

ГУАП
КАФЕДРА
№52

ПРЕПОДАВАТЕЛЬ

доц., к. т. н.		Е.М. Линский
должность, уч. ст., звание	дата, подпись	инициалы, фамилия

КУРСОВАЯ РАБОТА
по курсу: ТЕХНОЛОГИИ
ПРОГРАММИРОВАНИЯ

на тему: СИСТЕМА ОБМЕНА ФАЙЛАМИ
ЧЕРЕЗ ВЕБ

РАБОТУ ВЫПОЛНИЛ

Студент гр. 5723		А.А.Меллер
	дата, подпись	инициалы, фамилия

Содержание

Функциональная спецификация	3
Инструкция пользователя	4
Архитектура	6
Особенности реализации	6
Тестирование	8

1 Функциональная спецификация

Данная программа предоставляет возможность пользователям обмениваться файлами посредством сервера. Каждый пользователь имеет доступ не только к своей директории, но и к директориям, которые открыты ему другими пользователями. Админ имеет доступ ко всем директориям.

Интерфейс пользователя будет представлять из себя несколько веб-страниц. Основные страницы: страница регистрации, страница со списком видимых директорий и страница со списком доступных файлов в директории.

На Рисунок 1 изображает что-то интересное, строка для проверки правил.

Для обеспечения хотя бы минимального уровня безопасности будет организована аутентификация по логину и паролю.

Возможности программы:

1. Регистрация нового пользователя;
2. Скачивать файлы из открытых пользователю директорий;
3. Загружать файлы в свою директорию;
4. Делиться своей директорией с другими пользователями;
5. Искать файлы по открытым ему директориям;
6. При открытии директорий другим пользователям можно указать модификатор доступа: "Только чтение" и "Чтение и редактирование";
7. Директории имеющие модификатор доступа "Только чтение" невозможно редактировать;
8. Можно сгенерировать прямую ссылку на скачивание файла из открытых пользователю директорий;
9. Если пользователь является админом, то он имеет полный доступ (с возможностью изменения

всех элементов) ко всем директориям;

10. Админ так же может производить поиск по всем директориям.

2 Инструкция пользователя

После открытия сайта необходимо зарегистрировать новый аккаунт или войти в уже существующий. После входа появляется список доступных директорий (каталогов).



Log In!

Your login

Your password

Login

Register

Рисунок 1 - Форма входа

Зайдя в любую из них, появляются доступные для скачки файлы, а так же кнопка "Загрузить" для загрузки нового файла (доступно только для своей собственной директории). Вернуться назад можно будет нажатием кнопки "Назад". Для поиска файла необходимо заполнить форму вверху страницы и нажать кнопку "Найти".



Welcome in your disk!

+Napomosh
+admin
+Your links:

Поиск файла Search

Обзор... Файл не выбран. Napomosh Upload

Give access: admin Only read Read and modification Share

Log out

Рисунок 2 - Главная страница

Если пользователь хочет поделиться с кем-то своим каталогом он может нажать кнопку

"Поделиться"и еще один
Рисунок 1 для проверки правил

выбрать имя другого пользователя в списке, а также модификатор доступа. Если необходимо просто дать другому пользователю доступ к файлам (без возможности изменять и удалять файлы), то следует указать модификатор доступа "Только чтение".

ние". Если мы открываем полный доступ со всеми возможностями, то следует выбрать модификатор "Чтение и редактирование".

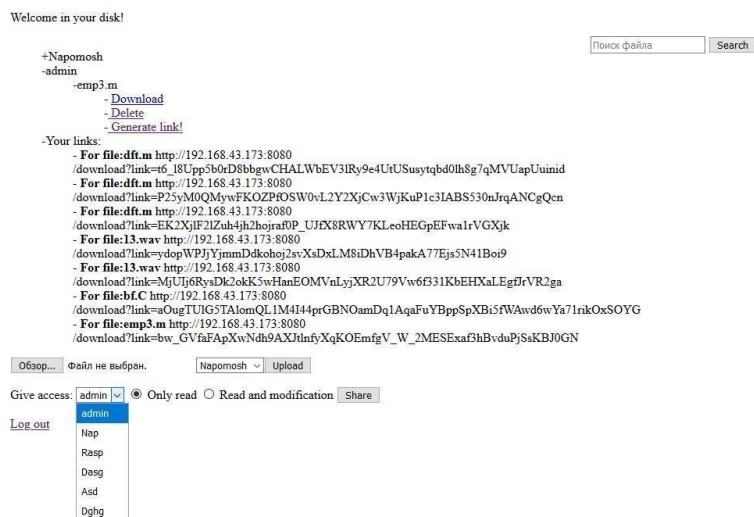


Рисунок 3 - Главная страница(вид с раскрытыми списками тех, кому можно расшарить)

Отметим, что в каталогах, которыми с пользователем кто-то поделился, нельзя производить изменения. То есть они доступны только для скачивания и просмотра списка файлов.

У админа после входа в аккаунт появляются сразу все существующие директории. Соответственно админу доступны все действия обычного пользователя, но с доступом ко всем директориям сразу. Стоит отметить, что для админа все директории имеют по умолчанию модификатор доступа "Чтение и редактирование".

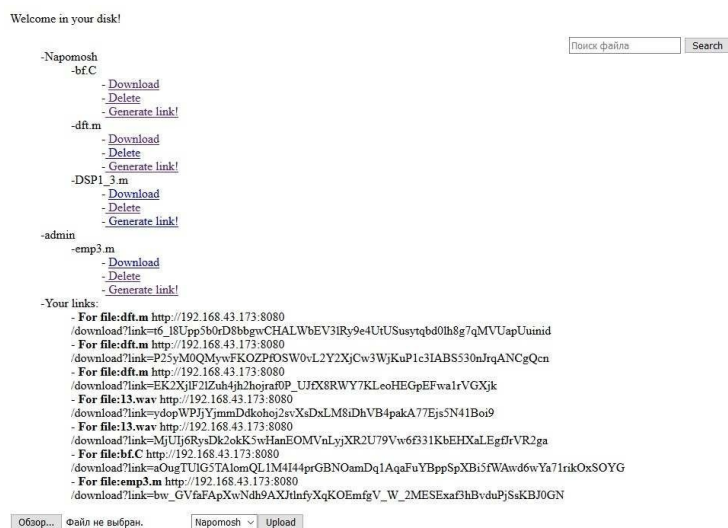


Рисунок 4 - Главная страница(вид с раскрытыми списками)

3 Архитектура

Программа состоит из трёх основных пакетов: auxiliary, controller и pages. Первый пакет представляет собой модель, то есть реализует работу с базами данных(читает и записывает txt файлы, передает информацию контроллеру по запросу). Второй - это контроллер. Он получает запрос клиента, обрабатывает его и передаёт в модель, затем полученный ответ опять обрабатывает и возвращает клиенту. Последний пакет является видом и содержит html страницы, css стили и скрипты JavaScript.

Вид ничего не знает о существовании модели и БД, он только переда- ет запросы пользователя в контроллер и отображает пользователю всю необходимую информацию.

Модель не знает о существовании контроллера и вида. В ней реали- зованы только методы работы с БД.

Контроллер занимается пересылкой запроса пользователя в БД(через модель) и результата обработки этого запроса к пользователю(через вид).

4 Особенности реализации

Рассмотрим основные методы программы по частям.

Начнём с модели. Модель содержит два класса: `WorkWithDirectory` и `WorkWithUsers`. Задача первого заниматься всем, что связано с директориями пользователей. Задача второго заниматься всем, что связано с

пользователями и их личными данными.

Основные методы класса `WorkWithDirectory`:

1. `public static ArrayList<String>getAllAvailableDirectoriesForUserWithoutMod(String user)` - возвращает список всех открытых пользователю директорий без учета модификаторов доступа;
2. `public static void addDirectoryToUser(String directory, String user, String mod)` - добавляет пользователю `user` новую директорию с соответствующим модификатором доступа `mod` и именем `directory`;
3. `public static ArrayList<String> doSearch(String user, String searchRequest)`
- производит поиск по запросу `searchRequest` по всем директориям пользователя `user`;
4. `public static ArrayList<String> getAllEditableDirectoryForUser(String user)` - возвращает все директории пользователя `user`, которые он может редактировать;
5. `public static void doDelete(String file, String directory)` - удаляет файл с именем `file` из директории `directory`;
6. `public static void uploadFile(Part file, String directoryForUpload)` - загружает файл пользователя `file` на сервер в директорию `directoryForUpload`;
7. `public static void downloadFile(String fileForDownload, String nameOfDir, PrintWriter writer)` - позволяет пользователю скачать файл `fileForDownload` из директории `nameOfDir`;
8. `public static void generateLinkKey(String directory, String fileName, String author)` - генерирует уникальный ключ для прямой ссылки на скачивание файла `fileName` из директории

directory.

Основные методы класса WorkWithUsers:

1. `public static boolean isUserRegistered(String name)`
- проверяет зарегистрирован ли пользователь `name` или нет;
2. `public static void addNewUser(String name, String password)` - добавляет пользователя `name` с паролем `password`.

Рассмотрим контроллер. Его цель - это посредничество между видом и моделью и грамотная обработка запросов пользователя (полученных с вида) и ответов модели. Рассмотрим основные классы:

1. DeleteServlet - наследуется от класса HttpServlet. Его роль в том, чтобы обработать запрос пользователя на удаление какого-то файла;
2. DownloadServlet - наследуется от класса HttpServlet. Его роль в том, чтобы отдать пользователю файл, на который поступил запрос на скачивание;
3. MainServlet - наследуется от класса HttpServlet. Его роль в том, чтобы отрисовать главную страницу системы пользователю, передавая виду информацию обо всём, что должно быть видно пользователю;
4. SearchServlet - наследуется от класса HttpServlet. Его роль в том, чтобы производить поиск по файлам в соответствии с запросом пользователя;
5. SharingServlet - наследуется от класса HttpServlet. Его роль в том, чтобы верно обработать действие пользователя "Поделиться";
6. UploadServlet - наследуется от класса HttpServlet. Его роль в том, чтобы загрузить файл пользователя на сервер.

Третий пакет хранит в себе:

1. mainPage.jsp - страничка для отображения всей информации, доступной пользователю;
2. loginPage.jsp - страничка с формой входа пользователя;
3. style.css - файл со стилями для всех страниц;
4. script.js - JavaScript для организации сворачивающихся списков.

5 Тестирование

В программе реализовано модульное тестирование публичных методов модели.

Есть два тестирующих класса `WorkWithDirectoryTest` и

WorkWithUsersTest, которые тестируют соответствующие классы модели WorkWithDirectory и WorkWithUsers. Перечислим тестирующие методы: public void testGetListOfFiles(), public void testIsDirectoryEmpty(), public void testGetAllAvailableDirectoriesForUserWithoutM public void testHasUserThisDirectory(), public void testDoSearch(), public

void testIsDirectoryEditable(), public void
testGetAllEditableDirectoryForUsers(), public void
testPrepareDownloadFileWithLink(), public void
testHasUserThisFile(), public void testIsUserRegistered(), public
void testIsAuthDataValid(), public
void testFindForbiddenSymbols(). Все эти методы
тестируют соответствующие им методы модели.
Тестирование происходит по одному плану: задаются
ожидаемые данные при определённых входных
параметрах, затем вызывается тестируемый метод с
теми же входными параметрами. Результата теста
определяется с помощью сравнения ожидаемых
данных и результатами работы тестируемого метода.