mean our application will work on both Firefox and Internet Explorer. Finally our application will be secure. For instance a user is looking his/her mails in a password protected area other users can not able to follow this user and also our application will not have any right to reach the information in this area. Concerning these basic principles, Xilent will include these features:

- Users will have the ability to create their accounts
- Users will have the ability to login with their username and password
- Users can see which WebPages are popular among Xilent users
- Users can be visible to everybody or a group according to his/her choice
- Users will have the ability to leave notes on a webpage
- Our application will be efficient enough regarding time constraints

3. RESEARCH

3.1. LITERATURE SURVEY

3.1.1. AJAX

As we explained earlier, we will make an Ajax based GUI to our plug-in and in this part of our report, we will mention what Ajax contributes to our application with some references from the related articles.

First of all, since we will code with Java on Eclipse, Ajax provides compatible tools with that platform, and Sam Thompson, IBM Internet Technologies Solution Architect, asserts that "... IBM recently integrated a series of useful technologies into the Ajax Toolkit Framework to help address this problem. The ATF (Ajax Toolkit Framework), freely downloadable from the Apache site, provides an Eclipse-based Ajax development environment."

Another feature of Ajax, which brings innovation to web applications, is declared by Julie Hanna Farris. That is "AJAX enables advanced features like drag 'n drop, drop down menus and faster performance capabilities, which are now making their way into Web applications. These kinds of capabilities represent a significant leap in the advancement of Web apps. More than just creature comforts, they represent a major step forward in terms of usability, productivity and application functionality." This feature is useful for our project, because user can be able to leave notes on the WebPages, in order to locate the notes and change their position such a drag and drop mechanism is needed.

Finally, Sam Thompson gives a good example and reaction to an Ajax Application. He states that "Lastly, in my opinion, the true tipping point for Ajax came when Google released the beta version of its Ajax-based Google Maps application. Anyone who had previously used a mapping Web site quickly saw the advantage Google's mapping software had over other mapping sites on the Internet. While non-technical people were amazed and wondered how Google did it, programmers who looked under the covers found Ajax and began to ponder how they could use Ajax-based techniques to improve the usability and responsiveness of their own applications."

3.1.2. JABBER SERVER

The Jabber standards and architecture help create a distributed IM system, reminiscent of the email systems distributed across the Internet, with users connecting to these systems locally. This approach is diametrically opposed to the monolithic system architecture provided by such current service providers as AIM (AOL Instant Messenger), ICQ, MSN (Microsoft Network), and Yahoo, where a single central server or group of centralized servers provides the messaging service. Jabber also resembles the email architecture in other ways: Jabber addresses its end-points (humans, machines, software) with an addressing scheme almost identical to the basic SMTP (Simple Mail Transfer Protocol) scheme. For example, myname@elogex.com is a valid Jabber address or JID (Jabber ID) in Jabber parlance. For these reasons, Jabber-based systems scale better than existing proprietary systems. Additionally, the protocol allows for gateways to proprietary instant messaging services, should that become necessary.

Where does Jabber fit into this standardization effort? According to the Jabber Website, Jabber is "committed to fully support any open real-time messaging protocols, including the IETF protocol"; if and when support for this IETF protocol grows, Jabber aims to position itself as a "leading open source platform" for the IETF protocol. So far, the IETF effort has concentrated mainly on gathering requirements rather than implementation. For the moment, Jabber is the only open instant messaging and presence service protocol with significant open source support. As a result, it has become the de facto standard for open instant messaging.



WHICH JABBER SERVER?

For the purpose of getting yourself up and running with Jabber, the Jabber server you choose doesn't matter, since they all accept standard Jabber XML and communicate with the end-point application to deliver the payload, which is also standard Jabber XML. But among all Jabber servers OPENFIRE is the best considering server features such as connecting, disconnecting, authentication, registration, private messages and group chat. OPENFIRE also

allows the developer to create new services integrate directly into the server. Moreover OPENFIRE's score among these server features is %97.

3.2. MARKET RESEARCH

At the beginning of our research, we were all familiar with web 2.0 applications. However we had very little knowledge about extensions. In order to handle this issue we have made a market research and we have analyzed the similar applications at the market. These are some of them:

Chatsum

Chatsum is a plug-in for web browsers that provide its user conversation opportunities with other users who are on the same web page or on same site. You can see users on the same page or on whole site. Actually, it is not efficient when it is compared to other applications. Its features are restricted and do not serve much preferences to user.



- Figure 1: chatsum, at the left side of the browser -

Dai.sy

Da.isy is also an add-on for web browsers that serves its users social network. User can communicate with each other separately. Users can create their chat room by giving a name to that room. Users can show their videos and images to others by using links of them. Other users can watch the videos and see the images on the toolbar of Da.isy. What is interesting with Daisy is that users can make file transfer with each other. First a user chooses a folder from his computer for sharing. After that, other users can download the files from the pool that is composed of users' shared folders. Also, a small blog is designed for offline messages.



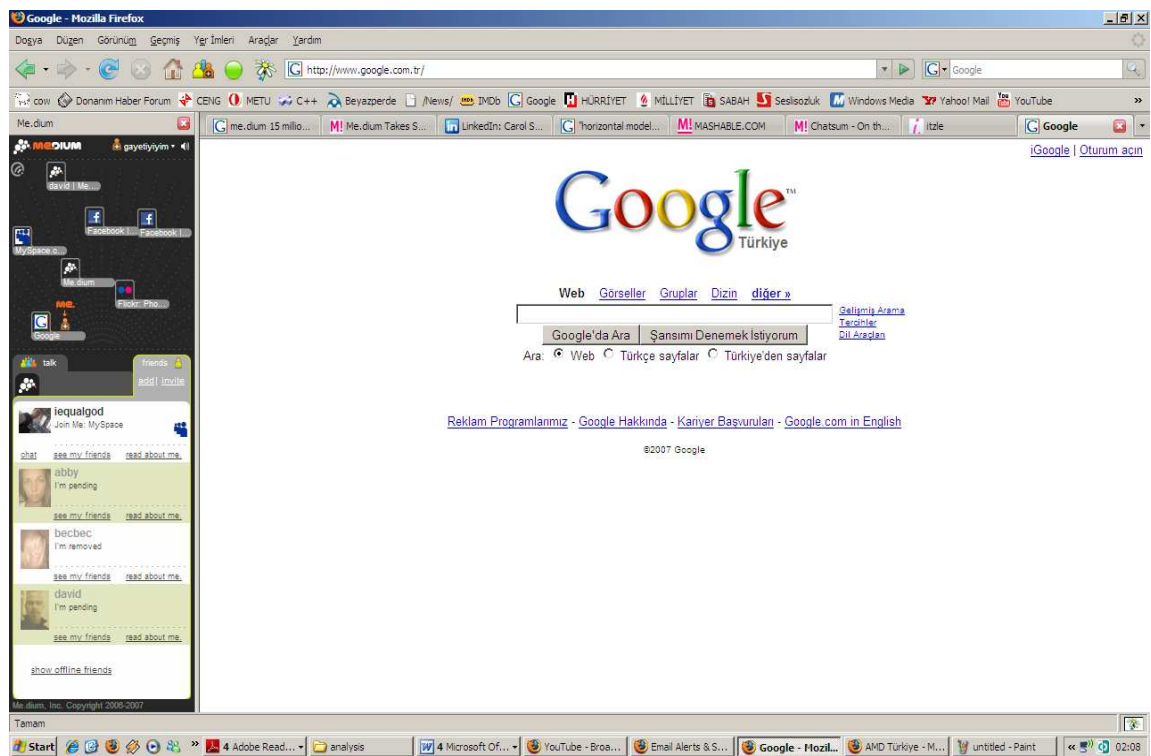
- Figure 2: Dai.sy, at the left side of the browser -

Me.dium

Me.dium is also a browser plug-in that enables XMPP-based social browsing. A user can interact with other users in real time. Its GUI is most developed one among other extensions. It has a radar display that shows moves of users between web pages. Users are capable of adding

their own photographs to their profile. Users who visit other pages can be observed and you can get in touch with them. Moreover, you do not have privacy problems while visiting web pages. You can be invisible anytime you want. Additionally, Me.dium automatically turns itself off when you visit a web page that require privacy like banking sites or secure connections.

Medium's server resources have been modified to improve the Me.dium matching engine, improve the data storage model and switch to the larger and more powerful Jabber XCP chat server. Me.dium seems to be developed by their creators in future. 15 million \$ was funded to advanced it, lastly.

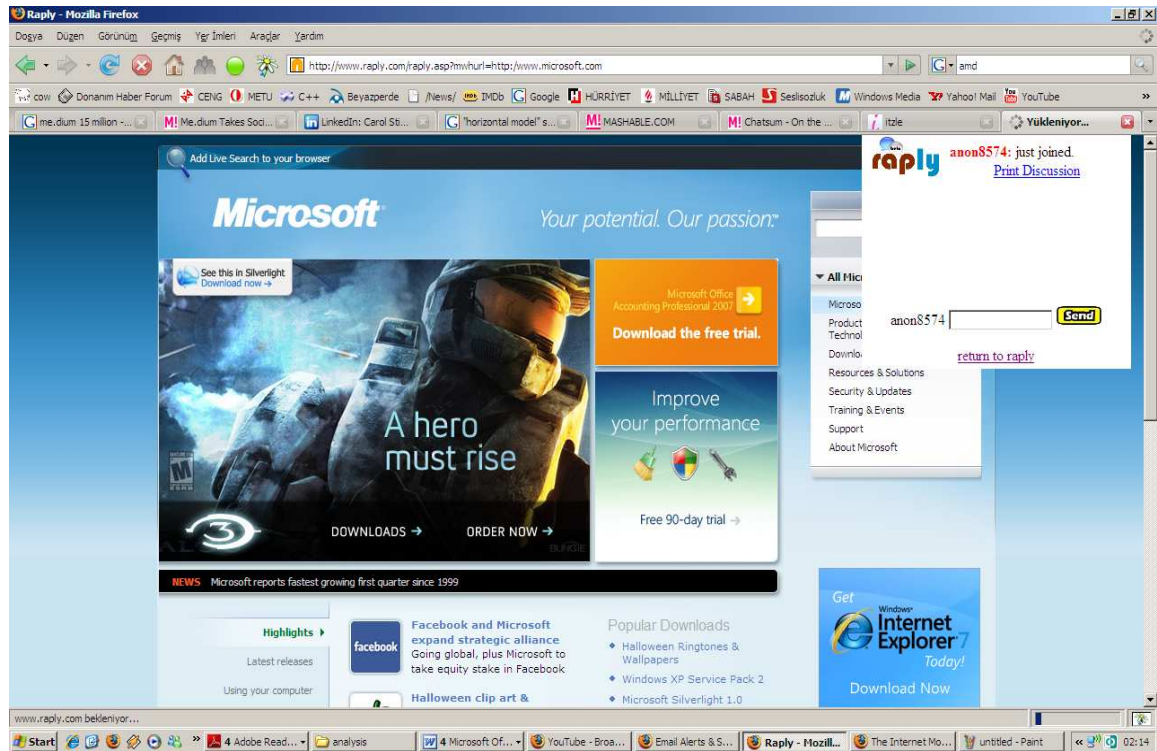


- Figure 3: Me.dium, at the left side of the browser -

Raply

Raply is a Web 2.0 social chat service from Numly Company. It does not require any extensions to web browsers. It gives people opportunity to chat with groups while surfing web pages. Registration is not needed for that application. You just type web address of page you

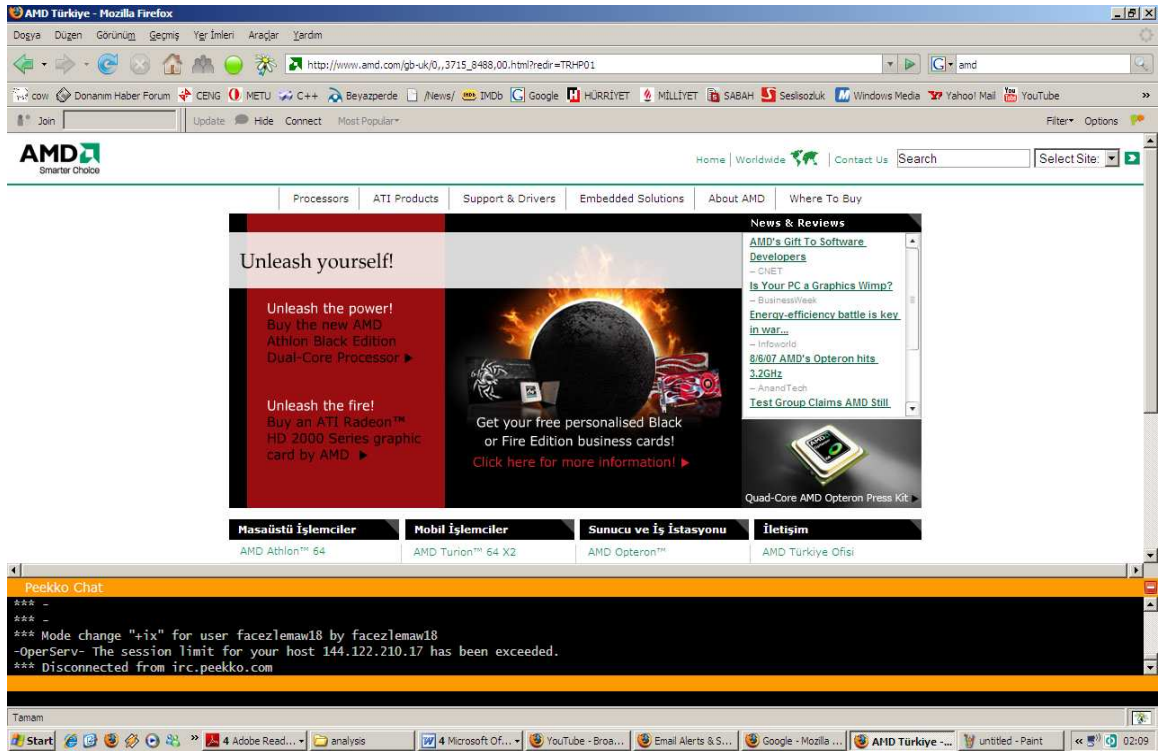
would like to visit on your web browser after “reply.com/”. Another way to use Rply is visit “reply.com” and use textbox at there for web page you want to scan.



- Figure 4: raply, at the upper right side of the browser -

Peekko Chat

Peekko Chat is another Firefox extension that makes every web page the place where people can communicate. It adds a toolbar to Firefox that shows how many people are on that web page currently. This extension let people who are at the same webpage instantly communicate together.



- Figure 5: Peekko Chat, at the bottom side of the browser -

3.3. CONTACTS

With David Mandell

David Mandell is one of the founders of Me.dium and vice president of Marketing. While we are exploring Me.dium we have had a chance to talk with him like an interview. He said that the idea is to bring people that are doing similar things together. However, it can serve different purposes. There is a map showing web pages that are visited at that time mostly by users. Therefore, it can direct a user to those pages, and help to see new things on internet. Moreover, if a user visits a web page that he does not know anything about earlier, he can want assistance from online users. Also, people can get rid of copying and pasting URLs when they want to show something to people with whom they are talking to. Only following his/her friend at Me.dium will be enough for this purpose.

We also ask about possible developments of Me.dium. He replied that there may be new features that let group activities. That will make Me.dium more preferable to users who want to socialize on the internet.

With Potential Users

Among all Firefox extensions, only Me.dium has active user groups. Therefore, we had the opportunity to talk with them and get their opinion about Me.dium. Also, we benefit from the forums of Me.dium web page to learn opinions of users.

After talking with potential users of our product, we saw that users have some requests related with system functionalities. Firstly users want to be capable of ignoring a user that he/she doesn't want to chat with by putting him/her to his/her blacklist. Moreover, chat history can be added as a new feature. It will be very good if a user could see a few lines of the last conversation that he has made. It will help people to remember what the last issue they are talking about was.

4. PROCESS

4.1. TEAM STRUCTURE

Due to the most of the project topics that we are going to work on being new for all AQUT members, we have no permanent leader. This situation leads us to choose Democratic Decentralized (DD) as a team structure. Each group member will work on specific area that is going to be defined as a group. Each member will also be responsible for his part but of course other group members may help him to solve the problem if he needs. In our weekly meetings, each AQUT member will explain on what he works during the week and suggest solutions for the problems that he encountered if any.

In order to obtain equal work participation in AQUT and clarify the roles of group members, we have prepared a role table as in Figure 6.

	M. Ali Özkeskin	Mustafa Çöçelli	Uğur Irmak	Şevket Dokgöz
Project Manager	✓			
Contact Person	✓			✓
Project Recorder		✓	✓	
Archive Keeper			✓	✓
Schedule Coordinator		✓		

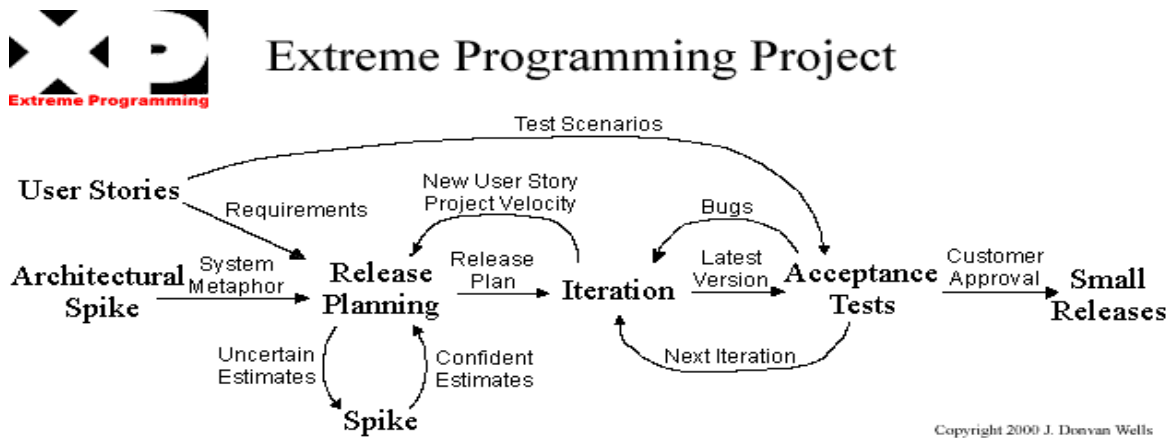
- Figure 6: Role Table -

The followings are our team rules to make our team life long:

- ✚ The tasks will be distributed equally among team members
- ✚ Team members have to come to the meetings on time
- ✚ All members are responsible for backup of their works
- ✚ Priority task of a team member is to complete his task on time unless he has a reasonable excuse
- ✚ Rules can be revised and some changes can be done on them if needed

4.2. PROCESS MODEL

Our plan while developing our project is to advance step by step and these steps will be defined on any stage of the process according to potential users' needs, our assistant's demand and our partners' requirement. Since requirements have a tendency to change during transition between steps, we decided to choose Extreme Programming (XP) as our process method. Therefore; we will start with simple work and improve our project on that work considering related people's requirements. In this methodology, since feedback from users, partners and assistant is very important for the development of the project, communication between these people should be strong. When we considered the need of this method, we believe that our productivity will increase by using XP. Extreme programming map can be seen in Figure 7 below:



- Figure 7: Extreme Programming Map -

4.3. MAJOR CONSTRAINTS

4.3.1. HARDWARE CONSTRAINTS

As we are going to supply instant messaging environment to the users, we need many of them within the context of our project especially in testing phase. This means we will need many computers in that phase.

4.3.2. SOFTWARE CONSTRAINTS

Since we have many tools that is widely used, we think we can easily solve our problems related with software by making little search.

4.3.3. TIME CONSTRAINTS

Due to strictly set deadlines for this project it becomes undeniable to make an excellent scheduling analysis. Also our heavy course load is another factor that takes us under pressure. But in order to complete this project, we know we should meet the strict deadlines of the project.

4.3.4. EXPERIENCE CONSTRAINTS

Since we haven't been in a web application project before, the tools that we are going to use are all new for us. Application servers like APACHE and TOMCAT, instant messaging servers like JABBER and OPENFIRE and AJAX are the applications that we newly started to get information about. Moreover we have created any plug-in for neither Internet Explorer nor Firefox before.

5. REQUIREMENTS

5.1. FUNCTIONAL REQUIREMENTS

5.1.1. SYSTEM FUNCTIONALITY

In our system a user can be a visitor or a registered user. And we have an admin. A visitor will only have the chance to use our system after registration so a visitor can only register to the system. Registration will be done through the website of the system. After registration process a user will be capable of these functionalities:

- ✖ User can send and receive messages instantly
- ✖ User can make friends and can be visible by everyone or only by his/her friends depending on his/her wishes
- ✖ User can see which webpage the other users are visiting right now
- ✖ User can leave notes on a webpage for himself/herself, for his/her friends or for everyone
- ✖ User can tag any paragraph or any image at the webpage to share them with other users and to help question answering agent
- ✖ User can vote for the WebPages
- ✖ User can see the popular WebPages
- ✖ User can see other users that are in the range if those users are visible to all
- ✖ User can learn information about the currently visited webpage from question answering agent
- ✖ User can evaluate the notes that are leaved by other users. Also user can view the evaluation of notes and tags
- ✖ User can flag the tags and the notes as inappropriate
- ✖ User can add other users in his/her blacklist that he/she doesn't want to see or talk

Admin must also login to the system and after the administration is approved, he/she will be capable of these functionalities:

- ✖ Admin will have all user functionalities that is defined above
- ✖ Admin can view all users
- ✖ Admin can ban the users that have complaints from other users. Moreover admin can remove this user from the system permanently or release this user's ban
- ✖ Admin can delete the notes or tags that are flagged as inappropriate

5.1.2. CORE FUNCTIONALITY

Since our project aims to develop a collaborative messaging environment based on Ajax and XMPP open platforms in the form of a Web 2.0 application, we have JABBER on the server side and AJAX on the client side.

❖ JABBER/XMPP SERVER

There are many instant messaging protocols such as MSNP, OSCAR and YMSG that is used in windows live messenger, aim, icq or yahoo messenger. The way used in XMPP server is combining these many disparate protocols inside the instant messaging (IM) server application. This approach moves the communication task to the server application so clients need not to know anything about other IM protocols.

❖ AJAX

On the client side AJAX is used to speed up the processes on the client side. The WebPages that users are currently visiting are always going to be seen without any need to refresh. Moreover when IM is being done with AJAX, XMLHttpRequest object is used in JavaScript to

send and receive commands to and from the server. No refreshing is needed for this web application to work because everything is updated in real-time via JavaScript.

5.2. NON-FUNCTIONAL REQUIREMENTS

5.2.1. USER FRIENDLINESS

The best user interface is one of the main goals of our project XILENT. We are planning to achieve this goal by providing the user the combination of both the best view and ease of use. Since the interface is one of the most important parts of the program for a user, it should be well defined and also can be modified according to user preferences. By this way the user can easily use the system functionalities without losing himself/herself in within the interface.

5.2.2. PORTABILITY

Actually by portability, we mean platform independency. Xilent will work both on Firefox and Internet Explorer. Since our project will be developed by java, the users of our product should have JavaScript support on their browsers. But for the first time we are planning to work on Mozilla Firefox browser, then we will try to adopt it for both browsers.

5.2.3. PERFORMANCE

As in every product, performance is a very important issue in our product. Since we are surrounded with WEB 2.0 applications, a slow program can not be acceptable by the users. At this point AJAX technology will make our job easier. In this way users will be satisfied.

5.3. SYSTEM REQUIREMENTS

5.3.1. HARDWARE REQUIREMENTS

We can analyze our hardware requirements in two groups. First one is at the development side and the other one is at the server side. We have guaranteed to have problem-free servers by the company. On the other hand we have fast enough computers that we are going to use in development phase. Even if there become crashes, we have the capability of supplying our needs. But briefly, our hardware requirements are:

- Pentium IV or equivalent AMD processor
- At least 512 MB ram
- At least 40 GB hard disk
- Internet connection

5.3.2. SOFTWARE REQUIREMENTS

We will need many software and tools for developing our project. These will help us at implementation phase, drawings and documentation phase. The followings will be used in order to complete these tasks:

- Apache Tomcat server
- Jabber server, most probably Openfire
- Mozilla Firefox and Internet Explorer
- Java Development Software Kit and Java Runtime Environment
- Microsoft Project
- Smart Draw
- Microsoft Office and Adobe Professional
- Development environment for JavaScript
- Dreamweaver

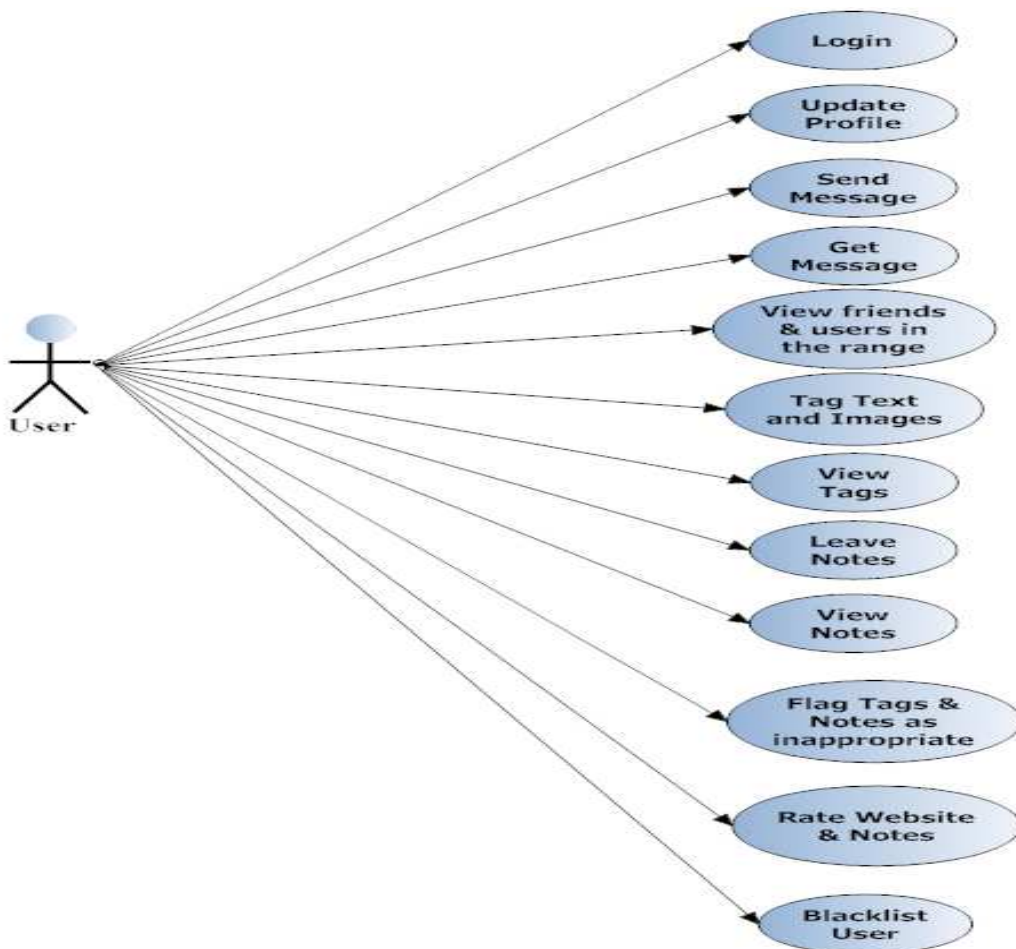
6. USAGE SCENARIO

6.1. USER PROFILES

Since our project will be a Web 2.0 application, we have visitors, users and administrators as usual. The visitors can register to be a user and benefit from system functionalities. On the other hand administrators are also users but have extra features.

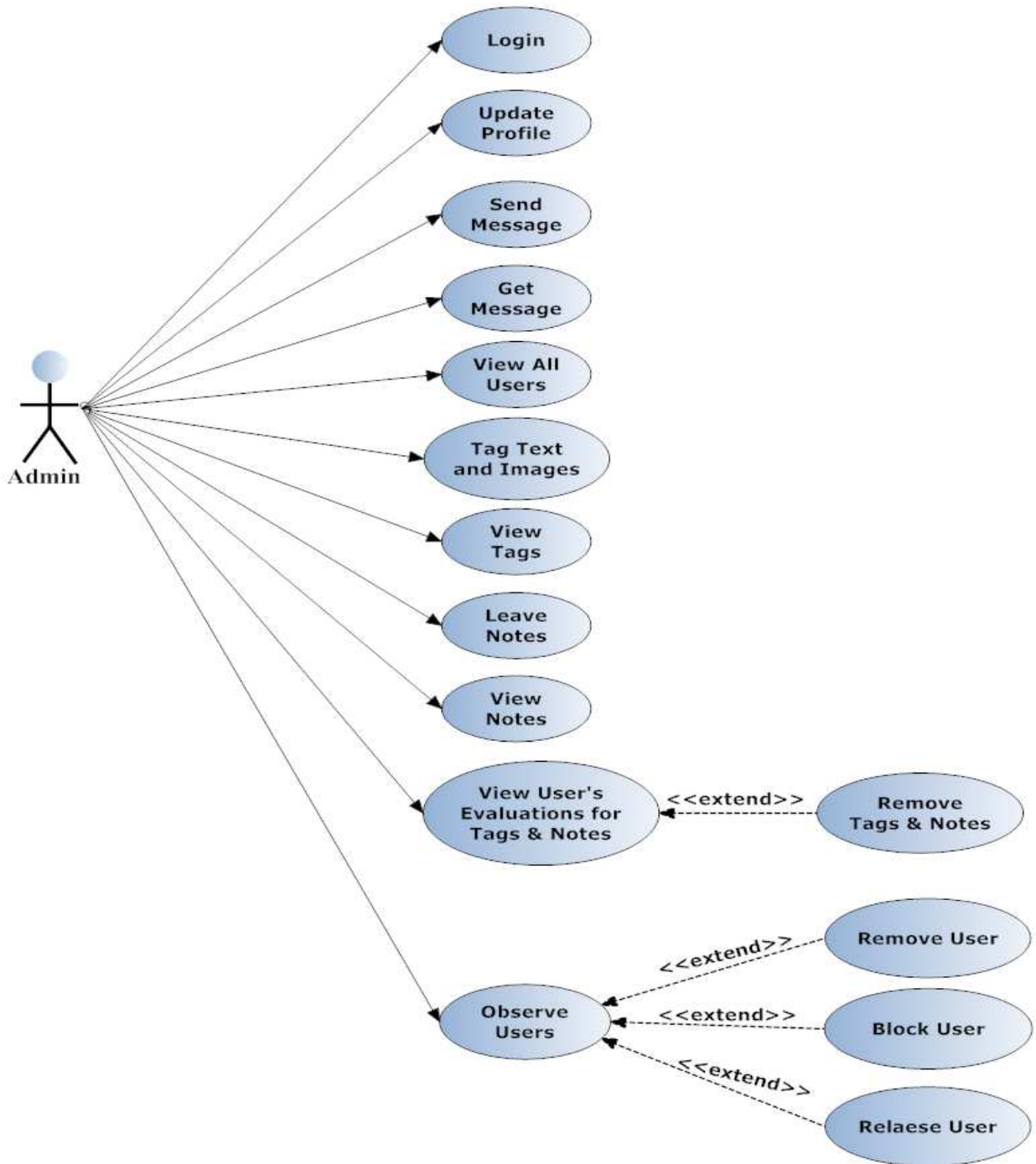
6.2. USE CASE DIAGRAMS

6.2.1. USER USE CASE DIAGRAM



- Figure 8: user use case diagram -

6.2.2. ADMIN USE CASE DIAGRAM



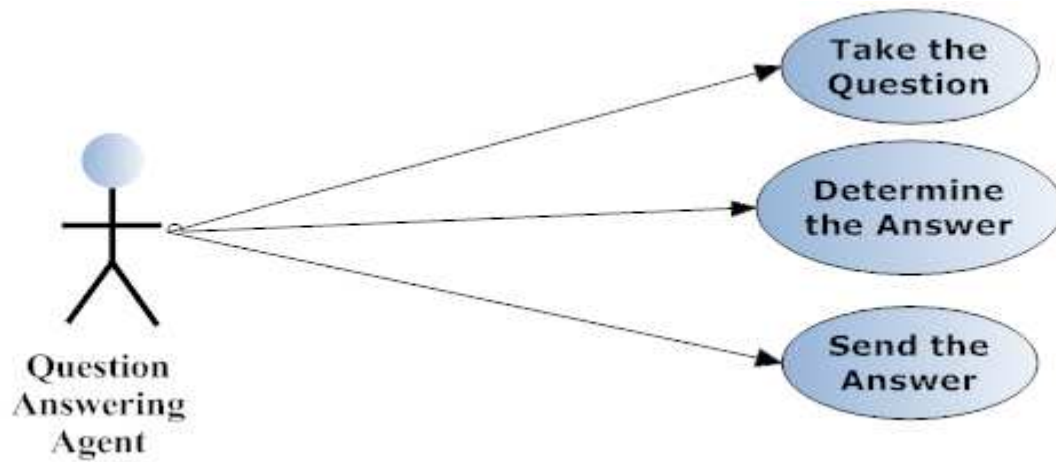
- Figure 9: admin use case diagram -

6.2.3. QA ASSISTANT USE CASE DIAGRAM



- Figure 10: qa assistant use case diagram -

6.2.4. QA USE CASE DIAGRAM

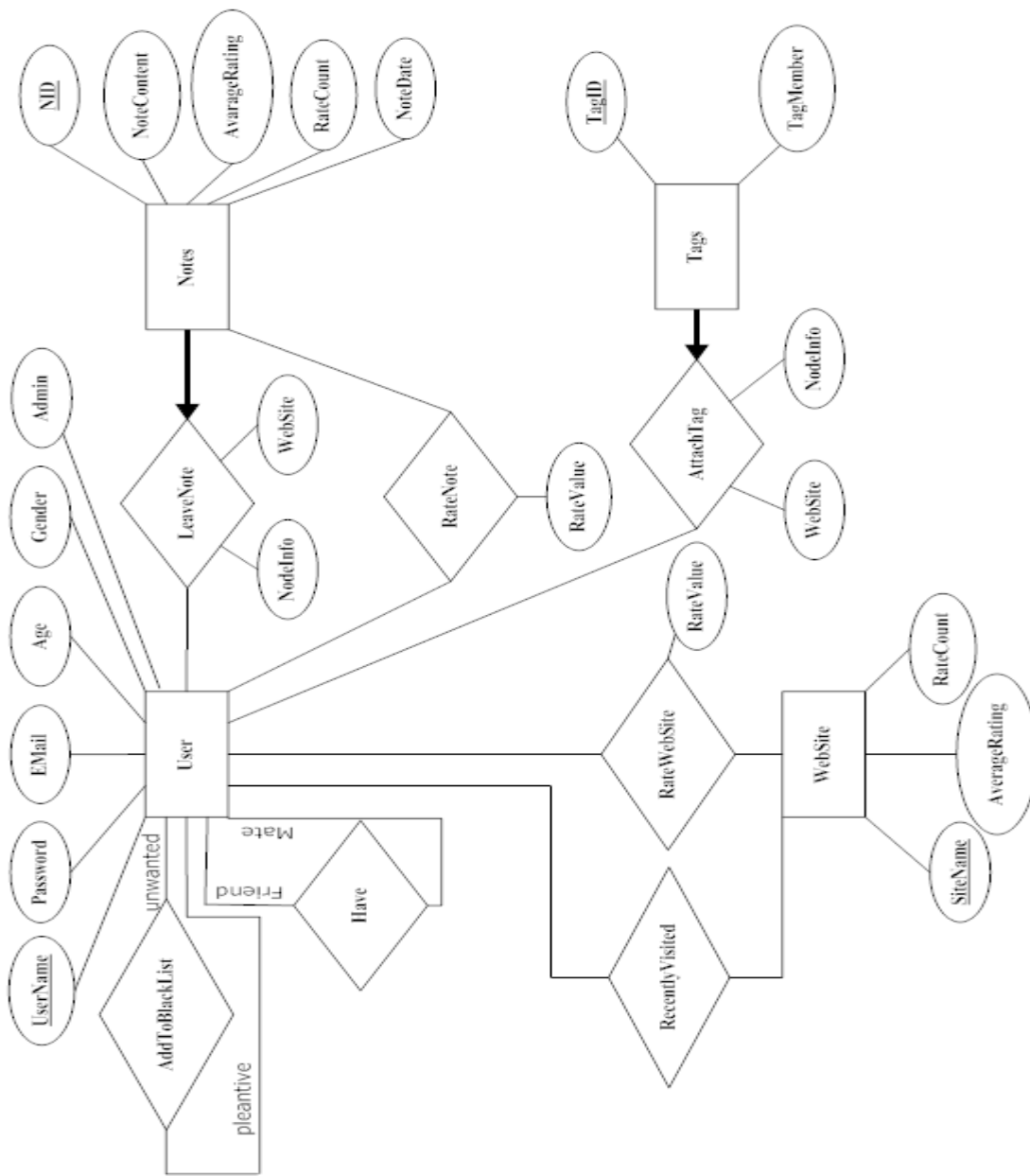


- Figure 11: qa use case diagram -

7. MODELING

7.1. DATA MODELING

7.1.1. ENTITY RELATIONSHIP DIAGRAM



- Figure 12: er diagram -

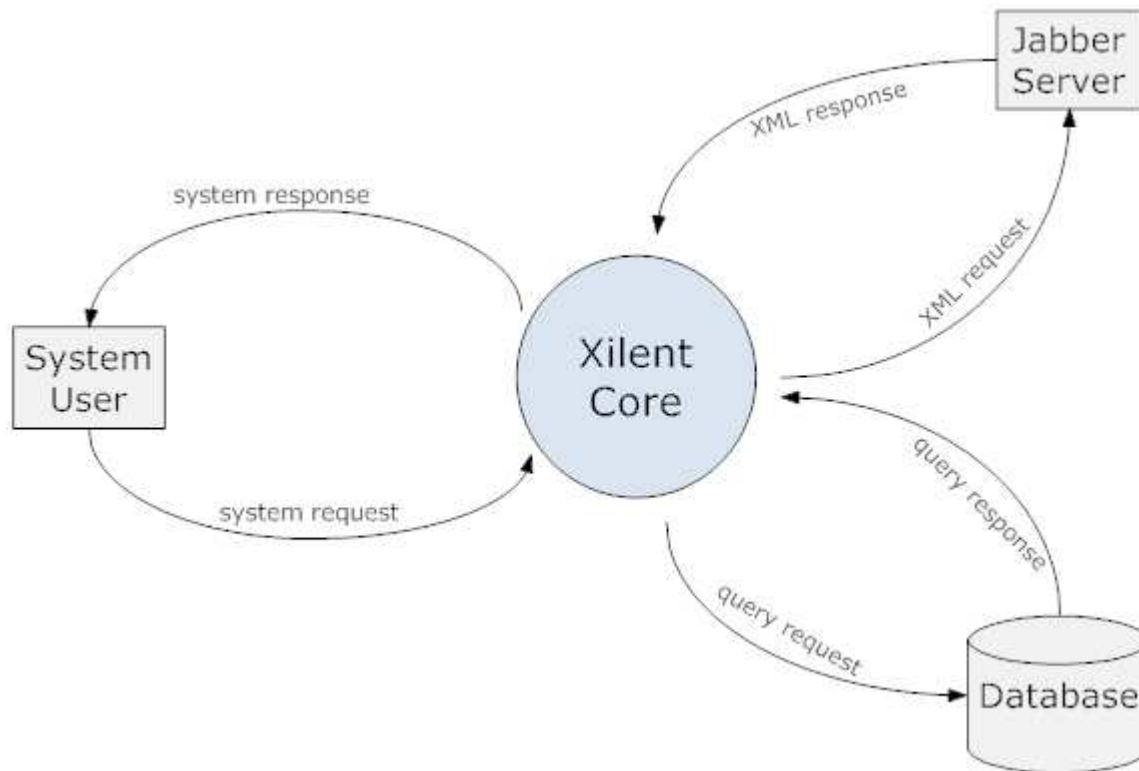
7.1.2. ER EXPLANATION

The database of the system holds information about users, notes left by users, attached tags and ratings of WebPages that are determined by users. There are four entities to show infrastructure of the database. *User* entity keeps information about users like user name, password of this user. There is an admin attribute in *User* entity that shows whether user is admin or not. A user can have friends. In other words, there is a *Have* relation from *User* entity to *User* entity. Also, a user can block message of another user by adding him to his black list. There is an *AddToBlackList* relation to handle this situation. It is also from *User* entity to *User* entity. Moreover, a user can leave notes to WebPages and other users can rate these notes. *Notes* entity, *LeaveNote* relation between *User* and *Notes* entities and *RateNote* relation between user and *Notes* entities are dealt with these actions. Beside these, a user can attach tags to any part of a web page. *Tags* entity and *AttachTag* relation between *User* and *Tags* entities keep information about attached tags. Lastly, to keep rating of WebPages that are rated by users, there are *WebSite* entity and *RateWebSite* relation between *User* and *Website* entities, to keep the WebPages that is visited by user during his/her last session *RecentlyVisited* relation will be used.

We have also another database that is called as *Backup Database*. Backup Database, backups the information from the database of the system periodically and recovers the system if needed.

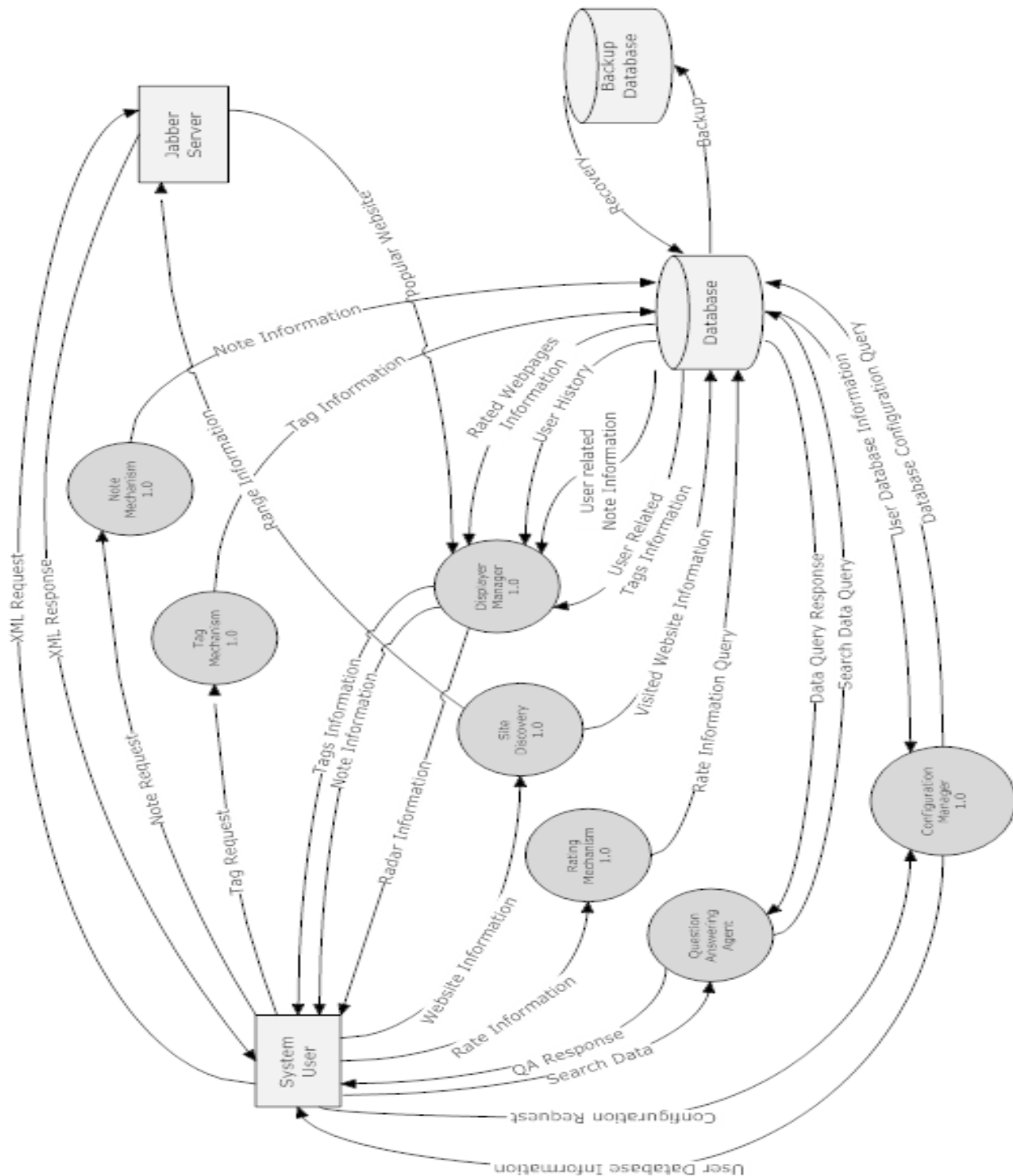
7.2. FUNCTIONAL MODELING

7.2.1. DATA FLOW DIAGRAM (LEVEL 0)



- Figure 13: dfd (level 0) -

7.2.2. DATA FLOW DIAGRAM (LEVEL 1)

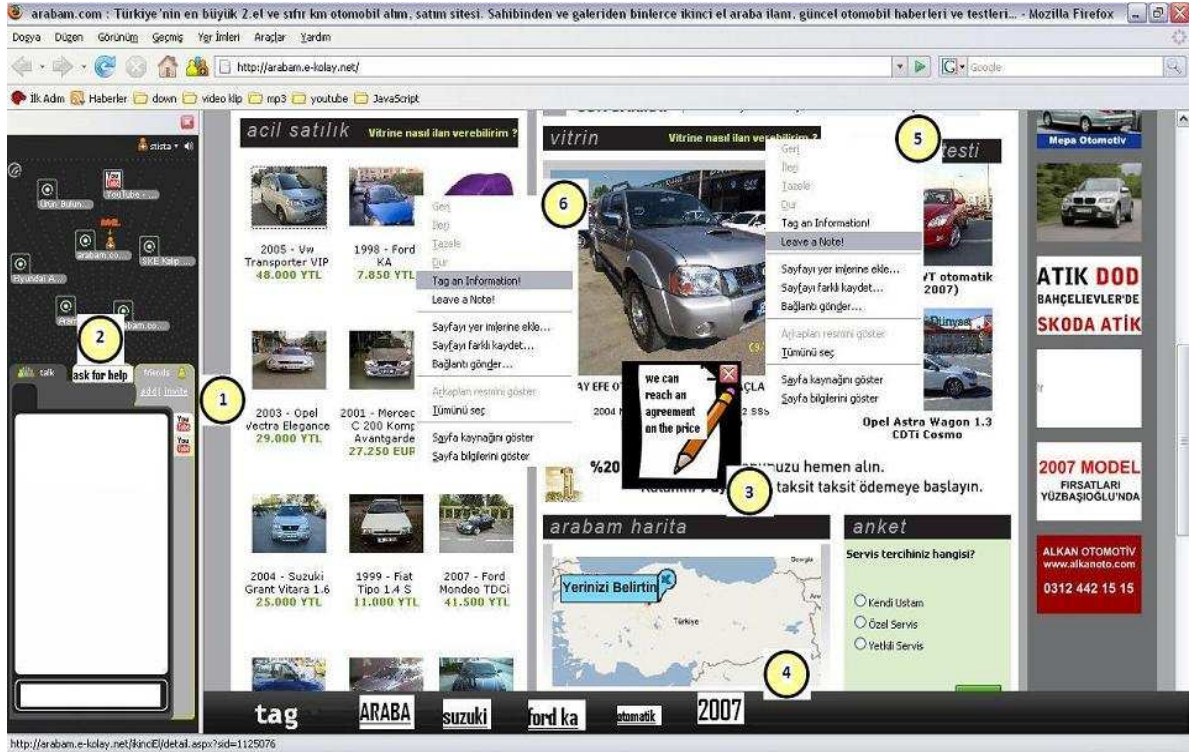


- Figure 14: dfd (level 1) -

7.2.3. DFD (LEVEL 1) EXPLANATION

Data flow diagram in Figure 14, simply explains the working principle of our core system XILENT. First of all, the visitor becomes a user by registering the system. After registering, user sends XML Requests to Jabber Server such as sending and receiving message. Subsequently Jabber Server forwards XML response to the user. User also can leave notes on a web page or can tag any kind of information on any web page. User's note requests will be handled by Note Mechanism, and note information will be sent to the database via this Note Mechanism. Tag requests will be handled in the same way but this time by Tag Mechanism. User also can update his/her profile or delete his/her notes that were previously leaved on a web page. These tasks will be handled by Configuration Manager. Users any configuration request will be sent to the database by Configuration Manager and the response will be taken by Configuration Manager to transmit it to the user. XILENT will also provide user to rate the WebPages to form the most favorite WebPages list. This issue will be handled by Rating Mechanism. This mechanism is used to take the rate information from the user and send this to the database. Our system will enable the messaging environment for the users who are visiting the WebPages that are in the same range. To achieve this goal, we have a Site Discovery tool that can learn which web page are currently visited by the user and send this Website information to both the database and Jabber Server. While Database uses this information to form the user's web page history, Jabber Server uses this information to determine which users are in the range of another. In XILENT to determine what the users see, we have Display Manager. Jabber Server sends currently popular WebPages information to Display Manager. On the other hand Database sends previously visited WebPages information, information about tags and notes whether they exist on the web page, which is currently visited by user and finally web page rating to Display Manager. And Display Manager shows this information to the user. In addition to these, we have Question Answering Agent in our project. This agent searches the webpage and also the database, which may have any tags related with the question, for an answer, when a user asks for help from the Question Answering Agent. Finally, in case of a crash, we have a Backup Database that can recover the original database.

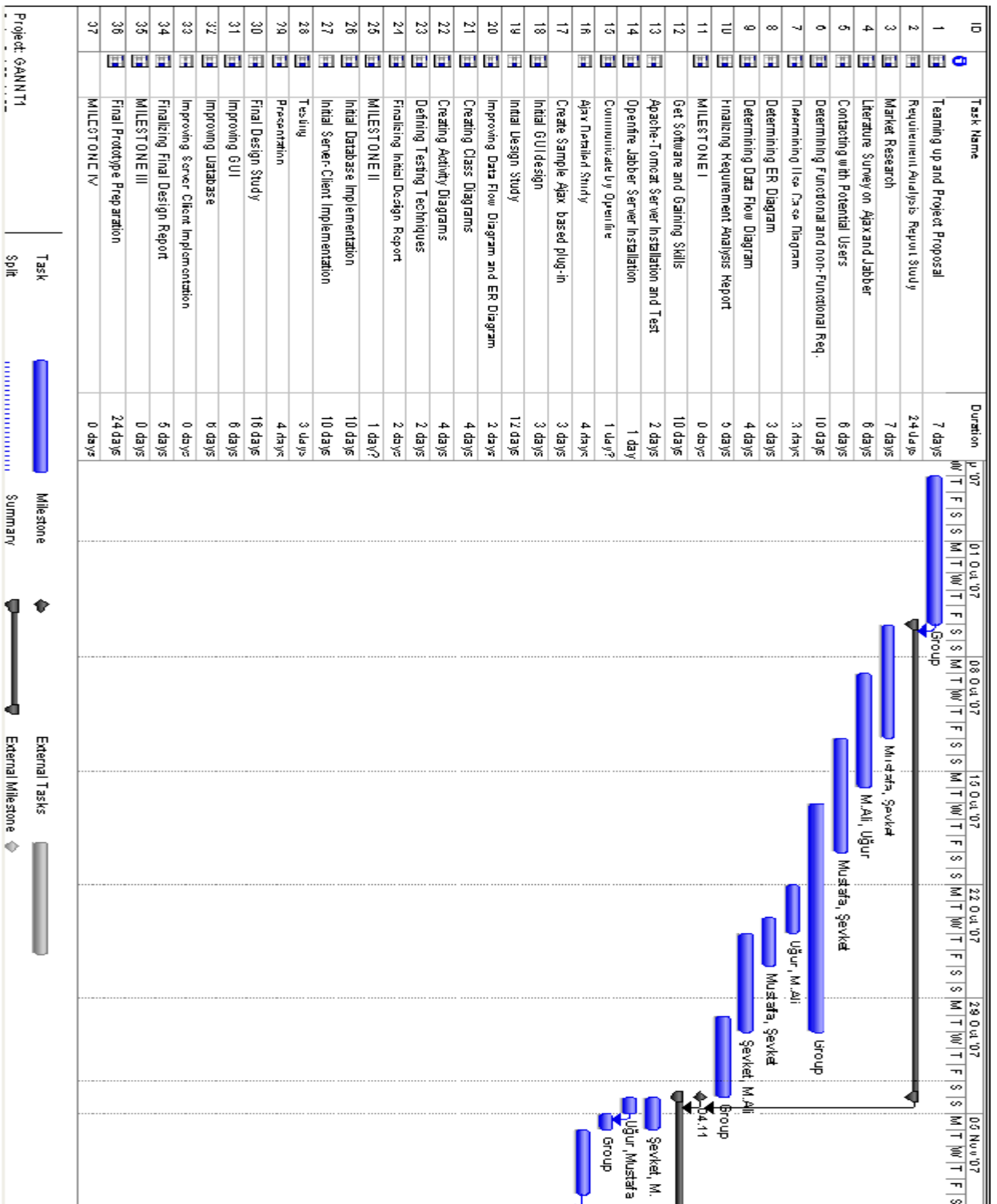
8. SIMPLE PROTOTYPE



- Figure 15: simple prototype -

This is a very simple prototype of our project. Just for the explanation, let us think the user is at arabam.com currently. Our instant messaging platform will be opened at the left side of the browser and push the web page to the right a little bit as it is marked as number 1. If you need any information about the webpage, you can ask that to our question answering agent as illustrated as number 2. User also can see the notes that are leaved on this web page as we can see from the label 3. If the user doesn't want to see this note, he/she can close it. Tags at the web page will be located at the bottom like in number 4. By just clicking on them, the user will be directed to the related place. And finally user can leave a note on the web page or tag any information at that page by right clicking the place that he/she wants and selecting "Tag an Information!" or "Leave a Note!" as it is marked as number 5 and 6 respectively.

9. GANTT CHART



- Figure 16: gantt chart part 1-

10. REFERENCES

Jabber away with instant messaging, Add open source, XML-based instant messaging to your Java applications by Jason Kitchen, JavaWorld.com, 07/26/02

Source page: <http://www.javaworld.com/javaworld/jw-07-2002/jw-0726-im.html?page=1>

AJAX Basics by Sang Shin, Java Technology Architect, Sun Microsystems, Inc. sang.shin@sun.com

Source page: <http://www.javapassion.com/ajax/AJAXBasics.pdf>

Very Dynamic Web Interfaces by Drew McLellan February 09, 2005

Source page: <http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>

XMPP: The Protocol for Open, Extensible Instant Messaging by JIVE SOFTWARE

Source page: http://www.igniterealtime.org/about/jive_xmpp_wp.pdf

Web 2.0 user interface technologies, 30 Jan 2007

Sam Thompson, Solution Architect, IBM Emerging Internet Technologies, IBM

Source page: <http://www.ibm.com/developerworks/web/library/wa-web2ui.html>

Why Microsoft and Google are cleaning up with AJAX?

What new features will AJAX bring to Web applications? , by Jan Stafford, Editor, 10 Nov 2005

Source page: <http://searchenterprise.linux.techtarget.com/>