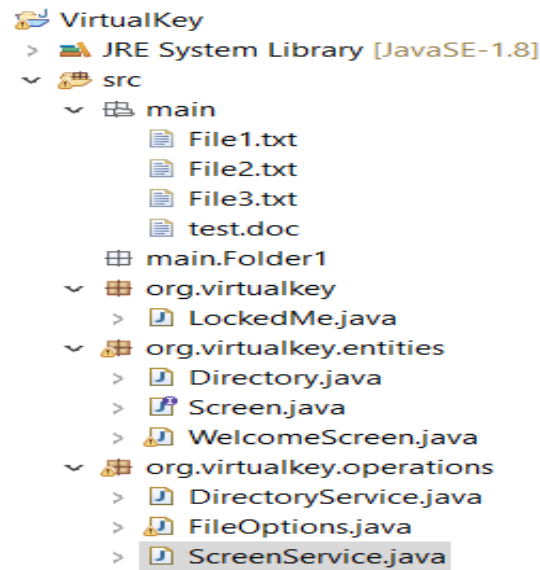


Project File Structure:



LockedMe.java

```
package org.virtualkey;

import org.virtualkey.entities.WelcomeScreen;

public class LockedMe {

    public static void main(String[] args) {

        WelcomeScreen welcome = new WelcomeScreen();
        welcome.welcomeScreen();
        welcome.GetUserInput();
    }

}
```

Directory.java

```
package org.virtualkey.entities;
import java.util.ArrayList;
import java.util.Collections;
import java.io.File;
import java.nio.file.FileSystems;
import java.nio.file.Path;

public class Directory {

    public static final String name = "src/main/";

    private ArrayList<File> files = new ArrayList<File>();
```

```

Path path = FileSystems.getDefault().getPath(name).toAbsolutePath();

File Dfiles = pathToFile();

public String getName() {
    return name;
}

public void print() {
    System.out.println("Existing Files: ");
    files.forEach(f -> System.out.println(f));
}

public ArrayList<File> fillFiles() {

    File[] directoryFiles = Dfiles.listFiles();
    files.clear();
    if(directoryFiles != null) {
        for (int i = 0; i < directoryFiles.length; i++) {
            if (directoryFiles[i].isFile()) {
                files.add(directoryFiles[i]);
            }
        }
        Collections.sort(files);
    }
    return files;
}

```

WelcomeScreen.java

```

package org.virtualkey.entities;

import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;

import org.virtualkey.operations.DirectoryService;
import org.virtualkey.operations.ScreenService;

public class WelcomeScreen implements Screen {
    private String welcomeText = "Welcome to LockedMe: A Virtual Key";
    private String developerText = "Developed By: BIBHU RANJAN MOHANTY";
    private ArrayList<String> options = new ArrayList<>();

    public WelcomeScreen() {
        options.add ("1. Display the current file names in ascending
order.");
        options.add ("2. Manage Files/Folders");
        options.add ("3. Quit");
    }
}

```

```

    }

    public void welcomeScreen() {
        System.out.println(welcomeText);
        System.out.println(developerText);
        System.out.println("\n");
        Show();
    }

    public void Show() {
        System.out.println("Main Menu");
        for (String s : options) {
            System.out.println(s);
        }
    }

    public void GetUserInput() {
        int selectedOption = 0;

        while (true) {
            selectedOption = this.getOption();
            if(selectedOption == 3) {
                System.out.println ("Program Terminated Successfully.");
                break;
            }
            this.NavigateOption(selectedOption);
        }
    }

    public void NavigateOption(int option) {
        switch(option) {
            case 1: //Show Files in the current Directory
                this.ShowFiles();
                this.Show();
                break;

            case 2: // Show File Options menu

ScreenService.setCurrentScreen(ScreenService.FileOptionsScreen);
                ScreenService.getCurrentScreen().Show();
                ScreenService.getCurrentScreen().GetUserInput();
                this.Show();
                break;
            default:
                System.out.println("Invalid Option");
                break;
        }
    }

    public void ShowFiles() {

        // Get the files from the Directory
        System.out.println("List of Files: ");
        DirectoryService.PrintFiles();
    }

```

```

        private int getOption() {
            Scanner in = new Scanner(System.in);

            int returnOption = 0;
            try {
                returnOption = in.nextInt();
            }
            catch (InputMismatchException ex) {

            }
            return returnOption;
        }
    }
}

```

Screen.java

```

package org.virtualkey.entities;

public interface Screen {
    public void Show();

    public void NavigateOption(int option);

    public void Get userInput();
}

```

DirectoryService.java

```

package org.virtualkey.operations;
import java.io.File;

import org.virtualkey.entities.Directory;
public class DirectoryService {
    private static Directory fileDirectory = new Directory();

    public static void PrintFiles() {

        fileDirectory.fillFiles();

        for (File file : DirectoryService.getFileDirectory().getFiles())
        {
            System.out.println(file.getName());
        }
    }
    public static Directory getFileDirectory() {
        return fileDirectory;
    }
}

```

```

        public static void setFileDirectory(Directory fileDirectory) {
            DirectoryService.fileDirectory = fileDirectory;
        }
    }
}

```

FileOption.java

```

package org.virtualkey.operations;

import java.io.File;
import java.io.IOException;
import java.nio.file.FileSystems;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;

import org.virtualkey.entities.Directory;
import org.virtualkey.entities.Screen;

public class FileOptions implements Screen {
    private Directory dir = new Directory();

    private ArrayList<String> options = new ArrayList<>();

    public FileOptions() {
        options.add("1. Add a File");
        options.add("2. Delete A File");
        options.add("3. Search A File");
        options.add("4. Return to Menu");
    }

    @Override
    public void Show() {
        System.out.println("File Options Menu");
        for (String s : options) {
            System.out.println(s);
        }
    }

    public void GetUserInput() {
        int selectedOption;
        while ((selectedOption = this.getOption()) != 4) {
            this.NavigateOption(selectedOption);
        }
    }
}

```

```

    }
}

@Override
public void NavigateOption(int option) {

    switch(option) {

        case 1: // Add File
            this.AddFile();
            this.Show();
            break;
        case 2: // Delete File
            this.DeleteFile();
            this.Show();
            break;
        case 3: // Search File
            this.SearchFile();
            this.Show();
            break;

        default:
            System.out.println("Invalid Option");
            break;

    }
}

public void AddFile() {
    System.out.println("Please Enter the Filename:");

    String fileName = this.getInputString();

    System.out.println("You are adding a file named: " + fileName);

    try {
        Path path = FileSystems.getDefault().getPath(Directory.name +
fileName).toAbsolutePath();
        File file = new File(dir.getName() + fileName);

        if (file.createNewFile()) {
            System.out.println("File created: " + file.getName());
            dir.GetFiles().add(file);
        } else {
            System.out.println("This File Already Exits, no need to add
another");
        }
    } catch (IOException e){
        System.out.println(e);
    }
}

public void DeleteFile() {

    System.out.println("Please Enter the Filename:");

```

```

        String fileName = this.getInputString();

        System.out.println("You are deleting a file named: " + fileName);
        Path path = FileSystems.getDefault().getPath(Directory.name +
fileName).toAbsolutePath();
        File file = pathToFile();
        if (file.delete()) {
            System.out.println("Deleted File: " + file.getName());
            dir.GetFiles().remove(file);
        } else {
            System.out.println("Failed to delete file:" + fileName + ", file was
not found.");
        }
    }

    public void SearchFile() {

        Boolean found = false;

        System.out.println("Please Enter the Filename:");

        String fileName = this.getInputString();

        System.out.println("You are searching for a file named: " + fileName);

        //TODO Fix it so ArrayList obtains files
        //Finished TODO

        ArrayList<File> files = dir.GetFiles();

        for(int i = 0; i < files.size(); i++) {
            if(files.get(i).getName().startsWith(fileName)) {
                System.out.println("Found " + files.get(i).getName());
                found = true;
            }
        }
        if (found == false) {
            System.out.println("File not found");
        }
    }

    private String getInputString() {

        Scanner in = new Scanner(System.in);
        return(in.nextLine());

    }

    private int getOption() {
        Scanner in = new Scanner(System.in);

        int returnOption = 0;
        try {
            returnOption = in.nextInt();
        }
        catch (InputMismatchException ex) {
            System.out.println("Invalid input");
        }
    }

```

```
        return returnOption;

    }

}
```

ScreenService.java

```
package org.virtualkey.operations;

import org.virtualkey.entities.Screen;
import org.virtualkey.entities.WelcomeScreen;

public class ScreenService {

    public static WelcomeScreen WelcomeScreen = new WelcomeScreen();
    public static FileOptions FileOptionsScreen = new FileOptions();

    public static Screen CurrentScreen = WelcomeScreen;

    public static Screen getCurrentScreen() {
        return CurrentScreen;
    }
    public static void setCurrentScreen(Screen currentScreen) {
        CurrentScreen = currentScreen;
    }

}
```