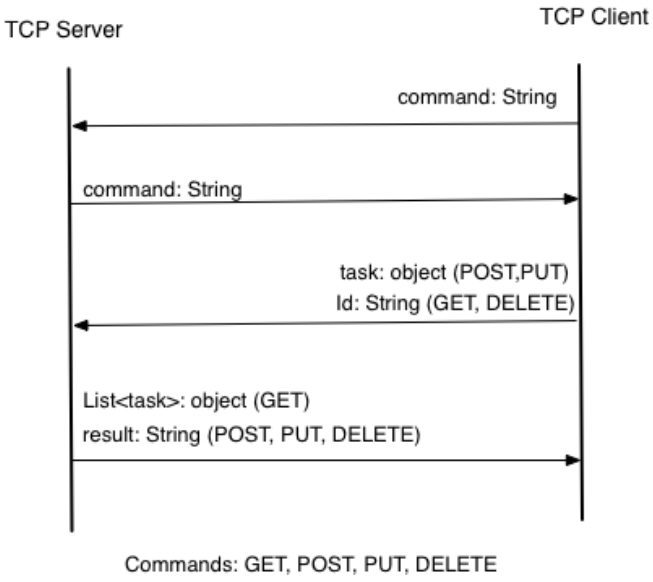
Task Manager

We've developed a simple implementation of a task manager TCP server, which main purpose is to maintain a list of tasks and allow a client to modify this list.
Modification of the list happens through the use of some basic HTTP-like commands: GET, POST, PUT, and DELETE in order to resemble regular HTTP requests.

We've modeled our implementation in accordance with the rules defined in the assignment including the conversional protocol.



Marshalling to XML
In order to use the JAXB java library for XML marshaling we've created 3 classes, each resembling the tags in the original xml-file provided in the assignment.
We've made a Calendar class which resembles the <cal>-tag in the XML. The Calendar class contains collections of instances of Task (<task>) and User (<user>) classes.
Task and User contain the rest of the data in the form of fields, which can be accessed through setters and getters.

The Client
The Client consists of the class TaskManagerTCPClient, which sets up the TCP connection to the server (through a socket) and awaits user input in a never-ending loop. In this loop it checks for the commands mentioned above[1] and executes corresponding methods upon receiving these.
As an example, the POST command will cause the client to send the command to the server and check whether the server returns the same command (in accordance with the assignment specifications). Afterwards the user is inquired to enter the data for a new task, which will ultimately be sent to the server and saved in the XML-file.

The Server
The server consists of the class TaskManagerTCPServer, which utilizes several RequestParser-instantiations (subclasses of Thread). Upon receiving a valid command it initializes the relevant RequestParser through the means of Javas reflection API. It uses a HashMap with IP and Request, to keep track of which client wants to use which command. The server has access to a RequestParser implementation for each of its commands.

Problems
At this point, our implementation does not work as intended. We have problems with server-client-communications. The problems seems to lie within the server. The first client request is received, read and returned, but the server does not handle the following client requests.

[1] We don't check the case of the command, so commands in lowercase will also be accepted.