

PROJET S4 2025 - Cahier des charges

Groupe Outta Control Rebel - Seekly

27 janvier 2025



Table des matières

1	Introduction	3
2	Le projet	3
2.1	L'équipe Outta Control Rebel	3
2.2	Origine	4
3	État de l'art	5
4	Objectifs	6
4.1	- Recherche de fichiers	6
4.2	- Recherche avancée	6
4.3	- Interface utilisateur	6
4.4	- Indexation	6
4.5	- Performances	7
4.6	- Personnalisation / Bonus	7
5	Utilisation de l'application	8
5.1	- Interface en ligne de commande (CLI)	8
5.2	- Interface graphique (GUI)	8
6	Découpage du projet	10
6.1	- Répartition des tâches	10
6.2	- Prévisions	10
7	Moyens mis en oeuvre	11
7.1	- Moyens humains	11
7.2	- Moyens matériels	11
7.3	- Moyens techniques	11
7.4	- Documentation et communication	11
8	Conclusion	12

1 Introduction

Nom du projet : Seekly

Type de projet : Application de recherche de fichiers et contenus, similaire à Spotlight, écrit en Rust pour un environnement Linux.

Objectif principal : Créer une application rapide, efficace et minimaliste permettant aux utilisateurs de rechercher facilement des fichiers, dossiers, et informations spécifiques. Elle vise à simplifier leur expérience en éliminant le besoin de naviguer manuellement dans le système de fichiers, tout en leur offrant des fonctionnalités avancées pour optimiser leur productivité

Notions clés : recherche avancée, interface intuitive, indexation performante, optimisation de la productivité, personnalisation des recherches, sécurité et rapidité grâce à Rust.

2 Le projet

2.1 L'équipe Outta Control Rebel

Le groupe Outta Control Rebel est composé de quatre jeunes passionnés par l'informatique, les jeux vidéos et l'intelligence artificielle :

- **KHAMCHANE Hamza**

Je m'appelle Hamza, j'ai 19 ans et je viens Drancy. Le monde de l'informatique est monde qui ma toujours fasciné que ce soit dans sa complexité ou son charme. Je suis attirée par ce monde depuis mes 13 ans. C'est surtout grace a l'influence des mes 3 oncles maternelle que j'ai decouvert ce monde. Depuis cette age j'ai essayer de decouvrir ce monde par plusieurs moyen, un stage , des recherche personnelles, programmation de mon cotée etc. Et de fil en aiguille je me suis retrouve aujourd'hui a Epita.

- **RICHARD Aubin**

Je suis passionné depuis plusieurs années par l'informatique. Le lycée est le moment où l'informatique est devenu encore plus importante pour moi. J'ai commencer à apprendre différents langage de programation et également à réaliser de mini-projets. J'ai ensuite décidé de rejoindre Epita pour continue de faire et d'apprendre l'informatique.

- **SUILLEROT Gaetan**

Mon intérêt pour l'informatique a commencé au collège avec Scratch. Au lycée, la création de projets en équipe est devenue une passion. J'ai choisi l'EPITA Paris pour sa réputation internationale. Mon rôle au sein de l'équipe me tient à cœur. Ce projet me fera découvrir de nouveaux domaines et enrichir mes compétences.

- VIDAL-LEFEBVRE Oscar

Je m'appelle Oscar, ma passion pour l'informatique remonte à mes histoires du soir quand j'étais enfant. L'intérêt que j'y porte vient des possibilités qu'elle offre, particulièrement dans l'électronique qui est le domaine dans lequel j'aimerais m'orienter. J'ai choisi de m'investir dans ce groupe pour leurs idées originales, leur assiduité et régularité qui m'ont particulièrement séduit.

2.2 Origine

L'idée de Seekly est née d'une analyse approfondie des besoins des utilisateurs Linux en matière de recherche de fichiers. Lors de nos recherches initiales, nous avons constaté que de nombreux outils de recherche existants manquaient de rapidité, d'ergonomie ou de flexibilité, surtout sur les systèmes ayant une grande quantité de fichiers. Inspirés par des solutions comme Spotlight sur macOS, nous avons cherché à combiner la simplicité d'utilisation avec les capacités avancées offertes par des technologies modernes.

Nous avons commencé par analyser les points faibles des outils actuels comme "locate" et "find" sous Linux, qui, bien qu'efficaces, ne permettent pas toujours une expérience utilisateur fluide. Cela nous a conduit à l'idée de développer une application reposant sur un moteur d'indexation puissant, tout en offrant une interface moderne et personnalisable. Rust a été choisi comme langage principal pour sa performance, sa gestion de mémoire sécurisée et sa communauté active, ce qui garantit la pérennité de notre projet.

3 État de l’art

Pour comprendre les besoins et lacunes actuelles dans le domaine des outils de recherche de fichiers, nous avons étudié plusieurs solutions existantes :

- Locate/Find (Linux) :

Ces outils sont largement utilisés dans l’écosystème Linux pour rechercher des fichiers. Cependant, ils souffrent d’un manque d’ergonomie et de fonctionnalités avancées. Par exemple, ils ne permettent pas une mise à jour en temps réel des modifications du système de fichiers et nécessitent souvent des commandes complexes peu accessibles aux utilisateurs non techniques.

- Spotlight (macOS) :

Spotlight est une référence en termes d’ergonomie et de rapidité. Il combine une interface utilisateur fluide avec des résultats de recherche rapides grâce à son système d’indexation performant. Toutefois, ce logiciel est exclusif à l’écosystème Apple, laissant un vide dans l’univers Linux.

- Catfish (Linux) :

Catfish est une solution graphique basée sur les commandes find et locate. Bien qu’intuitive, elle reste limitée en termes de performances pour des systèmes de fichiers contenant plusieurs millions de fichiers.

- Recoll (Linux) :

Cet outil offre une recherche full-text avancée mais peut être complexe à configurer pour les utilisateurs non expérimentés. Sa rapidité dépend également de la configuration matérielle et de la taille des index.

L’analyse des outils actuels nous a permis d’identifier les besoins suivants, une interface simple et moderne, accessible à tous les utilisateurs, un système d’indexation performant permettant une mise à jour en temps réel, une prise en charge des filtres avancés pour des recherches complexes et une solution open-source conçue spécifiquement pour l’écosystème Linux.

4 Objectifs

4.1 - Recherche de fichiers

- Offrir une recherche rapide et intuitive permettant de localiser des fichiers par leur nom, leur extension, ou des mots-clés spécifiques.
- Permettre la recherche dans des répertoires précis ou sur l'ensemble du système de fichiers pour maximiser la flexibilité.
- Inclure des options permettant de trier et filtrer les résultats en fonction de critères personnalisés comme l'ordre alphabétique ou la pertinence.

4.2 - Recherche avancée

- Proposer des filtres précis tels que la taille des fichiers, la date de dernière modification ou encore leur type.
- Intégrer la prise en charge des expressions régulières, offrant ainsi aux utilisateurs expérimentés la possibilité d'effectuer des recherches complexes et ciblées.
- Ajouter une fonction de recherche approximative pour corriger automatiquement les fautes de frappe dans les requêtes.

4.3 - Interface utilisateur

- Concevoir une interface en ligne de commande (CLI) qui soit conviviale, épurée et simple d'utilisation pour les utilisateurs avancés et débutants.
- Concevoir une interface graphique (GUI) pour une accessibilité accrue auprès des utilisateurs moins familiers avec les outils en ligne de commande.
- Intégrer un retour visuel clair, comme une barre de progression pour les recherches longues, ou un affichage des suggestions basées sur les recherches précédentes.

4.4 - Indexation

- Développer un moteur d'indexation robuste capable de traiter efficacement des systèmes de fichiers contenant plusieurs millions d'éléments.
- Mettre en place un mécanisme de mise à jour en temps réel de l'index pour refléter instantanément les changements effectués sur le système de fichiers.
- Prévoir une gestion optimisée des ressources pour minimiser l'impact de l'indexation sur les performances globales du système.

4.5 - Performances

- Assurer une recherche quasi-instantanée même dans des environnements où la densité de fichiers est élevée.
- Exploiter pleinement les capacités offertes par Rust pour garantir une exécution rapide et une gestion de mémoire sûre.
- Tester et optimiser l'application pour les environnements à faibles ressources (ordinateurs plus anciens ou serveurs modestes).

4.6 - Personnalisation / Bonus

- Permettre aux utilisateurs de configurer facilement des chemins de recherche exclus ou prioritaires pour affiner les résultats selon leurs besoins.
- Proposer des raccourcis personnalisables permettant d'accéder rapidement aux recherches fréquentes ou aux paramètres favoris.
- Inclure un mode « sombre » et des options de thèmes pour une meilleure expérience visuelle adaptée à tous les goûts.

5 Utilisation de l'application

L'application Seekly est conçue pour offrir une expérience utilisateur intuitive et efficace. Voici un aperçu de son fonctionnement, accompagné de croquis représentant l'interface utilisateur (CLI et GUI).

5.1 - Interface en ligne de commande (CLI)

L'interface CLI est conçue pour être rapide et accessible aux utilisateurs avancés. Voici un exemple de commande et son résultat :

```
seekly search --name "rapport.pdf" --directory /home/user/docume
```

Résultat attendu :

```
Recherche en cours...
Fichier trouvé : /home/user/documents/rapport.pdf
Taille : 2.3 Mo
Dernière modification : 12/01/2025
```

Croquis de l'interface CLI :

```
+-----+
| Seekly CLI - Recherche de fichiers |
+-----+
> Commande : seekly search --name "exemple.txt"

[Résultat 1] /home/user/exemple.txt
[Résultat 2] /home/user/documents/exemple_rapport.txt
```


5.2 - Interface graphique (GUI)

Une interface graphique est prévue pour les utilisateurs qui préfèrent une approche visuelle. Elle comportera les éléments suivants :

- Une barre de recherche en haut pour entrer des mots-clés.
- Des options de filtrage sur le côté gauche (taille, date de modification, type de fichier).
- Une liste des résultats affichés au centre avec des détails sur chaque fichier.

Croquis de l'interface GUI :

Seekly



fichier_1.exe - /home/user/bin/

fichier_2.png - /home/user/images

fichier_3.txt - /local/tmp/

fichier_4.wav - /usb/dev/media

Cette interface sera développée en utilisant des bibliothèques comme egui ou tui pour garantir une intégration fluide et rapide sur les distributions Linux.

6 Découpage du projet

6.1 - Répartition des tâches

	Gaetan	Hamza	Oscar	Aubin
CLI	Second	Main		
GUI			Main	Second
Site Web	Main		Second	
Design		Second		Main
Algorithmme	Main		Second	

6.2 - Prévisions

	soutenance 1	soutenance 2	soutenance 3
CLI	65%	100%	100%
GUI	0%	60%	100%
Site Web	75%	95%	100%
Design	0%	60%	100%
Algorithmme	45%	85%	100%

7 Moyens mis en oeuvre

Pour mener à bien le projet Seekly, l'équipe s'appuie sur une combinaison de technologies, d'outils et de compétences adaptés au développement d'une application performante et innovante. Voici un aperçu des moyens techniques, humains et matériels mobilisés :

7.1 - Moyens humains

- Équipe projet : Composée de quatre membres ayant des compétences complémentaires en programmation, gestion de projet.
- Réunions hebdomadaires : Organisation de points réguliers pour assurer l'avancement du projet et résoudre les éventuels problèmes.
- Formation continue : Chaque membre est encouragé à approfondir ses connaissances sur Rust, les moteurs d'indexation, et les outils associés.

7.2 - Moyens matériels

Ordinateurs personnels des membres de l'équipe équipés de distributions Linux.
Utilisation de logiciels de gestion de versions (Git).

7.3 - Moyens techniques

Utilisation de Rust, choisi pour ses performances et sa gestion sécurisée de la mémoire.
Bibliothèques et outils tiers : Nous allons utiliser différentes bibliothèques pour pouvoir réaliser une interface graphique propre. Egaleme nt nous utiliserons cargo.

7.4 - Documentation et communication

Documentation technique rédigée avec LaTeX.

Documentation utilisateur incluant des guides pratiques pour l'installation et l'utilisation de l'application.

Outils de communication comme Discord pour coordonner l'équipe et gérer les tâches.

Ces moyens combinés assurent une gestion efficace du projet, tout en garantissant la qualité et la pérennité de l'application.

8 Conclusion

Le projet Seekly est la construction d’une application de recherche de fichiers dans l’écosystème Linux. Ce cahier des charges constitue le fondement du développement de cette application et met en avant une méthodologie rigoureuse et une vision claire des objectifs.

En combinant des technologies modernes comme Rust, des bibliothèques performantes, et une approche utilisateur centrée sur l’ergonomie, Seekly se démarquera par sa rapidité, sa précision, et sa flexibilité. Notre équipe a pris soin d’identifier les lacunes des outils existants pour concevoir une solution sur mesure adaptée aux besoins des utilisateurs, qu’ils soient novices ou expérimentés.

Au-delà de l’aspect technique, ce projet est aussi une opportunité pour les membres de l’équipe de renforcer leurs compétences en gestion de projet, programmation, et travail collaboratif. Chacun des membres apporte des compétences uniques, et la répartition claire des rôles garantit une progression efficace.

Grâce à une organisation bien définie, des moyens matériels adaptés, et une motivation commune. Nous visons à livrer une application robuste, performante, et extensible, qui pourra continuer à évoluer et à répondre aux futurs besoins de la communauté open-source.

Ce cahier des charges servira de guide et de référence tout au long du projet, assurant que chaque étape est accomplie avec succès et que le résultat final reflète notre engagement technique et utilisateur.