

REPUBLICHE DU CAMEROUN

\*\*\*\*\*

Paix-Travail-Patrie

\*\*\*\*\*

MINISTERE DE L'ENSEIGNEMENT  
SUPERIEUR

\*\*\*\*\*

Université de Yaoundé 1

\*\*\*\*\*

Institut Saint Jean



REPUBLIC OF CAMEROON

\*\*\*\*\*

Peace-Work-Fatherland

\*\*\*\*\*

MINISTRY OF HIGHER  
EDUCATION

\*\*\*\*\*

University of Yaounde 1

\*\*\*\*\*

Saint Jean Institute

## PROJET SYSTEME D'EXPLOITATION

**Conception et mise en œuvre de File  
Analyzer, un analyseur avancé de fichiers :  
cas de EXT4 sous Debian**

### MEMBRE DU GROUPE :

- SONGMENE LADO Belviane

### Sous la direction de:

- M. NGUIMBUS Emmanuel

**ANNEE ACADEMIQUE: 2025-2026**



# SOMMAIRE

SOMMAIRE .....	i
LISTE DES TABLEAUX .....	ii
LISTE DES FIGURES .....	iii
PARTIE 1 : CAHIER DES CHARGES .....	1
INTRODUCTION.....	2
I. CONTEXTE ET JUSTIFICATION DU PROJET .....	3
II. PROBLEMATIQUE .....	4
III. LES OBJECTIFS DU PROJET .....	4
IV. EXPRESSION DES BESOINS .....	5
V. ACTEURS DU PROJET.....	8
VI. ESTIMATION DU COUT DU PROJET .....	8
VII. PLANIFICATION DU PROJET .....	10
VIII. LES CONTRAINTES DU PROJET .....	11
IX. LES LIVRABLES.....	11
CONCLUSION .....	12
PARTIE 2: CAHIER D'ANALYSE .....	13
INTRODUCTION.....	14
I. CONTEXTE ET OBJECTIF .....	15
II. PERIMETRE FONCTIONNEL.....	15
III. MODELISATION UML .....	16
IV. LIVRABLES .....	27
CONCLUSION .....	28
PARTIE 3: CAHIER DE CONCEPTION .....	29
INTRODUCTION.....	30
I. DIAGRAMMES DE SEQUENCE .....	31
CONCLUSION .....	33
TABLES DES MATIERES .....	34

## LISTE DES TABLEAUX

Tableau 1: Acteurs du projet.....	8
Tableau 2: Ressources Matérielles .....	8
Tableau 3: Ressources Logicielles .....	9
Tableau 4: Ressources humaines .....	9
Tableau 5: Tableau des taches .....	10
Tableau 6: CU1-Selectionner un fichier.....	18
Tableau 7: CU2-Analyser un fichier .....	19
Tableau 8: CU3-Visualiser les résultats.....	20
Tableau 9: CU4-Exporter le rapport d'analyse.....	21
Tableau 10: CU5-Changer de vue d'affichage .....	22
Tableau 11: CU6-Accéder à l'aide contextuelle .....	23
Tableau 12:CU7-Configurer les paramètres d'analyse .....	24
Tableau 13: CU8-Analyser plusieurs fichiers séquentiellement .....	25
Tableau 14: CU-Visualiser la carte physique des blocs .....	26

## LISTE DES FIGURES

Figure 1: Diagramme de Gantt.....	10
Figure 2: Diagramme des cas d'utilisation .....	17
Figure 3: séquence analyser un fichier.....	31
Figure 4: Séquence visualiser la carte physique des blocs.....	32

# PARTIE 1 : CAHIER DES CHARGES

# INTRODUCTION

Le cahier des charges est un document de référence qui présente, de manière structurée et synthétique, l'ensemble des fonctions que la solution finale devra offrir. Il décrit les besoins à satisfaire, les spécifications techniques auxquelles le projet devra se conformer, ainsi que les contraintes à respecter. Élaboré conjointement entre le commanditaire et le commandité, ce document contractuel sert de base commune pour orienter, encadrer et vérifier la réalisation du projet. Dans cette perspective, le présent cahier des charges expose le contexte dans lequel s'inscrit le projet, les objectifs visés, les besoins exprimés, les acteurs concernés et les différentes contraintes identifiées. Il précise également la planification générale, les livrables attendus ainsi que les critères d'évaluation qui permettront de mesurer la conformité de la solution développée. Ce document constitue ainsi un outil essentiel pour assurer le bon déroulement du projet et garantir que la solution proposée répondra fidèlement aux attentes définies en amont.

# I. CONTEXTE ET JUSTIFICATION DU PROJET

## 1. Contexte

Les systèmes de fichiers existent depuis plusieurs décennies. Dès les premières technologies de stockage, comme les cartes perforées ou les bandes magnétiques, ils permettaient déjà d'organiser les données, bien que de manière limitée. Ces anciens supports ne proposaient qu'un accès linéaire, nécessitant de parcourir séquentiellement l'ensemble du support pour atteindre une information précise, ce qui entraînait des temps d'accès importants.

Avec l'évolution des technologies de stockage, les systèmes de fichiers modernes offrent désormais un accès aléatoire, permettant de récupérer ou de modifier rapidement n'importe quelle donnée à partir de son emplacement. Un système de fichiers constitue ainsi un composant essentiel d'un système d'exploitation : il assure l'organisation, le stockage, la gestion et l'accès aux données sur un support tel qu'un disque dur, une clé USB ou une mémoire flash. Il joue également un rôle d'interface entre les applications et le matériel, rendant possibles des opérations comme la création, la lecture, l'écriture ou la suppression de fichiers.

Parmi les systèmes de fichiers largement utilisés dans les environnements Linux, **EXT4 (Extended filesystem-version4)** occupe une place de référence grâce à sa robustesse, ses performances et ses fonctionnalités avancées, notamment la gestion des extents, la journalisation et la réduction de la fragmentation.

## 2. Justification

Dans ce contexte, l'analyse du fonctionnement interne d'EXT4 reste une démarche complexe, souvent réservée aux outils spécialisés ou à des utilisateurs expérimentés. Les utilitaires standards offrent une vision partielle, tandis que les outils avancés sont parfois difficiles à interpréter.

Le projet propose donc la conception et la mise en œuvre d'un outil nommé **File Analyzer**, destiné à simplifier et à rendre accessible l'analyse détaillée des fichiers stockés sur une partition EXT4 sous Debian. Cet outil permettra d'extraire, d'interpréter et de présenter de manière claire l'ensemble des informations techniques et structurées relatives à un fichier : métadonnées, fragmentation, type de fichier, taille, permissions, et autres éléments internes.

## II. PROBLEMATIQUE

Malgré l'adoption généralisée d'EXT4 dans les environnements Linux, l'analyse approfondie de la structure interne des fichiers demeure complexe et fragmentée. Les utilisateurs sont confrontés à plusieurs obstacles :

- Les utilitaires standards (ls, stat, file) n'exposent qu'une vue superficielle des métadonnées, sans accéder aux structures internes d'EXT4 (inodes, extents, superblocks) ;
- L'analyse complète nécessite la combinaison de multiples outils spécialisés (debugfs, dumpe2fs, filefrag, xxd) dont l'utilisation requiert une expertise approfondie ;
- Aucune solution unifiée ne permet de corrélérer simultanément le type du fichier, ses métadonnées système, son organisation physique sur le disque et son état de fragmentation ;
- L'absence d'interface intuitive limite la compréhension concrète du fonctionnement interne d'EXT4, pourtant essentiel dans la formation en systèmes d'exploitation.

### Question principale :

*Comment concevoir un outil unique et accessible permettant d'analyser en profondeur la structure complète d'un fichier sur une partition EXT4 sous Debian, en exposant de manière intégrée ses métadonnées, son organisation physique et son état de fragmentation ?*

## III. LES OBJECTIFS DU PROJET

Les objectifs définissent la finalité de notre projet et se formulent en verbe à l'infinitif. Il est suffisamment détaillé pour permettre la planification et l'évaluation des activités de notre projet.

### 1. Objectif global

L'objectif principal, qui est le pourquoi de notre projet est **de Faciliter la compréhension et l'analyse approfondie des systèmes de fichiers EXT4** en fournissant un outil unifié permettant d'explorer de manière claire et accessible la structure interne, les métadonnées et l'organisation physique des fichiers sous Debian.

### 2. Objectifs spécifiques

Les axes de travail pour atteindre l'objectif global sont :

- Identifier automatiquement le type du fichier analysé et extraire l'ensemble des champs de son header (magic number, sections, métadonnées structurelles) ;

- Extraire et présenter les métadonnées système du fichier : permissions, propriétaire/groupe, taille logique, taille sur disque, horodatages (atime, mtime, ctime, btime), nombre de liens physiques et flags système (immutable, append-only, etc.) ;
- Cartographier l'organisation physique du fichier sur le disque en listant ses extents avec leurs caractéristiques (offset logique, longueur, numéros de blocs physiques) ;
- Évaluer le niveau de fragmentation du fichier en calculant son taux de fragmentation à partir de la distribution de ses blocs sur le disque ;
- Afficher le contenu brut du fichier lorsqu'il s'agit d'un fichier texte ou lisible, tout en indiquant clairement les fichiers binaires non affichables.

## IV. EXPRESSION DES BESOINS

Lorsque nous parlons de besoins ici, il s'agit des différentes fonctionnalités liées au projet. Il existe ainsi des **besoins fonctionnels** et des **besoins non fonctionnels**.

### 1. Besoins Fonctionnels

Les besoins fonctionnels représentent l'ensemble des actions que File Analyzer devra permettre d'effectuer.

#### i. Module « Détection et analyse du type de fichier »

Ce module constitue la première étape de l'analyse avancée. Il permet de :

- Déetecter automatiquement le type du fichier analysé (texte, binaire, exécutable, image, PDF, etc.) ;
- Extraire et afficher tous les champs du header du fichier (magic number, sections, métadonnées structurelles).

#### ii. Module « Extraction des métadonnées système »

Ce module permet d'accéder aux informations système associées au fichier. Il est chargé de :

- Extraire les métadonnées stockées dans l'inode du fichier (permissions, propriétaire, groupe, taille logique et physique) ;
- Récupérer les horodatages complets (atime, mtime, ctime, btime), le nombre de liens physiques et les flags système (immutable, append-only, etc.).

### iii. Module « Analyse de l'organisation physique »

Ce module cartographie la disposition réelle du fichier sur le disque. Il est chargé de :

- Identifier et lister l'ensemble des extents utilisés par le fichier ;
- Extraire pour chaque extent son offset logique, sa longueur en blocs et les numéros de blocs physiques correspondants sur le disque.

### iv. Module « Calcul de fragmentation »

Ce module évalue le niveau de dispersion du fichier sur le support de stockage. Il est chargé de :

- Calculer le taux de fragmentation en analysant le nombre d'extents et leur distribution sur le disque ;
- Fournir une interprétation qualitative du niveau de fragmentation (faible, modéré, élevé).
- Afficher le taux final sous forme de pourcentage.

### v. Module « Affichage du contenu »

Ce module permet de visualiser le contenu interne du fichier analysé. Il est chargé de :

- Afficher le contenu brut des fichiers texte ou lisibles de manière formatée ;
- De visualiser les informations analysées sous forme de panneaux ou onglets.
- Détecter les fichiers binaires et proposer un affichage hexadécimal ou signaler leur nature non affichable directement.

### vi. Module « Interface utilisateur graphique (GUI) »

Ce module constitue le point d'interaction entre l'utilisateur et l'outil d'analyse. Il est chargé de :

- Fournir une interface graphique intuitive permettant la sélection du fichier à analyser ;
- Organiser et présenter les résultats d'analyse de manière structurée (onglets, sections) et permettre leur exportation (texte, PDF, HTML).

### vii. Module « Gestion des erreurs et logs »

Ce module assure la robustesse et la traçabilité de l'application. Il est chargé de :

- Capturer et gérer les exceptions liées à l'accès aux fichiers ou aux structures EXT4 ;
- Afficher des messages d'erreur explicites à l'utilisateur et enregistrer les logs d'utilisation pour faciliter le débogage.

## 2. Besoins Non Fonctionnels

Les besoins non fonctionnels définissent les contraintes et les qualités que doit respecter l'outil.

### i. Performance

L'outil doit garantir des temps de réponse rapides, avec une analyse de fichier de taille moyenne (< 100 Mo) effectuée en moins de 5 secondes, tout en maintenant une interface réactive même pour les fichiers volumineux.

### ii. Compatibilité

L'outil sera développé en Python et fonctionnera exclusivement sur Debian 13.1.0 (Trixie) avec support du système de fichiers EXT4, nécessitant des privilèges root pour accéder aux structures bas niveau.

### iii. Fiabilité

L'outil doit fonctionner en mode lecture seule sans jamais modifier ni corrompre les fichiers analysés, tout en gérant de manière robuste les erreurs d'accès, les permissions insuffisantes et les structures de fichiers corrompues.

### iv. Utilisabilité

L'interface graphique doit être intuitive et accessible aux utilisateurs sans expertise technique avancée, avec des messages d'erreur clairs, une documentation complète et une aide contextuelle intégrée.

### v. Maintenabilité

Le code Python doit suivre les conventions PEP 8, être structuré en modules indépendants, correctement commenté et documenté pour faciliter les évolutions futures et l'ajout de nouvelles fonctionnalités.

### vi. Portabilité

L'application sera développée en Python 3.x en utilisant exclusivement des bibliothèques disponibles dans les dépôts officiels Debian pour garantir une installation et un déploiement simplifiés sans dépendances externes complexes.

## V. ACTEURS DU PROJET

Il sera question ici de présenter les acteurs du projet :

Tableau 1: Acteurs du projet

Dénomination	Fonction	Status	Rôles
<b>M. NGUIMBUS Emmanuel</b>	Maitre d'ouvrage	Enseignant	Définir les besoins du projet Orientation, supervision du projet
<b>SONGMENE LADO Belviane</b>	Maitre d'œuvre	Etudiante	Réalisation et mise en œuvre du projet

## VI. ESTIMATION DU COUT DU PROJET

Pour la réalisation de ce projet nous devons avoir à notre disposition des ressources matérielles, logicielles et humaines.

### 1. Ressources matérielles

Tableau 2: Ressources Matérielles

Matériels	Quantité	Utilité	Prix Unitaire (Fcfa)	Prix Total (Fcfa)
<b>Disques durs 1Tb</b>	1	-Utiliser pour les sauvegardes du projet.	40 000	40 000
<b>Laptop Core i5 Siem generation minimum 8Gb RAM 300GO SSD</b>	2	-Utiliser pour la conception et la mise en œuvre de File Analyzer	300 000	600 000
<b>Clé USB 64Go</b>	2	Utiliser pour le partage de fichiers	5000	10 000
<b>Serveur de développement</b>	1	Serveur pour tester et exécuter l'analyseur de fichiers	250 000	250 000

<b>Connexion Internet</b>	2	Accès à internet pour télécharger des outils de développement et mettre à jour les systèmes.	30 000	60 000
<b>Total</b>				<b>960 000</b>

## 2. Ressources logicielles

Tableau 3: Ressources Logicielles

Logiciels	Description	Prix Unitaire (Fcfa)	Prix Total (Fcfa)
<b>Système d'exploitation Debian 13.1.0</b>	Debian est un système libre et gratuit.	0	0
<b>VirtualBox</b>	Logiciel de virtualisation	0	0
<b>Git</b>	Outil de gestion de version.	0	0
<b>Total</b>			<b>0</b>

## 3. Ressources humaines

La réalisation d'un tel système nécessite l'intervention d'un analyste concepteur pour penser comment le système doit fonctionner, présenter les fonctionnalités possibles à développer.

Tableau 4: Ressources humaines

Rôle	Nombre	Durée de travail en jours	Montant (par jour) en Fcfa	Montant Total (Fcfa)
<b>Analyste concepteur</b>	1	20	80 000	1 600 000
<b>Chef de projet (suivi continu)</b>	1	55	90 000	4 950 000
<b>Développeur</b>	2	35	62 000	2 170 000
<b>Total</b>	<b>4</b>	<b>55</b>		<b>8 720 000</b>

## VII. PLANIFICATION DU PROJET

La **planification d'un projet** est l'activité qui consiste à ordonner les tâches à réaliser, qui ont été formalisées lors de la structuration du projet, estimer les charges associées, déterminer les profils nécessaires à leur réalisation. Tout activité étant toujours limitée dans le temps, il est préférable de planifier notre travail afin d'éviter les pertes de temps. Dans le cadre de notre planning scolaire, notre planning s'articule autour de :

### 1. Les différentes tâches

Tableau 5: Tableau des tâches

Tâches	Date de début	Date de fin	Durée (en jours)
<b>Etude du projet</b>	06/12/2025	13/12/2025	08
<b>Analyse</b>	14/12/2025	20/12/2025	07
<b>Conception</b>	21/12/2025	31/12/2025	11
<b>Réalisation</b>	01/01/2026	20/01/2026	20
<b>Déploiement</b>	21/01/2026	28/01/2026	08
<b>Clôture</b>	29/01/2026	31/01/2026	03

### 2. Diagramme de Gantt

Le diagramme de Gantt couramment utilisé en gestion de projet, est un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet.

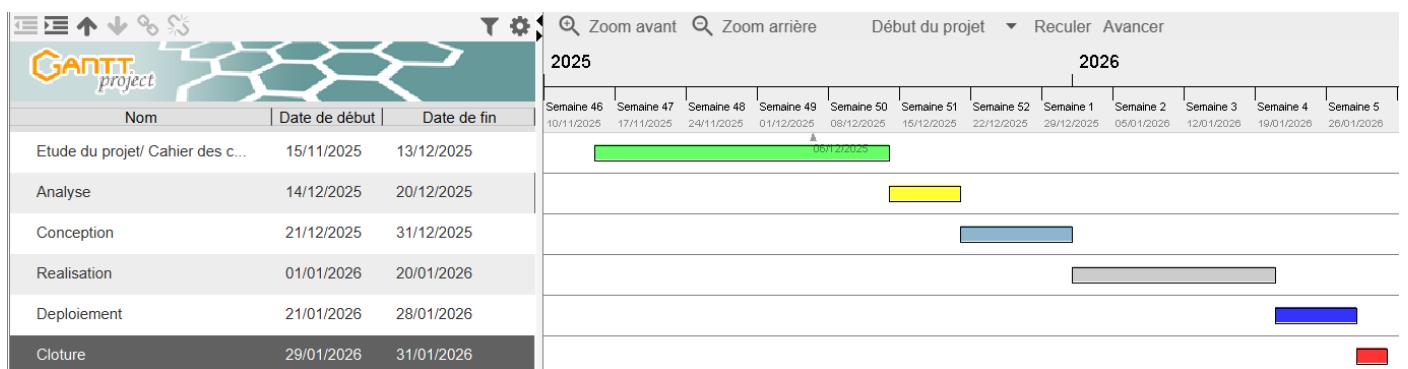


Figure 1: Diagramme de Gantt

## VIII. LES CONTRAINTES DU PROJET

### 1. Les contraintes de coût

Pour mener à terme la réalisation de notre projet, il sera indispensable de dépenser une somme totale de **9 680 000 FCFA** correspondant aux différentes ressources y intervenant.

### 2. Les contraintes de délais

La réalisation de notre projet devra être effectuée sur une période s'étalant sur 08 semaines de travail allant du **06 Décembre 2025** au **31 Janvier 2025**.

## IX. LES LIVRABLES

A la fin de ce projet, en plus du cahier des charges nous devons fournir un :

- Dossier d'analyse
- Dossier de conception
- Dossier de réalisation
- Rapport de déploiement
- Rapport du projet

# CONCLUSION

Au terme de notre cahier des charges, il est visible que nous avons une vue globale mais aussi détaillée de tout ce qu'il faut comme ressources pour la réalisation de ce projet. Ces ressources sont d'ordre matérielle, logicielle, et humaine. Toutefois, une estimation du plan financier étant déjà connue, il nous revient maintenant d'approcher la prochaine partie qui est le dossier d'analyse.

## PARTIE 2: CAHIER D'ANALYSE

# INTRODUCTION

Le projet d'un analyseur avancé de fichier **File Analyzer** vise à simplifier et à rendre accessible l'analyse détaillée des fichiers stockés sur une partition EXT4 sous Debian. Ce cahier d'analyse présente une étude approfondie des besoins fonctionnels et non fonctionnels du système, les méthodes et langages de modélisation utilisés ainsi que les différents diagrammes d'analyse pour mieux comprendre le fonctionnement de l'application.

# I. CONTEXTE ET OBJECTIF

## 1. Contexte

Avec l'évolution des technologies de stockage, les systèmes de fichiers modernes offrent désormais un accès aléatoire, permettant de récupérer ou de modifier rapidement n'importe quelle donnée à partir de son emplacement. Un système de fichiers constitue ainsi un composant essentiel d'un système d'exploitation : il assure l'organisation, le stockage, la gestion et l'accès aux données sur un support tel qu'un disque dur, une clé USB ou une mémoire flash. Il joue également un rôle d'interface entre les applications et le matériel, rendant possibles des opérations comme la création, la lecture, l'écriture ou la suppression de fichiers. Parmi les systèmes de fichiers largement utilisés dans les environnements Linux, **EXT4** (*Extended filesystem-version4*) occupe une place de référence grâce à sa robustesse, ses performances et ses fonctionnalités avancées, notamment la gestion des extents, la journalisation et la réduction de la fragmentation.

## 2. Objectif

L'objectif principal, qui est le pourquoi de notre projet est **de Faciliter la compréhension et l'analyse approfondie des systèmes de fichiers EXT4** en fournissant un outil unifié permettant d'explorer de manière claire et accessible la structure interne, les métadonnées et l'organisation physique des fichiers sous Debian.

# II. PERIMETRE FONCTIONNEL

- DéTECTer automatiquement le type du fichier analysé (texte, binaire, exécutable, image, PDF, etc.) ;
- Extraire et afficher tous les champs du header du fichier (magic number, sections, métadonnées structurelles) ;
- Extraire les métadonnées stockées dans l'inode du fichier (permissions, propriétaire, groupe, taille logique et physique) ;
- Récupérer les horodatages complets (atime, mtime, ctime, btime), le nombre de liens physiques et les flags système (immutable, append-only, etc.).
- Identifier et lister l'ensemble des extents utilisés par le fichier ;
- Extraire pour chaque extent son offset logique, sa longueur en blocs et les numéros de blocs physiques correspondants sur le disque.
- Afficher le contenu brut du fichier lorsqu'il s'agit d'un fichier texte ou lisible, tout en indiquant clairement les fichiers binaires non affichables.
- Calculer le taux de fragmentation en analysant le nombre d'extents et leur distribution sur le disque ;

- Fournir une interprétation qualitative du niveau de fragmentation (faible, modéré, élevé).
- Afficher le taux final sous forme de pourcentage.
- Afficher le contenu brut des fichiers texte ou lisibles de manière formatée ;
- De visualiser les informations analysées sous forme de panneaux ou onglets.
- Déetecter les fichiers binaires et proposer un affichage hexadécimal ou signaler leur nature non affichable directement.

### III. MODELISATION UML

#### 1. Choix de la méthode d'analyse

Au regard de la complexité croissante des systèmes d'information et suite à cette étude comparative, UML s'impose comme le langage le mieux adapté à notre projet. En tant que langage de modélisation unifié, UML permet de spécifier, visualiser, construire et documenter de manière précise l'architecture d'un système. Grâce à ses différents **diagrammes**, UML offre une description à la fois statique, dynamique et fonctionnelle du système. Il s'avère idéal pour :

- **Concevoir et structurer** une architecture logicielle orientée objet, en proposant des formalismes standards et universels ;
- **Modéliser les données** via les diagrammes de classes (utilisés ici pour représenter les entités), palliant ainsi certaines limites des modèles relationnels traditionnels par une approche plus proche du code ;
- **Formaliser les besoins métiers** à travers les diagrammes de cas d'utilisation, permettant de définir clairement les interactions entre les utilisateurs et le futur système.

## 2. Diagramme d'analyse : Diagramme de cas d'utilisation

### a. Diagramme de cas d'utilisation global du système

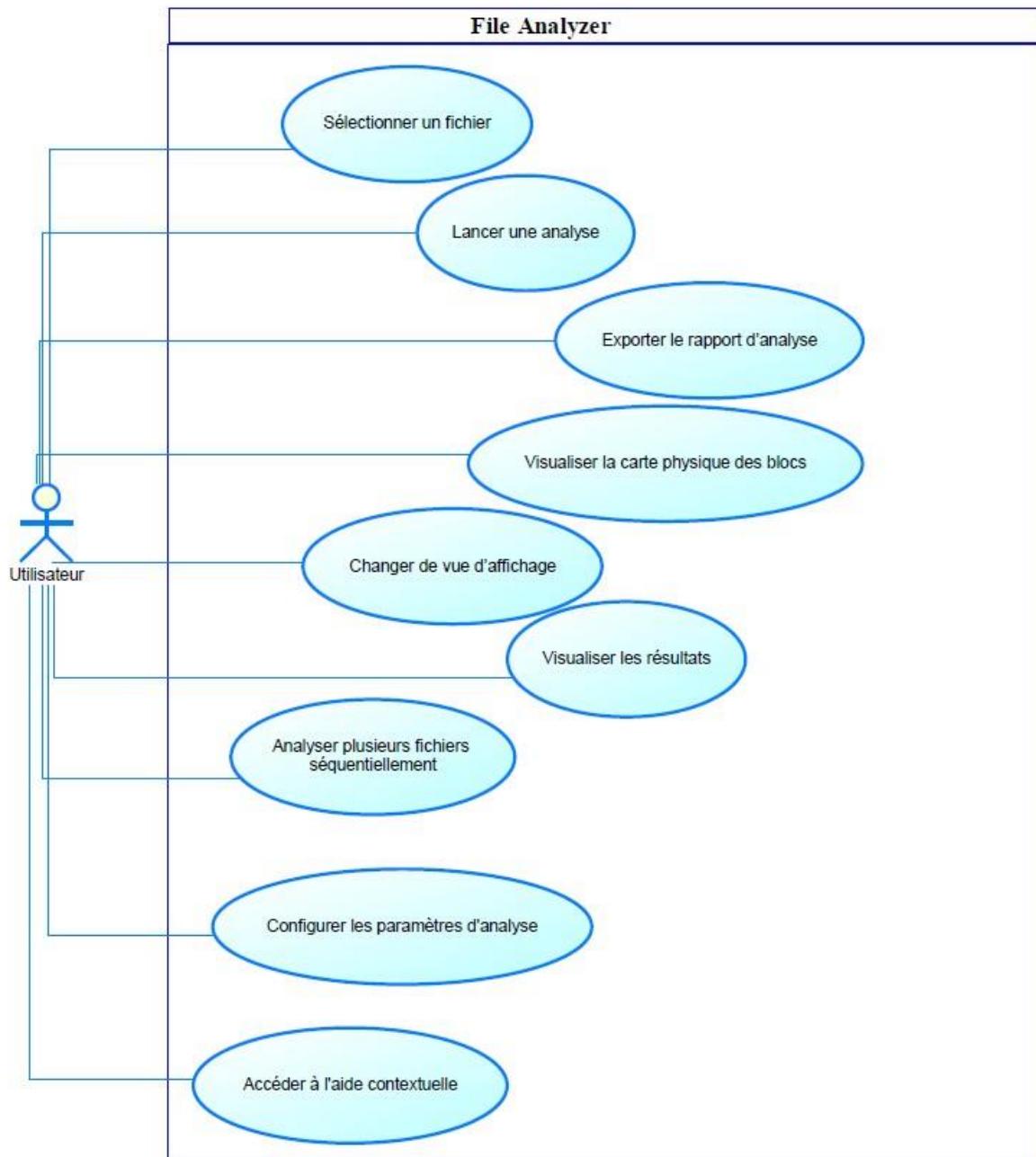


Figure 2: Diagramme des cas d'utilisation

### b. Description textuelle des cas d'utilisation

## 1. CU 1 – Sélectionner un fichier

Tableau 6: CUI-Selectionner un fichier

<b>Titre</b>	Sélectionner un fichier à analyser				
<b>Objectif</b>	Charger un fichier depuis le système de fichiers EXT4 pour préparation à l'analyse.				
<b>Résumé</b>	L'utilisateur navigue dans l'arborescence et choisit un fichier via l'interface graphique.				
<b>Acteur(s)</b>	Utilisateur				
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>			
Le système fonctionne sous Debian avec partition EXT4.	File Analyzer est lancé, interface chargée.	Le fichier est référencé, prêt pour analyse.			
<b>Scénario nominale</b>	<ol style="list-style-type: none"> <li>1. Utilisateur clique sur "Parcourir".</li> <li>2. Fenêtre de sélection s'ouvre.</li> <li>3. Il navigue vers le fichier.</li> <li>4. Il sélectionne le fichier → clique "Ouvrir".</li> <li>5. Le chemin s'affiche, bouton "Analyser" s'active.</li> </ol>				
<b>Scénario alternatif</b>	<ul style="list-style-type: none"> <li>• Fichier inaccessible → Message "Permission refusée".</li> </ul>				

## 2. CU 2 – Analyser un fichier

Tableau 7: CU2-Analyser un fichier

<b>Titre</b>	Analyser un fichier				
<b>Objectif</b>	Exécuter automatiquement tous les modules d'analyse sur le fichier sélectionné.				
<b>Résumé</b>	L'utilisateur déclenche l'exécution séquentielle des 5 modules définis.				
<b>Acteur</b>	Utilisateur				
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>			
Un fichier valide est sélectionné.	Fichier chargé, bouton "Analyser" actif.	Tous les résultats sont disponibles dans l'interface.			
<b>Scénario nominale</b>					
<ol style="list-style-type: none"> <li>1. Utilisateur clique "Analyser".</li> <li>2. Barre de progression apparaît.</li> <li>3. Modules s'exécutent dans l'ordre : a) Détection type + header, b) Extraction métadonnées, c) Cartographie physique, d) Calcul fragmentation, e) Préparation affichage contenu ;</li> <li>4. Message "Analyse terminée".</li> <li>5. Onglets résultats se remplissent.</li> </ol>					
<b>Scénario alternatif</b>					
<ul style="list-style-type: none"> <li>• Échec d'un module → Analyse continue avec mention "Échec partiel".</li> <li>• Fichier trop gros → Proposition d'analyse partielle.</li> </ul>					

### 3. CU 3 – Visualiser les résultats

Tableau 8: CU3-Visualiser les résultats

<b>Titre</b>	<b>Visualiser les résultats par catégorie</b>				
<b>Objectif</b>	Consulter les résultats d'analyse organisés par type d'information.				
<b>Résumé</b>	Navigation entre onglets pour voir différents aspects de l'analyse.				
<b>Acteur</b>	Utilisateur				
<b>Présumer</b>	<b>Précondition</b>	<b>Postcondition</b>			
Analyse préalablement effectuée.	Résultats disponibles.	Section choisie affichée à l'écran.			
<b>Scénario nominale</b>	<ol style="list-style-type: none"> <li>1. Utilisateur clique sur l'onglet "Type &amp; Header".</li> <li>2. Système affiche magic number, type détecté, structure header.</li> <li>3. Utilisateur clique sur "Métadonnées".</li> <li>4. Système affiche permissions, dates, taille, flags.</li> <li>5. Utilisateur clique sur "Organisation physique".</li> <li>6. Système affiche liste des extents avec détails.</li> </ol>				
<b>Scénario alternatif</b>	<ul style="list-style-type: none"> <li>• Onglet vide → Message "Exécutez d'abord l'analyse".</li> <li>• Données volumineuses → Pagination automatique.</li> </ul>				

#### 4. CU 4 – Exporter le rapport d'analyse

Tableau 9: CU4-Exporter le rapport d'analyse

<b>Titre</b>	<b>Exporter le rapport d'analyse</b>	
<b>Objectif</b>	Sauvegarder les résultats dans un fichier externe.	
<b>Résumé</b>	Génération d'un document contenant tout ou partie des résultats.	
<b>Acteur</b>	Utilisateur	
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>
Analyse effectuée.	Résultats disponibles.	Fichier créé sur disque.
<b>Scénario nominale</b>	<ol style="list-style-type: none"> <li>1. Utilisateur clique "Exporter".</li> <li>2. Dialogue s'ouvre : choix format (PDF/HTML/TXT).</li> <li>3. Choix contenu (complet/résumé/spécifique).</li> <li>4. Sélection emplacement.</li> <li>5. Clique "Sauvegarder".</li> <li>6. Confirmation "Export réussi".</li> </ol>	
<b>Scénario alternatif</b>	<ul style="list-style-type: none"> <li>• Espace insuffisant → Message d'erreur détaillé.</li> <li>• Échec génération → Proposition format alternatif.</li> </ul>	

## 5. CU 5 – Changer de vue d'affichage

Tableau 10: CU5-Changer de vue d'affichage

<b>Titre</b>	<b>Changer de vue d'affichage</b>	
<b>Objectif</b>	Adapter la présentation des résultats selon la préférence.	
<b>Résumé</b>	Basculer entre différentes représentations (texte, tableau, graphique).	
<b>Acteur</b>	Utilisateur	
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>
Données à afficher disponibles.	Résultats chargés.	Nouveau mode d'affichage actif.
<b>Scénario nominale</b>	<ol style="list-style-type: none"> <li>1. Dans "Organisation physique", utilisateur clique "Vue tableau".</li> <li>2. Système affiche les extents en tableau.</li> <li>3. Utilisateur clique "Vue graphique".</li> <li>4. Système affiche diagramme des blocs.</li> <li>5. Utilisateur clique "Vue texte".</li> <li>6. Système affiche description textuelle.</li> </ol>	
<b>Scénario alternatif</b>	<ul style="list-style-type: none"> <li>• Vue non disponible → Option grisée.</li> <li>• Données incompatibles → Vue par défaut.</li> </ul>	

## 6. CU 6 – Accéder à l'aide contextuelle

Tableau 11: CU6-Accéder à l'aide contextuelle

<b>Titre</b>	<b>Accéder à l'aide contextuelle</b>	
<b>Objectif</b>	Obtenir des explications sur les termes techniques.	
<b>Résumé</b>	Consultation de documentation intégrée.	
<b>Acteur</b>	Utilisateur	
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>
Documentation incluse.	Application en cours d'exécution.	Explication affichée.
<b>Scénario nominale</b>	<ol style="list-style-type: none"> <li>1. Utilisateur survole "extent" avec la souris.</li> <li>2. Info-bulle apparaît avec définition courte.</li> <li>3. Il clique sur "?" à côté de "fragmentation".</li> <li>4. Fenêtre d'aide détaillée s'ouvre.</li> <li>5. Il peut naviguer dans les sujets liés.</li> </ol>	
<b>Scénario alternatif</b>	<ul style="list-style-type: none"> <li>• Terme inconnu → "Documentation non disponible".</li> <li>• Demande hors contexte → Menu d'aide général.</li> </ul>	

## 7. CU 7 – Configurer les paramètres d'analyse

Tableau 12: CU7-Configurer les paramètres d'analyse

<b>Titre</b>	<b>Configurer les paramètres d'analyse</b>	
<b>Objectif</b>	Personnaliser le comportement de l'outil.	
<b>Résumé</b>	Modification des options et préférences.	
<b>Acteur</b>	Utilisateur	
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>
Aucun	Application lancée.	Paramètres sauvegardés.
<b>Scénario nominale</b>		
<ol style="list-style-type: none"> <li>1. Utilisateur ouvre "Paramètres".</li> <li>2. Modifie "Taille max d'affichage" à 50 Mo.</li> <li>3. Active "Toujours afficher l'hexadécimal".</li> <li>4. Désactive "Vérifier les mises à jour".</li> <li>5. Clique "Appliquer".</li> <li>6. Paramètres persistés.</li> </ol>		
<b>Scénario alternatif</b>		
<ul style="list-style-type: none"> <li>• Valeur invalide → Message d'erreur avec contrainte.</li> <li>• Redémarrage nécessaire → Avertissement.</li> </ul>		

## 8. CU 8 – Analyser plusieurs fichiers séquentiellement

Tableau 13: CU8-Analyser plusieurs fichiers séquentiellement

<b>Titre</b>	<b>Analyser plusieurs fichiers séquentiellement</b>	
<b>Objectif</b>	Analyser une série de fichiers sans réinitialisation.	
<b>Résumé</b>	Enchaînement d'analyses sur différents fichiers.	
<b>Acteur</b>	Utilisateur	
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>
Plusieurs fichiers à analyser.	Application lancée.	Résultats du dernier fichier affichés.
<b>Scénario nominale</b>	<ol style="list-style-type: none"> <li>1. Utilisateur analyse le fichier A.</li> <li>2. Il clique sur "Nouvelle analyse".</li> <li>3. Sélectionne le fichier B.</li> <li>4. Clique sur "Analyser".</li> <li>5. Interface se met à jour avec résultats de B.</li> </ol>	
<b>Scénario alternatif</b>	<ul style="list-style-type: none"> <li>• Fichier précédent non sauvegardé → Avertissement.</li> <li>• Limite mémoire atteinte → Nettoyage automatique.</li> </ul>	

## 9. CU 9 – Visualiser la carte physique des blocs

Tableau 14: CU-Visualiser la carte physique des blocs

<b>Titre</b>	Visualiser la carte physique des blocs				
<b>Objectif</b>	Voir graphiquement la répartition des données sur le disque.				
<b>Résumé</b>	Représentation visuelle des extents et de la fragmentation.				
<b>Acteur</b>	Utilisateur				
<b>Présupposer</b>	<b>Précondition</b>	<b>Postcondition</b>			
Analyse physique effectuée.	Données d'extents disponibles.	Diagramme affiché.			
<b>Scénario nominale</b>					
<ol style="list-style-type: none"> <li>1. Utilisateur ouvre onglet "Carte physique".</li> <li>2. Système génère un diagramme coloré.</li> <li>3. Chaque extent = rectangle coloré.</li> <li>4. Espaces vides = zones grises.</li> <li>5. Survol d'un bloc → infos détaillées.</li> </ol>					
<b>Scénario alternatif</b>					
<ul style="list-style-type: none"> <li>• Fichier non fragmenté → Message "Fichier contigu".</li> <li>• Trop d'extents → Vue simplifiée avec zoom.</li> </ul>					

## IV. LIVRABLES

- Dossier de conception
- Dossier de réalisation
- Rapport de déploiement
- Rapport du projet

# CONCLUSION

Le projet File Analyzer présenté dans ce cahier d'analyse ambitionne de fournir une solution complète et pédagogique pour l'analyse approfondie des fichiers sur les systèmes de fichiers EXT4 sous environnement Debian. En combinant des technologies d'analyse bas-niveau, une extraction intelligente des métadonnées et une interface graphique intuitive adaptée aux besoins des utilisateurs, cette solution contribuera à démystifier le fonctionnement interne des systèmes de fichiers et à renforcer les compétences techniques des utilisateurs. Dans la suite nous allons ressortir le dossier de conception de notre projet.

## PARTIE 3: CAHIER DE CONCEPTION

# INTRODUCTION

Dans la réalisation d'un projet informatique, le dossier de conception est celui qui présente l'architecture du système étudié. L'analyse UML nous permet à travers ses diagrammes structurels de représenter la structure de notre application ainsi que les différents composants matériels et logiciels. Dans cette partie, nous vous présenterons le diagramme de classe ainsi que le diagramme de composant.

# I. DIAGRAMMES DE SEQUENCE

## 1. SEQUENCE: Analyser un fichier

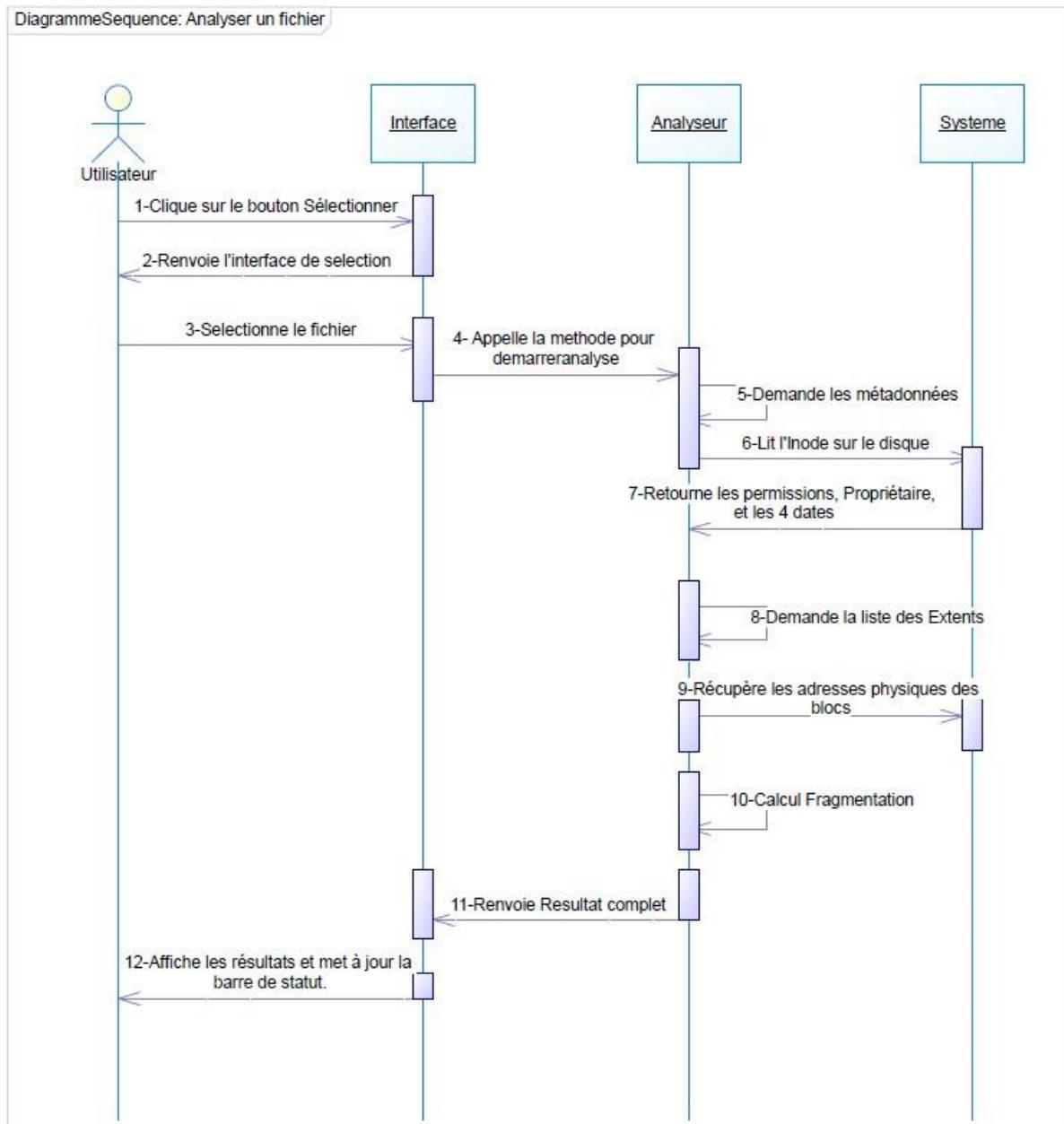


Figure 3: séquence analyser un fichier

## 2. SEQUENCE : Visualiser la carte physique des blocs

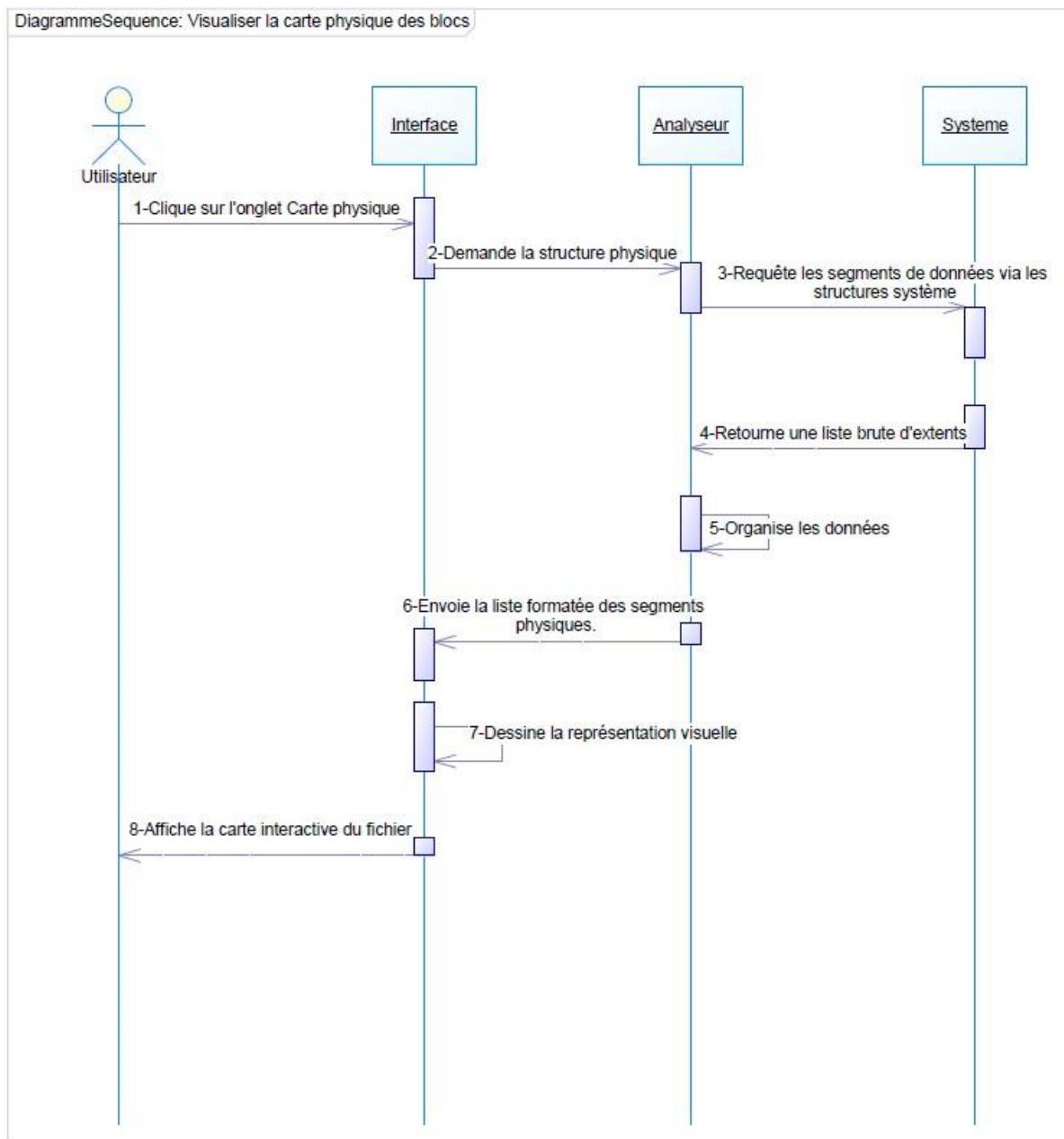


Figure 4: Séquence visualiser la carte physique des blocs

## CONCLUSION

La phase de conception est une étape primordiale dans la conception logicielle, car elle nous donne une vue globale de la structure de notre application. Après cette étape nous pouvons passer à la réalisation de notre projet.

# TABLES DES MATIERES

PROJET SYSTEME D'EXPLOITATION .....	1
PROJET SYSTEME D'EXPLOITATION .....	1
SOMMAIRE .....	i
LISTE DES TABLEAUX .....	ii
LISTE DES FIGURES .....	iii
PARTIE 1 : CAHIER DES CHARGES .....	1
INTRODUCTION.....	2
I. CONTEXTE ET JUSTIFICATION DU PROJET .....	3
1. Contexte .....	3
2. Justification .....	3
II. PROBLEMATIQUE .....	4
III. LES OBJECTIFS DU PROJET .....	4
1. Objectif global.....	4
2. Objectifs spécifiques .....	4
IV. EXPRESSION DES BESOINS .....	5
1. Besoins Fonctionnels.....	5
i. Module « Détection et analyse du type de fichier » .....	5
ii. Module « Extraction des métadonnées système ».....	5
iii.     Module « Analyse de l'organisation physique ».....	6
iv.     Module « Calcul de fragmentation ».....	6
v.     Module « Affichage du contenu ».....	6
vi.     Module « Interface utilisateur graphique (GUI) ».....	6
vii.    Module « Gestion des erreurs et logs » .....	6
2. Besoins Non Fonctionnels.....	7
i. Performance .....	7
ii. Compatibilité.....	7
iii.    Fiabilité.....	7
iv.    Utilisabilité .....	7
v.    Maintenabilité.....	7
vi.    Portabilité .....	7

V.	ACTEURS DU PROJET.....	8
VI.	ESTIMATION DU COUT DU PROJET .....	8
1.	Ressources matérielles .....	8
2.	Ressources logicielles .....	9
3.	Ressources humaines.....	9
VII.	PLANIFICATION DU PROJET .....	10
1.	Les différentes tâches .....	10
2.	Diagramme de Gantt .....	10
VIII.	LES CONTRAINTES DU PROJET .....	11
1.	Les contraintes de coût.....	11
2.	Les contraintes de délais .....	11
IX.	LES LIVRABLES.....	11
	CONCLUSION .....	12
	PARTIE 2: CAHIER D'ANALYSE .....	13
	INTRODUCTION.....	14
I.	CONTEXTE ET OBJECTIF .....	15
1.	Contexte .....	15
2.	Objectif.....	15
II.	PERIMETRE FONCTIONNEL.....	15
III.	MODELISATION UML .....	16
1.	Choix de la méthode d'analyse .....	16
2.	Diagramme d'analyse : Diagramme de cas d'utilisation .....	17
a.	Diagramme de cas d'utilisation global du système .....	17
b.	Description textuelle des cas d'utilisation.....	17
1.	CU 1 – Sélectionner un fichier .....	18
2.	CU 2 – Analyser un fichier .....	19
3.	CU 3 – Visualiser les résultats.....	20
4.	CU 4 – Exporter le rapport d'analyse.....	21
5.	CU 5 – Changer de vue d'affichage .....	22
6.	CU 6 – Accéder à l'aide contextuelle .....	23
7.	CU 7 – Configurer les paramètres d'analyse .....	24
8.	CU 8 – Analyser plusieurs fichiers séquentiellement.....	25

9. CU 9 – Visualiser la carte physique des blocs.....	26
IV. LIVRABLES .....	27
CONCLUSION .....	28
PARTIE 3: CAHIER DE CONCEPTION .....	29
INTRODUCTION.....	30
I. DIAGRAMMES DE SEQUENCE .....	31
1. SEQUENCE: Analyser un fichier .....	31
2. SEQUENCE : Visualiser la carte physique des blocs .....	32
CONCLUSION .....	33
TABLES DES MATIERES .....	34