

# **RAPPORT TECHNIQUE**

INTERFACE CAMÉRA - CAMIX

**IETI**



**2023**

# Table des Matières

**01 | Introduction**

**02 | Notre équipe**

**03 | Objectifs & Schéma de principe**

**04 | Cahier des charges**

**05 | Cas d'utilisation**

**06 | Fonctions**

**11 | Intégration**

**14 | Bilan**

# Introduction

---

## Notre mission

Notre entreprise vise à simplifier la vie des professionnels de l'industrie en leur offrant des logiciels d'interface caméra avancés qui leur permettent de détecter des formes et des couleurs de manière fiable et efficace, améliorant ainsi leur productivité et leur capacité à prendre des décisions éclairées.

## Nos valeurs

- **Innovation** : en se concentrant sur le développement de nouvelles technologies pour améliorer les logiciels d'interface caméra, CamiX veut montrer son engagement envers l'innovation et l'amélioration continue.
- **Fiabilité** : en veillant à ce que leurs produits soient optimisés et fiables, CamiX fait du service client et de leur satisfaction une priorité.
- **Collaboration** : en travaillant en étroite collaboration avec les clients de l'industrie pour comprendre leurs besoins et développer des solutions qui répondent à leurs défis uniques, CamiX veut montrer son engagement envers la collaboration et la coopération.



# ANIR MEJJAOUI

DIALOGUE INTERFACE-FONCTION



# CORENTIN PAVIOT

DIALOGUE INTERFACE-FONCTION

# ROMAIN ETIENNE

FONCTIONS



# ARMAND VERTERESSIAN

DOUBLE CASQUETTE FONCTIONS/INTERFACE

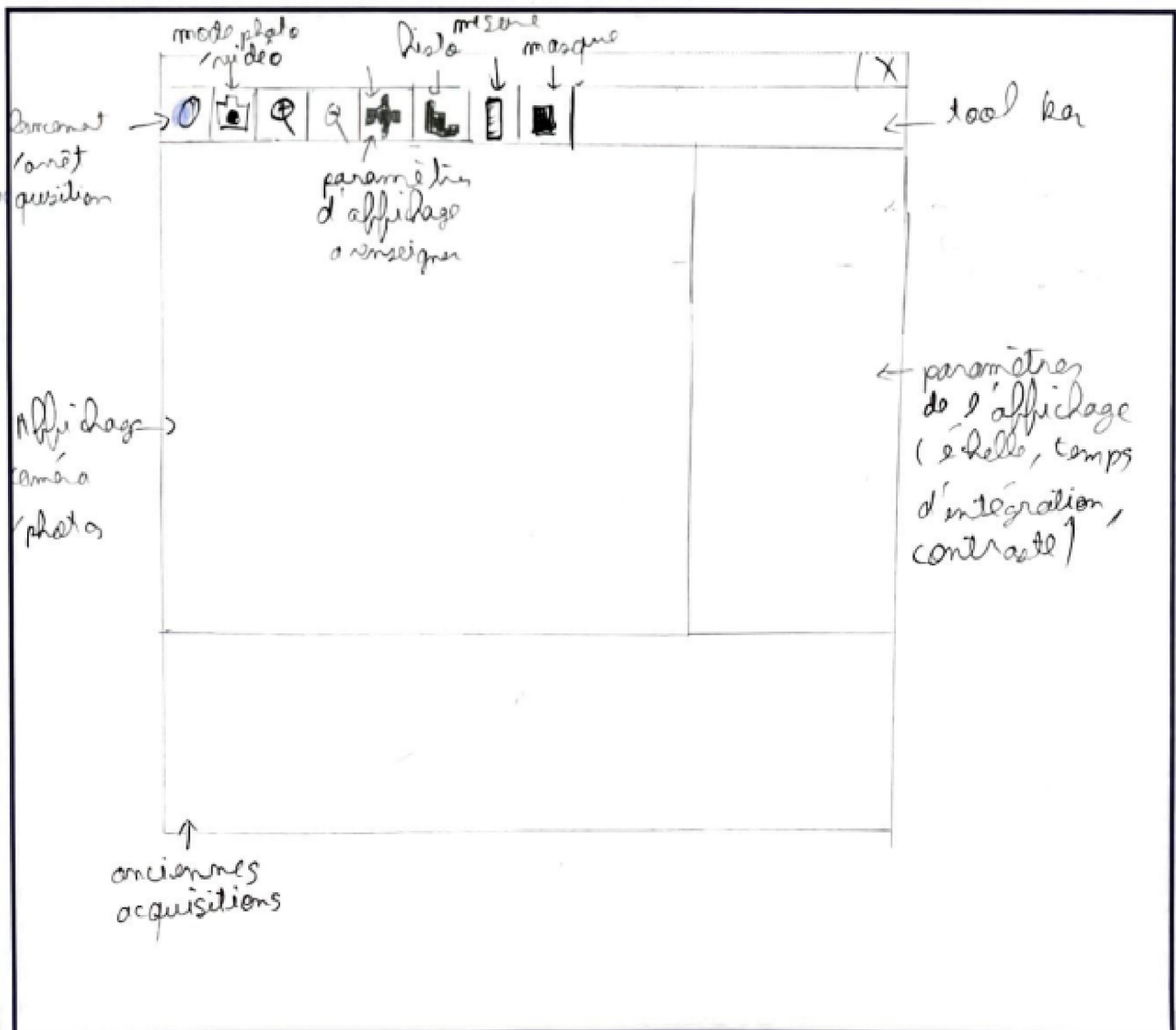
# MAUREEN SPICK

FONCTIONS ET  
GESTION DES OUTILS DE SUIVI DU PROJET



# Objectifs et schéma de principe

Le projet a pour but de créer un logiciel relié à une caméra *IDS 1280x1024* affichant l'image acquise par la caméra en "temps réel". On souhaite y ajouter notamment de nombreuses fonctionnalités de mesure et de traitement de l'image.



# Cahier des charges : contraintes et performances attendues

---

Fonctionnalités	Critère de validité
Affichage de l'image	Limitée par la résolution de la caméra
Temps d'affichage d'une mesure	Moins d'une seconde
Détection de la couleur	0.5% d'erreur de détection pour des couleurs "distinctes" (bleu, rouge, jaune)
Temps de traitement de l'image	Moins de cinq secondes
Taille de l'objet	Écart relatif <10 % , incertitude = taille du pixel
Masque/recadrage	Recadrer l'image à la "bonne" taille après un avoir choisit le masque
Affichage vidéo	En "temps réel" (<1s)
Transition mode photo/caméra	En "temps réel" (<1s)
Affichage des anciennes acquisitions	4 photos/vidéos, affichage des données au survol/ouverture d'une fenêtre
Critère de résolution	1 pixel = $180 \times 10^{-6} \times f'$
Histogramme d'une image en couleur	Afficher la superposition des 3 histogrammes pour les 3 couleurs en moins d'une seconde
Temps d'intégration	Limite de la caméra
Traitement d'image simultané	Contraste>95%,image non "floue", temps de traitement <10s

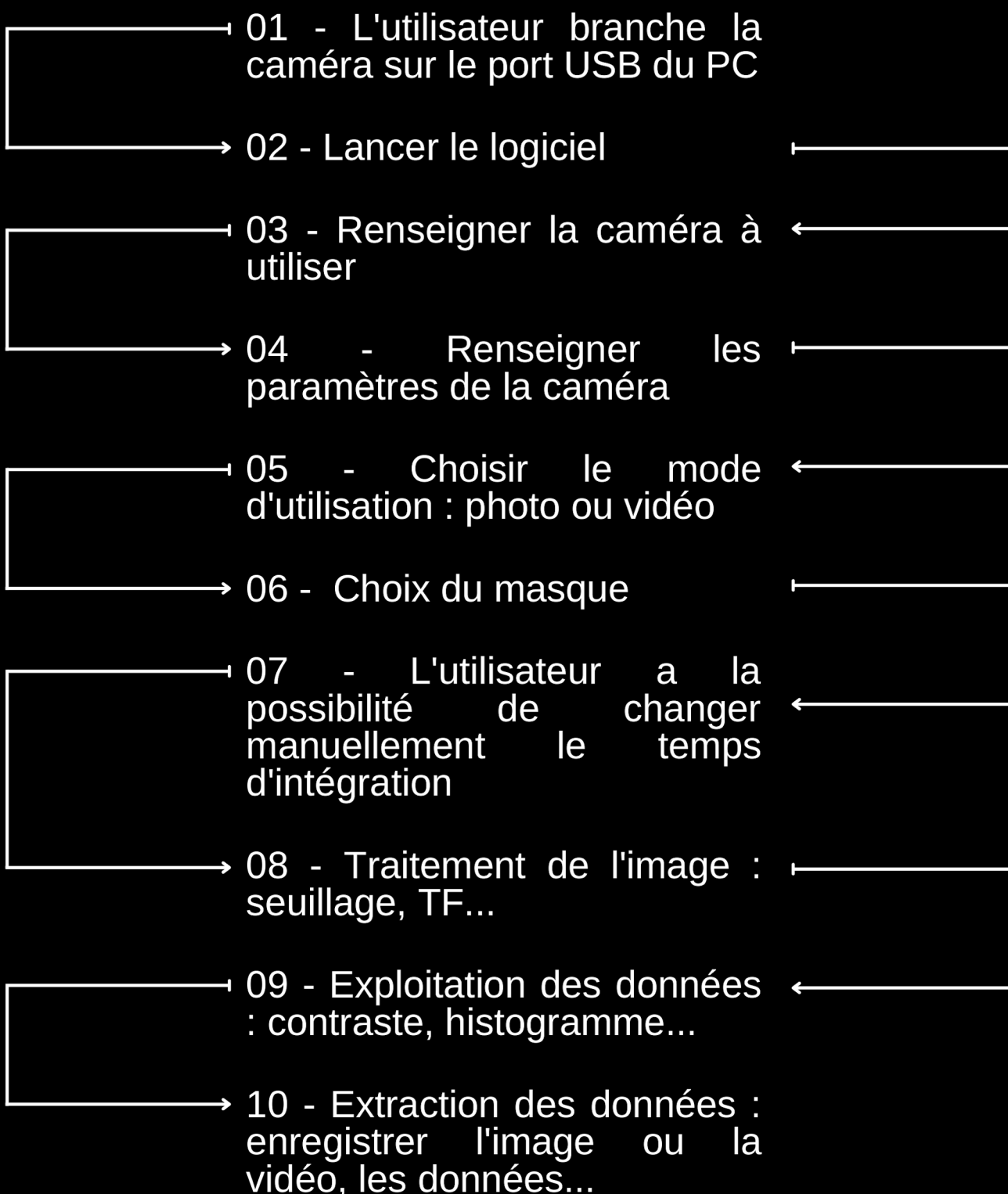
Chaque fonction Python doit être testée et approuvée avant d'être ajoutée au logiciel de gestion du projet. On s'assurera que les processus de test sont décrits dans la page associée à chaque fonction.

## Contraintes d'utilisation

L'utilisateur doit d'abord régler les paramètres optiques de la caméra avant d'utiliser l'interface.

# Cas d'utilisation

---



# Fonctions implémentées

---

Toutes les fonctions ont été développées sous Python et respectent les conventions d'écriture PEP 8 et PEP 257 pour maintenir un code cohérent et lisible.

Fonction	Objectifs	Test / Résultat
Histogramme_gris	Retourne l'histogramme d'une image en nuances de gris	Tableau des points de l'histogramme de type float32.
Histogramme_rgb	Retourne l'histogramme d'une image en couleur	Liste de trois tableaux, un pour chaque couleur, contenant les points de l'histogramme de type float32
mask_image	Masquer une image avec la forme de notre choix	Image masquée
recuperation_txt	Récupération des données sous forme de fichier .txt	Fichier .txt



# Fonctions implémentées

---

Toutes les fonctions ont été développées sous Python et respectent les conventions d'écriture PEP 8 et PEP 257 pour maintenir un code cohérent et lisible.

Fonction	Objectifs	Test / Résultat
<code>adjust_contrast</code>	Ajuster le contraste d'une image	Image dont le contraste a été modifié
<code>adjust_intensity</code>	Ajuster l'intensité d'une image	Image dont l'intensité a été modifiée
<code>taille_objet</code>	Mesurer la distance entre 2 points de l'image	Distance euclidienne en 2D entre les 2 points demandés
<code>couleur_objet_4</code>	Déterminer la couleur d'un objet	Nombre réel représentant le code de la couleur

# Fonctions implémentées

---

Toutes les fonctions ont été développées sous Python et respectent les conventions d'écriture PEP 8 et PEP 257 pour maintenir un code cohérent et lisible.

Fonction	Objectifs	Test / Résultat
detect_rectangles detect_circles detect_polygons	Détecter les formes rectangulaires, circulaires et polygonales dans une image	Image en nuances de gris dans laquelle les rectangles détectés sont dessinés
echelle	Rapport entre une distance mesurée entre deux pixels de l'image et taille d'un objet de référence	Distance euclidienne en 2D entre les 2 points demandés
seuillage_RGB	Seuiller une image en couleur	Image seuillée
zoom_image	Zoomer dans une image	Image zoomée

# Fonctions implémentées

---

Toutes les fonctions ont été développées sous Python et respectent les conventions d'écriture PEP 8 et PEP 257 pour maintenir un code cohérent et lisible.

Fonction	Objectifs	Test / Résultat
<code>apply_sharpening_filter</code>	Appliquer un filtre réhausseur à une image	Image filtrée
<code>apply_lowpass_filter</code>	Appliquer un filtre passe-bas à une image	Image filtrée
<code>filtre_moyenieur</code>	Appliquer un filtre moyenieur à une image	Image filtrée
<code>lissage</code>	Appliquer la méthode des moindres carrés aux points d'une image	Liste contenant les points solutions de la courbe de degré N

# Fonctions implémentées

---

Toutes les fonctions ont été développées sous Python et respectent les conventions d'écriture PEP 8 et PEP 257 pour maintenir un code cohérent et lisible.

Fonction	Objectifs	Test / Résultat
conversion_opencv_PIL	Conversion de l'image OpenCV vers PIL	Tableau contenant les valeurs des pixels de l'image
transforme_fourier	Appliquer la Transformée de Fourier à une liste de points	Liste des points de la FFT
transforme_fourier_inverse	Transformée de Fourier inverse	Liste des points de la FFT inverse

---

Pour obtenir plus d'informations sur les fonctions listées dans ce tableau, veuillez vous référer au fichier compressé joint avec ce rapport technique "**Fonctions Python**". Vous trouverez aussi parmi les fichiers annexes la documentation technique concernant la bibliothèque **OpenCV**.

---

# Intégration

---

## Objectif : Réaliser une interface avec Qt Designer

Pour réaliser l'interface graphique, on utilise QT Designer, un logiciel offrant une grande flexibilité pour concevoir des interfaces interactives et attrayantes de manière facile. On peut alors placer dans la fenêtre Designer les différents boutons, vues graphiques et la barre des tâches (QPushButton, QGraphicsView, QMenuBar).

La première tâche était de "convertir" le fichier Designer en code Python.

Ce code Python définit l'interface graphique, il faut ensuite pouvoir interagir avec l'interface.

## Objectif : Afficher la vidéo et une photo de la caméra dans notre interface

La première partie consistait à afficher une photo et une vidéo de la caméra industrielle après avoir appuyé sur les boutons correspondants.

Pour cela, il faut :

- Initialiser la caméra
- Récupérer les données de la caméra à chaque instant. Ces données sont stockées dans une "frame".
- Afficher la photo : il suffit pour cela d'afficher notre frame dans le QGraphicsView défini pour afficher la photo.
- Afficher la vidéo : il faut pour cela afficher notre frame dans le QGraphicsView à chaque instant défini par un timer.

# Intégration

---

## Objectif : Zoomer

Afin de zoomer et de dézoomer on définit deux booléen pour définir "l'état" du zoom, c'est-à-dire savoir si l'utilisateur veut zoomer / dézoomer ou aucun des deux lorsqu'il clique sur la photo.

Si le booléen associé au zoom est TRUE alors on zoomera à chaque clic de manière centrée sur la photo et d'un facteur 1,5. De même pour le dézoom (avec un facteur 0,75).

## Objectif : Enregistrer une image

Pour enregistrer une image il faut convertir la frame affichée par la caméra en mode photo en une image au format JPG. Pour cela on définit un booléen qui, quand il est TRUE, effectue cette action.

## Objectif : Intégrer les fonctions seuillage, filtre passe-haut, etc

Il faut tout d'abord créer des boutons associés à chaque fonction de filtre comme pour le zoom. Ainsi, si le booléen associé à une fonction de filtrage est TRUE, on renvoie l'image filtrée qui est retournée par la fonction associée codée sur Python.

### Exemple :

Pour la fonction seuillage, on crée l'action action\_seuil.

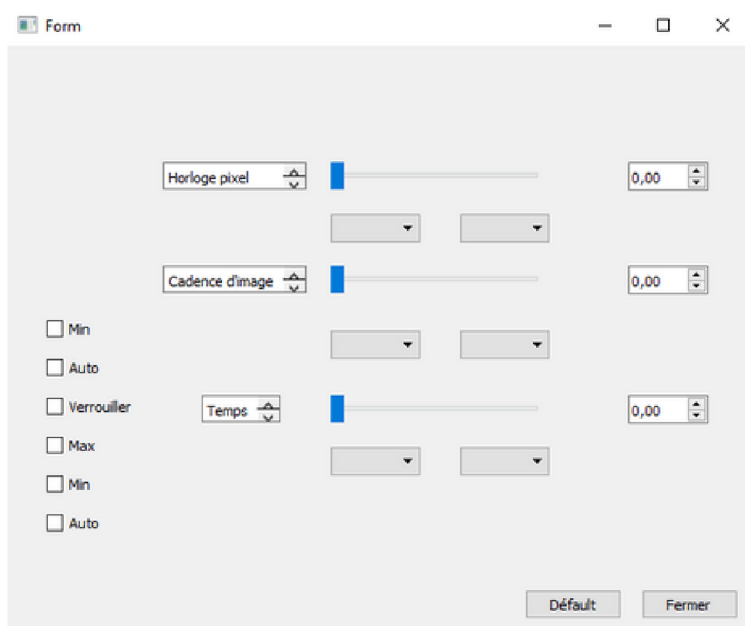
On crée un booléen : bool\_seuil. Ce booléen est modifié lorsque l'on réalise l'action action\_seuil.

On réalise une boucle if pour afficher la vidéo telle que : si le booléen est TRUE, on applique la fonction seuillage sur notre frame correspondant à l'image de la caméra.

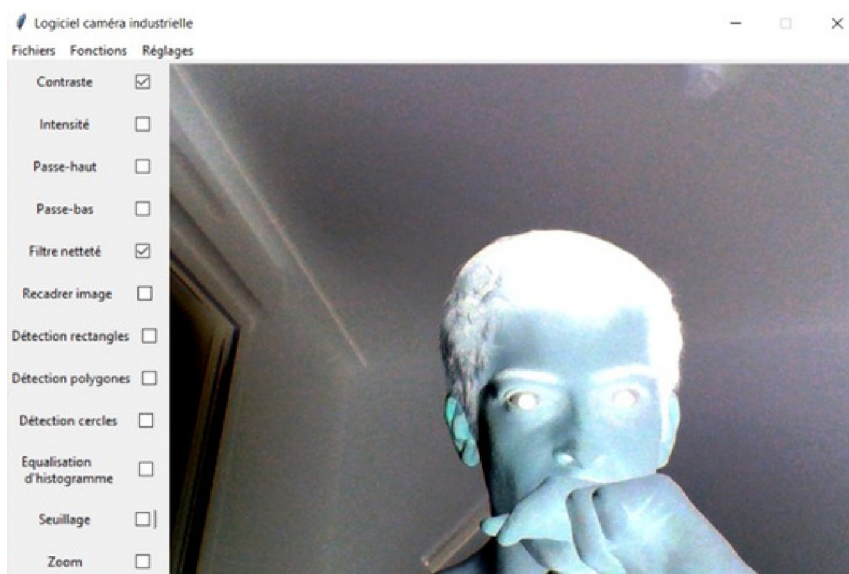
NB : seuillage n'est plus exactement une fonction puisqu'elle ne retourne rien, c'est une procédure modifiant la frame de l'image que l'on affiche.

# Intégration

---



*Onglet de réglage réalisé avec QT Designer*



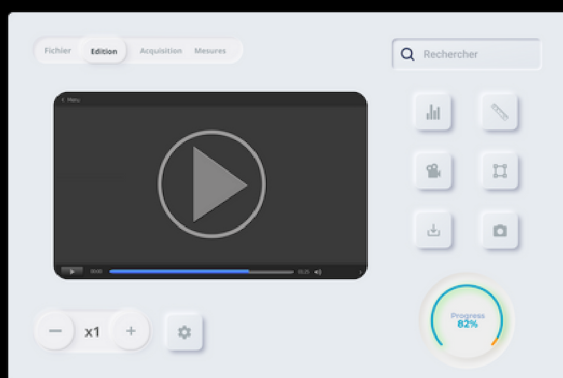
*Fenêtre principale du logiciel d'interface caméra*

# Bilan

---

## Bilan technique : avancement du projet

Le projet répond globalement au cahier des charges de départ et aux attentes de l'équipe. Toutefois, certaines fonctions n'ont pas pu être implémentées et l'interface graphique réalisée en CSS n'a pas pu être liée au fichier Python.



*Interface graphique codée en CSS*

---

## Retour d'expérience de l'équipe

Ce projet a été une excellente initiation à QT Designer et aux classes Python. Nous avons acquis des connaissances de base qui nous seront très utiles pour la suite de notre parcours. Ce projet nous a aussi permis de nous rendre compte de l'importance d'une communication efficace pour réaliser nos objectifs sans perdre de temps en utilisant notamment un logiciel de gestion de projet (**Notion**).

Dans l'ensemble, ce projet fut très enrichissant, tant sur le plan humain, en développant nos capacités à travailler en équipe, que pour l'apprentissage et le renforcement de nos compétences techniques. Cette expérience nous a motivé à réaliser et entreprendre de nouveaux projets, et de continuer à se former dans les langages informatiques.