

REPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR

Université de Yaoundé I

Institut Saint Jean



REPUBLIC OF CAMEROON

Peace-Work-Fatherland

MINISTRY OF HIGHER
EDUCATION

University of Yaounde I

Institut Saint Jean

Cahier d'analyse

Conception et mise en œuvre d'un gestionnaire de fichiers multi-utilisateurs: cas de Pop!_OS

Rédigé et présenté par :

TEMATIO Zixfiline powera
PAYONG Brice valery

Année Académique

2025-2026

ETUDE DU PROJET

Nom du projet :	Conception et mise en œuvre d'un gestionnaire de fichiers multi-utilisateurs: cas de Pop!_OS
Version du document :	Deuxième version
Date de création :	06/01/2026
Dernière mise à jour :	22/01/2026
Responsable projet :	<ul style="list-style-type: none">• <u>PAYONG Brice Valery</u>• <u>TEMATIO ZEBAZE Zixfiline Powera</u>
Validé par :	<u>M. NGUIMBUS Emmanuel</u>

SOMMAIRE

ETUDE DU PROJET	2
SOMMAIRE.....	3
LISTE DES TABLEAUX	4
INTRODUCTION.....	5
I. LES OBJECTIFS DU PROJET	5
a) Objectif global.....	5
b) Objectifs spécifiques	6
II. LES CAS D'UTILISATION DU SYSTEME.....	6
A. Les acteurs et leurs rôles	6
B. SPECIFIACIONS DES BESOINS	7
a) Besoins fonctionnels	7
c) Besoins non fonctionnels	8
Diagramme de cas d'utilisation	9
CONCLUSION GÉNÉRALE	20
TABLES DES MATIÈRES	21

LISTE DES TABLEAUX

Tableau 1: technologies de développement.....	Erreur ! Signet non défini.
Tableau 2: intervenant du projet.....	Erreur ! Signet non défini.
Tableau 3: Simulation du diagramme de gantt.....	Erreur ! Signet non défini.
Tableau 4: Ressources Matérielles	Erreur ! Signet non défini.
Tableau 5: Ressources Logicielles	Erreur ! Signet non défini.
Tableau 6: Ressources Humaines	Erreur ! Signet non défini.

INTRODUCTION

La gestion des fichiers sous Pop!_OS repose sur une structure hiérarchique rigoureuse et un système de permissions qui garantit l'intégrité des données dans un contexte multi-utilisateurs. Si le cahier des charges a posé les fondations du projet en identifiant la nécessité d'un outil de gestion intuitif, le présent **Cahier d'Analyse** a pour mission de traduire ces intentions en spécifications logiques détaillées.

L'objectif de ce document est de définir précisément le comportement attendu du système. Il s'agit d'analyser comment l'application devra interagir avec le système de fichiers pour répondre aux besoins de navigation, de manipulation (CRUD) et de gestion des droits d'accès. Cette étape est cruciale car elle permet de modéliser les processus métier — notamment la gestion des priviléges et l'isolation des utilisateurs — avant toute considération technique de réalisation.

Pour ce faire, l'analyse s'articule autour de trois axes principaux :

- **L'identification des acteurs** et des interactions possibles avec le gestionnaire de fichiers.
- **La définition du périmètre fonctionnel**, détaillant chaque opération que l'utilisateur pourra réaliser sur ses ressources.
- **La modélisation des cas d'utilisation**, permettant de visualiser le dialogue entre l'utilisateur et le système, ainsi que les règles de gestion qui régissent les accès sécurisés.

Ainsi, ce document constitue la référence fonctionnelle qui servira de base à la future phase de conception, en s'assurant que tous les besoins exprimés sont traduits en exigences logiques cohérentes et conformes aux standards de sécurité de Pop!_OS.

I. LES OBJECTIFS DU PROJET

a) Objectif global

L'objectif global du projet est de concevoir et de mettre en œuvre un gestionnaire de fichiers multi-utilisateurs fonctionnant sous Pop!_OS, permettant d'effectuer les opérations de base (création, lecture, modification et suppression) sur les fichiers et répertoires, tout en respectant les droits d'accès définis par le système. Ce gestionnaire doit offrir une interface graphique simple et intuitive, facilitant l'utilisation des fonctionnalités tout en illustrant les concepts fondamentaux liés aux systèmes de fichiers Linux.

b) Objectifs spécifiques

Les objectifs spécifiques du projet sont les suivants :

- Mettre en place une interface permettant de parcourir les fichiers et les répertoires.
- Intégrer les actions de création, lecture, modification et suppression.
- Prendre en compte les droits d'accès définis pour chaque utilisateur.
- Empêcher les opérations non autorisées selon les permissions du système.
- Afficher les informations de base des fichiers (type, taille, date de modification, permissions).
- Permettre la création et l'organisation de répertoires de manière simple.

II. LES CAS D'UTILISATION DU SYSTEME

A. Les acteurs et leurs rôles

Dans le cadre de cette application, les acteurs sont les entités externes (humaines ou techniques) qui interagissent avec le gestionnaire de fichiers. Nous distinguons les acteurs selon leur implication dans le déclenchement des processus.

L'Acteur Principal : L'Utilisateur connecté

L'utilisateur connecté est l'acteur **humain** et **initiateur**. Il se situe à l'origine de toutes les interactions avec le système.

- **Rôle** : Il utilise l'interface web pour piloter ses fichiers. C'est lui qui déclenche les cas d'utilisation tels que la navigation dans l'arborescence, la création de dossiers ou la modification des permissions.
- **Responsabilité** : Il est garant des décisions prises sur ses données. En mode "Session Système", son identité (UID) définit le périmètre de ce qu'il peut voir ou modifier. Lorsqu'une action dépasse ses droits standards, il intervient en fournissant ses informations d'authentification pour obtenir les priviléges nécessaires (Sudo).

L'Acteur Secondaire : Le Système de fichiers Linux (Noyau)

Le Système de fichiers, géré par le noyau de **Pop!_OS**, est un acteur **technique** et **serveur**. Il ne déclenche aucune action de lui-même, mais il est indispensable à la réalisation des requêtes de l'utilisateur.

- **Rôle** : Il agit comme la source de vérité et l'exécuteur technique. Lorsque l'application reçoit un ordre de l'utilisateur, elle sollicite cet acteur pour lire le contenu du disque ou appliquer des changements.
- **Responsabilité** : Il est le garant de la sécurité et du respect des droits d'accès. C'est lui qui valide ou rejette les opérations en fonction des permissions réelles du système. Il renvoie à l'application les métadonnées (taille, type, droits \$rwx\$) indispensables à l'affichage de l'interface.

B. SPECIFICATIONS DES BESOINS

a) Besoins fonctionnels

Catégorie A : Opérations de Navigation

- **Parcourir** le système de fichiers en partant du répertoire personnel
- **Afficher** le contenu des répertoires accessibles
- **Naviguer** de manière fluide entre les répertoires
- **Retourner** au répertoire parent ou précédent

Catégorie B : Opérations de Manipulation

- **Créer** des fichiers et répertoires nouveaux
- **Lire** le contenu textuel ou binaire des fichiers
- **Modifier** le contenu des fichiers avec les droits appropriés
- **Supprimer** des fichiers et dossiers selon les permissions
- **Renommer** des éléments du système de fichiers
- **Copier** des fichiers et répertoires
- **Déplacer** des éléments entre différents emplacements

Catégorie C : Opérations de Sécurité

- **Consulter** les permissions détaillées des éléments
- **Modifier** les droits d'accès (lecture/écriture/exécution)
- **Changer** le propriétaire des fichiers et répertoires
- **Changer** le groupe associé aux éléments
- **Appliquer** les modifications de permissions récursivement
- **Authentifier** l'utilisateur pour les opérations privilégiées

Catégorie D : Opérations d'Information

- **Afficher** les propriétés complètes des éléments
- **Calculer** la taille des fichiers et répertoires
- **Identifier** le type exact de chaque élément
- **Vérifier** les dates de création et modification
- **Consulter** les métadonnées techniques

Catégorie E : Opérations de Supervision

- **Consulter** le journal d'activités
- **Exporter** les historiques d'actions

c) Besoins non fonctionnels

Il s'agit des exigences qui ne concernent pas directement le comportement du système.

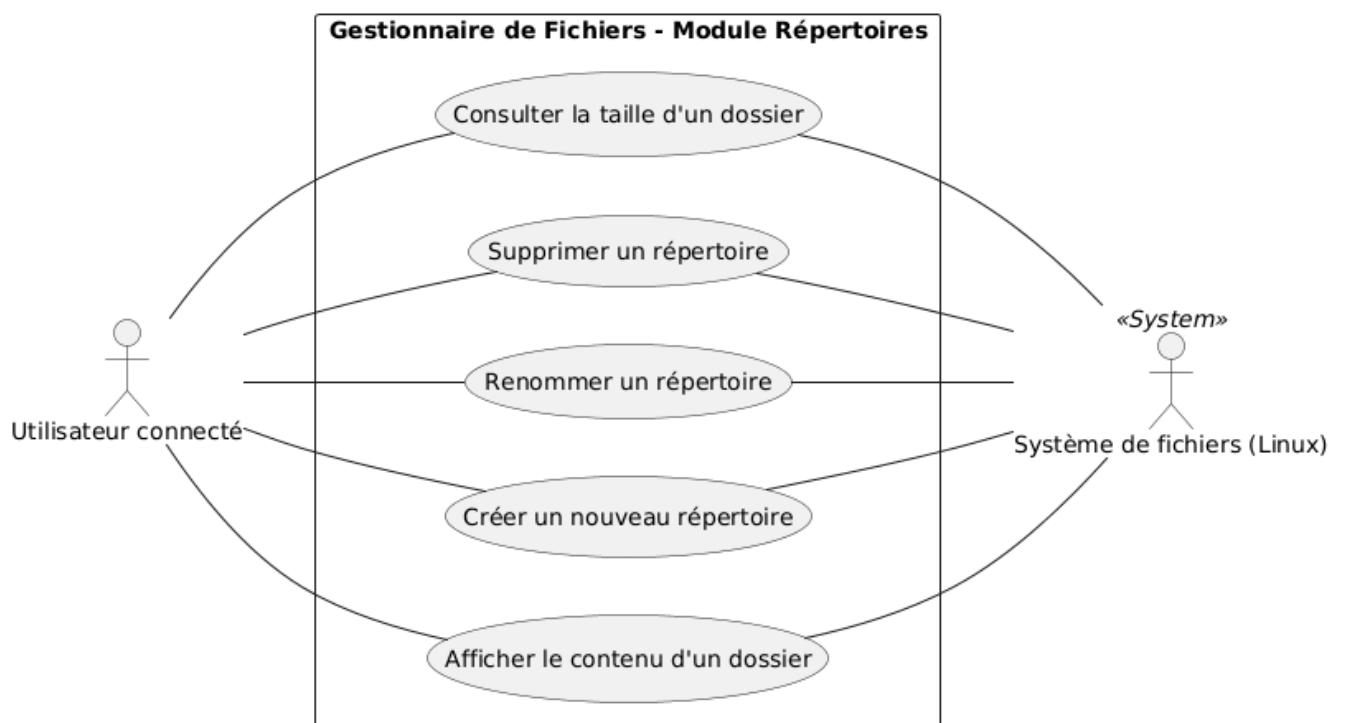
L'application devra respecter les points suivants :

- **Maintenabilité** : L'application devra être conçue de manière modulaire afin de permettre la correction rapide des anomalies et l'évolution aisée des fonctionnalités sans perturber le système existant.
- **Intégrabilité** : La structure du projet devra permettre l'ajout futur de nouvelles options (modules de sécurité, extensions, fonctionnalités avancées) sans nécessiter une refonte complète de l'application.

- **Performance** : L'application devra garantir un temps de réponse fluide lors de l'exploration des répertoires et des opérations CRUD, même dans des dossiers contenant un grand nombre de fichiers.
- **Fiabilité** : Les fonctionnalités doivent fonctionner de manière constante et cohérente, en évitant les crashs, erreurs inattendues ou pertes de données lors des manipulations.
- **Design** : L'interface graphique devra être intuitive, lisible et facile à prendre en main, afin de permettre aux utilisateurs, même novices, de gérer leurs fichiers sans difficulté.
- **Sécurité** : Le système devra appliquer strictement les droits d'accès Linux pour chaque utilisateur, garantissant : la confidentialité des données, leur intégrité lors des modifications et la disponibilité des fichiers autorisés.

Diagramme de cas d'utilisation

Module Répertoire



1. Cas d'utilisation « Afficher le contenu d'un dossier »

Résumé	Permet à l'utilisateur de visualiser la liste des fichiers et sous-dossiers présents dans un répertoire spécifique.

Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'application est lancée et l'utilisateur est authentifié.
Préconditions	L'utilisateur dispose des droits de lecture (\$r\$) sur le répertoire cible.
Scénario nominal	<p>1. L'utilisateur clique sur un dossier ou accède à un chemin.</p> <p>2. Le système sollicite le Système de fichiers pour lister le contenu.</p> <p>3. Le Système de fichiers renvoie la liste des éléments et leurs métadonnées.</p> <p>4. Le système affiche dynamiquement les icônes et noms des fichiers/dossiers.</p>
Scénario alternatif	2a. L'utilisateur n'a pas les permissions de lecture : le système affiche un message "Accès refusé" et reste sur le répertoire précédent.
Postconditions	Le contenu du répertoire est affiché à l'écran.

2. Cas d'utilisation « Créer un nouveau répertoire »

Résumé	Permet à l'utilisateur de générer un nouveau dossier dans l'arborescence actuelle.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur se trouve dans le répertoire parent où il souhaite créer le dossier.
Préconditions	L'utilisateur possède les droits d'écriture (\$w\$) sur le répertoire parent.
Scénario nominal	1. L'utilisateur clique sur le bouton "Nouveau dossier".

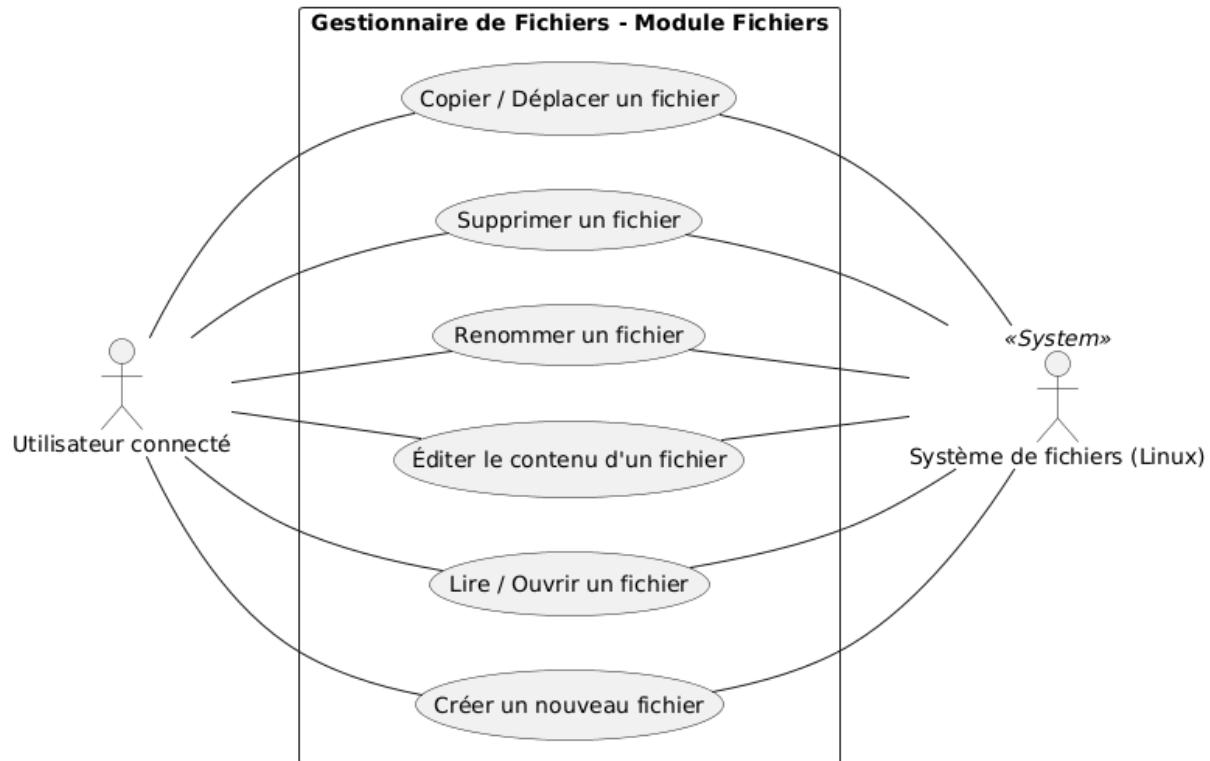
	<p>2. Le système affiche un champ de saisie pour le nom.</p> <p>3. L'utilisateur saisit le nom et valide.</p> <p>4. Le système demande au Système de fichiers de créer le dossier.</p> <p>5. Le Système de fichiers confirme la création.</p> <p>6. Le système actualise l'affichage pour montrer le nouveau répertoire.</p>
Scénario alternatif	<p>4a. Un dossier porte déjà le même nom : le système affiche une erreur et demande un autre nom.</p> <p>4b. Permissions insuffisantes : le système propose une élévation de privilèges (Sudo) ou annule l'action.</p>
Postconditions	Un nouveau répertoire est physiquement présent sur le disque.

Cas d'utilisation « Supprimer un répertoire »

Résumé	Permet de retirer définitivement un dossier et son contenu du système.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur a sélectionné le répertoire à supprimer.
Préconditions	L'utilisateur possède les droits d'écriture sur le répertoire parent.
Scénario nominal	<p>1. L'utilisateur sélectionne un dossier et clique sur "Supprimer".</p>

	<p>2. Le système demande une confirmation de suppression.</p> <p>3. L'utilisateur confirme l'action.</p> <p>4. Le système demande au Système de fichiers de supprimer l'élément.</p> <p>5. Le Système de fichiers exécute la commande et confirme la suppression.</p> <p>6. Le système retire l'élément de l'interface graphique.</p>
Scénario alternatif	<p>4a. Le répertoire n'est pas vide et l'utilisateur n'a pas les droits récursifs : le système affiche une erreur.</p> <p>4b. Le répertoire est protégé par le système : le système demande le mot de passe administrateur (Sudo).</p>
Postconditions	Le répertoire est effacé du support de stockage.

Module Fichiers



1- Cas d'utilisation « Créer un nouveau fichier »

Résumé	Permet à l'utilisateur de générer un nouveau fichier vide dans le répertoire courant.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur navigue dans un répertoire où il souhaite ajouter un fichier.
Préconditions	L'utilisateur possède les droits d'écriture (\$w\$) sur le répertoire parent.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur le bouton "Nouveau fichier". 2. Le système affiche une boîte de dialogue pour saisir le nom et l'extension. 3. L'utilisateur saisit le nom (ex: notes.txt) et valide.

	<p>4. Le système demande au Système de fichiers de créer le fichier vide.</p> <p>5. Le Système de fichiers crée le nœud sur le disque et confirme.</p> <p>6. Le système actualise la liste des fichiers pour afficher la nouvelle entrée.</p>
Scénario alternatif	<p>4a. Un fichier du même nom existe déjà : le système affiche une erreur et demande de renommer.</p> <p>5a. Espace disque insuffisant : le système lève une exception et informe l'utilisateur.</p>
Postconditions	Un fichier vide est présent sur le disque avec les permissions par défaut.

2. Cas d'utilisation « Modifier le contenu d'un fichier »

Résumé	Permet à l'utilisateur d'éditer le texte à l'intérieur d'un fichier existant.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur a ouvert l'éditeur de texte intégré à l'application.
Préconditions	Le fichier est de type texte et l'utilisateur possède les droits d'écriture (\$w\$).
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur modifie le texte dans l'interface web. 2. L'utilisateur clique sur "Enregistrer". 3. Le système envoie le nouveau contenu au serveur.

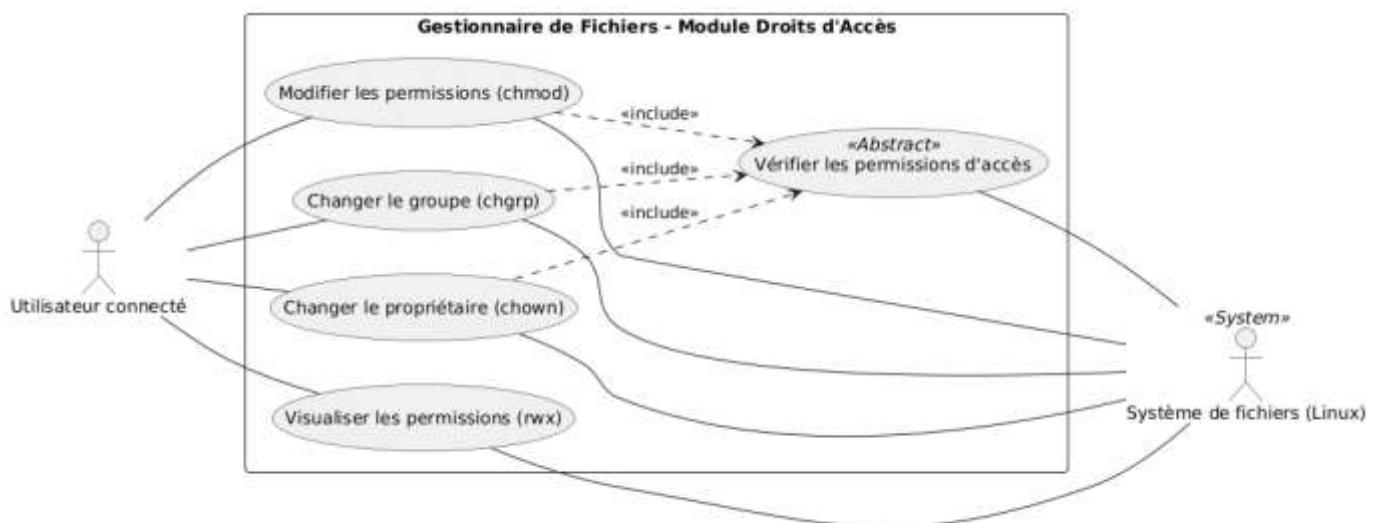
	<p>4. Le système demande au Système de fichiers d'écarter l'ancien contenu par le nouveau.</p> <p>5. Le Système de fichiers valide l'écriture et met à jour la date de modification.</p> <p>6. Le système affiche une notification de succès.</p>
Scénario alternatif	<p>4a. Le fichier est devenu "Lecture seule" entre-temps : le système affiche une erreur de permission.</p> <p>4b. Conflit d'accès (fichier ouvert ailleurs) : le système lève une exception.</p>
Postconditions	Le contenu du fichier sur le disque est mis à jour.

3- Cas d'utilisation « Déplacer un fichier »

Résumé	Permet de changer l'emplacement d'un fichier d'un répertoire source vers un répertoire cible.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur a sélectionné un fichier et choisi une destination.
Préconditions	L'utilisateur a les droits d'écriture sur le répertoire source et le répertoire cible.
Scénario nominal	<p>1. L'utilisateur sélectionne l'option "Déplacer vers".</p> <p>2. L'utilisateur choisit le répertoire de destination.</p> <p>3. Le système demande au Système de fichiers de déplacer le pointeur du fichier vers le nouveau chemin.</p>

	<p>4. Le Système de fichiers effectue l'opération (équivalent d'un mv).</p> <p>5. Le système confirme le succès du transfert.</p> <p>6. Le système actualise l'affichage de l'ancien et du nouveau répertoire.</p>
Scénario alternatif	<p>3a. Le fichier de destination existe déjà : le système demande s'il faut remplacer ou annuler.</p> <p>4a. Le déplacement traverse deux partitions différentes : le système effectue une copie suivie d'une suppression.</p>
Postconditions	Le fichier est présent dans le répertoire cible et supprimé du répertoire source.

Module droit d'accès



1. Cas d'utilisation « Modifier les permissions (chmod) »

Résumé	Permet à l'utilisateur de modifier les droits de lecture, écriture et exécution pour lui-même, son groupe ou les autres utilisateurs.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur a ouvert le panneau des propriétés d'un fichier ou dossier.
Préconditions	L'utilisateur est propriétaire de l'élément ou possède des droits d'administration.
Scénario nominal	<p>1. L'utilisateur coche ou décoche les cases \$rwx\$ (lecture, écriture, exécution).</p> <p>2. L'utilisateur valide les modifications.</p> <p>3. Le système demande au Système de fichiers d'appliquer le nouveau masque de permissions.</p> <p>4. Le Système de fichiers met à jour l'inode de l'élément sur le disque.</p> <p>5. Le système confirme le succès du changement.</p>
Scénario alternatif	<p>3a. L'utilisateur n'est pas propriétaire : le système demande le mot de passe Sudo.</p> <p>4a. Erreur système : le Système de fichiers refuse l'opération (ex: système de fichiers en lecture seule).</p>
Postconditions	Les nouveaux droits d'accès sont appliqués et immédiatement effectifs pour tout le système.

Cas d'utilisation « Voir les permissions »

Résumé	Permet à l'utilisateur de consulter l'état actuel des droits d'accès (\$rwx\$) et des propriétés d'un fichier ou d'un dossier.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur navigue dans l'arborescence et a sélectionné un élément.
Préconditions	L'élément cible doit être accessible en lecture par le système pour en extraire les métadonnées.
Scénario nominal	<p>1. L'utilisateur clique sur l'icône "Informations" ou fait un clic droit sur un élément.</p> <p>2. Le système interroge le Système de fichiers pour récupérer les attributs de l'élément.</p> <p>3. Le Système de fichiers renvoie les permissions (format octal ou symbolique), l'UID (propriétaire) et le GID (groupe).</p> <p>4. Le système traduit ces données techniques en une interface lisible (cases à cocher ou texte).</p> <p>5. Le système affiche les informations à l'utilisateur dans un volet ou une fenêtre dédiée.</p>
Scénario alternatif	<p>2a. Le fichier a été supprimé ou déplacé entre-temps : le système affiche un message d'erreur "Élément introuvable" et actualise la vue.</p> <p>3a. Erreur de lecture des métadonnées : le système informe l'utilisateur que les propriétés ne sont pas disponibles.</p>
Postconditions	L'utilisateur a une connaissance précise des droits appliqués à l'élément.

3. Cas d'utilisation « Changer le groupe (chgrp) »

Résumé	Permet de modifier le groupe de sécurité associé à un fichier ou un répertoire.
Acteurs	Utilisateur (Principal), Système de fichiers Pop!_OS (Secondaire).
Présupposé	L'utilisateur souhaite partager l'accès à un fichier avec les membres d'un groupe spécifique.
Préconditions	L'utilisateur doit appartenir au groupe cible ou posséder les droits d'administration.
Scénario nominal	<p>1. L'utilisateur sélectionne un nouveau groupe dans la liste proposée.</p> <p>2. L'utilisateur valide le changement.</p> <p>3. Le système demande au Système de fichiers de modifier le GID (Group ID) de l'élément.</p> <p>4. Le Système de fichiers confirme la mise à jour des métadonnées.</p> <p>5. Le système rafraîchit l'affichage pour montrer le nouveau groupe associé.</p>
Scénario alternatif	<p>3a. L'utilisateur ne fait pas partie du groupe cible : le système demande une confirmation administrateur.</p> <p>4a. Le groupe saisi est invalide : le système lève une exception et annule l'opération.</p>
Postconditions	Les membres du nouveau groupe associé héritent des permissions définies dans la colonne "Groupe".

CONCLUSION GÉNÉRALE

En conclusion, les bases fonctionnelles sont désormais solidement établies. La prochaine étape du projet consistera en la conception technique, incluant le choix définitif de l'architecture logicielle (Pattern MVC, API REST) et la mise en œuvre des logiciels pour transformer ces besoins en une application robuste et performante.

TABLES DES MATIÈRES

Table des matières

ETUDE DU PROJET	2
SOMMAIRE.....	3
LISTE DES TABLEAUX	4
INTRODUCTION.....	5
I. LES OBJECTIFS DU PROJET	5
a) Objectif global.....	5
b) Objectifs spécifiques	6
II. LES CAS D'UTILISATION DU SYSTEME.....	6
A. Les acteurs et leurs rôles	6
B. SPECIFIATIONS DES BESOINS	7
a) Besoins fonctionnels	7
Catégorie A : Opérations de Navigation.....	7
Catégorie B : Opérations de Manipulation	7
Catégorie C : Opérations de Sécurité	8
Catégorie D : Opérations d'Information	8
Catégorie E : Opérations de Supervision	8
c) Besoins non fonctionnels	8
Diagramme de cas d'utilisation	9
CONCLUSION GÉNÉRALE	20
TABLES DES MATIÈRES	21

