# 1. Résumé / Abstract

Ce mémoire présente la conception et la réalisation, dans le cadre de la semaine APOCAL’IPSSI 2025, d’un prototype fonctionnel d’assistant intelligent de synthèse de documents pdf.

Le projet s’inscrit dans un contexte pédagogique, où une équipe d’étudiants doit développer en quatre jours un outil permettant d’uploader un document (contrat, rapport, norme…) au format PDF et d’obtenir automatiquement, grâce à une API de model d’intelligence artificielle générative, un résumé, les points clés et des suggestions d’actions.

La gestion de projet s’appuie sur la méthodologie Scrum, adaptée à un environnement volontairement « chaotique », chaque journée a été rythmée par des incidents imprévus imposés par l’encadrement, afin de simuler la réalité du monde professionnel et de développer l’agilité de l’équipe.

Ce document détaille le contexte, l’organisation de l’équipe, les choix techniques, le déroulement du développement, la gestion des imprévus, ainsi que le bilan et les perspectives d’amélioration du projet.

# 2. Table des matières

Table des matières

[1. Résumé / Abstract 1](#_Toc202254246)

[2. Table des matières 2](#_Toc202254247)

[3. Introduction 4](#_Toc202254248)

[4. Vision et analyse du besoin 4](#_Toc202254249)

[5. Analyse utilisateur 5](#_Toc202254250)

[5.1. Personas 5](#_Toc202254251)

[5.2. Parcours utilisateur 5](#_Toc202254252)

[6. Conception fonctionnelle 5](#_Toc202254253)

[6.1 Story Map 5](#_Toc202254254)

[6.2 Définition du MVP 6](#_Toc202254255)

[7. Gestion du projet Agile 6](#_Toc202254256)

[7.1 Organisation de l’équipe 6](#_Toc202254257)

[7.2 Méthodologie Scrum et adaptation 6](#_Toc202254258)

[8. Planification et suivi 6](#_Toc202254259)

[8.1 Release Planning 7](#_Toc202254260)

[8.2 Product Backlog 7](#_Toc202254261)

[8.3 Sprint Backlogs 7](#_Toc202254262)

[8.4 Burndown Charts 8](#_Toc202254263)

[8.5 Burnup Chart projet 8](#_Toc202254264)

[9. Développement & Réalisations 8](#_Toc202254265)

[9.1 Architecture technique 8](#_Toc202254266)

[9.2 Développement front-end et back-end 9](#_Toc202254267)

[9.3 Intégration et tests 9](#_Toc202254268)

[9.4 Gestion des incidents chaotiques 9](#_Toc202254269)

[10. Livrable final 11](#_Toc202254270)

[10.1 MVP livré (fonctionnalité, démonstration) 11](#_Toc202254271)

[10.2 Qualité UX/UI et performance 11](#_Toc202254272)

[10.3 Plan d’amélioration et rétrospective 11](#_Toc202254273)

[11. Bilan et conclusion 12](#_Toc202254274)

[11.1 Résultats atteints, points forts/faibles 12](#_Toc202254275)

[11.2 Enseignements de l’expérience du module 13](#_Toc202254276)

[12. Annexes et documentations 14](#_Toc202254277)

[Annexe 1 Product vison board : 14](#_Toc202254278)

[Annexe 2 fiches personas : 15](#_Toc202254279)

[Annexe 3 Customer Journey Map: 16](#_Toc202254280)

[Annexe 4 Story: 16](#_Toc202254281)

[Annexe 5 Trello: 17](#_Toc202254282)

[Annexe 6 Backlog: 18](#_Toc202254283)

[Annexe 7 tableau Sprint Backlogs: 19](#_Toc202254284)

[Annexe 8 burndown charts : 20](#_Toc202254285)

[Annexe 9 burnup charts : 21](#_Toc202254286)

# 3. Introduction

La semaine APOCAL’IPSSI 2025 a pour objectif de faire réaliser à chaque groupe d’étudiants un prototype d’assistant automatisé de synthèse de documents. Le projet vise à permettre à tout utilisateur de déposer un document PDF afin d’obtenir, grâce à une intelligence artificielle, un résumé structuré, les points clés du contenu et des suggestions d’actions adaptées.

Le travail a été mené en équipe selon une organisation agile, avec un suivi quotidien de l’avancement et une répartition claire des rôles de l’équipe 22 :

* Théo Gravez : Rédaction du mémoire
* Julien Binet : Présentation
* Yanis Helali : Déploiement Docker, suivi Scrum
* Yannick Coulibaly : Développement back-end
* Duncan Costes : Tests
* Balthazar Beguin : Développement front-end

Ce mémoire retrace l’ensemble de la démarche adoptée, des outils mobilisés, des livrables réalisés et dresse le bilan du projet, depuis l’analyse du besoin jusqu’à la livraison du prototype.

# 4. Vision et analyse du besoin

L’objectif principal du projet est de faciliter l’accès à l’information contenue dans des documents complexes ou volumineux. De nombreux utilisateurs, qu’ils soient particuliers ou professionnels, sont confrontés à la nécessité de lire, d’analyser et de synthétiser rapidement des contrats, rapports, normes ou tout autre document textuel.

Le projet vise donc à développer un outil permettant à un utilisateur de déposer un document PDF et d’obtenir en retour un résumé automatique, les points clés à retenir et des suggestions d’actions. Cette approche doit permettre de gagner du temps, d’améliorer la compréhension des documents et d’aider à la prise de décision.

La vision portée par ce POC est de démontrer la faisabilité technique d’un tel assistant de synthèse basé sur l’intelligence artificielle, avec une expérience utilisateur simple. Un dépôt de fichier, un résultat synthétique et exploitable en quelques secondes. À ce stade, le projet s’adresse à un large public sans distinction de secteur ou de métier, l’enjeu étant avant tout de valider le principe et l’intérêt du service.

L’analyse du besoin s’appuie sur l’observation d’usages courants, la multiplication des documents à traiter et la difficulté, pour beaucoup d’utilisateurs, d’extraire rapidement l’essentiel de contenus longs ou techniques. La solution proposée doit ainsi répondre à trois attentes majeures :

* La simplicité d’utilisation.
* La rapidité de traitement.
* La fiabilité des résumés générés.

*[Voir annexe 1 Product vision board](#_Annexe_1_Product)*

# 5. Analyse utilisateur

Dans le cadre de ce projet, l’analyse utilisateur vise à identifier les profils susceptibles d’utiliser un assistant de synthèse de documents et à comprendre leurs besoins spécifiques. Le produit étant un POC, la cible reste volontairement large, allant des particuliers souhaitant gagner du temps sur la lecture de documents personnels aux professionnels confrontés à de gros volumes de texte à traiter.

## 5.1. Personas

Plusieurs profils d’utilisateurs potentiels peuvent être envisagés :

* Le particulier qui souhaite résumer rapidement des documents administratifs, des contrats ou des rapports personnels.
* Le professionnel (ex : chef de projet, juriste, consultant) amené à analyser fréquemment des documents longs, techniques ou réglementaires.
* L’étudiant qui doit lire et synthétiser de la documentation dans le cadre de ses études.

Chaque persona présente des attentes communes : gagner du temps, simplifier la compréhension de documents, faciliter la prise de décision ou la gestion de tâches.

[*Voir annexe 2* fiches personas](#_Annexe_2_fiches)

## 5.2. Parcours utilisateur

Pour chaque persona identifiée, un parcours utilisateur a été décrit afin de visualiser les étapes de l’utilisation du service, depuis la découverte de l’outil jusqu’à l’obtention du résumé synthétique.

Les principales étapes sont :

* Accès à l’application (web ou mobile)
* Dépôt du document PDF
* Attente du traitement
* Visualisation du résumé, des points clés et des suggestions d’actions

[*Voir a*nnexe 3 Customer Journey Map](#_Annexe_3_Customer)

# 6. Conception fonctionnelle

## 6.1 Story Map

La Story Map permet de visualiser l’ensemble du parcours utilisateur à travers les différentes fonctionnalités nécessaires pour répondre au besoin initial. Cet outil aide à prioriser les développements en distinguant les fonctionnalités essentielles des options secondaires, tout en gardant une vue d’ensemble du produit.

[*Voir a*nnexe 4 Story Map :](#_Annexe_4_Story:)

## 6.2 Définition du MVP

Le Minimum Viable Product (MVP) retenu pour ce POC comprend les fonctionnalités essentielles permettant de valider le principe du service et d’offrir une première expérience utilisateur fonctionnelle.

Les fonctionnalités du MVP sont :

* Upload d’un document PDF via une interface simple.
* Traitement automatique du document (extraction du texte, envoi à une API d’intelligence artificielle, récupération du résumé)
* Affichage du résumé structuré avec les points clés et suggestions d’actions.

Les fonctionnalités secondaires (gestion d’historique, personnalisation de la synthèse, authentification, etc.) sont identifiées mais ne sont pas prioritaires à ce stade du projet.

# 7. Gestion du projet Agile

La gestion du projet s’est appuyée sur les principes de l’agilité, en particulier la méthode Scrum. L’organisation du travail a été structurée autour de la répartition claire des rôles, d’une planification et d’une communication régulière au sein de l’équipe.

## 7.1 Organisation de l’équipe

L’équipe a été constituée selon les spécialités de chaque membre, frontend, backend, base de données, gestion du mémoire et de la présentation. Chacun a pris en charge le développement d’une partie du projet en fonction de ses compétences. Cette spécialisation a permis d’optimiser le temps et de faciliter la coordination des tâches.

## 7.2 Méthodologie Scrum et adaptation

Le projet a été découpé en sprints de courte durée (2 journées). Un point d’avancement collectif était organisé chaque jour sur Teams afin de faire le bilan des tâches réalisées, de lever les éventuels blocages et de planifier les priorités pour la journée suivante. La planification des tâches et le suivi du projet ont été réalisés à l’aide de l’outil Trello, permettant à chaque membre de visualiser l’état d’avancement et les prochaines étapes.

[*Voir annexe 5 tableau Trello en annexe*](#_Annexe_5_Trello:)

L’équipe s’est adaptée aux éventuels imprévus/incidents survenus en cours de projet, en privilégiant l’entraide et la flexibilité dans la gestion des priorités. La démarche agile a favorisé la réactivité face aux obstacles et a permis de garantir une progression continue vers l’objectif final.

# 8. Planification et suivi

## 8.1 Release Planning

La planification du projet a été organisée en deux sprints couvrant les quatre jours du module. Chaque sprint regroupe un ensemble de fonctionnalités, depuis l’architecture technique jusqu’à la livraison finale du POC. Les jalons clés correspondent à la fin de chaque sprint et à la livraison des livrables majeurs.

Un calendrier prévisionnel a été établi en début de semaine, précisant les dates clés (démarrage, sprints, démo finale) et les ressources mobilisées sur chaque phase.

Calendrier prévisionnel :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Dates** | **Fonctionnalités principales** | **Jalons / Livrables attendus** | **Responsables** |
| Sprint 1 | J1 & J2 | Initialisation, architecture, page d’accueil, upload PDF, base de données, premier affichage du PDF uploadé | Fonction “upload PDF” opérationnelle, page d’accueil, base de données fonctionnelle | Balthazar (front), Yannick (back), Duncan (BDD) |
| Sprint 2 | J3 & J4 | Intégration IA, génération résumé, affichage résultat, téléchargement résumé, suggestions d’actions, correction bugs, déploiement docker CI/CD, préparation mémoire et soutenance | Résumé IA fonctionnel affiché, téléchargement OK,  Déploiement docker CI/CD OK, démo prête, mémoire et soutenance en cours | Balthazar (front), Yannick (back), Duncan (BDD), Yanis (Docker), Théo (mémoire), Julien (Soutenance) |

## 8.2 Product Backlog

Le Product backlog recense l’ensemble des fonctionnalités et tâches nécessaires à la réalisation du projet.  
Chaque user story a été priorisée selon la valeur apportée à l’utilisateur et les contraintes techniques, et enrichie de critères de définition de prêt (**DoR**) et de définition de fini (**DoD**).

[*Voir annexe 6 tableau backlog.*](#_Annexe_6_Backlog:)

## 8.3 Sprint Backlogs

Pour chaque sprint, un sprint backlog a été défini avec les tâches à réaliser, les responsables, et les critères d’acceptation associés.  
Cela a permis de suivre au quotidien l’avancée des développements et de réagir rapidement en cas de difficulté.

[*Voir annexe 7 tableau Sprint Backlogs*](#_Annexe_7_tableau)

## 8.4 Burndown Charts

Des burndown charts ont été utilisés pour chaque sprint afin de visualiser la progression du travail restant à effectuer au fil du temps.  
Ces graphiques permettent de vérifier si l’équipe suit le rythme prévu et d’identifier d’éventuels retards ou avances par rapport au planning initial.

L’analyse des burndown charts (voir Annexe 8) montre que, pour chaque sprint, l’équipe est restée parfaitement alignée avec la courbe idéale.  
Au Sprint 1, la réalisation des tâches a suivi exactement le plan prévu, sans retard ni avance.  
Au Sprint 2, une légère avance a été prise dès le premier jour, ce qui a permis de terminer le sprint avec sérénité et sans pression de dernière minute.  
Globalement, la bonne gestion du temps et la coordination de l’équipe ont permis de respecter les échéances fixées.

[*Voir annexe 8 graphique Burndown Charts*](#_Annexe_8_burndown)

## 8.5 Burnup Chart projet

Le burnup chart (voir Annexe 9) complète l’analyse des burndown charts en illustrant la progression cumulative du travail accompli tout au long des deux sprints, comparée à l’objectif final fixé. On constate que la courbe du travail réalisé monte régulièrement jusqu’à atteindre la ligne du scope total à la fin de chaque sprint, ce qui confirme la bonne tenue du planning et l’absence de retard significatif.  
Aucun changement de périmètre n’ayant été observé (le scope total est resté constant), l’ensemble des tâches prévues a été accompli dans les délais.

[*Voir annexe 9 graphique Burnup Charts*](#_Annexe_9_burnup)

# 9. Développement & Réalisations

## 9.1 Architecture technique

Le projet adopte une séparation entre le front-end (interface utilisateur), le back-end (traitement et logique métier) et la base de données (stockage).

* **Front-end** : développé avec **TypeScript** et **Next.js**, il propose une interface web légère et réactive. Le choix de Next.js a permis d’accélérer la mise en place du projet grâce à ses fonctionnalités de rendu côté serveur (SSR) et à la gestion optimisée des routes. L’utilisation de la librairie **shadcn/ui** a facilité la création rapide de composants.
* **Back-end** : construit avec **Python** et le framework **Flask**, il assure la gestion des requêtes, le traitement des documents et la communication avec l’API d’intelligence artificielle. Flask a été retenu pour sa simplicité, sa rapidité de mise en œuvre et la richesse de son écosystème pour les applications REST.
* **Base de données** : le stockage des documents et des métadonnées sur **PostgreSQL**.
* **Déploiement et intégration continue** : l’ensemble du projet est conteneurisé avec **Docker**, facilitant le déploiement sur tout environnement compatible et assurant la reproductibilité des builds. L’intégration continue / déploiement continu (CI/CD) a été mise en place via des pipelines Docker pour garantir la qualité et l’automatisation des livraisons.

## 9.2 Développement front-end et back-end

**Front-end**  
L’interface a été conçue pour garantir une expérience utilisateur simple : upload du PDF, affichage du résumé.  
Next.js a facilité l’intégration des composants visuels. L’upload de document et l’affichage des résultats ont été traités en priorité, conformément au MVP défini. Des hooks React ont permis de gérer l’état des composants et de simplifier la remontée d’informations entre les différentes pages.

**Back-end**  
Le serveur Flask reçoit les fichiers PDF, effectue une extraction de texte (via la librairie pdfminer ), puis appelle l’API d’intelligence artificielle externe Hugging Face. Les résultats sont renvoyés vers le front pour affichage.  
Des endpoints RESTful clairs ont été définis pour l’upload et la génération du résumé.

## 9.3 Intégration et tests

L’intégration des différents modules (front, back, base de données) a été réalisée progressivement au fil des sprints. Des tests manuels ont été menés à chaque étape clé :

* Vérification de l’upload et du stockage des PDF
* Test des échanges front ↔ back-end (API)
* Validation de la génération de résumés par l’IA
* Tests de téléchargement et de restitution des résultats

Des scripts de test simples ont permis de simuler différents cas d’usage (fichiers volumineux, documents non conformes). L’intégration continue via Docker a permis de détecter rapidement les éventuelles régressions lors des builds.

## 9.4 Gestion des incidents chaotiques

Tout au long du projet, plusieurs incidents imprévus ont perturbé le déroulement des sprints, simulant des situations de crise fréquentes dans le monde professionnel.  
Chaque incident a été traité collectivement, et l’équipe a pris soin de documenter la situation ainsi que les solutions trouvées pour rebondir rapidement.

Incident 1 : Indisponibilité soudaine de l’IA

Résumé :  
En pleine session de développement, l’équipe s’est retrouvée confrontée à une maintenance imprévue de l’outil d’IA principal, rendant le service inaccessible pour plusieurs heures.

Réaction / Solutions mises en place :

Réorganisation temporaire du travail : priorisation des tâches ne nécessitant pas l’IA (UI, base de données, documentation, etc.)

Recherche et test rapide d’outils alternatifs pour pallier l’indisponibilité

Mise en place d’une procédure de veille : surveillance régulière du statut des outils critiques

Rétrospective en fin de journée pour identifier les bonnes pratiques à adopter en cas de nouvelle panne

Incident 2 : Fiche persona incomplète

Résumé :  
Un persona utilisateur transmis au groupe s’est avéré incomplet (absence d’informations clés : fréquence d’utilisation, besoins métier, etc.), compromettant la qualité de la Customer Journey Map.

Réaction / Solutions mises en place :

Organisation d’un point d’équipe rapide pour compléter la fiche à partir de discussions et d’exemples réels

Recherche d’informations complémentaires auprès du Product Owner et d’utilisateurs cibles

Validation de la fiche persona enrichie avant de poursuivre la conception des parcours et des priorités produit

Incident 3 : Changement de modèle d’IA imposé en dernière minute

**Résumé :**  
Lors du Sprint Planning, le Product Owner a annoncé un changement soudain de politique : l’utilisation de l’API OpenAI, initialement prévue, est désormais interdite pour des raisons de budget et de souveraineté. L’équipe doit donc basculer vers un modèle open source à héberger en interne. Ce pivot implique de nouveaux défis techniques :

Réaction / Solutions mises en place :

* Analyse rapide de l’état de l’art des modèles open source disponibles et sélection d’une solution adaptée
* Planification de sessions de montée en compétence sur l’installation et la sécurisation de modèles IA en serveur local
* Mise en place d’un suivi spécifique des risques liés à la sécurité et à la performance
* Réévaluation du planning avec le Product Owner pour intégrer l’incertitude et prioriser les tâches critiques
* Partage régulier d’un état d’avancement avec l’ensemble des parties prenantes afin d’anticiper d’autres éventuelles adaptations

**Mettre les prochain evenement ici**

# 10. Livrable final

## 10.1 MVP livré (fonctionnalité, démonstration)

Le livrable final du projet APOCAL’IPSSI est un MVP (Minimum Viable Product) fonctionnel, conforme au périmètre défini lors de la phase de conception.  
L’outil permet à l’utilisateur de :

* Accéder à une page d’accueil claire
* Importer un document PDF via une interface dédiée
* Lancer la génération automatique d’un résumé à l’aide d’une IA externe par API
* Consulter le résumé synthétique ainsi que des suggestions d’actions

## 10.2 Qualité UX/UI et performance

Un effort particulier a été porté à l’ergonomie de l’interface et à la simplicité du parcours utilisateur.  
L’utilisation de la librairie shadcn/ui a permis d’obtenir des composants modernes et accessibles.  
Des tests utilisateurs rapides (auto-tests par l’équipe) ont permis d’identifier et de corriger les points de friction majeurs (ex : feedback sur l’upload, gestion des erreurs en cas de fichier non valide).

Sur le plan technique, l’application a démontré de bonnes performances sur l’ensemble du MVP :

* Temps de traitement IA raisonnable (quelques secondes pour un document de taille standard)
* Réactivité de l’interface web (chargement rapide, transitions fluides)
* Robustesse du back-end (peu ou pas de crash, gestion correcte des erreurs)

Des marges d’amélioration subsistent sur la gestion des très gros fichiers ou l’optimisation de l’expérience mobile, qui pourront être adressées lors de futures itérations.

## 10.3 Plan d’amélioration et rétrospective

À l’issue du projet, plusieurs pistes d’amélioration ont été identifiées pour enrichir le service et renforcer sa robustesse :

* Fonctionnalités à ajouter :
  + Prise en charge d’autres formats de documents (Word, images)
  + Personnalisation du résumé (niveau de synthèse, choix du style)
  + Historique des fichiers traités et gestion des comptes utilisateur
* Optimisations UX/UI :
  + Amélioration du feedback utilisateur (états de chargement, messages d’erreur plus explicites)
  + Adaptation de l’interface pour mobile et tablette
* Performance et fiabilité :
  + Parallélisation du traitement pour réduire encore les temps d’attente
  + Mise en place d’un échange via streaming avec l’API IA notamment avec SSE
  + Mise en place de tests automatisés et de monitoring

Rétrospective d’équipe  
La méthodologie agile et la communication régulière sur Teams ont favorisé la résolution rapide des problèmes et la montée en compétence de chacun.  
L’équipe a souligné l’importance de la flexibilité face aux imprévus, et identifié la valeur ajoutée d’une documentation partagée et vivante (Trello, mémoire collaboratif).  
Pour un prochain projet, il serait pertinent d’impliquer davantage les utilisateurs finaux dès le début pour affiner les besoins et de prévoir des solutions de secours en cas d’indisponibilité d’un outil critique.

# 11. Bilan et conclusion

## 11.1 Résultats atteints, points forts/faibles

Le projet APOCAL’IPSSI a permis de livrer un MVP opérationnel, conforme aux attentes initiales.  
Tous les parcours essentiels ont été couverts : import de PDF, génération et restitution d’un résumé automatique, interface simple et efficace.  
La démonstration a montré la faisabilité technique de l’outil et son utilité pour les utilisateurs cibles identifiés (particuliers, professionnels, étudiants).

Points forts :

* Cohésion de l’équipe et réactivité face aux imprévus (cf. incidents documentés)
* Respect du planning agile, bonne gestion des sprints et des livrables intermédiaires
* Simplicité et clarté de l’interface, rapidité de prise en main

Points faibles ou limites :

* Dépendance forte à des outils externes (API IA, plateformes tierces) qui a entraîné des perturbations
* Peu de tests utilisateurs externes, difficulté à valider la pertinence fonctionnelle auprès d’un large public
* Améliorations possibles sur l’expérience mobile, la gestion des fichiers volumineux ou la personnalisation des résultats

## 11.2 Enseignements de l’expérience du module

Au terme de cette semaine de développement intensif, plusieurs enseignements clés peuvent être retenus.  
Tout d’abord, ce projet a clairement mis en évidence l’importance de l’adaptabilité et d’une gestion de crise efficace lorsqu’on évolue dans un environnement caractérisé par des contraintes fortes de temps et de ressources. Malgré les imprévus, l’équipe a su réagir avec réactivité et solidarité, en ajustant la répartition des tâches et en maintenant une dynamique de groupe constructive.

Par ailleurs, l’expérience a contribué à renforcer la maîtrise des outils agiles : la planification par sprints, la priorisation des tâches, ainsi que l’organisation et la documentation du travail d’équipe sont devenues des réflexes. De plus, il a fallu préserver la motivation et l’implication de chacun, même lors des phases de blocage ou d’incertitude.

En somme, trois principaux enseignements se dégagent de cette expérience :

* Premièrement, il est essentiel d’anticiper systématiquement une solution de repli en cas d’indisponibilité d’un outil critique, afin d’éviter toute paralysie du projet.
* Deuxièmement, la documentation de chaque incident ou adaptation constitue un véritable atout pour capitaliser sur l’expérience acquise et améliorer continuellement les processus collectifs.
* Enfin, il apparaît que l’agilité ne porte ses fruits que si la communication d’équipe reste active, régulière et transparente à chaque étape du projet.

Dans cette perspective, plusieurs axes de progrès peuvent être envisagés pour de futurs projets.  
D’une part, il serait pertinent d’intégrer plus en amont des utilisateurs “réels” dans la boucle de feedback, afin de valider l’adéquation du produit aux besoins concrets.  
D’autre part, l’élaboration d’un plan de tests plus complet, intégrant notamment des tests automatisés et des scénarios d’erreur variés, permettrait d’accroître la fiabilité et la robustesse de l’application.  
Enfin, il conviendra de poursuivre les efforts engagés en matière d’ergonomie et d’évolution de l’architecture technique, afin d’accompagner sereinement de futurs développements.

En définitive, cette expérience s’est révélée particulièrement formatrice, aussi bien sur le plan technique qu’humain, et pose des bases solides pour l’approche et la réussite de projets collectifs à venir.

# 12. Annexes et documentations

## Une image contenant texte, capture d’écran, nombre, Police Le contenu généré par l’IA peut être incorrect.Annexe 1 Product vison board :

## Une image contenant texte, capture d’écran, Police Le contenu généré par l’IA peut être incorrect.Une image contenant texte, capture d’écran Le contenu généré par l’IA peut être incorrect.Annexe 2 fiches personas :

## Annexe 3 Customer Journey Map:



## Annexe 4 Story:

|  |  |  |
| --- | --- | --- |
| **Étape** | **MVP (1ère ligne)** | **Futur (2ème ligne)** |
| Découverte | Accès à l’application, page d’accueil claire | Tutoriel, FAQ, Authentification |
| Upload du document | Sélectionner PDF, bouton “Envoyer” | Multi-upload, drag & drop |
| Traitement IA | Génération de résumé via API IA | Choix du type de synthèse, résumé personnalisé |
| Résultats | Affichage résumé, points clés, suggestions | Export formats variés, surlignage |
| Exploitation | Télécharger/copie du résumé | Partage email, historique, intégration cloud |

## Annexe 5 Trello:

**Une image contenant texte, logiciel, Système d’exploitation, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.**

## Annexe 6 Backlog:



## Annexe 7 tableau Sprint Backlogs:

Sprint 1 :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tâche** | **Responsable** | **Critère d’acceptation** | **Durée tâche** |
| Initialiser le dépôt Git | Yannick | Dépôt accessible à l’équipe | 0h15 |
| Mettre en place l’architecture front | Balthazar | Page d’accueil fonctionnelle | 2h |
| Mettre en place le backend Flask | Yannick | Serveur Flask lancé | 4h |
| Créer la base PostgreSQL | Duncan | Connexion testée | 4h |
| Développer l’upload PDF | Balthazar | Fichier uploadé, stocké | 2h |
| Stocker les fichiers | Duncan | PDF présent en base | 2h |
| Premier affichage PDF uploadé | Balthazar | PDF affiché après upload | 2h |

Sprint 2 :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tâche** | **Responsable** | **Critère d’acceptation** | **Durée tâche** |
| Intégrer l’API IA pour la synthèse | Yannick | Résumé IA généré | 1h |
| Afficher le résumé et suggestions | Balthazar | Résumé et suggestions affichés | 2h |
| Permettre le téléchargement du résumé | Balthazar | Résumé téléchargeable | 0h30 |
| Corriger les bugs & tests | Toute l’équipe | Aucun bug bloquant | 8h |
| Déployer Docker CI/CD | Yanis | Déploiement automatisé opérationnel | 4h |
| Rédiger et finaliser le mémoire | Théo | Mémoire complet à J4 | 8h |
| Préparer la soutenance | Julien | Slides de soutenance prêtes | 6h |
| Préparer la démo finale | Toute l’équipe | Démo prête, livrable remis | 4h |

## Annexe 8 burndown charts :

## Annexe 9 burnup charts :