

Project 1 Report

Our program is a knowledge repository, it functions solely on the command line user interface. The program has a main menu and a few sub menus such as:

```
Welcome to the Knowledge Depository

Please select from the following options:
Press 1 to view information pages.
Press 2 to create a new information page.
Press 3 to change admin password.
Press x to exit.

Please Enter the Administrator Password:
Admin
Enter the name of the file: DeclanDetails
File created: DeclanDetails.txt
Please add your information
My name is Declan and I was born on the 16th of March 2002
Text added to file: DeclanDetails.txt



You have selected 'DeclanDetails.txt' Please select an option.
Press 1 to view information.
Press 2 to edit information.
Press 3 to view edit history.
Press 4 to delete file.
Press b to go back.
Press X to go exit.
```

There is also an administrator lock feature that only allows users with a password (in this program "Admin") to create, edit, or delete information.

Admins can:

- Create new information pages.
- Edit existing information pages.
 - o Find and replace information in a specific page.
 - o Or append new information to a specific.
 - o Delete a specific page. (And its 'EDITS_' file)

Our program works by managing a small database of text files that contain information and their corresponding 'EDITS_' files which contain a record of every time a text file is edited (through the program) You will notice in the relative directory of the Java project; each text file has a corresponding 'EDITS_' file. (DeclansDetails.txt will have an edits file named EDITS_DeclanDetails.txt made at creation, provided the user uses the program through the CLI to create the file)

 DeclanDetails.txt	3/09/2023 2:44 am	Text Document	1 KB
 EDITS_DeclanDetails.txt	3/09/2023 2:44 am	Text Document	0 KB

Every time an edit is made through the program using the CLI, a record is added to the corresponding 'EDITS_' files.

Edits made by Admin at: 02:50:25 03-09-2023

APPENDED DATA:

I like to play video games and Software Contruction is my favourite class!
End of edit.

For planning purposes, we created a few mock up menu screens and built out program upon that.

Main Menu contents:

```
Welcome to the knowledge database! please select from the following options:
Press 1 to view files.
Press 2 to create a text file. (Admin permission required)
Press 3 to delete a text file. (Admin permission required)

Press X to exit the program.
```

A will show a submenu that lists all text files
B will create a text file (*****.txt) AND a EDITS_*****.txt file

```
Choose a file:
Press 1 for: text.txt
Press 2 for: myFile.txt
Press 3 for: bradley.txt

Press B to go back.
Press X to exit the program.
```

```
You have selected "text.txt", would you like to:
Press 1 to view the file.
Press 2 to view the summary. (Stored in metadata)
Press 3 to edit the file. (Admin permission required)

Press B to go back.
Press X to exit the program.
```

```
Please select from the following options:
Press 1 to edit file contents.
Press 2 edit file metadata. (Stored in document class variables)
Press 3 View all changes.
Press 4 to delete the document. (Are you sure screen?)

Press B to go back.
Press X to exit the program.
```

Contribution and program breakdown

Bradley	Declan
GroupProject.java <ul style="list-style-type: none">- Main function of program (loop)- Instance of GroupProject.java stores the Admin password- Instance of AbstractFile is stored in the instance of GroupProject.java.- Using polymorphism and inheritance, the program switches between modes (CreateFile, DisplayContents and EditFile) on request.	EditFile.java (Polymorphic Mode of AbstractFile) <ul style="list-style-type: none">- Extends AbstractFile.java.- Only users with an admin password have access to editing.- EditFile has the functionality of finding/replacing information or appending new information to the file.- EditFile also has to functionality of deleting Files and their corresponding edits file.
CreateFile.java (Polymorphic Mode of AbstractFile) <ul style="list-style-type: none">- Extends AbstractFile.java- When a new file is created, a mirror file for edit history is also created. (Software.txt would have another file paired with it called EDITS_Software.txt, these files are not visible in the CLI or editable from the CLI)	AbstractFile.java <ul style="list-style-type: none">- Contains a variable File type that stores the File path of the current file.- Using encapsulation, AbstractFile can return either the path of the current file or the path of the current file edits file.- AbstractFile contains the empty methods that are overridden and inherited by the other functions.
ListFiles.java <ul style="list-style-type: none">- Simply print out the list of the filtered documents (EDITS_ files are hidden).	
DisplayContents.java (Polymorphic Mode of AbstractFile) <ul style="list-style-type: none">- Extends AbstractFile- Sets the current file path Displays the contents of a selected file.- Sets the contents of the current file path stored in the superclass of AbstractFile to the file the user wishes to view.- Displays the contents of either the file or the 'edits' version of the file depending on the arguments passed through to it.	
Contribution: 50%	Contribution: 50%

Finally, we used Git Version control and GitHub to collaborate on the program together and provide us with a solution if we needed to rollback to a previous version of the program.