App.java

```
package com;

import java.io.IOException;
import java.util.InputMismatchException;
import java.util.Scanner;

public class App {

	App() {
		DevInfo info = new DevInfo();

		info.displayInfo();

	}

	public static void main(String args[]) throws IOException {

		int option,subOption;
		String file;


		@SuppressWarnings("unused")
		App app = new App();
		Options options = new Options();

		System.out.println("Enter a the folder");
		Scanner scanner = new Scanner(System.in);
		try {
			String path = scanner.nextLine();

			while (true) {
				options.displayOptions();
				System.out.println("Select an Option");
				option = scanner.nextInt();
				scanner.nextLine();

				switch (option) {
				case 1:
					options.displayFileNames(path);
					break;

				case 2:
```

```
		loop:while(true){
			System.out.println("Select a sub-catogery");
			subOption = scanner.nextInt();
			scanner.nextLine();

		switch (subOption) {

		case 1:
			System.out.println("Enter a file name to add:\n");
			file = scanner.nextLine();
			options.addFile(path, file);
			break;

		case 2:
			System.out.println("Enter a file name to delete");

			if(scanner.hasNextLine()) {
				file = scanner.nextLine();
				options.deleteFile(path, file);
			}
			break;

		case 3:
			System.out.println("Enter a file name to search:\n");
			file = scanner.nextLine();
			options.searchFile(path, file);
			break;
		case 4:
			 System.out.println("Closed sub-menu");
	break loop;

	default:
	   System.out.println("Please select a valid sub-category");
			break;
		}}break;
	case 3:
		options.closeApplication();
		System.out.println("Application Terminated");
		break;
	default:
		System.out.println("Please select a valid option");
```

```
                        }

                    }
                }
                catch(InputMismatchException e) {
                    System.out.println("Invalid input. Please enter a valid option number.");
            scanner.nextLine();

                }
                scanner.close();
            }}
```

BinarySearch.java

```
package com;

public class BinarySearch {
    public void Search(String arr[],int begin,int end, String fileName) {
        int mid = (begin+end)/2;

        while(begin<=end) {
            int res = arr[mid].compareToIgnoreCase(fileName);
            if(res<0) {
                begin = mid+1;
            }
            else if(res == 0) {
                System.out.println("File found " + fileName);
                break;
            }
            else {
                end = mid-1;
            }
            mid = (begin+end)/2;
        }

        if(begin>end) {
            System.out.println("File not found");
        }
    }
}
```

DevInfo.java

```
package com;
```

```
public class DevInfo {
	public String applicationName = "LockedMe.com";
	public String developerName = "Abrar";

	public String getApplicationName() {
		return applicationName;
	}

	public void setApplicationName(String applicationName) {
		this.applicationName = applicationName;
	}


	public void setDeveloperName(String developerName) {
		this.developerName = developerName;
	}

	void displayInfo() {
		System.out.println("---------------------------------------");
		System.out.println(applicationName + "\n" + developerName);
		System.out.println("---------------------------------------");
	}


}
```

Options.java

```
package com;

import java.io.File;
import java.io.IOException;


public class Options {

	public void displayOptions() {

		System.out.println("---------------------------------------" + "\n1.List all the files in the
directory."
				+ "\n2.Perform file operations\n" + "\t1.Add a file\t" + "\t2.Delete a
file" + " \t 3.Search a file"+" \t 3.Main menu"
				+ "\n3.Close the application\n" + "---------------------------------------");
```

```
    }


    public void displayFileNames(String path) {
        File folder = new File(path);

        if (folder.isDirectory()) {
            String[] files = folder.list();

            QuickSort quickSort = new QuickSort();
            quickSort.Sort(files, 0, files.length - 1);

            System.out.println("---------------------------------------");
            System.out.println("Sorting by filename in ascending order");

            for (String file : files) {
                System.out.println(file);
            }
            System.out.println("---------------------------------------");
        } else {
            System.out.println(folder.getAbsolutePath() + " is not a directory");
        }
    }

    public void deleteFile(String path, String filename) {

                String filePath= path + File.separator + filename;
                File file = new File(filePath);
                if(file.exists()) {
                if (file.delete()) {
                        System.out.println("Deleted file scuccessfully!");

                } else {
    System.out.println("Failed to delete the file.");
}
                }else {System.out.println("File not found");}
                }




    public void searchFile(String path, String file) {
                File folder = new File(path);
                BinarySearch Binary = new BinarySearch();
```

```
                String[] files = folder.list();
                Binary.Search(files,0, files.length, file);
        }

        public void closeApplication() {
                System.out.println("---------------------------------------" + "Application closed" +
"---------------------------------------");
                System.exit(0);
        }

        public void addFile(String path, String file) throws IOException {
                try {
                        File newFile = new File(path + "\\" + file);
                        if ((newFile.createNewFile())) {
                                System.out.println("File created: " + newFile.getName());
                        } else {
                                System.out.println("File already exists.");
                        }
                } catch (IOException e) {
                        System.out.println("An error occurred.");
                        e.printStackTrace();
                }

        }
}
```

QuickSort.java

```
package com;

public class QuickSort {
        public void Sort(String[] arr, int begin, int end) {
                if(begin<end) {
                        int partitionIndex = partition(arr,begin,end);
                        Sort(arr,begin,partitionIndex-1);
                        Sort(arr,partitionIndex+1,end);
                }
        }

        private int partition(String[] arr, int begin, int end) {
                String pivot = arr[end];
                int i = (begin-1);

                for(int j=begin;j<end;j++ ) {
```

```
                int result = arr[j].compareTo(pivot);

                if(result <= 0) {
                        i++;
                        String swapTemp = arr[i];
                        arr[i]=arr[j];
                        arr[j]=swapTemp;
                }
        }
        String swapTemp = arr[i+1];
        arr[i+1] = arr[end];
        arr[end] = swapTemp;
        return i+1;
    }


}
```