



Rapport de conception

Projet Fil Rouge (PFR) – Phase 02 – 2023

UPSSIWEB – Julien PINQUIER

Equipe : Flavie THIBAUT – Emilien VESIN – Bastien LALANNE – DEGBEGNI
Andy – Alexis GIBERT – Elio GENSON



Maven™



Sommaire

- 1 UPSSIWEB.....4
 - 1.1 Les équipes.....4
 - 1.1.1 Equipe conception C vers JAVA4
 - 1.1.2 Equipe UI design.....4
 - 1.1.3 Equipe JAVA.....4
 - 1.2 Environnement logiciel de développement5
 - 1.3 Environnement organisationnel.....5
- 2 Architecture générale du logiciel (Quels sont les acteurs du système et qui fait quoi).....6
 - 2.1 Architecture initiale.....6
 - 2.2 Alternative JAVA.....6
- 3 Description générale du logiciel.....7
 - 3.1 Cas d'utilisation7
 - 3.2 Diagramme de classe7
 - 3.3 Description détaillée du logiciel.....9
- 4 Interface graphique (IHM)11
 - 4.1 Version prototype (Simple fenêtre).....11
 - 4.2 Version UI design (Multi fenêtre).....11



Introduction

Le Projet Fil Rouge (PFR) est un projet mis en place au sein de la 1ère année de formation à l'UPSSITECH de Toulouse. Fédéré par Isabelle FERRANE, Julien PINQUIER et Julien VANDERSTRAETEN. Celui-ci permet aux étudiants d'aborder de nombreux aspects techniques, organisationnels et professionnels à travers un projet transversal pour mettre en œuvre les différentes compétences scientifiques et techniques de la spécialité SRI telles que l'informatique, l'analyse de contenu (audio, image et texte), l'interaction, et enfin, la robotique.

Il permet également de mettre en pratique différentes compétences de gestion de projet en mettant en relation un client (ici Julien PINQUIER) et notre équipe projet : Flavie THIBAUT, Emilien VESIN, Bastien LALANNE, Alexis GIBERT, Andy DEGBEGN et Elio GENSON.

Cette année l'objectif du projet fil rouge est de réaliser un moteur de recherche suivant trois types de fichier (texte, image, audio) tout en respectant les contraintes posées par le client. Sa conception a été découpée en 3 phases distinctes :

- Phase 1 : La conception d'une application textuelle en langage C sous UNIX.
- Phase 2 : La conception d'un bus de communication virtuel JAVA-C et d'une interface graphique afin de contrôler plusieurs moteurs de recherche.
- Phase 3 : La mise en place d'un lien entre le travail effectué et la robotique

Ainsi, ce second livrable aura pour but d'exposer les contraintes et de régir les fonctionnalités de la partie interface du projet suivant les concepts « Objets » tout en effectuant de nombreux tests logiciels pour valider son fonctionnement.

1 UPSSIWEB

UPSSIWEB est une entreprise de 6 membres dont la vision commune est de parfaire les moteurs de recherche de demain pour les rendre plus ergonomiques et plus intuitifs afin de les intégrer aux futurs équipements domotiques et robotiques.

1.1 Les équipes

1.1.1 Equipe conception C vers JAVA

Alexis GIBERT : Coordinateur projet en charge de la gestion de projet, Développeur backend JAVA-C en charge du bus virtuel, de la conception de la recherche de données textuelles et la recherche de données photographiques.

Issue d'une double formation de BTS CPI et DUT GEII, je suis capable d'intervenir sur des points techniques en mécanique, électronique et informatique. Je possède de solides compétences en développement embarqué (C), en développement orienté objet (Python) et en gestion de projet.

Andy DEGBEGNI : Développeur backend C pour l'amélioration du moteur de recherche, de l'indexation de données textuelles et l'indexation de données photographiques.

J'ai suivi une CPGE MPSI et je suis titulaire d'une licence en informatique. Je suis donc en mesure d'intervenir aussi bien sur les questions relatives à l'analyse et la modélisation de problèmes, que sur des points techniques en informatique. Je possède des compétences en programmation procédurale (langage C) et en programmation orientée objet (Python, Java, C++).

1.1.2 Equipe UI design

Bastien LALANNE : Développeur backend en charge du menu administrateur et de la conception de l'indexation et la recherche de données sonores et frontend JAVA en charge de l'IHM.

Grâce à mon DUT GEII, j'ai pu acquérir des compétences techniques dans les domaines de l'électronique et de l'informatique, mais aussi des compétences plus transversales telles que le travail d'équipe, l'organisation ou encore dans la gestion de projet. Je possède également de solides connaissances en langage C. Durant mon stage j'ai pu acquérir de l'autonomie, mais aussi du relationnel.

Flavie THIBAULT : Coordinatrice client & Développeuse backend en charge de la conception de l'indexation et la recherche de données photographiques et frontend JAVA en charge de l'IHM.

Après l'obtention d'un bac STL spécialité physique chimie je suis rentrée en classe préparatoire aux grandes écoles TSI. Ces deux ans en CPGE m'ont permis d'acquérir une grande capacité de travail. Je possède des compétences en mathématiques, en physique, en algorithmique et en gestion de projet.

1.1.3 Equipe JAVA

Emilien VESIN : Responsable de l'intégration, développeur backend en charge du menu général et de la conception de l'indexation et la recherche de données sonores.

En 2020 j'ai obtenu mon bac S option sciences de l'ingénieur spécialité physique chimie. Puis j'ai poursuivi ma formation en DUT GEII à Toulouse où j'ai pu approfondir mes compétences en langage C ainsi qu'en organisation au travers de différents projets. Mon stage de fin d'année en entreprise m'a également permis de développer mon travail en autonomie et en équipe.

Elio GENSON : Développeur backend JAVA en charge de la recherche complexe

Après avoir obtenu mon BAC STL, j'ai décidé de poursuivre mes études en obtenant un BTS CIRA (Contrôle Industriel et Régulation Automatique), ce qui m'a permis de développer des compétences en régulation, asservissement et automatisation des procédés. Par la suite, j'ai intégré la classe préparatoire ATS option sciences de l'ingénieur de Déodat de Séverac, ce qui m'a permis d'acquérir une solide capacité de travail ainsi que d'autres compétences essentielles pour le métier d'ingénieur.

1.2 Environnement logiciel de développement

Les outils logiciels suivants (voir Tableau 1) seront utilisés par tous les membres de l'équipe.

Type d'outil	Nom de l'outil	Version de l'outil
Machine virtuelle (VM)	Virtual Box	6.1.40 r154048
Système d'exploitation	Linux/Ubuntu	22.04.1 LTS
Compilateur C	GCC	11.3.0
Makefile	GNU Make	4.3
JDK	OpenJDK	19.0.2
Framework	OpenJFX	11.0.11
Gestion de projet logiciel	Maven	3.6.3
Environnement de développement	Visual Studio Code	1.73.1
Extension	Doxygen Documentation Generator	V1.4.0
Extension	C/C++	V1.13.4
Extension	C/C++ Extension Pack	V1.3.0
Extension	GitHub Pull Requests and Issues	V0.54.1
Extension	Live Share	V1.0.5776
Extension	JAVA Extension Pack	
Gestionnaire de configuration local	Git	2.34.1
Gestionnaire de configuration Web	Github	/

Tableau 1 : Environnement logiciel de développement

1.3 Environnement organisationnel

La gestion de projet sera réalisée sous un Trello partagé au sein de l'équipe et mis à jour régulièrement (au minimum 1 fois par semaine) lors de la réunion hebdomadaire avec l'équipe. Il permettra de fixer les objectifs au sein de l'équipe et d'identifier les dates de livraisons (jalons) soit :

- Un diagramme de **GANTT interne** qui détaille les micro-tâches (fonctions ou groupe de fonctions) pour faciliter l'intégration, donner des objectifs chiffrés à chacun, identifier les éventuels dépassement et retard.
- Un diagramme de **GANTT client** issu du GANTT interne qui détaille uniquement les groupes de tâches et les livrables qui permettront au client de se rendre compte de l'avancement du projet.

Tout au long du projet nous intégrerons une démarche de conception en suivant un **cycle en V** (voir figure 1).

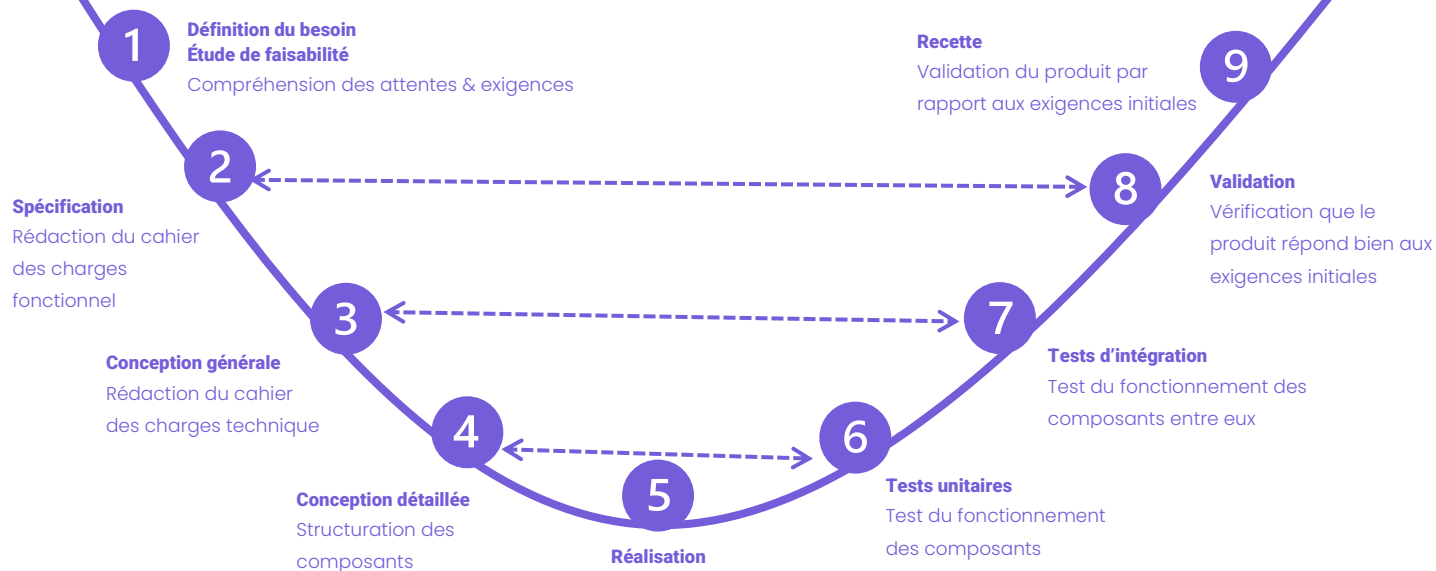


Figure 1 : Cycle en V

2 Architecture générale du logiciel (Quels sont les acteurs du système et qui fait quoi)

2.1 Architecture initiale

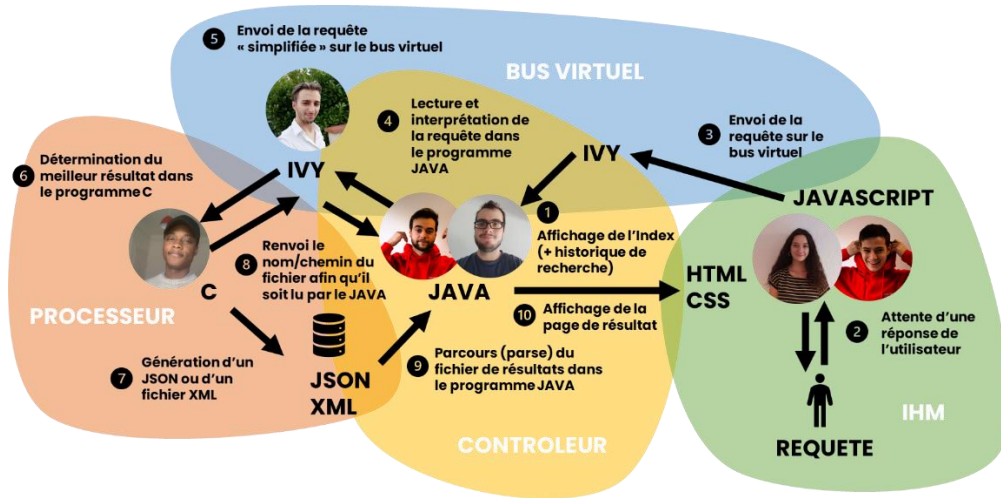


Figure 2 : Architecture 1.0

Une première idée serait de partir sur la création d'une IHM web en HTML, CSS et Javascript. Plus simple à mettre en place, elle nous permettrait d'utiliser le potentiel de la bibliothèque logicielle IVY. De plus, bien que l'ouverture et la fermeture de fichier de résultats JSON génèrera un temps de recherche long, cela reste une éventualité facile à mettre en place pour communiquer avec le JAVA. Néanmoins cette architecture ne correspond pas aux attentes du cahier des charges, nous allons donc poursuivre nos recherches pour proposer une solution plus adaptée.

2.2 Alternative JAVA

Une alternative efficace serait de construire l'interface sur SceneBuilder puis de l'intégrer via le framework open-source JavaFX afin de limiter au maximum la transmission de donnée via le bus virtuel. De plus, on peut imaginer renvoyer directement le score et le chemin du fichier via le bus virtuel sans avoir à ouvrir et fermer de fichier JSON inutilement (perte de temps).

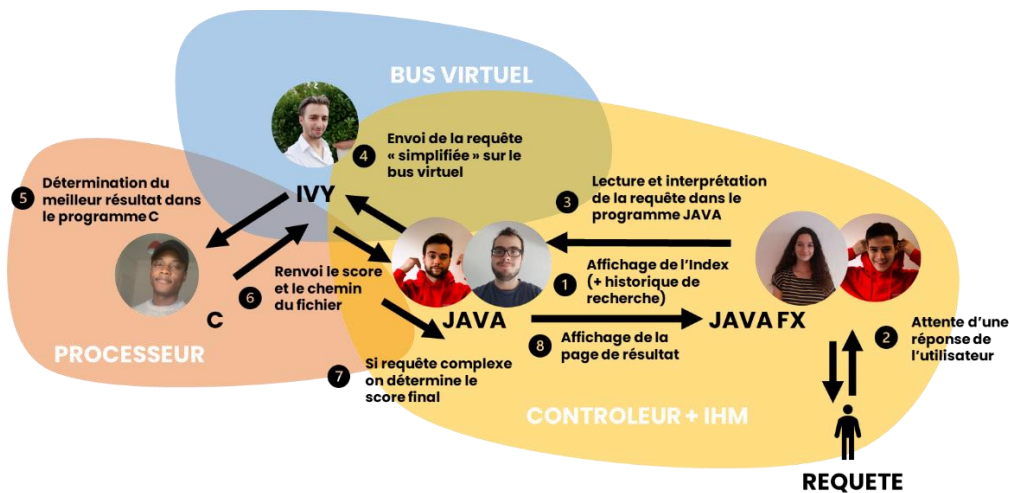


Figure 3 : Architecture 2.0

3 Description générale du logiciel

3.1 Cas d'utilisation

Intitulé du cas	Acteur	Description	Obligatoire	Commentaire
Connexion	Utilisateur, Administrateur	Connexion utilisateur ou connexion administrateur si utilisation du mot de passe	OUI	Si le mot de passe est correct, accès en mode utilisateur.
Fermeture de l'application	Utilisateur, Administrateur	Fermer l'application en fin d'utilisation	OUI	
Ouverture de l'application	Utilisateur, Administrateur	Démarrage de l'application par un utilisateur	OUI	
Recherche par mots clés	Utilisateur, Administrateur	Recherche d'un fichier en saisissant un mot clé	OUI	Le système recherche les documents répondant à la demande liée au mot clé. Affichage des résultats selon un seuil de présence du mot clé
Recherche par similitude		Recherche de fichiers par similitude à un autre fichier	OUI	Le système vérifie si le document entré comme référence existe, sinon affiche un message d'erreur. S'il existe, affichage décroissant des résultats par rapport à un seuil de similitude.
Modification des paramètres	Administrateur	Modification du seuil de similitude Activation ou désactivation du mode multimoteur Modification du mode d'indexation.	NON	Le moteur prendra en compte les modification

Tableau 2 : Cas d'utilisation

3.2 Diagramme de classe

Nous avons décidé de représenter l'architecture de notre code avec des diagrammes de classe.

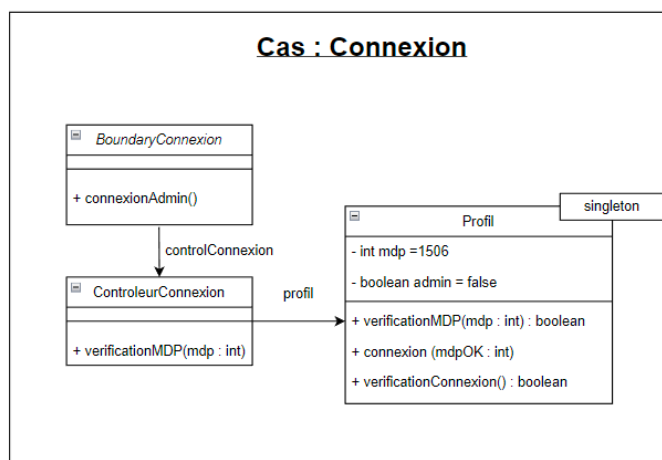


Figure 4 : Cas de la connexion

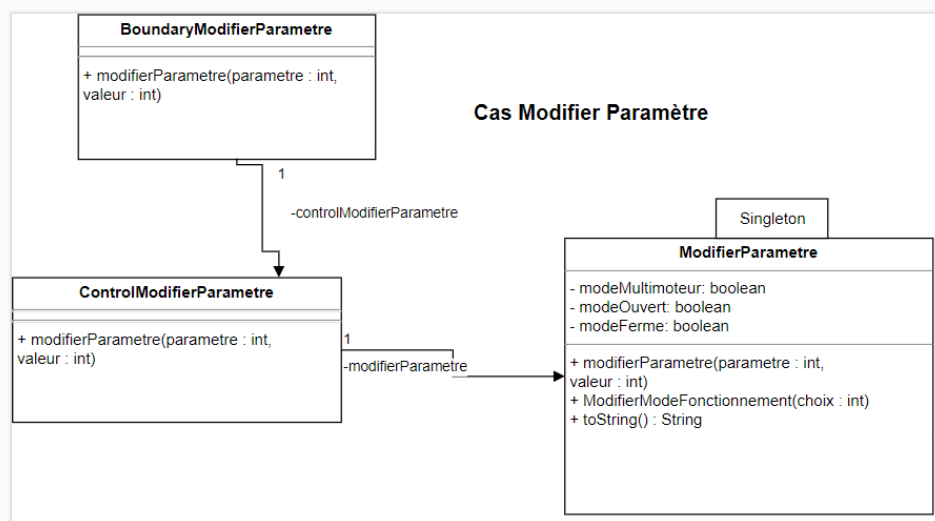


Figure 5 : Cas de la modification de paramètres

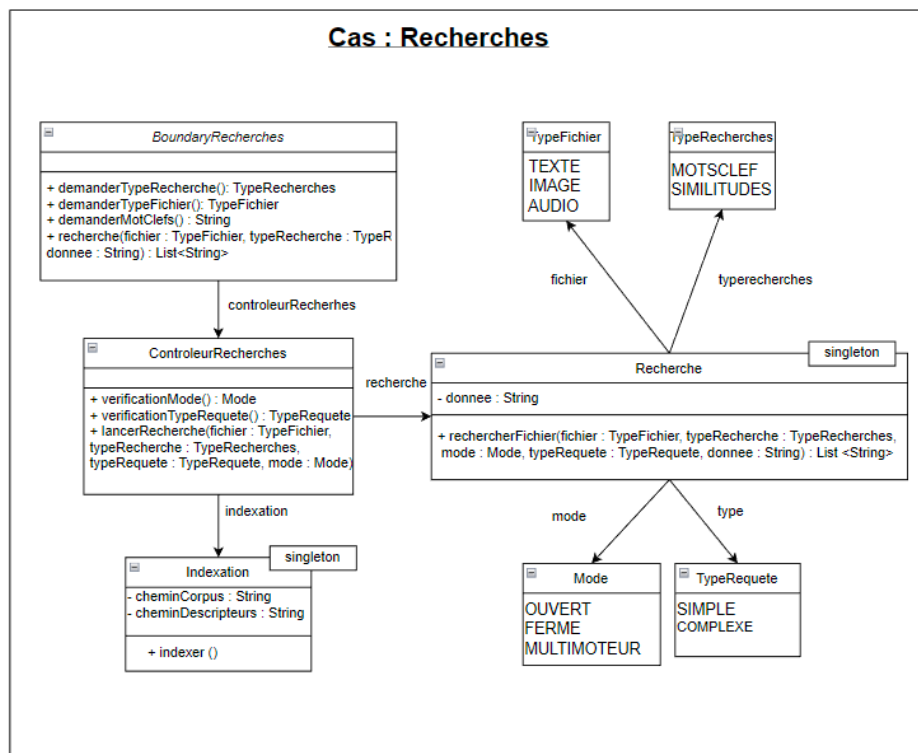


Figure 6 : Cas de la recherche

3.3 Description détaillée du logiciel

Nous avons associé à chaque diagramme de classe un diagramme de séquence système :

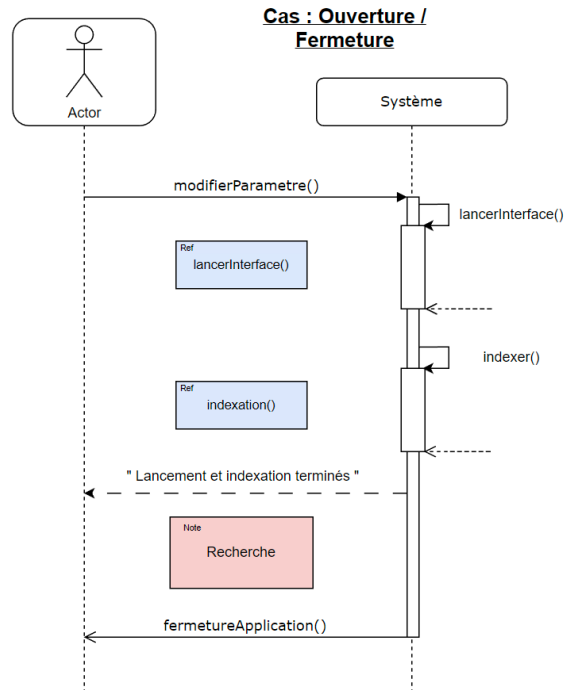


Figure 7 : Diagramme de séquence détaillé de l'ouverture / fermeture

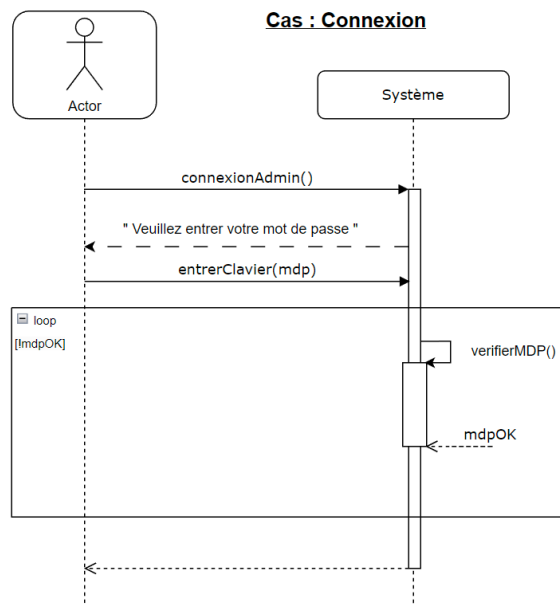


Figure 8 : Diagramme de séquence détaillé de la connexion

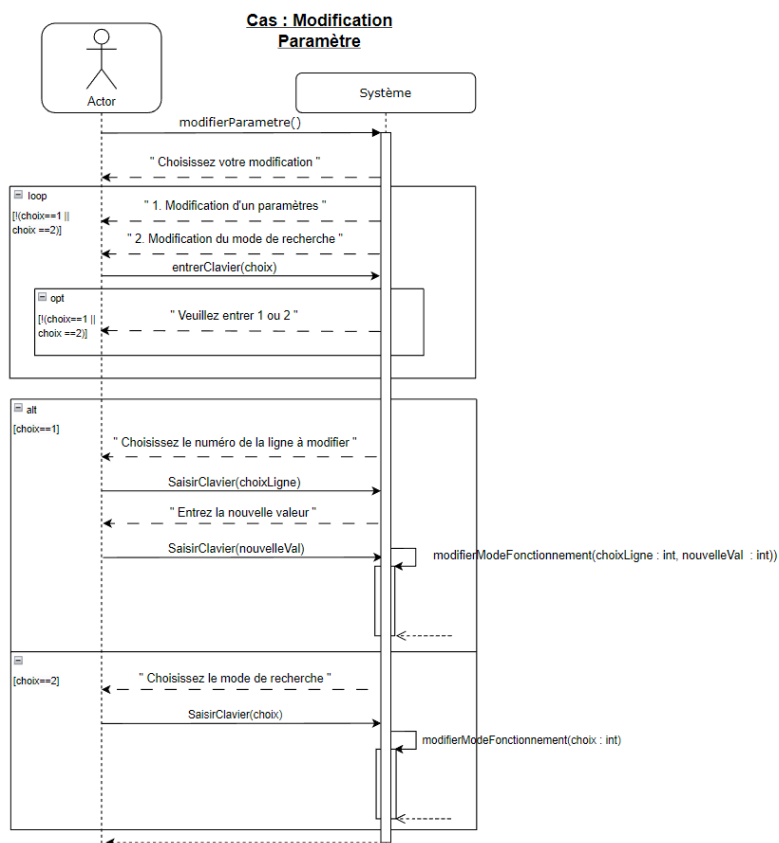


Figure 9 : Diagramme de séquence détaillé de la modification de paramètres

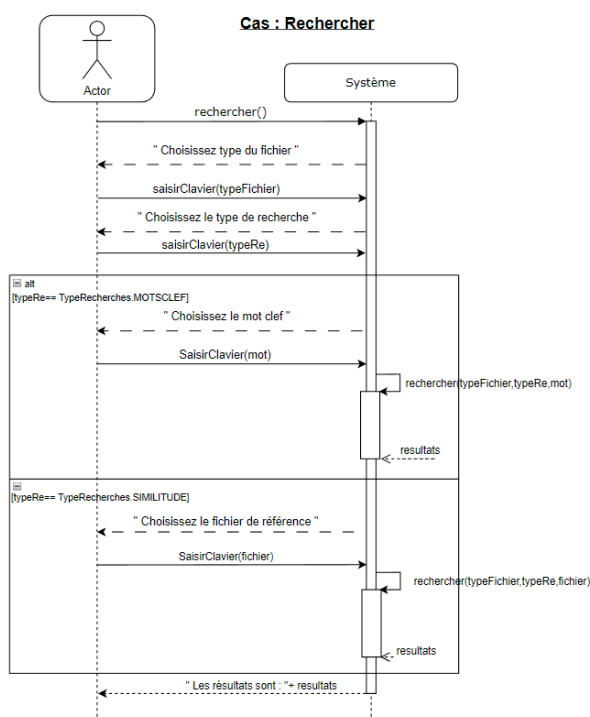


Figure 10 : Diagramme de séquence détaillé de recherche

4 Interface graphique (IHM)

4.1 Version prototype (Simple fenêtre)

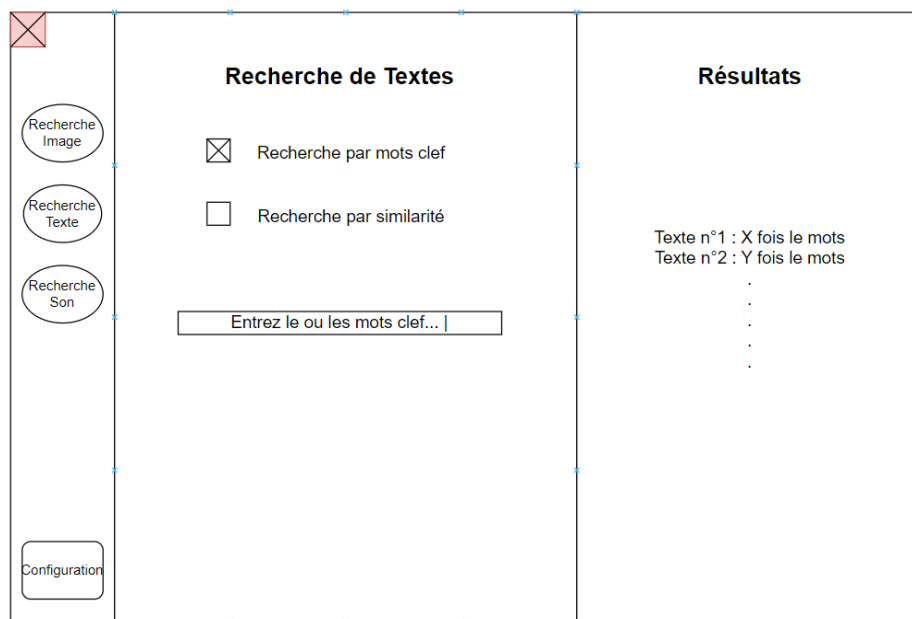


Figure 11 : Prototype 1.0 IHM

4.2 Version UI design (Multi fenêtre)

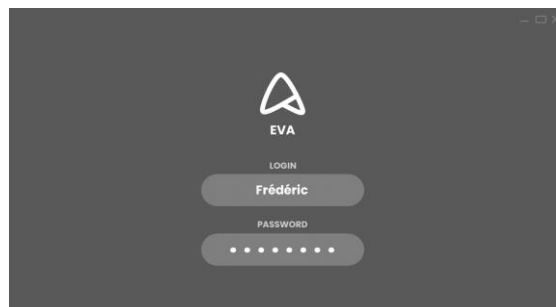


Figure 12 : Prototype 2.0 - Page d'accueil

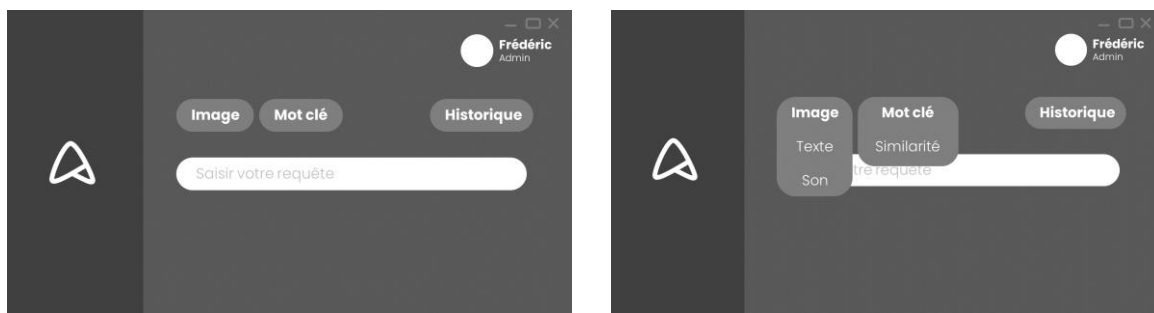


Figure 13 : Prototype 2.0 - Choix et menu déroulant

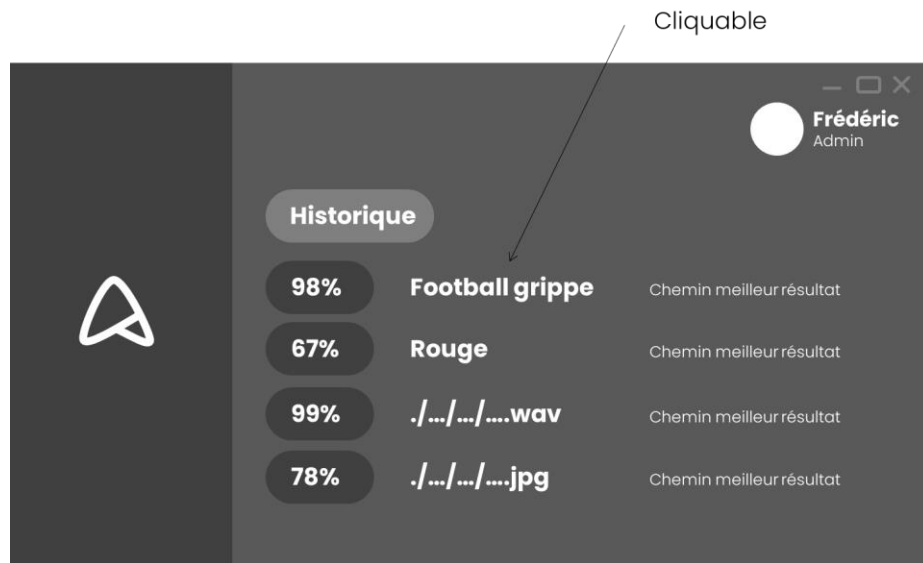


Figure 14 : Prototype 2.0 - Historique



Figure 15 : Prototype 2.0 - Paramètres

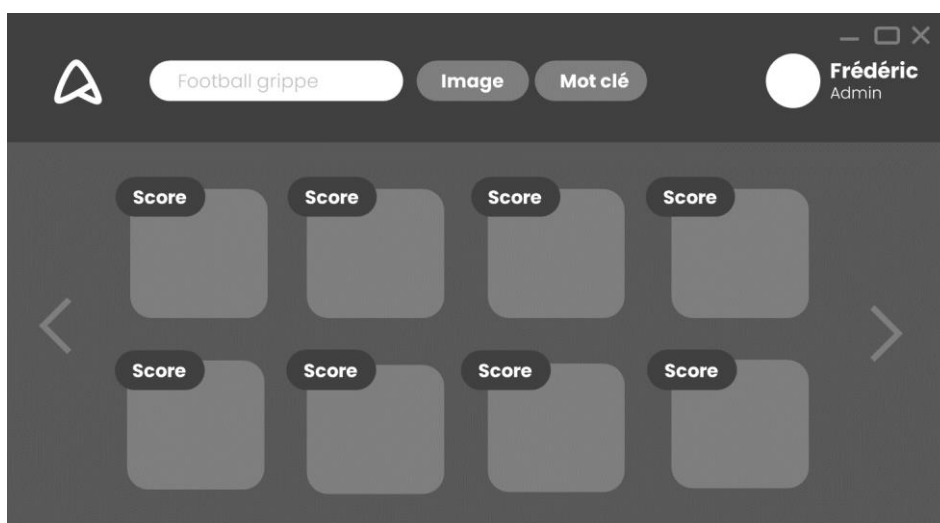


Figure 16 : Prototype 2.0 - Recherche image par mot clé

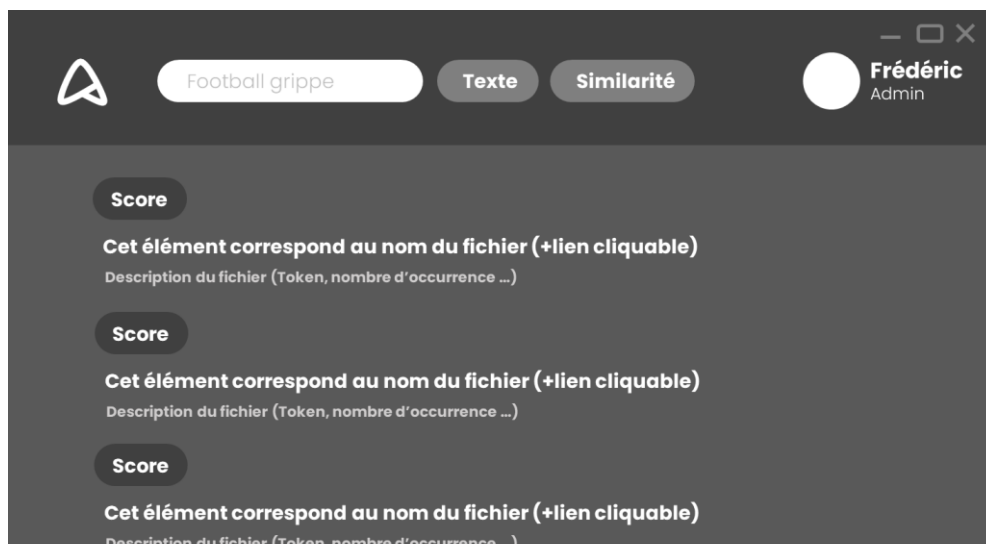


Figure 17 : Prototype 2.0 – Affichage des résultats de recherche par similarité de fichier

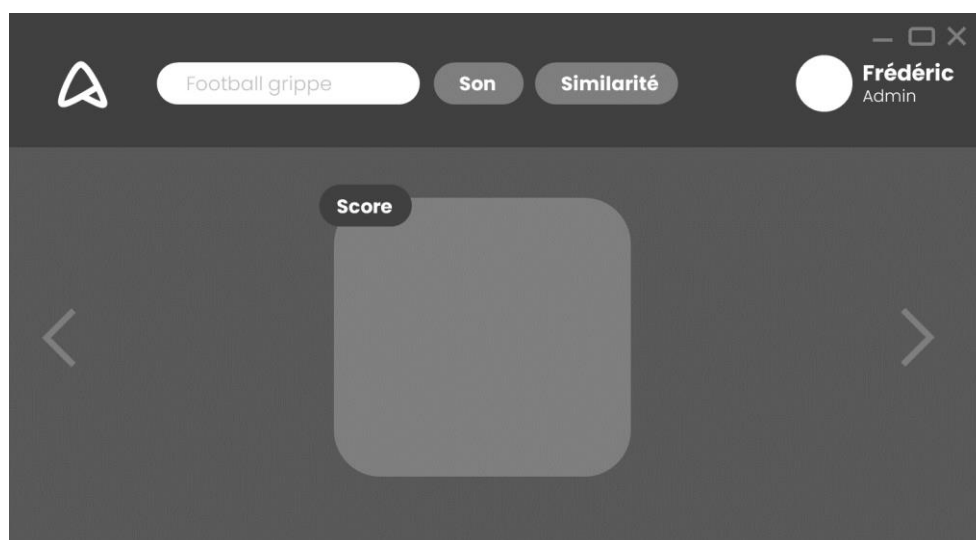


Figure 18 : Prototype 2.0 – Affichage des résultats de recherche son par similarité

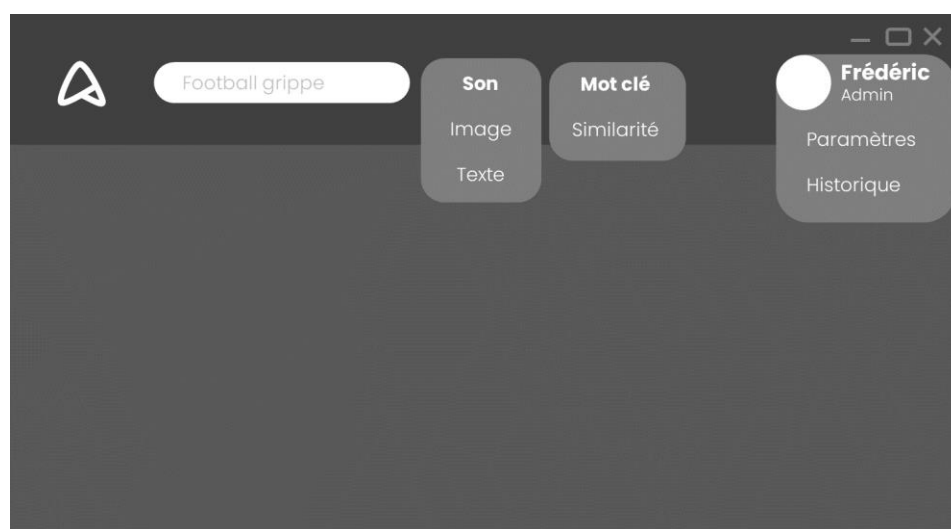


Figure 19 : Prototype 2.0 – Onglet déroulant après l'affichage des résultats de recherche