GROUP WEEKLY REPORT:

Our development team started to gain technical competence to achieve technical tasks. We learned about general architecture and important classes of ADO.NET. We have not chosen our DBMS system yet but the ADO.NET library provides same functionality for different systems. So our decision of DBMS will just be an issue of changing the connection object. We made an extensive research on possible Jabber libraries for .NET and we will probably use the AGS-XMPP library. Also we began digging openfire features and networking literature.

PERSONAL WEEKLY REPORTS:

Mine KARAKAYA
Devil's Advocate, Recorder

During the week, I studied the ADO.NET Object Model. The ADO.NET Object hierarchy consists of two main parts; Connected Objects (consisting of Connection, Transaction, DataAdapter, Command, Parameter, DataReader) and Disconnected Objects (consisting of DataSet, DataTable, DataRow, DataColumn, Constraint, DataRelation, DataView). I learned the general definitions and relationships between them. For the time being, I am not sure about which one of the two .NET Data Providers; namely the SQL Client .NET Data Provider and the OLE DB .NET Data Provider; I am going to use. However, this will not cause any problems as both implement the same base classes and the code written looks similar regardless of the provider. I also started to learn some features of Visual Studio .NET such as design-time data-access features. I bind a database created in FireBird 1.5 to a DataAdapter created in a DataSet of a sample project in Visual Studio. I want to give information about some of the methods and properties that will be beneficial for the project:

- the Fill method of the DataAdapter Object: Used for fetching data from the database into a DataSet.
- the Update method of the DataAdapter Object: Used for submitting pending changes stored in DataSet to the database.
- the TableMappings property of The DataAdapter Object: Used for tracking which table in the database corresponds to which table in DataSet Object.
This week I looked for available jabber libraries for .NET framework. There was not much option, only Jabber.NET and AGS-XMPP libraries were available. Jabber.NET is a popular and open source library. It is easy to find 3rd party documentation and samples about this library. On the contrary, I found out that the library is no more supported by the development team. I think this is a risk which can cause disaster through the later stages of the project. Our other option, AGS-XMPP is a two license library, one is commercial and the other one is for open source projects. Since our project is open source the second kind of license is proper for our company. AGS-XMPP has its drawbacks, there is almost no sample or documentation provided by a 3rd party. All is provided by the developer company, and to be fair it seems enough.

This working period, I focused on the plug-in issue, instant messaging protocols, jabber client software and jabber server software. I did a brief search about the features that makes "openfire" more powerful than the other server softwares. I also picked up some information about "TLS", "SSL" and "handshaking protocol". I started to use the "eclipse platform" and used it to run a simple plug-in. I also read articles on "simple" and "sip" protocols which are also used for instant messaging.