# Virtual Key for Your Repositories

## Table of Contents

| S.no | | Page.No |
|---|---|---|
| 1 | Project and developer details | 1 |
| 2 | Sprint planning and task completion | 1 |
| 3 | Core concepts used in projects | 2 |
| 4 | Flowcharts of the application | 2 |
| 5 | Demonstrating the product capabilities,appearance and user interactions | 3 |
| 6 | Links to the github repository to verify project completion | 15 |
| 7 | Conclusion | 17 |

The code of the project is hosted at https://github.com/Megantha/JAVA-CODES/tree/main/Virtual%20Key%20Repository
The project developed by Megantha P.

**Sprints planning and Task completion**
The project is planned to be completed in 1 sprint.Tasks assumed to be completed in sprint are:

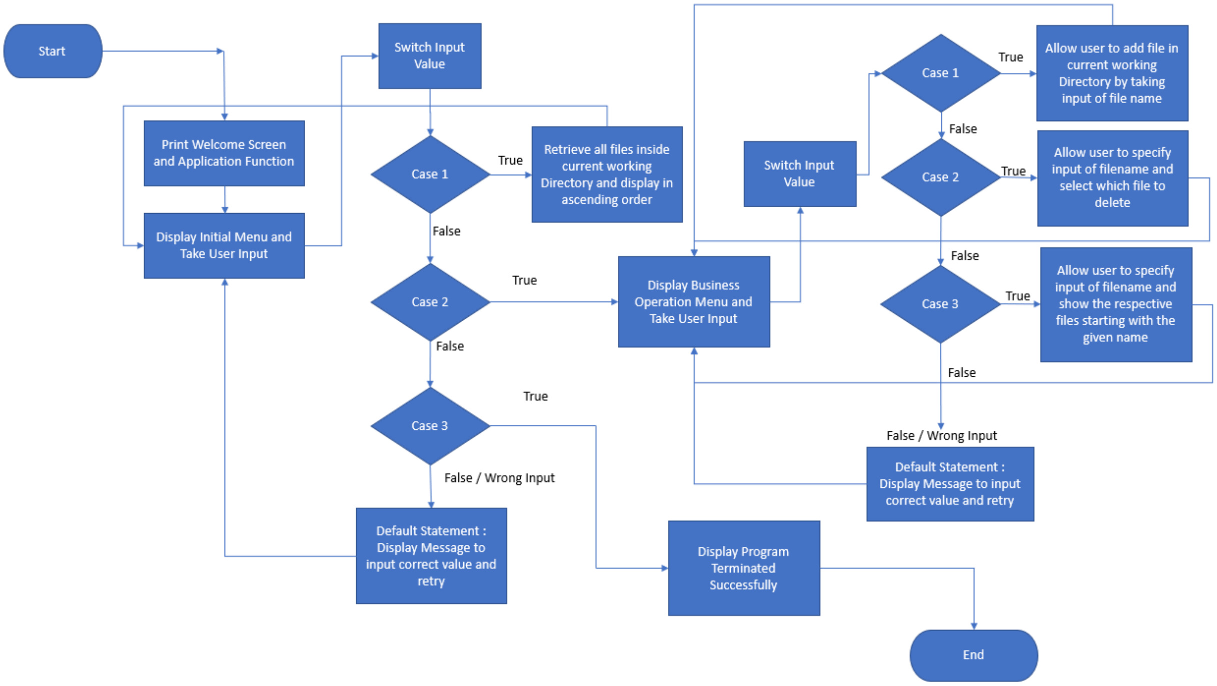
- Creating flow for the application.
- Initializing the git repository to track changes as development progresses.
- Writing the java program to fulfill the requirements of the project.
- Testing the java program with different kinds of user input.

- Pushing code to github
- Creating this specification document highlights application capabilities,appearance, and user interactions.

### Core concepts used in project

Collection framework, File handling, Sorting, Flow control, Recursion, exception Handling, Streams API.

### Flowcharts of the Application:


Start
Print Welcome Screen and Application Function
Display Initial Menu and Take User Input
Switch Input Value
Case 1
True
Retrieve all files inside current working Directory and display in ascending order
False
Case 2
True
False
Case 3
True
False / Wrong Input
Default Statement : Display Message to input correct value and retry
Display Business Operation Menu and Take User Input
Switch Input Value
Case 1
True
Allow user to add file in current working Directory by taking input of file name
False
Case 2
True
Allow user to specify input of filename and select which file to delete
False
Case 3
True
Allow user to specify input of filename and show the respective files starting with the given name
False
False / Wrong Input
Default Statement : Display Message to input correct value and retry
Display Program Terminated Successfully
End

**Demonstrating the product capabilities,appearance and user interactions :**
To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

1.Creating a project in eclipse.

2. Writing a program in java for the entry point of the application(Virtualkey.java)

3.Writing a program in java to perform the file operations as specified by(BusinessOperations.java)

4.Pushing the code to Github repository.

**Step 1: Creating a new project in Eclipse**

- Open Eclipse.
- Go to file →New→Project→Java Project→Next.
- Type in any Project name and click on Finish.
- Select your project and go to file → New→ Class.
- Enter **Virtualkey** in any class name, check the checkbox "public static void main(String[] args)" and click on "Finish".

**Step 2: Writing a program in java to display options available for the user(VirtualKey.java)**

- Select your project and go to File → New → Class.
- Enter MenuOptions in the class name and click on "Finish."

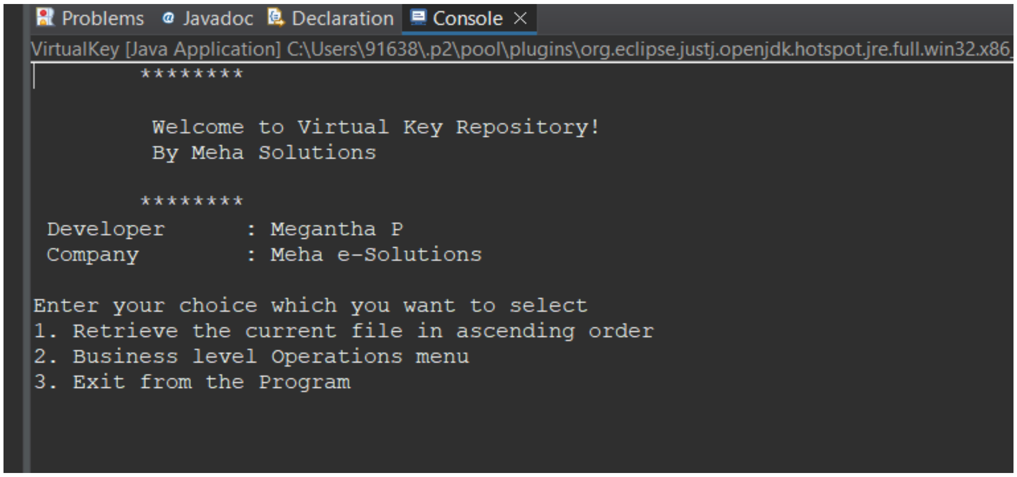
Menu options consists of methods for:

2.1 Displaying Welcome Screen.
2.2 Displaying Initial Menu.
2.3 Displaying the Secondary menu for File Operations available.

**Step 2.1 Writing method to display Welcome to VirtualKey Repository.**

```
package Phase1FinalProject;

import java.util.Scanner;

public class VirtualKey {

    public static void main(String[] args) {
        int a=0,choice=0;

        Scanner scanner=new Scanner(System.in);

        System.out.println("\t********\n");
        System.out.println("\t Welcome to Virtual Key Repository! ");
        System.out.println("\t By Meha Solutions\n");
        System.out.println("\t********");
        System.out.println(" Developer\t: Megantha P\n Company\t: Meha e-Solutions \n");

        while(true)
        {
            System.out.println("Enter your choice which you want to select");
            System.out.println("1. Retrieve the current file in ascending order");
            System.out.println("2. Business level Operations menu");
            System.out.println("3. Exit from the Program");
            try{
                a = scanner.nextInt();
            }
            catch(Exception e)
     {
     System.out.println("Null Exception occurred");
     }
            switch(a)
            {
            case 1: //List function feature to list all files in ascending order.
                BusinessOperations.listFiles();
                break;
            case 2:

                    System.out.println("Please choose one of the
following options :");
                    System.out.println("1. Add a File to the
directory");
                    System.out.println("2. Delete a File from
directory");
                    System.out.println("3. Search for a File");
                    try{
                         choice = scanner.nextInt();
                    }
                    catch(Exception e)
           {
            System.out.println("Null Exception occurred");
           }
                    switch(choice)
                    {
                    case 1:

                        System.out.println("Enter the name of a file
to be created: ");

                        String fileCreate = scanner.next();
                        BusinessOperations.createFile(fileCreate);
                        break;

                    case 2:
                        System.out.print("Enter the name of a file to
be deleted: ");
                        String fileDelete = scanner.next();
                        BusinessOperations.deleteFile(fileDelete);
                        break;

                    case 3:

                        System.out.println("Enter the name of a file
to be searched: ");
                        String fileSearch = scanner.next();
                        BusinessOperations.searchFile(fileSearch);
                        break;

                default:

                        System.out.println("\n Opps! Invalid
Input,Try again\n");
                        break;
                }

                    break;
            case 3:


                scanner.close();
                System.out.println("\n Thank for using this application.It
was a nice experience with you here! See you again. Good bye... and take care");
                System.exit(1);
                break;

            default:

                System.out.println("\n\n Opps! Invalid Input, Please
select the range between 1-3\n");
```

```
                        break;

                    }

                }
        }
}
```



```
VirtualKey [Java Application] C:\Users\91638\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_
        ********

         Welcome to Virtual Key Repository!
         By Meha Solutions

        ********
 Developer      : Megantha P
 Company        : Meha e-Solutions

Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
```

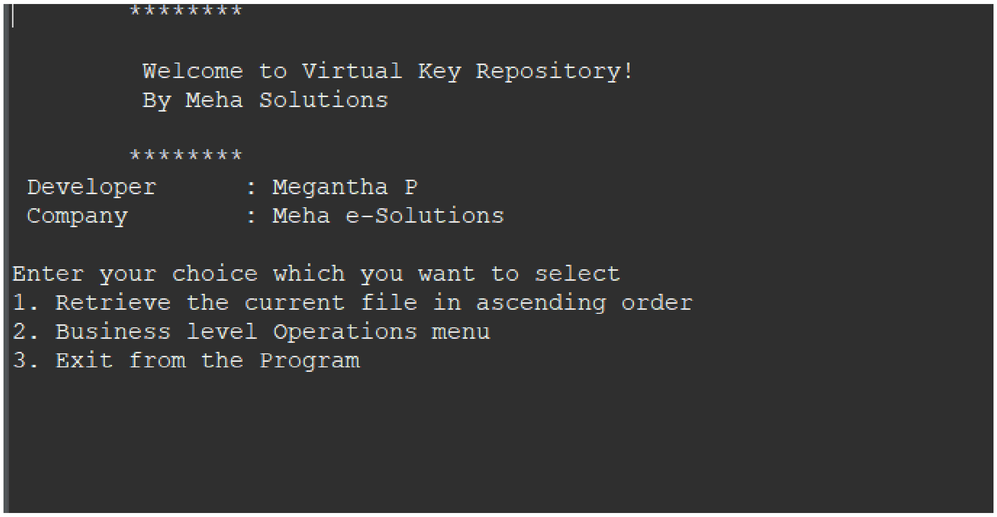
## Step 2.2 Writing method to display Initial menu

Now created the program to choose any one of the options like

- Retrieve current files in ascending order
- Business level operations menu
- Exit from the application

```
while(true)
        {
                System.out.println("Enter your choice which you want to select");
```

```
                    System.out.println("1. Retrieve the current file in ascending
order");
                    System.out.println("2. Business level Operations menu");
                    System.out.println("3. Exit from the Program");
                    try{
                            a = scanner.nextInt();
                    }
                    catch(Exception e)
            {
             System.out.println("Null Exception occurred");
            }
```

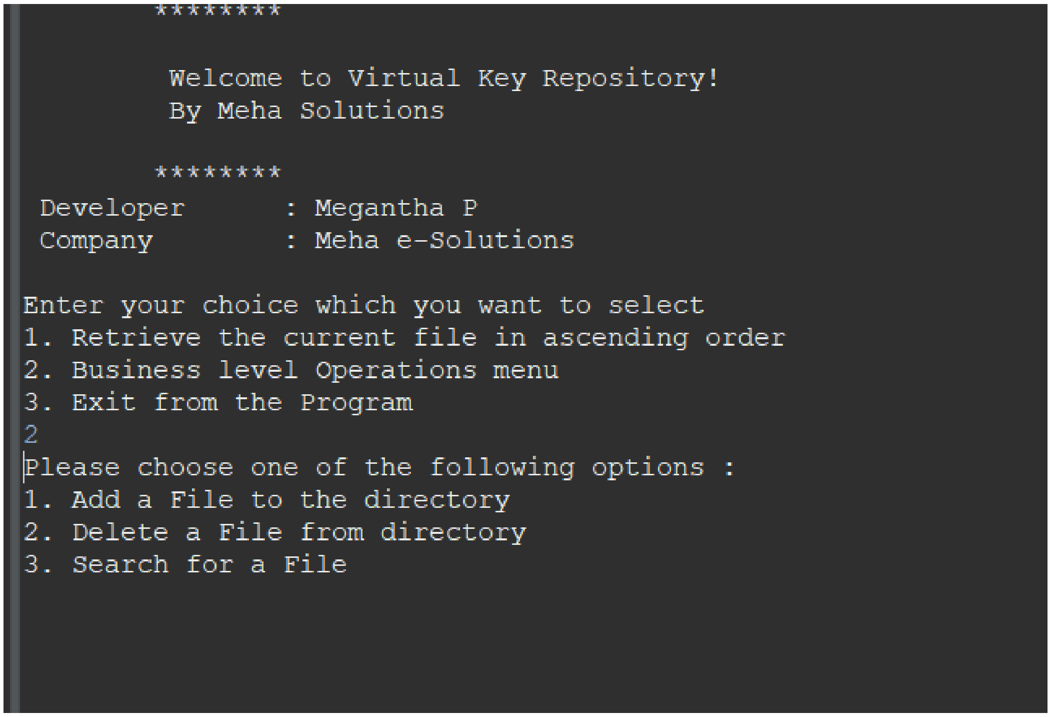


```
        ********

         Welcome to Virtual Key Repository!
         By Meha Solutions

        ********
 Developer      : Megantha P
 Company        : Meha e-Solutions

Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
```

### Step 2.3 Writing method to display Secondary menu for File Operations

```
switch(a)
{
        case 1: //List function feature to list all files in ascending order.
        BusinessOperations.listFiles();
        break;
        case 2:
```

```
System.out.println("Please choose one of the following options :");
System.out.println("1. Add a File to the directory");
System.out.println("2. Delete a File from directory");
System.out.println("3. Search for a File");
try{
        choice = scanner.nextInt();
    }
        catch(Exception e)
          {
                System.out.println("Null Exception occurred");
          }
```



```
         ********

          Welcome to Virtual Key Repository!
          By Meha Solutions

         ********
 Developer      : Megantha P
 Company        : Meha e-Solutions

Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
2
Please choose one of the following options :
1. Add a File to the directory
2. Delete a File from directory
3. Search for a File
```

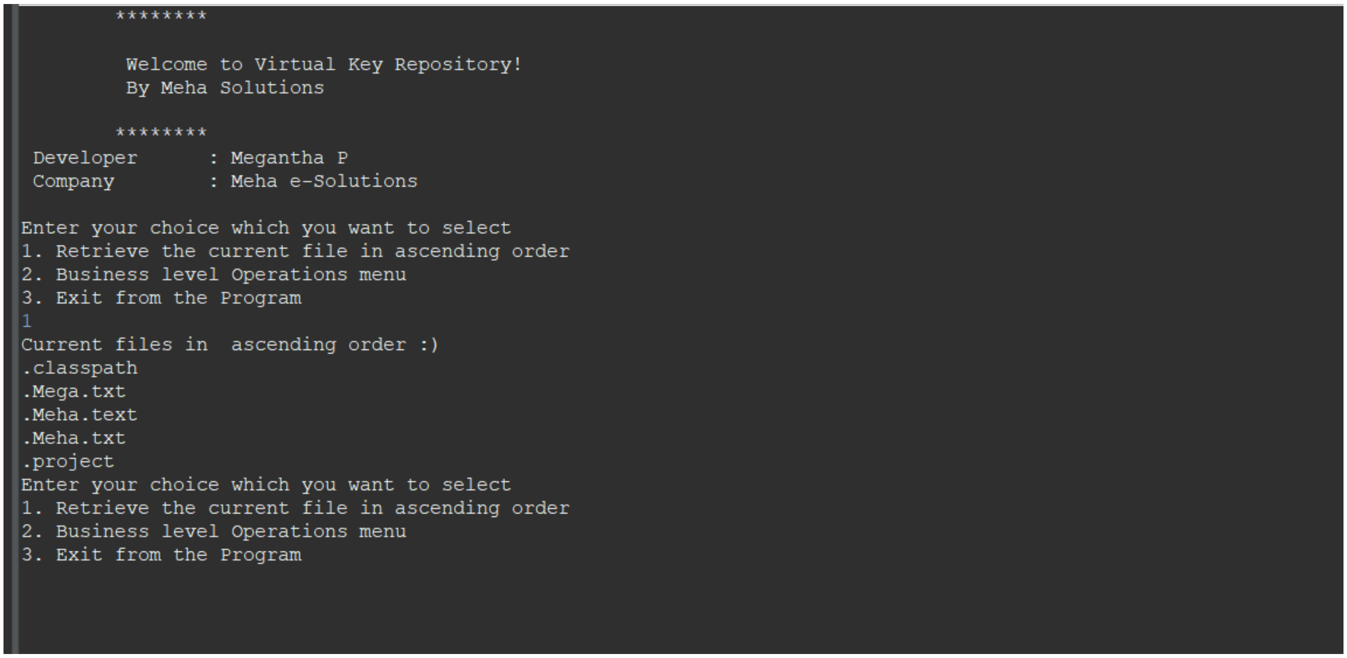
## Step 3: Writing a program in java to handle Menu options selected by user(BusinessOperations.java)

- Select your project and go to File→New→Class.
- Enter BusinessOperations in the class name and click on “Finish”
- BusinessOperations consists of a method for:
  3.1 Retrieve current files in ascending order.
  3.2 To add a file is created.
  3.3 To delete a file is created.
  3.4 To search a file is also created.

## Step 3.1 Writing a method to sort the files in ascending order.

```
public class BusinessOperations {
        protected static String[] sort_sub(String array[], int size){
                String temp = "";
                for(int i=0; i<size; i++){
                        for(int j=1; j<(size-i); j++){
                                if(array[j-1].compareToIgnoreCase(array[j])>0){
                                        temp = array[j-1];
                                        array[j-1]=array[j];
                                        array[j]=temp;
                                }
                        }
                }
                return array;
        }



        //Listing the file
        protected static void listFiles() {
```

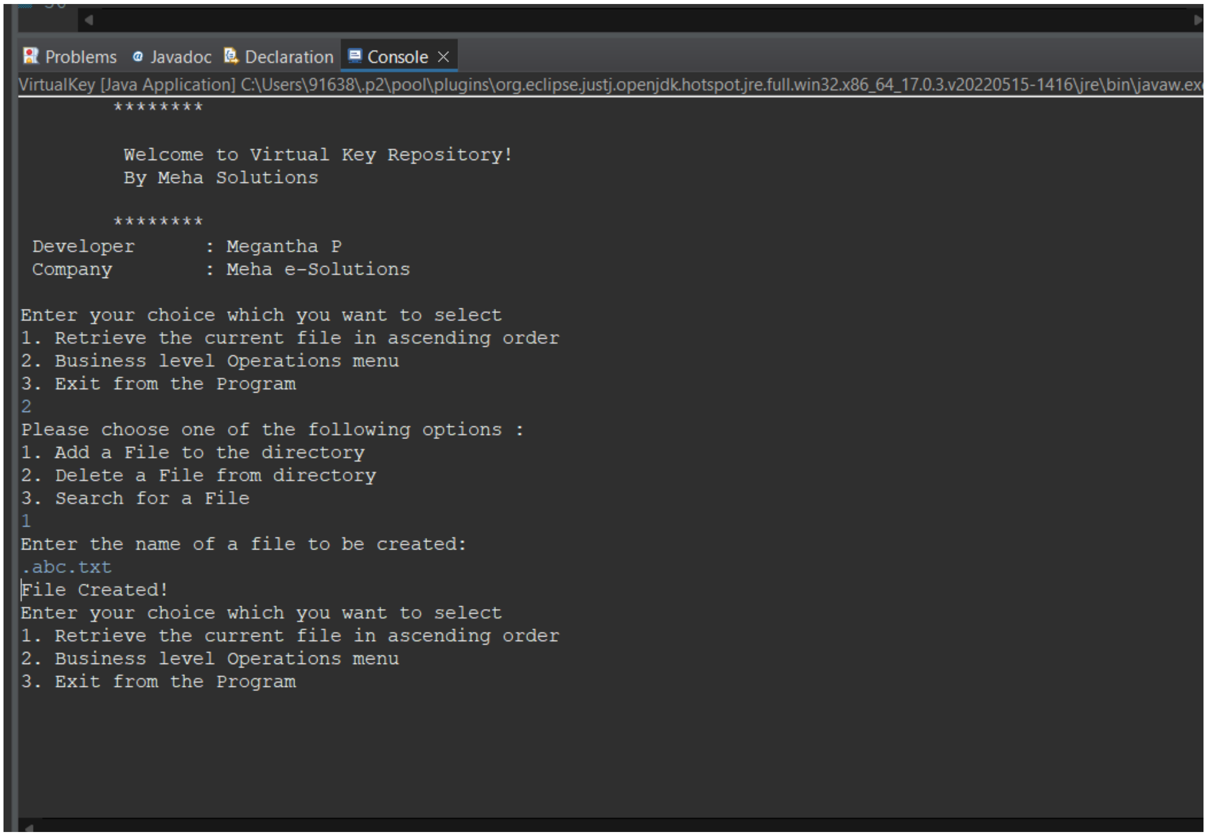
```
        int fileCount = 0;
    ArrayList<String> filenames = new ArrayList<String>();

        File directoryPath = new File(System.getProperty("user.dir"));
        File[] listOfFiles = directoryPath.listFiles();
        fileCount = listOfFiles.length;


        System.out.println("Current files in  ascending order :) ");
        for (int i = 0; i < fileCount; i++) {
         if (listOfFiles[i].isFile()) {
          filenames.add(listOfFiles[i].getName());
         }
        }

        String[] str = new String[filenames.size()];

    for (int i = 0; i < filenames.size(); i++) {
      str[i] = filenames.get(i);
    }

    String[] sorted_filenames = sort_sub(str, str.length);

        for(String currentFile: sorted_filenames) {
                System.out.println(currentFile);
        }

}
```



```
        ********

         Welcome to Virtual Key Repository!
         By Meha Solutions

        ********
 Developer      : Megantha P
 Company        : Meha e-Solutions

Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
1
Current files in  ascending order :)
.classpath
.Mega.txt
.Meha.text
.Meha.txt
.project
Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
```

**Step 3.2 Writing a method to create a file.**

```
//File creation
                protected static void createFile (String fileToBeCreated) {
                        File file = new File( (System.getProperty("user.dir") ) + "\\" + fileToBeCreated );

                        try {
                                if (file.createNewFile() ) {
                                        System.out.println("File Created!");
                                } else {
                                        System.out.println("File already exists :(");
                                }
                        } catch (IOException e) {

                                e.printStackTrace();
                        }
                }
```

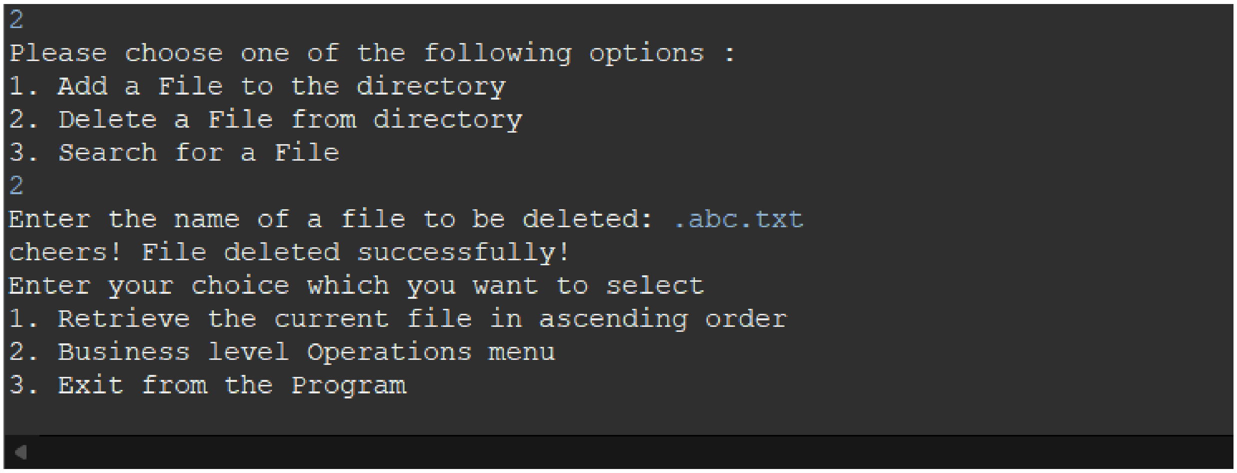


```
VirtualKey [Java Application] C:\Users\91638\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.ex
        ********


         Welcome to Virtual Key Repository!
         By Meha Solutions

        ********
 Developer      : Megantha P
 Company        : Meha e-Solutions

Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
2
Please choose one of the following options :
1. Add a File to the directory
2. Delete a File from directory
3. Search for a File
1
Enter the name of a file to be created:
.abc.txt
File Created!
Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
```

### Step 3.3 Writing method to delete the files

```
	//File deletion
	protected static void deleteFile(String fileToBeDeleted) {

		File file = new File( (System.getProperty("user.dir") ) + "\\" +
fileToBeDeleted );

		if(file.exists()) {
			if ( file.delete() ) {
				System.out.println("cheers! File deleted successfully!");
			}
		} else {
			System.out.println("Sorry, Your file wasn't deleted (File Not
Found)");
```

```
                }
        }
```

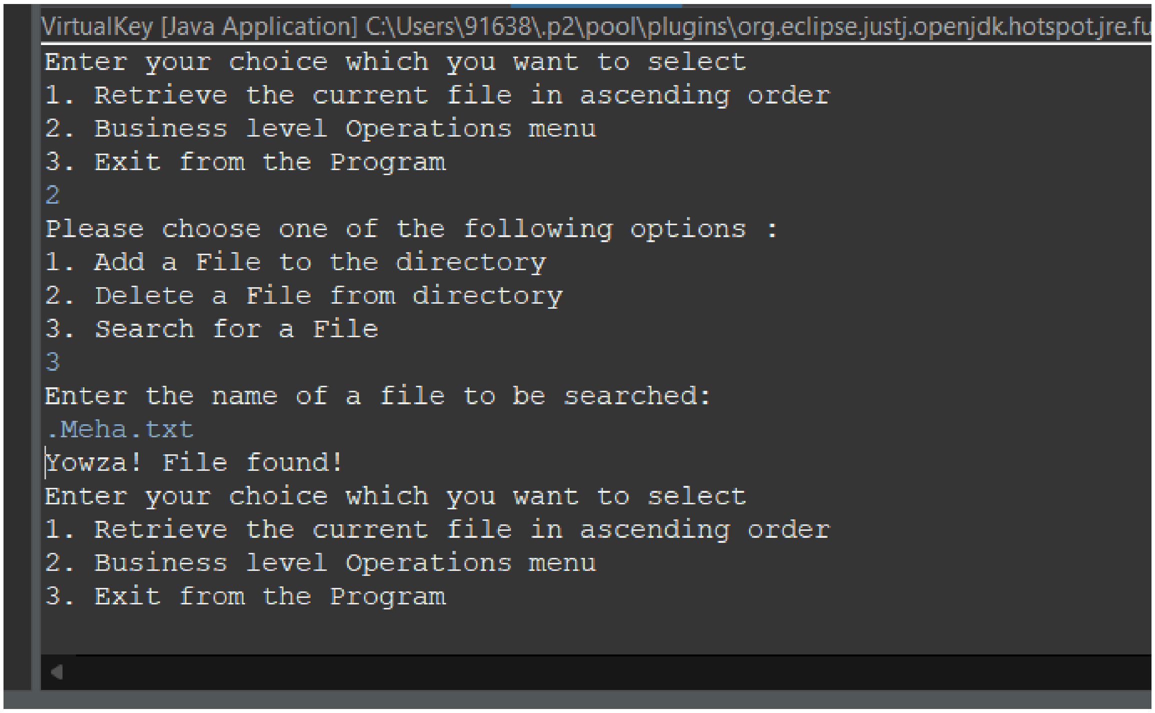


```
2
Please choose one of the following options :
1. Add a File to the directory
2. Delete a File from directory
3. Search for a File
2
Enter the name of a file to be deleted: .abc.txt
cheers! File deleted successfully!
Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
```

### Step 3.4 Writing a method to search the files

```
//File searching
        protected static void searchFile(String fileToBeSearched) {

                File file = new File( (System.getProperty("user.dir") ) + "\\" +
fileToBeSearched );

                //Check whether file whether file exists or not.
                //If yes then display associated message
                if(file.exists()) {
                        System.out.println("Yowza! File found!");
                } else {
                        System.out.println("Ohh, Your file is not there (File Not
Found)");
                }       PrintWriter pw;
      try {
         pw = new PrintWriter(fileToBeSearched);
         pw.println("saved");
      }
```

```
    // providing the checked exception handler
    catch (FileNotFoundException e) {

        System.out.println(e);
    }
```



VirtualKey [Java Application] C:\Users\91638\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu

```
Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
2
Please choose one of the following options :
1. Add a File to the directory
2. Delete a File from directory
3. Search for a File
3
Enter the name of a file to be searched:
.Meha.txt
Yowza! File found!
Enter your choice which you want to select
1. Retrieve the current file in ascending order
2. Business level Operations menu
3. Exit from the Program
```

### Step 4: Pushing the code to GitHub repository

- Open your command prompt and navigate to the folder where you have created your files.
  **cd<folder path>**
- Initialize repository using the following command:
  **git init**
- Add all the files to your git repository using the following command:
  **git add .**
- Commit the changes using the following command:
  **git commit . -m <commit message>**

- Push the files to the folder you initially created using the following command:
  **git push -u origin master**

**Github link:https://github.com/Megantha/JAVA-CODES/tree/main/Virtual%20Key%20Repository**

**Unique Selling Points of the Application**

1. The application is designed to keep on running and taking user inputs even after exceptions occur. To terminate the application, appropriate option needs to be selected.
2. The application can take any file/folder name as input. Even if the user wants to create nested folder structure, user can specify the relative path, and the application takes care of creating the required folder structure
3. User is also provided the option to write content if they want into the newly created file.
4. The application doesn't restrict user to specify the exact filename to search/delete file/folder. They can specify the starting input, and the program searches all files/folder starting with the value and displays it. The user is then provided the option to select all files or to select a specific index to delete
5. The application also allows user to delete folders which are not empty.
6. The user is able to seamlessly switch between options or return to previous menu even after any required operation like adding, searching, deleting or retrieving of files is performed.
7. When the option to retrieve files in ascending order is selected, user is displayed with two options of viewing the files.
   7.1. Ascending order of folders first which have files sorted in them,
   7.2. Ascending order of all files and folders inside the "main" folder
8. The application is designed with modularity in mind. Even if one wants to update the path, they can change it through the source code. Application has been developed keeping in mind that there should be very less "hardcoding" of data.

## Conclusions

Further enhancements to the application can be made which may include:

- Conditions to check if user is allowed to delete the file or add the file at the specific locations.
- Asking user to verify if they really want to delete the selected directory if it’s not empty.
- Retrieving files/folders by different criteria like Last Modified, Type, etc.
- Allowing user to append data to the file