File Management Application

Specifications

Application to displays and manages files stored in company facilities.

The user interacts with the application by reading options on menu and submenu and making valid choices and entering file names to be added or deleted or retrieved.

Features

Main menu

- Greets the user with information on the application.
- Provides the user with 3 options:
    - Option 1: display existing files in ascendant order.
    - Option 2: Advanced options to manage the files.
    - Option 6: Exit the application.
- The main menu is capable of handling incorrect entries from the user.

Sub menu

- Provides the user with 4 options
    - Option 1: Adding files.
        - Prompts the user for the file to be added.
        - Checks if the file already exists and reports to the user.
        - Adds the file if not already in the storage.
        - Prompts the user if willing to add more files.
        - Always adds the files to the nearest location available (lowest index) to optimize storage.
    - Option 2: Deleting files.
        - Prompts the user for the file to be deleted.
        - Deletes the file if already exists and informs the user.
        - Reports to the user if the file does not exist.
        - Prompts the user if willing to delete more files.
        - Keeps track of available locations if file deleted to use it when adding new files.
    - Option 3: Searching for files.
        - Prompts the user for the file to be search for.
        - Search for the file and reports its location to the user.
        - Reports to the user if the file does not exist.
        - Prompts the user if willing to search for more files.
    - Option 6: Exit the sub-menu and go back to the main menu.
- The sub-menu is capable of handling incorrect entries from the user.

Development

The work is planned to be completed in 3 sprints.

Sprint 1: 1 week

- Identify and build the data structure.
- Provide methods for adding, deleting, and retrieving files from the data structure.
- Test manually the operations.

Sprint 2: 1 week

- Main menu for display, sub-menu and app-exit.
- Sub-menu for files management.
- Management of wrong user entries.

Sprint 3: 1 week

- Connecting the menus with the data structure operations.
- Optimizing the user display feedback.
- Updating documentation.

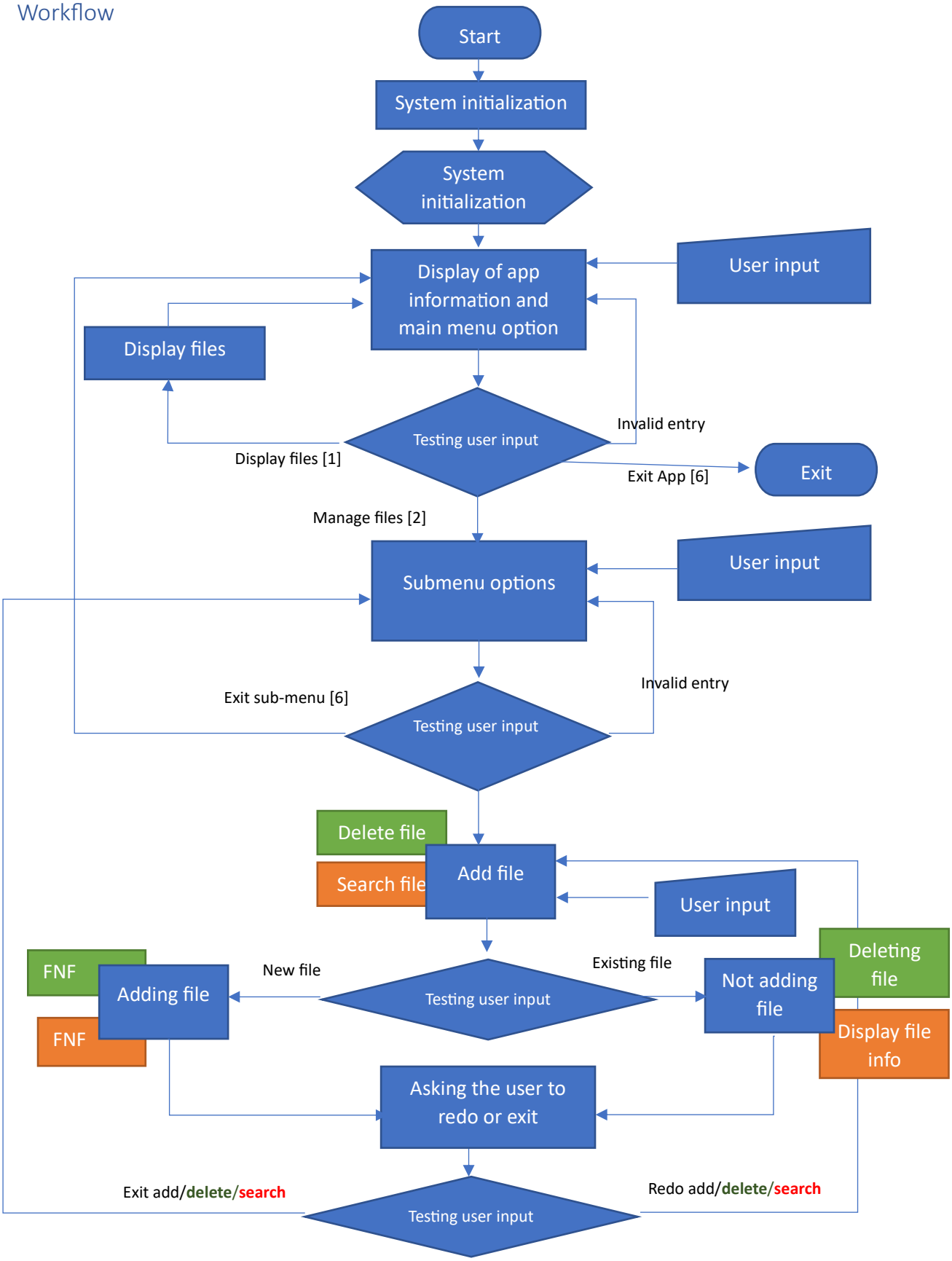
Methodology

- Application is written in Java.
- Code version control using Git and stored in GitHub-repository.
- Files elements are stored in FileNode objected containing file name and storage location.
- Storage listing uses a linked-list of files (FileNode elements).
    - Data structure chosen to accommodate unlimited size limit.
    - Elements added in ascending order using **Insertion Sort**.
    - List attributes are made private and accessible via getters.
    - Arrays of files and storage locations can be generated from the linked-list to facilitate searching.
    - Searching for files is done using **Binary Search** since the list of fields is already sorted.
    - When files are deleted, available storage locations are stored in **Queue** so these locations will be used first when new files are added.
    - Try/Catch blocks are used to capture invalid entries from the user and manage unexpected behavior of the application.
    - Some methods are overridden to accommodate various ways to add, delete and retrieve files.

Conclusion

- Application is interactive and robust against user's wrong entries.
- Inserting new files in sorted order allows fast retrieval of files.
- Storage is optimized since the lowest index locations are filled first.
- Future enhancement can be:
    - Directory management.
    - Using GUI / webapp.

Workflow


Start
System initialization
System initialization
Display of app information and main menu option
User input
Display files
Testing user input
Invalid entry
Display files [1]
Exit App [6]
Exit
Manage files [2]
Submenu options
User input
Exit sub-menu [6]
Invalid entry
Testing user input
Delete file
Search file
Add file
User input
FNF
Adding file
FNF
New file
Testing user input
Existing file
Not adding file
Deleting file
Display file info
Asking the user to redo or exit
Exit add/delete/search
Testing user input
Redo add/delete/search