

JAVA Project : A File Management System

Gaurav Yadav

Date: Sept 15, 2020

1 Project Description

The Objective of the Project is to make a viable product for file management with the following functionalities. The project contains the following functionalities:

1. **Add File:** This option is used to add a file to the system. Duplication of files are not allowed. The filename can only contain 3 to 15 character which allows only alphabets and () underscore symbol to form the name.
2. **Delete File:** This options deletes the file if it is in the system. The file names are case-sensitive.
3. **Find File:** This options is used to display the file info of the file if it is contained in the system.
4. **Get Files by Ascending order:** This options retrieves all the files detains in the alphabetical order of their name.

These options can be selected from the main context which is available right after program execution. To exit the program use the last option.

```
Welcome to Custom File Management System V1.0
Author : Gaurav Yadav

Main Menu
=====
1)Create a File
2)Delete a File
3)Search File By Name
4)List All Files
5)EXITTTTTT
Please Enter your appropriate choice(1-5)
```

Figure 1: Welcome Screen for the user

2 Git Repository

HTTPS Link: https://github.com/GauravYadav83/Java_Project.git

SSH Link: [git@github.com:GauravYadav83/Java_Project.git](ssh://git@github.com:GauravYadav83/Java_Project.git)

3 Project Structure

1. **Model:** This package contains the class file of the file_object. This package defines a Serializable class with a Comparator for the file_object type.
2. **DAO and DAOImpl:** This Layer deals with the actual creation and deletion of the data objects. This service implements the Data structure for storing the file_objects and also the retrieval of the file_objects in a sorted order.

3. **Service and ServiceImpl:** This Layer is responsible for the checks on the input provided by the user. This layer handles most of the exceptions and also for the desired effect of the certain operations on the data objects.
4. **Exception:** This package defines the Exception class to be used in our Exception Handling. Our FileNotFoundException and Duplicate Files exists error are handles by this class.
5. **Main:** This class implements the interface of the project. It interacts with the service layer to provide functionalities to the user and interacts with the user to take input and manage the flow of the program.

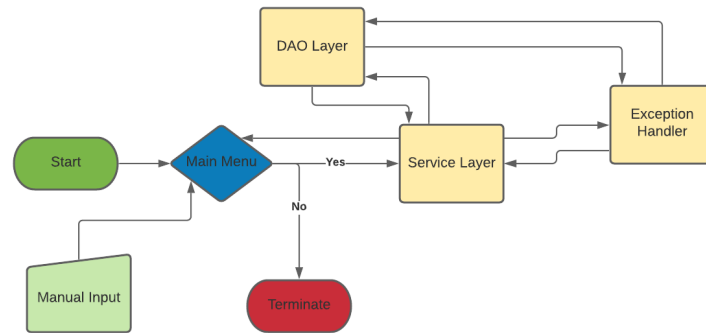


Figure 2: Flow Chart for the Program

4 Scrum Details

The Scrum Master, Team Leader, Consumer Head and Developer position were all part of my job. So a unclear segregation of jobs is inevitable and lead to bigger case stories. The scrum divided the project into two sprints.

1. **Sprint 1:** This sprint comprised of Service layer and DAO Layer Implementation. The objective of this sprint is provide a working framework for the project without handing exceptions. This sprint required a total 3 hours.
2. **Sprint 2:** The objective of this sprint was to implement the main function and the handle the various exceptions. This provides the interface to the user. This phase also involved testing. This sprint required a total of 5 hours.

5 Java Concepts Used

1. **Interface:** This is required to represent the Service layer and Data layer as two interfaces which can be used to communicate between the layers. The Service layer performs the various user info checks while the DAO layer implements the core functionalities.
2. **Exceptions:** This class is used for the exceptions handling. This is very useful for the user experience to hide the error details and have a seamless flow of the program without any interruption. This class provide a prompt to the user on errors in a easy and compact way and how to handle the problem. It involved a heavy use of try catch blocks.
3. **File IO:** This part of the program was used in the Service layer to generate the actual files in the working directory and populating it with some data. It also involves the cleanup on deletion of files. The is also required in the serialization and de-serialization of the file objects.
4. **Comparators:** This is useful for the efficient sorting of file objects according to the file name. This was used to sort the file in alphabetical order.

6 Data Structures

1. **File_object Class** : This is the core class of the project to represent a file in the system. The class is Serializable and is used to fillup our list and Map Data Structures.
2. **HashMap**: This is used to store our mapping of filename to file_object. This is a static member of our DAO class to represent all the file objects in a single mapping. The storage and retrieval operations are constant order operations to have a fast retrieval.
3. **List**: This is used to hold the file objects to present to the user. This is created using Collectors. Collections is then used to sort this list according to the Comparator defined for the base class.

```
Enter File Details Below.....
Enter File Name
logfile
File created with details File_object data for fd=0
filename=logfile
file_size=1
file_path=/logfile
access_permissions=777
data :
This is a file created using File Object implementation
The file name is logfile
```

Figure 3: File object details

7 Future Work

1. The current system is not persistent. Support for serialization is to be added to store the HashMap on the system and populate the directory structure using it.
2. Various Metadata has been defined for the File object class to have a future support for the data manipulation layer and Access control layer. Currently they are defined as constants.
3. The interface need to be defined in a more clean way providing more control to the user.