



République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Supérieure Privée des Technologies de l'Information et de Management de Nabeul



Rapport de Projet de Fin d'Etudes soumis afin d'obtenir le
Diplôme National d'Ingénieur en Génie Logiciel
Réalisé par

OUMAIMA Dghaies

*Système automatique de recommandation des réservations dans le
domaine logistique*

Réalisé au sein de



Encadré par

Encadrant(s) universitaire(s)

NESRINE Meddeb

Encadrant(s) professionnel(s)

RAOUF Terres

Année universitaire
2024 – 2025

Dédicaces

Avec tout le respect et tout l'amour,

Je dédie cet humble travail à celui qui m'a appris que là où il y a une volonté, il y a un chemin... .

À l'âme de mon père

Ce travail est dédié à mon père, HABIB, aujourd'hui disparu, qui m'a toujours soutenue et motivée dans mes études. J'espère que, du monde qui est désormais le sien, il apprécie ce modeste geste comme une preuve de reconnaissance de la part d'une fille qui n'a jamais cessé de prier pour le salut de son âme.

Puisse Dieu, Le Tout-Puissant, l'accueillir en Sa sainte miséricorde.

À ma mère, MARIEM

Aucune dédicace ne pourrait suffire à exprimer tout l'amour et l'attachement que j'ai pour toi.

Aucun mot ne saurait traduire toute ma gratitude, mon amour et mon respect.

Puisse Le Bon Dieu nous accorder santé et longue vie ensemble.

À mon frère FRAJ, ma sœur AMANI, ma belle-sœur MAHA et ses enfants NOUR & AMINE

Merci pour votre affection et vos encouragements. Je ne pourrais jamais imaginer la vie sans vous.

Que Dieu vous protège et vous accorde tout le succès et le bonheur que vous méritez.

À l'âme de ma nièce, KAMILIA

*Puisse Dieu t'accueillir en Sa sainte miséricorde,
et que ce travail soit une prière dédiée à ton âme.*

À mes amis et à ma grande famille

Qui ont su me redonner force et confiance, simplement parce que j'étais moi... Merci de m'avoir protégée, guidée, conseillée et comblée par votre amour et votre affection inestimables.

Remerciements

Avant de présenter mon travail, je tiens à exprimer ma profonde reconnaissance envers toutes les personnes qui m'ont soutenue, de près ou de loin. Qu'elles trouvent ici, collectivement et individuellement, l'expression de ma gratitude sincère.

Je tiens à exprimer mes sincères remerciements à mon encadrant professionnel, Monsieur Raouf Terres, pour m'avoir accueillie au sein de l'équipe inoubliable de CargoWeen et pour la confiance qu'il m'a témoignée. Je le remercie également pour ses précieux conseils et ses orientations qui m'ont permis de mener à bien ce projet.

Ma profonde gratitude s'adresse aussi à mon encadrante académique, Madame Nesrine Meddeb, pour son encadrement bienveillant et constant, qui m'a grandement aidée à accomplir ce travail dans les meilleures conditions.

J'exprime également toute ma reconnaissance à l'équipe de CargoWeen pour leur accueil chaleureux et leur aide précieuse tout au long de cette expérience.

Un grand merci s'adresse à mes enseignants de l'ITBS pour la richesse de leur apport scientifique, qui m'a été d'une grande utilité lors de l'élaboration de ce projet, ainsi qu'à toutes les personnes qui m'ont encouragée et soutenue, de près ou de loin, dans cette aventure.

Enfin, je remercie chaleureusement les membres du jury pour l'honneur qu'ils me font en évaluant ce travail. Puissent-ils y trouver tout l'intérêt et le sérieux que j'ai mis à sa réalisation.

Table des matières

Introduction générale	1
1 Cadre générale du projet	2
1.1 Introduction	2
1.2 Présentation de l'organisme d'accueil	2
1.2.1 Les services du Cargoween	2
1.2.2 Les partenaires de Cargoween	3
1.2.3 Organigramme de l'entreprise	3
1.3 Présentation du projet	3
1.3.1 Cadre du projet	3
1.3.2 Contexte du projet	4
1.3.3 Problématique	4
1.3.4 Analyse et critique de l'existant	4
1.3.5 Solution proposée	5
1.4 Méthodologie de travail	6
1.4.1 Choix d'approches de gestion du projet	6
1.4.2 Méthodologie adoptée	8
1.5 Méthodologie Scrum	9
1.5.1 Les rôles	9
1.5.2 Les artefacts	9
1.5.3 Les cérémonies	10
1.5.4 Les méthodes de classification	10
1.5.4.1 La méthode WSJF	10
1.5.4.2 La méthode MoSCoW	11
1.5.5 Méthodologie Crisp ML (Q)	12
1.6 Planification de projet	14
1.7 Planification des sprints	15
1.8 Conclusion	15
2 État de l'art et spécifications des besoins	17
2.1 Introduction	17
2.2 intelligence artificielle (IA)	17
2.3 Machine Learning (ML)	18
2.3.1 Apprentissage supervisé	19
2.3.2 Apprentissage non supervisé	20
2.3.3 Apprentissage par renforcement	20
2.3.4 MLOps	21
2.3.5 MLflow	21

2.4	Deep Learning (DL)	22
2.5	Traitements du langage naturel (NLP)	23
2.5.1	Technologies liées au NLP	23
2.5.2	Fonctionnement du NLP	24
2.5.2.1	Saisie de texte et collecte de données	24
2.5.2.2	Prétraitement du texte	25
2.5.2.3	Représentation du texte	25
2.5.2.4	Extraction des caractéristiques	25
2.5.2.5	Sélection et formation du modèle	26
2.5.2.6	Déploiement et inférence du modèle	26
2.5.2.7	Évaluation et optimisation	26
2.5.2.8	Itération et amélioration	26
2.6	Hugging Face	27
2.7	Reconnaissance optique de caractères (OCR)	27
2.7.1	Les types OCR	28
2.7.2	Fonctionnement d'OCR	28
2.7.3	Les modèles OCR	30
2.8	Analyse des besoins	31
2.8.1	Besoins fonctionnels	31
2.8.2	Besoins non fonctionnels	31
2.8.3	Diagramme de cas d'utilisation	32
2.8.4	Diagramme de classe	32
2.8.5	Backlog de produit	33
2.9	Conclusion	34
3	Release 1 : Système d'extraction et de recommandation	35
3.1	Introduction	35
3.2	Architecture du projet	35
3.3	Sprint1 : Système d'extraction de données	36
3.3.1	Fonctionnement d'OCR	37
3.3.2	Fonctionnement de NLP	39
3.4	Sprint 2 : Système de recommandation	41
3.4.1	Géocodage de l'aéroport d'arrivée	42
3.4.2	Géocodage des Transitaire	42
3.4.3	Calcul de la distance	42
3.4.4	Prédiction ML	43
3.4.5	Déploiement	49
3.4.5.1	Conteneurisation avec Docker	49
3.4.5.2	Orchestration avec Kubernetes	49
3.4.5.3	Déploiement sur Google Cloud Platform (GCP)	49
3.4.5.4	Automatisation CI/CD	50
3.4.6	Calcul des tarifs	52
3.4.7	Recommandation des meilleures sociétés	52
3.5	Conclusion	53

4 Release 2 : Développement d'application web	54
4.1 Introduction	54
4.2 Architecture du projet	54
4.3 Planification de release 2	55
4.4 Sprint 1 : Système de recommandation	56
4.4.1 Raffinement de cas d'utilisation "Authentification"	56
4.4.2 Raffinement de cas d'utilisation «Gérer profil»	61
4.4.3 Raffinement de cas d'utilisation "Consulter toutes les sociétés recommandées"	66
4.5 Sprint 2 : Système de réservation	71
4.5.1 Raffinement de cas d'utilisation "Gestion des réservations"	72
4.5.2 Raffinement de cas d'utilisation "Paiement en ligne"	77
4.6 Intégration du module CargoWeen dans l'application principale	79
4.7 Choix technique	79
4.7.1 Environnement logiciel	79
4.7.2 Moyen de programmation	81
4.8 Conclusion	81
Conclusion générale	81
Appendices	83
A Annexe 1 : Fiche société Cargoween	84
Bibliographie	87

Table des figures

1.1	Logo CargoWeen	2
1.2	Les partenaires	3
1.3	Organigramme de l'entreprise	3
1.4	Site PagesJaunes	5
1.5	Processus de la méthodologie Scrum	9
1.6	WSJF	11
1.7	MoSCoW	11
1.8	Processus du cycle de vie du développement de l'apprentissage automatique	12
1.9	Assurance qualité à chaque étape	13
1.10	Planification de projet	15
1.11	Planification des sprints	15
2.1	intelligence artificielle	18
2.2	Machine Learning	19
2.3	Composant mlflow	21
2.4	processus NLP	24
2.5	Hugging Face	27
2.6	OCR	27
2.7	processus d'OCR	28
2.8	Tesseract Logo	30
2.9	EasyOCR Logo	30
2.10	Doctr Logo	30
2.11	Surya OCR Logo	31
2.12	Diagramme de cas d'utilisation global	32
2.13	Diagramme de classe	33
3.1	architecture du projet	36
3.2	Architecture de système d'extraction	37
3.3	Exemple de reçu de réservation	37
3.4	Réultat Tesseract	38
3.5	Réultat EasyOcr	38
3.6	Text extrait avec Tesseract	38
3.7	Réultat de FlauBert	39
3.8	Réultat CamemBert	39
3.9	Exemple des informations extraites	41
3.10	Architecture de sprint 2	42
3.11	Géocodage de l'aéroport	42
3.12	pipeline de prédition en ML	44
3.13	Les colonnes et types	45

3.14	Exemple de préfixe du code postal	45
3.15	Exemple de médiane de la colonne	46
3.16	Exemple des données transformées	46
3.17	Rapport de classification de 20% test	46
3.18	Rapport de classification de 30% test	47
3.19	Modèle en Production	48
3.20	MLFlow déployer	50
3.21	Vérification de service	50
3.22	MLFlow déployer	50
3.23	Workflow github actions	51
3.24	Email de notification	51
3.25	MLFlow déployer	52
3.26	Recommandation des entreprises	52
4.1	Architecture technique	54
4.2	Plannification Release 2	56
4.3	Diagramme de cas d'utilisation «Authentification»	57
4.4	Diagramme de sequence d'authentification	58
4.5	Prototype d'interface d'inscription	58
4.6	Prototype d'interface d'authentification	59
4.7	Interface d'inscription	60
4.8	Interface d'authentification	60
4.9	Diagramme de cas d'utilisation «Gestion profil»	61
4.10	Diagramme de sequence de Changer mot de passe	63
4.11	Prototype d'interface de consulter profil	64
4.12	Prototype d'interface pour changer le mot de passe	64
4.13	Interface de profil	65
4.14	Interface pour changer le mot de passe	65
4.15	Email de confirmation	66
4.16	Diagramme de cas d'utilisation «Consulter les sociétés recommandées»	66
4.17	Diagramme de sequence de Consulter les sociétés recommandées	68
4.18	Prototype d'interface de recherche	69
4.19	Prototype d'interface détail de réservation	69
4.20	Partie 1 d'interface de Recherche	70
4.21	Partie 2 d'interface de Recherche	70
4.22	Partie 1 d'interface détail de réservation	71
4.23	Partie 2 d'interface détail de réservation	71
4.24	Diagramme de cas d'utilisation «Gestion des réservations»	72
4.25	Diagramme de sequence de Modification statut de réservation	74
4.26	Prototype d'interface de réservation reçue	75
4.27	Prototype d'interface détail de réservation reçue	75
4.28	Interface de réservation reçue	76
4.29	Interface détail de réservation reçue	76
4.30	Email de confirmation	77
4.31	Diagramme de cas d'utilisation «Paiement en ligne»	77
4.32	Diagramme de sequence de Paiement en ligne	78
4.33	GitHub	79
4.34	MongoDB	79

4.35 Visual Studio Code	80
4.36 Figma	80
4.37 Trello	80
4.38 Postman	80
4.39 Lucidchart	81
4.40 LaTex	81
4.41 Python	81
4.42 NextJs	81

Liste des tableaux

1.1	Comparaison entre l'approche classique et agile	7
1.2	Comparaison entre Scrum, Kanban, XP et CrispML(Q)	8
1.3	Classification des éléments par niveau de priorité	12
2.1	Les types OCR	28
2.2	Backlog de produit	33
3.1	Tableau comparatif des modèles	47
4.1	Backlog de Sprint 1	56
4.2	Description textuelle du cas d'utilisation « Authentification »	57
4.3	Description textuelle du cas d'utilisation « Changer mot de passe »	62
4.4	Description textuelle du cas d'utilisation « Consulter les sociétés recommandées »	67
4.5	Backlog de Sprint 2	72
4.6	Description textuelle du cas d'utilisation « Modifier statut de réservation »	73
4.7	Description textuelle du cas d'utilisation « Payer en ligne »	78

Liste des abréviations

- **PO** : Product Owner
- **WSJF** : Weighted Shortest Job First
- **ML**) : Machine Learning
- **CRISP-ML(Q)** : Cross Industry Standard Process for Machine Learning with Quality assurance
- **IA** : Intelligence Artificielle
- **MLOps** : Machine Learning Operations"
- **CI/CD** : Intégration Continue/Livraison Continue
- **DL** : Deep Learnin
- **CNN** : Réseaux Neuronaux Convolutifs
- **RNN** : Réseaux Neuronaux Récursents
- **NLP** : Traitement du Langage Naturel
- **TALN** : Traitement Automatique du Langage Nature
- **NLTK** : Natural Language Toolkit
- **NER** : Reconnaissance d'Entités Nommée
- **OCR** : Reconnaissance Optique de Caractères
- **RIC** : Reconnaissance Intelligente des Caractères
- **OWR** : Reconnaissance Optique de Mots
- **OMR** : Reconnaissance Optique des Marque

Introduction générale

Dans un contexte économique mondialisé et en constante évolution, la logistique joue un rôle stratégique dans la performance des entreprises. Le secteur du fret aérien, en particulier, représente un maillon essentiel pour le transport rapide et sécurisé des marchandises à l'échelle internationale. Cependant, le processus de réservation et de sélection d'un agent transitaire reste souvent complexe, peu digitalisé et chronophage pour les utilisateurs finaux.

C'est dans ce cadre que s'inscrit ce Projet de Fin d'Études, qui vise à concevoir et développer une plateforme intelligente dédiée aux transitaires. Cette solution permet aux utilisateurs de rechercher, comparer et réserver des services de transport auprès de diverses compagnies et sociétés de transit. Le cœur du projet repose sur l'intégration de l'intelligence artificielle, notamment à travers un système de recommandation et de prédition, afin de proposer les transitaires les plus pertinents selon des critères géographiques, économiques et opérationnels.

Ce rapport de Projet de Fin d'Études s'articule autour de quatre chapitres principaux.

- Le premier chapitre présente l'organisme d'accueil, le contexte général du projet, ainsi que la méthodologie adoptée pour sa réalisation, notamment l'étude de l'existant.
- Le deuxième chapitre est consacré à l'état de l'art des technologies liées à l'intelligence artificielle, tout en détaillant la phase de conception de la solution.
- Le troisième chapitre décrit la mise en œuvre des systèmes d'extraction d'informations à partir de documents et le système de recommandation des transitaires.
- Le quatrième et dernier chapitre traite du développement de l'application web, en mettant en avant les choix techniques, les outils utilisés et les principales fonctionnalités implémentées.

Chapitre 1

Cadre générale du projet

1.1 Introduction

Dans le cadre de ce projet de fin d'études, il est essentiel de comprendre l'environnement dans lequel il a été réalisé ainsi que les besoins ayant motivé sa conception. Ce premier chapitre a donc pour objectif de présenter l'organisme d'accueil et de situer le projet dans son contexte général. Il inclut également une étude de l'existant, étape clé permettant d'identifier les limitations actuelles et de définir les axes d'amélioration. Enfin, ce chapitre décrit la méthodologie adoptée pour la mise en œuvre du projet.

1.2 Présentation de l'organisme d'accueil

CargoWeen (S.A.R.L) est une start-up tunisienne établie à Nabeul, en Tunisie, qui a gagné le label Startup Act en octobre 2022.

Spécialisée dans la numérisation du fret aérien, l'entreprise propose une plateforme en ligne permettant aux professionnels du transport de rechercher, comparer et réserver des services de fret aérien de manière simple et accessible, 24/24 et 7/7. CargoWeen joue un rôle crucial dans la modernisation du secteur en proposant des solutions novatrices pour répondre aux besoins croissants du transport aérien (voir annexe A).



FIGURE 1.1 – Logo CargoWeen

1.2.1 Les services du Cargoween

CargoWeen met l'accent sur la numérisation et l'automatisation du processus de réservation de fret aérien pour les professionnels du transport. Voici un aperçu des principaux services proposés par CargoWeen :

- **Plateforme de Réservation en Ligne :** Cette solution est spécialement conçue pour faciliter la recherche, le comparatif et la réservation d'espaces de fret aérien auprès des compagnies aériennes, pour les transitaires, les commissionnaires en douane et les voyageurs.

- **Suivi et Traçabilité en Temps Réel :** CargoWeen propose une solution de suivi et de traçabilité en temps réel, ce qui permet aux clients de suivre l'état de leurs réservations à tout moment.

1.2.2 Les partenaires de Cargoween

La réussite de notre plateforme repose principalement sur des partenariats stratégiques avec des acteurs clés du secteur, assurant ainsi un service fiable et performant.



FIGURE 1.2 – Les partenaires

1.2.3 Organigramme de l'entreprise

Ci-dessous figure l'organigramme de l'entreprise, détaillant les postes clés, les lignes de reporting et les interactions entre les équipes.

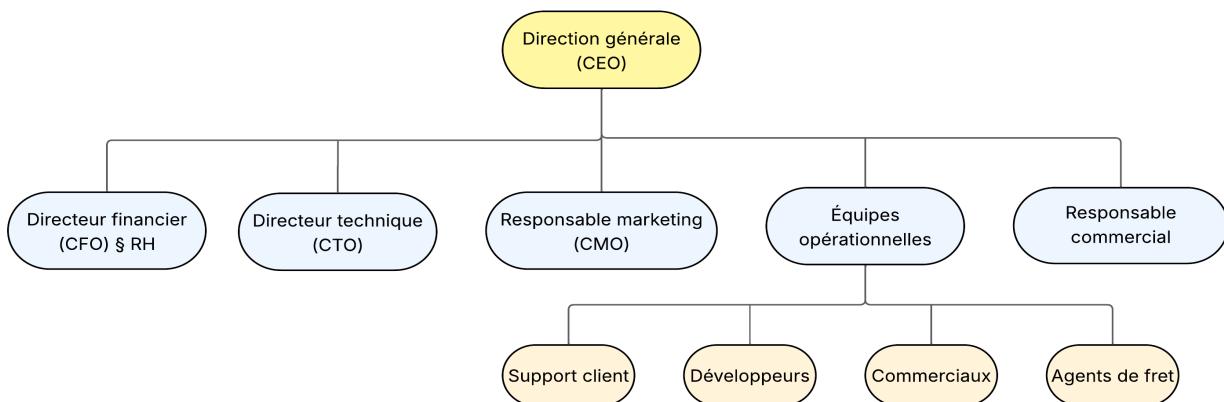


FIGURE 1.3 – Organigramme de l'entreprise

1.3 Présentation du projet

Dans le cadre de notre projet de fin d'études, nous allons développer un système de réservation intelligent, spécialement conçu pour les transitaires. Ce système vise à optimiser la recherche, la comparaison et la réservation des services proposés par les compagnies de transit.

1.3.1 Cadre du projet

Le projet s'inscrit dans le domaine de la logistique et du transport de marchandises, un secteur essentiel pour le bon fonctionnement du commerce mondial. Il vise à répondre aux défis liés à la gestion des flux, à la coordination des acteurs et à l'optimisation des opérations. Dans ce contexte, les outils numériques jouent un rôle crucial pour améliorer l'efficacité, la traçabilité et la réactivité des services logistiques.

1.3.2 Contexte du projet

Dans un contexte de mondialisation accrue et d'accélération des échanges internationaux, la gestion du transport de marchandises représente un défi logistique majeur pour les entreprises. Les transitaires jouent un rôle clé dans cette chaîne : ils agissent comme intermédiaires entre les expéditeurs et les différents acteurs du transport (compagnies aériennes, douanes, transporteurs terrestres), en assurant la coordination, le suivi et le bon déroulement des opérations logistiques. Actuellement, les utilisateurs doivent comparer manuellement les offres de transport, ce qui entraîne une perte de temps et une baisse d'efficacité. Ce projet s'inscrit dans une démarche de transformation numérique du secteur du fret aérien, en réponse à des besoins concrets d'optimisation des processus, de gain de temps et de fiabilité dans la sélection des partenaires logistiques.

1.3.3 Problématique

Le processus actuel de gestion des réservations auprès des agents transitaires présente plusieurs lacunes, ce qui complique et ralentit les opérations logistiques pour les clients.

Tout d'abord, il s'agit d'un processus largement manuel : après avoir réservé leur fret aérien, les clients doivent eux-mêmes rechercher une société de transit, ce qui implique des démarches supplémentaires, telles que des recherches en ligne ou des appels téléphoniques, entraînant ainsi une perte de temps considérable. Ensuite, on constate un manque d'intégration entre les différentes étapes du transport, notamment entre la réservation du fret aérien et l'organisation du transport terrestre, ainsi qu'une absence de recommandations automatiques pour faciliter la réservation. Cela engendre une rupture dans la chaîne logistique et nuit à l'efficacité globale du processus.

Par ailleurs, la gestion du paiement constitue également une difficulté, les clients étant souvent contraints d'effectuer des transactions manuelles ou de contacter directement les transitaires pour convenir des modalités, ce qui complique et ralentit davantage le processus.

Enfin, cette gestion manuelle peut entraîner des retards de livraison, notamment lorsque les clients ont des difficultés à trouver rapidement un transitaire fiable ou disponible.

1.3.4 Analyse et critique de l'existant

Après avoir réservé un fret aérien auprès des compagnies aériennes, le client doit effectuer des démarches manuelles ou passer des appels téléphoniques pour trouver une société transitaire chargée de transporter la marchandise de l'aéroport à l'adresse de livraison.

Après avoir effectué des recherches approfondies les méthodes existantes de recherche et de réservation manquent fréquemment d'automatisation et d'intelligence nous allons identifier un système novateur susceptible de résoudre certains des problèmes précédemment mentionnés.

- **PagesJaunes** : est une plateforme qui permet aux utilisateurs de rechercher des entreprises en fonction de divers critères, y compris la localisation [1].

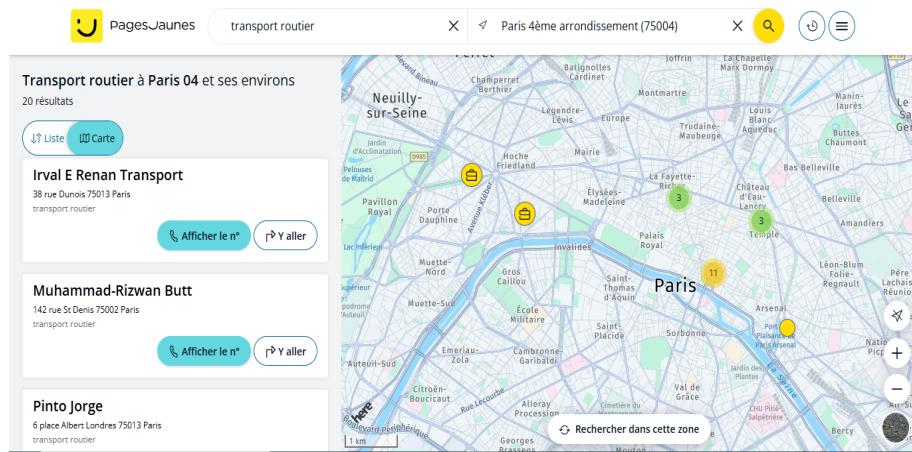


FIGURE 1.4 – Site PagesJaunes

Cependant, même si ce système apporte des solutions importantes, il ne parvient pas à résoudre tous les problèmes existants, car il présente également certains défauts :

- la recherche et de la comparaison manuelle.
- L'adresse doit être saisie manuellement deux fois par l'utilisateur, au début de la recherche pour afficher la liste des entreprises disponibles et lorsqu'il choisit une entreprise, il doit recalculer la distance entre l'adresse saisie et l'emplacement choisi.
- L'utilisateur ne peut pas finaliser le processus en effectuant une réservation sur le site.

1.3.5 Solution proposée

Suite à cette étude et à ce recensement des problèmes, on a conclu que le client a besoin d'une solution innovante, d'automatiser ce processus et de mettre en place un système de recommandation pour optimiser le processus de recherche, de sélection et de réservation des transitaires. Les objectifs sont les suivants :

- **Automatisation de l'extraction des informations** : Extraction automatique de l'aéroport et d'autres informations clés à partir du reçu de réservation d'un fret aérien en PDF.
- **Recommandation intelligente des transitaires** : Mise en place d'un modèle de machine learning pour classer les entreprises de transit en fonction de la distance calculée à partir de l'adresse extraite.
- **Affichage dynamique des résultats** : Les transitaires sont présentés sous forme de liste classée en fonction de leur proximité.
- **Facilité de réservation** : Intégration d'un système permettant à l'utilisateur de sélectionner un transitaire et de finaliser la réservation en quelques clics.
- **Paiement en ligne sécurisé** : Mise en place d'une solution de paiement intégrée permettant aux utilisateurs d'effectuer leurs réservations et de régler directement via la plateforme.

1.4 Méthodologie de travail

Avant de lancer un projet informatique, il est crucial de décider d'une méthodologie de travail et d'un processus de suivi pour obtenir un logiciel fiable. Cette méthodologie présente un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système, afin de rendre ce développement plus fidèle aux besoins du client.

1.4.1 Choix d'approches de gestion du projet

Il est donc primordial, durant la phase de préparation et de mise en œuvre d'un projet, de sélectionner la méthode de gestion qui convient le mieux. Dans ce but le tableau 1.1 offre une étude comparative entre approche classique et approche agile [5].

TABLE 1.1 – Comparaison entre l'approche classique et agile

Étape du projet	Approche classique [5]	Approche agile [5]
Cycle de vie	Phase séquentielle, en cascade ou en V, aucune rétroaction n'est possible, c'est-à-dire que lorsqu'une tâche est terminée, on ne peut pas revenir en arrière	Processus d'itération et incrémental. Si on a un projet de 10 mois, l'itération durera 10 mois et l'incrémental peut être fait tous les mois.
Planification	Au début du projet, le planning est détaillé se basant sur un périmètre d'exigences	La planification se fait sur plusieurs niveaux (macro et micro) et les adaptations sont faites au fur et à mesure selon les changements qui surviennent
Documentation	Elle est produite en quantité importante qui sert de support de communication	la documentation est axée sur des incréments fonctionnels opérationnels pour obtenir le retour client
Equipe	Composée de ressources spécialisées dirigées par un chef de projet	Responsabilité et initiative est de rigueur dans une équipe agile
Qualité	Le contrôle qualité se fait à la fin du cycle de développement	Le contrôle qualité se fait de manière permanente, le client a l'opportunité de visualiser les résultats fréquemment
Changement	Résistance au changement	Le changement fait partie inhérente du processus
Suivi de l'avancement	Analyse différentielle	Le pilotage se fait par des stories et le backlog
Gestion des risques	Processus rigoureux de gestion des risques	Pilotage par les risques, on identifie les risques et on essaie d'identifier les causes d'échecs
Mesure du succès	Les engagements initiaux sont respectés (coûts, budget, niveau de qualité)	La satisfaction du client

Après avoir examiné de près les deux approches, nous avons finalement opté pour l'approche Agile, qui semble vraiment mieux correspondre aux besoins spécifiques de notre projet. En effet, comme le client a des attentes encore floues, il est crucial d'avoir une solution qui soit à la fois flexible et évolutive. L'approche Agile nous permet de maintenir une qualité constante, de repérer et de résoudre rapidement les problèmes, tout en gérant les risques de manière efficace. De plus, elle nous donne une meilleure visibilité sur l'avancement du projet et nous aide à nous adapter plus facilement aux changements. Ainsi, cette méthode est la réponse idéale aux exigences de notre projet.

1.4.2 Méthodologie adoptée

TABLE 1.2 – Comparaison entre Scrum, Kanban, XP et CrispML(Q)

Critère	Scrum [30]	Kanban [30]	XP (Extreme Programming) [30]	CrispML(Q) [2]
Organisation	Itérative en sprints	Flux continu	Itérative et incrémentale	Basée sur 6 phases itératives orientées ML
Rôles	Product Owner, Scrum Master, Équipe de dev	Pas de rôles définis formellement	Client, Coach, Développeurs	Data Scientist, Data Engineer, Domain Expert, Ops
Planification	Sprint Planning (2 à 4 semaines)	Planning flexible et adaptatif	User stories livrées en petites itérations	Planification centrée sur les objectifs métier et validation continue
Suivi de projet	Daily Scrum, Burndown chart	Tableau Kanban, WIP	Tests continus, Intégration continue	Suivi des expériences, des métriques, reproductibilité (MLflow, etc.)
Adaptabilité au changement	Élevée (à la fin de chaque sprint)	Très élevée (changement à tout moment)	Très élevée (feedback rapide, refactoring)	Élevée (approche itérative et orientée qualité du modèle)
Livrables	Incréments de produit à chaque sprint	Livraison continue	Logiciel fonctionnel à chaque itération	Modèles ML validés, pipelines reproductibles, documentation qualité

Nous avons opté pour la méthode Scrum, car elle offre une meilleure prévisibilité des délais de livraison grâce à ses sprints bien définis. De plus, sa structure claire permet de responsabiliser l'équipe et de gérer les priorités de manière plus efficace.

Dans le cadre de la première release du projet, nous avons choisi d'intégrer la méthodologie CRISP-ML(Q) (Cross Industry Standard Process for Machine Learning with Quality assurance) afin de structurer le cycle de vie des modèles de machine learning, en mettant l'accent sur la qualité, la traçabilité et la reproductibilité des résultats.

1.5 Méthodologie Scrum

Scrum est une méthode incrémentale qui consiste en des sprints où l'équipe Scrum va produire un incrément potentiellement livrable du produit. La durée des sprints est déterminée en amont du projet. En général, elle est de 2 ou 3 semaines pour que l'équipe puisse produire l'incrément du produit. Le cadre est organisé autour de trois rôles, d'artéfacts et de cérémonies [3].

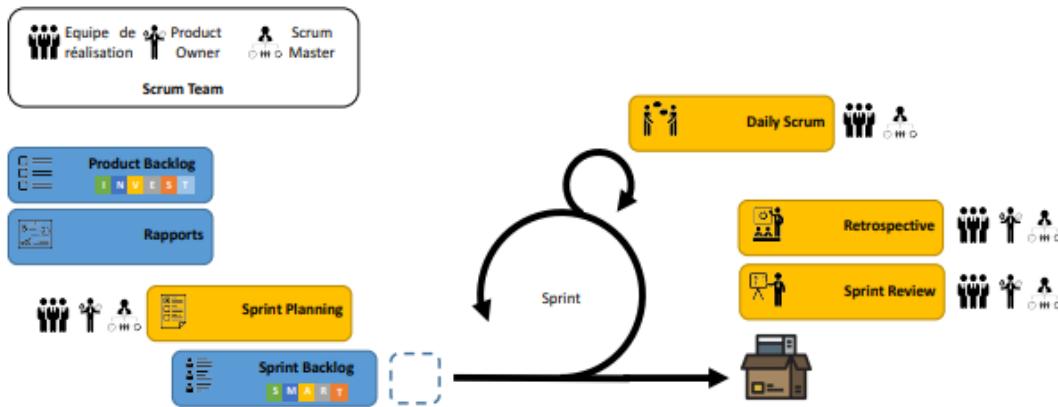


FIGURE 1.5 – Processus de la méthodologie Scrum

1.5.1 Les rôles

- **Le Product Owner (PO)** est responsable du produit, il doit définir sa vision et ses priorités pour maximiser sa valeur. Il est responsable de recueillir les besoins des utilisateurs et de les inscrire dans le 'product backlog', qu'il doit également maintenir et prioriser. C'est lui qui valide si une fonctionnalité a été correctement réalisée.
- **Le Scrum Master** veille à ce que la méthodologie Scrum soit respectée et à ce que l'équipe ne soit bloquée pendant le sprint. Il est chargé d'animer les cérémonies Scrum et de développer l'agilité. En tant qu'intermédiaire entre le Product Owner (PO) et l'équipe, il favorise leur développement autonome et leur amélioration. Il ne se positionne pas comme un chef de projet, mais plutôt comme un coach et un facilitateur, encourageant la collaboration et l'auto-organisation sans imposer de directives.
- **L'équipe de réalisation** a pour mission de concevoir le produit. Elle est autonome et détermine elle-même son organisation pour atteindre ses objectifs. Aucune contrainte ne lui est imposée, et il est crucial d'avoir un dialogue régulier avec le Product Owner (PO) pour parvenir à un consensus sur les décisions.

1.5.2 Les artéfacts

- **Le Product Backlog** est une liste de fonctionnalités, besoins et exigences qui définit la vision du produit. La responsabilité de l'organisation et de la priorisation en fonction de la valeur apportée incombe au Product Owner (PO).
- **Le Sprint Backlog** est un ensemble de récits choisis par l'équipe pour le sprint. L'équipe s'engage à les terminer et à les prouver à la fin du sprint. Ensuite, ces récits sont découpés en tâches

pour faciliter leur accomplissement. Les récits fournissent des réponses aux questions qui, quoi et pourquoi.

1.5.3 Les cérémonies

Sont les différentes réunions cadrant le sprint et ayant lieu de manière fixe à chaque sprint :

- **Sprint Planning** : Pendant cette réunion, l'équipe Scrum examine la complexité, sélectionne les récits qu'elle estime pouvoir démontrer à la fin du sprint, et fixe un but pour le sprint.
- **Daily Scrum** : C'est une réunion qui se tient habituellement debout tous les jours au même endroit. Elle est planifiée dans le temps (Time-boxing) comme toutes les réunions Scrum et ne doit pas dépasser 15 minutes.
- **Sprint Review** : Qui permet à l'équipe de montrer ce qu'elle a fait pendant le sprint. Ensuite, le 'Product Owner' valide si le récit est accepté ou non, en fonction des critères d'acceptation.
- **Retrospective** : Cette réunion marque la fin du sprint et permet à l'équipe Scrum de faire le bilan et de partager son expérience sur le sprint. C'est une étape cruciale qui favorise l'amélioration continue de l'équipe.

1.5.4 Les méthodes de classification

MoSCoW et WSJF sont deux méthodes populaires pour la priorisation des backlogs, mais elles répondent à des besoins différents.

1.5.4.1 La méthode WSJF

Le WSJF “Weighted Shortest Job First” est une méthode de priorisation des besoins (User Stories, Features, Epic, etc.) qui se rapproche du fameux rapport qualité/prix, mais en version agile [31]. Cette méthode est particulièrement utile dans les situations suivantes :

- Quand il y a de nombreux stakeholders (parties prenantes) qui collaborent avec une équipe Scrum.
- Lorsque la planification se fait au niveau des Epics ou des Features, avec plusieurs équipes qui travaillent sur un même produit dans un cadre SAFe.
- Quand une équipe doit gérer plusieurs flux de produits, avec un ou plusieurs Product Owners pour chaque produit. Bien que cette approche ne suive pas les principes du Scrum traditionnel – que je ne soutiens pas – il est courant de voir ce genre de configuration où une même équipe doit jongler avec plusieurs produits en même temps.

Le WSJF est un outil pratique qui, grâce à une formule simple, aide à trouver le meilleur compromis entre la valeur qu'une fonctionnalité apportera une fois livrée et le temps qu'il faudra pour la réaliser. [31]

Pour calculer le WSJF, on se base sur plusieurs critères, évalués selon une échelle inspirée de la suite de Fibonacci, avec des valeurs qui augmentent.

- **User-Business Value** : c'est-à-dire la valeur estimée de la fonctionnalité pour nos utilisateurs, sans notion de temps (en partant du principe qu'elle est déjà en production)

- **Time Criticality** : c'est-à-dire l'urgence à la mettre à disposition de nos utilisateurs (réglementaire, évènements, etc.), et ce peu importe la valeur même de cette fonctionnalité
- **Risk Reduction or Opportunity Enablement** : c'est-à-dire sa capacité à réduire/annuler les risques existants (dépendances, instabilité, sécurité, etc.) et/ou à créer de la valeur indirecte (mieux comprendre les besoins de nos utilisateurs, tester la viabilité d'une solution, etc.)
- **Job Size** : Correspond à l'effort estimé pour développer la fonctionnalité. Il s'agit d'une mesure relative de la complexité ou du volume de travail nécessaire, évaluée par l'équipe de développement.

Par définition la formule du WSJF est la suivante :

$$\text{WSJF} = \frac{\text{Business Value} + \text{Time Criticality} + \text{Risk Reduction and/or Opportunity Enablement}}{\text{Job Size}}$$

FIGURE 1.6 – WSJF

1.5.4.2 La méthode MoSCoW

Dai Clegg, consultant pour Oracle UK, a développé cet outil qui ordonne les exigences d'un projet en fonction de leur degré de criticité. L'idée est d'aller au-delà d'une classification plus ou moins binaire reposant sur l'importance, afin de mieux comprendre pourquoi choisir tel ou tel élément plutôt qu'un autre. Le MoSCoW utilisé dans les projets Agile est un outil qui aide à la décision [9].

- **M - Must have this** Ce qui doit être fait
- **S - Should have this if at all possible** Ce qui devrait être fait dans la mesure du possible.
- **C - Could have this if it does not affect anything else** Ce qui pourrait être fait dans la mesure où cela n'a pas d'impact sur les autres tâches.
- **W - Won't have this time but would like in the future** Ce qui est exclus dans l'immédiat, mais qui doit être conservé pour un traitement ou une intégration ultérieure.

Les « o » ont un rôle cosmétique, et servent à rendre le mot phonétiquement prononçable.



FIGURE 1.7 – MoSCoW

TABLE 1.3 – Classification des éléments par niveau de priorité

Acronyme	Priorité	Description
M	1	Nous plaçons dans cette catégorie, des éléments critiques, ou vitaux. Sans ces éléments, le projet sera considéré comme en échec. Les éléments classés dans cette catégorie ne sont pas négociables.
S	2	Ces éléments sont importants, et ils contribuent à la réalisation des objectifs du projet, mais ils sont jugés moins prioritaires que les éléments de la précédente catégorie.
C	3	L'absence de ces éléments ne remet pas en cause la réussite du projet. Ces éléments ne sont pas prioritaires, et peuvent donc être retirés si nous n'avons ni les ressources, ni le temps de les construire. On parle d'éléments de confort.
W	4	Ces éléments peuvent être abandonnés sans aucun impact au niveau du projet. Il peut être bon de les conserver pour un développement ultérieur.

1.5.5 Méthodologie Crisp ML (Q)

Actuellement, il n'existe pas de pratique standard pour créer et gérer des applications de machine learning (ML). Les projets de machine learning sont souvent mal structurés, manquent de répétabilité et ont tendance à échouer sur le long terme. Il est donc crucial de mettre en place un processus qui assure la qualité, la durabilité, la robustesse et le contrôle des coûts tout au long du cycle de vie du machine learning [2].

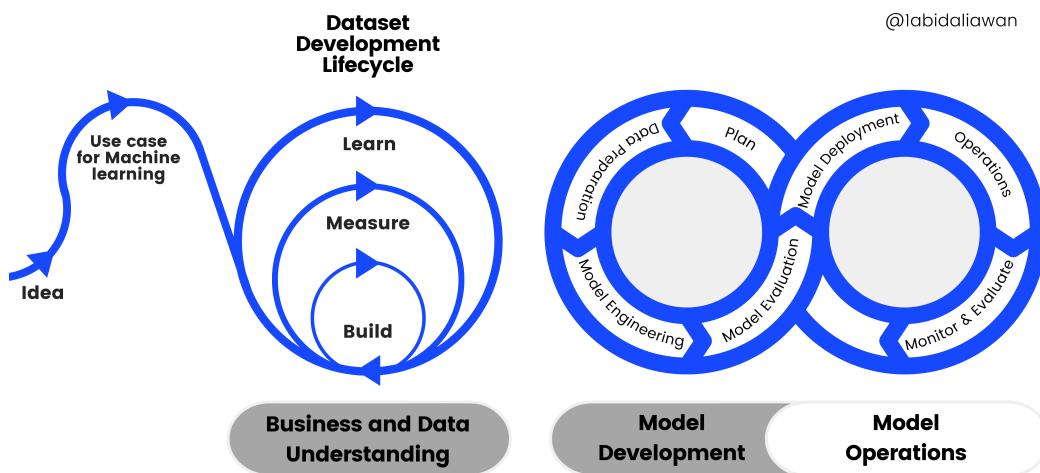


FIGURE 1.8 – Processus du cycle de vie du développement de l'apprentissage automatique

Le processus standard intersectoriel pour le développement d'applications d'apprentissage automatique, connu sous le nom de CRISP-ML(Q) "Cross Industry Standard Process for Machine Learning with Quality assurance", est une version améliorée de CRISP-DM. Il a été conçu pour garantir la qualité des produits d'apprentissage automatique.

CRISP-ML (Q) comporte six phases distinctes :

- Compréhension des affaires et des données

- Préparation des données
- Ingénierie des modèles
- Évaluation du modèle
- Déploiement du modèle
- Surveillance et maintenance

Ces étapes nécessitent une itération et une exploration continues pour construire une meilleure solution. Même si le cadre est ordonné, les résultats d'une étape ultérieure peuvent déterminer s'il est nécessaire de réexaminer l'étape précédente.

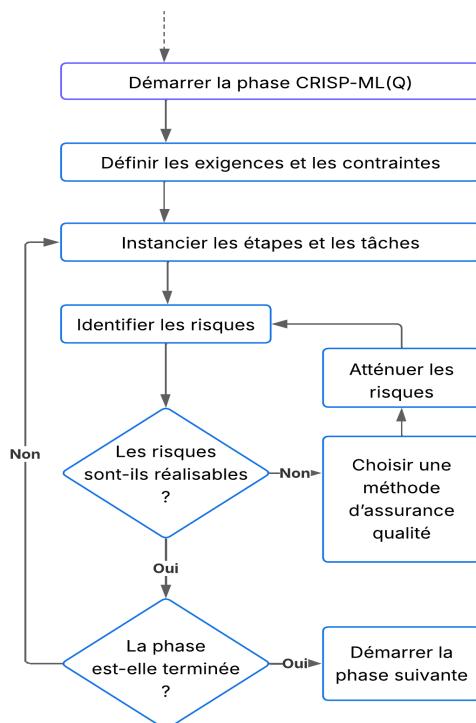


FIGURE 1.9 – Assurance qualité à chaque étape

Des méthodes d'assurance qualité sont intégrées à chaque étape du cadre. Cette approche inclut des exigences et des contraintes, comme des indicateurs de performance, des critères de qualité des données et de robustesse. Elle aide à minimiser les risques qui pourraient nuire au succès des applications de machine learning. Sa mise en œuvre repose sur une surveillance et une maintenance continues de l'ensemble du système [2].

Dans ce qui suit, nous décrivons chacune des six étapes du CRISP-ML(Q) [14] :

- **Compréhension des affaires et des données** Cette phase initiale vise à définir avec précision les objectifs commerciaux du projet en s'appuyant sur une compréhension approfondie des données disponibles. Elle commence par une identification claire du problème à résoudre et des attentes de l'entreprise, en tenant compte des besoins spécifiques des parties prenantes concernées. Il s'agit également de recueillir, d'examiner et d'évaluer les données pertinentes afin d'en vérifier la disponibilité, la qualité et la pertinence pour le projet. Enfin, cette étape permet de fixer des critères de réussite mesurables qui serviront à évaluer l'efficacité et l'impact des solutions mises en œuvre.

- **Préparation des données** Cette phase est consacrée à la préparation des données en vue de la modélisation, en assurant leur qualité, leur cohérence et leur pertinence. Elle comprend des opérations essentielles telles que le nettoyage et le prétraitement des données, la gestion rigoureuse des valeurs manquantes et des valeurs aberrantes, ainsi que l'ingénierie des fonctionnalités pour extraire des informations utiles à partir des variables brutes. Enfin, une sélection des caractéristiques les plus pertinentes est effectuée afin d'optimiser la performance des futurs modèles d'apprentissage.
- **Ingénierie des modèles** Cette phase a pour objectif de développer et d'évaluer plusieurs modèles d'apprentissage automatique afin d'identifier la solution la plus adaptée au problème posé. Elle implique la sélection d'algorithmes pertinents en fonction de la nature des données et des objectifs du projet, l'entraînement de ces modèles sur les jeux de données prétraités, ainsi que l'analyse comparative de leurs performances à l'aide de métriques d'évaluation appropriées. Le modèle offrant les meilleurs résultats est ensuite retenu pour être utilisé dans l'étape suivante du projet.
- **Évaluation du modèle** Cette phase vise à assurer la qualité globale du modèle et du projet en mettant en place des mécanismes de contrôle rigoureux. Elle inclut la validation et les tests approfondis du modèle afin de vérifier sa robustesse, sa cohérence et son alignement avec les objectifs fixés. Elle aborde également les questions liées à la qualité des données, aux biais potentiels et aux enjeux d'équité. L'interprétabilité et l'explicabilité du modèle sont également prises en compte pour garantir la transparence des résultats. Enfin, les performances du modèle sont évaluées selon les critères de réussite définis en amont du projet.
- **Déploiement du modèle** Cette phase correspond au déploiement du modèle en environnement de production afin de le rendre accessible aux utilisateurs finaux. Elle comprend le packaging du modèle, son intégration avec les systèmes existants et la mise en place de mécanismes de surveillance continue pour évaluer ses performances dans le temps réel. Cette étape inclut également la détection et la gestion de la dérive des concepts, nécessitant éventuellement le recyclage du modèle pour maintenir sa pertinence et sa précision. Enfin, il est essentiel de communiquer clairement les résultats et les insights générés par le modèle aux parties prenantes pour favoriser une prise de décision éclairée.
- **Surveillance et maintenance** Cette phase est dédiée à la surveillance continue du modèle déployé afin d'assurer son bon fonctionnement et sa performance dans le temps. Elle implique le suivi régulier des indicateurs de performance pour détecter toute dégradation éventuelle, l'identification proactive des problèmes et l'ajustement du modèle en conséquence, notamment par son réentraînement avec de nouvelles données. Elle comprend également la maintenance de l'infrastructure technique et des dépendances associées, garantissant ainsi la stabilité et l'évolutivité du système en production.

1.6 Planification de projet

Pour bien organiser un projet, il est essentiel de commencer par la planification. Cette méthode permet d'estimer la durée totale du projet ainsi que le temps moyen requis pour exécuter chaque tâche. Le projet est fractionné en N tâches, qui se dérouleront du 3 février au 10 juin 2021. Le diagramme de GANTT ci-dessous illustre la planification détaillée de ce projet.

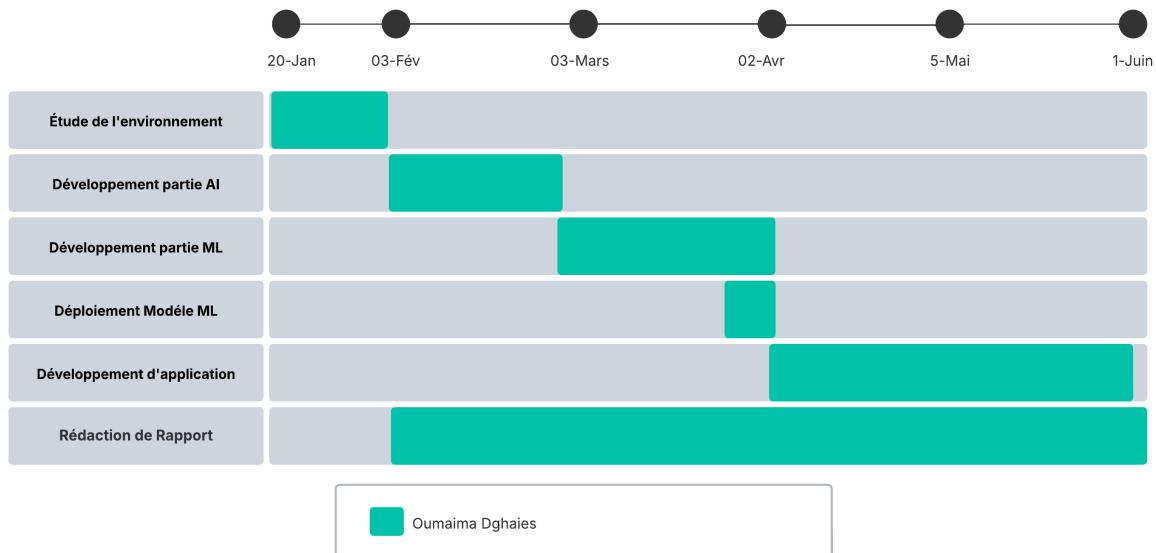


FIGURE 1.10 – Planification de projet

1.7 Planification des sprints

Dans le but de garantir une gestion efficace du projet, nous avons adopté la méthodologie agile Scrum, qui se base sur un développement itératif et incrémental. Le projet est séparé en deux sprints, pour livrer progressivement les fonctionnalités et garantir des tests continus.



FIGURE 1.11 – Planification des sprints

1.8 Conclusion

Ce premier chapitre a permis d'introduire le cadre institutionnel et opérationnel du projet, en mettant en lumière les attentes de l'organisme d'accueil et les contraintes à prendre en compte. L'étude de

l'existant a révélé les besoins spécifiques en matière d'automatisation et d'optimisation des processus logistiques, posant ainsi les bases sur lesquelles s'appuie la conception de la solution proposée dans les chapitres suivants.

Chapitre 2

État de l'art et spécifications des besoins

2.1 Introduction

Afin de concevoir une solution adaptée aux besoins identifiés, il est indispensable de s'appuyer sur les avancées récentes dans le domaine de l'intelligence artificielle. Ce chapitre est consacré à l'état de l'art des technologies utilisées, notamment dans les domaines de l'extraction d'informations, de la reconnaissance de texte, de la géolocalisation et des systèmes de recommandation et il présente également la phase de conception de la solution.

2.2 intelligence artificielle (IA)

L'intelligence artificielle est un domaine scientifique qui cherche à concevoir des ordinateurs et des machines capables de réfléchir, d'apprendre et d'agir comme le ferait un humain, ou qui implique des données dont l'échelle dépasse ce que les humains peuvent analyser [21].

L'IA englobe un large éventail de sujets, allant de l'informatique et de l'analyse de données à l'ingénierie matérielle et logicielle, sans oublier la linguistique, les neurosciences, et même des domaines comme logistique, la philosophie et la psychologie.

Intelligence artificielle dans le domaine logistique, s'étendent de la prévision de la demande à l'optimisation des itinéraires de transport, sans oublier la gestion proactive des risques. Avec cette technologie, les entreprises ont la possibilité d'anticiper les besoins de leurs clients, de diminuer les coûts opérationnels et d'améliorer la qualité du service.

Pour les entreprises, l'IA se traduit par un ensemble de technologies, principalement axées sur le machine learning et le deep learning. Ces outils sont utilisés pour analyser des données, faire des prédictions, classer des objets, traiter le langage naturel, formuler des recommandations, et même récupérer des données de manière intelligente, entre autres.

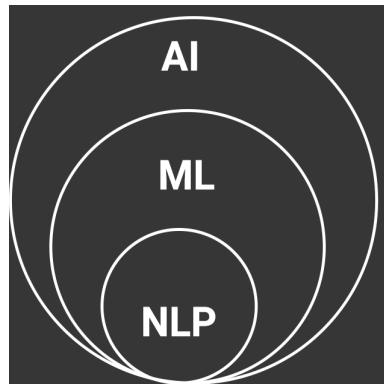


FIGURE 2.1 – intelligence artificielle

Le Machine Learning, le Deep Learning et le Natural Language Processing (NLP) sont indissociables et constituent des domaines imbriqués de l'IA. Pour comprendre comment le Machine Learning, le Deep Learning et le NLP interagissent entre eux, il faut les voir comme des éléments complémentaires [26] :

- **Machine Learning comme base** : Le Machine Learning va fournir les bases des techniques générales permettant de développer des modèles d'apprentissage permettant de traiter des types de données variés, notamment du texte.
- **Deep Learning pour la complexité** : Le Deep Learning, suite à sa capacité à travailler avec des réseaux de neurones profonds, est notamment utile pour traiter la complexité et les nuances du langage naturel dans le NLP.
- **NLP comme application spécifique** : Le NLP est un domaine d'application où le Machine Learning et le Deep Learning sont appliqués pour résoudre des difficultés spécifiques liées au langage humain.

2.3 Machine Learning (ML)

L'apprentissage automatique est une technique qui permet aux ordinateurs d'apprendre et de prendre des décisions à partir de données sans être au préalable programmés explicitement. Il s'agit d'un entraînement d'un modèle informatique sur un ensemble de données, celui-ci soit capable par la suite de faire des prédictions ou d'effectuer des décisions sur la base de modèles et de relations dans les données [34].

Il existe trois grands types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

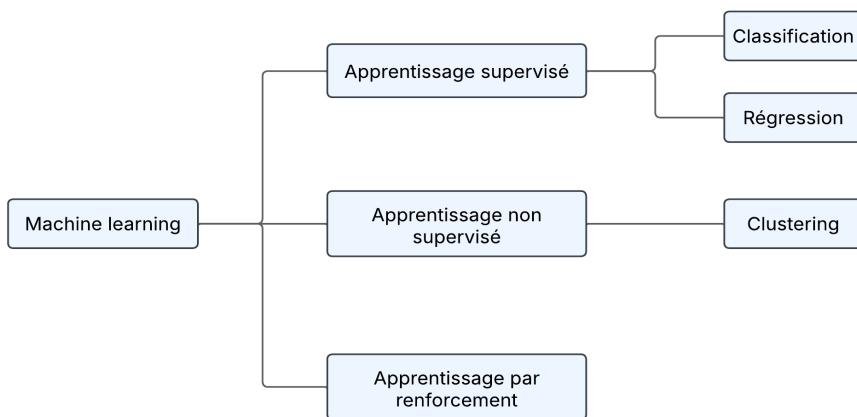


FIGURE 2.2 – Machine Learning

2.3.1 Apprentissage supervisé

le modèle reçoit des données d'apprentissage étiquetées, y compris des données d'entrée et les résultats corrects correspondants. L'objectif est de permettre au modèle de faire des prédictions sur la sortie pour de nouvelles données inédites en se basant sur les modèles qu'il a appris à partir des données d'apprentissage [34].

Les algorithmes de régression, d'une part, et de classification, d'autre part, font partie des algorithmes d'apprentissage supervisé.

- **Classification :**

La classification est une méthode d'apprentissage automatique qui permet à un modèle d'apprendre à donner une étiquette de classe à une donnée d'entrée. Il s'agit d'une tâche d'apprentissage supervisé, où le modèle est entraîné sur un ensemble de données qui comprend des exemples de données d'entrée et les étiquettes de classe y associées.

Pour effectuer de la classification, de nombreux algorithmes peuvent être employés, tels que [35] :

- **Régression logistique** s'agit d'un modèle linéaire de classification binaire. Il sert à prédire la probabilité de survenance d'un événement. L'objectif de la régression logistique est de trouver les meilleurs coefficients (poids) tels que l'erreur entre la probabilité prédite et le résultat observé.
- **Arbres de décision** s'agit de modèles en forme d'arbre qui prennent des décisions basées sur les valeurs des différentes caractéristiques. Ils sont parfaits pour la classification, que ce soit binaire ou multi-classes. Les arbres de décision ont plusieurs atouts, comme leur simplicité et leur capacité à s'intégrer facilement avec d'autres systèmes.
- **Classification par forêt aléatoire** se profile comme une méthode d'ensemble, dans la mesure où elle combinera les prédictions de plusieurs arbres de décision afin d'en obtenir une plus précise et plus stable. Elle est moins portée à l'overfitting qu'un arbre de décision unique puisque ici les prédictions des arbres individuels sont moyennées.
- **K-Voisins les plus proches** est une méthode non paramétrique qui associe un point à la classe de ses plus proches voisins. La méthode KNN présente des avantages notamment sa simplicité et sa facilité de mise en œuvre, elle peut traiter des données numériques et catégorielles et ne fait aucune hypothèse sur la distribution sous-jacente des données.

- **Régression :**

Les algorithmes de régression ont pour but de modéliser les relations entre les entrées et les sorties et d'effectuer des prédictions sur cette relation. La régression peut être utilisée pour des variables dépendantes continues ou catégorielles.

Pour effectuer de la régression, de nombreux algorithmes peuvent être employés, tels que [35] :

- **Régression linéaire** utilisé pour prédire une valeur continue à partir d'un ensemble de caractéristiques. Il permet de modéliser la relation entre les caractéristiques et la variable cible en ajustant une droite aux données.
- **Régression polynomiale** s'agit d'un modèle non linéaire utilisé pour ajuster une courbe aux points de données. Il permet de modéliser les relations entre les caractéristiques et la variable cible lorsque la relation n'est pas linéaire. Il repose sur l'idée d'ajouter des termes d'ordre supérieur au modèle linéaire pour saisir les relations non linéaires entre les variables dépendantes et indépendantes.
- **Régression vectorielle de soutien** Tout comme les SVM, la régression vectorielle de support est un modèle linéaire qui cherche à s'ajuster aux données en identifiant l'hyperplan qui maximise la marge entre les variables dépendantes et indépendantes.

2.3.2 Apprentissage non supervisé

le modèle ne reçoit pas de données d'apprentissage étiquetées mais à découvrir des régularités ou des relations dans les données de façon autonome. Cette approche peut être employée pour détecter des classes ou des grappes dans les données ou pour repérer des anomalies ou des modèles particuliers. Les algorithmes de Clustering font partie des algorithmes d'apprentissage non supervisé [34].

- **Clustering :**

est une méthode qui permet d'explorer des données brutes et non étiquetées, en les regroupant en fonction de leurs similitudes ou différences. On l'utilise dans de nombreuses applications, comme la segmentation de la clientèle, la détection de fraudes et l'analyse d'images. Les algorithmes de clustering s'attachent à diviser les données en groupes naturels en identifiant des structures ou des motifs similaires dans des ensembles de données qui ne sont pas catégorisés [35].

- **K-Means** sont des algorithmes spécifiquement utilisés pour résoudre des problèmes de clustering. Ils ont pour principe de diviser et assigner un ensemble de points de données non affectés d'un label en un groupe appelé « cluster ». Ainsi, pour chaque itération de l'algorithme, chaque point se trouve assigné dans un groupe d'éléments présentant des caractéristiques communes. Les points de données peuvent être suivis dans le temps pour pouvoir identifier les changements dans les clusters.

2.3.3 Apprentissage par renforcement

C'est un agent qui apprend à agir sur son environnement afin de maximiser une récompense. Il s'agit d'entraîner un modèle à prendre des décisions en fonction des informations qu'il reçoit de l'environnement [34].

Le Machine Learning permet de construire des modèles prédictifs performants, mais sans une infrastructure adéquate pour gérer leur cycle de vie, leur utilité reste limitée. C'est dans ce contexte que le MLOps entre en jeu.

2.3.4 MLOps

MLOps signifie "Machine Learning Operations" et désigne le processus de gestion du cycle de vie du machine learning, du développement au déploiement et à la surveillance. Il implique des tâches telles que suivi des tests, déploiement de modèles, surveillance des modèles, réentraînement des modèles [13].

La maturité d'un processus de machine learning est généralement évaluée selon trois niveaux, en fonction du degré d'automatisation du workflow [19] :

- **Niveau 0 du MLOps : processus entièrement manuel**

Il s'agit d'un processus classique de science des données, exécuté au début de la mise en œuvre du ML. Ce niveau est de nature expérimentale et itérative. Chaque étape de chaque pipeline, comme la préparation et la validation des données, l'entraînement et les tests des modèles, est exécutée manuellement.

- **Niveau 1 du MLOps : pipeline de ML automatisé**

Le niveau suivant permet d'automatiser complètement l'entraînement du modèle. Ici, nous parlons de l'entraînement continu du modèle. Dès que de nouvelles données sont prêtes, le processus de réentraînement se met en marche. Ce niveau d'automatisation comprend également les étapes de validation des données et du modèle.

- **Niveau 2 du MLOps : système CI/CD automatisé**

Dans la dernière étape, il est essentiel d'implémenter un système CI/CD pour assurer des déploiements rapides et fiables de modèles de machine learning en production. La grande différence avec l'étape précédente réside dans l'automatisation de la création, des tests et du déploiement des données, du modèle de machine learning et des différents éléments du pipeline d'entraînement.

Dans le cadre de la mise en œuvre concrète des pratiques MLOps, l'outil MLflow s'impose comme une solution clé pour le suivi des expériences, la gestion des modèles et l'automatisation du cycle de vie du machine learning.

2.3.5 MLFlow

MLflow est un projet open source qui a pour but de rendre le cycle de vie des projets d'apprentissage automatique beaucoup plus simple. Il offre des fonctionnalités pour suivre les expériences, gérer les flux de travail et déployer des modèles [13].

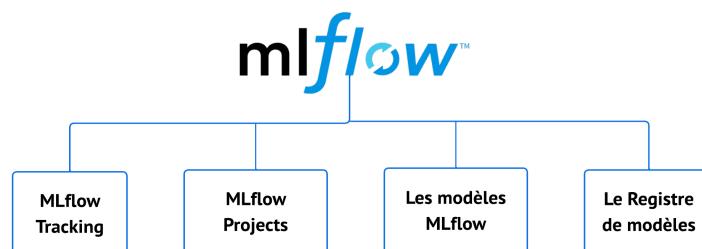


FIGURE 2.3 – Composant mlflow

Il propose quatre composants permettant de gérer le cycle de vie de tout projet de ML :

- **MLflow Tracking** est une API qui vous permet de garder une trace des paramètres, de gérer les versions de vos modèles, de suivre les métriques et de stocker les artefacts, comme le modèle sérialisé, tout au long du cycle de vie de votre projet ML.
- **MLflow Projects** est un format, ou plutôt une convention, qui permet de structurer du code d'apprentissage automatique de manière à ce qu'il soit à la fois réutilisable et reproductible. Il permet de décomposer un code d'apprentissage automatique en petits blocs répondant à des cas d'utilisation très spécifiques , puis les assemble pour créer le workflow d'apprentissage automatique final.
- **Les modèles MLflow** sont une convention de packaging MLflow permettant leur réutilisation ultérieure .
- **Le Registre de modèles** est un espace centralisé qui facilite la gestion du cycle de vie d'un modèle MLflow. Cela inclut des aspects comme son stockage, sa promotion en production ou son archivage. Il enregistre toutes les métadonnées liées à ce cycle de vie, ce qui permet d'assurer une traçabilité : quelle expérience MLflow a produit un modèle donné, qui a effectué la transition du modèle de la phase de test à la phase de production.

L'approche MLOps repose sur l'automatisation et l'intégration continue du cycle de vie des modèles de machine learning, dont MLflow fait partie intégrante. Pour cela, plusieurs outils sont utilisés :

- **Docker** : outils de développement, de services, de contenu fiable et d'automatisations, utilisés individuellement ou ensemble, pour accélérer la livraison d'applications sécurisées.Docker permet de créer facilement des conteneurs et des applications basées sur les conteneurs [8].
- **Kubernetes** : Kubernetes, souvent appelé K8s, est un système open source qui facilite l'automatisation du déploiement, de la mise à l'échelle et de la gestion des applications conteneuriées [18].
- **GitHub Actions** : GitHub Actions simplifie l'automatisation de tous vos workflows logiciels, désormais avec une CI/CD de pointe. Créez, testez et déployez votre code directement depuis GitHub. Améliorez vos revues de code, la gestion des branches et le tri des problèmes [7].
- **Google Cloud Platform** : Google Cloud Platform est une solution de cloud computing proposée par Google, qui offre un hébergement sur la même infrastructure que celle utilisée en interne pour des services comme son moteur de recherche [24].

2.4 Deep Learning (DL)

Le deep learning constitue un sous-domaine du machine learning qui repose sur l'apprentissage de représentations de données, par opposition à des algorithmes ou modèles spécifiques à chaque tâche. L'apprentissage est dit supervisé (avec labels), semi-supervisé ou non supervisé. Les différentes architectures peuvent être classées en trois catégories : réseaux neuronaux profonds, réseaux de croyance profonds, réseaux neuronaux récurrents... etc. Et leurs domaines d'application incluent, notamment : la vision par ordinateur, la reconnaissance vocale, le traitement du langage naturel, la reconnaissance à partir de sons [20].

Il y a essentiellement deux grands types de deep learning :

- **Réseaux neuronaux convolutifs (CNN)** : Les CNN s’organisent autour de la structure du cortex visuel dans lequel sont utilisées des couches convolutives pour l’extraction des caractéristiques de l’image d’entrée. Les CNN sont utilisés en reconnaissance d’images et se sont avérés très efficaces pour détecter des objets dans des images.
- **Réseaux neuronaux récurrents (RNN)** : Les RNN reposent sur un principe simple selon lequel si quelque chose s’est produit au passé, il sera probable qu’il se reproduise dans le futur. Les RNN ont connu une grande réussite dans la modélisation de séquences telles que la parole ou le texte, car ils peuvent apprendre à prédire ce qui va se passer ensuite en fonction de ce qui s’est passé déjà.

2.5 Traitement du langage naturel (NLP)

Le traitement automatique du langage naturel (TALN) est un sous-domaine de l’informatique et de l’intelligence artificielle, qui vise à permettre aux ordinateurs d’interpréter le langage humain. S’appuyant sur la linguistique computationnelle qui consiste à comprendre le fonctionnement du langage, le TALN repose sur divers de modèles basés sur les statistiques, l’apprentissage automatique et l’apprentissage profond. Ces technologies permettent aux ordinateurs d’analyser et de traiter des données sous forme textuelle ou vocale tout en comprenant leur sens y compris les intentions et les émotions du locuteur ou de l’auteur [15].

Au cœur de très nombreuses applications mobilisant le langage, le TALN est impliqué dans les champs de la telles que la traduction de texte, la reconnaissance vocale, la synthèse de texte et des chatbots.

2.5.1 Technologies liées au NLP

Différentes technologies sont utilisées dans le traitement automatique du langage naturel, afin d’analyser et comprendre le langage humain. Parmi les plus courantes, on trouve [15] :

- **Apprentissage automatique** : le PNL s’appuie en très grande partie sur des techniques d’apprentissage automatique comme l’apprentissage supervisé et non supervisé, l’apprentissage profond et l’apprentissage par renforcement pour former des modèles à comprendre et à générer le langage humain.
- **Natural Language Toolkits (NLTK)** : NLTK est, en Python, une bibliothèque open source la plus connue qui fournit des outils pour les tâches de traitement du langage naturel (TALN) telles que la tokenisation, la recherche de radicaux ou l’étiquetage des parties du discours. SpaCy, OpenNLP, CoreNLP font également partie des nombreuses bibliothèques disponibles.
- **Analyseurs** : ce sont des outils qui servent principalement à réaliser les analyses syntaxiques des phrases.
- **Systèmes de synthèse vocale (TTS) et de synthèse vocale (STT)** : les systèmes TTS sont offerts pour convertir du texte écrit en mots prononcés, alors que les systèmes STT offrent plutôt la conversion des mots parlés en texte écrit.
- **Systèmes de reconnaissance d’entités nommées (NER)** : ce sont des systèmes qui effectuent la reconnaissance et l’extraction d’entités nommées (Figure : personnes, lieux, organisations, etc.) au sein du texte.

- **Analyse des sentiments** : c'est une technique qui cherche à saisir les sentiments (ou opinions) exprimés dans un texte, en utilisant diverses techniques telles que les méthodes basées sur le lexique, l'apprentissage automatique et l'apprentissage profond.
- **Traduction automatique** : Le PNL est utilisé pour la traduction d'une langue à une autre via un ordinateur.
- **Chatbots** : Le PNL est utilisé pour les chatbots selon leurs dispositifs de communication avec le ou les autres chatbots et les humains variés : auditifs (oral) ou textuels (écrit).
- **Logiciel d'IA** : le PNL est mis en œuvre dans les logiciels de questions-réponses pour la représentation des connaissances, le raisonnement déductif ainsi que la recherche de l'information.

2.5.2 Fonctionnement du NLP

Travailler dans le traitement automatique du langage naturel (TALN) implique généralement l'utilisation de techniques informatiques destinés à l'analyse et à la compréhension du langage humain [15].

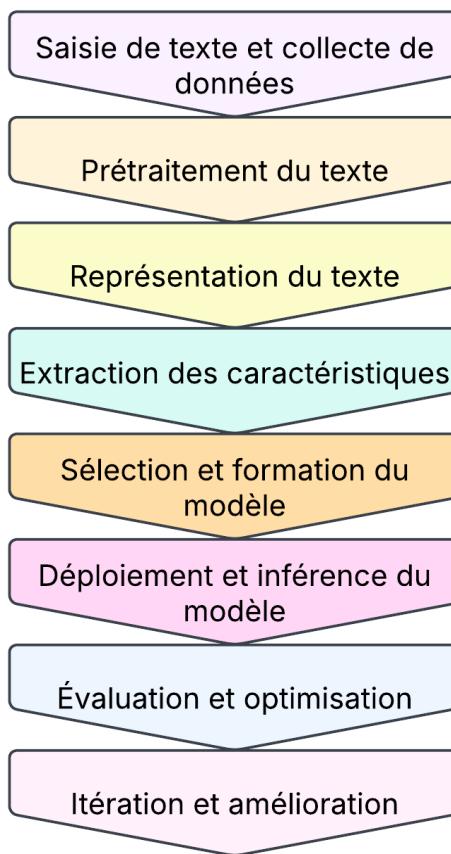


FIGURE 2.4 – processus NLP

2.5.2.1 Saisie de texte et collecte de données

- **Collecte de données** : textuelles à partir de plusieurs sources telles que des centaines des sites Internet, des livres, des réseaux sociaux ou des bases de données

- **Stockage de données** :récupérées sous un format structuré, par exemple une base de données ou une collection de documents.

2.5.2.2 Prétraitement du texte

Le prétraitement est essentiel pour le nettoyage et la préparation des données textuelles brutes en vue de leur analyse. Les étapes de prétraitement courantes comprennent :

- **Tokenisation** :Le texte est décomposé en unités plus petites, mots ou phrases.
- **Minuscules** :La totalité du texte est mise en minuscules pour garantir l'uniformité.
- **Suppression des mots vides** :Suppression des mots les plus fréquents qui n'ont pas de signification significative par rapport à l'objet d'étude (ex. « et », « le », « est »,etc.).
- **Suppression de la ponctuation** :Suppression des signes de ponctuation.
- **Racine et lemmatisation** :réduction des mots à leur forme fondamentale. La racine éliminé les suffixes, tandis que la lemmatisation prend en compte le contexte et transformer les mots à leur forme fondamentale significative.
- **Normalisation du texte** :normalisation du format du texte, y compris la correction des fautes d'orthographe, l'expansion des contractions et la gestion des caractères spécifiques.

2.5.2.3 Représentation du texte

- **Sac de mots (BoW)** :Représenter un texte comme une rassemblement de mots, en ignorant la grammaire et l'ordre des mots mais en gardant une trace de la fréquence des mots.
- **Fréquence du terme-Fréquence inverse du document (TF-IDF)** :une statistique qui représente l'importance d'un mot dans un document par rapport à un ensemble de documents.
- **Incorporation de mots** :utilisation de représentations vectorielles denses de mots où les mots sémantiquement similaires sont plus proches les uns des autres .

2.5.2.4 Extraction des caractéristiques

Extraction des caractéristiques significatives des données textuelles qui peuvent être utilisées pour les tâches de PNL variées.

- **N-grammes** : Capture de séquences de N mots pour garder un certain contexte et rang des mots.
- **Fonctionnalités syntaxiques** : Utilisation de balises de parties du discours, de dépendances syntaxiques et d'arbres d'analyse.
- **Fonctionnalités sémantiques** : Utiliser les plongements de mots ou d'autres représentations pour saisir le sens et le contexte des mots.

2.5.2.5 Sélection et formation du modèle

Choix et formation d'un modèle d'apprentissage automatique ou d'apprentissage en profondeur pour des tâches PNL spécifiques.

- **Apprentissage supervisé** : l'utilisation de données étiquetées pour l'apprentissage de modèles tels que les machines à vecteurs de support (SVM), les forêts aléatoires, ou des modèles d'apprentissage en profondeur tels que les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN).
- **Apprentissage non supervisé** : utilisation de méthodes telles que le clustering ou la modélisation de sujets à partir de données non étiquetées.
- **Modèles pré-entraînés** : l'utilisation de modèles de langage pré-entraînés tels que BERT, GPT ou des modèles basés sur des transformateurs pré-entraînés sur de grands corpus.

2.5.2.6 Déploiement et inférence du modèle

Mettre en œuvre le modèle formé, c'est-à-dire l'utiliser pour faire des prédictions ou extraire des informations sur de nouvelles données textuelles.

- **Classification de texte** : faire correspondre un texte à une ou plusieurs classes prédéfinies.
- **Reconnaissance d'entités nommées (NER)** : identifier et classifier les entités présentes dans le texte.
- **Traduction automatique** : traduire un texte d'une langue à une autre.
- **Réponse aux questions** : répondre aux questions selon le contexte apporté par les données textuelles

2.5.2.7 Évaluation et optimisation

Évaluation de la performance de l'algorithme PNL en utilisant les différentes mesures de performances telles que la précision, le rappel, le score F1 et autres.

- **Réglage des hyperparamètres** : ajustement des paramètres du modèle pour améliorer la performance.
- **Analyse des erreurs** : étudier les erreurs pour comprendre les faiblesses et améliorer la robustesse du modèle.

2.5.2.8 Itération et amélioration

L'algorithme sera continuellement amélioré grâce aux nouvelles données ajoutées, à l'affinage des méthodes de prétraitement, à l'expérimentation des différents modèles et à l'optimisation des fonctionnalités.

2.6 Hugging Face

Hugging Face, fondée en 2015 par Julien Chaumond et Clément Delangue, est une startup française dont l'objectif est de rendre l'intelligence artificielle accessible à tous.



FIGURE 2.5 – Hugging Face

Hugging Face dispose d'une librairie open-source dédiée au NLP (Natural language Processing) afin d'offrir à ses clients une API permettant d'accéder à des modèles pré-entraînés. Comme ces modèles de machine learning sont déjà entraînés, leur apprentissage et expérimentation sont simplifiés. Par ailleurs, il propose aussi des outils pour gérer les données et les modèles existants, développer et entraîner de nouveaux modèles [33].

Parmi les modèles pré-entraînés par exemple CamemBERT est un modèle BERT pré-entraîné spécifiquement pour le français, visant à pallier les limites des modèles majoritairement entraînés sur des données anglaises ou multilingues. Il surpasse les modèles multilingues dans plusieurs tâches NLP comme la reconnaissance d'entités nommées ou l'inférence. Ce modèle est publié pour encourager la recherche en traitement automatique du français [6].

- **Camembert-base(Fill-Mask)** : Ce modèle pré-entraîné sur des textes français peut prédire un mot manquant dans une phrase.
- **Camembert-ner** : Ce modèle est spécialisé dans la reconnaissance d'entités nommées. Il identifie automatiquement des entités comme des noms , des organisations...

2.7 Reconnaissance optique de caractères (OCR)



FIGURE 2.6 – OCR

Le processus de reconnaissance optique de caractères, ou OCR, vise à convertir une image de texte au format du texte exploitable par une machine. Par exemple, si vous scannez un document, ou

que vous photographiez un ticket de caisse, votre ordinateur va conserver la numérisation sous forme d'un fichier image au format JPEG ou PNG, etc., qui ne pourra pas être ouvert ni modifié dans un éditeur de texte, ou encore interrogé, par exemple pour effectuer une recherche dans son contenu ou une extraction du nombre de mots [22].

2.7.1 Les types OCR

L'OCR ne signifie donc pas seulement numériser des textes sur papier mais aussi de texte dans tout autre format autre que les documents. Puisque ses types et ses applications étant différents, les techniques et approches déployées sont également distinctes [22].

TABLE 2.1 – Les types OCR

Type de ROC	Ce que cela implique
Reconnaissance intelligente des caractères (RIC)	Ceci est très similaire à la reconnaissance de mots, mais au lieu de scanner l'intégralité du texte, il recherche des caractères spécifiques.
Reconnaissance optique de caractères (OCR)	Celui-ci détecte le texte tapé mais, comme son nom l'indique, il n'identifie qu'un seul caractère à la fois.
Reconnaissance optique de mots (OWR)	Semblable à la reconnaissance de caractères, cette fonctionnalité identifie les mots et le texte au lieu de simplement les caractères dans les images avec des textes saisis.
Reconnaissance optique des marques (OMR)	Les données marquées par l'homme telles que les réponses OMR, les marques sur les bulletins de vote, les coches sur les feuilles de réponses et bien plus encore sont identifiées grâce à cette technique.

2.7.2 Fonctionnement d'OCR

Le fonctionnement de l'OCR (Reconnaissance Optique de Caractères) s'apparente à la capacité humaine à lire un texte, à reconnaître des motifs et des caractères. Dans un processus classique, l'homme lit, puis saisit les informations manquantes en recopiant manuellement les données nécessaires dans un système, un fichier de données ou une base de données.

L'OCR procède de manière un peu différente. La technologie améliore la qualité d'un texte ou d'une image scannée et suit plusieurs étapes pour extraire les données qui ont été capturées.

Examinons en détail les étapes suivantes du processus d'OCR [32] :



FIGURE 2.7 – processus d'OCR

- **Étape 1 : Pré-traitement de l'image**

Dans le cadre de l'extraction des données, la qualité de l'image est doit être améliorée. Le processus d'amélioration des images est aussi appelé phase de pré-traitement des images. Plus l'image ou le document numérisé est clair et de bonne qualité, plus la sortie de données est précise.

Au cours de l'étape de prétraitement, le moteur OCR effectue de manière automatique la recherche d'erreurs et la correction des problèmes. Les techniques couramment utilisées pour améliorer les images ou les documents numérisés sont les suivantes :

- **Redressement** : Le processus consistant à redresser la photo ou le document scanné et à corriger son angle.
- **Binarisation** : Le processus qui consiste à convertir l'image ou le document scanné en noir et blanc. La binarisation améliore la séparation du texte et de l'arrière-plan.
- **Zonage** : Appelé aussi analyse de la mise en page, permettant de repérer les colonnes, les rangées, les blocs, les légendes, les paragraphes, les tableaux et autres éléments.
- **Normalisation** : Le fait de réduire le bruit en modifiant la valeur de l'intensité des pixels en fonction des moyennes des intensités des pixels voisins.

- **Etape 2 :Segmentation**

La segmentation est le processus de reconnaissance d'une ligne de texte à la fois. La segmentation comprend les étapes suivantes :

- **Détection des mots et des lignes de texte** : L'identification des lignes de texte et des mots qui leur sont associés.
- **Reconnaissance du script** : Processus d'identification du script à partir de documents, de pages, de lignes de texte, de paragraphes, de mots et de caractères.

- **Etape 3 :Reconnaissance de texte**

Ainsi, lors de cette étape, l'image ou le document est décomposé en pièces, sections ou zones. Une fois la séparation effectuée, les caractères qui s'y trouvent sont reconnus.

Deux méthodes sont utilisées dans l'étape de reconnaissance des caractères :

- **La mise en correspondance de matrices** : processus qui consiste à faire correspondre chaque caractère à une bibliothèque de matrices de caractères dans laquelle est effectuée la comparaison. Le modèle OCR effectue une comparaison pixel par pixel afin d'étiqueter l'image d'un caractère au caractère correspondant.
- **Reconnaissance des caractéristiques** : Le processus de reconnaissance des modèles de texte et des caractéristiques des caractères à partir d'images. Par exemple, la taille, la hauteur, la forme, les lignes et la structure d'un caractère sont comparées à celles de la bibliothèque existante.

- **Etape 4 :Post-traitement du résultat**

Cette étape recouvre les techniques et les algorithmes qui améliorent la précision des données extraites pour obtenir un résultat optimal. Suivant le processus, les données sont d'abord détectées, puis, si nécessaire, corrigées.

Les données extraites sont comparées à un vocabulaire ou à une bibliothèque de caractères pour des vérifications des grammaires que de la considération contextuelle afin de compléter la phase de post-traitement.

2.7.3 Les modèles OCR

Nous avons utiliser l'OCR pour convertir l'image en document texte, dont le contenu sera stocké en tant que données textuelles grâce à des modèles OCR différents comme Tesseract, EasyOCR... Ces eoutils rendent l'extraction de texte et d'informations précieuses beaucoup plus facile pour les entreprises [25].

- **Tesseract**



FIGURE 2.8 – Tesseract Logo

est un moteur OCR open-source n'utilisant comme entrée que des images de texte imprimé ou écrit. Il a été développé à l'origine par Hewlett-Packard, et son développement a ensuite été repris par Google. C'est pourquoi il est maintenant connu sous le nom de “Google Tesseract OCR”.

- **EasyOCR**



FIGURE 2.9 – EasyOCR Logo

est apprécié pour sa simplicité d'intégration et sa bonne performance, notamment avec des images peu qualitatives. Il propose plus de 80 langues et est facilement intégrable dans des projets Python.

- **Doctr**



FIGURE 2.10 – Doctr Logo

est un outil de reconnaissance de documents qui se spécialise dans les documents structurés (formulaire, fichiers pdf, documents scannés ou similaires) utilisant le deep learning pour la reconnaissance des documents, il est performant pour traiter des documents avec des mises en page variées.

- **Surya OCR**



FIGURE 2.11 – Surya OCR Logo

se distingue par son aptitude pour traiter les documents complexes, par exemple, ceux qui comportent des tableaux ou des éléments mathématiques.

2.8 Analyse des besoins

Dans cette section, nous allons explorer les divers besoins fonctionnels et non fonctionnels nécessaires pour mettre en œuvre notre solution.

2.8.1 Besoins fonctionnels

Les besoins fonctionnels décrivent les principales fonctionnalités attendues par l'utilisateur. Dans notre solution, un seul acteur est pris en compte : le transitaire. :

- Authentification
- Mise en oeuvre d'un système de recommandation des sociétés avec AI
- Mise en oeuvre d'un système de réservation
- Paiement en ligne
- Gestion profil
- Gestion des notifications (notification instantanée)

2.8.2 Besoins non fonctionnels

Les besoins non fonctionnels se réfèrent aux exigences de qualité et aux contraintes techniques que la solution doit respecter, peu importe les fonctionnalités principales. Ils assurent la robustesse, la flexibilité et la durabilité de l'application. Parmi ces besoins, on peut citer :

- **Sécurité** : Le système doit garantir la protection des données sensibles des utilisateurs et prévenir les accès non autorisés.
- **Disponibilité** : La plateforme doit être toujours accessible et fonctionner sans interruption, avec un minimum de pannes.
- **Scalabilité** : La solution doit être capable de s'ajuster à une hausse du volume de données ou à un plus grand nombre d'utilisateurs, sans dégradation notable des performances.
- **Compatibilité hétérogène** : L'application doit fonctionner de manière fluide sur environnements pour garantir une expérience utilisateur cohérente.

- **Simplicité et ergonomie :** L'interface utilisateur doit être conviviale, facile à comprendre et simple à utiliser.
- **Maintenabilité :** Le code source et l'architecture de la solution doivent être organisés de façon à rendre les évolutions, la correction des erreurs et l'ajout de nouvelles fonctionnalités plus simples à l'avenir.

2.8.3 Diagramme de cas d'utilisation

Nous avons traduit la phase de capture des besoins fonctionnels par le diagramme de cas d'utilisation qui regroupe dans un schéma les périphériques du système d'information, l'acteurs et les cas d'utilisation avec les relations qui existent entre eux comme montre la figure 2.12.

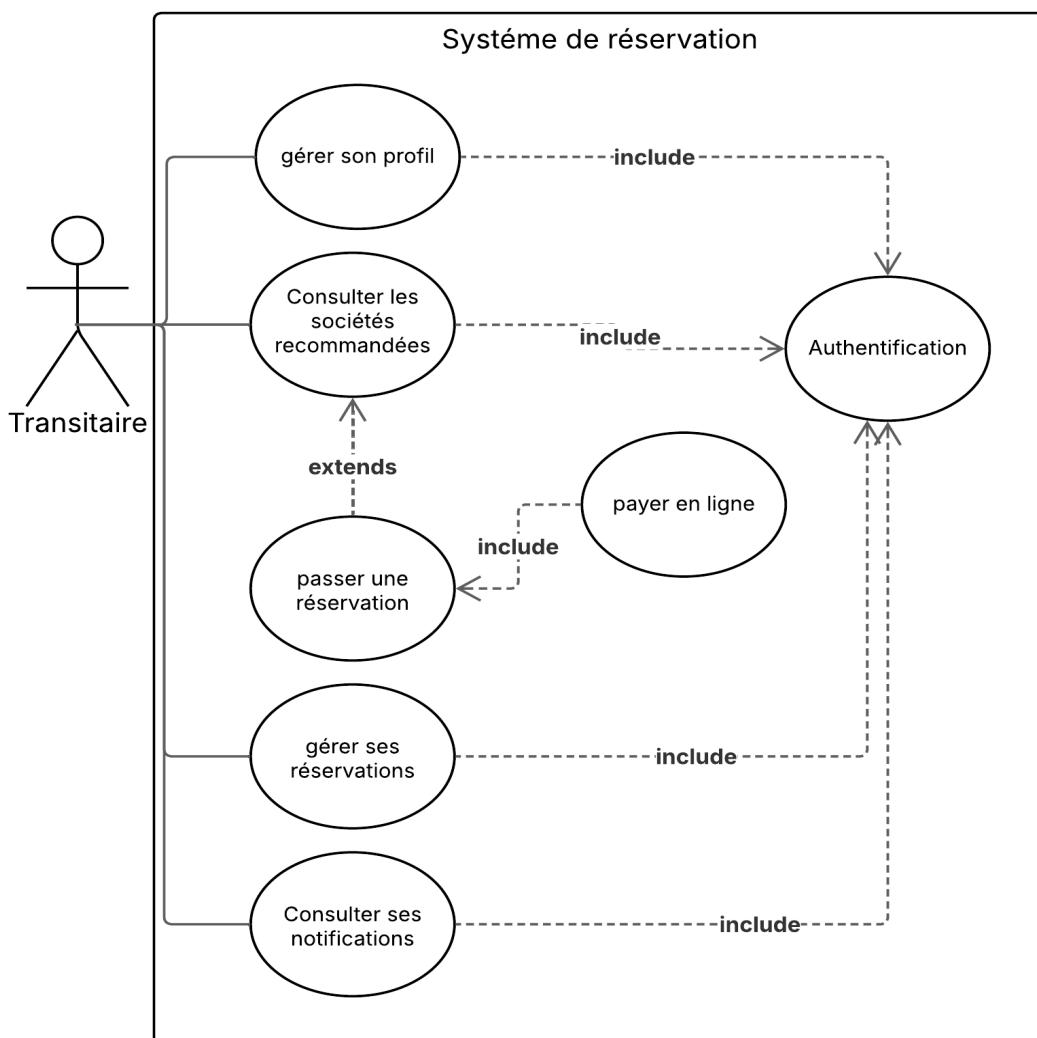


FIGURE 2.12 – Diagramme de cas d'utilisation global

2.8.4 Diagramme de classe

Afin de mieux représenter la structure statique de notre système, nous avons élaboré un diagramme de classe. Ce dernier met en évidence les principales entités de l'application ainsi que les relations entre elles. Il constitue une base essentielle pour la conception et l'implémentation du modèle métier de la solution comme montre la figure 2.13.

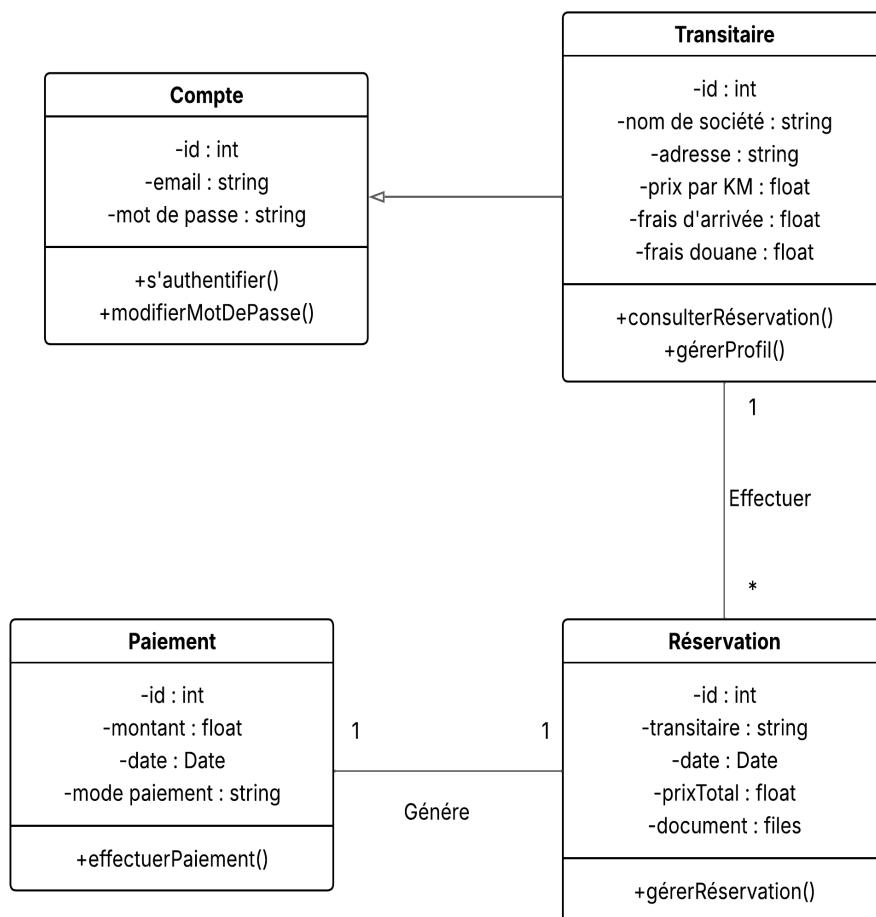


FIGURE 2.13 – Diagramme de classe

2.8.5 Backlog de produit

Dans ce qui suit nous allons décrire les différents user stories de notre projet. Un user story est identifié par un ID. Nous avons choisi la méthode MoSCoW pour sa simplicité et sa capacité à s'adapter aux principes Agiles. Elle facilite une priorisation rapide et collaborative des fonctionnalités, sans nécessiter de calculs compliqués.

TABLE 2.2 – Backlog de produit

ID	Acteur	Fonction globale	id	User story	Priorité
1	Transitaire	Authentification	01	En tant que transitaire, je dois m'authentifier avec un identifiant et un mot de passe pour sécuriser l'accès à mes informations	1
			02	En tant que transitaire, je veux m'inscrire	2
	Gérer son profil	Gérer son profil	03	En tant que transitaire, je veux consulter mon profil	3
			04	En tant que transitaire, je veux modifier mes informations de mon profil	2

ID	Acteur	Fonction globale	id	User story	Priorité
			05	En tant que transitaire, je veux changer mon mot de passe avec confirmation par Email	2
			06	En tant que transitaire, je veux désactiver mon compte	3
		Gérer ses réservations	07	En tant que transitaire, je veux consulter mes réservations	2
			08	En tant que transitaire, je veux consulter mes réservations reçues	1
			09	En tant que transitaire, je veux modifier le statut d'une réservation reçue	1
			10	En tant que transitaire, je veux télécharger mes documents et mes facture	2
			11	En tant que transitaire, je veux consulter les sociétés recommandées avec AI	1
		Consulter les sociétés recommandées	12	En tant que transitaire, je veux passer une réservation d'une société	1
			13	En tant que transitaire, je veux payer une réservation passée	1
		Gérer ses notifications	14	En tant que transitaire, je veux consulter mes notifications	3

2.9 Conclusion

Ce chapitre a permis d'établir une base théorique solide pour le développement du projet. L'analyse de l'état de l'art a mis en évidence les outils et techniques les plus pertinents à intégrer dans la solution, tandis que la phase de conception a permis de définir une architecture claire et cohérente. Ces fondations ouvrent la voie à la mise en œuvre concrète du projet, qui sera détaillée dans le chapitre suivant.

Chapitre 3

Release 1 : Système d'extraction et de recommandation

3.1 Introduction

Ce chapitre est consacré à la première release du projet, structurée autour de deux sprints majeurs : le premier porte sur le développement du système d'extraction d'informations basé sur l'intelligence artificielle, tandis que le second concerne la mise en place d'un système de recommandation exploitant des techniques de machine learning et une approche MLOps.

3.2 Architecture du projet

L'architecture de ce projet est conçue autour de plusieurs modules interconnectés, ce qui permet un traitement complet et automatisé des données liées au transport aérien. Tout commence par une phase d'extraction de données, où l'utilisateur peut soit importer un fichier PDF, soit entrer directement une adresse de livraison, ce qui facilite l'extraction des informations nécessaires.

Ces données alimentent un système de recommandation intégré dans une approche MLOps. Ce système réalise le géocodage des adresses, calcule les distances, effectue des prédictions de coûts grâce à un modèle de machine learning géré par MLflow, puis calcule les tarifs finaux avant de proposer une recommandation. Toutes les données sont centralisées dans une base MongoDB Atlas. En parallèle, un processus d'automatisation via GitHub Actions garantit l'entraînement quotidien du modèle et envoie des notifications par email pour suivre l'état des workflows. En fin de compte, notre modèles MLflow est prêt pour un déploiement à grande échelle.

Le projet comprend également une partie dédiée au développement, qui se divise en une interface utilisateur construite avec Next.js et une API backend en Python. Cela rend la communication entre l'utilisateur et les services de machine learning beaucoup plus fluide.

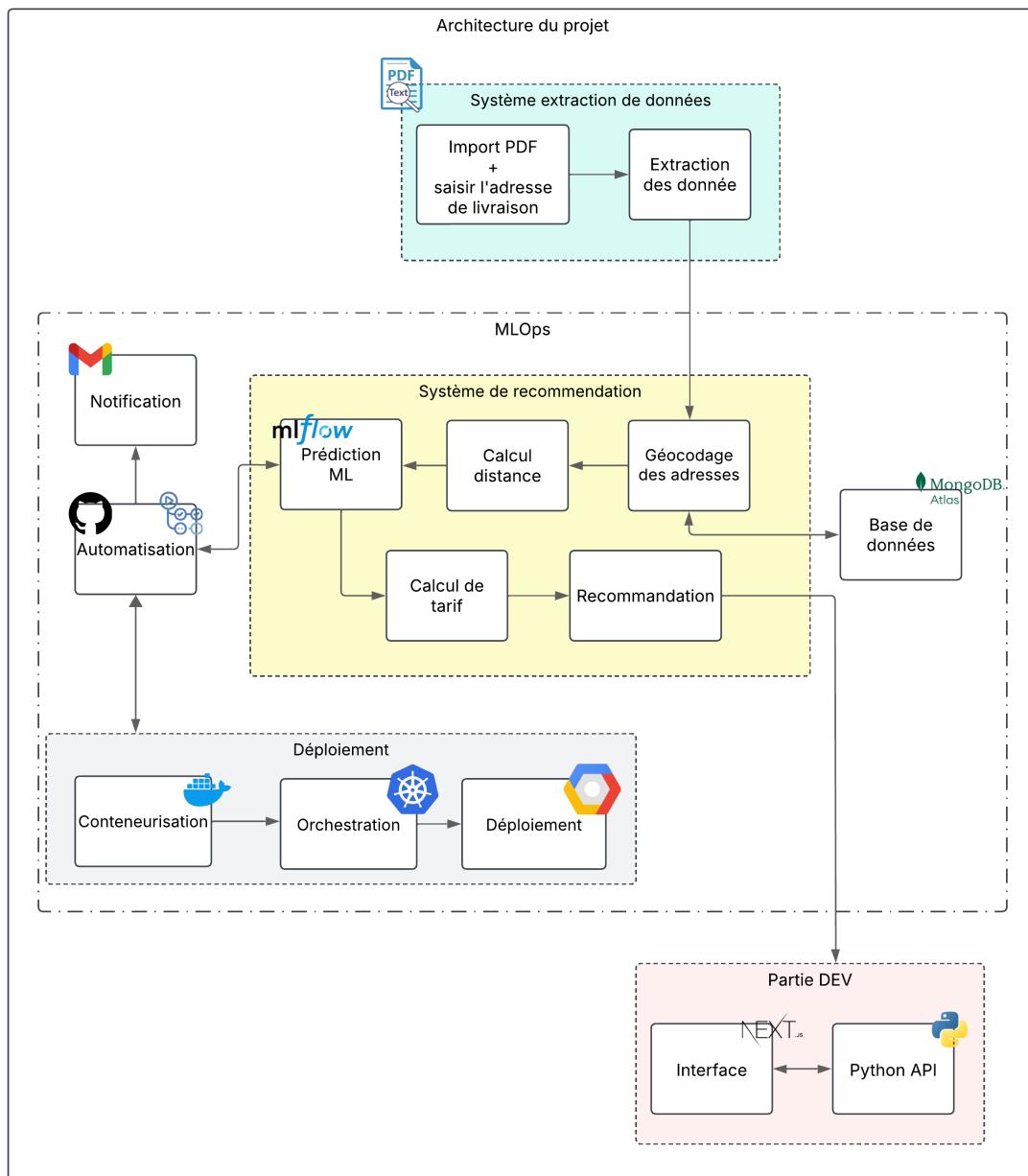


FIGURE 3.1 – architecture du projet

3.3 Sprint1 : Système d'extraction de données

L'architecture de notre système d'extraction de données, illustrée la figure 3.2, repose sur une chaîne de traitement structurée en plusieurs étapes. Nous commençons par la réception d'un document PDF, souvent un reçu ou une confirmation de réservation. Ce document est ensuite soumis à une étape d'OCR qui nous permet de convertir le contenu visuel en texte brut exploitable. Une fois le texte extrait, nous appliquons des techniques de NLP afin de corriger les erreurs éventuelles, compléter les informations manquantes et identifier automatiquement des entités clés. À l'issue de ce processus, nous obtenons les données extraites, prêtes à être utilisées pour les étapes suivantes de notre pipeline ou pour alimenter d'autres systèmes d'information.

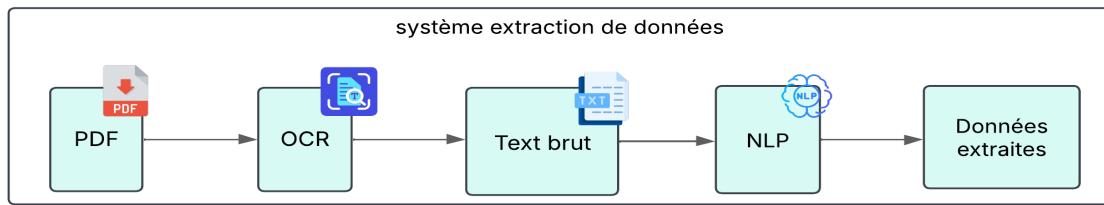


FIGURE 3.2 – Architecture de système d'extraction

3.3.1 Fonctionnement d'OCR

Avant de pouvoir extraire le texte d'un document PDF, il faut d'abord passer par une étape de reconnaissance optique des caractères (OCR), surtout si le PDF est simplement une image. Cette étape est essentielle car elle permet de convertir les images en texte que l'on peut ensuite utiliser pour d'autres traitements.

Étape 1 : Pré-traitement de l'image

Dans notre pipeline, nous commençons par convertir le fichier PDF en images. Pour cela, nous utilisons la bibliothèque pdf2image, qui permet de transformer chaque page du document PDF en une image au format PIL.

La figure 3.3 illustre un exemple de document PDF, représentant un reçu de réservation de vol pour le transport de marchandises.

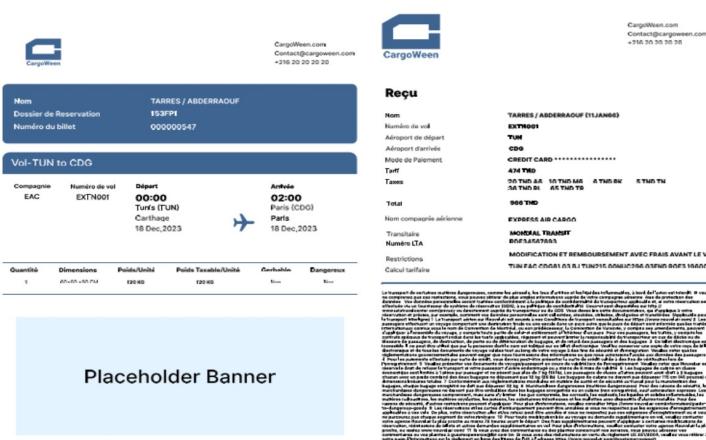


FIGURE 3.3 – Exemple de reçu de réservation

Étape 2 : Segmentation

Pour cette étape, nous avons testé EasyOCR et Tesseract, puis nous avons choisi de travailler avec Tesseract en raison de sa meilleure précision et de son temps de réponse plus rapide. La segmentation du texte est automatiquement prise en charge par Tesseract, le moteur OCR que nous utilisons. Il détecte les lignes, les mots et même les caractères directement à partir des images extraites des fichiers PDF.

La figure 3.4 illustre les résultats des tests effectués avec Tesseract sur un document PDF.

```

    "Tesseract": {
        "Execution Time (s)": 0.56,
        "Text Similarity (%)": 64.87,
        "Raw Text": "nom et prenom : foulen ben foulen\nemail: foulen@email.com\nprix : 320 $\nDétail Réservation\nNuméro de vol : EXTN001\nNom compagnie aérienne : EXPRESS AIR CARO\nAéroport de départ : TUN\nAéroport d'arrivée : CDG\nDate de départ : 18/12/2023 00:00\nNum Lta : RDE34567893\nTransitaire : MONDIAL TRANSIT\nDétail cargaison\nQuantité Dimensions Poids/Unité Poids Taxable/Unité Gerbable\nDangerous\n60x60 x90 CM 120 KG 120 KG Non Non",
        "Extracted Address": "CDG",
        "Flight Number": "EXTN001",
        "Airline Name": "EXPRESS AIR CARO"
    },
}

```

FIGURE 3.4 – Resultat Tesseract

La figure 3.5 illustre les résultats des tests effectués avec EasyOcr sur un document PDF.

```

    "EasyOCR": {
        "Execution Time (s)": 32.14,
        "Text Similarity (%)": 43.22,
        "Raw Text": "nom et prenom foulen ben foulen email: foulen@email.com prix 320 $ Détail Réservation Numéro de vol EXTN001 Nom compagnie aérienne EXPRESS AIR CARO Aéroport de départ TUN Aéroport d'arrivée CDG Date de départ 18/12/2023 00.00 Num Lta RDE34567893 Transitaire MONDIAL TRANSIT Détail cargaison [Quantité [Dimensions [Poids/Unité [Poids Taxable/Unité [Gerbable [Dangereux [60x60 x90 CM 120 KG [120 KG Non Non",
        "Extracted Address": "Adresse d'arrivée non trouvée",
        "Flight Number": "Numéro de vol non trouvé",
        "Airline Name": "Nom de compagnie non trouvé"
    }
}

```

FIGURE 3.5 – Resultat EasyOcr

Étape 3 : Reconnaissance de texte

Une fois que l'image est segmentée, Tesseract applique ses modèles internes pour reconnaître les caractères dans chaque zone identifiée. Il s'appuie sur une approche hybride, combinant la mise en correspondance de formes et la reconnaissance par caractéristiques. Cela nous permet d'obtenir un texte brut, retourné sous forme de chaîne de caractères contenant l'ensemble des informations détectées.

Voici la figure 3.6 le texte que nous avons extrait à l'aide de Tesseract, à partir du reçu de réservation.

```

CargoWeen.com
Contact@cargoween.com
CargoWeen +216 20 20 20 20

Nom TARRES / ABDERRAOUF

Dossier de Reservation 153FP1
Numero du billet 000000547

VolI-TUN to CDG
Compagnie Numéro de vol Départ Arrivée
EAC EXTN001 00:00 02:00
Tunis (TUN) Paris (CDG)
Carthage ) > Paris
18 Dec, 2023 18 Dec, 2023
Quantité Dimensions Poids/Unité Poids Taxable/Unité Gerbable Dangereux
1 60x60 x90 CM 120 KG 120 KG Non Non

Placeholder Banner

```

FIGURE 3.6 – Text extrait avec Tesseract

Étape 4 : Post-traitement du résultat

Le texte brut issu de l'OCR subit ensuite un post-traitement destiné à améliorer sa qualité. Une fonction de correction orthographique est appliquée afin de rectifier les erreurs fréquentes dues à une mauvaise reconnaissance. De plus, un traitement linguistique est réalisé à l'aide du modèle CamemBERT pour effectuer de la reconnaissance d'entités nommées (NER). Cette phase permet d'extraire des informations clés structurées, comme les noms de sociétés, les adresses ou les montants, en apportant un niveau sémantique au texte extrait.

3.3.2 Fonctionnement de NLP

Après l'extraction du texte brut par OCR, nous avons mis en place un pipeline de traitement NLP afin d'extraire des informations structurées à partir des documents scannés. Ce pipeline suit une démarche progressive, décrite comme suit :

Étape 1 : Pré-traitement du texte

Après l'extraction de texte par OCR, nous avons observé que certaines erreurs de reconnaissance pouvaient altérer la qualité des textes (par exemple : lettres mal reconnues, séparations de mots incorrectes, etc.). Pour corriger ces erreurs, nous avons intégré un modèle de type **fill-mask** basé sur **CamemBERT**, un modèle pré-entraîné pour la langue française.

Pour traiter le texte provenant de l'OCR, nous commençons par le tokeniser avec CamembertTokenizer. Nous le découpons en segments de 510 tokens pour respecter la limite des modèles Transformers. Un token <mask> est inséré à la troisième position de chaque segment, puis nous utilisons CamemBERT pour prédire ce mot. Si la prédiction est correcte, le texte est automatiquement corrigé ; sinon, il reste inchangé.

Étape 2 : Représentation du texte

Une fois le texte extrait et corrigé, nous l'affichons page par page dans la console afin de permettre une vérification visuelle. Cette étape nous permet de confirmer que l'OCR et le processus de correction ont bien fonctionné, tout en facilitant la traçabilité en cas d'erreurs ou d'incohérences dans l'extraction des informations.

Etape 3 : Extraction des donnee

Nous procémons ensuite à l'analyse du texte à l'aide d'un pipeline de reconnaissance d'entités nommées (NER) fondé sur plusieurs modèles ont été évalués, notamment FlauBERT et CamemBERT, mais ce dernier a été retenu en raison de ses performances supérieures en extraction d'entités. Par exemple, comme illustré dans la figure 3.7 et la figure 3.8, CamemBERT identifie correctement la localisation "CDG" avec un score de confiance de 0,99, alors que FlauBERT ne détecte que "CD" avec un score de 0,66.

FIGURE 3.7 – Resultat de FlauBert

FIGURE 3.8 – Resultat CamemBert

Le modèle Jean-Baptiste/camembert-ner nous permet d'extraire automatiquement des entités clés telles que les noms, les lieux, ainsi que des informations plus spécifiques comme les numéros de vol, les compagnies aériennes ou les montants. En parallèle, nous utilisons des expressions régulières pour identifier des données structurées supplémentaires (par exemple : poids, dimensions, horaires, dates), en tirant parti de motifs textuels récurrents. Parmi les informations essentielles à extraire pour la suite du traitement figure notamment l'aéroport d'arrivée, souvent représenté sous la forme d'un code IATA (International Air Transport Association).

Étape 4 : Sélection du modèle

Dans notre pipeline, nous sélectionnons les modèles en fonction de la nature spécifique de chaque tâche de traitement. Pour la correction ou la complétion des textes extraits via OCR, nous utilisons le modèle camembert-base, qui s'appuie sur une approche de masking typique des architectures de type BERT. Ce modèle nous permet de combler les lacunes dans les textes en générant des séquences plausibles à partir du contexte, grâce à sa performance reconnue sur la langue française.

Pour l'extraction des entités nommées (NER), nous avons opté pour le modèle Jean-Baptiste/camembert-ner, spécialement entraîné pour identifier des entités clés dans des textes en français, telles que les noms, les lieux, les dates ou encore les montants, éléments essentiels à l'analyse des reçus de réservation.

Notre stratégie de sélection repose ainsi sur une approche modulaire : nous choisissons des modèles adaptés à chaque type de tâche (vision, traitement du langage, extraction structurée), en privilégiant des modèles pré-entraînés de haute qualité, optimisés pour le français et éprouvés dans des cas d'usage concrets.

Étape 5 : Déploiement du modèle

Les modèles CamemBERT (à la fois en mode fill-mask pour la correction et en mode NER pour l'extraction d'entités) sont directement intégrés dans notre pipeline de production via la bibliothèque Hugging Face Transformers. Nous avons conçu notre script pour qu'il puisse être utilisé facilement dans un environnement réel, notamment grâce à une interface utilisateur basée sur Tkinter (bibliothèque graphique libre d'origine pour le langage Python), qui permet à l'utilisateur de sélectionner un fichier PDF de manière intuitive. Cette approche rend notre outil accessible, interactif et prêt à être déployé dans un contexte opérationnel.

La figure 3.9, nous présentons un exemple de données extraites à partir d'un reçu de réservation affiché dans la console.

```
Heure de Départ: 00:00
Heure d'Arrivée: 02:00
Quantité: 1
Dimensions: 60x60 x90 CM
Poids/Unité: 120 KG
Poids Taxable/Unité: 120 KG
Gerbable: Non
Dangereux: Non
Date de départ: 18 Dec,2023
Date d'arrivée: 18 Dec,2023
Nom: TARRES / ABDERRAOUF
Numéro de vol: EXTNO01
Aéroport de départ: TUN
Aéroport d'arrivée: CDG
Total: 966 TND
Nom compagnie aérienne: EXPRESS AIR CARGO
Transitaire: MONDIAL TRANSIT
Numéro LTA: RDE34567893
```

FIGURE 3.9 – Exemple des informations extraites

Étape 6 : Évaluation et amélioration continue

L'évaluation de notre pipeline NLP repose principalement sur une validation manuelle. Pour cela, nous affichons dans la console les textes extraits, leurs versions corrigées, ainsi que les entités nommées reconnues. Cette approche facilite la vérification visuelle de chaque étape du traitement, permettant d'identifier rapidement d'éventuelles erreurs ou anomalies. Elle constitue ainsi un premier pas vers l'amélioration continue de notre système.

3.4 Sprint 2 : Système de recommandation

Dans cette architecture MLOps, nous avons conçu un système de recommandation intelligent pour le domaine logistique. Notre pipeline commence par la collecte et la gestion des données via MongoDB Atlas, qui alimente notre système en informations à jour. Nous utilisons MLflow pour le suivi, la gestion et le déploiement de nos modèles de machine learning. Ces modèles sont intégrés dans un processus plus large incluant le géocodage des adresses, le calcul de distances, le calcul des tarifs, et enfin la recommandation des transitaire les plus pertinents. L'ensemble du système est automatisé grâce à GitHub Actions, ce qui permet de déclencher la conteneurisation avec Docker, l'orchestration avec Kubernetes, et le déploiement sur Google Cloud Platform. En parallèle, un système de notification (via Gmail) permet d'informer les utilisateurs ou les administrateurs des mises à jour importantes. Grâce à cette architecture, nous assurons un déploiement fluide, une maintenance facilitée et une scalabilité optimale de notre solution.

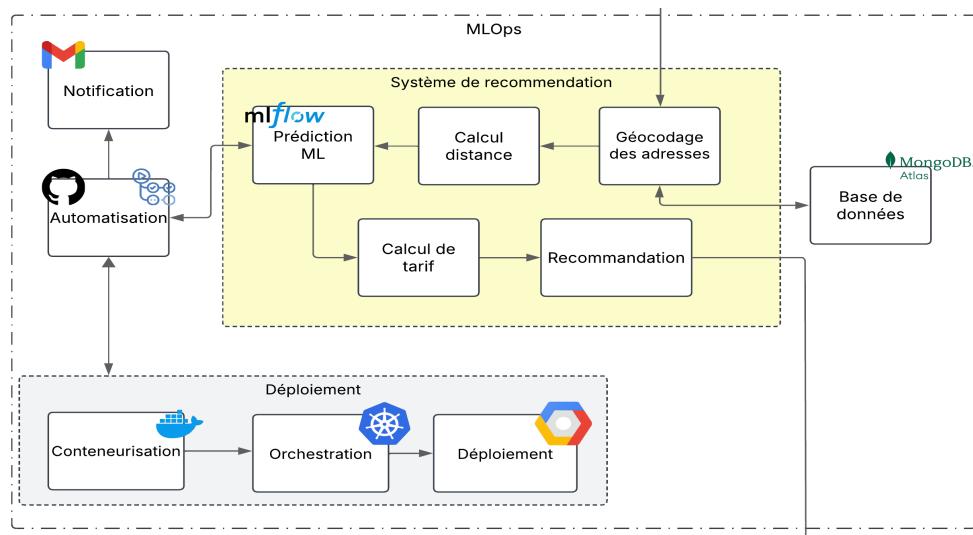


FIGURE 3.10 – Architecture de sprint 2

3.4.1 Géocodage de l'aéroport d'arrivée

Après avoir extrait le code IATA de l'aéroport d'arrivée à partir du reçu de réservation, nous devons impérativement déterminer sa position géographique, notamment le nom, la latitude et la longitude. Cette étape est essentielle pour pouvoir enchaîner avec les calculs de distance entre l'aéroport et les différentes sociétés de transit.

Pour ce faire, nous utilisons le service Nominatim fourni par OpenStreetMap, qui nous permet d'effectuer une recherche géographique à partir d'une requête textuelle, par exemple en saisissant un code IATA comme "CDG".

Ce service nous retourne les coordonnées précises, que nous intégrons ensuite dans notre pipeline pour la suite du traitement comme indiqué la figure 3.11.

```
Nom de l'aéroport: Aéroport de Paris-Charles-de-Gaulle, Tremblay-en-France, Le Raincy, Seine-Saint-Denis, Île-de-France, France métropolitaine, 93290, France
Latitude: 49.0068908
Longitude: 2.5710820
```

FIGURE 3.11 – Géocodage de l'aéroport

3.4.2 Géocodage des Transitaires

Pour calculer la distance entre un transitaire et l'aéroport d'arrivée, nous avons besoin des coordonnées géographiques (latitude et longitude) de chaque point. Or, dans notre base de données, les transitaires sont enregistrés uniquement avec une adresse postale sous forme textuelle. Pour rendre ces données exploitables, nous avons mis en place un processus de géocodage. Cela consiste à utiliser une API de géocodage telle que Nominatim, basée sur les données d'OpenStreetMap. Grâce à cette API, nous transformons chaque adresse textuelle en coordonnées GPS précises, ce qui nous permet ensuite de procéder aux calculs de distances de manière fiable et automatisée.

3.4.3 Calcul de la distance

Une fois que nous avons récupéré les coordonnées géographiques de l'aéroport d'arrivée et de tous les transitaires grâce au processus de géocodage, nous passons à l'étape suivante : le calcul de la distance routière entre ces points. Pour cela, nous utilisons les coordonnées GPS (latitude et

longitude) de chaque transitaire et de l'aéroport, et nous faisons appel à une API spécialisée dans le calcul d'itinéraires, telle que OpenRouteService. Cette approche nous permet de tenir compte des routes réelles empruntées, plutôt que de simples distances à vol d'oiseau. L'API nous retourne une distance exprimée en kilomètres, ce qui rend les résultats facilement intégrables dans nos calculs de tarif ou nos logiques de recommandation opérationnelle.

3.4.4 Prédiction ML

Une fois que nous avons calculé les distances routières entre l'aéroport et les transitaires, nous passons à la phase de machine learning afin de prédire les sociétés les plus pertinentes, en tenant compte des distances calculées.

Le schéma ci-dessus illustre notre pipeline de prédiction en machine learning. Tout débute par une étape de nettoyage des données, indispensable pour assurer la qualité de l'entraînement. Ensuite, nous procédons à une phase de feature engineering où nous construisons les variables explicatives les plus significatives pour notre modèle.

Plusieurs algorithmes sont alors entraînés en parallèle notamment KNN, Random Forest, Régression Logistique et SVM afin d'évaluer diverses approches. Une évaluation rigoureuse des performances permet de comparer ces modèles selon des métriques définies. Le modèle le plus performant est ensuite sélectionné et déployé à l'aide de MLFlow, ce qui nous permet de gérer efficacement le cycle de vie du modèle, de sa mise en production à son suivi continu.

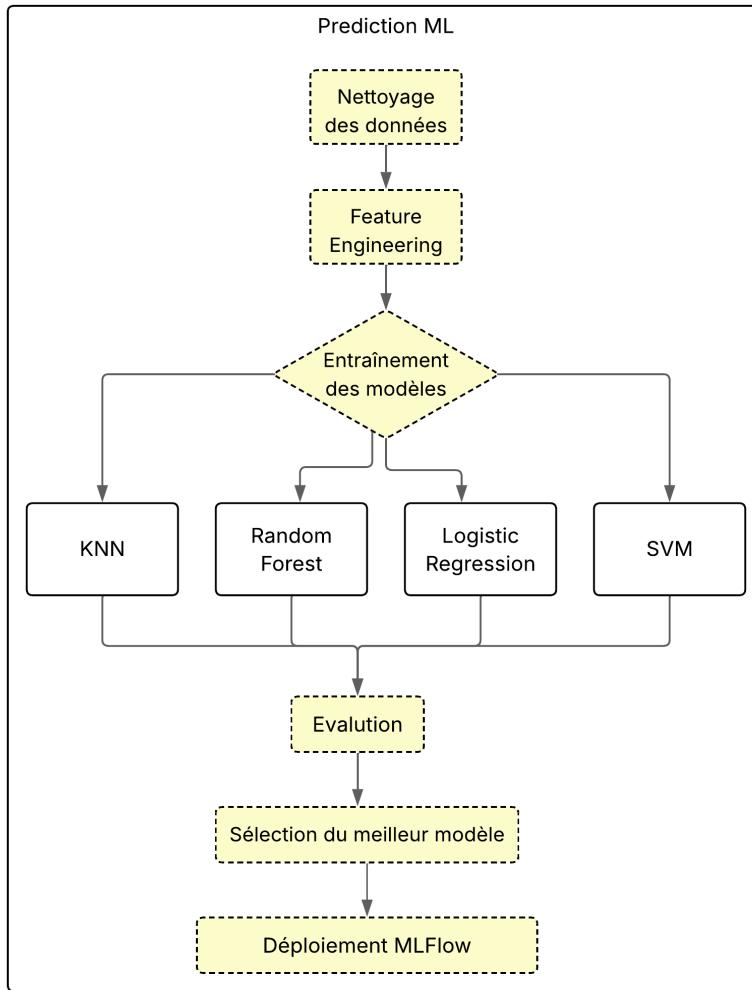


FIGURE 3.12 – pipeline de prédiction en ML

Nettoyage des données

Alors, nous avons utilisé un script Python pour effectuer le nettoyage les données extraites depuis la base MongoDB, pour assurer qu'elles soient cohérentes et de bonne qualité. Pour ça, nous commençons par éliminer les doublons, parce que ça fausse les analyses.

La figure 3.13 illustre les noms des colonnes de notre base de données ainsi que le type associé à chacune d'elles.

Colonnes et types:	
_id	object
Company Name	object
Postal Address	object
Email Address	object
Website URL	object
Responsible Person Name	object
Phone Number	object
Fax	object
Region	object
Pays	object
Ville	object
Code Postal	int64
Prix KM	float64
prix arrivée	int64
douane	int64
dtype:	object

FIGURE 3.13 – Les colonnes et types

Ensuite, on s'occupe des champs textuels **postalAddress**. On va uniformiser ces champs en enlevant les espaces inutiles, et s'il y a des valeurs vides, on les remplace par une valeur par défaut pour ne pas laisser de trous dans les données.

Pour les colonnes qui contiennent des nombres importants, **prixKm**, **prixArrivee** et **douane**, Nous avons transformé ces données en format numérique. Avant ça, nous avons enlevé tous les caractères qui ne sont pas des chiffres, et remplacé les virgules par des points pour que les décimales soient bien reconnues. Par exemple, la colonne **prixKm** :

- avant le nettoyage : 1,5€
- après le nettoyage : 1.5

Pour les colonnes catégorielles, telles que la région, la ville, le pays ou le nom de l'entreprise, nous avons rempli les valeurs manquantes avec le mot "Inconnu", pour garder une certaine cohérence et ne pas avoir de données vides qui pourraient poser problème.

Enfin, nous avons créé un nouveau champ qui correspond au préfixe du code postal, c'est-à-dire les deux premiers chiffres du code postal. Ça nous permet d'avoir une sorte de regroupement géographique simplifié. Et là aussi, nous avons géré les cas où le code postal serait manquant ou incorrect, pour ne pas introduire d'erreurs comme indiqué la figure 3.14.

Traitement du code postal...		
Avant conversion:		
0	75017	
1	75011	
2	1055	
Name:	Code Postal	, dtype: int64
Après conversion:		
	Code Postal	Code_Prefix
0	75017	75
1	75011	75
2	1055	10
Valeurs manquantes dans Code Postal: 0		

FIGURE 3.14 – Exemple de préfixe du code postal

Feature Engineering

Le Feature Engineering joue un rôle essentiel dans la préparation des données pour le machine learning. Il s'agit de transformer les données brutes en variables pertinentes et exploitables par les algorithmes.

Dans notre cas, nous commençons par transformer la variable cible "Prix KM" en une variable binaire : elle est qualifiée de "élevée" ou "faible", en fonction de sa médiane comme indique la figure 3.15. Cela permet de reformuler le problème comme une tâche de classification binaire, plus adaptée à certains modèles.

```
Médiane de la colonne cible (Prix KM): 1.2
Distribution des classes cibles:
faible    76
élevé     47
Name: count, dtype: int64
```

FIGURE 3.15 – Exemple de médiane de la colonne

Ensuite, nous traitons les variables catégorielles telles que la région, le pays ou la ville. Celles-ci sont converties en valeurs numériques via le **LabelEncoder**, afin de les rendre exploitables par les algorithmes de machine learning.

Par ailleurs, les colonnes numériques sont séparées, puis standardisées grâce à **StandardScaler**.

```
Noms des features: ['Code Postal', 'prix arrivée', 'douane', 'Postal Address', 'Region', 'Pays', 'Ville']
Exemple de données transformées (première ligne):
[ 0.73092645  0.          0.        47.         12.          3.
 23. ]
```

FIGURE 3.16 – Exemple des données transformées

Enfin, toutes les données transformées sont regroupées pour constituer la matrice finale d'entraînement, prête à être utilisée pour l'apprentissage de différents modèles de classification.

Entraînement des modèles

Une fois nos données prêtes, nous avons entamé la phase d'entraînement des modèles de classification pour prédire si le prix par kilomètre (prixKm) est élevé ou faible.

La figure 3.17 présente le rapport de classification obtenu en utilisant une répartition de 80% des données pour l'entraînement et 20% pour les tests.

	precision	recall	f1-score	support
faible	0.75	0.88	0.81	17
élevé	0.60	0.38	0.46	8
accuracy			0.72	25
macro avg	0.68	0.63	0.64	25
weighted avg	0.70	0.72	0.70	25

FIGURE 3.17 – Rapport de classification de 20% test

La figure 3.18 présente le rapport de classification obtenu en utilisant une répartition de 70% des données pour l'entraînement et 30% pour les tests.

	precision	recall	f1-score	support
faible	0.69	0.92	0.79	24
élevé	0.60	0.23	0.33	13
accuracy			0.68	37
macro avg	0.64	0.57	0.56	37
weighted avg	0.66	0.68	0.63	37

FIGURE 3.18 – Rapport de classification de 30% test

Pour cela, nous avons d’abord divisé notre jeu de données en deux ensembles :

- 80% pour l’entraînement,
- 20% pour les tests.

Ce choix s’est avéré pertinent, car les scores de F1 (notamment en précision) obtenus avec cette répartition étaient meilleurs que ceux obtenus avec une division 70% entraînement et 30% test.

Ensuite, nous avons sélectionné quatre algorithmes de machine learning adaptés aux problèmes de classification :

TABLE 3.1 – Tableau comparatif des modèles

Modèle	Fonctionnement dans notre code	Pourquoi ce choix ?	Paramètres utilisés
K-Nearest Neighbors (KNN)	Calcule la distance entre l’adresse du client et celles des transitaires, puis recommande les 5 plus proches.	Adapté à la recommandation géographique, facile à interpréter.	n_neighbors=5, metric='euclidean'
Random Forest	Crée plusieurs arbres de décision et combine leurs prédictions pour une meilleure robustesse.	Performant, gère bien les données non linéaires et évite le surapprentissage.	n_estimators=100, max_depth=10, random_state=42
SVM (Support Vector Machine)	Tente de séparer les données à l’aide d’un hyperplan optimal dans un espace transformé.	Très bon pour des jeux de données à faible dimension et des marges claires entre classes.	kernel='rbf', C=1.0, gamma='scale'
Logistic Regression	Modèle de classification linéaire qui prédit la probabilité qu’un transitaire soit pertinent ou non (sortie binaire entre 0 et 1).	Simple, rapide à entraîner, interprétable, et sert souvent de modèle de base (baseline).	penalty='l2', C=1.0, solver='liblinear'

Pour chaque modèle, nous avons sauvégardé les hyperparamètres utilisés, et le processus d’entraînement a été orchestré à l’aide de scripts Python automatisés et suivis via MLflow.

Évaluation des modèles

Une fois que nous avons entraîné les modèles, nous les testons sur l’ensemble de test (X_{test}) afin d’évaluer leur capacité à généraliser. Pour chaque modèle, nous calculons plusieurs métriques de performance :

- l'accuracy : le taux global de bonnes prédictions
- la précision : le taux de bonnes prédictions parmi celles détectées comme positives
- le rappel : la capacité à identifier toutes les vraies positives

Ces métriques nous permettent de comprendre non seulement si le modèle fonctionne correctement dans l'ensemble, mais aussi s'il est bien équilibré en termes de détection et de fiabilité pour chaque classe, en particulier pour les cas où le Prix KM est élevé. Toutes les performances sont automatiquement enregistrées dans MLflow, ce qui nous permet d'assurer un suivi structuré et de comparer efficacement les différents modèles.

Sélection du meilleur modèle

Après avoir évalué les modèles, nous comparons les résultats de chacun d'eux. Le critère principal que nous utilisons pour faire notre choix est l'accuracy obtenue sur l'ensemble de test. Le modèle qui présente la meilleure accuracy est sélectionné comme le meilleur candidat pour passer en production. À ce stade, nous ne nous contentons pas d'identifier le modèle ; nous enregistrons également son identifiant de run dans MLflow. Cela nous permet de suivre précisément quelle version du modèle a été choisie, garantissant ainsi la reproductibilité et la traçabilité de notre processus de sélection.

Déploiement avec MLflow

Une fois que nous avons identifié le meilleur modèle, nous le déployons directement grâce à MLflow Model Registry. Dans un premier temps, nous enregistrons le modèle dans MLflow, puis nous marquons sa version comme « Production », ce qui signifie qu'il est prêt à être utilisé dans un environnement réel. Cette transition est effectuée via le client MLflow (MlflowClient), qui recherche la version correspondant au meilleur run. Une fois en production, le modèle peut être chargé dynamiquement pour réaliser des prédictions futures. Ce processus garantit que seule la version validée et la plus performante du modèle est utilisée en production, tout en permettant des mises à jour contrôlées et un audit rigoureux des modèles déployés.

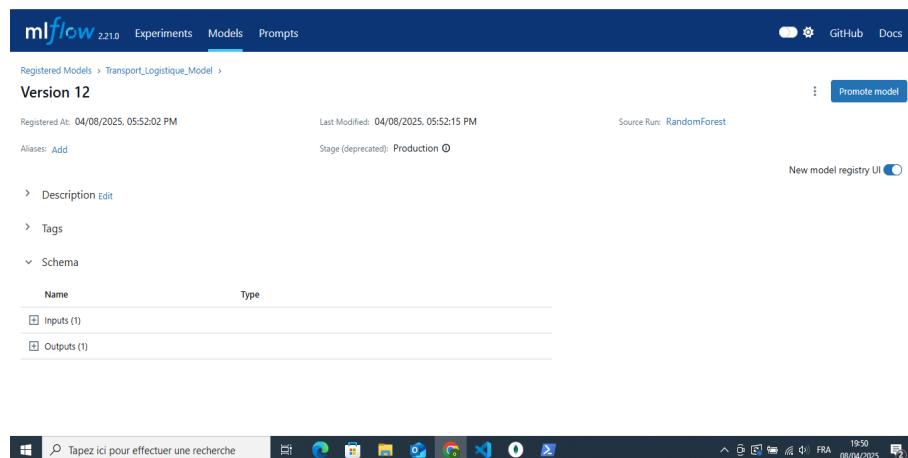


FIGURE 3.19 – Modèle en Production

3.4.5 Déploiement

Nous avons intégré et déployé MLflow dans notre système afin d'assurer le suivi, la gestion et le versionnement des expériences de machine learning. Cette solution nous permet de tracer les paramètres, les métriques, les artefacts des modèles, et de faciliter leur mise en production dans un environnement conteneurisé et orchestré.

3.4.5.1 Conteneurisation avec Docker

Afin de garantir un environnement isolé et reproductible, nous avons conteneurisé le service de suivi des expériences MLflow à l'aide de Docker. Pour cela, nous nous sommes appuyés sur l'image officielle python :3.9-slim. Nous y avons installé les dépendances système nécessaires, copié les fichiers de l'application, puis installé les dépendances Python définies dans le fichier requirements.txt.

Le fichier Dockerfile expose le port 5000 et lance le serveur MLflow avec les paramètres suivants :

- backend-store-uri : nous avons utilisé une base SQLite pour le suivi des métadonnées ;
- default-artifact-root : les artefacts générés sont stockés localement.

3.4.5.2 Orchestration avec Kubernetes

Nous assurons l'orchestration de notre système à l'aide de Kubernetes, ce qui nous permet de gérer automatiquement le cycle de vie des conteneurs. Pour cela, nous avons défini un fichier deployment.yaml qui décrit l'infrastructure suivante :

- Un Deployment Kubernetes pour le conteneur MLflow ;
- Un Service de type LoadBalancer afin d'exposer l'interface MLflow vers l'extérieur ;
- Deux PersistentVolumeClaim permettant d'assurer la persistance des données, à savoir :
 - Les artefacts des expériences (modèles, métriques, fichiers générés...);
 - La base de données SQLite utilisée pour le tracking des expériences.

Les volumes montés garantissent ainsi une sauvegarde durable des éléments essentiels au suivi et à la reproductibilité des expériences de machine learning.

3.4.5.3 Déploiement sur Google Cloud Platform (GCP)

Notre cluster Kubernetes est hébergé sur Google Kubernetes Engine (GKE). Nous avons construit l'image Docker de notre serveur MLflow, que nous avons ensuite poussée dans le Google Container Registry (GCR). Cette image est référencée dans notre fichier deployment.yaml via la directive suivante :

image : gcr.io/cargoweenapp/mlflow-server : v2

La figure 3.20 présente le résultat de la commande kubectl get pods, montrant que notre

podmlflow-server-6cd699689d-626k6

est en cours d'exécution (Running), avec un statut prêt (READY 1/1). Nous remarquons également qu'il a redémarré trois fois.

```
^oumaimadghaiess@mlflow-vm:~/CargoWeenApp/ML$ kubectl get pods
  NAME           READY   STATUS    RESTARTS   AGE
  mlflow-server-6cd699689d-626k6   1/1     Running   3 (2m ago)   7m39s
```

FIGURE 3.20 – MLFlow déployer

Cette approche nous permet de bénéficier d'une mise à l'échelle simplifiée, d'une surveillance centralisée et d'une gestion efficace des versions de nos services. Grâce à cette architecture, l'interface de suivi MLflow est accessible à l'adresse suivante :

`http://34.76.98.147/`

La figure 3.21 illustre le résultat de la commande `kubectl get svc mlflow-service`. On y observe que le service MLflow est exposé via un LoadBalancer. L'adresse IP externe qui lui a été attribuée est 34.76.98.147, ce qui nous permet d'accéder à l'interface de MLflow depuis l'extérieur.

```
oumaimadghaiess@mlflow-vm:~/CargoWeenApp/ML$ kubectl get svc mlflow-service
  NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
  mlflow-service   LoadBalancer   34.118.226.30   34.76.98.147   80:32084/TCP   91m
```

FIGURE 3.21 – Vérification de service

La figure 3.22 illustre l'accès à l'interface de MLflow via un navigateur Chrome. Le serveur MLflow y est déployé à l'adresse 34.76.98.147, ce qui confirme que le service est bien accessible depuis l'extérieur.

The screenshot shows a web browser window with the URL `http://34.76.98.147/#/experiments/0?searchFilter=&orderByKey=attributes.start_time&orderAsc=false&startTime=ALL&lifecycleFilter=Active&m...` . The page title is "mlflow 2.2.0 Experiments Models Prompts". The main content area is titled "Experiments" and shows a table header with columns: Run Name, Created, Dataset, Duration, Source, and Models. Below the table, there is a large message: "No runs logged" with a flask icon, followed by the text "No runs have been logged yet. Learn more about how to create ML model training runs in this experiment."

FIGURE 3.22 – MLFlow déployer

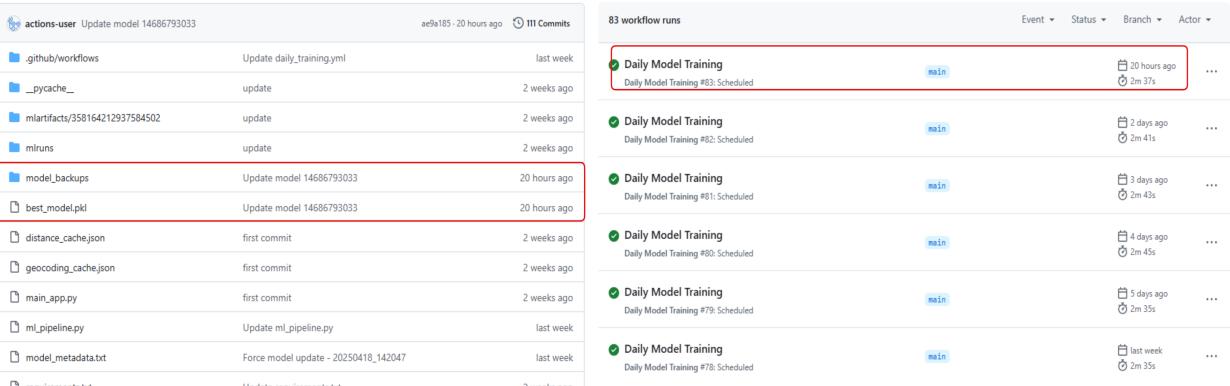
3.4.5.4 Automatisation CI/CD

Pour garantir que l'entraînement de notre modèle de machine learning soit automatisé, nous avons mis en place un workflow GitHub Actions qui se connecte à MLflow. Ce processus est programmé pour s'exécuter automatiquement chaque jour à minuit (UTC) grâce à une tâche cron, tout

en offrant également la possibilité d'un déclenchement manuel.

À chaque exécution, nous clonons le dépôt sur une machine virtuelle Ubuntu. Nous configurons ensuite l'environnement Python et installons toutes les dépendances nécessaires. Puis, nous lançons le serveur MLflow avec cette adresse "http://34.76.98.147/" pour gérer le suivi des expérimentations et assurer la traçabilité des modèles entraînés. Avant de redémarrer l'entraînement, nous nous assurons de supprimer toute ancienne version du modèle afin d'éviter les conflits. Le script d'entraînement (`train_model.py`) est alors exécuté, ce qui génère un nouveau modèle `best_model.pkl`. Nous vérifions et affichons les métadonnées de ce modèle pour un contrôle qualité. Une fois validé, nous versionnons et sauvegardons automatiquement le modèle dans le dépôt GitHub via un commit et un push automatisés comme indique la figure 3.23.

À la fin du processus, un email de notification est envoyé, via le protocole SMTP, pour informer les parties prenantes du succès ou de l'échec de l'exécution, avec un lien direct vers le workflow qui a été lancé comme indique la figure 3.24. Cette automatisation nous assure une régularité dans l'entraînement du modèle, tout en garantissant un suivi rigoureux et une historisation complète grâce à GitHub et MLflow.



The screenshot shows the GitHub Actions history for a repository. On the left, a list of commits is shown, with the last two being 'model_backups' and 'best_model.pkl', both of which have a red box around them. On the right, a list of '83 workflow runs' is displayed, with the first run, 'Daily Model Training #83: Scheduled', also having a red box around it. The table has columns for the workflow name, status, branch, and timestamp.

Workflow	Status	Branch	Timestamp
Daily Model Training	success	main	20 hours ago
Daily Model Training	success	main	2 days ago
Daily Model Training	success	main	3 days ago
Daily Model Training	success	main	4 days ago
Daily Model Training	success	main	5 days ago
Daily Model Training	success	main	last week

FIGURE 3.23 – Workflow github actions



FIGURE 3.24 – Email de notification

La figure 3.25 illustre l'interface du serveur MLflow, affichant les différentes expériences lancées automatiquement grâce à notre pipeline d'automatisation.

Run Name	Created	Dataset	Duration	Source
LogisticRegression	8 hours ago	-	3.5s	[train_mo...]
RandomForest	8 hours ago	-	3.7s	[train_mo...]
KNN	8 hours ago	-	4.2s	[train_mo...]
SVM	13 hours ago	-	3.3s	[train_mo...]
LogisticRegression	13 hours ago	-	3.3s	[train_mo...]
RandomForest	13 hours ago	-	3.5s	[train_mo...]
KNN	13 hours ago	-	3.9s	[train_mo...]
SVM	13 hours ago	-	3.5s	[train mo...]

FIGURE 3.25 – MLFlow déployer

3.4.6 Calcul des tarifs

Le processus débute par la saisie de l'adresse de livraison et l'extraction de l'aéroport d'arrivée à partir des documents fournis. Avec ces informations, nous calculons la distance routière entre l'aéroport et l'adresse de livraison en utilisant l'API OpenRouteService. Ensuite, pour chaque entreprise répertoriée dans la base de données, nous déterminons le tarif selon la formule suivante :

$$\text{Tarif total} = (\text{Distance de livraison} \times \text{Prix par km}) + \text{Prix d'arrivée} + \text{Frais de douane}$$

Cela nous permet d'obtenir un tarif total sur mesure pour chaque entreprise, en tenant compte de la distance réelle à parcourir.

3.4.7 Recommandation des meilleures sociétés

Une fois que nous avons calculé les tarifs, notre système sélectionne les entreprises les plus proches de l'aéroport d'arrivée en analysant à la fois les distances et les coûts. Pour affiner cette sélection, nous utilisons un modèle de machine learning qui prend en compte plusieurs facteurs : la distance, le prix par kilomètre, le coût d'arrivée, les frais de douane ainsi que le tarif total. Ce modèle attribue un score à chaque entreprise, ce qui nous permet de recommander en priorité celles qui offrent la meilleure proximité.

Comparaison des tarifs (triés par proximité à l'aéroport):						
Company Name	Distance to Airport (km)	Delivery Distance (km)	Prix KM	Prix arrivée	Douane	Tarif Total (Euro)
Novalis Transport Biologique	31.2621	20.5665	1.40	150.0	80.0	258.79
Cora2 LTM	31.2621	20.5665	1.40	150.0	80.0	258.79
MANUPORT Logistics	155.8189	20.5665	1.20	150.0	80.0	254.68
CENTRIMEX	724.4906	20.5665	1.30	150.0	80.0	256.74
CIACAM	724.4906	20.5665	0.96	150.0	80.0	249.74

FIGURE 3.26 – Recommandation des entreprises

La figure 3.26 illustre un exemple concret de comparaison des tarifs entre différentes sociétés de transport, triées par proximité à l'aéroport d'arrivée. Elle met en évidence les écarts de coûts totaux,

influencés par des facteurs tels que la distance, le prix au kilomètre et les frais fixes. Ces informations permettent d'appuyer les décisions du modèle en visualisant clairement les paramètres pris en compte pour la recommandation finale.

3.5 Conclusion

Ce chapitre a permis de présenter concrètement la mise en œuvre des deux principaux modules du projet : l'extraction intelligente d'informations et la recommandation des transitaires. Grâce à l'intégration de techniques d'intelligence artificielle et de géolocalisation, ces modules offrent une automatisation efficace et une prise de décision assistée. Cette implémentation constitue un socle fonctionnel solide qui sera exploité dans le développement de l'interface web, sujet du prochain chapitre.

Chapitre 4

Release 2 : Développement d'application web

4.1 Introduction

L'aboutissement de ce projet repose sur le développement d'une application web permettant aux utilisateurs d'interagir de manière simple et intuitive avec les fonctionnalités implémentées. Ce chapitre présente les différentes étapes du développement de l'interface, en mettant en lumière les choix techniques effectués, les outils et frameworks utilisés, ainsi que l'intégration des modules d'intelligence artificielle développés précédemment. L'objectif est de proposer une solution complète, ergonomique et performante, répondant aux besoins des utilisateurs finaux.

4.2 Architecture du projet

L'architecture repose sur une application web construite avec Next.js, qui joue un double rôle : à la fois frontend (interface utilisateur) et backend léger via ses API routes. L'utilisateur interagit directement avec cette application Next.js, qui traite les requêtes, gère la logique métier et communique avec une base de données MongoDB Atlas.

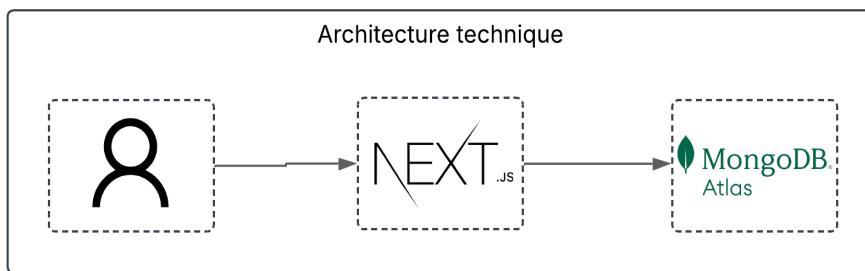


FIGURE 4.1 – Architecture technique

- Transitaire : accède à l'application web depuis un navigateur pour effectuer des actions comme la recherche, la réservation, ou la consultation de l'historique.
- Next.js : framework React qui permet :
 - le rendu des pages côté serveur.
 - la création d'API Routes (fichiers dans /src/app/api/) servant d'API backend sans avoir besoin d'un serveur Node.js séparé.

- la gestion de l'authentification, la validation des données, les appels à la base MongoDB.
- MongoDB Atlas : base de données NoSQL cloud hébergée sur une plateforme scalable. Elle stocke les données dynamiques de l'application comme les utilisateurs, les réservations, les sociétés de transit, les historiques, etc.

Cette architecture simplifie le déploiement et la maintenance en gardant l'ensemble du projet dans un monorepo Next.js, tout en assurant une scalabilité grâce à l'utilisation de MongoDB Atlas.

4.3 Planification de release 2

Dans notre application CargoWeen, le déroulement est le suivant : lorsqu'un transitaire non inscrit souhaite utiliser notre plateforme, il doit d'abord passer par un formulaire d'inscription, puis se connecter. Ensuite, il peut effectuer sa recherche en fournissant le reçu de réservation de vol ainsi que l'adresse de livraison où la marchandise doit être livrée. Notre plateforme affiche alors une liste, ainsi qu'une carte interactive, des sociétés transitoires déjà inscrites chez nous. Le transitaire choisit une société parmi cette liste, puis peut passer une réservation en ajoutant les documents nécessaires, comme la facture. Il procède ensuite au paiement en ligne grâce au module de paiement intégré "Konnect", ce qui facilite grandement le processus de réservation.

Lorsqu'une réservation est reçue par un autre transitaire, nous utilisons la technologie Socket pour assurer un système de notification en temps réel. Le transitaire destinataire peut alors modifier le statut de la réservation en "Livrée" une fois que la marchandise a été remise à l'adresse de livraison. Dans ce cas, un email de confirmation de livraison est automatiquement envoyé depuis notre plateforme à l'expéditeur. Pour cette fonctionnalité d'envoi d'emails, nous avons utilisé le protocole SMTP.

Pour cette release, notre travail a été réparti en deux sprints :

- **Sprint 1 : Système de recommandation :** comprend l'implémentation de la fonctionnalité d'authentification, la gestion de profil, ainsi que la consultation des sociétés recommandées (avec l'intégration de la partie Python dans l'application Next.js).
- **Sprint 2 : Système de réservation :** inclut la gestion des réservations, la mise en place d'un système de notifications, et l'intégration d'un module de paiement.

La figure 4.2 présente une capture illustrant le suivi d'avancement de la réalisation de cette deuxième release.

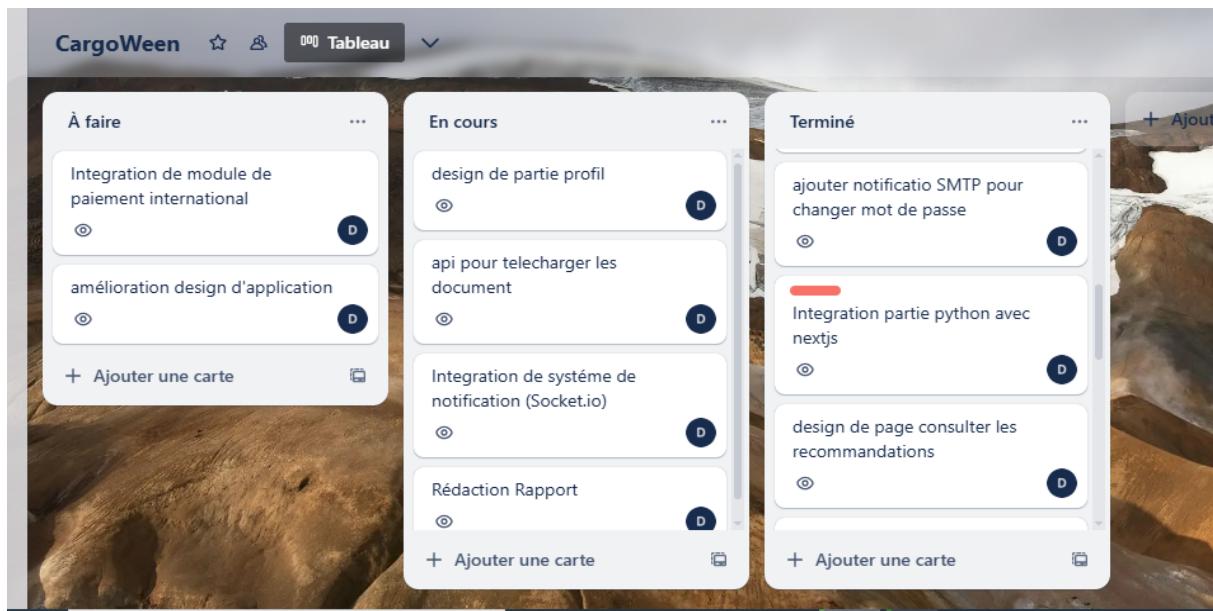


FIGURE 4.2 – Plannification Release 2

4.4 Sprint 1 : Système de recommandation

Le tableau 4.1 ci-dessous présente le backlog du premier sprint et décrit les tâches à réaliser et éventuellement la décomposition des tâches en sous-tâches.

TABLE 4.1 – Backlog de Sprint 1

Tâche	Sous tâche	Durée
Authentification	- Se connecter - s'inscrire	4 jours
Gérer son profil	- Consulter son profil - Modifier ses informations - Désactiver son compte - Changer son mot de passe	5 jours
Consulter les sociétés recommandées	- Consulter les sociétés recommandées par AI - Passer une réservation	15 jours

4.4.1 Raffinement de cas d'utilisation "Authentification"

Dans cette étape, nous illustrons le cas d'utilisation « Authentification », présenté dans la figure 4.3, suivi d'une description textuelle, d'un diagramme de séquence, ainsi que de leur prototype et de leur réalisation.

- **Diagramme de cas d'utilisation**

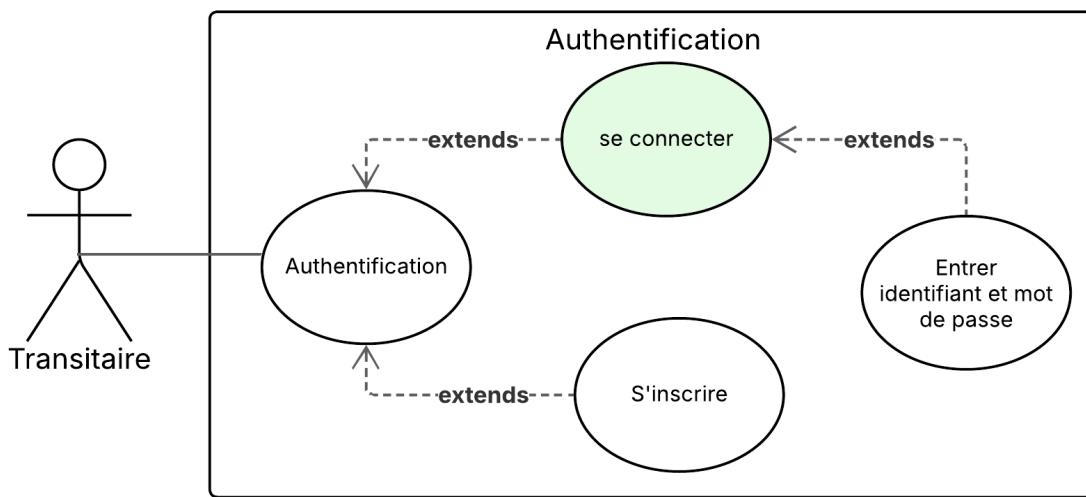


FIGURE 4.3 – Diagramme de cas d'utilisation «Authentification»

- **Description textuelle**

TABLE 4.2 – Description textuelle du cas d'utilisation « Authentification »

Cas d'utilisation	Authentification
Acteur principal	Transitaire
Préconditions	L'utilisateur accède à la page de connexion.
Postconditions	L'utilisateur est connecté à son espace sécurisé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le transitaire accède à l'interface de connexion. 2. Il saisit son identifiant et son mot de passe. 3. Le système vérifie les identifiants. 4. Les identifiants sont corrects. 5. L'utilisateur est connecté avec succès.
Scénario alternatif	<ol style="list-style-type: none"> 3a. Les identifiants sont invalides. → Message : « Identifiant ou mot de passe incorrect. »

- **Diagramme de séquence**

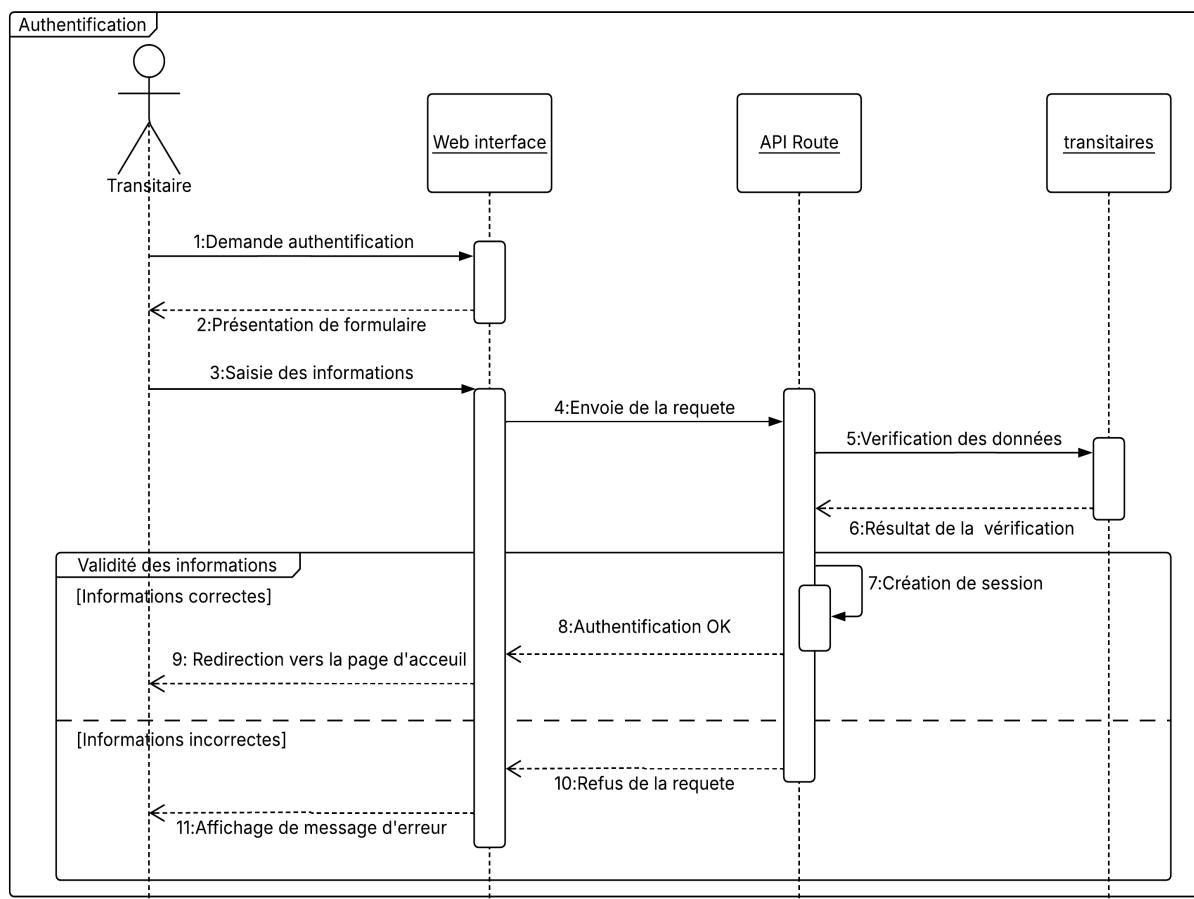


FIGURE 4.4 – Diagramme de sequence d'authentification

• Prototype

La figure 4.5 illustre le prototype de notre future interface permettant à chaque transitaire de s'inscrire sur notre plateforme.

The prototype shows a registration form for 'Inscription Transitaire' (CargoWeen). The form includes fields for company name, postal address, telephone, fax, city, postal code, website, password, and password confirmation. A 'Créer mon compte' button is at the bottom.

FIGURE 4.5 – Prototype d'interface d'inscription

La figure 4.6 illustre le prototype de notre future interface permettant à chaque transitaire d'accéder à notre plateforme.

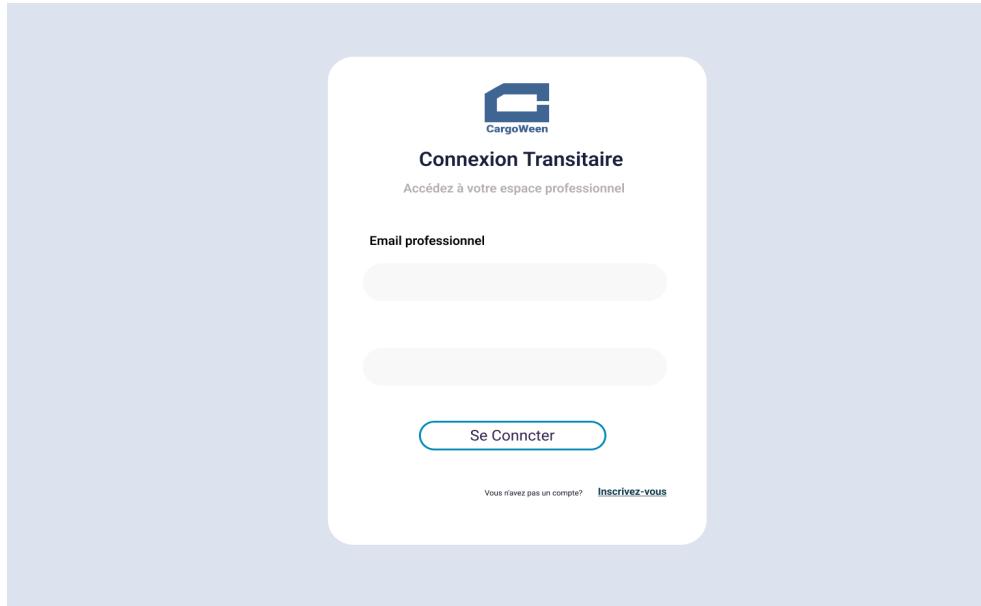


FIGURE 4.6 – Prototype d'interface d'authentification

- **Réalisation**

La figure 4.7 présente l'interface d'inscription, où un transitaire non encore inscrit peut créer son compte en remplissant les champs requis.



Inscription Transitaire

Créez votre compte professionnel

Informations de l'entreprise		Contacts et Tarifs	
Nom de l'entreprise *	Votre entreprise	Email *	email@entreprise.com
Adresse postale *	Adresse complète	Téléphone *	+33 1 23 45 67 89
Ville *	Code postal *	Fax	Fax (optionnel)
Région *	Pays *	Site web	https://www.example.com
Code postal	Pays	Personne responsable	Nom du responsable
Prix/KM (€) *	Prix arrivée (€) *	Douane (€) *	0.00 € 0.00 € 0.00 €
Sécurité			
Mot de passe *	Confirmer mot de passe *		
*****	*****		
Minimum 8 caractères			
Créer mon compte			
Déjà inscrit ? Connectez-vous			

FIGURE 4.7 – Interface d'inscription

La figure 4.8 présente l'interface de connexion où le transitaire saisit son email et son mot de passe pour accéder à l'application. En cas d'erreur sur l'email ou le mot de passe, un message d'erreur est affiché.



Connexion Transitaire

Accédez à votre espace professionnel

Email professionnel	votre@email.com
Mot de passe	*****
Mot de passe oublié ?	
Se connecter	
Pas encore de compte ? Créer un compte	

FIGURE 4.8 – Interface d'authentification

4.4.2 Raffinement de cas d'utilisation «Gérer profil»

Dans cette étape, nous illustrons le cas d'utilisation « Gérer profil », présenté dans la figure 4.9, suivi d'une description textuelle, d'un diagramme de séquence, ainsi que de leur prototype et de leur réalisation.

- **Diagramme de cas d'utilisation**

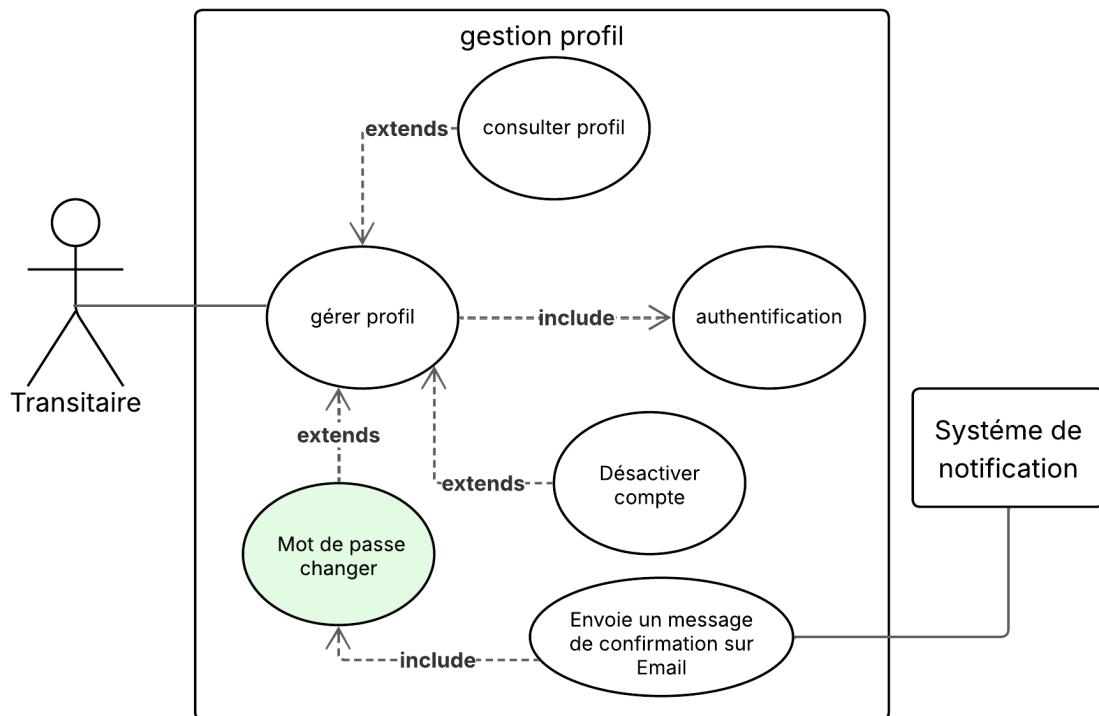


FIGURE 4.9 – Diagramme de cas d'utilisation «Gestion profil»

- **Description textuelle**

TABLE 4.3 – Description textuelle du cas d'utilisation « Changer mot de passe »

Cas d'utilisation	Changer mot de passe
Acteur principal	Transitaire
Parties prenantes	Système de notification
Préconditions	Le transitaire est authentifié dans l'application.
Postconditions	Le mot de passe est mis à jour et un Email de confirmation est envoyé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le transitaire accède à l'application. 2. Il se connecte avec ses identifiants. 3. Il accède à la section « Gérer profil ». 4. Il choisit l'option « Changer mot de passe ». 5. Le système demande l'ancien mot de passe. 6. Il saisit l'ancien mot de passe et le nouveau mot de passe. 7. Le système vérifie l'ancien mot de passe. 8. Le mot de passe est mis à jour. 9. Un message de succès s'affiche. 10. Le système de notification envoie un email de confirmation.
Scénario alternatif 1	<p>5a. Le mot de passe actuel est incorrect. → Un message d'erreur s'affiche : « Mot de passe actuel incorrect. » → Le transitaire peut réessayer ou annuler l'opération.</p>
Scénario alternatif 2	<p>10a. L'Email ne peut être envoyé (erreur réseau, Email invalide). → Le système affiche : « Mot de passe modifié, mais la confirmation par Email a échoué. »</p>

- **Diagramme de séquence**

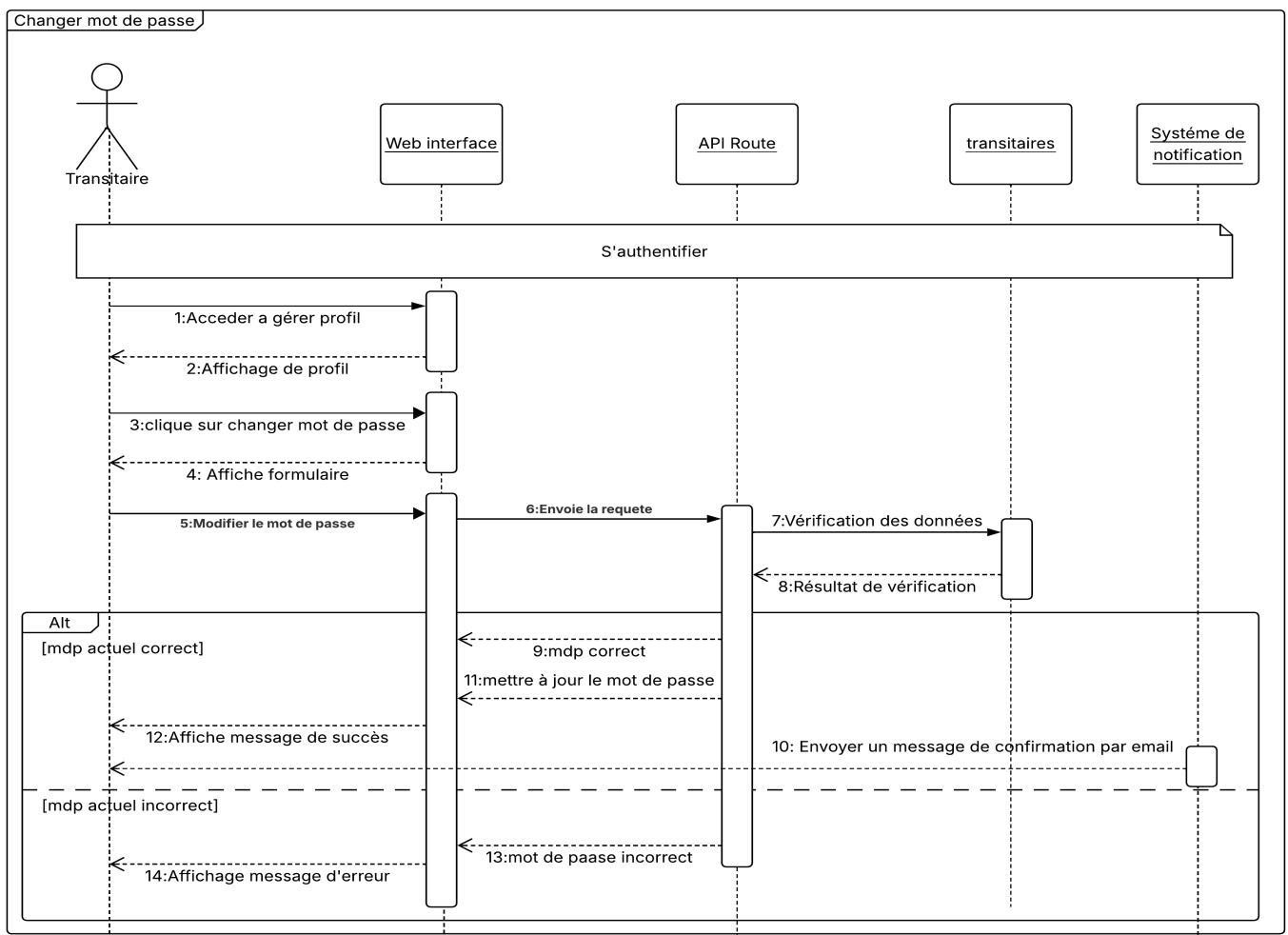


FIGURE 4.10 – Diagramme de séquence de Changer mot de passe

• Prototype

La figure 4.11 illustre le prototype de notre future interface permettant à chaque transitaire de consulter son profil, de modifier ses informations, de désactiver son compte ou de changer son mot de passe.

Information d'entreprise

Nom d'entreprise	Téléphone
ITBS	11111111
Email	Fax
itbs@gmail.com	11111111

Adresse

Adresse postale	Ville
Nabeul, Tunis	nabeul
Région	Pays
nabeul	Tunisie

Tarifs

Prix par km (€)	Douane (€)	Prix arrivée (€)
1.4	80	150

FIGURE 4.11 – Prototype d’interface de consulter profil

La figure 4.12 illustre le prototype de notre future interface permettant à chaque transitaire de changer son mot de passe.

Changer le mot de passe

Mot de passe actuel

Nouveau mot de passe

Confirmer le nouveau mot de passe

Annuler enregistrer

FIGURE 4.12 – Prototype d’interface pour changer le mot de passe

• Réalisation

La figure 4.13 présente l’interface où le transitaire peut consulter, modifier ses informations, désactiver son compte et changer son mot de passe.

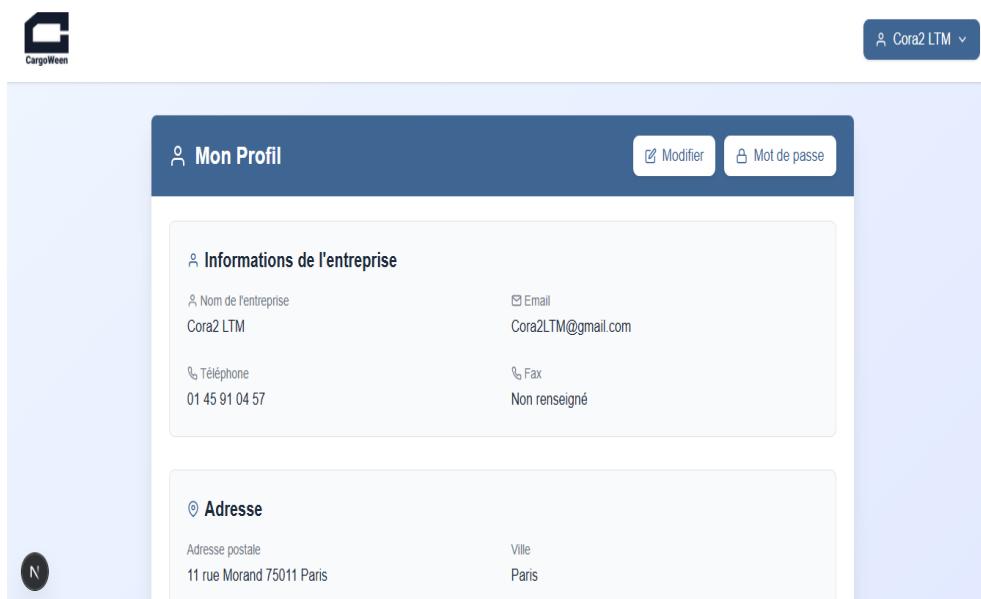


FIGURE 4.13 – Interface de profil

La figure 4.14 présente l'interface permettant au transitaire de changer son mot de passe.

The screenshot shows the 'Changer le mot de passe' (Change Password) form. It has three input fields: 'Mot de passe actuel' (Current password), 'Nouveau mot de passe' (New password), and 'Confirmer le nouveau mot de passe' (Confirm new password). At the bottom are two buttons: 'Annuler' (Cancel) and 'Enregistrer' (Register).

FIGURE 4.14 – Interface pour changer le mot de passe

La figure 4.15 présente l'email de confirmation envoyé après que le transitaire a terminé la modification de son mot de passe.



FIGURE 4.15 – Email de confirmation

4.4.3 Raffinement de cas d'utilisation "Consulter toutes les sociétés recommandées"

Dans cette étape, nous illustrons le cas d'utilisation « Consulter toutes les sociétés recommandées », présenté dans la figure 4.16, suivi d'une description textuelle, d'un diagramme de séquence, ainsi que de leur prototype et de leur réalisation.

- **Diagramme de cas d'utilisation**

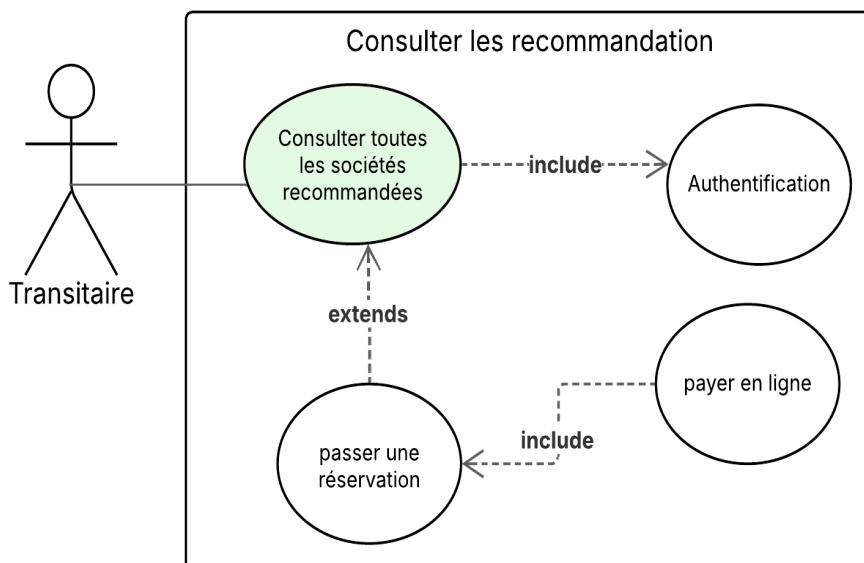


FIGURE 4.16 – Diagramme de cas d'utilisation «Consulter les sociétés recommandées»

- **Description textuelle**

TABLE 4.4 – Description textuelle du cas d'utilisation « Consulter les sociétés recommandées »

Cas d'utilisation	Consulter toutes les sociétés recommandées
Acteur principal	Transitaire
Préconditions	Le transitaire est authentifié dans l'application.
Postconditions	Les sociétés recommandées affichées
Scénario nominal	<ol style="list-style-type: none"> 1. Le transitaire accède à l'application. 2. Il se connecte avec ses identifiants. 3. Il accède à la section « Recherche ». 4. Il charge un reçu de réservation en PDF 5. Il Saisir l'adresse de livraison. 5. Il clique sur le bouton «Rechercher». 6. Le système demande les sociétés. 7. Le système affiche les tous les recommandations.
Scénario alternatif	<p>5a. L'adresse de livraison introuvable. → Un message d'erreur s'affiche : «Adresse de livraison introuvable. » l'opération.</p>

- **Diagramme de séquence**

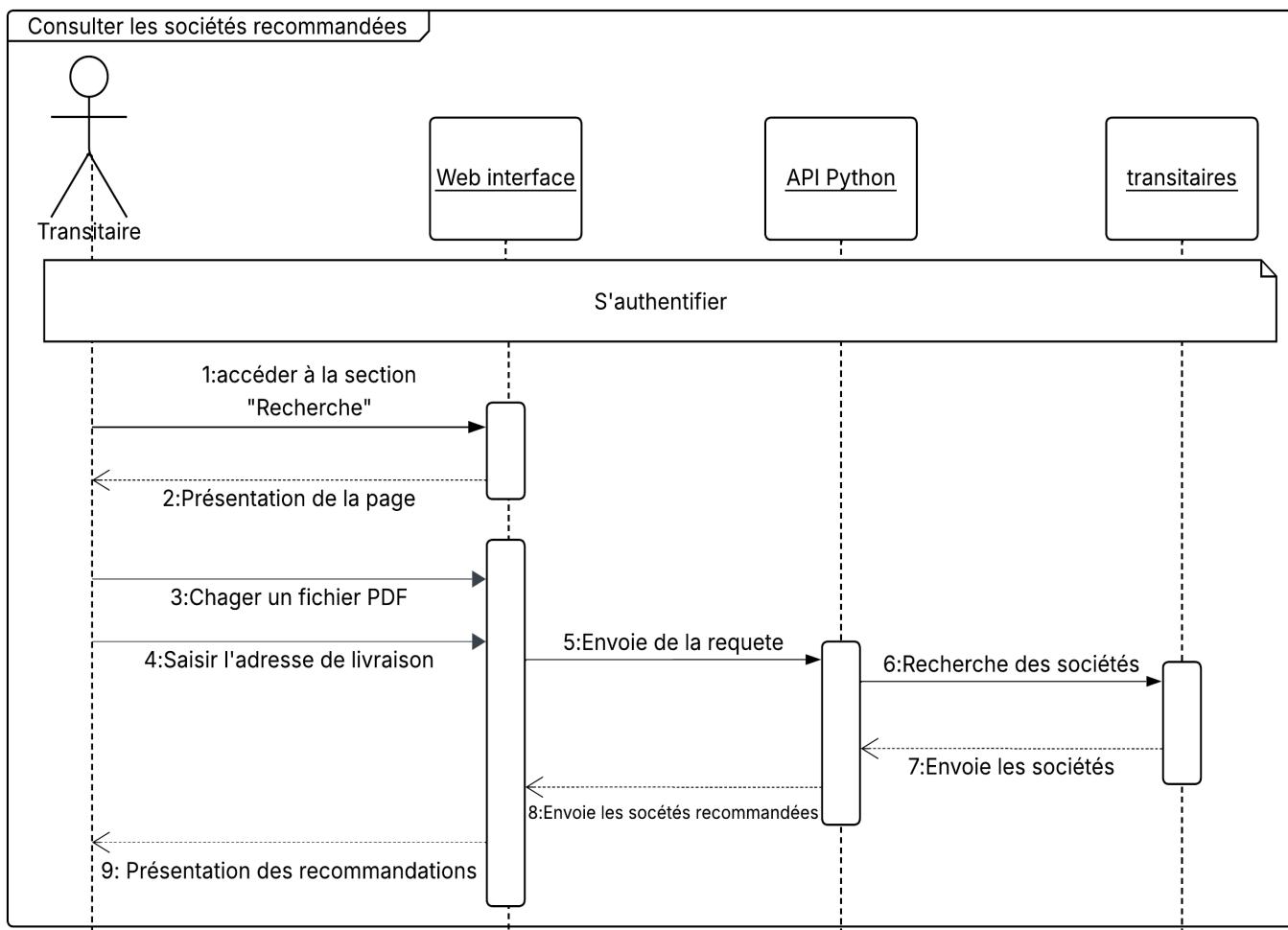


FIGURE 4.17 – Diagramme de séquence de Consulter les sociétés recommandées

• Prototype

La figure 4.18 présente le prototype de notre future interface de recherche, où le transitaire peut consulter les sociétés recommandées.

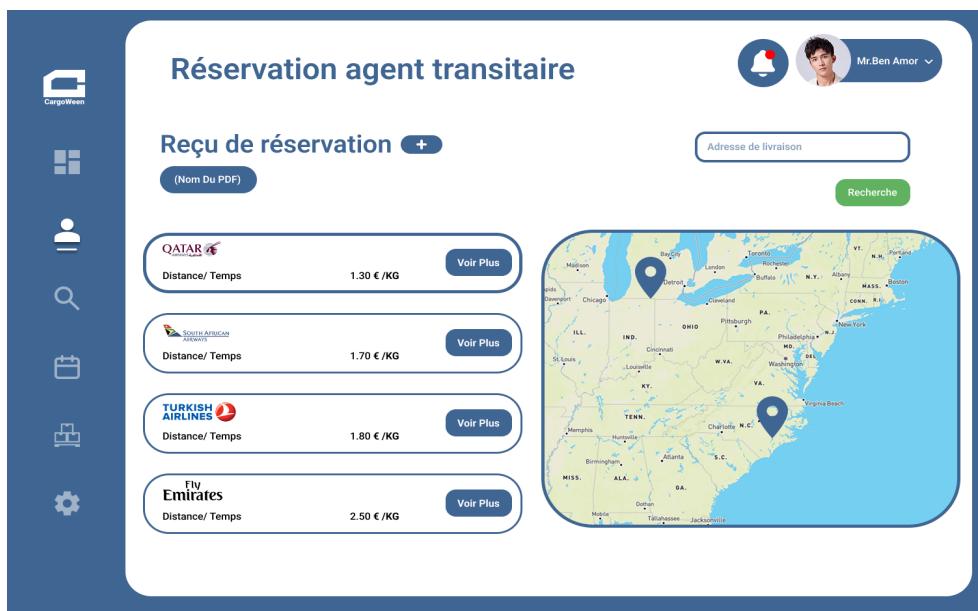


FIGURE 4.18 – Prototype d'interface de recherche

La figure 4.19 présente le prototype de notre future interface détail de réservation, où le transitaire peut passer une réservation.



FIGURE 4.19 – Prototype d'interface détail de réservation

• Réalisation

La figure 4.20 présente l'interface correspondant à la première étape de recherche, où le transitaire peut parcourir un fichier PDF — représentant le reçu de réservation de vol — saisir l'adresse de livraison, puis cliquer sur le bouton de recherche.

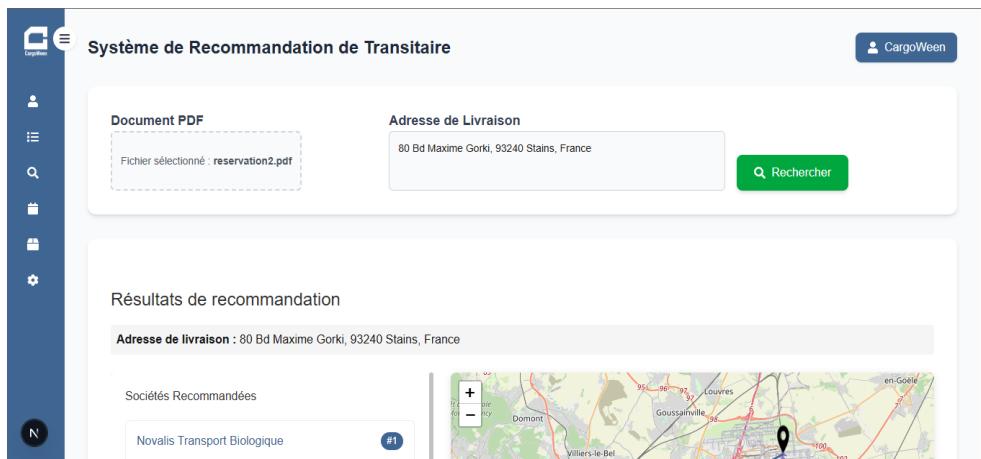


FIGURE 4.20 – Partie 1 d'interface de Recherche

La figure 4.21 présente l'interface correspondant à la deuxième étape de recherche, où le système affiche une liste des sociétés transitaires indiquant pour chacune leur nom, adresse, tarif et la distance les séparant de l'aéroport d'arrivée. Sur la carte, les adresses sont représentées par différents points :

- **le point noir** : représente l'aéroport d'arrivée.
- **le point vert** : correspond à l'adresse de livraison de la marchandise.
- **les points rouges** : indiquent les emplacements des sociétés transitaires.

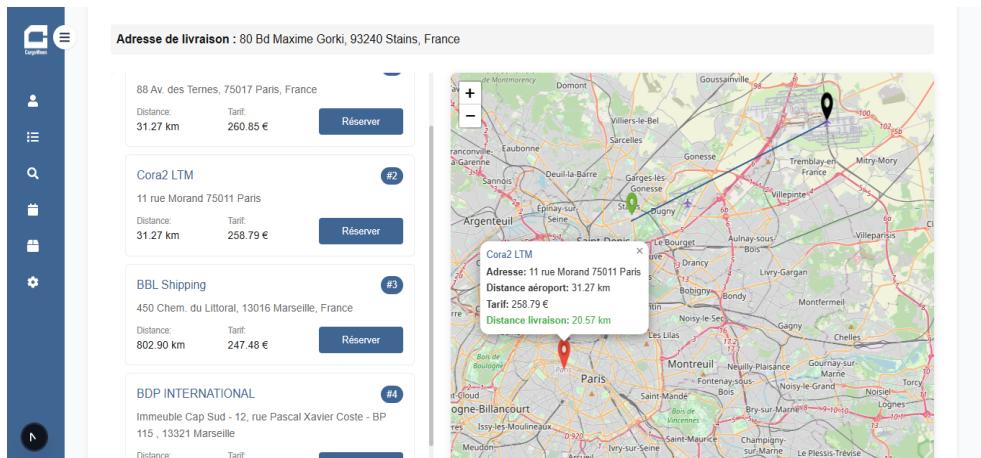


FIGURE 4.21 – Partie 2 d'interface de Recherche

Figure 4.22 présente la première partie de l'interface "Détail de réservation". Cette section affiche les informations de chaque société de transit, telles que l'email, le numéro de téléphone, l'adresse, ainsi que leur position géographique visualisée sur une carte interactive.

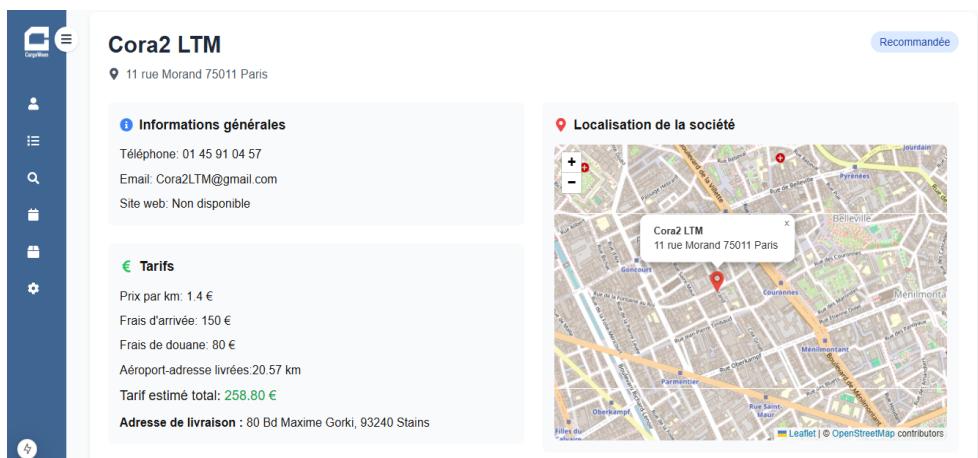


FIGURE 4.22 – Partie 1 d'interface détail de réservation

Figure 4.23 présente la deuxième partie de l'interface "Détail de réservation". Cette section permet au transitaire de fournir les documents nécessaires, tels que les factures ou les bons de livraison. Une fois les documents déposés, un clic sur le bouton dédié permet de passer à l'étape de paiement.

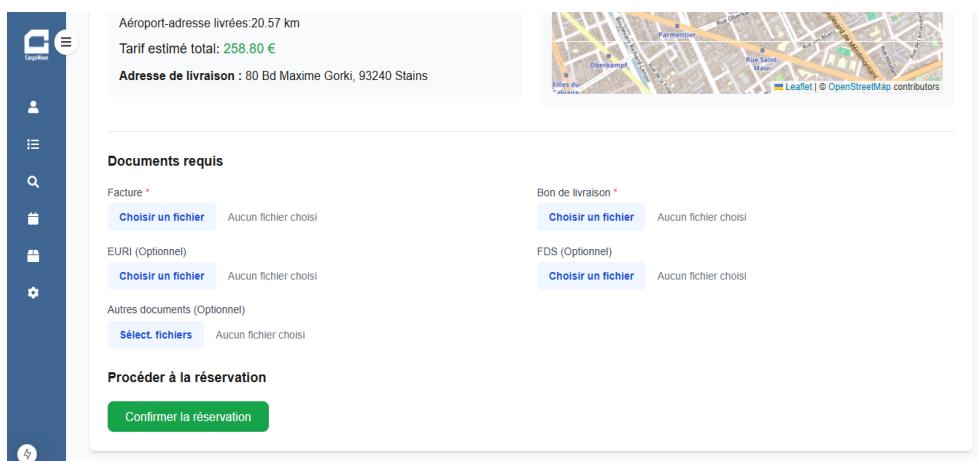


FIGURE 4.23 – Partie 2 d'interface détail de réservation

4.5 Sprint 2 : Système de réservation

Le tableau 4.5 ci-dessous présente le backlog du premier sprint et décrit les tâches à réaliser et éventuellement la décomposition des tâches en sous-tâches.

TABLE 4.5 – Backlog de Sprint 2

Tâche	Sous tâche	Durée
Paiement en ligne	- Paiement d'une réservation passée	4 jours
Gérer ses réservations	- Consulter ses réservations - Modifier statut de réservation - Consulter ses réservations reçues - Télécharger ses documents et factures	12 jours
Consulter ses notifications	- Consulter ses notifications	3 jours

4.5.1 Raffinement de cas d'utilisation "Gestion des réservations"

Dans cette étape, nous illustrons le cas d'utilisation « Gestion des réservations », présenté dans la figure 4.24, suivi d'une description textuelle, d'un diagramme de séquence, ainsi que de leur prototype et de leur réalisation.

- **Diagramme de cas d'utilisation**

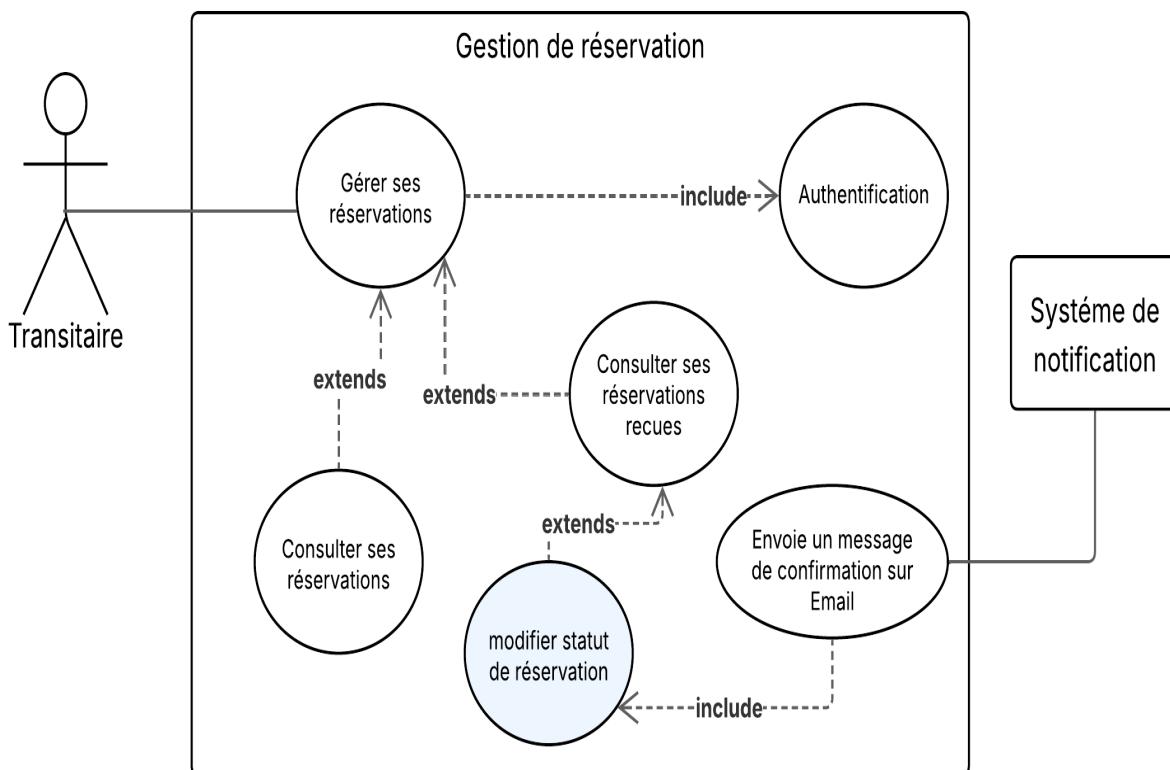


FIGURE 4.24 – Diagramme de cas d'utilisation «Gestion des réservations»

- **Description textuelle**

TABLE 4.6 – Description textuelle du cas d'utilisation « Modifier statut de réservation »

Cas d'utilisation	Modifier statut de réservation
Acteur	Transitaire
Pré-condition	Le transitaire doit être connecté à l'application et avoir au moins une réservation reçue
Post-condition	Le statut de la réservation est mis à jour et un Email de confirmation est envoyé.
Scénario nominal	<ol style="list-style-type: none"> 1. Le transitaire accède à l'application. 2. Il se connecte avec ses identifiants. 3. Il accède à la section « Mes réservations reçues ». 4. Il sélectionne une réservation. 5. Il modifie le statut de la réservation « Livrée ». 6. Le système enregistre la modification. 7. Un message de confirmation s'affiche. 8. Le système de notification envoie un email de confirmation.
Scénario alternatif 1	<ol style="list-style-type: none"> 5a. Il tente de modifier le statut de la réservation. 6a. Le système détecte une erreur 7a. Un message d'erreur s'affiche. → Message : «Connexion réseau perdue»
Scénario alternatif 2	<ol style="list-style-type: none"> 8a. L'Email ne peut être envoyé (erreur réseau, Email invalide). → Le système affiche : « Mot de passe modifié, mais la confirmation par Email a échoué. »

- **Diagramme de séquence**

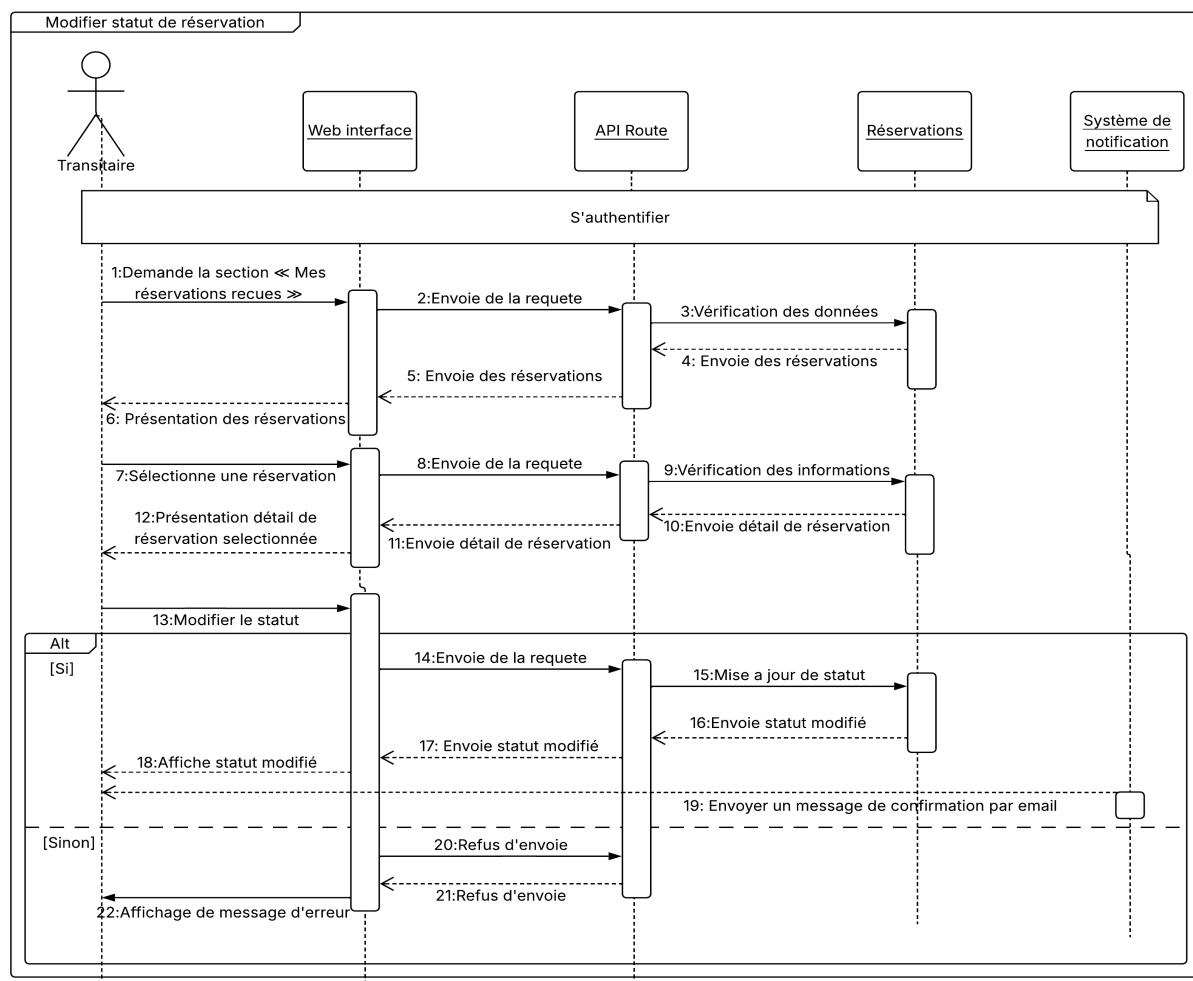


FIGURE 4.25 – Diagramme de séquence de Modification statut de réservation

- **Prototype** La figure 4.26 présente un prototype de notre future interface pour la gestion des réservations reçues.



FIGURE 4.26 – Prototype d'interface de réservation reçue

La figure 4.27 présente un prototype de notre future interface pour la détail de réservation reçue.

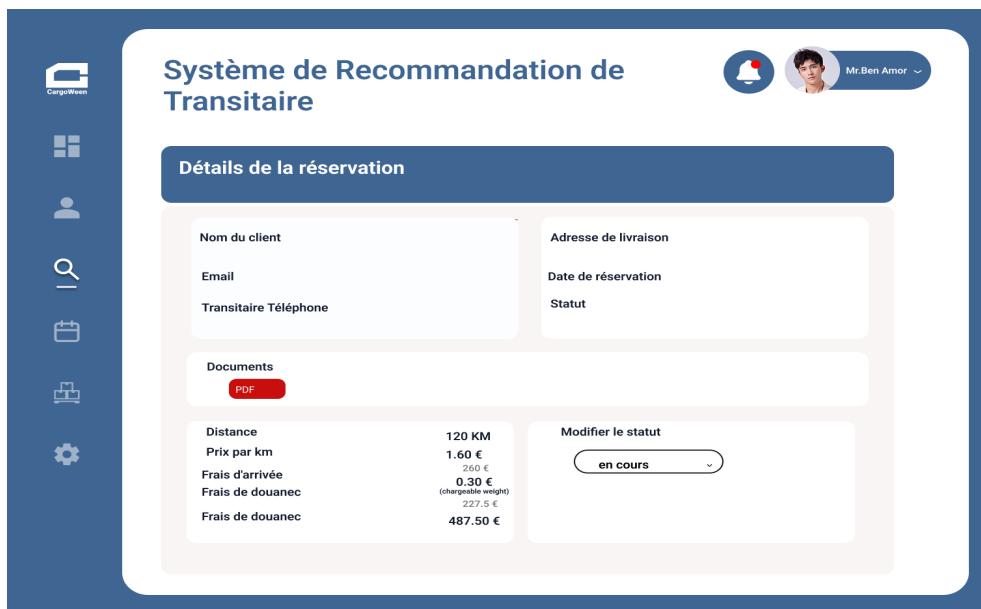


FIGURE 4.27 – Prototype d'interface détail de réservation reçue

- **Réalisation** La figure 4.28 présente l'interface de la liste des réservations reçues, où sont affichés le nom des clients, l'adresse de livraison de leur marchandise, la date de leur réservation ainsi que le statut de chaque réservation, qu'il soit en cours ou livré.

The screenshot shows a web application interface titled "Système de Recommandation de Transitaire". On the left is a vertical sidebar with icons for user management, reports, search, and settings. The main area has a header "Réservations reçues" and a sub-header "Gérez les demandes de réservation de vos clients". Below this is a table with columns: CLIENT, ADRESSE DE LIVRAISON, DATE, STATUT, and DÉTAILS. Two rows are visible:

CLIENT	ADRESSE DE LIVRAISON	DATE	STATUT	DÉTAILS
Novalis Transport Biologique	80 Bd Maxime Gorki, 93240 Stains, France	dim. 11 mai 2025, 12:31	livré	Voir
CargoWeen	80 Bd Maxime Gorki, 93240 Stains, France	mer. 21 mai 2025, 15:13	en cours	Voir

FIGURE 4.28 – Interface de réservation reçue

La figure 4.29 présente une interface détaillée des réservations reçues, permettant au transitaire de modifier le statut de la réservation directement depuis cette interface.

The screenshot shows a detailed view of a reservation. The top bar says "Détails de la réservation" and "Informations complètes sur la réservation". It includes "Retour" and "Imprimer" buttons. The interface is divided into sections:

- Informations client:** Nom du client (CargoWeen), Email (dghaisoumaia0@gmail.com), Transitaire Téléphone (Non spécifié).
- Détails de livraison:** Adresse de livraison (80 Bd Maxime Gorki, 93240 Stains, France), Date de réservation (mercredi 21 mai 2025 à 15:13), Statut (en cours).
- Documents:** Two PDF files: "Doc1.pdf" (Facture) and "111.pdf" (Bon de livraison).
- Détails tarifaires:** Distance (20.567 km), Prix par km (1.4 €/km), Frais d'arrivée (150 €), Frais de douane (80 €). Total: 258.79 €.
- Actions:** A dropdown menu with "Modifier le statut" and "En cours".

FIGURE 4.29 – Interface détail de réservation reçue

La figure 4.30 montre l'email de confirmation de réservation, dont le statut a déjà été modifié en « Livrée ». Cet email indique également le numéro de réservation, la date de réservation ainsi que l'adresse de livraison de la marchandise.

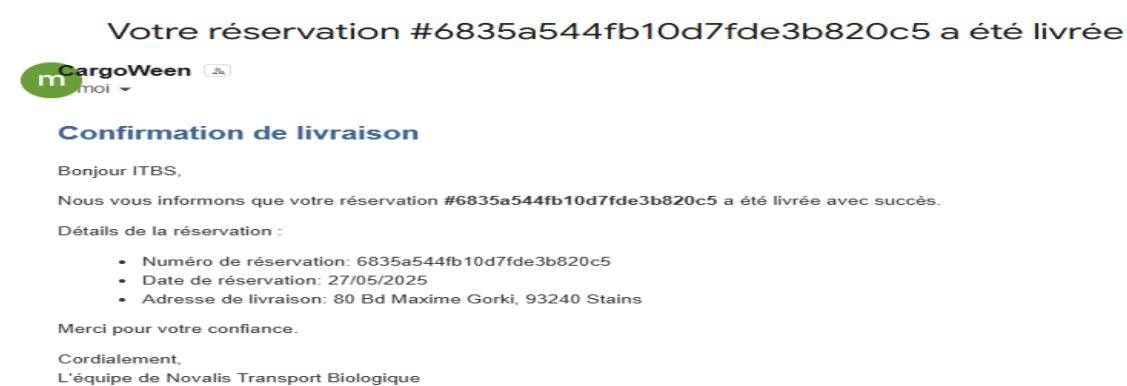


FIGURE 4.30 – Email de confirmation

4.5.2 Raffinement de cas d'utilisation "Paiement en ligne"

Dans cette étape, nous illustrons le cas d'utilisation « paiement en ligne », présenté dans la figure 4.31, suivi d'une description textuelle, d'un diagramme de séquence, ainsi que de leur prototype et de leur réalisation.

- **Diagramme de cas d'utilisation**

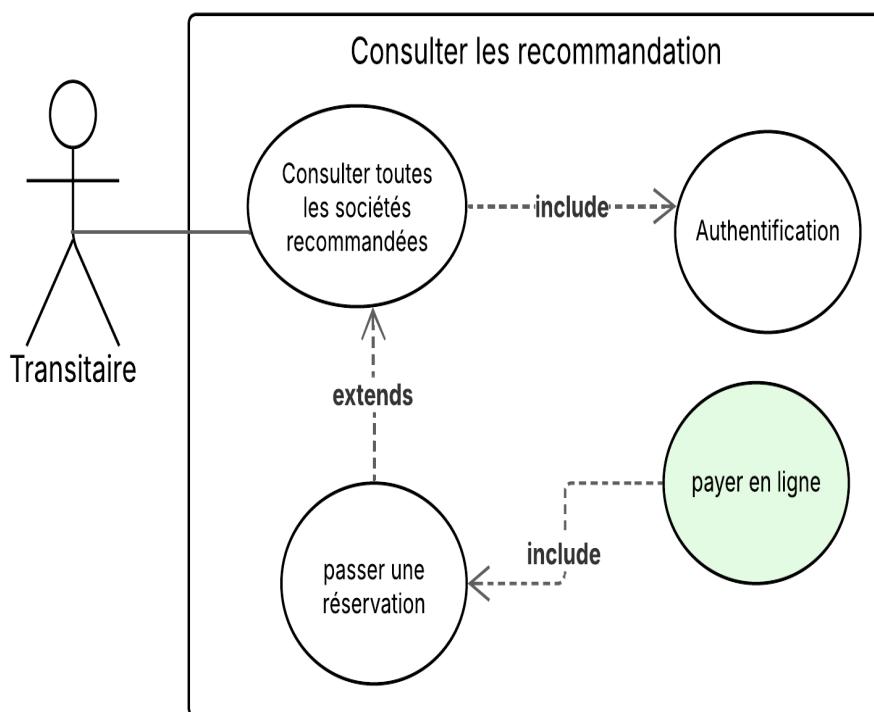


FIGURE 4.31 – Diagramme de cas d'utilisation «Paiement en ligne»

- **Description textuelle**

TABLE 4.7 – Description textuelle du cas d'utilisation « Payer en ligne »

Cas d'utilisation	Payer en ligne
Acteur	Transitaire
Pré-condition	Le transitaire doit avoir validé une réservation et être connecté à l'application
Post-condition	Le paiement est validé et la réservation est confirmée
Scénario nominal	<ol style="list-style-type: none"> 1. Le transitaire accède à l'application. 2. Il se connecte avec ses identifiants. 3. Il accède à la page de paiement. 4. Il sélectionne un mode de paiement 5. Il saisit les informations de paiement requises. 6. Il confirme le paiement. 7. Le système vérifie et valide le paiement. 8. Un message de confirmation s'affiche.
Scénario alternatif	<p>4a. Le paiement échoue (fonds insuffisants, erreur de carte).</p> <p>5a. Un message d'erreur s'affiche → Message : « Échec du paiement. Veuillez réessayer. »</p> <p>6a. L'utilisateur peut corriger les données ou choisir un autre moyen de paiement.</p>

• Diagramme de séquence

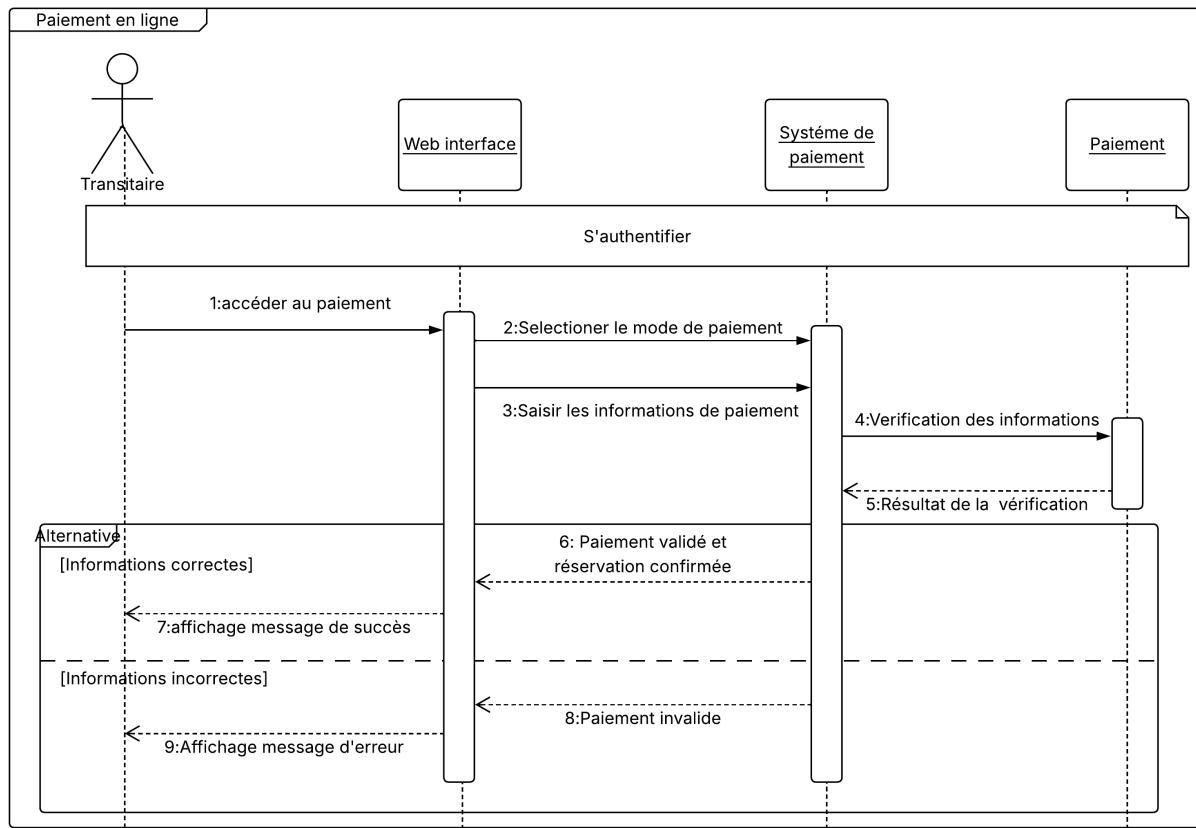


FIGURE 4.32 – Diagramme de sequence de Paiement en ligne

- Prototype
- Réalisation

4.6 Intégration du module CargoWeen dans l'application principale

Le module que nous avons créé s'intègre parfaitement à l'application principale CargoWeen, qui a pour but de gérer l'ensemble du processus de réservation de vols de fret : recherche, comparaison, réservation et paiement.

Une fois qu'un utilisateur a finalisé sa réservation et son paiement via l'application, il est automatiquement dirigé vers le module de gestion des transitaires, que nous avons spécialement développé. Ce module permet à l'utilisateur de choisir une société de transit pour livrer sa marchandise. Il inclut le téléchargement du reçu de réservation, la saisie de l'adresse de livraison, l'affichage des transitaires disponibles sur une carte interactive, la réservation, l'ajout de documents justificatifs, ainsi que le paiement via une passerelle intégrée.

Cette intégration garantit la continuité du processus logistique et améliore l'expérience utilisateur en centralisant toutes les démarches sur une seule plateforme.

4.7 Choix technique

La réalisation de notre application nécessite de faire un bon choix de technologies. Cette partie sera consacrée à la présentation de l'environnement de développement logiciel de l'application ainsi qu'aux langages de programmation utilisés.

4.7.1 Environnement logiciel

- **Github** : est une plateforme cloud qui vous permet de stocker, partager et collaborer avec d'autres pour écrire du code [4].



FIGURE 4.33 – GitHub

- **MongoDB** : est une base de données orientée documents. En d'autres termes, vous avez la scalabilité et la flexibilité dont vous avez besoin, tout en bénéficiant des fonctionnalités d'interrogation et d'indexation essentielles [23].



FIGURE 4.34 – MongoDB

- **Visual Studio Code :** est un éditeur de code source autonome qui fonctionne sur Windows, macOS et Linux. C'est le choix idéal pour les développeurs JavaScript et web, avec des extensions qui supportent presque tous les langages de programmation [28].



FIGURE 4.35 – Visual Studio Code

- **Figma :** est une plateforme en ligne gratuite qui permet aux utilisateurs de collaborer pour créer les meilleurs sites Web au monde. Que ce soit pour le wireframe, le design, le prototype ou le développement, Figma rend le web design beaucoup plus accessible et agréable [17].

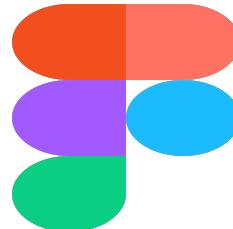


FIGURE 4.36 – Figma

- **Trello :** l'outil idéal pour faire avancer votre travail. Que vous gériez vos tâches quotidiennes, que vous fixiez des objectifs personnels ou que vous organisiez votre vie, Trello vous aide à vous concentrer sur l'essentiel et à toujours rester au top de votre productivité. [27].



FIGURE 4.37 – Trello

- **Postman :** est une plateforme tout-en-un pour gérer les API, facilitant leur création et leur utilisation. Elle rend chaque étape du cycle de vie des API plus simple, que ce soit la conception, les tests, la livraison ou la surveillance. Pensée pour les équipes, Postman améliore la collaboration, l'organisation et permet un développement rapide d'API à la fois sécurisées et fiables [29].



FIGURE 4.38 – Postman

- **Lucidchart :** Créez des diagrammes de nouvelle génération grâce à l'IA, aux données et à l'automatisation dans Lucidchart. Apprenez à maîtriser et à optimiser tous vos systèmes et processus [12].



FIGURE 4.39 – Lucidchart

- **Latex** : est un outil de composition de documents professionnels. Il peut servir à écrire des textes pour écrire des livres [11].

LATEX

FIGURE 4.40 – LaTex

4.7.2 Moyen de programmation

- **Langage Python** : Python est un langage de programmation à la fois puissant et accessible. Il offre des structures de données de haut niveau et permet d'adopter une approche simple mais efficace de la programmation orientée objet [10].



FIGURE 4.41 – Python

- **Framework NextJs** : Next.js est un framework gratuit et open source s'appuyant sur la bibliothèque JavaScript React et sur la technologie Node.js [16].



FIGURE 4.42 – NextJs

4.8 Conclusion

Ce dernier chapitre a permis de démontrer comment les modules techniques du projet ont été intégrés dans une application web fonctionnelle et accessible. Le choix des outils et technologies a contribué à garantir une interface fluide, tout en assurant la cohérence avec les traitements en backend. Cette application concrétise l'ensemble des travaux réalisés durant ce projet de fin d'études, en offrant une solution innovante et opérationnelle pour l'optimisation du processus de réservation dans le domaine du transport aérien.

Conclusion générale

Ce Projet de Fin d'Études a été l'occasion de mettre en pratique les compétences techniques et méthodologiques acquises tout au long de la formation, au service d'une problématique concrète liée à l'optimisation des services de transport aérien. En partant d'une analyse approfondie du besoin et de l'environnement de travail, le projet a conduit à la conception et au développement d'une solution intelligente intégrant des technologies d'intelligence artificielle, de géolocalisation et de recommandation.

La mise en œuvre d'un système d'extraction d'informations à partir de documents, couplée à un moteur de recommandation basé sur la localisation, a permis de proposer une approche innovante et automatisée du processus de recherche et de sélection de transitaires. Le développement de l'application web a ensuite permis de rendre ces fonctionnalités accessibles via une interface conviviale, répondant aux attentes des utilisateurs.

Parmi les perspectives, l'intégration d'un système de collecte et d'analyse des avis et évaluations des transitaires par les utilisateurs apparaît comme un axe majeur d'amélioration. Ce dispositif permettrait d'enrichir le moteur de recommandation avec des critères qualitatifs, favorisant ainsi une meilleure transparence et une sélection plus fiable des partenaires transitaires, tout en renforçant la confiance des utilisateurs dans la plateforme.

Appendices

Annexe A

Annexe 1 : Fiche société Cargoween

Fiche société :

- Nom de la société Cargoween
- Date de création Octobre 2022.
- Statut S.A.R.L .
- Adresse

1-Siege :Avenue de la République Immeuble de L'Oranger,
App Nr 13. 1 er étage, Nabeul 8000- Tunisie.

2- Annexe : The Dot Rue Lac Malären1053, Tunis

- Téléphone +216 58 447 405 .
- web : cargoween.com
- Email : hello@cargoween.com

Statut Startup

-Starupact Label : 6/10/2022 (en pièce jointe)

-frenchtech label : 25/11/2022

Présentation Générale de Cargoween

Cargoween est une startup innovante basée en Tunisie, spécialisée dans le design, le développement et le déploiement d'applications digitales dédiées à l'industrie du fret aérien et plus largement au secteur de la supply chain.

Avec une approche résolument tournée vers l'intelligence artificielle, Cargoween propose des solutions technologiques avancées visant à optimiser les processus de réservation, de paiement et de gestion des itinéraires, contribuant ainsi à une économie plus verte et plus efficiente. Sa plateforme de réservation connectée aux compagnies aériennes facilite les échanges entre l'Europe et l'Afrique, avec un accent particulier sur la digitalisation et la réduction de la consommation de papier grâce à l'échange de documents numériques.

Cargoween se distingue comme la première solution technologique basée sur l'IA à révolutionner le secteur du fret aérien sur le corridor Europe-Afrique et dans la région MENA. En alliant expertise technique et engagement écologique, Cargoween accompagne les acteurs de la supply chain dans leur transition numérique tout en répondant aux défis logistiques actuels.

Organigramme simplifié

1. **Direction générale (CEO)**
 - Responsable de la vision, de la stratégie et de la gestion globale de l'entreprise.
2. **Directeur financier (CFO) § RH**
 - Gère la trésorerie, les finances, les budgets et les investissements.
3. **Directeur technique (CTO)**

- Responsable de l'innovation, du développement technologique et de l'infrastructure IT.

4. Responsable marketing (CMO)

- Développe les stratégies marketing, gère les campagnes et analyse les résultats.

5. Responsable commercial

- Dirige les efforts de vente, la gestion des relations clients et l'expansion du marché.

6. Équipes opérationnelles

- Comprend les divers départements qui contribuent au bon fonctionnement de l'entreprise (le support client, développeurs, commerciaux , Agents de fret).

Vous pouvez mettre ces infos dans un organigramme CHART GRAPHIQUE

Bibliographie

- [1] <https://www.pagesjaunes.fr/>. [Accessed 29-01-2025].
- [2] <https://www.mdpi.com/2504-4990/3/2/20>. [Accessed 19-05-2025].
- [3] <https://travaux.master.utc.fr/wp-content/uploads/sites/16/2021/05/2018-15-qpoap-cahiers-qualite-utc-vol-3-extrait-article-p137-144-compressed.pdf>. [Accessed 01-05-2025].
- [4] About GitHub and Git — GitHub Docs — [docs.github.com](https://docs.github.com/en/get-started/start-your-journey/about-github-and-git). <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>. [Accessed 26-05-2025].
- [5] Approche traditionnelle ou agilité : quelles différences ? — [cegos.fr](https://www.cegos.fr/ressources/mag/projet/agilite/approche-traditionnelle-agilite-differences). <https://www.cegos.fr/ressources/mag/projet/agilite/approche-traditionnelle-agilite-differences>. [Accessed 22-02-2025].
- [6] CamemBERT — [huggingface.co](https://huggingface.co/docs/transformers/model_doc/camembert). https://huggingface.co/docs/transformers/model_doc/camembert. [Accessed 05 – 2025].
- [7] GitHub Actions — [github.com](https://github.com/features/actions). <https://github.com/features/actions>. [Accessed 27-05-2025].
- [8] Home — [docker.com](https://www.docker.com/). <https://www.docker.com/>. [Accessed 27-05-2025].
- [9] La méthode MoSCoW pour prioriser les tâches de vos projets — [manager-go.com](https://www.manager-go.com/gestion-de-projet/dossiers-methodes/methode-moscow). <https://www.manager-go.com/gestion-de-projet/dossiers-methodes/methode-moscow>. [Accessed 23-02-2025].
- [10] Le tutoriel Python — [docs.python.org](https://docs.python.org/fr/3/tutorial/). <https://docs.python.org/fr/3/tutorial/>. [Accessed 26-05-2025].
- [11] Learn LaTeX in 30 minutes — [fr.overleaf.com](https://fr.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes). https://fr.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes. [Accessed 05 – 2025].
- [12] Lucidchart | Diagrammes alimentés par l'intelligence — [lucidchart.com](https://www.lucidchart.com/pages/fr). <https://www.lucidchart.com/pages/fr>. [Accessed 26-05-2025].
- [13] Machine learning operations landscape : platforms and tools - Artificial Intelligence Review — [link.springer.com](https://link.springer.com/article/10.1007/s10462-025-11164-3). <https://link.springer.com/article/10.1007/s10462-025-11164-3>. [Accessed 21-03-2025].
- [14] ml-ops.org — [ml-ops.org](https://ml-ops.org/content/crisp-ml). <https://ml-ops.org/content/crisp-ml>. [Accessed 01-05-2025].
- [15] Natural Language Processing (NLP) - Overview - GeeksforGeeks — [geeksforgeeks.org](https://www.geeksforgeeks.org/natural-language-processing-overview/). <https://www.geeksforgeeks.org/natural-language-processing-overview/>. [Accessed 26-03-2025].

- [16] Next.js by Vercel - The React Framework — nextjs.org. <https://nextjs.org/>. [Accessed 26-05-2025].
- [17] Outil de web design en ligne gratuit pour les équipes | Figma — figma.com. <https://www.figma.com/fr-fr/web-design>. [Accessed 26-05-2025].
- [18] Production-Grade Container Orchestration — kubernetes.io. <https://kubernetes.io/>. [Accessed 27-05-2025].
- [19] Que sont les MLOps ? – Explication des opérations de machine learning – AWS — aws.amazon.com. <https://aws.amazon.com/fr/what-is/mlops/> :text=MLOps [Accessed 19-05-2025].
- [20] Qu'est-ce que le Deep Learning ? - IA School — intelligence-artificielle-school.com. <https://www.intelligence-artificielle-school.com/deep-learning/>. [Accessed 26-03-2025].
- [21] Qu'est-ce que l'intelligence artificielle (IA) ? | Google Cloud — cloud.google.com. <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=fr>. [Accessed 19-03-2025].
- [22] Qu'est-ce que l'OCR ? – La reconnaissance optique de caractères expliquée – AWS — aws.amazon.com. <https://aws.amazon.com/fr/what-is/ocr/>. [Accessed 10-04-2025].
- [23] Qu'est-ce Que MongoDB ? — mongodb.com. <https://www.mongodb.com/fr-fr/company/what-is-mongodb>. [Accessed 26-05-2025].
- [24] Services de cloud computing | GoogleCloud | Google Cloud — cloud.google.com. https://cloud.google.com/?synth_kw=google+cloud+platform&gad_source=1&gad_campaignid=22561271914&gbraid=0AAAAAOXl6HoEfPQOuR&jSF2QyQPK&gl2rgclid=Cj0KCQjwxdXBhDEARIsAAUkP6iiQZEZBWY4qGYWCrui2Mo3lr2ubfNheWJyqbZ9KEksGNgnPi&fr. [Accessed 27-05-2025].
- [25] Top 10 des OCR Open Source en 2025 — koncile.ai. <https://www.koncile.ai/ressources/les-10-meilleurs-ocr-open-source-en-2025-doctr>. [Accessed 15-04-2025].
- [26] Traitement du langage naturel (NLP) : guide complet | Diabolocom — fr.diabolocom.com. <https://fr.diabolocom.com/blog/traitement-langage-naturel-nlp/>. [Accessed 19-03-2025].
- [27] Trello : qu'est-ce que c'est ? Découvrez les fonctionnalités, les utilisations et plus encore | Trello — trello.com. <https://trello.com/fr/tour>. [Accessed 26-05-2025].
- [28] Visual Studio : IDE et Éditeur de Code pour les Développeurs de Logiciels et les Équipes — visualstudio.microsoft.com. <https://visualstudio.microsoft.com/fr/>. [Accessed 26-05-2025].
- [29] What is Postman ? Postman API Platform — postman.com. <https://www.postman.com/product/what-is-postman/>. [Accessed 26-05-2025].
- [30] BURAK, A. Agile Frameworks Compared : Scrum, Kanban, Lean, XP — kanbanzone.com. <https://kanbanzone.com/2024/agile-frameworks-compared/>. [Accessed 22-02-2025].
- [31] HTTPS://WWW.FACEBOOK.COM/SCRUM-MASTERORG 105560232058267/. C'est Quoi Le WSJF - Weighted Shortest Job First ? - Scrum-Master.Org — scrum-master.org. <https://scrum-master.org/definition-du-wsjf-weighted-shortest-job-first/>. [Accessed 28-05-2025].

- [32] NIETERS, R. Qu'est-ce que l'OCR ? - Guide ultime OCR - Klippa — klippa.com. <https://www.klippa.com/fr/blog/informations/ocr-guide-ultime/> : :text=fonctionne [Accessed 10-04-2025].
- [33] ROBERT, J. Hugging Face : Tout ce qu'il faut savoir sur cette startup IA — datascientest.com. <https://datascientest.com/hugging-face-tout-savoir>. [Accessed 19-05-2025].
- [34] SAH, S. Machine learning : a review of learning types. *Preprints* (2020).
- [35] TEAM, G. Explication de la régression et de la classification dans l'apprentissage automatique — geekflare.com. <https://geekflare.com/fr/regression-vs-classification/>. [Accessed 19-03-2025].

RESUME

Vous devez mettre un résumé de votre rapport en langue française, en indiquant les mots clés. Si l'on prend l'exemple d'un projet qui traite du commerce électronique, un des mots clés peut être le commerce électronique. Le résumé ne doit pas dépasser les 10 lignes.

Mots clés : commerce électronique, mot, mot, mot.

SUMMARY

You should give an English abstract for your report. You have to indicate your key-words. For example, if your project is interesting on the e-commerce, this word can be a key-word. The abstract should not exceed 10 lines.

Key words : First Key word, Second Key Word, Third Key Word, Fourth Key Word.