**Black-Box**

Ordnerverwaltung

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Resultat |
| BB-01 | Neuen Ordner "Rechnungen" erstellen | Ordner erscheint im Ordnerstruktur |
| BB-02 | Nochmal Ordner "Rechnungen" erstellen | Fehlermeldung: "Name existiert bereits" |
| BB-03 | Ordner "Rechnungen" in "Invoices" umbenennen | Name wird geändert |
| BB-04 | Ordner "2024" in Ordner "Rechnungen" verschieben | Ordner ist jetzt unter "Rechnungen" |
| BB-05 | Leeren Ordner löschen | Ordner ist weg |
| BB-06 | Ordner mit Dokumenten löschen | Warnung: "Ordner enthält Dokumente" |

Dateiimport

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Resultat |
| BB-07 | PDF manuell importieren | Datei erscheint im Ordnerstruktur |
| BB-08 | DOCX manuell importieren | Datei erscheint, Text wird extrahiert |
| BB-09 | .txt Datei importieren | Fehlermeldung: "Nur PDF/DOCX erlaubt" |
| BB-10 | PDF in Watcher-Ordner kopieren | Datei wird automatisch importiert |

Suche

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Resultat |
| BB-11 | Nach "Rechnung" suchen | Dokumente mit "Rechnung" werden gefunden |
| BB-12 | Nach "xyz123" suchen (existiert nicht) | Meldung: "Keine Treffer gefunden" |
| BB-13 | Suchergebnis anklicken | Dokument wird rechts angezeigt, Suchbegriff gelb markiert |

GUI

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Resultat |
| BB-14 | Ordner auf-/zuklappen | Unterordner werden ein-/ausgeblendet |
| BB-15 | Dokument anklicken | Inhalt wird rechts angezeigt |
| BB-16 | Fenster vergrössern | Layout passt sich an |

**White-Box**

Datenbank-Funktionen

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Prüft |
| WB-01 | CreateFolder() | Wird Ordner in DB gespeichert? Bekommst du eine ID zurück? |
| WB-02 | GetFoldersByParentId() | Werden nur Unterordner vom richtigen Ordner geholt? |
| WB-03 | ValidateFolderName() | Erkennt die Funktion doppelte Namen? |
| WB-04 | SaveDocumentBlob() | Werden Datei UND Text gespeichert? |

Text-Extraktion

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Prüft |
| WB-05 | ExtractPdf() | Datei erscheint im Ordnerstruktur? |
| WB-06 | ExtractPdf() mit kaputtem PDF | Gibt es eine Fehlermeldung statt Absturz? |
| WB-07 | ExtractDocx() | Wird Text aus Word extrahiert? |

Suche

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Prüft |
| WB-08 | SearchFullText("Rechnung") | Werden richtige Dokumente gefunden? |
| WB-09 | GenerateSnippet() | Wird Textausschnitt mit Highlight erstellt? |

File Watcher

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Prüft |
| WB-10 | FileWatcherService.OnCreated() | Wird Event ausgelöst, wenn Datei hinzugefügt wird? |
| WB-11 | StartWatching() | Startet Watcher den richtigen Ordner? |

GUI-Code

|  |  |  |
| --- | --- | --- |
| Test Nr. | Funktion | Prüft |
| WB-12 | Button Click → Service aufrufen | Wird richtige Funktion aufgerufen? |
| WB-13 | TreeView Binding | Aktualisiert sich der Ordnerbaum automatisch? |
| WB-14 | Input-Validierung | Werden leere Ordnernamen abgelehnt? |
| WB-15 | Exception-Handling | Wird Fehler abgefangen und Meldung gezeigt? |

Old.

**White-Box**

**Data Layer:**

* CreateFolder() → DB-Insert korrekt ausgeführt, ID zurückgegeben
* GetFoldersByParentId() → Hierarchie-Query liefert nur Unterordner
* ValidateFolderName() → Prüft Eindeutigkeit nur auf gleicher Ebene (ParentId)
* SaveDocumentBlob() → BLOB und PlainText werden korrekt gespeichert
* DeleteFolder() → Cascade Delete funktioniert (oder Exception bei Dokumenten)

**Business Layer:**

* TextExtractionService.ExtractPdf() → Text wird korrekt extrahiert, Exception bei fehlerhafte Datei
* TextExtractionService.ExtractDocx() → Text wird korrekt extrahiert, Exception-Handling
* DocumentService.ValidateFileType() → Nur PDF/DOCX erlaubt, andere abgelehnt
* SearchService.SearchFullText() → LIKE-Query korrekt konstruiert mit Contains()
* SearchService.GenerateSnippet() → Snippet mit korrekter Länge und Hervorhebung
* FolderService.ValidateUniqueName() → Doppelte Namen auf gleicher Ebene erkannt

**File Watcher:**

* FileWatcherService.OnCreated() → Event wird ausgelöst bei neuer Datei
* FileWatcherService.OnCreated() → Nur gültige Dateitypen werden verarbeitet
* FileWatcherService.StartWatching() → Watcher startet und überwacht korrekten Pfad
* Exception bei nicht existierendem Watcher-Pfad → Fehler wird abgefangen
* Thread-Safety → Mehrere gleichzeitige Dateien werden korrekt verarbeitet

**GUI-Logik:**

* Event-Handler für Button-Clicks → Korrekte Service-Methode wird aufgerufen
* TreeView DataBinding → Ordnerstruktur aktualisiert sich bei DB-Änderung
* ListView DataBinding → Suchergebnisse werden korrekt angezeigt
* Input-Validierung → Leere Ordnernamen werden abgelehnt
* Exception-Handling in GUI → Try-Catch fängt Fehler ab, zeigt Meldung

**Code-Coverage:**

* Alle CRUD-Operationen durchlaufen → Mindestens 80% Coverage
* Alle Exception-Handler getestet → Catch-Blöcke durch provozierte Fehler erreicht
* Boundary-Tests → Null-Werte, leere Strings, Maximalwerte getestet