

File Catalog Functional Test Plan				
Objectives	Test Case ID	Test Case Description	Test Data	Expected Results
Test if source files are returned after searching for file name that exists	TC_SEARCHDISK_001	Enter existing file name	File name that exists: "Test.c"	Source files with those names are returned
Test if source files are returned after searching for file name that doesn't exist	TC_SEARCHDISK_002	Enter nonexistent file name	File name that doesn't exist: "NonExistent.c"	No source files are returned
Test if source files are returned after searching for file type that exists	TC_SEARCHDISK_003	Enter existing file type	File type that exists: ".c"	All files with the ".c" extension are returned
Test if source files are returned after searching for file type that doesn't exist	TC_SEARCHDISK_004	Enter nonexistent file type	File type that doesn't exist in disk	No source files are returned
Test if files with unsupported file types are skipped during file system search	TC_SEARCHDISK_005	Enter unsupported file type	File type: ".txt"	No source files are returned
Test if selected source files are added to catalog successfully	TC_ADDTOCATALOG_001	Select source file and add to catalog	File name: "Test.c"	The source file is added to the catalog with the correct metadata and matches the selection
Test if duplicate source file names are handled correctly in the catalog	TC_ADDTOCATALOG_002	Select source file with duplicate name and add to catalog	File name that has duplicate: "Duplicate.c"	The source file is added to the catalog with the correct metadata and matches only the selection and not its duplicates
Test if attached annotations of source files are added to catalog	TC_ADDTOCATALOG_003	Select source file and add annotations	File name: "Test.c" Annotation: "This is a test program"	The annotation is added to the catalog to the corresponding file
Test if empty annotations of source files show up as empty in catalog	TC_ADDTOCATALOG_004	Select source file and add no annotation	File name: "Test2.c" Annotation: ""	Empty annotation is added to the catalog to the corresponding file
Test if annotations for two different files with the same name are handled correctly	TC_ADDTOCATALOG_005	Select source file with duplicate name and add annotation	File name: "Duplicate.c" Annotation: "This one should have annotation"	The annotation is added to the catalog only to the corresponding file and not its duplicates
Test if location attribute of file in catalog is updated after source file is moved to different location	TC_VALIDATION_001	Move source file to different location and validate	File name: "Test.c" Original location: "C:/TestDirectory" New location: "C:/NewDirectory"	Location attribute to corresponding file must be changed in catalog
Test if file is in catalog after source file has been deleted off disk	TC_VALIDATION_002	Delete file and validate	File name: "Test2.c"	Deleted file should not appear in catalog
Test if file name in catalog is updated after name of source file in disk is modified	TC_VALIDATION_003	Change file name and validate	Old File name: "Test3.c" New File name: "ChangedName.c"	File name to corresponding file must be changed in catalog
Test if invalid metadata entries are flagged during catalog validation	TC_VALIDATION_004	Add files with intentionally invalid or corrupted metadata and trigger c.	File name with incorrect metadata format, e.g., "InvalidMeta.	An error should be reported, and the system should prompt the user to correct or delete the invalid entry.
Test if deleted files on disk are auto-updated in the catalog	TC_VALIDATION_005	Delete a file on the disk and refresh the catalog view.	File to delete: "OldTest.c".	The file should automatically be flagged as missing in the catalog, and the user should receive a prompt to delete or update the entry.
Test if location attribute and file name in catalog is updated if location + name are both modified	TC_VALIDATION_006	Change file name and move to different location then validate	Old File name: "ChangedName.c" New File name: "Test3.c" Original location: "C:/TestDirectory" New location: "C:/NewDirectory"	Location attribute and file name to corresponding file must be changed in catalog
Test whether the program can find the differences between two files on the disk	TC_COMPARE-DISK-DISK_001	Select two files that have differences in source code	First File: "Compare1.c" Second File: "Compare2.c"	Difference between two files are highlighted (lines of code)
Test whether the program can recognize that two identical files on the disk	TC_COMPARE-DISK-DISK_002	Select two files that have no differences in source code	First File: "Compare1.c" Second File: "Compare1Copy.c"	No differences are highlighted
Test whether the program can find the differences between two files on the catalog	TC_COMPARE-CAT-CAT_001	Select two files in catalog that have differences in source code	First File: "Compare1.c" Second File: "Compare2.c"	Difference between two files are highlighted (lines of code)
Test whether the program can recognize that two identical files on the catalog	TC_COMPARE-CAT-CAT_002	Select two files in catalog that have no differences in source code	First File: "Compare1.c" Second File: "Compare1Copy.c"	No differences are highlighted
Test whether the program can find the differences between a file on the disk and a file on the catalog	TC_COMPARE-CAT-DISK_001	Select two files in catalog that have differences in source code	First File: "Compare1.c" Second File: "Compare2.c"	Difference between two files are highlighted (lines of code)
Test whether the program can recognize that two identical files between the disk and catalog	TC_COMPARE-CAT-DISK_002	Select two files in catalog that have no differences in source code	First File: "Compare1.c" Second File: "Compare1Copy.c"	No differences are highlighted
Test if catalog files are returned after searching for file name that exists	TC_SEARCHCATALOG_001	Enter existing file name in catalog	File name: "Test.c"	Source files with those names are returned
Test if catalog files are returned after searching for annotation that exists	TC_SEARCHCATALOG_002	Enter nonexistent file name in catalog	File name: "NonExistent.c"	No source files are returned
Test if catalog files are returned after searching for annotation that doesn't exist	TC_SEARCHCATALOG_003	Enter existing annotation in catalog	Annotation: "This is a test program"	Source files with those annotations are returned
Test if catalog files are returned after searching for annotation that doesn't exist	TC_SEARCHCATALOG_004	Enter nonexistent annotation in catalog	Annotation: "Blablabla"	No source files are returned
Test if catalog files are returned after searching for file type that exists	TC_SEARCHCATALOG_005	Enter existing file type in catalog	File type: ".c"	Source files with those file types are returned
Test if catalog files are returned after searching for file type that doesn't exist	TC_SEARCHCATALOG_006	Enter nonexistent file type in catalog	File type that doesn't exist	No source files are returned
Test if catalog files are returned after searching for last edited date that exists	TC_SEARCHCATALOG_007	Enter existing last edited date in catalog	Last edited date that exists	Source files with those last edited dates are returned
Test if date-based searches consider time zones correctly	TC_SEARCHCATALOG_008	Search for files by last modified date using different time zones.	Search date in specific timezone.	The search should adjust for timezone differences and provide accurate results based on the user's local time zone.
Test if combined search criteria return accurate results	TC_SEARCHCATALOG_009	Enter multiple search criteria (e.g., file type and annotation) simultaneously to filter results.	File type ".c"; Annotation "Test Program".	Only files matching both criteria should be returned.
Test if catalog files are returned after searching for last edited date that doesn't exist	TC_SEARCHCATALOG_010	Enter nonexistent last edited date in catalog	Last edited date that doesn't exist	No source files are returned
Test if edited annotations are updated in catalog	TC_EDITANNOTATION_001	Select file in catalog and edit its annotations	File name: "Test.c" Annotation: "This is a edited"	Annotations for selected file is shown accordingly to edit
Test if edited empty annotations show up as empty in catalog	TC_EDITANNOTATION_002	Select file in catalog and edit its annotations to empty string	File name: "Test.c" Annotation: ""	Annotations for selected file is empty
Test if edited annotations for two different files with the same name are handled correctly	TC_EDITANNOTATION_003	Select two files in catalog with same name and edit its annotations	First File: "Duplicate.c" Second File: "Duplicate.c" First Annotation: "A" Second Annotation: "B"	Annotations for same file names are shown accordingly to edit

Test if file can be moved to new location	TC_MOVEFILE_001	Select file and move to existing location	File name: "Test.c" Original location: "C:/TestDirectory" New location: "C:/NewDirectory"	Selected file is moved to new location
Test if file can be moved to non-existent location	TC_MOVEFILE_002	Select file and move to non-existent location	File name: "Test.c" Original location: "C:/TestDirectory" New location: "C:/blabla"	Error to display that selected file cannot be moved
Test if moving a file with duplicate names can be moved to new location	TC_MOVEFILE_003	Select file which has duplicate names and move to new location	File name: "Duplicate.c" Original location: "C:/TestDirectory" New location: "C:/NewDirectory"	Only selected file is moved to new location
Test if moving a file to a location that already has a file with the same name is handled correctly	TC_MOVEFILE_004	Select file which has name that is the same as file already in new location	File name: "Duplicate.c" Original location: "C:/TestDirectory" New location: "C:/NewDirectory"	User is prompted to overwrite file, rename, or keep
Test if catalog files are deleted from catalog	TC_DELETEFROMCATALOG_001	Select file and delete from catalog	File name: "Test.c"	File is deleted from catalog
Test if catalog files with the same name are affected by deletion	TC_DELETEFROMCATALOG_002	Select file with duplicate names and delete from catalog	File name: "Duplicate.c"	Only selected file is deleted from catalog and not its duplicate with a different extension
Test if an existing file can be queried and read from the backend	TC_QUERY_FILE_001	Query and read the value of an existing file	File name: "Test.c" Data: "test"["test"]["test].test,"	The program should return the contents of the file correctly
Test if the program can handle querying a non-existent file	TC_QUERY_FILE_002	Query and read the value of a non-existent file	File name: "NonExistent.c"	The program should return an error
Test the impact of file renaming conflicts	TC_RENAME_CONFLICT_001	Test the impact of file renaming conflicts	Original file name: "Test.c", New file name: "Duplicate.c".	A prompt should notify the user of the conflict and offer options (overwrite, rename, keep both).
Test the impact of concurrent file modifications	TC_CONCURRENCY_001	Simultaneously modify a file in the catalog and on the disk.	File name: "ConcurrentTest.c"	The system should handle concurrency gracefully, showing any conflicts and offering a merge option if applicable.
Test if the UI accurately displays search options and processes user input correctly.	TC_UI_001	Verify that the UI displays all available search options and processes user inputs correctly.	Search by annotation: "example annotation" Search by date: "2024-10-20" Search by file type: ".c" Search by file name: "TestFile.c"	UI displays options: Annotation, Date, File Type, File Name. After entering test data, the system performs a search. The correct results are displayed in the result table if data matches the catalog entries. For invalid formats or empty fields, a prompt error message should be displayed.
Verify that the UI displays detailed information about a selected file.	TC_UI_002	Check if the UI correctly shows the file details when a source file is	Select file "TestFile.c" from search results.	A file details panel/window should open. Information like file location, last modification date, and annotations should be accurately displayed. If the file has annotations, they should be visible; otherwise, an empty state should be shown.
Ensure that the UI correctly allows users to add files from disk to the catalog	TC_UI_003	Test the process of selecting and adding a file from the disk to the catalog using the UI.	Choose "NewFile.c" using JFileChooser.	The file should be added to the catalog. A confirmation dialog should pop up indicating the addition was successful. The file metadata should be correctly visible in the catalog. Any validation errors (e.g., unsupported file type) should trigger error dialogs.
Confirm that search results are properly displayed using UI components.	TC_UI_004	Validate that the search results are listed correctly in table component	Search for file type ".c", multiple .c files should be in the catalog.	All .c files are listed in the table. Columns include file name, file type, location, and modification date. Sorting and filtering options function correctly.
Test the file deletion functionality via the UI.	TC_UI_005	Remove a selected file from the catalog using the UI and confirm it's removed.	File name to delete: "DeleteTestFile.c"	The file should disappear from the search results. A confirmation dialog should ask the user to verify the action. Post-deletion, a success message should be displayed. The catalog should no longer list the deleted file.
Verify that the file comparison viewer highlights differences correctly.	TC_UI_006	Check if the UI's file comparison viewer accurately displays the differences between selected files.	First file: "File1.c" Second file: "File2.c"	The viewer should show differences line-by-line. Changes should be highlighted (e.g., additions in green, deletions in red). No differences should be shown if files are identical.
Ensure users can edit file annotations using the UI.	TC_UI_007	Modify an annotation of a file in the catalog via the UI.	File: "AnnotatedFile.c" Old Annotation: "Old Annotation" New Annotation: "New Annotation"	The annotation should update without error. The change should be saved in the database. The UI should immediately reflect the new annotation.
Check the client-side validation of user inputs.	TC_UI_008	Validate inputs in search fields for errors or incorrect formats.	Annotation: "%\$%" Date: "2024/20/20" File Type: "abcxyz" File Name: "averylongfilenameexceedinglimit.c"	Each invalid input should trigger a warning/error dialog. The system should not proceed with the search until corrected. Correct inputs should allow the search to continue.