# TABLE OF CONTENTS

1 INTRODUCTION .................................................................................................................. 2

1 PROJECT TITLE .................................................................................................................... 2

   1.2 PROBLEM DEFINITION & SCOPE................................................................................. 2
      1.2.1 PROBLEM DEFINITION......................................................................................... 2
      1.2.2 PROJECT SCOPE ................................................................................................. 2

2 PROJECT REQUIREMENTS .................................................................................................... 4

   2.1 SYSTEM REQUIREMENTS ............................................................................................ 4
      2.1.1 HARDWARE REQUIREMENTS .............................................................................. 4
      2.1.2 SOFTWARE REQUIREMENTS .............................................................................. 5

   2.2 USER REQUIREMENTS ............................................................................................... 6
      2.2.1 USE CASE DIAGRAM ........................................................................................ 6
      2.2.2 FUNCTIONAL REQUIREMENTS ......................................................................... 7
      2.2.3 NON-FUNCTIONAL REQUIREMENTS .................................................................. 13

3 MODELING ........................................................................................................................ 14

   3.1 FUNCTIONAL MODELING :: DATA FLOW DIAGRAMS ................................................ 14
      3.1.1 LEVEL0 DFD ........................................................................................................ 14
      3.1.2 LEVEL1 DFD ........................................................................................................ 15
      3.1.3 LEVEL2 DFD ........................................................................................................ 16

   3.2 BEHAVIORAL MODELING::STATE TRANSITION DIAGRAM .................................. 32

   3.3 DATA MODEL ............................................................................................................. 33
      3.3.1 DATA OBJECTS ................................................................................................... 33
      3.3.2 RELATIONS ........................................................................................................ 39
      3.3.3 E/R DIAGRAMS ................................................................................................. 40

4 PROCESS ............................................................................................................................ 46

   4.1 TEAM STRUCTURE ...................................................................................................... 46

   4.2 PROCESS MODEL ....................................................................................................... 46

   4.3 ESTIMATION .............................................................................................................. 46

   4.4 GANTT CHART ......................................................................................................... 49

5 LITERATURE SURVEY ..................................................................................................... 51

   5.1 SIMILAR APPLICATIONS .......................................................................................... 51
      5.2.1 QUNU .................................................................................................................. 51
      5.2.2 NOWNOW .......................................................................................................... 53
      5.2.3 FIXYA .................................................................................................................. 54

   5.2 QUESTIONNAIRE ...................................................................................................... 55

6 CONCLUSION .................................................................................................................... 59

7 REFERENCES ....................................................................................................................... 60
1 INTRODUCTION

Internet has become very essential in our daily life. We use Internet for getting information on various subjects, sharing our videos and photos with friends and for finding people with similar interests. New trend in internet technology is Web 2.0 applications. In the old days, Internet had one-way information traffic. Internet sites were sending information but users could not contribute their knowledge or resources at all. Web 2.0 changed the face of internet with a genius approach and let the users to contribute to internet sites. Some famous Web 2.0 applications are wikipedia, youtube and facebook. These internet sites serve an environment to users and users add content to internet sites.

Ajax is a technology used in Web 2.0 application which does not need to reload the entire page, it only reloads required parts. This feature increases the effectiveness of internet sites by shortening loading time.

1.1 PROJECT TITLE

Our company name is Tombeki++. We wanted to name our project in harmony with our company name so we decided to name our project Marpuch.

1.2 PROBLEM DEFINITION & SCOPE

1.2.1 PROBLEM DEFINITION

Internet has become information highway of the new world; however it is very easy to get lost while searching for information. You can find hordes of information on a subject however most of what you find will be totally useless or unrelated. Even if you find a useful site with related information, you will need to skim between many pages of tutorials to find information on a certain paragraph. This process wastes quite a time and also it will be quite boring. Moreover many users do not like/need generic answers. Users want solution for their very own, specific problems and they want it fast.

Furthermore, sometimes users need trivial information such as “the day with the longest night time” or “price of a certain Mercedes model”. Building a Web 2.0 site is the perfect answer for this problem.

1.2.2 PROJECT SCOPE

Marpuch will be an Ajax-based Web 2.0 application which answers users’ questions much faster than traditional ways.

Marpuch will provide users a question box. When a user asks a question in English language, Marpuch will analyze the question via natural language processing
techniques. Then Marpuch will search its database for an appropriate answer. If the question is trivial, such as "what is boiling point of water?" then answer will be shown. Otherwise we will show similar questions and their answers, we will talk about this later.

If no similar question is found or user is not satisfied with those questions then Marpuch will search online Marpuch volunteers who are expert in that field. This is the Web 2.0 side of Marpuch. As we described earlier, Web 2.0 applications take advantage of users' knowledge, Marpuch is no exception. We will allow users to 'tag' themselves as 'Expert' in a subject. As I said earlier, when a user cannot find a satisfactory answer in a certain field, Marpuch will search for volunteers who are expert in that subject. User will be able to talk with these experts via browser. When a user wants to talk with an expert, expert will be notified by his/her instant messenger software. This process will be done via Marpuch's jabber server. We think that user-expert inputs are extremely important so after an expert-user chat; experts will be able to mark some of user's sentences as 'the real question' and also will be able to mark his/her sentences as 'the real answer' of that question. These questions will be saved to Marpuch's database. On the other hand, we will ask registered users to rate expert with respect to our expert criteria. Moreover registered users will be able to comment on volunteers. Anyone with a browser will be able to use Marpuch without registering to system however we will only take feedback from registered users to prevent abuse of rating system. We will have a ladder system for volunteers (a.k.a. experts). User ratings, chat count and amount of online time will affect the total rating and class of volunteer.

We believe that if an expert chats with many users at the same time, his/her efficiency will be reduced so Marpuch will not let users to ask question to volunteers who are already in a chat with another user. However we will show volunteers who are in chat with another user in expert search as busy. These are the core elements of Marpuch.

If no expert is online when a question is asked, then registered users will be able to pin that question. When an expert on that subject becomes online, he/she will be notified about ‘pinned question’. These are the core features of Marpuch.

Some other features of Marpuch are:

- Ajax-based keyword suggestion
- Experts will be able to add ‘tags’
- Registered users may add some experts as favorite experts
- ‘Online Topics’ will be shown on main page
- If Marpuch matches experts’ country with user’s country, then it will be pointed out. Matching will be done via IP address.
2 PROJECT REQUIREMENTS

2.1 SYSTEM REQUIREMENTS

2.1.1 HARDWARE REQUIREMENTS

Our hardware requirements are in 3 groups. First one is our hardware requirements for developing process; second one is hardware requirements for server side, last one is requirements for client side.

These are the hardware requirements for developing process:

- Intel Pentium4 3.2 GHz processor
- 1 GB Ram
- 5 GB HDD Space
- Internet connection

These are hardware requirements for server side:

- Intel Pentium Core 2 Duo 2.24 GHz processor
- 2 GB Ram
- 40 GB HDD Space
- Fast Internet connection

These are hardware requirements for client side (non mobile):

- Celeron 1.6 GHz processor
- 256 MB Ram
- Internet connection

These are hardware requirements for client side (mobile):

- Any mobile device which can run jabber client
- Internet connection
2.1.2 SOFTWARE REQUIREMENTS

Our software requirements are in 3 groups. First one is our software requirements for developing process; second one is software requirements for server side, last one is software requirements client side.

These are the software requirements for developing process:

- Windows XP / Linux
- Apache & Tomcat
- JDeveloper IDE
- JRE
- Smartdraw
- Oracle
- Gimp
- OpenOffice
- Firefox & IE(6.0,7.0) & Opera & Safari
- JDK
- Microsoft Project

These are the software requirements for server side:

- Apache & Tomcat
- Openfire
- JRE
- Linux

These are the software requirements for client side:

- Any browser
2.2 USER REQUIREMENTS

2.2.1 USE CASE DIAGRAM
2.2.2 FUNCTIONAL REQUIREMENTS

1. Registering

Registration is not mandatory to ask questions however it is a must for pinning questions, rating experts and adding experts to ‘favorite experts’ list. Experts need to register as well. Users need to fill these fields:

- Username: Nick name of user. This is a mandatory field. We will control uniqueness of lowercase of nickname, which is username can include uppercase characters but our system is not case sensitive when processing nicknames. A validater controller will check this box and warn if user chooses a nickname which is already registered or nickname textbox is empty.
- Password: Password of user. This is a mandatory field. This field is disabled until a unique nickname is entered. Passwords will be shown as stars. It will be encoded when inserted to database. Password must be at least 6 character and at most 12 characters. Moreover it must contain a numeric and alpha numeric character.
- Name: Real name of user. This is another mandatory field. It is also disabled until password and username are entered correctly.
- Surname: Real surname of user. This is another mandatory field. It is also disabled until password, username and Name fields are entered correctly.
- Location: Which country is user from? It consists of 3 different parts.
  - Country: This is a dropdown menu which contains all the countries of the world.
  - City: This is also a dropdown menu which is filled after the user selects his country.
  - Zip code: This textbox is enabled when user chooses city part. This part is not mandatory

- E-mail address: This textbox is enabled after mandatory fields in previous part are filled correctly. This is a mandatory field. We will use this e-mail address when a user’s pinned question is answered and also more importantly it will send a new password and his old username (nickname) to users and experts who forget his/her password. Validater checker will check whether e-mail address is syntaxly valid or not.
• Education Level: This is a mandatory field. It is for users and experts to know each other’s knowledge level. It is enabled after mandatory fields in previous part are filled correctly.

• Department: This is implemented with Ajax hiding facility. So it is visible only if education level is either undergraduation or graduation.

• Become an Expert: This is not a mandatory field. If the users check “I want to be an expert” checkbox then hidden expertise subjects (tags) will be shown in a table in alphabetical order with help of Ajax. Expert candidate can select tags from this table with the help of check boxes.

• Instant Messaging Address: This part is enabled after previous mandatory parts are filled correctly and also “Became an Expert” checkbox is checked. This is a row which consists of three basic html elements. These elements are textbox, label, and dropdown list. Account name is written in textbox, “@” is written in the label. Available instant messaging servers will be in the dropdown menu. User will chose one of them. We cannot allow every instant messaging server because some instant messaging servers do not support jabber. Marpuch’s jabber server will make a friend request to this address. If the volunteer accepts Marpuch server’s friend request, Marpuch will notify him/her that he/she become an expert. From now on expert will use this IM address whenever he/she chats with users.

• Register Button: This button will be enabled only if the user fills all steps correctly. When this button is clicked user will be registered to Marpuch’s database and he/she will be able to login. Moreover he/she will be redirected to Marpuch’s main page as a logged in user.

2. Logging in

Login is not a mandatory process for asking questions to experts however, when a user logins he/she will be able to look for their favorite experts. Moreover users must login in order to pin unanswered questions. Experts do not need to login to answer questions since question-answering process is done via instant messaging application. Users, experts and moderating stuff use same page for logging in.
Users/Experts need to fill these fields:

- **Username**: Username comparisons are case-insensitive. Validate checker warns if username does not exist.
- **Password**: Password comparisons are case-sensitive. Validate checker warns user if password is less than 6 characters or more than 12 characters
- **Login button**: When a user fills both of these fields, this button will be enabled. Login process works this way: Username will be transformed to lowercase. Password will be encrypted. This translated username and encrypted password will be compared with database usernames and passwords which are in the entity named “Membership”. If the login process success user will be redirected to main Marpuch page. Otherwise user will be redirected to login page.
- **Forget Password?**: This is an Ajax based button. When this button is clicked, a hidden textbox and a button will appear. When the user enters his/her email address and click the button, Marpuch’s database will make a control if the email address entered by the user exists. If email address exists then a new password and old username will be sent to this email address. However if the email address does not exist an Ajax based notification will be shown with the text “There is no match with this email address”.

### 2. Asking question

Ask question page is where users enter their question. Marpuch parses this question and search its database for an appropriate answer and similar questions in that subject.

Users need to fill these fields:

- **Question Box**: Users enter their questions in English language to this Ajax based textbox. Questions should be meaningful English sentences so that Marpuch can extract as much detail as possible with the help of Marpuch’s machine learning tool. If user writes only keywords such as “Firefox, c++, Athletics etc.” Marpuch will still be able to offer possible questions-answers and experts. Marpuch will be able to suggest question completion with the help of Ajax.
- **Ask Button**: This button will trigger the Marpuch’s machine learning tool to parse the question, also trigger the Marpuch’s database for searching online experts which are online and relevant to question’s subject. Moreover user will be redirected to the question-answering page.
3. Question answering

This page has mainly three parts. First part will contain answer of the question if Marpuch’s machine learning tool finds an appropriate answer from its database. Second part will consist of the online expert list. There will be boxes in this list. Each box will contain nickname of expert, chat count of expert, education level of expert and also some extra information about expert. Third part is a list of relevant questions on that subject.

Users need to fill these fields:

- Chat buttons: Each of these buttons is inside experts’ information boxes. When this button is clicked a chat request will be send to the selected expert’s instant messaging application via Marpuch’s jabber server. If the expert confirms chat request then Ajax based chat interface will be opened.

- Relevant questions: This is an Ajax based hyperlink. When it is clicked a hidden form will be shown. In this form Marpuch will show question and answer pairs.

4. Chatting

This is an Ajax based interface looks like msn chat interface, which contains chat history, send button, rating box and a textbox for writing a message.

- Textbox: User and expert will write their messages here. Expert can enter specific key words here. These key words are:
  - <tag=“expertise” , level=“number (between 1 to 10)” />
  - <question = “question sentence” , answer = “answer sentence”/>

- Rating box: There are five stars here. Experts’ average rating will be shown by coloring these starts in red. User will be able to rate experts by clicking these stars. An example of this rating system can be seen at “www.youtube.com”.
5. Reading Help

User can take help by reading Marpuch’s help page. There are 6 main help topics in this page. These main topics are:

1. Getting Started: This part includes a glossary and information about many subjects such as: account types, how a question should be asked to get a better result.

2. Becoming an Expert: This part explains how to become an expert in details.

3. The Marpuch Community: This part is all about Marpuch community. This part includes information about:
   a. How to add an expert to favorite experts list.
   b. What does it means adding an expert to favorite experts list?
   c. How do I invite my friends to join Marpuch?

4. Account and Policies: This part includes information about:
   a. I forget my username and password. How do I log back to Marpuch?
   b. How do I close my account?
   c. How do I report abuse?
   d. How do I rate experts?
   e. Why was my account suspended?
   f. How do I block another user?
   g. My account was hacked. What do I do?
   h. How do I change my password?

5. Advanced: This part includes information about:
   a. Will my mobile device work with Marpuch Mobile?
   b. How do I add a new expertise subject?
c. I am expert in an area which is not in suggested expertise list. What do I do?

d. I want to add a new question-answer pair which deducted from a chat with a user. What do I do?

6. Troubleshooting: This part includes information about:
   
a. Solving General Site Issues
   
b. Common Error Messages

6. Pinning a question

Users can pin a question if Marpuch cannot find an appropriate answer and all expert in that area are offline. It is shown as a suggestion in question-answering page.

7. Reporting abuse

Users and experts can report abuse by sending screenshot of abuse to admin@marpuch.com with the topic [ABUSE].

8. Reporting bug

Users and experts can report bugs by sending screenshot of abuse to admin@marpuch.com with the topic [BUG].
2.2.3 NON-FUNCTIONAL REQUIREMENTS

I - Usability

Marpuch will be an easy to use Web 2.0 application. This is an important requirement for us because our main claim is making life easier for people who search information. In each step we will clearly show all choices of user in one page. This will be done by using Ajax’s menu hiding tools.

II – Reliability

We will keep profiles of registered users and experts. Expert profiles are extremely important because without experts Web 2.0 applications are quite weak. Another important issue is chatting. Chat process should be consistent. Dead links are quite annoying for users so we should not have dead links in our system.

Marpuch may be displayed by different browsers. We should tune Marpuch to as much browser as possible.

III - Security

Many websites abuse registration process by selling e-mail address of their users or by not taking sufficient security means. Marpuch’s security should be tight. We wouldn’t want to lose our precious volunteers by letting their e-mail address get stolen.

IV - Portability

Marpuch is a Web 2.0 application so it can be accessed from any platform. We will also tune Marpuch to make it nice looking in every browser.

V - Performance

Performance is an important issue for Marpuch because our aim is to make it faster to get information. Our database should be big but our serve time should also be fast. We use Ajax so Marpuch will not be reloaded every time you make some request.
3  MODELING

3.1 FUNCTIONAL MODELING :: DATA FLOW DIAGRAMS

3.1.1 LEVEL0 DFD
3.1.2 LEVEL1 DFD
3.1.3 LEVEL2 DFD

1.0 AUTHENTICATION

- GUI UNIT
  - User input

- AUTHENTICATION UNIT
  - Input: username, password
  - Output: filtered data

- ENCRYPTION UNIT
  - Input: filtered data
  - Output: encrypted data

- REGISTER UNIT
  - Input: filtered data
  - Output: username, password

- LOGIN UNIT
  - Input: username, password
  - Output: login successful

- KEYBOARD EVENT HANDLER
  - Input: username, password
  - Output: filtered data

- DATABASE
  - Input: filtered data
<table>
<thead>
<tr>
<th>Name</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>LOGIN 1.1</td>
</tr>
<tr>
<td>To</td>
<td>ENCRYPTION UNIT 1.2</td>
</tr>
<tr>
<td>Description</td>
<td>The data taken from user to approve log-in</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>FilteredData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>REGISTER UNIT 1.3</td>
</tr>
<tr>
<td>To</td>
<td>ENCRYPTION UNIT 1.2</td>
</tr>
<tr>
<td>Description</td>
<td>Data is filtered because of preserving abuse</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>EncryptedData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>ENCRYPTION UNIT 1.2</td>
</tr>
<tr>
<td>To</td>
<td>AUTHENTICATION UNIT 1.4</td>
</tr>
<tr>
<td>Description</td>
<td>Some data (ex. password) are used after encrypted to preserve security</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>FilteredInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>AUTHENTICATION UNIT 1.4</td>
</tr>
<tr>
<td>To</td>
<td>DATABASE</td>
</tr>
<tr>
<td>Description</td>
<td>The data contains user info and filtered by previous units</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FilteredUserInfo</td>
<td>Gathered data from database</td>
</tr>
<tr>
<td>DATABASE</td>
<td></td>
</tr>
<tr>
<td>AUTHENTICATION UNIT 1.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApprovalUserLogin</td>
<td>Includes required info about user who has logged-in</td>
</tr>
<tr>
<td>AUTHENTICATION UNIT 1.4</td>
<td></td>
</tr>
<tr>
<td>GUI UNIT 4.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApprovalOfUser</td>
<td>Includes required info about user who has logged-in</td>
</tr>
<tr>
<td>AUTHENTICATION UNIT 1.4</td>
<td></td>
</tr>
<tr>
<td>TRIGGER UNIT 5.3</td>
<td></td>
</tr>
</tbody>
</table>
2.0 INPUT CONTROLLER

- LOGIN
- REGISTER
- PARSER UNIT
- DATABASE
- TRIGGER UNIT
- MOUSE EVENT HANDLER
- MOUSE KEYBOARD SPLITTER UNIT
- KEYBOARD EVENT HANDLER

Flows:
- UserInput
- UserRequest
- MouseInput
- KeyboardInput
- ChatRequest
- ParseData
- UserInput
- Register
- ParseData
- Login
<table>
<thead>
<tr>
<th>Name</th>
<th>MouseAndKeyboardInput</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>USER</td>
</tr>
<tr>
<td>To</td>
<td>MOUSE EVENT HANDLER 2.2</td>
</tr>
<tr>
<td>Description</td>
<td>Total input from user</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>MouseInput</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>MOUSE KEYBOARD SPLITTER UNIT 2.1</td>
</tr>
<tr>
<td>To</td>
<td>MOUSE EVENT HANDLER 2.2</td>
</tr>
<tr>
<td>Description</td>
<td>Splitted mouse input by splitter unit. This data will be splitted again for specific uses.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>KeyboardInput</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>MOUSE KEYBOARD SPLITTER UNIT 2.1</td>
</tr>
<tr>
<td>To</td>
<td>KEYBOARD EVENT HANDLER 2.4</td>
</tr>
<tr>
<td>Description</td>
<td>Splitted keyboard input by splitter unit. This data will be splitted again for specific uses.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>KeyboardInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>MOUSE EVENT HANDLER 2.2</td>
</tr>
<tr>
<td>To</td>
<td>TRIGGER UNIT 2.3</td>
</tr>
<tr>
<td>Description</td>
<td>Data that processed by trigger unit to make triggers.</td>
</tr>
<tr>
<td>Name</td>
<td>KeyboardInfoRequest</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>From</td>
<td>TRIGGER UNIT 2.3</td>
</tr>
<tr>
<td>To</td>
<td>KEYBOARD EVENT HANDLER 2.4</td>
</tr>
<tr>
<td>Description</td>
<td>Data that triggers KEYBOARD EVENT HANDLER and the unit gives true data with the help of this data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>ChatRequestFrom</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>MOUSE EVENT HANDLER 2.2</td>
</tr>
<tr>
<td>To</td>
<td>TRIGGER UNIT 5.3</td>
</tr>
<tr>
<td>Description</td>
<td>Triggers data control unit to send chat request and related info to Jabber server</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>ChatData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>KEYBOARD EVENT HANDLER 2.4</td>
</tr>
<tr>
<td>To</td>
<td>PARSER UNIT 5.1</td>
</tr>
<tr>
<td>Description</td>
<td>Includes raw chat data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>SearchData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>KEYBOARD EVENT HANDLER 2.4</td>
</tr>
<tr>
<td>To</td>
<td>PARSER UNIT 3.1</td>
</tr>
<tr>
<td>Description</td>
<td>Raw search data which submitted by user</td>
</tr>
</tbody>
</table>
### Requirements Analysis Report | TOMBEKI++

<table>
<thead>
<tr>
<th>Name</th>
<th>UsernamePassword</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>KEYBOARD EVENT HANDLER 2.4</td>
</tr>
<tr>
<td>To</td>
<td>LOGIN 1.1</td>
</tr>
<tr>
<td>Description</td>
<td>Includes username and password</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>PersonalData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>KEYBOARD EVENT HANDLER 2.4</td>
</tr>
<tr>
<td>To</td>
<td>REGISTER UNIT 1.3</td>
</tr>
<tr>
<td>Description</td>
<td>Includes data that taken from register</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>RateOfExpert</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>MOUSE EVENT HANDLER 2.2</td>
</tr>
<tr>
<td>To</td>
<td>DATABASE</td>
</tr>
<tr>
<td>Description</td>
<td>Rate data includes expert info and rating</td>
</tr>
</tbody>
</table>
3.0 MACHINE LEARNING UNIT
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParsedData</td>
<td>Users search data parsed and passed to analyzer to reach result</td>
</tr>
<tr>
<td>ErrorFreeExtensibleInfo</td>
<td>Data that will add to database to extent our database and improve our answer ability</td>
</tr>
<tr>
<td>EncodedDataToMachineLearn</td>
<td>Data is encoded to another form that our application can use.</td>
</tr>
<tr>
<td>OldAnswerAndOnlineExpert</td>
<td>Result of queries to display search results</td>
</tr>
<tr>
<td>Name</td>
<td>AddNewQuestionAnswer</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>From</td>
<td>RECORDER UNIT 3.4</td>
</tr>
<tr>
<td>To</td>
<td>DATABASE</td>
</tr>
<tr>
<td>Description</td>
<td>Data which Filtered and encoded to application form, that submitted by expert. This data is for improving best results capability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>RecorderQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>RECORDER UNIT 3.4</td>
</tr>
<tr>
<td>To</td>
<td>DATABASE</td>
</tr>
<tr>
<td>Description</td>
<td>Query from recorder these are mostly insert queries</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>AnalyzerQuery</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>ANALYZER UNIT 3.3</td>
</tr>
<tr>
<td>To</td>
<td>DATABASE</td>
</tr>
<tr>
<td>Description</td>
<td>Query from Analyzer, these are mostly select queries</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>DataToDisplay</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>ANALYZER UNIT 3.3</td>
</tr>
<tr>
<td>To</td>
<td>DECODER UNIT 4.1</td>
</tr>
<tr>
<td>Description</td>
<td>Includes all online experts data and old question&amp;answers</td>
</tr>
</tbody>
</table>
4.0 DISPLAY UNIT
## Requirements Analysis Report

### TOMBEKI++

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnlineExperts</td>
<td>Includes online expert information that will be displayed to user</td>
</tr>
<tr>
<td>OldAnswers</td>
<td>Includes old answers data that is relevant to current question</td>
</tr>
<tr>
<td>Styles</td>
<td>Includes info about registered users’ visual choices such as font, colours...</td>
</tr>
<tr>
<td>Display</td>
<td>Includes all the data about page that will be shown to user</td>
</tr>
<tr>
<td>LoginUserInfo</td>
<td>Includes required info about user who has logged-in</td>
</tr>
</tbody>
</table>

**From**
- DECODER UNIT 4.1
- GUI UNIT 4.2
- USER

**To**
- DISPLAYER UNIT 4.3
5.0 DISPLAY UNIT
### Name: QuestionAndAnswer

<table>
<thead>
<tr>
<th>From</th>
<th>SPECIAL ENCODER UNIT 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Data that will be added to database to enrich future results. Contains added question and answer</td>
</tr>
</tbody>
</table>

### Name: ExtendTag

<table>
<thead>
<tr>
<th>From</th>
<th>SPECIAL ENCODER UNIT 5.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Data will be added to experts’ tags space. Include tags and expertise levels of expert</td>
</tr>
</tbody>
</table>

### Name: ChatDataFrom

<table>
<thead>
<tr>
<th>From</th>
<th>DATA CONTROLLER UNIT 5.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Data that will be sent to Jabber server. Cleared from special tags</td>
</tr>
</tbody>
</table>

### Name: ChatDataTo

<table>
<thead>
<tr>
<th>From</th>
<th>DATA CONTROLLER UNIT 5.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Chat data that will parsed by parser unit</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>ChatDataToServer</td>
<td>Communication Data</td>
</tr>
<tr>
<td>DATA CONTROLLER UNIT 5.4</td>
<td>To JABBER SERVER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatDataFromServer</td>
<td>Communication Data</td>
</tr>
<tr>
<td>JABBER SERVER</td>
<td>From DATA CONTROLLER UNIT 5.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatRequest</td>
<td>Chat request to Jabber Server. Handled by Jabber Server</td>
</tr>
<tr>
<td>DATA CONTROLLER UNIT 5.4</td>
<td>To JABBER SERVER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatDataToUser</td>
<td>Final Data that will be displayed to User</td>
</tr>
<tr>
<td>PARSER UNIT 5.1</td>
<td>From USER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerData</td>
<td>Trigger Data Controller to</td>
</tr>
<tr>
<td>TRIGGER UNIT 5.3</td>
<td>From DATA CONTROLLER UNIT 5.4</td>
</tr>
<tr>
<td>DATA CONTROLLER UNIT 5.4</td>
<td>To DATA CONTROLLER UNIT 5.4</td>
</tr>
</tbody>
</table>
### Requirements Analysis Report

<table>
<thead>
<tr>
<th>Name</th>
<th>SpecialData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>SPECIAL ENCODER UNIT 5.2</td>
</tr>
<tr>
<td>To</td>
<td>PARSER UNIT 3.1</td>
</tr>
<tr>
<td>Description</td>
<td>Data that will be added to database to enrich our database. These data are given by experts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>ExtensibleData</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>SPECIAL ENCODER UNIT 5.2</td>
</tr>
<tr>
<td>To</td>
<td>PARSER UNIT 3.1</td>
</tr>
<tr>
<td>Description</td>
<td>Includes data which will extend our database and enrich search results</td>
</tr>
</tbody>
</table>
3.2 BEHAVIORAL MODELING::STATE TRANSITION DIAGRAM
3.3 DATA MODEL

3.3.1 DATA OBJECTS

This section describes our tables and their properties which will be used in Marpuch project. These tables are made under the consideration of getting rid of unnecessary attributes and normalization factors.

Users

This table is always needed in order to be known in server side. For this case we attempted to create as less attributes as we could. Here UserName which is Nick in the view of user is primary key and foreign key to other tables. This is must in almost all web services dealing with these type account systems. In order to avoid user login with his/her nickname to the system wrongly by making mistake in large/small letter cases we created LoweredUserName attribute. This attribute also does not permit system registering same nick with same character string. Since registration is not compulsory for users visiting our website we created attribute IsAnonimous. This attribute also determines visitors which are registered or not. LastActivityDate is supposed to record the users’ last activity within in website. By this, after some time user is logged out automatically by the system. In addition to all these we have attributes like Name, SurName, BirthDay, Phone, MobilePhone, Sex describing users’ personal characteristics.

Briefly attributes for table Users are:

- UserId
- UserName
- LoweredUserName
- IsAnonimous
- LastActivityDate
- Name
- SurName
- BirthDay
- Country
- Experience
- Job
- Sex
Membership

This table consists of attributes that are related with the security problems. UserId is the primary key coming from table Users. This key determines to which user attributes belongs that are in table Membership. PasswordEncoded is a string that user should type with his/her username to login. This string is determined by the user in registration phase and kept in the database, where it is controlled single sided. PasswordFormat keeps information of password format which is encrypted. Email is an attribute that is used when user forgets password or there is a need in sending notifications to him/her. LoweredEmail is used if the user types wrong email in aspect of big/small character. Afterwards user is informed or warned about mistake. PasswordQuestion is the question if the user wants to recover his/her password. This question is filled by the user in the registration phase. PasswordAnswer is the answer to the question that is filled by the user. IsLockedOut concerns with whether user has been locked out or not for any reason. LastLoginDate keep information about how long the user is active. In order to get rid of mess in database, users which are not active more than 2 years are deleted from the system automatically. FailedPasswordAttemptCount is an integer that is assigned to zero when user logs in. In other case, if user attempts to log in with wrong password twenty times continuously, then the system recognize this system as it is attacked. Afterwards related user account is hibernated. FailedPasswordAttemptWindowStart supports the previous explained attribute. Comment is an attribute that keep the notes of abuse. In case of repetition user is locked.

Briefly attributes for table Membership are:

- UserId
- Password
- PasswordFormat
- PasswordSalt
- Email
- LoweredEmail
- PasswordQuestion
- PasswordAnswer
- IsLockedOut
- CreateDate
- LastLoginDate
- LastPasswordChangedDate
- LastLoggedOutDate
- FailedPasswordAttemptCount
- FailedPasswordAttemptWindowStart
- Comment
Profile

This table is concerned with users’ profile settings. UserId is kept for primary and foreign key. PreferredCulture supply requested language information of user to the server, to give priority in finding expert speaking same language. MainWindowColor and FontColor as seen from their names are to personal flavor.

Briefly attributes for table Profile are:

- UserId
- PreferredCulture
- MainWindowColor
- FontColor

QuestionAnswerTuples

TupleId is primary key and also foreign key to question tags table. Question is the key question that has been asked by user and marked by expert. Answer is the key answer for this question suggested and also marked by the expert. UserId is the ExpertId which marks the question and answer.

Briefly attributes for table QuestionAnswerTuples are:

- TupleId
- Question
- Answer
- UserId
**PinnedQuestion**

Question is primary key. UserId is supply the information who asked the question. GroupId is supply the information to the server that question is about what issue. QuestionAskedDate and LastAnsweredDate keep dates when question is asked and answered.

Briefly attributes for table PinedQuestion are:

- **Question**
- **UserId**
- **GroupId**
- **QuestionAskedDate**
- **QuestionAnsweredDate**

**Tags**

TagId is primary and foreign key. TagName is tags names typed by experts.

Briefly attributes for table Tags are:

- **TagId**
- **TagName**

**ExpertRatings**

UserId is kept as a primary key. Since users can be experts, this facility only belongs to users who are experts. Rating is average rating that has been voted by the users after getting answers from experts. Rating count is for determining how many times related expert was voted.

Briefly attributes for table ExpertRatings are:

- **UserId**
- **Rating**
- **RatingCount**
Roles

Since in our application requires control and maintenance we must specify some roles. These roles are admin, moderator and simple user. These roles determine information that a user can see or settings or another features of application. Also new added tags must be control by moderators or admin in order to prevent abuses.

Briefly attributes for table Roles are:

- RoleId
- RoleName
- Description

Group

In our application there are too many areas that we can interest. Some tags are related each other actually. This relation must be used when we search an expert or giving old questions’ answers. Clustering the tags make easy our work and give more precise results.

Briefly attributes for table Group are:

- GroupId
- GroupName

UserDefaults

In a Web 2.0 application some properties of page must be determined by user. We do not want to display a page with colors or fonts that users do not like. In this table we store some information about users’ preferences about layout of the pages that application will create.

Briefly attributes for table UserDefaults are:

- UserId
- Font
- MainColor
**CommentsToExperts**

Users rates experts in order to answers that they gave. Besides rating mechanism we collect comments to experts from users. Every registered user writes comment about experts.

Briefly attributes for table CommentsTo Experts are:
- **UserId**
- **Comments**
3.3.2 RELATIONS

UsersInRoles

We have users and roles. Some users are more than users. They can be in moderator or admin or post defined roles. We store this information in UsersInRoles table. In our application security of this table is really important.

Briefly attributes for table UsersInRoles are:
- UserId
- RoleId
- Since

TagsOfExperts

In our application every expert has one or more experienced topic. We define these topics as tags in database. Also every expert must define expertise level of that topic. We store all this information in our database. This relation is too important for us to know experts’ abilities.

Briefly attributes for table TagsOfExperts are:
- UserId
- TagId
- ExpertLevel

TagsandGroups

We cluster the tags and with clustered tags we give more precise results to users. Tags may belong to more than one tag-group. Some tags are related to some groups but not too much so we must store info about relevance level of tags and groups.

Briefly attributes for table TagsandGroups are:
- TagId
- GroupId
- RelevanceLevel
3.3.3 E/R DIAGRAMS

Users

- UserName
- LoweredUserName
- IsAnonymous
- LastActivityDate
- Name
- Sex
- Experience
- UserId
- Surname
- Birthday
- Country
- Job

Membership

- UserId
- Password
- PasswordFormat
- PasswordSalt
- Email
- LoweredEmail
- PasswordAnswer
- PasswordQuestion
- IsLockedOut
- LastPasswordChangedDate
- LastLoggedOutDate
- FailedPasswordAttemptWindowStart
- FailedPasswordAttemptCount
- LastLoginDate
- CreateDate
- Comment
Requirements Analysis Report | TOMBEKI++

Profile

UserId

PreferredCulture

MainWindowColor

FontColor

PROFILE

QuestionAnswerTuples

TupleId

Question

Answer

UserId

QUESTIONANSWERTUPLES

PinnedQuestions

UserId

GroupId

QuestionAskedDate

QuestionAnsweredDate

Question

PINEDQUESTION
UsersInRoles

TagsOfExperts
TagsAndGroups

- TagId
- TagName
- GroupId
- GroupName

TAGS

GROUP

RelevanceLevel

TAGSANDGROUPS
4 PROCESS

4.1 TEAM STRUCTURE

We have decided that Controlled Decentralized (CD) team structure fits us. Our team has a leader and our leader assigns sub topics to members. We have a collaborative decision mechanism. Communication between members is really important. We have meetings in strict hours and every time we are in touch.

4.2 PROCESS MODEL

We will follow a step by step program. We will do analysis, initial design, more detailed design, prototype, implementation, testing & debugging and maintenance. The most suitable process model is “Waterfall Model” for us. After we finished one step we won’t turn back to previous steps. So, all the features of a step must be well-understood and applied.

4.3 ESTIMATION

**FP BASED ESTIMATION**

<table>
<thead>
<tr>
<th>Measurement parameter</th>
<th>count</th>
<th>simple</th>
<th>average</th>
<th>complex</th>
<th>Functional Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of user inputs</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>Number of user outputs</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Number of user inquires</td>
<td>1</td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Number of files</td>
<td>2</td>
<td></td>
<td>10</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Number of external interfaces</td>
<td>6</td>
<td>5</td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73</td>
</tr>
</tbody>
</table>
User Inputs: Register, Login, Rate Expert.

User Outputs: Question-Answer page, Login.

User Inquires: Ask question

Files: Main database, Server log

External Interfaces: Register, Login, Question-Answer, Help, Report Abuse, Relevant Questions Pages

<table>
<thead>
<tr>
<th>Questions</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the system require reliable backup and recovery?</td>
<td>4</td>
</tr>
<tr>
<td>Are data communications required?</td>
<td>5</td>
</tr>
<tr>
<td>Are there distributed processing functions?</td>
<td>1</td>
</tr>
<tr>
<td>Is performance critical?</td>
<td>2</td>
</tr>
<tr>
<td>Will the system run in an existing, heavily utilized operational environment?</td>
<td>0</td>
</tr>
<tr>
<td>Does the system require on-line data entry?</td>
<td>4</td>
</tr>
<tr>
<td>Does the on-line data entry require the input transaction to be built over multiple screens or operations?</td>
<td>2</td>
</tr>
<tr>
<td>Are the master files updated on-line?</td>
<td>0</td>
</tr>
<tr>
<td>Are the inputs, outputs, files, or inquires complex?</td>
<td>5</td>
</tr>
<tr>
<td>Is the internal processing complex?</td>
<td>5</td>
</tr>
<tr>
<td>Is the code designed to be reusable?</td>
<td>2</td>
</tr>
<tr>
<td>Are conversion and installation included in the design?</td>
<td>0</td>
</tr>
<tr>
<td>Is the system designed for multiple installations in different organizations?</td>
<td>0</td>
</tr>
</tbody>
</table>
Is the application designed to facilitate change and ease of use by the user?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong></td>
<td>35</td>
</tr>
</tbody>
</table>

\[\sum (F_i) = 35\]
\[FP = 73^*[0.65 + 0.01*35] = 73\]

Now if we have the following assumptions:

1. The organization produces on an average 12 FP per month

2. And labor costs is $10000.00 per month

We can now make an estimate of the software:

At $10000.00 per month with 12 FP per month, then the cost per FP is $(10000/12) which is approximately $833.30.

For 73 FP the project will cost $(833.30x 73) . $60,833.
5 LITERATURE SURVEY

5.1 SIMILAR APPLICATIONS

There are a few applications like Marpuch. Some of them are Qunu, Nownow and Fixya. Qunu is the major application in this area. We will discuss some info about these applications.

5.1.1 QUNU

Qunu is the most professional application in this area. Some of the key features of Qunu below

- Clear interface
- Easy to use
- Able to search experts
- Jabber-based
- Free to access
- Instant messaging
- Multi-browser (limited) support
- Well-prepared help section and wiki
- Anonymous help

These are the most important features of Qunu. To get help you must follow these steps

1. Enter our search query into the search bar
2. We will pick an expert that we believe that he/she helps us.

3. After choosing corresponding expert we will click “CHAT” button.

4. The site must be connecting you to expert using with chat window.

Finally we reach the expert.
5.1.2 NOWNOW

NowNow is a question & answer system like Marpuch. NowNow is a service that mobile users can use to find answers to any question via mobile email. When you post a question to NowNow via email (ask@nownow.com), the workers will surf the web and try to find the answer for you.

The disadvantage of this system is not being of instant. But NowNow has a big supplier behind it which is Amazon.com.

NowNow is not a free system. If you want to ask questions to system you must be a registered user. Registration is free but for every question they take money. They have workers so the system cannot be free.

But since the system cannot answer questions immediately and it is not a free service NowNow is not a strong opponent of Marpuch.

To get help with NowNow is not difficult. You ask the question to a simple textbox and submit. The system answers question as soon as possible. The answer will be sent to your mail address.

There is a screenshot from homepage of NowNow below.
5.1.3 FIXYA

Fixya provides free tech support and technical help for gadgets, electronic equipment and consumer products. Fixya’s technical experts advise on fixing problems and provide instructions on proper usage of products either by chat or message posting. Fixya stores manuals and troubleshooting guides for over half a million products. Fixya’s tech support community will provide a quick solution for your "how to" problem.

Fixya is not an instant system. It provides answers from asked questions. It analyzes the question and finds most suitable topics from the database and shows to the user. It does not require registered user, you can easily use system anonymously. It is a free system.

One of the disadvantage of the system is system is only for computer or computer related hardware. It does not support any kind of question.

Below, there is a screenshot from the site http://www.fixya.com/
5.2 QUESTIONNAIRE

Since our major aim is to meet user needs, we must ask users to learn about their needs. Questionnaire is the one of the best ways to learn user needs. We performed a questionnaire to possible users.

✔ Egitim Duzeyiniz?
Bilgiye ulaşmak için en çok hangi yöntem kullanıyorsunuz?

- Ansiklopedi
- Arkadaş/Oğretmen
- Internet

Günlük hayatta bilgiye ulaşmadaki öncelikli amacınız nedir?

- Akademik Çalışma
- Merak
- Kısisel Gelişim
- Problemlerin Çözülmesi
**Internet ortamında hangi kaynak sizin için daha çekicidir?**

- Konunun Uzmanından Eşzamanlı: 18
- Dokumantasyonlar: 4
- Google: 10
- Forumlar: 8
- Mail Listeleri: 2

**Sorularınıza cevap ararken karşınıza bir insan olması (muhtemelen sizinle aynı dili konuşan) sizin işinizi kolaylaştırır mı?**

- Evet: 35
- Hayır: 5
- Farketmez: 0
Bilgi sahibi oldugunuz konularda diger insanlara yardımda bulunmak ister misiniz?
CONCLUSION

Preparing this requirement report was hard and tiring but we, Marpuch Team, think that it will be very beneficial to us in the coming months. First of all, we have had a solid idea about our project since we write functional requirements and draw use cases in details. Secondly, we will not need to think about who will do what at a certain time because it is already written in Gantt chart. Moreover we made a market research and that gave invaluable ideas about what we can add to Marpuch to make it compete with senior market players such as (Amazon's www.NowNow.com and www.qunu.com). Furthermore we have a deep understanding of architectural design of Marpuch project since we prepared level 2 DFD diagrams.

To summarize, Marpuch team believes that this requirement report will be of great help to our team during this semester.
REFERENCES

- http://qunu.com/
- http://www.nownow.com/nownow/index.jsp
- http://www.fixya.com/
- Software Engineering A Practitioners Approach. Pressman S. Roger