# Specifications

This command-line based application allows a user to manage the contents of a pre-defined file repository. The application greets a user with a title card and prompts the user for input form a list of functions – list the repository contents, manage files within the repository, or close the application. If the user chooses to manage the files, they are prompted again to add, delete, or search for a file, or to return to the main menu.

These menus accept case-agnostic partial-text inputs, i.e. “s”, “Search”, “SE”, or “search for a file” will all successfully bring up the searching function. For the simple functions – search, return to main menu, manage files, list contents, and close application – the output is immediate. For the more complicated functions – add and delete – the user is prompted for confirmation or if they would like to retry the operation if the task failed in the case of user error.

All errors are handled and the cause is relayed to the end user, most commonly the issue is a mis-input or filename mismatch.

# Sprint Planning

Sprint 1 – 1 week

Develop flowchart for application and functions based on user specifications

Write pseudocode for primary menu

Title card

Display menu

Input handling and logic

List contents

Close Application

Manage Files

Write pseudocode for secondary menu

Display menu

Input handling and logic

Add file

Delete File

Search for a file

Return to main menu

Code minimum prototype

(See previous steps in sprint 1 for individual code tasks)

Sprint 2 – 1 week

Implement OOP

Parent class for menus

Include Scanner and File object

Input handling code

Subclass for each menu

Unique menu outputs

Unique menu methods

Expand input handling

Partial input

File search

User confirmation

User feedback

Implement error handling

No matches found

I/O issues

Filename conflicts

Partial filename matches

Sprint 3 – 1 week

Test application and apply bugfixes

Null inputs

# [Github Repository](https://github.com/braxdorn/vk-repo) (hyperlink)

Java Concepts

Parent class and inheritance for two menu loops (one for outer, one for inner)

Common methods in parent class

Common method types abstracted in parent

Common attributes declared in parent class

Common constructor for both children

Child classes both implement abstracted parent methods

Child classes both have unique methods separate from parent and each other

# Data Structures

ArrayList and Collections.sort() are used for sorting the content before listing it to the end user.

ArrayList and matches([regex]) are used to filter the registry content in both Add and Delete.

This simplified error handling for zero or multiple matches.

Also Allowed expanded user inputs (partial match).

# Features and Operations

* Visually appealing title card with Developer information and production date
* Allows a user to complete the following operations:
  + List repository contents
  + Add a file to the repository
  + Delete a file from the repository
  + Locate a file within the repository
  + Close the application
* Organized between high-level and low-level operations
  + General application functions vs file management
* Expanded input handling for intuitive use
* Error handling with user-friendly feedback
  + Confirmation prompts in the case of filename collisions