

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

-----*



BÀI TẬP LỚN

IT4371 - CÁC HỆ PHÂN TÁN

Đề tài 4: MyStorage

Nhóm sinh viên thực hiện:

Trần Trung Hiếu	20121700
Phan Công Huân	20121776
Phạm Đức Hùng	20121841

Giảng viên hướng dẫn: TS. Nguyễn Bình Minh

Hà Nội 4/2016

Mục Lục

LỜI NÓI ĐẦU	3
Danh mục hình ảnh	4
Danh mục bảng	5
Phân công công việc.....	6
I. MÔ TẢ BÀI TOÁN.....	7
1. Giới thiệu đề tài.....	7
2. Mục đích của đề tài.....	7
3. Hướng nghiên cứu.....	7
II. CƠ SỞ LÝ THUYẾT	8
1. Công nghệ Remote Method Invocation – RMI.....	8
1.1 Giới thiệu về RMI.....	8
1.2 Kiến trúc Java RMI.....	9
1.3 Áp dụng RMI vào hệ thống	10
2. Network Time Protocol - NTP	11
3. Ngôn ngữ lập trình.....	12
III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13
1. Phân tích hệ thống	13
1.1 Usecase của hệ thống.....	13
1.2 Biểu đồ hoạt động.....	14
1.3 Các lớp sử dụng trong hệ thống.....	15
2. Thiết kế giao diện.....	16
2.1 Giao diện cho ứng dụng trên PC	16
2.2 Giao diện cho ứng dụng trên mobile.....	17
IV. TRIỂN KHAI CÀI ĐẶT VÀ CHẠY CHƯƠNG TRÌNH	18
1. Hướng dẫn cài đặt đối với chương trình trên PC	18
2. Hướng dẫn cài đặt ứng dụng trên Mobile.....	19

TÀI LIỆU THAM KHẢO.....	20
--------------------------------	-----------

LỜI NÓI ĐẦU

Hệ phân tán trải qua những giai đoạn thay đổi đáng kể và điều này có thể được bắt nguồn từ ảnh hưởng một số xu hướng như sự xuất hiện của công nghệ mạng phổ biến, sự phát triển của máy tính ở khắp mọi nơi cùng với mong muốn hỗ trợ người dùng di động trong hệ phân tán, nhu cầu ngày càng cao của dịch vụ đa phương tiện, và nhiều xu hướng khác....

Xây dựng kho lưu trữ phân tán cũng là một trong xu hướng của hệ phân tán giải quyết nhu cầu lưu trữ tăng cao, và sử dụng đa nền tảng trên nhiều thiết bị khác nhau.

Nhóm chúng em xây dựng hệ thống còn nhiều thiếu điểm, sẽ tiếp tục bổ sung và phát triển trong thời gian tới. Xin cảm ơn đến sự hướng dẫn nhiệt tình của thầy TS. Nguyễn Bình Minh giúp nhóm em hoàn thành đề tài

Danh mục hình ảnh

Hình 2.1 Thành phần quan trọng nhất trong kiến trúc của RMI là các Interfaces

Hình 2.2. Sơ đồ thực thi dịch vụ Interface của client và server.

Hình 2.3 Các tầng chính trong kiến trúc RMI

Hình 2.4 Minh họa các nền tảng nhóm đã triển khai ứng dụng

Hình 3.1 Biểu đồ Usecase của hệ thống

Hình 3.2. Biểu đồ hoạt động của hệ thống

Hình 3.3 Các class sử dụng trong chương trình

Hình 3.4. Màn hình đồng bộ của Client trên PC

Hình 3.5 Màn hình file Exflorer của Server

Hình 3.6 Giao diện ứng dụng trên Mobile

Danh mục bảng

Bảng 1 Phân công công việc các thành viên

Bảng 2. Sự khác nhau giữa cơ chế Remote Object và Local Object.

Phân công công việc

Tên thành viên	Công việc
Phan Công Huân	Nghiên cứu công nghệ RMI, triển khai và cài đặt trên Client và Server trên PC, kiểm thử
Trần Trung Hiếu	Nghiên cứu, triển khai ứng dụng Client trên Android, viết báo cáo, kiểm thử
Phạm Đức Hùng	Viết báo cáo, Xây dựng giao diện cho client trên PC

Bảng 1 Phân công công việc các thành viên

I. MÔ TẢ BÀI TOÁN

1. Giới thiệu đề tài

Đề tài 4: MyStorage

Viết ứng dụng phân tán đơn giản chia sẻ và đồng bộ hóa dữ liệu.

Yêu cầu:

- Dữ liệu được phía client đưa lên máy Server
- Server thực hiện việc đồng bộ hóa tự động giữa các máy khách (Clients) dựa trên 2 yếu tố đồng thời:
 - kích thước của file dữ liệu
 - thời gian thay đổi file dữ liệu.
- Client tự động upload và download dữ liệu
- Cung cấp giao diện đơn giản (dạng Explorer) để hiển thị dữ liệu dạng phía Client
- Cung cấp giao diện đơn giản (có thể là CLI) để hiển thị dữ liệu phía Server
- Cho phép người sử dụng thoát ứng dụng tự động đồng bộ phía Client.

2. Mục đích của đề tài

Việc thực hiện đề tài này sẽ giúp chúng em hiểu được các kiến thức căn bản về hệ thống thông tin phân tán, các đặc điểm của hệ phân tán...

Việc triển khai cài đặt các thiết bị chứa dữ liệu và đồng bộ chúng với nhau sẽ giúp chúng em hiểu được bản chất của các hệ phân tán cũng như các vấn đề đặc trưng cần nghiên cứu phát triển trong hệ phân tán.

Nghiên cứu đề tài này sẽ giúp nhóm hiểu sâu hơn các kiến thức trên lớp, biết mô hình hóa bài toán và ứng dụng vào giải quyết vấn đề trong thực tế.

3. Hướng nghiên cứu

Nghiên cứu các công nghệ hỗ trợ truyền dữ liệu, đồng bộ dữ liệu theo thời gian thực.

Triển khai trên nhiều nền tảng khác nhau như PC, Android, Web...

II. CƠ SỞ LÝ THUYẾT

1. Công nghệ Remote Method Invocation – RMI

1.1 Giới thiệu về RMI

RMI là một công nghệ phân tán hướng đối tượng, nó cho phép ứng dụng gọi phương thức của các đối tượng xác định từ xa, chia sẻ tài nguyên và xử lý tải giữa các hệ thống. Không giống như tất cả các hệ thống thực thi từ xa khác, RMI cho phép mọi loại đối tượng java sử dụng, thậm chí nếu client hoặc server chưa từng kết hợp với nhau từ trước. RMI cho phép cả client và server tải động các loại đối tượng mới giống như đã yêu cầu.

Mục đích chính của việc thiết kế ra công nghệ RMI là cho phép các lập trình viên phát triển các chương trình Java phân tán với cùng cú pháp và ngữ nghĩa đã sử dụng cho các trường trình không phân tán.

Đặc điểm của RMI

Các nhà thiết kế RMI đã cố gắng tạo cách sử dụng các đối tượng Java phân tán giống như việc sử dụng các đối tượng Java ở cục bộ. Dưới đây là một vài so sánh giữa đối tượng cục bộ và đối tượng được gọi từ xa.

	Local Object	Remote Object
Định nghĩa đối tượng	Một đối tượng cục bộ thì được định nghĩa bởi các lớp trong Java	Hành vi của các đối tượng Remote thì được định nghĩa bởi một lớp kế thừa lớp Remote interface.
Đối thực thi (Object Implementation)	Một đối tượng cục bộ thì được thực thi bởi chính lớp Java đó	Hành vi của đối tượng remote thì được thực thi bởi lớp Java mà implements một Remote Interface
Khởi tạo đối tượng	Một thể hiện của đối tượng cục bộ được tạo ra bằng toán tử new	Một thể hiện mới của đối tượng remote được tạo ra trên máy chủ với toán tử new. Một client không thể trực tiếp tạo ra 1 đối tượng remote.

Truy cập đối tượng	Truy cập trực tiếp thông qua biến tham chiếu đến đối tượng	Thông qua một biến trỏ tới một <i>proxy stub implimentaion</i> của <i>remote interface</i>
--------------------	--	--

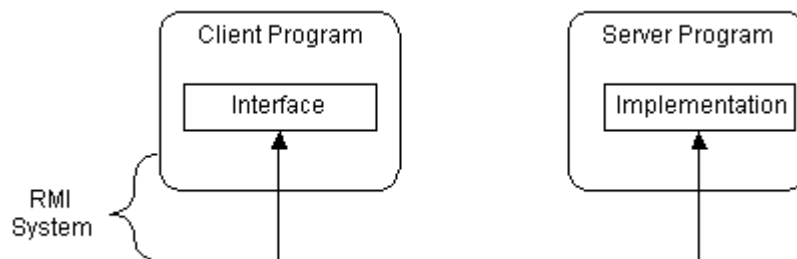
Bảng 2. Sự khác nhau giữa cơ chế Remote Object và Local Object.

1.2 Kiến trúc Java RMI

Interfaces

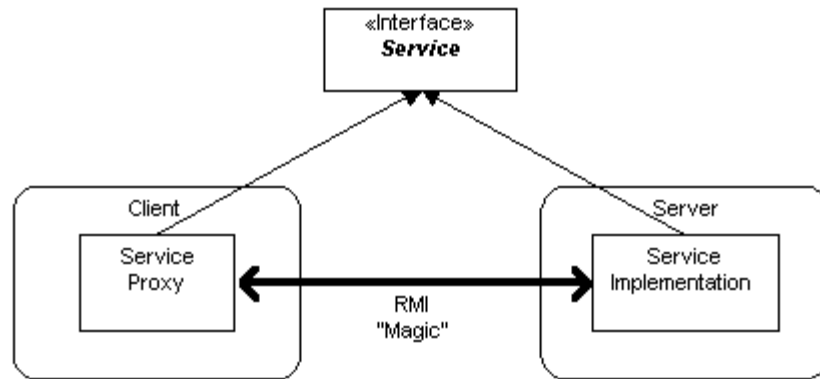
Kiến trúc RMI được xây dựng dựa trên một nguyên lý quan trọng: RMI cho phép các đoạn mã định nghĩa các hành vi và mã thực thi chúng được thực hiện riêng rẽ và chạy trên các máy ảo Java riêng rẽ.

Đặc biệt, trong RMI, việc định nghĩa một dịch vụ remote thì được mã hóa sử dụng Java Interface. Việc thực thi các remote service thì được viết mã trong các class. Bởi vậy từ khóa để hiểu được hệ thống RMI là *interfaces define behavior* và *classes define implementation*.



Hình 2.1 Thành phần quan trọng nhất trong kiến trúc của RMI là các Interfaces

Lưu ý rằng các lớp Interface trong java thì không chứa mã thực thi. RMI hỗ trợ 2 class được thực thi cùng một interface. Lớp đầu tiên thực thi các hành vi của Interface và chạy trên server. Lớp thứ 2 thông qua một proxy cho các dịch vụ remote và nó chạy trên client. Dưới đây là mô hình thực thi các interface của client và server.



Hình 2.2. Sơ đồ thực thi dịch vụ Interface của client và server.

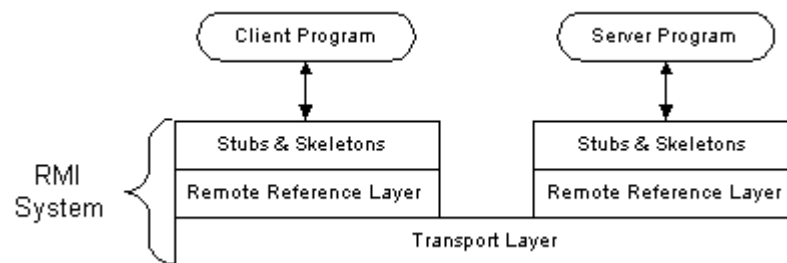
Các tầng của kiến trúc RMI

Bản chất hoạt động của RMI được xây dựng trên 3 tầng abstraction.

Đầu tiên là tầng *Stub and Skeleton layer*, tầng này trong suốt đối với lập trình viên. Tầng này chặn đứng các lời gọi hàm từ client tới các biến tham chiếu interface và chuyển hướng những lời gọi đó tới RMI service.

Tầng tiếp theo là *Remote Reference Layer*. Tầng này sẽ biết cách làm thế nào để chặn đứng và quản lý các tham chiếu tới các đối tượng Remote từ client.

Tầng *Transport Layer* được xây dựng dựa trên giao thức TCP/IP – kết nối giữa các máy trong mạng. Nó cung cấp một giao thức kết nối cơ bản, giống như chiến lược ngăn chặn sự thâm nhập từ bên ngoài của tường lửa.



Hình 2.3 Các tầng chính trong kiến trúc RMI

1.3 Áp dụng RMI vào hệ thống

Với Đề tài là MyStorage, nhóm cần phải triển khai được một hệ thống đồng bộ file đơn giản. Sử dụng công nghệ RMI, triển khai chúng cần các thành phần sau:

- Định nghĩa Interface cho các dịch vụ remote
- Implementations các dịch vụ remote

- Stub và Skeleton các file dữ liệu
- Một server để quản lý các dịch vụ remote
- Một service RMI Naming cho phép các client tìm thấy các dịch vụ service.
- Một lớp cung cấp 1 HTTP server
- Một chương trình client cần remote tới các dịch vụ của server.

Các bước triển khai trên hệ thống:

1. Viết và biên dịch các lớp Java Interface
2. Viết và biên dịch mã Java cho các class implimentaion
3. Sinh ra các file class Stub và Skeleton từ các lớp implementation
4. Viết mã Java cho service remote chương trình máy chủ
5. Phát triển code Java cho chương trình RMI tại Client
6. Cài đặt và chạy hệ thống RMI

2. Network Time Protocol - NTP

Tổng quan về NTP

Network Time Protocol là một giao thức mạng được sử dụng để đồng bộ các đồng hồ giữa các hệ thống máy tính với nhau. NTP là một trong những giao thức Internet cũ nhất mà hiện tại vẫn được sử dụng.

NTP cho phép đồng bộ đồng hồ của các máy tính khác nhau trong hệ phân tán thông qua mạng. NTP giao tiếp giữa client và server thông qua giao thức UDP tại cổng 123. Gói phần mềm NTP bao gồm 1 chương trình nền để đồng bộ thời gian giữa các máy tính trong điều kiện các máy tính được kết nối Internet. NTP hỗ trợ hệ thống các server phân cấp để quản lý tải lớn hơn từ các request client.

Trong NTP, server đã được kết nối trong một hệ phân cấp logic thì được gọi là *synchronization subnet*. Những cấp bậc của *synchronization subnet* được gọi là *strata*.

1. Tầng 1, máy chủ có thông tin thời gian chính xác nhất
2. Các máy chủ tại mỗi tầng như một máy chủ thời gian cho các máy chủ ở tầng thấp hơn.

Cách NTP hoạt động

Khi Server A liên lạc với Server B để thực hiện đồng bộ.

If $\text{stratum}(A) \leq \text{stratum}(B)$, thì A không đồng bộ với B

If $\text{stratum}(A) > \text{stratum}(B)$, thì:

Time server A đồng bộ với B

Thực hiện đồng bộ

Cập nhật Statum Time server A

$\text{stratum}(A) = \text{stratum}(B) + 1$

Đặc điểm của NTP

1. Đồng bộ chính xác theo thời gian của UTC
2. Có khả năng mở rộng: NTP Server được tổ chức phân cấp để tăng tốc độ đồng bộ và quy mô để có thể phục vụ một lượng lớn client truy cập
3. Tin cậy
4. Bảo mật: NTP protocol sử dụng việc xác thực để kiểm tra thông tin thời gian có từ các nguồn đáng tin cậy hay không.

3. Ngôn ngữ lập trình

Platform

Sau khi nghiên cứu đề tài, tìm hiểu công nghệ, nhóm đã quyết định triển khai trên 2 nền tảng chính. Một là trên máy tính cá nhân (PC) và trên Mobile chạy nền tảng Android.

Hình 2.4 Minh họa các nền tảng nhóm đã triển khai ứng dụng



III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

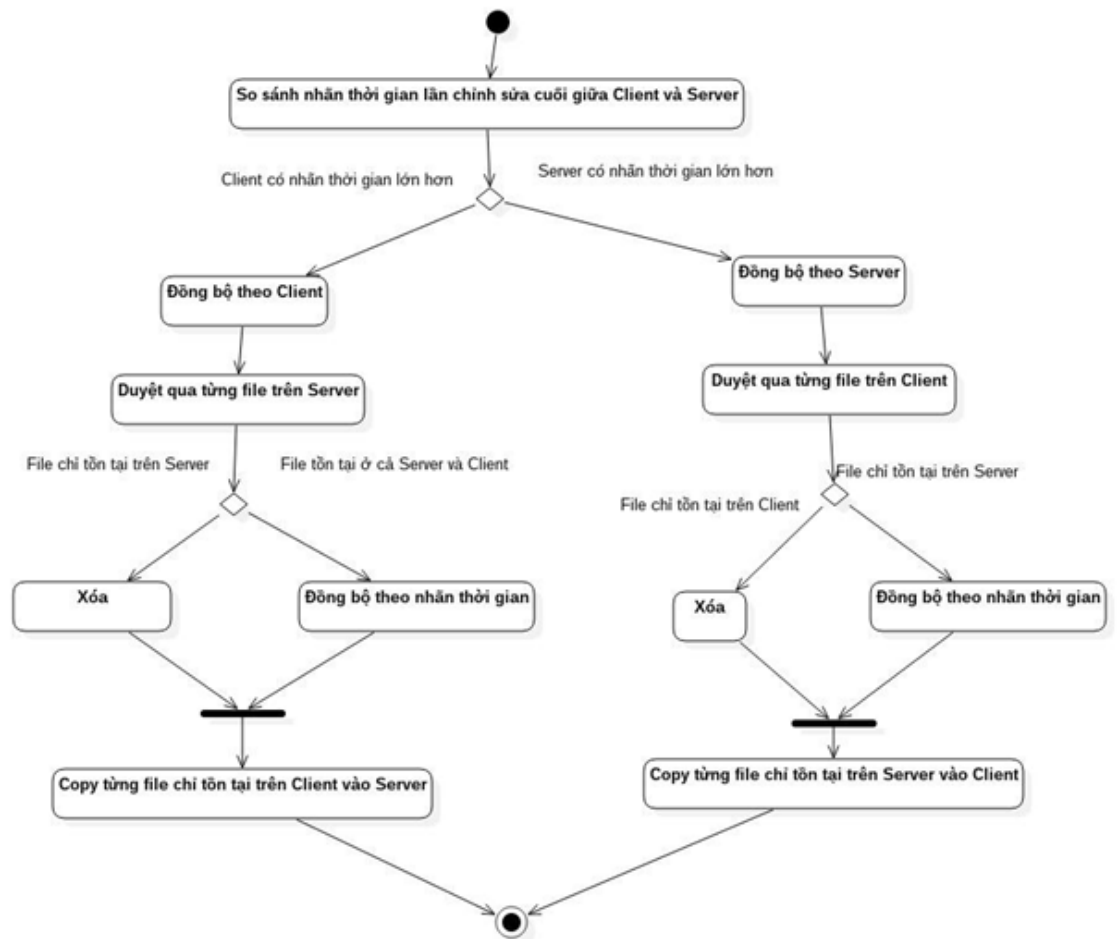
1. Phân tích hệ thống

1.1 Usecase của hệ thống



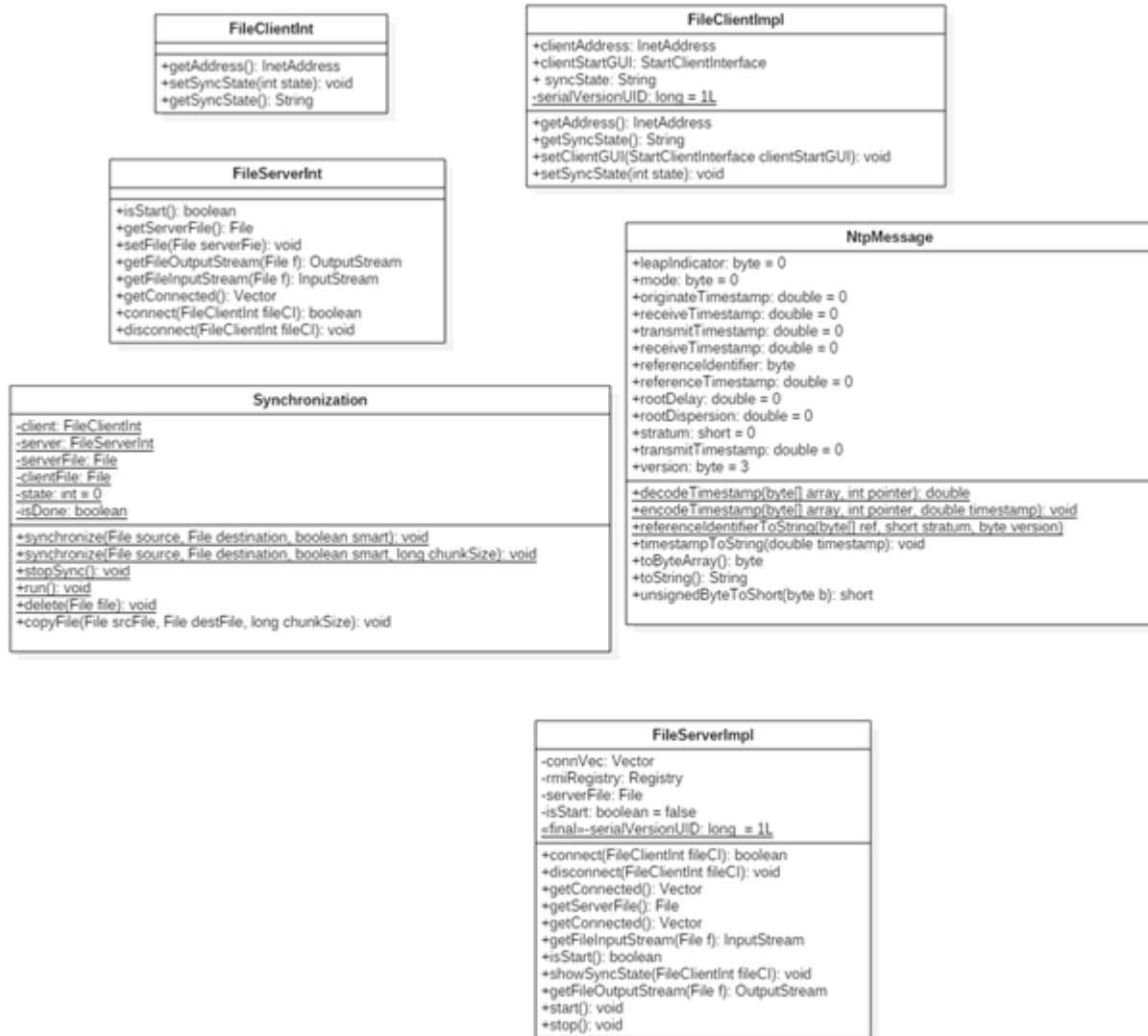
Hình 3.1 Biểu đồ Usecase của hệ thống

1.2 Biểu đồ hoạt động



Hình 3.2. Biểu đồ hoạt động của hệ thống

1.3 Các lớp sử dụng trong hệ thống

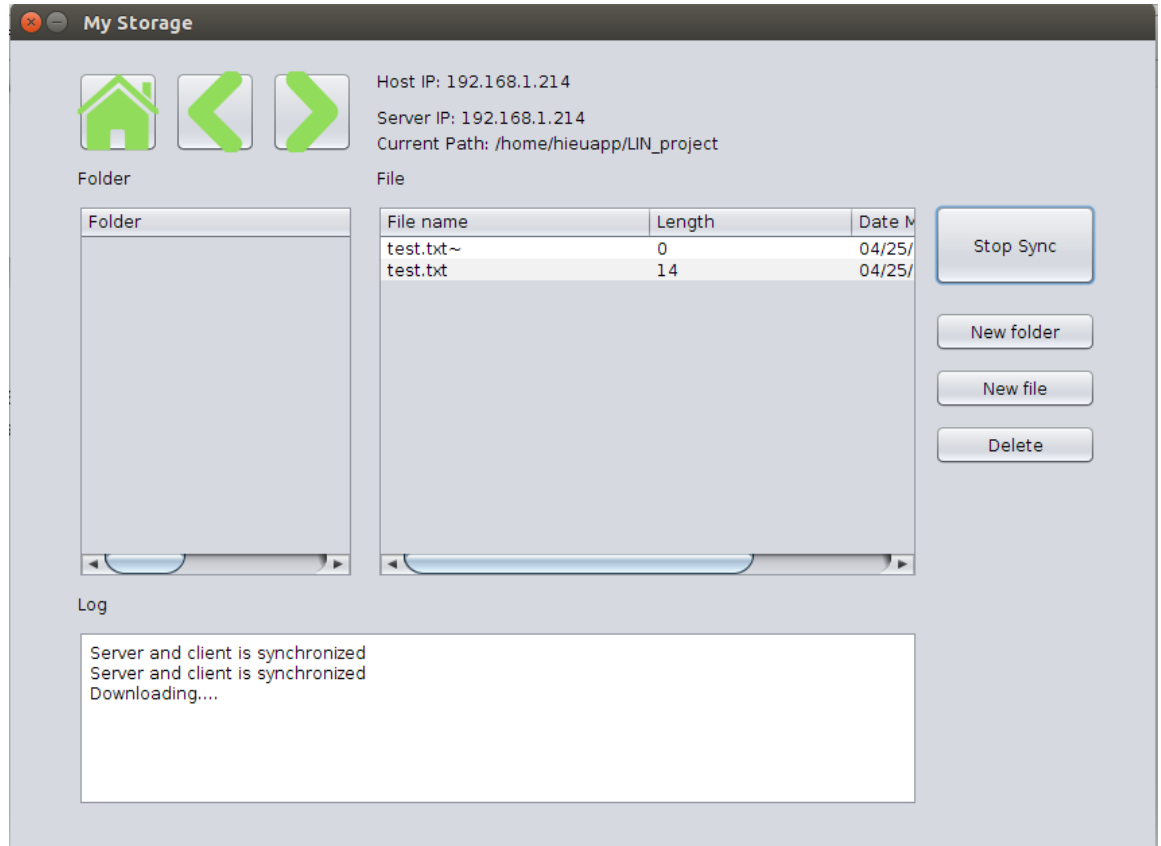


Hình 3.3 Các class sử dụng trong chương trình

2. Thiết kế giao diện.

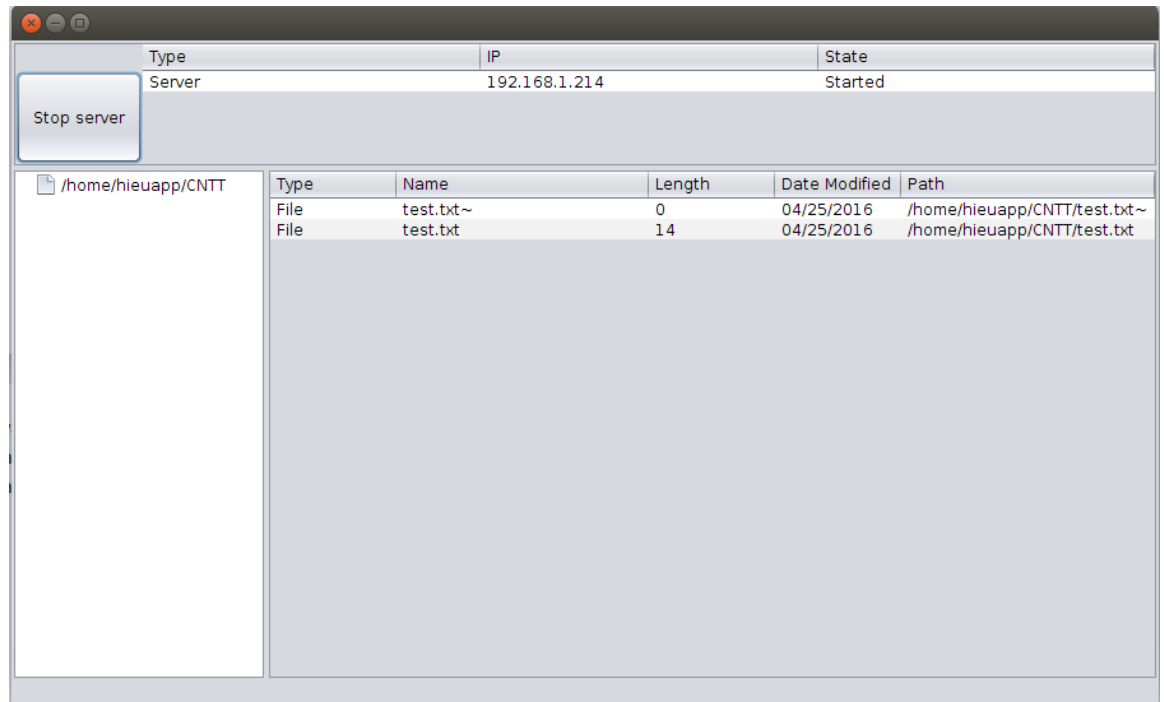
2.1 Giao diện cho ứng dụng trên PC

Giao diện Client



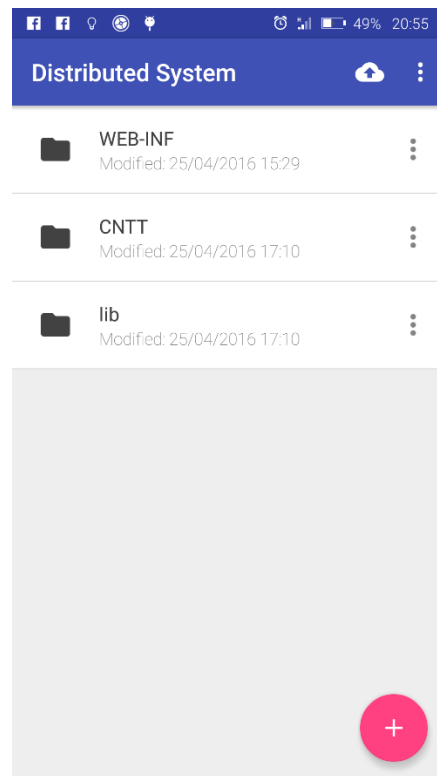
Hình 3.4. Màn hình đồng bộ của Client trên PC

Giao diện phía Server



Hình 3.5 Màn hình file Exflorer của Server.

2.2 Giao diện cho ứng dụng trên mobile



Hình 3.6 Giao diện ứng dụng trên Mobile

IV. TRIỂN KHAI CÀI ĐẶT VÀ CHẠY CHƯƠNG TRÌNH

1. Hướng dẫn cài đặt đối với chương trình trên PC

Trên PC cần phải cài đặt 2 chương trình, một dành cho client và một dành cho server.

Đối với client

Bước 1: Import mã nguồn vào IDE dành cho java (Eclipse).

Bước 2: Chọn class LoginGUI.java và run as java application

Bước 3: Nhập địa chỉ IP server vào ô textbox trên giao diện.

Bước 4: Chọn Folder để đồng bộ.

Bước 5: Nhấn Connect để kết nối tới server.

Bước 6: Sau khi kết nối được đến server, giao diện file Explorer hiện ra với danh sách file hiện có tại client. Nhấn và nút Đồng bộ để thực hiện đồng bộ file với Server.

Bước 7: Nhấn các nút Add File, Delete File để thêm file mới hoặc xóa file.

Bước 8: Nhấp nút X ở góc trên bên phải cửa sổ để kết thúc chương trình

Đối với server

Bước 1: Import mã nguồn vào IDE (Eclipse)

Bước 2: Chọn class LoginGUI.java và run as java application.

Bước 3: Chọn Folder root chứa file của server.

Bước 4: Nhấn start để bắt đầu chạy server.

2. Hướng dẫn cài đặt ứng dụng trên Mobile

Bước 1: Copy file cài đặt .apk vào bộ nhớ.

Bước 2: Mở file .apk lên, hệ thống sẽ tự động cài đặt.

TÀI LIỆU THAM KHẢO

1. TS. Nguyễn Bình Minh, *Slide bài giảng môn Các hệ phân tán*
2. Wikipedia.com, *Remote Method Invocation - RMI*