

Virtual Key for Your Repositories

Project 1

INDEX

- Project and developer details
- Sprints planned and the tasks achieved in them
- Algorithms and flowcharts of the application
- Core concepts used in the project
- Links to the GitHub repository to verify the project completion
- Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)

Project and developer details

Project:

Java Application program to create a Virtual Key Repository having following features:

- Display welcome screen with company and developer details.
- Generic features and three operations:
 1. Retrieving the file names in an ascending order
 2. Business-level operations:
 - Option to add a user specified file to the application.
 - Option to delete a user specified file from the application
 - Option to search a user specified file from the application
 - Navigation option to close the current execution context and return to the main context
 3. Option to close the application

The goal of the company is to deliver a high-end quality product as early as possible.

Developer Details:

Name: Ragula Rupasi

Email: rrupasiragula@teksystems.com

Organization: Tek Systems Global Services

Sprints planned and the tasks achieved in them

A total of four sprints were planned in completing the application.

Different tasks achieved in sprint planning were:

- Selecting items from the Backlog
- Estimating time to complete each Backlog item
- Self-organization
- Tracking progress
- During the Sprint to changes must be done
- Final Thoughts
- After the sprint adapting to changes

In all the four sprints above mentioned steps are repeated in cyclic order.

A briefing of activities in three sprints are as below:

Sprint 1

Construction of navigation menus – main menu and business level operations menu.

Sprint 2

Adding functionalities to option in menus.

Sprint 3

Optimizing algorithms and error handling.

Sprint 4

Testing the application and deployment.

Algorithms and flowcharts of the application:

Application has two flowcharts as mentioned below:

- **Main Flow** - Used to initiate the application and perform three generic operations.
- **Business Level Operations Flow** – Used to represent business level operations used and navigation to main menu.

Algorithms used in application are of two types as mentioned below:

- Searching Algorithm
- Sorting Algorithm

Searching Algorithm is used to search a file from a list of files present in the folder. Since files are not always in sorted order, the search algorithm implemented in application is **Linear Search**.

Time complexity of Linear Search is **$O(n)$** , where n is size of the list.

Sorting Algorithm is used to sort the files in ascending order in the current working directory. Built in method called **Arrays.sort()** is used to sort the array.

Big-O time complexity is **$O(N \log(N))$** . The sorting algorithm is a modified mergesort (in which the merge is omitted if the highest element in the low sublist is less than the lowest element in the high sublist)

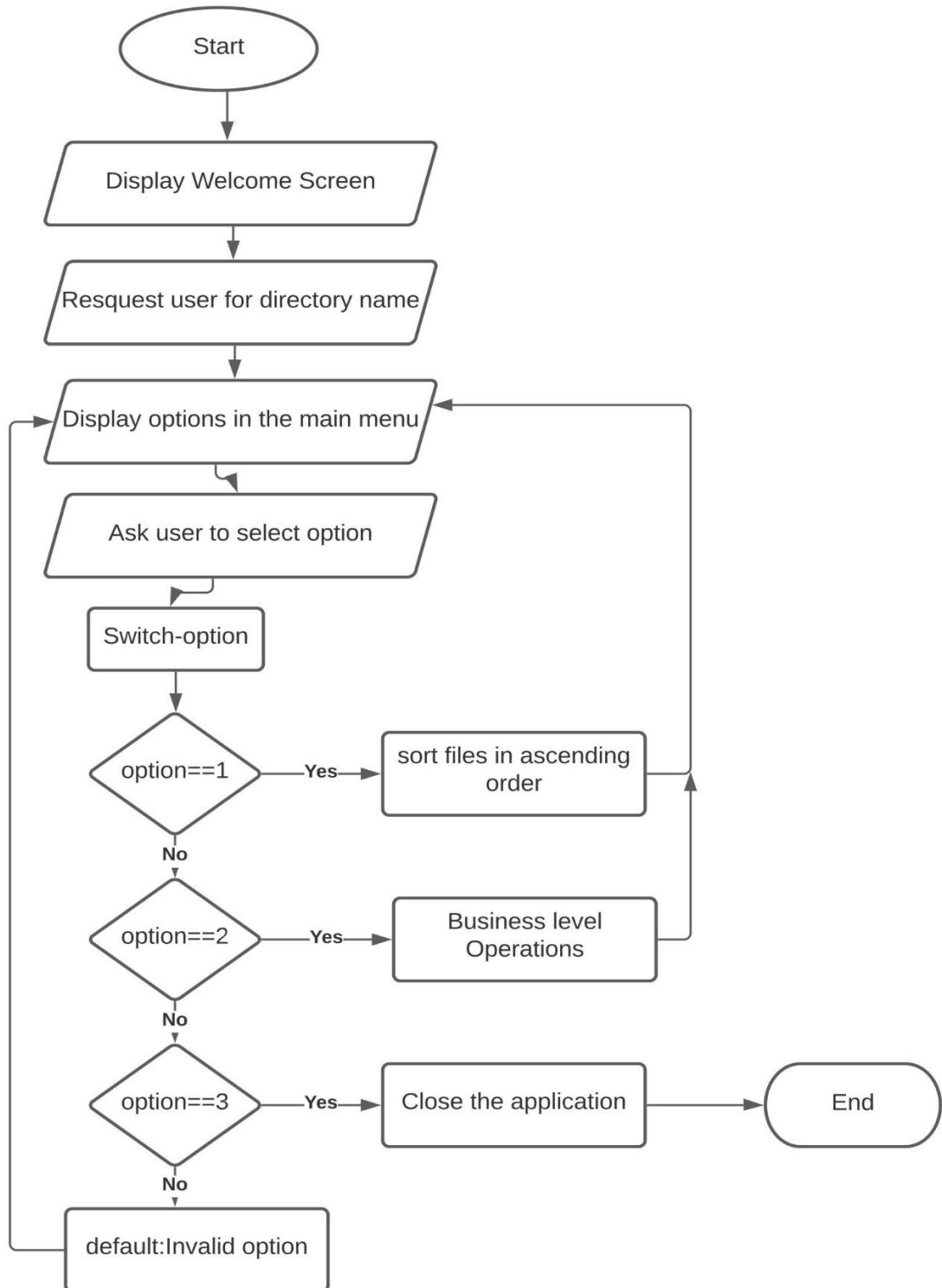
```
import java.util.Arrays;
```

This statement must be included to use built in function.

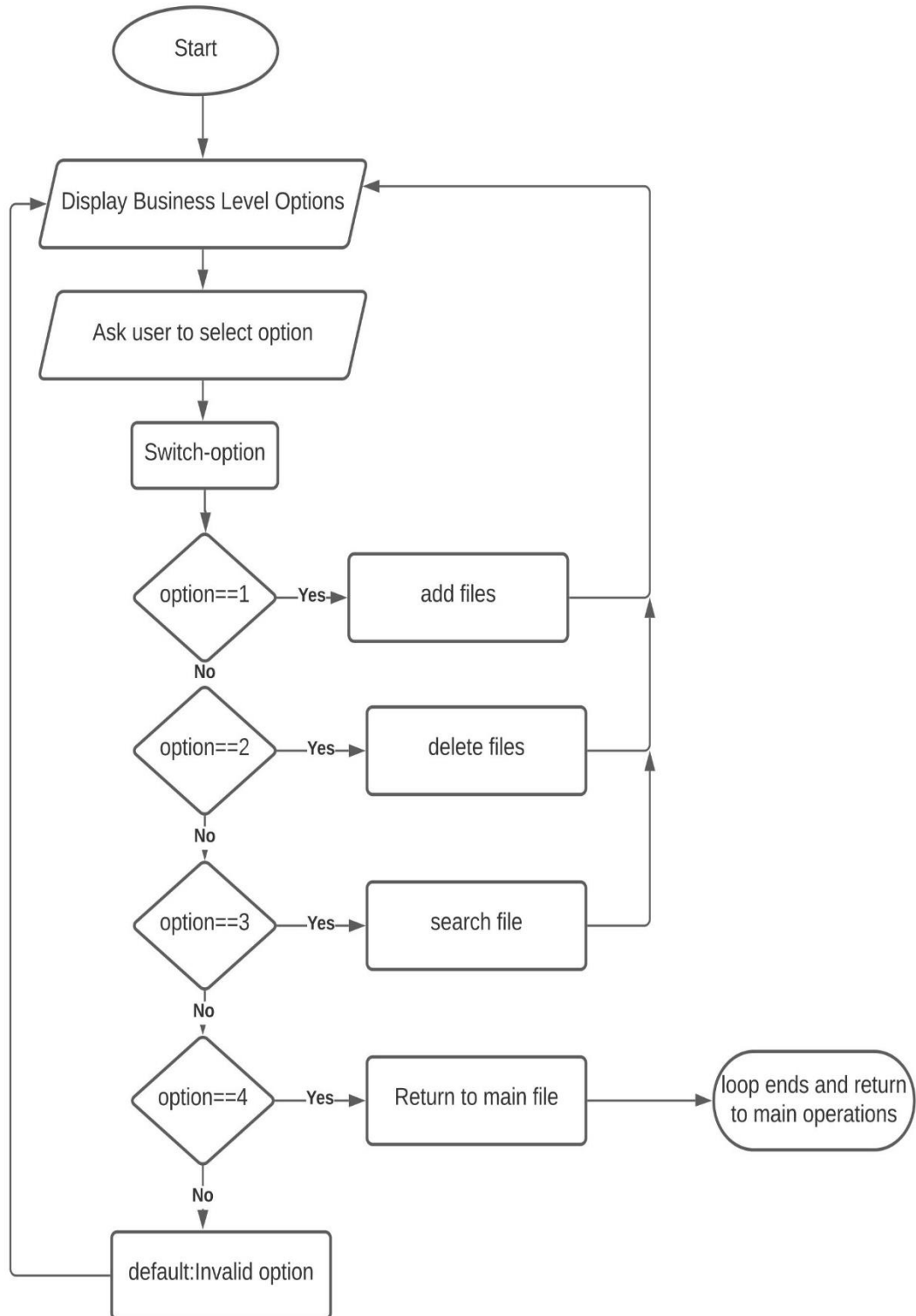
```
Arrays.sort(list, String.CASE_INSENSITIVE_ORDER);
```

CASE_INSENSITIVE_ORDER is used to display in ascending order irrespective of case.

Main Flow Chart



Business Level Operations Flow Chart



Core concepts used in the project

Following are the list of core concepts used in the project

1. Classes
2. Constructors and objects
3. Data Structures and Algorithms – ArrayList (Generics)
4. Package
5. File Handling
6. Exception Handling
7. Code Modularity
8. String Handling
9. Scanner for input handling

Overview of project workflow:

Whole project is included in a package titled “com.virtualKeyRepository”

Project functionality is divided into different files as follows:

- Main.java – Consist of welcome screen and basic operations as follows:
 - Retrieving the file names in an ascending order
 - Business-level operations:
 - Option to close the application
- displayFiles.java – Used to retrieve file names in ascending order
 - java.io.File – Package is used for file handling
 - java.util.Arrays- Package is used for sorting file name in ascending order.
 - If the folder is empty it displays “Folder empty message”.
- businesslevelOperations.java- Displays following business level operations:
 - Option to add a user specified file to the application
 - Option to delete a user specified file from the application
 - Option to search a user specified file from the application
 - Navigation option to close the current execution context and return to the main context.

- addFiles.java- Add files to a folder

java.util.Scanner- Package used to read filename from user

java.io.IOException- Package used to handle exceptions

java.io.File –Package is used for file handling

If file already exists it displays “File exists message”.

- deleteFiles.java- Delete a user specified file from the application

java.util.Scanner- Package used to read filename from user

java.io.File –Package is used for file handling

If file not found displays message “Deletion Failed-File Not Found”

- searchFile.java - search a user specified file from the application.

java.io.File –Package is used for file handling.

If file not found displays message “File not found”.

Links to the GitHub repository to verify the project completion

https://github.com/Rupasi-Ragula/VKR_version1

Conclusion

Enhancements

- The application can be further enhanced with following features:
- Write option to add data to a file
- Append option to modify existing data in a file.
- Since move and copy are frequently used operations these must be included in the application.

USP (Unique Selling Points)

- Application handles all types of exceptions in an interactive.
- Each time user has the option to choose the current working directory. Current working directory may be new one or existing directory.
- Navigation from one menu to another is easy.