**Project Name:** Virtual Key For Repositories

**Developer Name:** N.Vijay

This document contains sections for:

● Sprint planning

● Core concepts used in project

● Flow of the Application

● Project Description

● Unique Selling Points of the Application

● Conclusions

**Sprints planning:**

The project is planned to be completed in 2 sprints. Tasks assumed to be completed in the sprint are:

· Creating the flow of the application

· Initializing 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

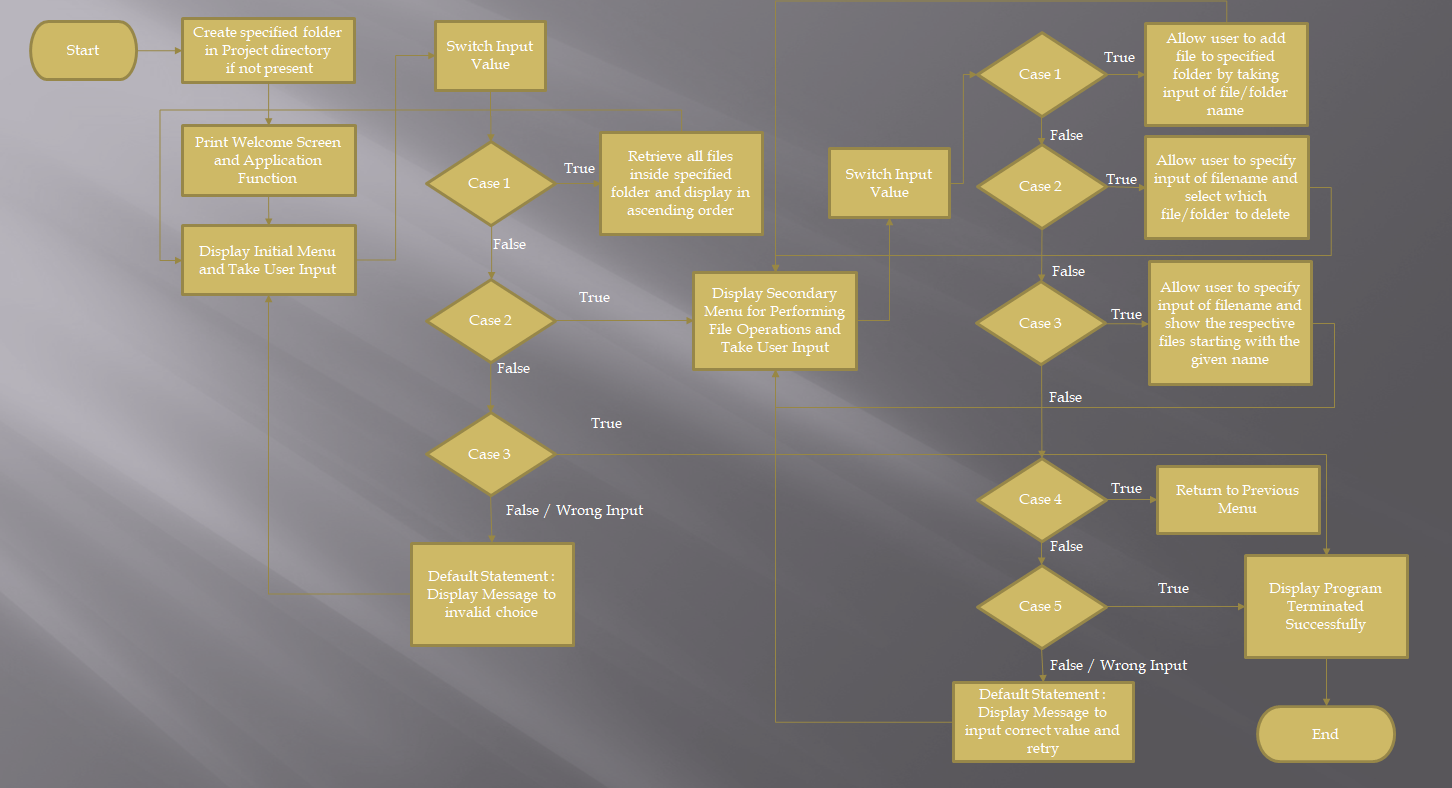
· Pushing code to GitHub.

· Creating this specification document highlighting application capabilities, appearance, and user interactions.

**Core concepts used in project:**

Collections framework, File Handling, Sorting, Flow Control, Recursion, Exception Handling etc…

**Application Flow:**

****

**Project Description:**

In this Project , I developed a prototype of the application. In which it can display name and developer details, The details of the User Interface such as options displaying the User Interactions information and features to accept the user input to select one of the options Listed.

In this project it will display the welcome information and then ask for the user interaction as accessing files in the given folder, Display menu for Business level operation and exit application. In Display menu options we can add, delete, search a file in the folder.

To complete this project the following steps are used:

Step 1: Created a class named Business level operation, in which implemented the following methods.

1. Method to Create a folder if not exist in the given directory.
2. Method to return the list of files that are available in the given folder.
3. Method to create a file and add it to folder in the directory.
4. Method to delete a file from the given directory.
5. Method to search a file in the given directory.

Step 2: Created a class named Appmain in which we developed welcome screen, menu Options by using the following methods:

1. Method to print the welcome screen containing the application name and developer details.
2. Methods to give the list of options as a menu.

Step 3: Created a interface name FileInterface to choose which work we need to do using the switch cases.

**Unique Selling Points of the Application:**

1. The Application is developed to implement to take the inputs from the user even some exceptions happen. To exit from the application, some options 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 need to specify the relative path, and the application takes care of creating the required folder.

3. User is also provided the option to write content if they want into the newly created file.

4. The application also allows user to delete folders which are not empty.

5. 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.

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 specified folder.

8. The application is developed with modularity in mind. Even if we wants to update the path, they can change it through the source code.

**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 by different criteria like Last Updated, Type, etc.

· Allowing user to add the data to the file.