Окно авторизации

Окно регистрации

Управление файлами (переименовать/удалить/скачать) --

Не показывать корневую директорию (запретить изменения

Отображать список файлов пользователя - таблица «

Прогресс бар для загрузки (ромашка - ожидание

выпадающее меню, кнопки

корневой директории)

загрузки/скачивания файла)

Автологин для последнего пользователя •

Кнопка Logout

Окно для загрузки файла на сервер 🥌

ServerClass - создание серженного сокета, прием подключений, подключение к БД ClientHandler – обработака подключённого клиента 🥌 Server Создание отдельного thread под клиента (отдельного или через ThreadPool) Common Authorization < Выбрать технологию 🛑 Окно авторизации 🥌 Client ДЗ Простой список отображения (можно только имена файлов) Подключение к серверу, создание сокета « Выбрать технологию < БД Создать табличку Swing/FX/AndroidApp/JSF < Интрефейс авторизации Клиент Основной интерфейс Общая идея У каждого пользователя своя папка по уникальному ID 🥌 Упрощенная структура - один пользователь одрна папка с файлами Основная структура - папка пользователя с подпапками Изменять имена одинаковых файлов -Хранение данных Конфликт имён - как его разрешать? Не разрешать пользователю операции, приводящие к появлению файлов с одинаковыми именами Работа через JDBC драйвер SQL база (SQLite/MsSQL/MySQL/PostgreSQL) Таблица с пользователями: id - autoincrement, login, password, timestamp, путь к папке на сервере Таблица со структурой каталогов Логгирование в БД Работа через Hibernate

Облачное хранилище (файлы пользователя хранятся на Отображение структуры файлов в папке пользователя Клиент/серверное взаимодействие работает на сокетах ObjectInputStream/ObjectOutputStream (File) Передача файлов в виде ByteArray Отображение структуры файлов и папок в папке пользователя на сервере Класс User (отвечающий за объект из базы) Логгирование Консольное приложение (графика очень опционально) Создание серверного сокета Принимает запрос на сокет от клиента, устанавливается соединение и создается новый объект ClientHandler (обработчик клиента) Сервер Массив с подключенными клиентами (не использовать Vector) Допускается использование Collections.synchronizedList / или класса, подходящего под ваши требования из java.util.concurrent Отдельный Thread на каждого клиента. Для практики можно использовать ThreadPool Файлы хранятся на жестком диске сервера в специально выделенной директории Работа с файлами (загрузка файла на сервер/переименование/удаление) Завершение сессии пользователя (условия корректного завершения работы - передача файлов/работа с БД/закрытие сокета...) Работа с папками (создание папки/перемещение файлов между папками/удаление папки) Упрощенная форма - пользователь вводит несуществующий логин и тогда создается запись по этому логину и введенному Регистрация Основная форма - открывать в новом окне и предлагать пользователю придумать свои данные для входа в систему Проверяем, не занят ли предложенный логин Создать пользователю дисковое пространство (уникальную папку с названием по логинуіd/придумать свой алгоритм) Сервис авторизации Создать пользователя в БД с указанными данными и папкой Сразу залогинить пользователя, который зарегистрировался Шифрование пароля Запрос на восстановление пароля Проверяем в БД данные пользователя

Подключаем, если все ок.

Можно ли заходить под одним логином с разных устройств?

Авторизация