Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

**Table des matières**

[Introduction générale 4](#_Toc136452944)

[1 Cadre général du projet 6](#_Toc136452945)

[Introduction 6](#_Toc136452946)

[1.1 Problématique 6](#_Toc136452947)

[1.2 Motivation 6](#_Toc136452948)

[1.3 Cadre du projet 6](#_Toc136452949)

[Conclusion 7](#_Toc136452950)

[2 Spécification de besoins 9](#_Toc136452951)

[Introduction 9](#_Toc136452952)

[2.1 Identification des acteurs 9](#_Toc136452953)

[2.2 Spécification des besoins 9](#_Toc136452954)

[**2.2.1** **Les besoins fonctionnels** 9](#_Toc136452955)

[**2.2.2** **Les besoins non fonctionnels** 10](#_Toc136452956)

[2.3 Conception 10](#_Toc136452957)

[**2.3.1** **UML** 10](#_Toc136452958)

[**2.3.2** **Diagramme de cas d’utilisation** 11](#_Toc136452959)

[**2.3.3** **Diagramme de classe** 12](#_Toc136452960)

[**2.3.4** **Diagramme de séquence** 12](#_Toc136452961)

[2.4 Architecture de l’application 13](#_Toc136452962)

[Conclusion 13](#_Toc136452963)

[3 Réalisation 15](#_Toc136452964)

[Introduction 15](#_Toc136452965)

[3.1 Environnement de travail 15](#_Toc136452966)

[**3.1.1** **Choix technologiques** 15](#_Toc136452967)

[**3.1.2** **Environnements logiciels** 16](#_Toc136452968)

[3.2 Réalisation 16](#_Toc136452969)

[3.3 Chronogramme 17](#_Toc136452970)

[Conclusion 17](#_Toc136452971)

[Conclusion générale 18](#_Toc136452972)

**Table des figures**

[**Figure 1:** Langage de modélisation UML 12](file:///C:\Users\Samah%20Raad\Desktop\rapport%20PFA%202023\rapport%20PFA.docx#_Toc136464005)

[**Figure 2**: Diagramme de cas d’utilisation 12](#_Toc136464006)

[**Figure 3:** Diagramme de classe 13](#_Toc136464007)

[**Figure 4:** Diagramme de séquence "Export to dataset" 14](#_Toc136464008)

[**Figure 5:** Architecture de l’application 15](file:///C:\Users\Samah%20Raad\Desktop\rapport%20PFA%202023\rapport%20PFA.docx#_Toc136464009)

[**Figure 6:** Node.js 17](file:///C:\Users\Samah%20Raad\Desktop\rapport%20PFA%202023\rapport%20PFA.docx#_Toc136464010)

[**Figure 7:** JavaScript 17](file:///C:\Users\Samah%20Raad\Desktop\rapport%20PFA%202023\rapport%20PFA.docx#_Toc136464011)

[**Figure 8:** MongoDB 18](file:///C:\Users\Samah%20Raad\Desktop\rapport%20PFA%202023\rapport%20PFA.docx#_Toc136464012)

[**Figure 9:** Visual Studio Code 18](#_Toc136464013)

[**Figure 10:** Page d'accueil MakeSense 19](#_Toc136464014)

[**Figure 11**: Page de sélection d'images MakeSense 20](#_Toc136464015)

[**Figure 12:** Créer des étiquettes et démarrer le projet 20](#_Toc136464016)

[**Figure 13:** Choisissez l'annotation manuelle et Annoter l'image 21](#_Toc136464017)

[**Figure 14:** Etiquette d'exportation 21](#_Toc136464018)

[**Figure 15:** message d'exportation 22](#_Toc136464019)

[**Figure 16:** diagramme de Gantt 22](#_Toc136464020)

# **Introduction générale**

Aujourd'hui, dans un monde de plus en plus interconnecté et axé sur les données, la gestion efficace des données est essentielle pour les entreprises, les organisations et les individus. Les plateformes web collaboratives de gestion de données sont devenues des outils indispensables pour collecter, organiser, analyser et partager des ensembles de données.

Le développement de l'intelligence artificielle (IA) a ouvert de nouvelles perspectives et possibilités dans de nombreux domaines, mais il nécessite une quantité considérable de données pour alimenter les algorithmes d'apprentissage automatique. Pour répondre à ce besoin croissant, des plateformes web collaboratives de gestion de datasets ont été développées, offrant aux chercheurs, aux ingénieurs et aux entreprises un moyen efficace de collecter, organiser et partager des données.

Parmi ces plateformes, makesense.ai se démarque par sa vision novatrice et sa technologie avancée.

Makesense.ai est un site web dédié à la gestion des images, offrant une expérience utilisateur intuitive et des fonctionnalités avancées pour faciliter le processus de gestion des données. Cette plateforme est conçue pour aider les utilisateurs à trouver, accéder et collaborer sur des ensembles de données de différentes sources, tout en garantissant leur qualité et leur intégrité.

makesense.ai et d'autres plateformes web collaboratives de gestion des images jouent un rôle essentiel dans le développement de l'intelligence artificielle en offrant aux utilisateurs un accès centralisé à des ensembles de données diversifiés, en favorisant la collaboration et en garantissant la qualité des données. Ces plateformes contribuent à accélérer le processus de recherche et d'innovation dans de nombreux domaines, ouvrant la voie à de nouvelles applications et découvertes basées sur l'IA.

*Chapitre 1*

CADRE GENERAL DU PROJET

# 

# **Cadre général du projet**

## **Introduction**

Le but de ce chapitre introductif est de mettre le projet dans son contexte général. Nous abordons premièrement le sujet autour duquel s’articule notre projet de fin d’année en décrivant la problématique

## **Problématique**

L'émergence des plateformes web de gestion d'images a ouvert de nouvelles perspectives dans le domaine de la recherche et de l'analyse de données.

Parmi ces plateformes novatrices, makesense.ai se distingue en proposant des solutions visant à simplifier la gestion et l'étiquetage efficace des images. Cependant, une problématique essentielle se pose :

Comment assurer la qualité et l'intégrité des données tout en favorisant la collaboration entre les utilisateurs ?

Cette question soulève des enjeux cruciaux en matière de sauvegarde des données et met en évidence la nécessité de développer des mécanismes fiables et des protocoles robustes pour garantir la préservation des ensembles de données collaboratifs.

## **Motivation**

La gestion et l'analyse de datasets sont des aspects essentiels dans de nombreux domaines tels que la recherche scientifique, l'intelligence artificielle, l'apprentissage automatique, et bien d'autres.

Ces datasets servent de fondements pour l'entraînement de modèles, la génération de connaissances et la prise de décisions éclairées.

Cependant, avec l'explosion des données disponibles et la nécessité croissante de collaborations entre chercheurs et praticiens, les plateformes web collaboratives de gestion de datasets ont émergé comme des solutions prometteuses.

Dans cette partie, nous allons explorer les défis spécifiques auxquels font face les plateformes web de gestion et d'étiquetage des images, en mettant l'accent sur le cas de makesense.ai.

Notre attention se portera principalement sur l'enregistrement des données dans une base persistante.

## **Cadre du projet**

MakeSense ne propose pas de moyen d'enregistrer vos projets d'annotation en cours. Cela signifie que si vous actualisez votre navigateur Web, que ce soit intentionnellement ou par erreur, toute votre progression d'annotation sera perdue et vous devrez recommencer depuis le début.

De plus, la plateforme ne dispose pas d'une option permettant de travailler en groupe, ce qui rend difficile la collecte du travail d'un groupe dans un seul ensemble de données.

L'objectif principal de ce projet est de développer une solution intégrée sur la plateforme makesense.ai, permettant d'enregistrer les données collectées dans une base de données centralisée.

Cette initiative vise à améliorer la gestion des datasets sur la plateforme, en offrant aux utilisateurs une infrastructure fiable et sécurisée pour stocker et organiser leurs données de manière efficace.

Plus spécifiquement, les objectifs du projet sont les suivants :

Intégration de la fonctionnalité de collecte de données : Nous travaillerons à l'implémentation d'un mécanisme qui permettra aux utilisateurs de makesense.ai de collecter et d'enregistrer facilement des données sur la plateforme. Cela inclura la conception d'une interface conviviale et intuitive pour faciliter le processus de collecte.

Mise en place d'une base de données centralisée : Nous mettrons en place une infrastructure de base de données robuste et sécurisée pour stocker les données collectées. Cette base de données permettra une gestion efficace des datasets, en offrant des fonctionnalités telles que la recherche, le tri et la récupération des données.

Intégration avec les fonctionnalités existantes de makesense.ai : Notre objectif est d'assurer une intégration fluide de la solution de gestion de données avec les fonctionnalités existantes de la plateforme makesense.ai. Cela permettra aux utilisateurs de profiter pleinement de l'expérience collaborative offerte par la plateforme tout en bénéficiant d'une meilleure gestion de leurs datasets.

En conclusion les objectifs principaux de notre projet, qui consistent à intégrer une solution de gestion de données sur la plateforme makesense.ai. En développant cette solution, nous visons à améliorer la collecte, le stockage et la gestion des datasets, tout en assurant la sécurité et la confidentialité des données pour les utilisateurs de la plateforme.

## **Conclusion**

Dans ce premier chapitre nous avons pu situer le projet dans son cadre général Dans

le chapitre suivant, nous allons spécifier les différents besoins auxquels doit répondre notre application et nous allons faire la conception de notre projet.

*Chapitre 2*

SPECIFICATION DE BESOINS ET CONCEPTION

# 

# **Spécification de besoins**

## **Introduction**

L’analyse des besoins est une étape importante dans la réalisation de tout projet informatique. De ce fait, ce chapitre sera consacré à la présentation des besoins. Nous commençons par l’identification des acteurs (rôles des utilisateurs) et leurs fonctionnalités principales, ensuite nous entamons la spécification fonctionnelle et non fonctionnelle.

## **Identification des acteurs**

Un acteur est l’idéalisation d’un rôle joué par une personne externe, un processus

ou une chose qui interagit avec un système. Le système est conçu pour être utilisé par un profil d’acteur.

—Utilisateur : C’est la personne qui consulte la plateforme pour résoudre des problèmes spécifiques, accéder aux fonctionnalités et bénéficier des services fournis.

## **Spécification des besoins**

L’application envisagée doit satisfaire les besoins fonctionnels qui seront exécutés

par le système et les besoins non fonctionnels qui perfectionnent la qualité logicielle du système. Dans ce qui suit, nous identifions les fonctionnalités requises pour l’acteur.

### **Les besoins fonctionnels**

Chaque solution de conception doit satisfaire des besoins fonctionnels qui

représentent les attentes l’acteur de la future application.

— Pour chaque utilisateur :

* Insérer des images à annoter : L'utilisateur doit pouvoir importer des images sur la plateforme pour les annoter.
* Sélectionner les zones à annoter sur l’image : L'utilisateur doit pouvoir choisir les parties spécifiques de l'image à annoter. Cela peut être fait en utilisant des points, des lignes, des rectangles ou des polygones pour encadrer les régions d'intérêt.
* Donner à chaque zone un label : L'utilisateur doit pouvoir attribuer des étiquettes ou des catégories aux zones annotées sur l'image. Par exemple, dans le cas de la détection d'objets, il peut attribuer des étiquettes telles que "voiture", "arbre" ou "personne" aux zones correspondantes.
* Importer le contenu de datasets : L'utilisateur doit avoir la possibilité d'importer des ensembles de données préexistants dans la plateforme. Cela permet d'utiliser des données existantes pour l'annotation et l'entraînement des modèles.

### **Les besoins non fonctionnels**

Outre les besoins fonctionnels, il est nécessaire de prendre en considération certains

besoins nécessaires et non fonctionnels pour une utilisation simple des différentes fonctionnalités offertes :

**Sécurité :** La plateforme makesense.ai doit garantir la sécurité des données, des modèles et des utilisateurs. Cela inclut des mesures de protection telles que le chiffrement des données, la gestion des droits d'accès, l'audit des activités et la détection des menaces potentielles.

**Extensibilité :** La plateforme makesense.ai doit être conçue de manière à pouvoir être facilement étendue pour prendre en charge de nouvelles fonctionnalités et technologies à l'avenir. Il est important de concevoir une architecture modulaire et évolutive qui facilite l'ajout de nouvelles fonctionnalités sans compromettre la stabilité du système existant.

**Convivialité :** La plateforme makesense.ai doit offrir une interface utilisateur conviviale et intuitive pour faciliter son utilisation par des utilisateurs de différents niveaux de compétence. Il est essentiel de concevoir une expérience utilisateur agréable qui permet aux utilisateurs de tirer pleinement parti des fonctionnalités de la plateforme.

## **Conception**

Les fonctionnalités mentionnées ci-dessus seront présentées dans la partie qui suit

par un diagramme de cas d’utilisation et un diagramme de classes pour la partie ajouter à la plateforme. Reconnu comme étant le standard industriel par excellence de la modélisation objet, nous avons opté au langage de modélisation UML. Cette illustration permettra de mieux structurer les interactions entre l’acteur et le système.

### **UML**

UML, abréviation de Unified Modeling Language, est un langage de modélisation

normalisé composé d’un ensemble intégré de diagrammes, développé pour aider les développeurs de systèmes et de logiciels à spécifier, visualiser, construire et documenter les artefacts des systèmes logiciels,

ainsi qu’à modéliser et d’autres systèmes non logiciels. L’UML représente un ensemble de meilleures pratiques d’ingénierie qui ont fait leurs preuves dans la modélisation de systèmes grands et complexes. L’UML est une partie très importante du développement de logiciels orientés objet et du processus de développement logiciel Nous avons choisi UML pour les points ci-dessous :

— Fournir des mécanismes d’extensibilité et de spécialisation pour étendre les concepts de base.

— UML n’impose pas de méthode de travail particulière, il peut être intégré à n’importe quel processus de développement logiciel de manière transparente

— Fournir une base formelle pour comprendre le langage de modélisation.

— UML permet d’améliorer les méthodes de travail, tout en préservant les modes de fonctionnement.

Une image contenant Graphique, graphisme, Police, conception

Description générée automatiquement**Diagramme de cas d’utilisation**

**Figure 1:** Langage de modélisation UML

La figure 2 présente le diagramme de cas d’utilisation de notre partie qui est

intégrée dans la plateforme de notre application. Il donne une vue globale des fonctionnalités ajouté, celle qui consiste à visualiser et analyser le comportements des utilisateurs.

Une image contenant capture d’écran, diagramme, ligne, texte

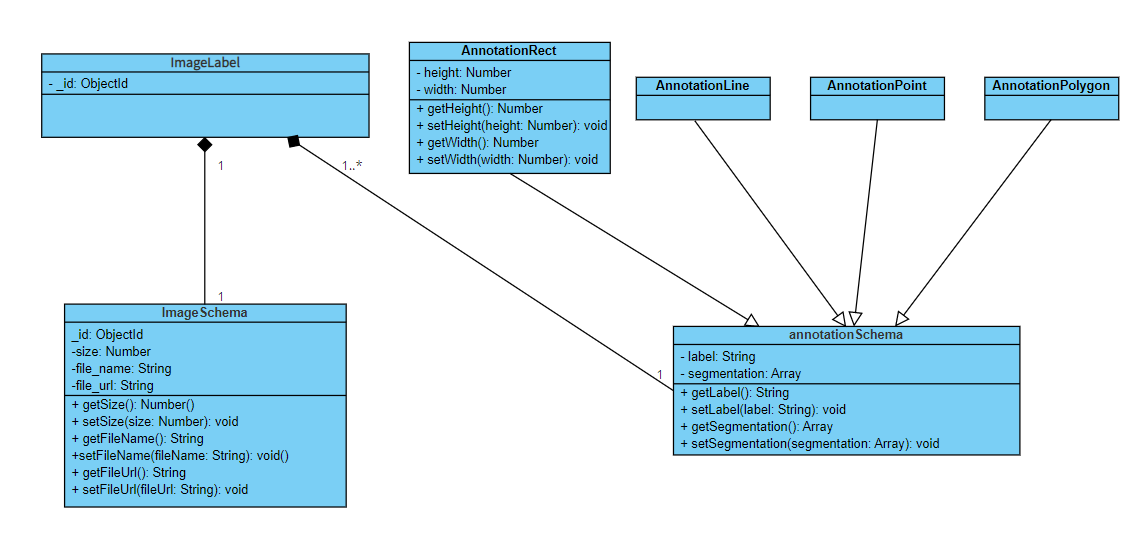
Description générée automatiquement

**Figure 2**: Diagramme de cas d’utilisation

### **Diagramme de classe**

Le diagramme de classes explique la structure statique du système en termes de

classes et de relations entre elles. La figure 3 présente le diagramme de classes de notre partie qui est intégrée dans la plateforme.

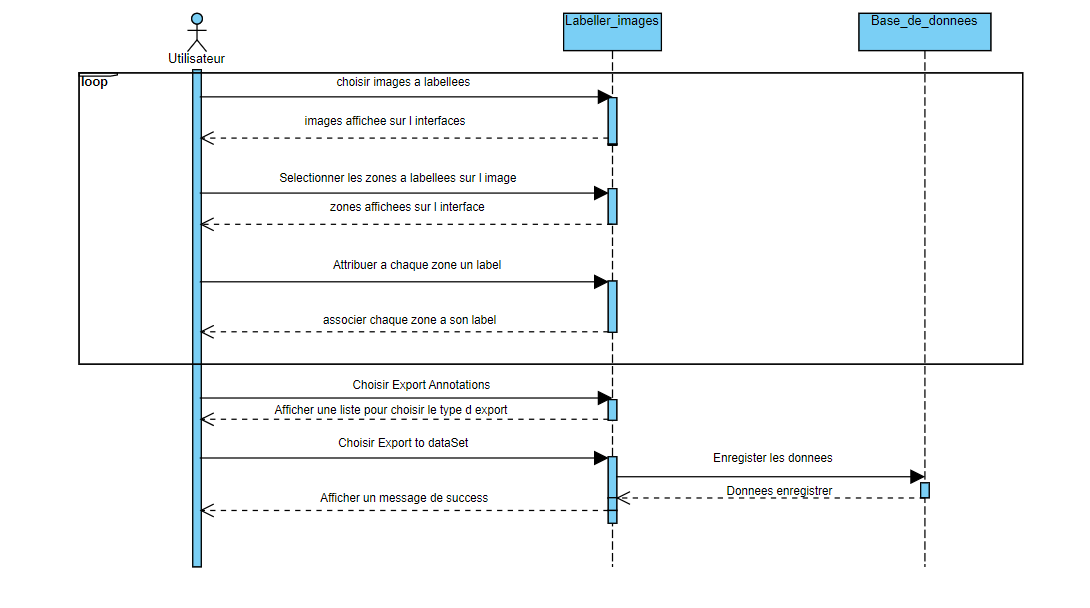


**Figure 3:** Diagramme de classe

### **Diagramme de séquence**

L'utilisateur commence par choisir une ou plusieurs images, puis ces images seront

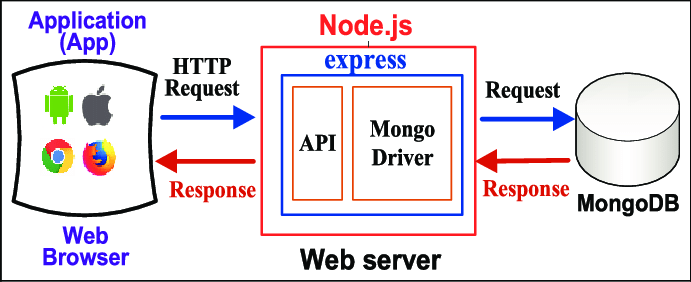
affichées sur l'interface de la plateforme. L'utilisateur sélectionne les zones à étiqueter sur chaque image et attribue un label à chaque zone. Une fois que l'utilisateur a terminé, il choisit d'exporter les annotations vers la base de données. Le système collecte les données et les envoie à la base de données pour les enregistrer. Si l'enregistrement est réussi, un message sera affiché à l'utilisateur.



**Figure 4:** Diagramme de séquence "Export to dataset"

## **Architecture de l’application**

La figure 5 présente l’architecture physique de notre application. Notre

application doit être déployée sur un serveur web. Ce dernier aura besoin de communiquer avec un serveur de base de données pour l’exécution des requêtes assurant l’enregistrement des données.

**Figure 5:** Architecture de l’application

## **Conclusion**

Dans ce chapitre nous avons énuméré les besoins fonctionnels et non fonctionnels de

l’application pour avoir une vision globale du sujet. Puis, on a fait la conception de notre projet, ce qui nous conduit à définir dans le prochain chapitre les choix technologiques et les étapes de réalisation.

*Chapitre 3*

REALISATION

# 

# **Réalisation**

## **Introduction**

Après l’étape de conception de l’application, nous allons dans ce chapitre, décrire

la phase de réalisation, nous allons présenter, en premier lieu, l’environnement du travail utilisé pour le développement de l’application, ensuite nous allons donner un aperçu sur le travail accompli à travers des captures d’écran.

## **Environnement de travail**

La réalisation de l’application demandera plusieurs outils ainsi qu’un

environnement de développement qui faciliteront cette tâche. Dans ce qui suit nous présenterons dans une première partie les outils (environnement technique) utilisés, ensuite nous présenterons l’environnement logiciel

### **Choix technologiques**

#### **Node.js**

Node.js est un environnement d’exécution JavaScript open source et multi-

plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d’exécution JavaScript V8 et utilise une architecture

d’E / S



**Figure 6:** Node.js

#### **JavaScript**

JavaScript est un langage de programmation léger et interprété. Il est conçu pour

créer des applications centrées sur le réseau. Il est complémentaire et intégré à Java. JavaScript est très facile à mettre en œuvre car il est intégré à HTML. Il est ouvert et multiplateforme.



**Figure 7:** JavaScript

### **Environnements logiciels**

#### **MongoDB**

MongoDB est un programme de base de données multiplateforme orienté document

disponible en source. Classé comme programme de base de données NoSQL, MongoDB utilise des documents de type JSON avec des schémas optionnels.

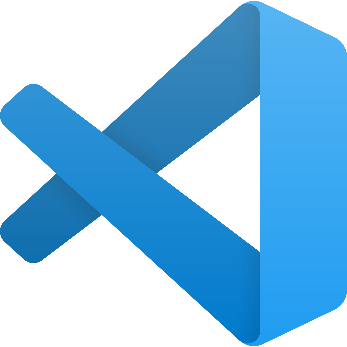


**Figure 8:** MongoDB

#### **Visual Studio Code**

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété

de langages de programmation, notamment Java, JavaScript, Node.js et C++.



**Figure 9:** Visual Studio Code

## **Réalisation**

Dans cette section, nous allons présenter les principales fonctionnalités de notre

application à travers quelques captures d’écran.

L'interface utilisateur est facile à utiliser. Il vous suffit de télécharger les images que vous souhaitez annoter, d'annoter les images et d'exporter les étiquettes.

MakeSense prend en charge plusieurs annotations : zone de délimitation, polygone et annotation de point. Vous pouvez exporter les étiquettes dans différents formats, notamment YOLO, VOC XML, VGG JSON et CSV. Selon le site Web, MakeSense est maintenant stocke les images dans une base de données.

Voici un guide étape par étape pour utiliser l'outil d'annotation MakeSense.

1. Allez sur [www.makesense.ai](http://www.makesense.ai)

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

**Figure 10:** Page d'accueil MakeSense

1. Cliquez sur la case en bas à droite pour accéder à la page d'annotation et vous verrez la page suivante où vous pourrez télécharger les images que vous souhaitez annoter.
2. Après avoir sélectionné et téléchargé les images, cliquez sur le bouton " image recognition".

Une image contenant texte, capture d’écran, Police, graphisme

Description générée automatiquement

**Figure 11**: Page de sélection d'images MakeSense

1. Comme aucune étiquette n'est chargée, il vous sera demandé de créer une liste de noms d'étiquettes pour votre projet.

Pour ajouter une nouvelle étiquette, cliquez sur le signe + dans le coin supérieur gauche de la boîte de message et saisissez l'étiquette dans le champ de texte «*Insérer une étiquette ».*Répétez l'opération pour toutes les étiquettes.

Lorsque vous avez ajouté toutes les étiquettes, sélectionnez « *Démarrer le projet* ».

Une image contenant texte, capture d’écran, Logiciel multimédia, logiciel

Description générée automatiquement

**Figure 12:** Créer des étiquettes et démarrer le projet

1. Ici, vous êtes en mesure de sélectionner la forme d'annotation qui vous convient et d'effectuer l'étiquetage manuellement selon votre préférence.

Une image contenant Logiciel multimédia, logiciel, Logiciel de graphisme, nourriture

Description générée automatiquement

**Figure 13:** Choisissez l'annotation manuelle et Annoter l'image

1. Lorsque vous cliquer sur le bouton « export annotation » une liste est afficher pour choisir le type d’export.

Une image contenant capture d’écran, Logiciel multimédia, logiciel, Logiciel de graphisme

Description générée automatiquement

**Figure 14:** Etiquette d'exportation

1. Après que vous choisir l’exportation pour une base de données, un message de succès est affiché.

Une image contenant texte, Logiciel multimédia, logiciel, capture d’écran

Description générée automatiquement

**Figure 15:** message d'exportation

## **Chronogramme**

Nous avons planifié le projet comme suit, la figure ci-dessous représente le

diagramme de Gantt.

Une image contenant texte, capture d’écran, Caractère coloré, conception

Description générée automatiquement

**Figure 16:** diagramme de Gantt

## **Conclusion**

Dans ce chapitre, nous avons présenté l’environnement matériel et logiciel utilisé

lors du développement du site web. Ensuite, on a présenté quelques captures d’écran montrant le bon fonctionnement de notre application.

# **Conclusion générale**

En conclusion, dans un monde de plus en plus axé sur les données et l'intelligence artificielle, les plateformes web collaboratives de gestion de données sont devenues indispensables pour les entreprises, les organisations et les individus. Makesense.ai se démarque par sa vision novatrice et sa technologie avancée, offrant aux utilisateurs une expérience intuitive et des fonctionnalités avancées pour faciliter la gestion des images.

L'intégration d'une solution de base de données centralisée sur makesense.ai représente une avancée significative dans le domaine de la gestion des données. Cette fonctionnalité permet d'enregistrer les données collectées de manière organisée, ce qui facilite leur accès et leur utilisation ultérieure. De plus, en garantissant la qualité et l'intégrité des données, makesense.ai contribue à la fiabilité des résultats obtenus grâce à l'IA.

Cependant, il convient de noter que makesense.ai présente également quelques limitations. Par exemple, il ne propose pas d'option pour télécharger des fichiers zip, ce qui peut rendre le processus de téléchargement des données plus long et fastidieux. De plus, la plateforme ne permet pas encore la collaboration en temps réel sur un même projet d'annotation, ce qui peut limiter la productivité des équipes travaillant sur des tâches d'annotation complexes.

Malgré ces limitations, makesense.ai et d'autres plateformes web collaboratives de gestion des images jouent un rôle essentiel dans le développement de l'intelligence artificielle. Elles offrent un accès centralisé à des ensembles de données diversifiés, favorisent la collaboration et garantissent la qualité des données, ce qui accélère le processus de recherche et d'innovation dans de nombreux domaines.

Dans l'ensemble, makesense.ai représente une solution prometteuse pour la gestion efficace des images et des données, et ses fonctionnalités avancées ouvrent la voie à de nouvelles applications et découvertes basées sur l'IA. En continuant d'améliorer et d'élargir ses fonctionnalités, makesense.ai peut jouer un rôle clé dans l'avancement de la recherche et de l'innovation dans l'ère de l'intelligence artificielle et des données massives.