Assignment A2

***Distributed Objects***

Student: Lorinczi Istvan

Group: 30444

# Requirements

Consider a distributed application called “job-searching service”. The application should keep record of all job offers in the past year mentioning whether the job is already taken or not. A user should be able to post a job offer and search through the job offers either by a date (or an interval) or by categories (e.g. IT, economy, food service, etc.).

When implementing this system using Distributed Objects technologies consider the following constraints:

1. A Remote Object should expose the “post jobs” system functionality, letting the user to add a job to the system.

2. A Remote Object should expose the “search jobs” functionality, which allows the user to search for jobs either by a date or by a category.

3. All job offers must have a job title, company name, deadline, contact details and job specification.

# Tasks

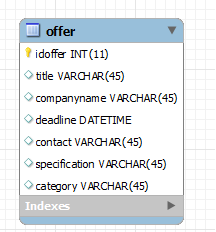
Design, implement and test the distributed application using Java or .NET specific technologies. (RMI or .NET Remoting technologies are compulsory)

# Conceptual Architecture

Since the job offers have a deadline, but after the deadline passes they should still be available to see, a database is used to keep the job offers. The database keeps all the jobs and users can query it to see the jobs and can post other jobs into the database.

A Job has:

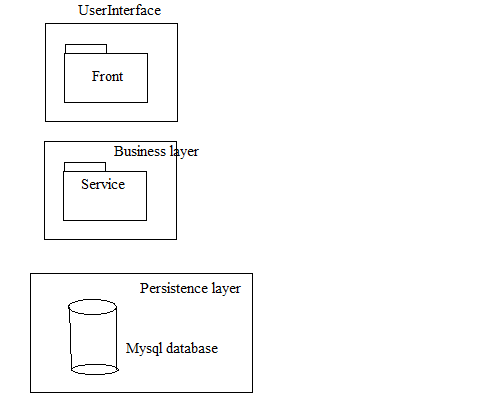
* Title
* Company
* Specifications
* Category
* Deadline
* Contact



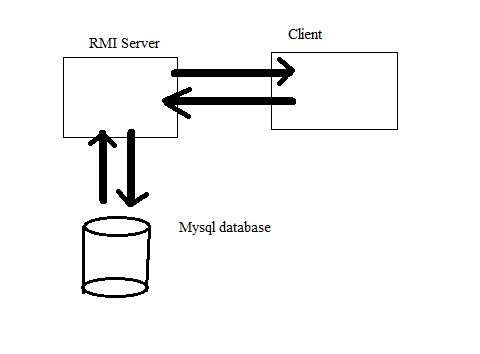
The users have access to 2 types of activities: post a new job offer and check the available job offers. The users can check available job offers using 2 types of filters – by category or by date (interval) – and see the list stored in the database.

When a user wants to perform an action the rmi server must run in order to have access to the database and the operations.

The conceptual diagram of the application:



Deployment diagram:



The application runs on a desktop PC which hosts the server, the database and the client as well. The server posts the services that the client can access. The services are implemented on the server which accesses the database through the services. The client runs on the same device and accesses the server through a socket.

# Unit Testing

The services are tested:

* Posting a job and checking if it is stored in the database
* Searching by title and checking the result
* Searching by company and checking the result
* Searching by date interval and checking the result