
# CAHIER D'ANALYSE

**THEME: CONCEPTION ET REALISATION DE BOOT INSPECTOR, UN OUTIL D'ANALYSE DU DEMARRAGE D'UNE DISTRIBUTION LINUX: Cas de Manjaro**

Rédigé par:

- TCHINDA METEWOUE Florelle Ecladore
- OMGBA ESSENGUE Patrice

**ENCADRANT: M. NGUIMBUS Emmanuel**

**Master 1 Sécurité des Systèmes d'Information 2025/2026**

# PLAN :

1. INTRODUCTION
2. CONTEXTE ET OBJECTIFS
3. PERIMETRE FONCTIONNEL
4. MODELISATION UML
5. CONCLUSION

# INTRODUCTION

Ce cahier d'analyse a pour objectif de détailler les besoins et les fonctionnalités du projet de **CONCEPTION ET REALISATION DE BOOT INSPECTOR, UN OUTIL D'ANALYSE DU DEMARRAGE D'UNE DISTRIBUTION LINUX : Cas de Manjaro.** Ce projet vise à Développer un outil capable de **surveiller**, **analyser**, **diagnostiquer** et **visualiser** l'ensemble des étapes du processus de démarrage d'un système **Manjaro Linux**, afin d'optimiser ses performances et d'identifier efficacement les anomalies.

FR

# OBJECTIFS

1. Compréhension et visibilité du démarrage
2. Diagnostic rapide et automatisé
3. Optimisation des performances
4. Amélioration de l'expérience utilisateur par
   une automatisation complète du processus de collecte de
   données et une interface intuitive

# PERIMETRE FONCTIONNEL

FR

**1. Module d'Analyse du Démarrage du Système :**

- Analyse du **temps global de démarrage**
- Décomposition du démarrage en **phases** :
  - Firmware (UEFI/BIOS)
  - Bootloader (GRUB)
  - Kernel
  - Initramfs
  - Userspace (systemd)
- Identification des **phases lentes** du boot

**2. Module de Visualisation du Démarrage :**

- Affichage d'une **timeline interactive du démarrage;**
- Visualisation graphique des **phases du boot;**
- Représentation visuelle des **durées de démarrage;**
- Graphiques interactifs :
  - timelines
  - histogrammes
  - diagrammes circulaires

# PERIMETRE FONCTIONNEL

FR

**3. Module d'Analyse des Services systemd :**

- Affichage de la **liste complète des services systemd;**
- Mesure du **temps d'exécution de chaque service;**
- Identification automatique des **services lents (blame);**
- Analyse de la **chaîne critique systemd;**
- Visualisation graphique des dépendances entre services.

**4. Module de Diagnostic des Anomalies du Démarrage :**

- Détection automatique :
  - erreurs système
  - avertissements (warnings)
  - timeouts
  - blocages
  - erreurs du noyau (kernel panics)
- Analyse des **journaux systemd et kernel** de manière **automatique** en arrière-plan dès le lancement de l'outil, sans intervention manuelle de l'utilisateur**;**
- Corrélation des anomalies avec : les services systemd , les phases du démarrage, Accès ciblé aux **logs filtrés;**

# PERIMETRE FONCTIONNEL

**5. Module d'Optimisation des Performances du Démarrage :**

- **Recommandations basées sur des règles d'analyse internes** (par exemple, si un service dépasse X secondes, suggérer sa désactivation ou sa parallélisation).**;**
- Détection des **tendances de dégradation** du temps de démarrage;
- Aide à la prise de décision :
  - désactivation de services non essentiels
  - amélioration de la parallélisation
  - ajustement de l'initramfs
- Support à la **maintenance préventive**

**6. Module de Reporting et d'Export :**

- Génération de **rapports d'analyse**
- Export des résultats aux formats :
  - HTML
  - JSON
- Conservation des analyses pour :
  - documentation
  - audit
  - suivi des performances

# DESCRIPTION TEXTUELLE D'UN CAS D'UTILISATION(MODÉLISATION UML )

# Description textuelle des cas d'utilisations :

## Analyser les données de démarrage

| **Titre :** | Analyser les données de démarrage |
|---|---|
| Objectif : | Permettre à l'utilisateur d'obtenir une analyse complète et visuelle de son processus de démarrage Linux pour l'optimiser. |
| Résumé : | L'application **collecte automatiquement** les logs et données de démarrage (systemd-analyze, journalctl,etc) au lancement, puis les analyse pour afficher des graphiques, des listes de services et génère des **recommandations basées sur ses règles internes**. |
| Acteur : | **Principal** : Utilisateur<br>**Secondaire** : **Système Linux, SGBD Local (stockage des données).** |

| **Présupposé** | **Précondition** | **Post condition** |
|---|---|---|
| L'application est installée localement sur le système Manjaro cible et dispose des permissions nécessaires (ex: via pkexec ou sudo transparent) pour exécuter les commandes système en arrière-plan | • L'utilisateur est sur la page /analysis.<br>• L'application a terminé la **collecte automatique** des données brutes. | • Un rapport d'analyse visuel est affiché.<br>• Des **recommandations pertinentes** sont affichées.<br>• Le résultat de l'analyse est **enregistré dans le SGBD**. |

# Description textuelle des cas d’utilisations : Analyser les données de démarrage

FR

| Scénario nominal |
| --- |
| 1. L'utilisateur lance l'application "Boot Inspector".<br>2. L'application affiche un indicateur de chargement ("Collecte et analyse en cours...").<br>**3. Le système exécute automatiquement** les commandes système (systemd-analyze, journalctl, dmesg, etc) en arrière-plan et valide les données collectées.<br>4. (Include) L'application analyse les données pour en extraire les informations pertinentes (temps, services, logs).<br>5. Les visualisations de base (graphiques, listes) sont immédiatement affichées.<br>6. Le système exécute le **module de règles internes** pour générer des suggestions d'optimisation.<br>7. Les recommandations s'affichent dans l'interface.<br>8. L'analyse complète est **enregistrée dans le SGBD** (base de données locale) pour l'historique. |

| Scénario alternatif |
| --- |
| **A - Données invalides** : L'utilisateur soumet des données trop courtes. Le système affiche une erreur de validation et s'arrête.<br>**B - Erreur de parsing** : Les données sont mal formatées. Le système affiche une notification d'échec et présente les sections qu'il a pu analyser.<br>**C - Échec du module de règles internes** : L’exécution du processus de consultation des métriques de démarrage pour les suggestions d’optimisation échoue. Seule l'analyse de base (sans les recommandations) est affichée et sauvegardée. |

# Description textuelle des cas d’utilisations: Exporter un rapport d'analyse

FR

| | |
|---|---|
| **Titre :** | Exporter un rapport d'analyse |
| Objectif : | Permettre à l'utilisateur de sauvegarder le rapport d'analyse actuel dans un fichier local pour archivage, partage ou consultation hors ligne |
| Résumé : | L'utilisateur sélectionne un format de fichier (JSON, HTML, PDF) depuis un menu, ce qui déclenche la génération et le téléchargement du rapport correspondant. |
| Acteur : | **Principal** : Utilisateur |

| **Présupposé** | **Précondition** | **Post condition** |
|---|---|---|
| Le navigateur web de l'utilisateur est configuré pour autoriser le téléchargement de fichiers générés côté client (via JavaScript). | • Une analyse a été effectuée et son rapport est actuellement affiché dans l'interface.<br>• Le bouton "Exporter le rapport" est visible et actif. | • Un fichier contenant les données du rapport dans le format choisi est sauvegardé sur l'ordinateur de l'utilisateur.<br>• L'état de l'application reste inchangé après l'export. |

## Scénario nominal

1. L'utilisateur clique sur le bouton "Exporter le rapport" pour ouvrir le menu déroulant des options.
2. L'utilisateur clique sur l'option "PDF" dans le menu.
3. Le système active la fonction exportToPdf.
4. La librairie html2canvas capture le contenu du rapport affiché et le convertit en une image (canvas).
5. La librairie jspdf prend cette image et l'insère dans un nouveau document PDF.
6. Le système déclenche le téléchargement du fichier PDF généré, avec un nom de fichier contenant la date et l'heure de l'analyse (ex: boot-inspector-report-2023-10-27T10:30:00.000Z.pdf).
7. Le navigateur de l'utilisateur affiche une boîte de dialogue pour enregistrer le fichier.

## Scénario alternatif

**Scénario A : Aucune analyse affichée**

Pour une raison quelconque, l'utilisateur tente d'exporter alors qu'aucun rapport n'est affiché.
Le clic sur une option d'export ne produit aucun effet car la fonction correspondante s'arrête immédiatement (la condition if (!analysisResult) return; est vraie).

**Scénario B : Échec de la génération du fichier (pour PDF/HTML)**

L'utilisateur clique sur "PDF".
La librairie html2canvas rencontre une erreur lors de la capture de la page (par exemple, à cause de contenu non supporté ou de restrictions de sécurité du navigateur).
L'application intercepte l'erreur.
Une notification d'erreur s'affiche à l'écran ("Échec de l'exportation PDF").
Aucun fichier n'est téléchargé.

# Description textuelle des cas d’utilisations: Consulter l'historique d'analyse

FR

| | |
|---|---|
| **Titre :** | Consulter l'historique d'analyse |
| Objectif : | Permettre à l'utilisateur de revoir et de réafficher une analyse précédemment effectuée sans avoir à saisir à nouveau les données. |
| Résumé : | L'utilisateur sélectionne une analyse passée depuis une liste, récupérée depuis le SGBD local. |
| Acteur : | **Principal** : Utilisateur |

| **Présupposé** | **Précondition** | **Post condition** |
|---|---|---|
| Le SGBD local est accessible et configuré. | L'utilisateur a déjà effectué et enregistré au moins une analyse dans le SGBD. | • Le rapport complet de l'analyse sélectionnée (graphiques, listes, recommandations d’optimisation) est affiché dans la zone de contenu principale.<br>• L'application est prête pour une nouvelle analyse ou une nouvelle sélection dans l'historique. |

# Description textuelle des cas d’utilisations : Consulter l'historique d'analyse

FR

**Scénario nominal**

1. Le système charge la liste des analyses sauvegardées depuis le SGBD et les affiche dans la barre latérale de l'historique, groupées par date (Aujourd'hui, Hier, etc.).
2. L'utilisateur identifie l'analyse qu'il souhaite revoir dans la liste.
3. L'utilisateur clique sur l'entrée correspondante (identifiée par son heure).
4. Le système récupère l'objet de données complet de cette analyse depuis son état interne (le state React).
5. Le système met à jour la zone d'affichage principale pour rendre les composants (BootSummary, ServiceList, etc.) avec les données de l'analyse sélectionnée.
6. Le rapport s'affiche instantanément.

**Scénario alternatif**

**Scénario A : Aucun historique n'est sauvegardé**

Le système tente de charger l'historique depuis le SGBD mais n'en trouve aucun.
La barre latérale affiche un message indiquant "Aucun historique récent trouvé".
L'utilisateur ne peut pas interagir avec la fonctionnalité.

**Scénario B : L'historique est corrompu ou invalide**

Le système tente de lire les données mais celles-ci sont corrompues.
L'application intercepte l'erreur de parsing.
La barre latérale affiche un état vide ou un message d'erreur discret.
L'historique est considéré comme vide, et la fonctionnalité n'est pas disponible

FR

# Modélisation UML Diagramme de Cas d'Utilisations :

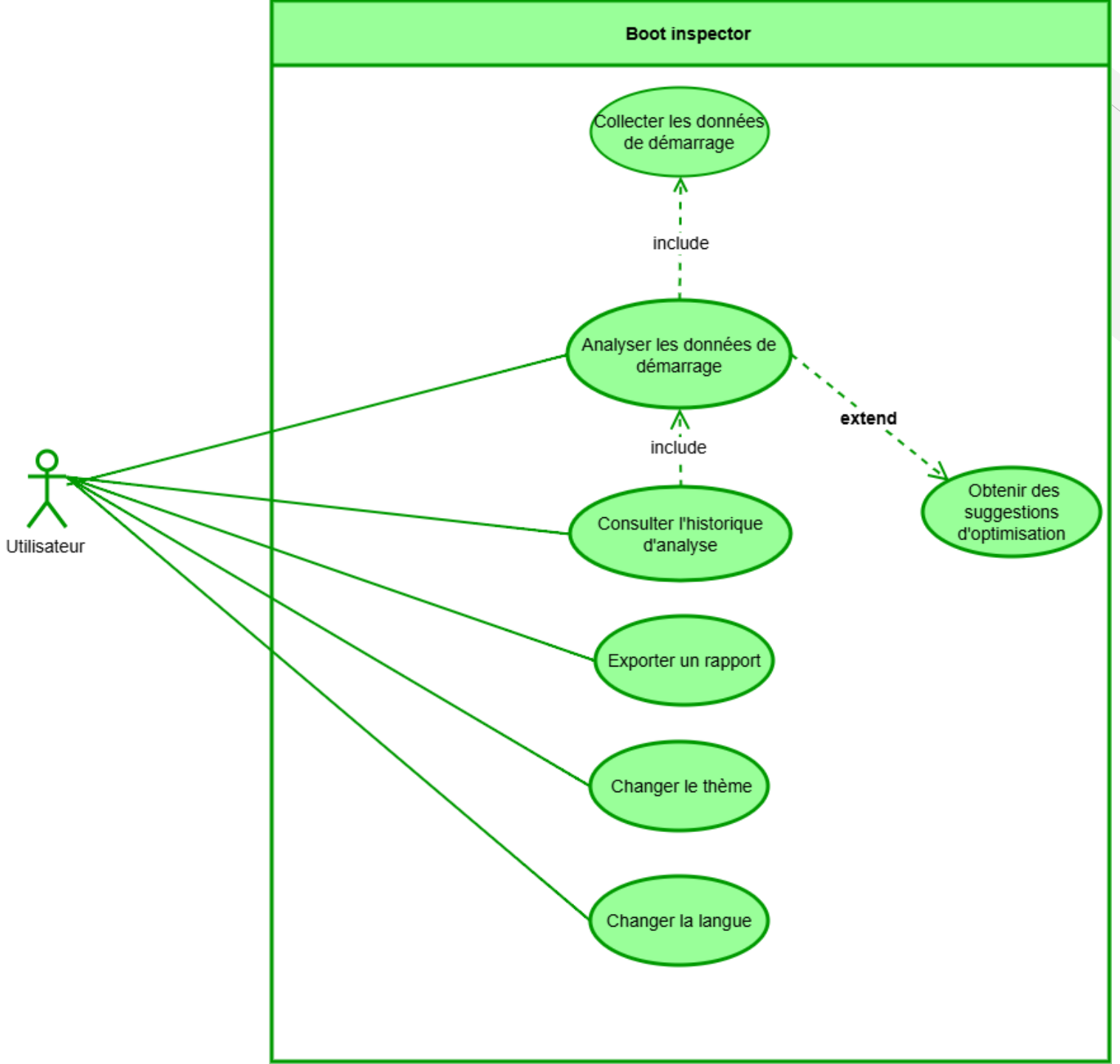
Acteurs :

- Utilisateur : La personne (administrateur système, développeur) qui utilise l'application pour optimiser le démarrage de son système Linux. Il est l'initiateur de toutes les actions principales.
- (Acteurs secondaires, non-humains) : Système Linux : Représente le système d'exploitation cible **avec lequel l'application interagit directement** (via des appels système/API internes) pour **collecter automatiquement** les données de démarrage nécessaires à l'analyse.

Cas d'Utilisation :

- Collecter les données de demarage
- Analyser les données de démarrage
- Obtenir des suggestions d'optimisation
- Consulter l'historique des analyses
- Exporter un rapport d'analyse
- Personnaliser l'interface
- Modifier la langue

# DIAGRAMME USE CASE GLOBAL




FR

FR

# CONCLUSION

Ce cahier d'analyse a permis d'identifier les besoins fonctionnels et techniques du projet *Boot Inspector*. Il a mis en évidence les objectifs, le périmètre fonctionnel ainsi que les différents cas d'utilisation liés à l'analyse et à l'optimisation du démarrage d'un système Linux. Cette phase constitue une base solide pour la conception de l'application et garantit que la solution développée répondra efficacement aux attentes des utilisateurs.