COMP3100 Group Project

## Project title

Intelligent cloud storage distributed system (ICSDS)

## Introduction

With the rapid development of science and technology in recent decades, especially the popularity of the Internet and the comprehensive construction of network infrastructure, humans have become increasingly dependent on network applications in the daily life and work job, and intelligent cloud storage applications are among the fastest-growing.

## Aims

Our goal is to enable customers to upload/share/transfer data through our ICSDS designed between different locations, devices, and even different OS, so that users can operate seamlessly data between multiple devices.

## Background

### Architecture

The architecture diagram of the ICSDS is shown in Figure 1. It separates the application services, data services, and log services. When one system service changes in the future, it will not affect other service systems in normal operation. At the same time, it is also convenient when upgrading one group of servers. Application services serve as interfaces that are directly connected to users and will access corresponding services, such as login services, data services, and log services, depending on business needs. The login service is mainly used for user login, the data service is used to handle user file upload and download, and the log service is used to record the user's login and logout and records of data operations.

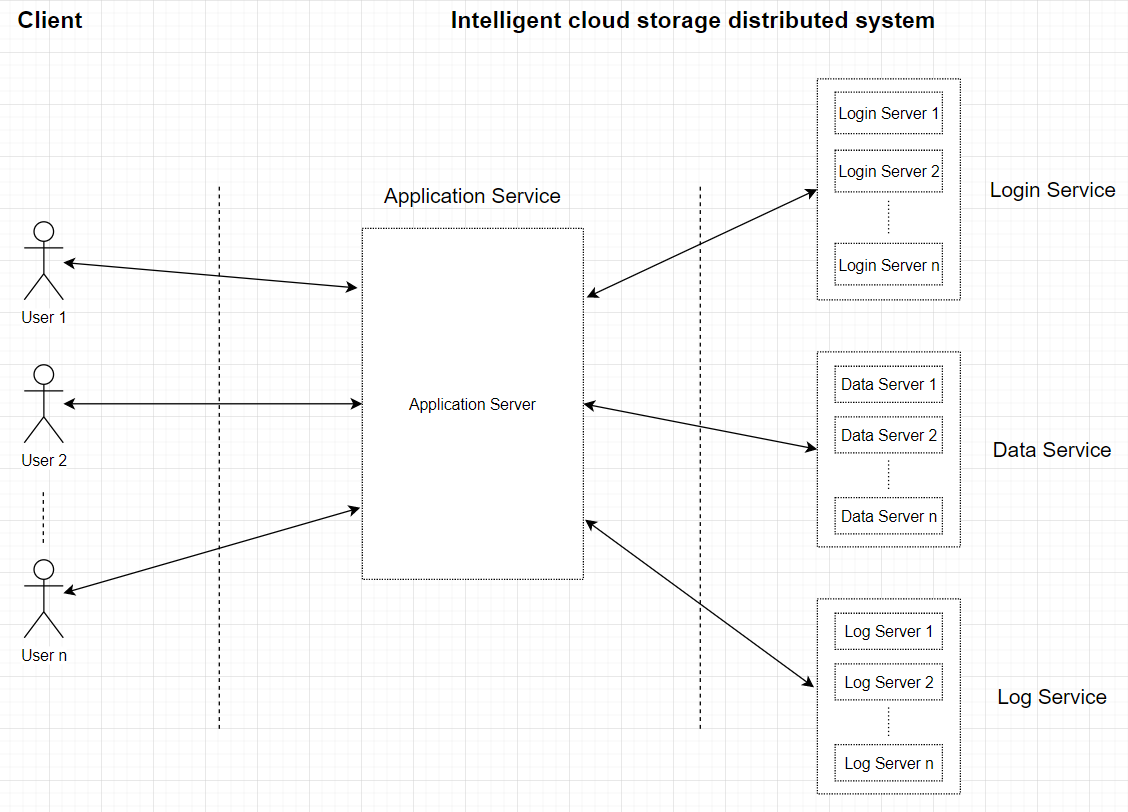


Figure ICSDS architecture

### Discrete even simulation

**initialise** system state variables by reading

• Simulation configuration file (e.g., ds-server-config.xml) in the server-side simulator

• Simulation configuration file (e.g., ds-client-config.xml) in the client-side simulator.

**Initialise** Clock (usually starts at simulation time zero)

Client **sign in**

* Login verification

**Insert a log** of user login

Client **choosing work mode**

* upload
* download

**While** all files are uploaded/downloaded **do**

• Insert a log when a file transfer completed

**End while**

**Client sign off**

**Insert a log** of user sign off

### Client-server communication model

ICSDS decided to use persistent and synchornous as the client-server communication model, because when users are transferring files, it needs to keep the communication persistent. Moreover, when the communication is interrupted, the server can retain the message which is sent unsuccessful and resend it when the communication is restored, rather than discarding the message directly.

### Simulation steps

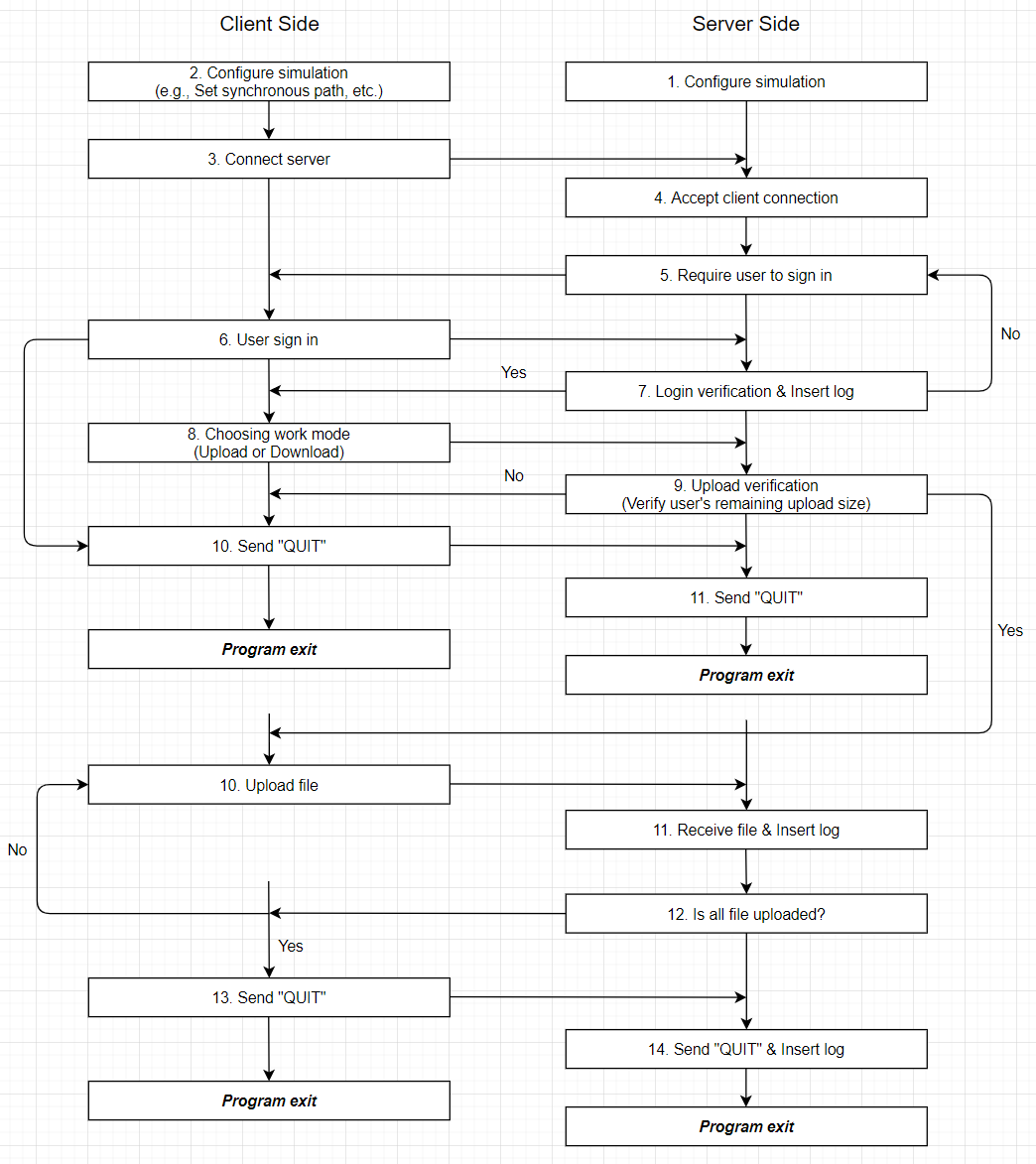


Figure simulation steps – upload

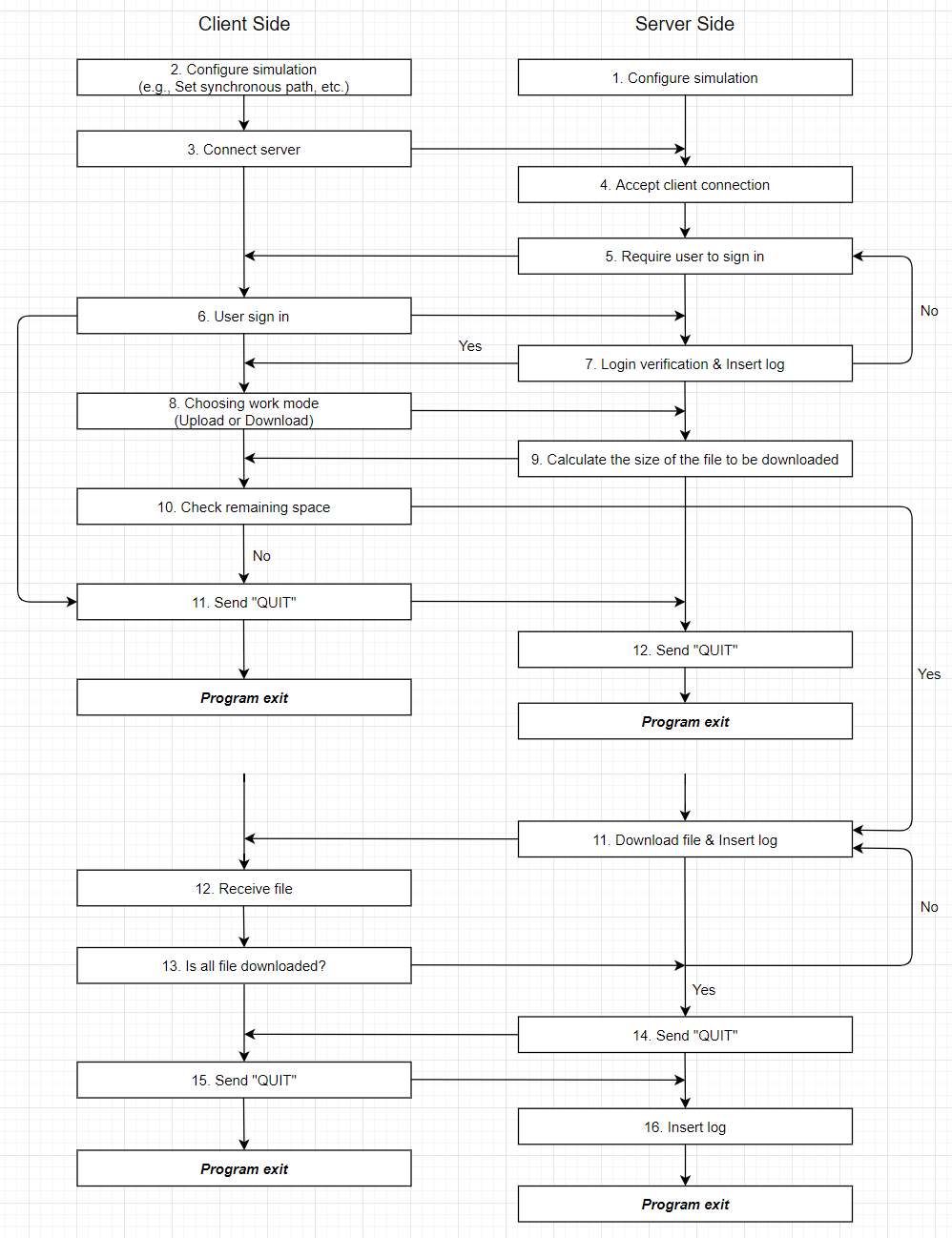


Figure simulation steps - download

## Project plan

## References