

Assignment 2 – Distributed File Server

Pierre Thube and Ashley Holmes



Table of Contents

Abstract.....	3
1. Introduction.....	3
2. Design Criteria	3
3. Description of Implementation	4
4. Design Choices.....	4
5. Elaboration of Design	5
Conclusion	7

Abstract

Cloud storage services allow our files to be accessed and shared among multiple users or devices with ease. It has revolutionised the way we interact with our file systems and device storage. It has brought upon a change with regards to the way we even view our computers of this day and age. They used to be our main focus point but now it is the gateway to this ever growing and evolving thing called the “Cloud”.

1. Introduction

The overall goal of this project is to develop a distributed file server using Java RMI. It is a follow on from the previous assessment. Where a group chat application was developed using Java Socket Programming to implement the remote communication. This assessment also helps contrast the difference between programming with Java Sockets and programming with the simplicity of Java RMI. What is meant by this is that the programmer would need to decide whether to use TCP versus UDP over IP.

Explicitly create the sockets with port numbers for communication and then add the InputStream and OutputStream for the send and receive messages. Remote Method Invocation greatly simplifies the creation of distributed applications as it does this all for us and abstracts away from all of what was mentioned. As with the group chat application where a client GUI was created and designed for the user to interact with the system. This process will be used again for the distributed file server.

2. Design Criteria

This document describes the design process and implementation of a File Server. The design criteria are comprised of two major components. That being a GUI interface and the File Server.

The requirements for the File Server include:

- Getting the files from the File Server on a specific directory
- Creating new files on the File Server
- Creating new directories on the File Server
- Deleting files of directories on the File Server

The client GUI:

- This is a GUI with support of a file browser which offers operations to explore the file system on the file server with mechanisms to view files, create files and deletion of these files. Also including renaming operations of the files and directories and methods to add and also delete these directories.

3. Description of Implementation

The first step of this process is creating the graphical user interface or GUI for the application. Using the Netbeans IDE GUI Builder and using the Java Swing components. These components will provide the front end that is needed for the user interface. Next the functionality is added to the back end which provide the methods needed to invoke the File Server Node. The File Server interface would have the same methods as the user interface as the methods and properties which will be available to the users which are publically available from the File Server.

An RMIFileServer class is needed. This is the class which contains the method main. It is this method which sets the connection if you like. A reference is obtained between the client and the remote object. It is made possible through the RMI Name Registry. Sometimes called a namespace. It is accomplished through the Java Naming.lookup method. Like an id it creates the binding between the client and the File Server. This binding gives us the ability to pass the objects around from the client to the remote object.

An implementation class otherwise known as the servant class is generated. It represents the remote object skeleton and implements the remote interface methods. As with the RMIFileServer class a similar class is used for the client side. It contains the method main and has a Naming.lookup method.

4. Design Choices

The design choices made for this assessment are straight forward. Some templating has been provided with regards to the Server Side including the interface class and remote file server. Included in this is the Client Side class and a few basic ideas to have some form of permissions policy. The

design of the GUI features has also been provided and it will serve all the purposes for the functions needed. A reference of this is depicted in the figure below.

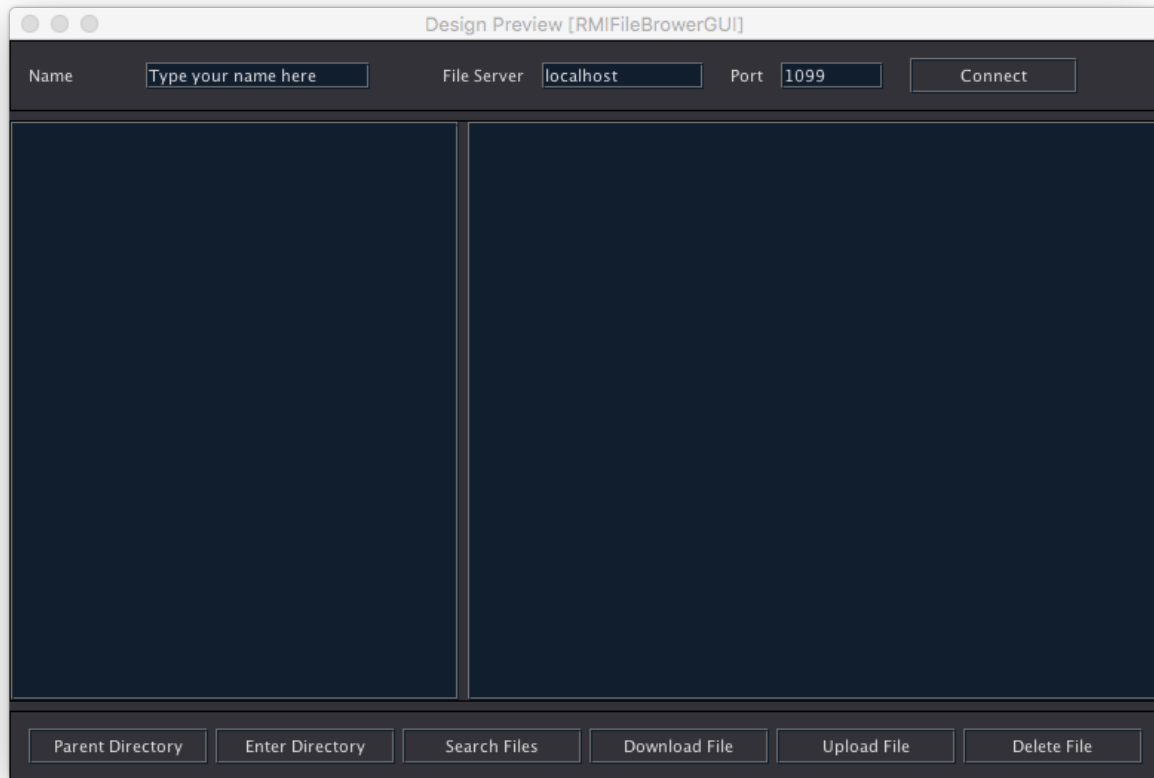


Figure 1 – GUI Design Features

5. Elaboration of Design

As shown in the Figure 2 below it outlines the interaction between the user and the File Server. It stipulates the functions that are available on the GUI and these correspond to the methods and functionality which the file server can deliver to the user. In Figure 3 the relationship between the components responsible for delivering the requests and the processes of how this is achieved.

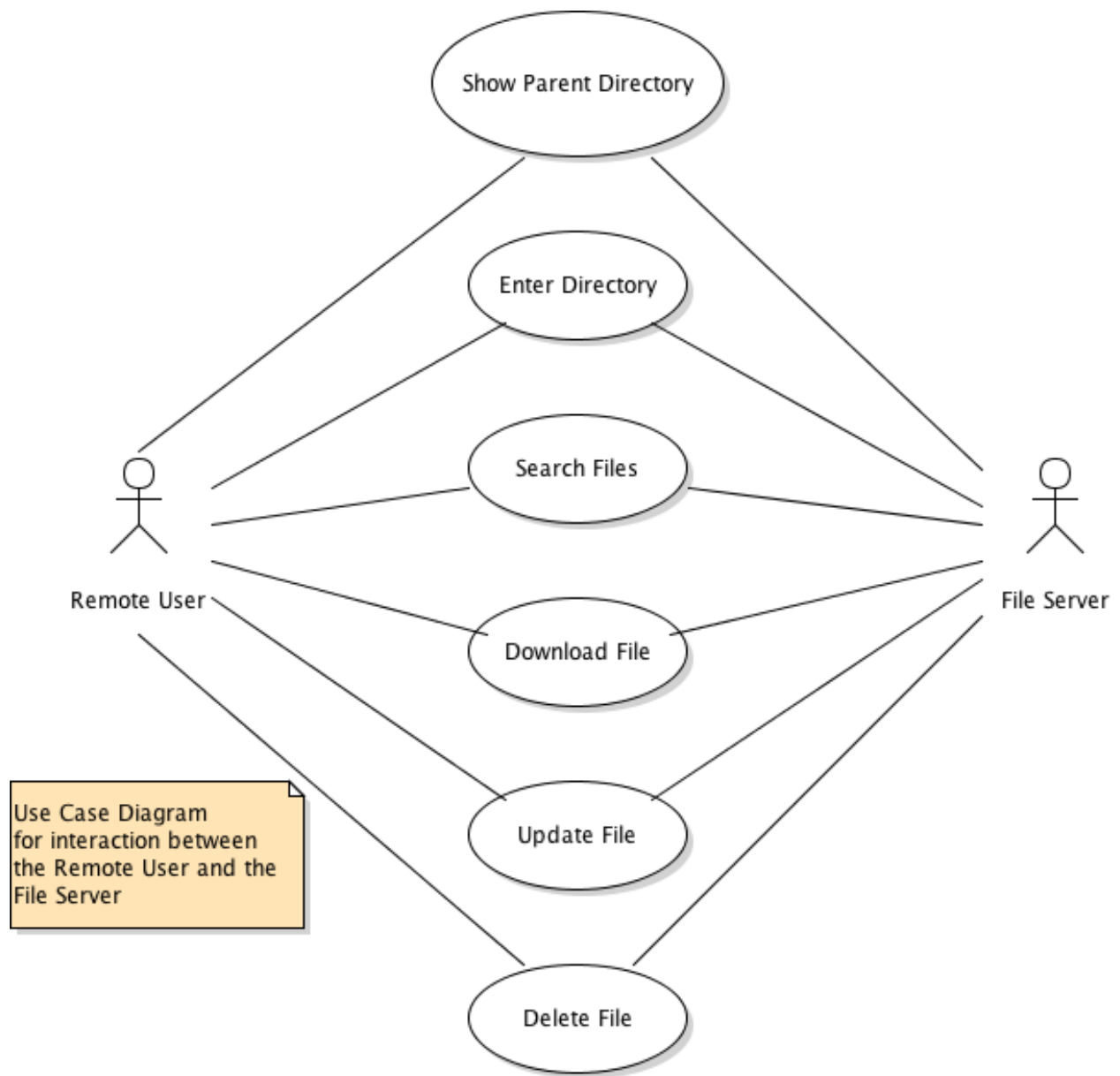


Figure 2 – Use Case Diagram showing interaction between File Server and User

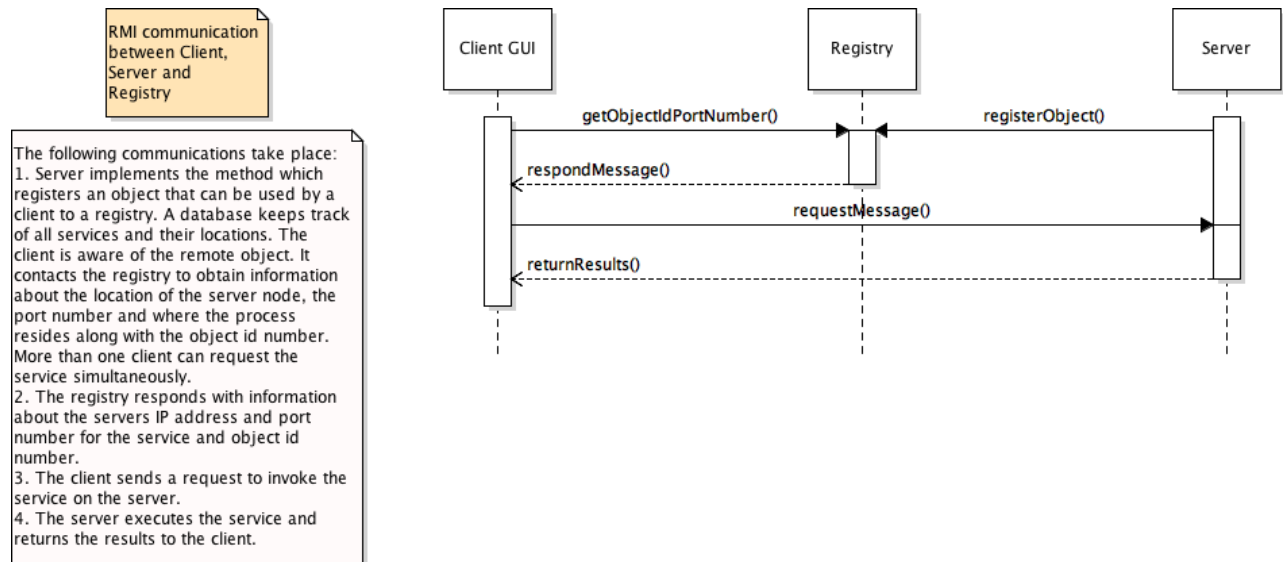


Figure 3 – Sequence Diagram of overall system

Conclusion

The knowledge taking away from the learnings and outcome of this project provides a good basis to develop other RMI based software. For instance, this development can also be implement in the Chat Application created in the previous assessment and modified in such a way that is uses RMI as opposed to using Java Socket Programming.