**Weekly Progress Report**

**Group Name:** Qracle

**Erdem KARAHAN**

*Recorder, gatekeeper, optimist*

This week’s task is providing a basic plug-in for the first project prototype. Despite having a basic structure, the plug-in, still have to handle some preferred basic tasks, by using the Openfire API. I made some trials on eclipse, I include the Openfire library, used some methods in the library, but I’ve still some serious problems about deciding the event to trigger my plug-in. I am planning to go over a task about doing some simple activity, triggered by starting up a session. Since this is a user activity, and I don’t have a user to communicate with now, I’m thinking about another option like an admin related event. For now it seems to me possible that I will have the ability to activate my plug-in without fully implementing a user interface, by using an admin based event. For next week, I’m planning to work on my plug-in structure, to enhance its abilities.

**Mine Karakaya**

*Devil’s Advocate, Recorder*

Firstly, for the prototype I read about working with strongly typed DataSet objects. The heads of the topic were creating strongly typed DataSet objects, using them, working with hierarchical data with them, when to use them, design-time and run-time benefits of them. Secondly, I have developed my part in the prototype. Then I read about Mono as we chose Windows as our platform. I figured out that Mono/Linux/GNOME development stack component provides connectivity to databases including MSSQL and Microsoft compatibility stack component espouse ADO.NET. Yet, I could not prepare the expected environment for ADO.NET and MS SQL Server 2005 Express Edition. Thirdly, I read about Yahoo! Protocol, whose communications use TCP over IP communication.

Next Week

Submitting updates to database
I've learned how to use the ReadXML and WriteXML events. A ReadXML event handle has the following prototype:

```c
void XmppCon_OnReadXml(object sender, string xml);
```

Obviously xml parameter is the xml string read from the sockets. Although it seems as the sender parameter is the 'sender' of the message, it is not. It is the sender of the event, which is another object. Sender information of the message takes place in the XML. Normally we don't have to deal with the XML because the AGS-XMPP library triggers an On_Message event when the xml contains a regular message.

Also I inspected On_Presence event which has the following prototype for its handler:

```c
void XmppCon_OnPresence(object sender, Presence pres);
```

Again sender parameter is the object who triggered the event. By the pres parameter we can learn whose presence is changed and what it is changed to:

```c
if(pres.Type == PresenceType.unsubscribe)
```

**Next week:** We are planning to complete our prototype tasks and go over our designs to enhance their functionalities.