

GNG2501
Mise à jour du progrès du projet de conception

FB2- 4

Soumis par:

Ahmed Moustapha, 300362736

Genadri Alfred, 300170731

Masena Bébélo-Noah, 300189788

Taleb Nizar, 300371104

Sep 28 2024

Université d'Ottawa

Table des matières

Table des matières.....	i
Liste des figures	iii
Liste des tables	iv

Liste des acronymes et glossaire	v
1 Introduction.....	1
2 Rapport de développement durable et CPX.....	2
2.1 Rapport de développement durable	2
2.2 Conception pour X.....	5
3 Définition du problème, développement de concepts et plan de projet.....	9
3.1 Définition du problème	9
3.2 Développement des concepts	13
3.3 Plan de projet	23
4 Conception détaillé et NDM	24
4.1 Facteurs CPX	25
4.2 Conception détaillé	27
4.3 Compétences et ressources disponibles	28
4.4 Estimation du temps et disponibilité.....	30
4.5 Hypothèses critiques	30
4.6 NDM.....	31
4.7 Plan de Projet	31
5 Conclusions.....	32
6 Bibliographie.....	33

Liste des figures

Insert your list of figures here (click-droit pour mettre à jour ce champ).

Liste des tables

Table 1. Acronyms.....	v
Table 2. Glossary	v

Liste des acronymes et glossaire

Table 1. Acronymes

Acronyme	Définition
AI	Artificial Intelligence
CD	Cloud Deployment
DB	Database
FS	Feedback System
UI	User Interface

Table 2. Glossaire

Terme	Acronyme	Définition
Application Programming Interface	API	Set of rules or protocols that enables software applications to communicate with each other

1 Introduction

Notre objectif serait de livrer un produit présentable et fonctionnel d'un prototype bilingue d'un assistant d'AI qui permet de supporter les individus et leurs familles selon les besoins spécifiques du client. Un autre objectif serait de produire un prototype qui atteint un niveau d'accessibilité et qui garantir la sécurité des individus qui auront accès aux produits.

2 Rapport de développement durable et CPX

2.1 Rapport de développement durable

Le développement de notre produit engendre des impacts importants sur les plans social, environnemental et économique. Dans cette section, nous analysons les impacts de notre projet d'IA d'assistance à travers ces trois dimensions, en tenant compte des concepts du résultat net triple et de l'analyse de cycle de vie (ACV). D'un point de vue social, l'IA d'assistance vise à améliorer l'accessibilité pour les personnes en situation de handicap. En leur fournissant un accès facile à des ressources et à de l'aide, le projet contribue à une meilleure inclusion sociale. Cependant, il existe un risque que les utilisateurs deviennent trop dépendants de la technologie, ce qui pourrait diminuer les interactions humaines essentielles. Le projet d'IA d'assistance utilise des infrastructures cloud pour l'hébergement des données et le traitement des réponses. Cela permet de minimiser l'utilisation de serveurs physiques, ce qui réduit la consommation d'énergie. Toutefois, il est important de noter que l'hébergement cloud a tout de même un impact environnemental lié à l'énergie utilisée par les centres de données. Une amélioration potentielle pourrait être l'utilisation de services cloud écologiques qui réduisent leur empreinte carbone en utilisant des énergies renouvelables. Économiquement, l'IA d'assistance permet de diminuer les dépenses liées aux services de soutien en automatisant les processus, ce qui réduit le besoin d'interventions humaines. Cela aide les organisations à économiser sur les coûts de main-d'œuvre et à améliorer l'efficacité. Cependant, la mise en œuvre de cette technologie peut entraîner des coûts initiaux importants, notamment pour le développement, l'infrastructure technologique, et la maintenance continue.

Résultat triple	Impact positif	Impact négatif
Social	Amélioration de l'accessibilité et inclusion sociale accrue pour les personnes en situation de handicap	Risque de dépendance excessive à la technologie, réduisant les interactions humaines essentielles

Environnemental	Réduction de l'utilisation de serveurs physiques grâce à l'hébergement cloud	Consommation d'énergie élevée par les centres de données, nécessitant une amélioration potentielle vers des services cloud écologiques
Économique	Réduction des coûts de main-d'œuvre grâce à l'automatisation des processus	Coûts initiaux élevés pour le développement, l'infrastructure et la maintenance continue

2.1.2 Analyse du cycle de vie

On base cette analyse sur un produit similaire existant comme une plateforme web avec un assistant virtuel (je me base sur le site Acuity Insights; l'entité responsable des tests CasPer).

2.1.2.1 Objectif et champ d'étude

L'objectif est d'évaluer l'impact environnemental d'un site web qui utilise un chatbot pour répondre aux questions des utilisateurs. L'ACV permettra d'identifier les principales sources d'impacts, telles que la consommation d'énergie et les émissions de CO2 liées au fonctionnement du site et à l'utilisation du chatbot.

Le système inclut les infrastructures matérielles (serveurs, data centers), les réseaux de communication (transfert de données) ainsi que l'infrastructure nécessaire au chatbot (modèles d'IA et leurs besoins en ressources).

Les phases du cycle de vie couverte sont: conception, utilisation, maintenance, et fin de vie des entités (serveurs, équipements, IA).

2.1.2.2 Analyse d'inventaire

Notre produit requiert plusieurs parties: l'infrastructure dont la plateforme sera déployée, l'apprentissage et l'inférence du modèle IA, les réseaux et le transfert de donnée.

Le site web et le chatbot auront besoin d'une infrastructure. Similairement à Acuity Insights, on utilisera une infrastructure d'un fournisseur cloud. On considère les serveurs dédiés à l'hébergement du site web, la base de données qui stocke les informations. Les modèles d'IA, en particulier ceux basés sur des algorithmes complexes (comme les réseaux neuronaux, LLMs), seront utilisés pour l'inférence.

Les données sur la consommation d'énergie peuvent être basées sur des moyennes connues pour l'hébergement cloud (kWh utilisés). On considère aussi l'énergie nécessaire pour maintenir les serveurs et autres infrastructures pour le fournisseur Cloud (e.g., refroidissement). On considère aussi la fréquence des interactions avec le chatbot et la durée moyenne d'une session d'interaction afin d'estimer la quantité d'information (Mb utilisés). On regardera le volume de données transférées entre les serveurs et les utilisateurs, et l'énergie consommée par le réseau pour ces transferts.

2.1.2.3 Évaluation d'impact

On évaluera notre impact carbone en convertissant la consommation énergétique des serveurs, réseaux et infrastructure en émissions de CO₂. On pourrait aussi considérer la consommation énergétique des dispositifs des utilisateurs lors de l'accès au site ou l'utilisation du chatbot. On analysera la consommation des serveurs et leur impact en termes de consommation de matières premières (e.g., serveurs, batteries). On pourrait aussi évaluer les impacts indirects, tels que le besoin d'infrastructure de soutien (e.g., tours de communication, câblage).

2.1.2.4 Interprétation

L'analyse pourrait révéler que l'étape d'utilisation requiert le plus d'énergie. Par exemple, si le chatbot est basé sur un modèle d'IA complexe, la consommation énergétique du modèle pourrait représenter une partie importante de l'empreinte carbone totale.

On pourrait considérer l'optimisation du chatbot ou réduire la complexité du modèle afin de diminuer les coûts énergétiques. On pourrait aussi migrer vers un fournisseur cloud qui est plus efficace en termes de stockage de données, de refroidissement, par exemple. On pourrait développer un site web léger afin d'utiliser moins d'énergie du côté de l'utilisateur. De plus, on pourrait encoder les données afin de réduire la quantité d'information envoyées sur les réseaux.

2.2 Conception pour X

N°	Besoin	CPX	5
1	L'application est accessible aux personnes atteintes d'un handicap	Accessibilité	5
2	L'assistant est sensible aux besoins psychologiques de l'utilisateur	Support (Réconfort)	5
3	L'application est sécuritaire	Sécurité	5
4	L'assistant donne une réponse selon un délai raisonnable	Vitesse	3
5	L'assistant permet d'être utilisé en posant des questions par voix et par écrit ainsi que répondre par audio ou par écrit	Facilité d'utilisation	4

1. Accessibilité

- Niveau de Conformité AA

- L'application suivra les directives de WCAG 2.1 niveau AA, un standard reconnu pour garantir une accessibilité acceptable pour les utilisateurs ayant différents types de handicaps.
- Contient le texte alternatif (descriptions textuelles des images), contraste suffisant (ratio supérieur à 4) pour garantir lisibilité, média synchronisé (sous-titres).
- Fonctionnalités doivent être accessibles via le clavier, indicateur de focus lorsqu'on utilise le clavier.

- Accessibilité des composants interactifs

- Étiquetage descriptif des composants interactifs (boutons, formulaires) pour lecteurs d'écran.
- Messages d'erreur, confirmations et alertes sont accessibles aux lecteurs (aria-live).
- Tester la compatibilité avec lecteurs d'écran.
- L'application accommode l'utilisateurs selon le handicap de l'utilisateur, exp :

1. Visuelle : Synthèse vocale du texte

2. Neuropathie : Communication avec l'application par voix

- Métriques

- i. Atteindre un taux de conformité de 95% au niveau AA pour toutes les pages.
- ii. 90+% des fonctionnalités doivent être accessibles uniquement via le clavier.

2. Support

- Communiquer bien et être sensible aux situations de détresse, ainsi qu'offrir un soutien psychologique et/ou social à l'utilisateur.
- Tests d'utilisateurs pour déterminer le niveau.
- Estimer le niveau d'engagement.
- Sensibilité au soutien psychologique, rendement des réponses
 - i. Pourcentage où l'assistant identifie correctement les signaux de détresse.
 - ii. Taux de satisfaction supérieur à 80% sur la pertinence des réponses fournies.
 - iii. Moyenne de messages par session
 - iv. Niveau de rétention d'utilisateurs (retour)

3. Sécurité

- Empêcher la fuite de données sensibles; encryption des données
- Protection des données
 - i. Aucune faille critique détectée lors des tests de pénétration.
 - ii. Utilisation de TLS 1.2+ pour toutes communications, avec un objectif de 100% de conformité.
 - iii. Pourcentage de données encryptées (au repos et en transit)
 - iv. Pourcentage d'essais d'authentification réussis vs non-réussis.
 - v. Pourcentage de conformité aux standards de sécurité (GDPR, HIPAA, AES-256)

4. Vitesse

- Assurer une expérience fluide et rapide (éviter l'aspect robotique d'un chatbot et rendre la conversation plus réelle)
- Performance de l'application

- i. Moins de **2 secondes** pour les réponses basiques et moins de **45 secondes** pour des réponses plus complexes.
- ii. L'accès initial au site ne doit pas dépasser **1.5 secondes** sur une connexion à haut débit.

5. Facilité d'utilisation

- Plusieurs objectifs: rendre l'application intuitive, offrir du support multilingue.
- Simplicité de l'assistant et l'application; vitesse auquel une personne peut comprendre les fonctionnalités et l'utilisation de l'application
 - i. Objectif est de rendre l'application intuitive aux utilisateurs.
 - ii. Un nouvel utilisateur doit comprendre les fonctionnalités principales en moins de 3 minutes, mesuré par des tests utilisateurs.
 - iii. Niveau de satisfaction supérieur à **80%** sur la simplicité d'utilisation, mesuré par des tests d'utilisateurs.
- Multiples méthodes d'interaction et de communication
 - i. Support pour au moins **2 modes de communication** (texte et voix).
 - ii. Précision d'au moins 85% pour la reconnaissance vocale.
- Support multilingue
 - i. Supporter au moins 2 langues (anglais et français).
 - ii. Niveau de satisfaction supérieur à 80% des utilisateurs pour la qualité des réponses dans leur langue.

3 Définition du problème, développement de concepts et plan de projet

3.1 Définition du problème

3.1.1 Liste des besoins des clients

- Accès aux ressources de soutien psychologique et social: Les personnes en situation d'handicap ou en détresse psychologique rencontrent des obstacles à l'accès de ressources, notamment l'isolement social, l'inaccessibilité numérique ou physique, le manque de sensibilisation.
- Support personnalisé: Les solutions génériques ne répondent pas aux besoins spécifiques des individus, notamment ceux avec des handicaps sensoriels ou cognitifs.
- Accès multilingue (bilingue au moins): La plateforme doit être disponible en français et en anglais pour atteindre un public large et diversifié.
- Protection des données personnelles: Le respect de la confidentialité et la sécurité des informations sensibles des utilisateurs sont essentiels.
- Récolte de rétroaction: Un système qui permet de recueillir les avis des utilisateurs pour améliorer continuellement la plateforme.

3.1.2 Informations connues

- Des chatbots IA sont déjà utilisés dans le domaine de la santé mentale pour offrir un soutien émotionnel, mais ils manquent de fonctionnalités spécifiques requises.
- Les Web Content Accessibility Guidelines (WCAG) sont une norme internationale pour l'accessibilité du web, assurant que les sites sont utilisables par les personnes ayant différents types de handicaps.

3.1.3 Informations inconnues

- Quelle est la profondeur de personnalisation nécessaire pour répondre aux besoins spécifiques des personnes en situation de handicap ? Cela inclut la capacité de l'IA à comprendre et s'adapter aux préférences personnelles, au niveau de handicap, et aux conditions psychologiques des utilisateurs.
- Le coût de fonctionnement et de maintenance d'une plateforme offrant un service continu, notamment la disponibilité 24h/7, n'est pas encore connu.
- Il reste à comprendre dans quelle mesure les utilisateurs fourniront des réponses pertinentes et honnêtes via le système de feedback. S'ils sont en détresse psychologique ou s'ils souffrent de handicaps cognitifs, cela peut influencer la manière dont ils interagissent avec les systèmes de rétroaction.

3.1.4 Énoncé du problème

Le problème touche les personnes en situation de handicap et celles en détresse psychologique qui n'ont pas accès aux ressources de soutien adéquates en raison d'infrastructures inadaptées, de services limités, et d'un manque de personnalisation. Les solutions doivent prendre la forme d'une plateforme en ligne fiable qui inclut un chatbot d'intelligence artificielle, capable de fournir un soutien personnalisé, sécurisé et accessible dans une période acceptable (i.e., CPX) ainsi qu'un système de rétroaction.

3.1.5 Liste des métriques

- Multilinguisme
 - Langues du Chatbot: pas d'unités mais plutôt une liste finie d'options (e.g., [Anglais, Français]).
- Accessibilité
 - Niveau de conformance à WCAG: en niveau (e.g., A, AA).
 - Compatibilité avec les lecteurs d'écran: en pourcentage.
- Efficacité
 - Temps de réponse du chatbot: en secondes.
- Sécurité
 - Niveau d'encryption: liste finie d'options (e.g., AES-256)

- Disponibilité: en pourcentage.
- Facilité d'utilisation
 - Taux de satisfaction d'utilisateurs: en pourcentage.
 - Temps requiert pour faire une tâche: en secondes.

3.1.6 Étalonnage

- ChatGPT : IA multilingue, disponible en français et anglais, fournit des réponses précises et contextuelles.
- Google Assistant : Support multilingue avec une bonne reconnaissance vocale pour l'accessibilité, mais limitée pour les utilisateurs avec des besoins spécifiques.
- Wysa: IA conçue pour la santé mentale qui guide les utilisateurs à travers des exercices de thérapie cognitive et fournit des interventions personnalisées. Offre un soutien en santé mentale avec des interventions basées sur l'IA, mais n'est pas optimisé pour les personnes en situation de handicap ou multilingue.

	Facteurs	ChatGPT		Google Assistant		Wysa IA	
		Score		Score		Score	
Langages	1	4	4	3	3	2	3
Accessibilité	5	4	20	4	20	4	20
Temps de réponse	4	3	12	3	12	4	16
Encryption	5	5	25	5	25	4	20
Disponibilité	5	5	25	5	25	5	25
Taux de satisfaction	4	4	20	4	20	4	20
Facilité d'utilisation	4	4	16	4	16	5	20
Pointage total			122		121		124

Tableau 1. Matrice décisionnelle de étalonnages de produits similaires qui mesure les spécifications cibles de notre conception. Les facteurs varient entre 1 et 5 où 1 est moins important et 5 est très important.

Justifications des facteurs des métriques

- **Langages – 1** : Il est seulement nécessaire que le concept puisse communiquer en français et anglais, les autres langues ne sont pas importantes au succès du produit finale.
- **Accessibilité – 5**: C'est la métrique la plus important puisque ceci affecte directement notre clientèle cible.
- **Temps de réponse – 4** : Notre conception serait utilisée dans des moments de détresse, ainsi il est important que le logiciel puisse donner une réponse le plus rapide que possible.
- **Encryption – 5** : Étant donné que le logiciel a accès à de l'information personnelle et sensible de l'utilisateur, il est très important que ces données ne puissent pas être divulguer.
- **Disponibilité – 5** : Dû à la nature du problème que notre conception tente de résoudre, notre conception doit être toujours disponible ainsi cette spécification est très importante.
- **Taux de satisfaction – 4** : Ce facteur est assez important puisqu'il nous permet de mieux accommoder le client. Bref il nous est très important que le client continue à utiliser notre concept.
- **Facilité d'utilisation – 4** : Le produit doit être facile à utiliser puisqu'il sert comme outil que le client pourrait utiliser quotidiennement et nous aurions des utilisateurs avec un handicap physique ainsi le concept devrait être facile à utiliser.

3.1.7 Spécifications cibles

- **Accessibilité** : Certification WCAG 2.1 niveau AA minimum pour assurer une utilisation facile pour tous les types d'utilisateurs.
- **Langues**: Contient au moins [Anglais, Français] pour garantir une expérience bilingue.
- **Temps de réponse**: Moins de 5 secondes pour s'assurer d'une interaction fluide, efficace et réduire l'aspect robotique.
- **Sécurité (protection de données)**: Cryptage AES-256 pour protéger les informations sensibles.
- **Disponibilité** : Taux de disponibilité d'au moins 90% pour garantir un accès constant à la plateforme.

Les choix de ces spécifications sont motivés par la nécessité s'assurer que la plateforme soit non seulement fonctionnelle, mais aussi adaptée aux besoins spécifiques des personnes en situation de handicap et en détresse psychologique. Cela garantit l'accessibilité, la sécurité, l'efficacité et la personnalisation des services proposés.

3.2 Développement des concepts

3.2.1 Sous-systèmes

Interface utilisateur (UI)

L'interface utilisateur permet aux utilisateurs de poser des questions, soit par texte, soit par voix, et de recevoir des réponses de l'assistant AI. Cette interface est connectée au backend via une API, qui assure la transmission des requêtes vers l'intelligence artificielle (GPT-4 par exemple), puis retourne les réponses pour qu'elles soient affichées.

Backend (API IA)

Le backend sert de lien entre l'interface utilisateur et le modèle d'intelligence artificielle. Il gère les requêtes entrantes, applique la logique métier, et envoie les informations entre le front-end et l'IA. De plus, il interagit avec la base de données pour enregistrer les journaux d'utilisation et les feedbacks des utilisateurs.

Base de données (DB)

La base de données est utilisée pour stocker les informations de l'utilisateur, les journaux de conversation, ainsi que les retours d'expérience. Cela permet d'améliorer les interactions futures en offrant un historique et une analyse des données précédentes. Elle est directement liée au backend, qui enregistre automatiquement les logs et feedbacks.

Déploiement Cloud (CD)

Le déploiement sur le cloud permet d'héberger l'application et tous ses composants, tels que le backend, la base de données, et le modèle d'intelligence artificielle, sur des plateformes comme AWS ou Azure. Ce type de déploiement assure une évolutivité facile, une disponibilité élevée, et garantit que le système fonctionne de manière fluide et continue.

Système de feedback

Après chaque session de chat, un formulaire est proposé aux utilisateurs pour qu'ils puissent fournir des commentaires détaillés sur leur expérience. Ces retours sont transmis au backend, stockés dans la base de données, puis analysés par un modèle d'IA qui évalue automatiquement le niveau de satisfaction des utilisateurs.

3.2.2 Concepts

Frontend

1. Interface web responsive HTML/CSS/JS: Interface utilisateur traditionnelle pour assurer la compatibilité avec tous les navigateurs modernes.
2. Interface web responsive avec React, Angular ou Bootstrap: Interface utilisateur plus moderne, responsive pour s'adapter aux différents écrans.
3. Application Web Progressive: Concevoir le site web comme une PWA pour offrir une expérience utilisateur similaire à une application native. Cela permettrait aux utilisateurs d'installer l'application sur leur appareil et d'accéder au contenu hors ligne.
4. Interface accessible conforme aux normes WCAG: Créer une interface centrée sur l'accessibilité en respectant les directives WCAG 2.1 niveau AA. Cela inclut une navigation au clavier, des contrastes de couleurs appropriés, et une compatibilité avec les lecteurs d'écran.

Backend

1. Intégration avec une API d'IA externe: Utiliser une API existante (comme OpenAI GPT-4) pour gérer les fonctionnalités du chatbot. Cela réduit le temps de développement et assure des réponses de haute qualité.
2. Développement d'un modèle IA personnalisé: Entraîner un modèle d'IA spécifique aux besoins des utilisateurs cibles, en utilisant des données pertinentes pour améliorer la pertinence des réponses.
3. Architecture Microservices: Structurer le backend en microservices pour séparer les responsabilités (gestion du chatbot, authentification, API). Cela améliore la scalabilité et la maintenance du système.
4. Architecture Modulaire: L'application est développée et déployée comme une seule unité mais contient plusieurs modules. Tous les composants (interfaces utilisateur, logique métier, accès aux données) sont regroupés dans une seule application.

DB

1. Base de données relationnelle SQL: Utiliser une base de données SQL (comme MySQL ou PostgreSQL) pour stocker les données structurées, telles que les informations des utilisateurs et les sessions de chat.
2. Base de données NoSQL: Adopter une base de données NoSQL (comme MongoDB) pour stocker des données non structurées ou semi-structurées, comme les logs de conversation et les rétroactions.
3. Stockage sécurisé avec encryption: Mettre en place un chiffrement au repos pour toutes les données sensibles stockées, renforçant la sécurité et la conformité aux réglementations.

FS

1. Formulaire de rétroaction post-interaction: Après chaque session de chat, présenter un formulaire permettant aux utilisateurs de fournir des commentaires détaillés sur leur expérience.
2. Boutons de notation en temps réel: Intégrer des boutons pendant la conversation (par exemple, "Cette réponse a-t-elle été utile ? Oui/Non") pour recueillir des retours immédiats.
3. Analyse automatisée de la satisfaction: Utiliser l'analyse de sentiment sur les interactions et le formulaire de rétroaction pour évaluer automatiquement la satisfaction des utilisateurs. L'analyse de sentiment requiert un modèle d'IA.

CD

1. Hébergement sur une plateforme Cloud publique: Utiliser les services d'un fournisseur cloud public (comme AWS, Azure) pour bénéficier d'une infrastructure scalable et fiable, avec des services gérés pour le déploiement.
2. Déploiement local: Pour des raisons de sécurité ou de conformité, déployer l'application localement, contrôlant ainsi entièrement l'environnement.
3. Utilisation de conteneurs et orchestration: Conteneuriser les applications (Docker) pour faciliter le déploiement et la scalabilité, en utilisant Kubernetes pour l'orchestration des conteneurs.

3.2.3 Analyse et évaluation de concepts

Frontend

- Critères d'évaluation:
 - Accessibilité (WCAG)
 - Efficacité
 - Facilité d'utilisation
 - Disponibilité
 - Temps de développement

- Concept 1:
 - Accessibilité: Peut être optimisée, mais nécessite un effort supplémentaire pour atteindre la conformité WCAG.
 - Efficacité: Développement plus complexe puisque la structuration d'éléments est manuelle.
 - Facilité d'utilisation: Standard, mais peut manquer de fonctionnalités avancées.
 - Disponibilité : Accessible sur tous les navigateurs modernes.
 - Temps: Nécessite plus de temps de développement (pas d'utilisation de frameworks).

- Concept 2:
 - Accessibilité: Peut être optimisée, mais nécessite un effort supplémentaire pour atteindre la conformité WCAG.
 - Efficacité: Développement plus rapide, moins complexe.
 - Facilité d'utilisation: Facile à utiliser (possibilité de drag et drop).
 - Disponibilité : Accessible sur tous les navigateurs modernes.
 - Temps: Moins de temps de développement grâce aux ressources fournies.

- Concept 3:
 - Accessibilité: Peut être pleinement conforme aux normes WCAG avec un développement approprié.
 - Efficacité: Offre des performances améliorées, chargement rapide.
 - Facilité d'utilisation: Expérience utilisateur riche, semblable à une application native.
 - Disponibilité: Fonctionne hors ligne, installable sur les appareils.
 - Temps: Temps de développement plus élevé.

- Concept 4:
 - Accessibilité : Priorise l'accessibilité, essentiel pour les utilisateurs handicapés.
 - Efficacité : Peut nécessiter un temps de développement supplémentaire.
 - Facilité d'utilisation : Améliore l'expérience pour tous les utilisateurs.
 - Disponibilité : Accessible sur tous les dispositifs.

- Temps: Investissement supplémentaire pour tests et validations d'accessibilité.

Backend

- Critères d'évaluation:
 - Exactitude et pertinence des réponses
 - Fiabilité et efficacité
 - Sécurité des données
 - Coût et temps de développement
- Concept 1:
 - Exactitude : Haute, en utilisant des modèles éprouvés comme GPT-4.
 - Fiabilité : Dépend de la disponibilité du service externe.
 - Sécurité : Risque lié à la transmission de données à un tiers.
 - Coût : Coûts opérationnels récurrents (estimé à 0,06\$ par 1 000 tokens).
 - Temps de développement : Faible, intégration rapide.
- Concept 2:
 - Exactitude : Peut être optimisé pour le public cible.
 - Fiabilité : Contrôle total sur le modèle.
 - Sécurité : Données restent internes.
 - Coût : Élevé (coûts de développement et d'entraînement).
 - Temps de développement : Long (plusieurs mois).
- Concept 3:
 - Efficacité : Haute scalabilité et maintenabilité.
 - Complexité : Augmentée, nécessite une gestion avancée.
 - Temps de développement : Plus long en raison de la complexité.
- Concept 4:
 - Efficacité: Meilleures performances pour les applications de petite à moyenne envergure. Cependant, scalabilité, dépendance et évolution sont toutes des inconvénients potentiels si l'application n'est pas conçue de manière appropriée.
 - Complexité: Moins complexe à mettre en place et à gérer que les microservices, ce qui est avantageux pour une équipe avec des ressources limitées.
 - Temps de développement: Permet un développement plus rapide puisque la logique est centrée dans une région.

DB

- Critères d'évaluation:
 - Fiabilité et efficacité
 - Sécurité des données
 - Scalabilité
 - Coût et temps
- Concept 1:

- Fiabilité : Haute pour les transactions structurées.
- Efficacité : Moins flexible pour les données non structurées.
- Scalabilité : Verticale, limitée.
- Coût et temps: Modéré.
- Concept 2:
 - Fiabilité : Bonne pour les données non structurées.
 - Efficacité : Haute flexibilité.
 - Scalabilité : Horizontale, excellente.
 - Coût et temps: Modéré.
- Concept 3:
 - Sécurité : Protège les données sensibles.
 - Efficacité : Peut impacter légèrement les performances.
 - Coût et temps: Faible à modéré (implémentation du chiffrement).

FS

- Critères d'évaluation:
 - Facilité d'utilisation
 - Taux de participation des utilisateurs
 - Coût et temps de mise en œuvre
 - Efficacité des retours
- Concept 1:
 - Facilité d'utilisation : Commentaires généraux donc facile.
 - Taux de participation : Faible (~10%).
 - Coût et temps: Faible.
- Concept 2:
 - Facilité d'utilisation : Simple, intégré à l'expérience.
 - Taux de participation : Plus élevé (~30-40%).
 - Coût et temps: Faible.
- Concept 3:
 - Efficacité : Fournit des insights sans sollicitation directe.
 - Précision : Dépend de la qualité de l'algorithme.
 - Coût et temps: Moyen (développement de l'algorithme).

CD

- Critères d'évaluation:
 - Disponibilité

- Efficacité
- Coût et temps
- Sécurité
- Concept 1:
 - Disponibilité : Haute, infrastructure fiable.
 - Efficacité : Accès à des services avancés.
 - Coût et temps: Paiement à l'usage, économique pour les petits volumes, pas trop de temps requis.
 - Sécurité : Standards élevés, mais dépend du fournisseur.
- Concept 2:
 - Disponibilité : Contrôlable.
 - Efficacité : Personnalisable.
 - Coût et temps: Élevé (infrastructure propre).
 - Sécurité : Contrôle total.
- Concept 3:
 - Efficacité : Scalabilité et déploiement simplifiés.
 - Complexité : Requiert une expertise DevOps.
 - Coût et temps: Élevé.

Justification du processus et des méthodes utilisées:

- Chaque concept a été évalué en fonction des spécifications cibles, CPX et autres contraintes tels que le coût et temps de développement.
- Les concepts ont été comparés les uns aux autres pour identifier les plus adaptés pour le projet. Le choix de concept logiciels est difficile à quantifier. Donc, notre approche est de discuter à propos des avantages et désavantages de chaque concept.

3.2.4 Concept global

Pour le frontend, on utilisera une combinaison des concepts 2, 3 et 4: Une PWA multilingue développée à l'aide de React Mantine et se concentre sur les normes WCAG. Cette stratégie se centre sur l'accessibilité, permet une interface multilingue, offre une meilleure performance, une disponibilité accrue et une vue moderne, et facilite l'utilisation des composantes, malgré un temps de développement plus élevé.

Pour le backend, on utilisera une architecture modulaire et on intégrera l'application avec l'API de ChatGPT (Concepts 1 et 4). On utilisera Python pour le développement (langage de programmation) et Django comme web framework. On aura plusieurs couches (modules) qui organisera le backend.

Une couche de présentation qui gère la communication avec le frontend via des API RESTful. Une couche de logique métier (business logic) qui contient la logique principale de l'application, y compris l'intégration avec le chatbot IA. Une couche d'accès aux données qui gère les interactions avec la base de données. Cette approche simplifie le développement et le déploiement tout en répondant aux spécifications cibles du projet. Cette approche est pragmatique pour le stade actuel du projet, offrant une solution efficace pour fournir rapidement un service fiable et sécurisé aux utilisateurs. Elle permet également de poser des bases solides qui pourront être évoluées à l'avenir en fonction des besoins grandissants du projet.

Pour la DB, on utilisera une combinaison des concepts 2 et 3. Cette stratégie concerne une base de données NoSQL, spécifiquement MongoDB, pour sa flexibilité et scalabilité, avec chiffrement des données pour la sécurité. De plus, la structure des documents n'est pas nécessaire puisque la majorité des données seront non-structurées (demandes et réponses).

Pour le FS, on utilisera un formulaire de rétroaction ainsi que des boutons de notation, l'analyse des réponses pourrait être considérée si le temps permet (Concepts 1,2 et possiblement 3). Cela permet d'avoir de la rétroaction immédiate, de collecter des données pertinentes sans surcharger l'utilisateur et serait raisonnable pour le coût et temps alloués.

Pour le CD, on utilisera la plateforme cloud publique AWS avec conteneurs Docker pour le backend afin d'équilibrer efficacité, coût et contrôle (Concepts 1 et 3). Cela permet d'avoir une plateforme disponible, efficace et raisonnable en termes de coût.

3.2.5 Représentation visuelle

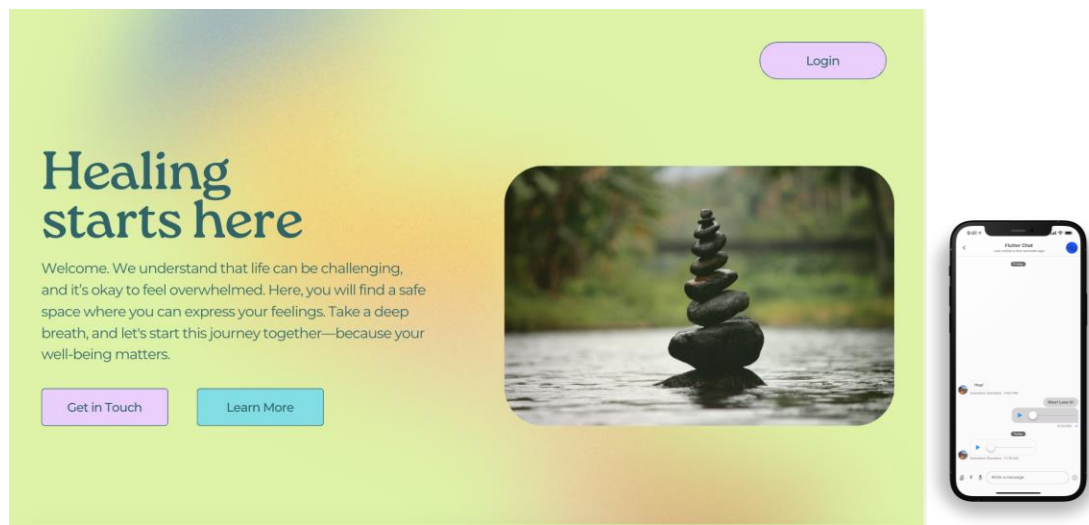


Figure 1 : Prototype du sous-concept d'un frontend

Figure 2 : Exemple de chatbox potentiel pour l'application

Notre concept adoptera un format similaire aux figures ci-dessus. L'utilisateur serait accueilli par une page de connexion afin que les ressources y soient personnalisées. Cependant, il serait toutefois possible d'accéder au chat sans de profile de l'utilisateur. En second lieu, l'utilisateur pourrait communiquer avec l'assistant IA sous forme d'une conversation qui pourrait être écrite ou bien à l'orale.

3.2.6 Rapport entre concept et spécifications cibles

Le concept global développé pour l'assistant d'IA répond aux spécifications cibles de plusieurs manières

1. **Accessibilité:** Conformité aux normes WCAG 2.1 niveau AA minimum pour assurer une utilisation facile pour tous les types d'utilisateurs.
2. **Langues:** Contient au moins Anglais, Français pour garantir une expérience bilingue.

3. **Temps de réponse:** Moins de 10 secondes pour s'assurer d'une interaction fluide, efficace et réduire l'aspect robotique.
4. **Sécurité:** Tokenization pour protéger les informations sensibles.
5. **Disponibilité:** La plateforme doit être accessible en tout temps, avec un objectif de disponibilité d'au moins 90% pour assurer que les utilisateurs puissent y accéder sans interruption.

Ces éléments montrent que le concept a été aligné avec les spécifications cibles pour maximiser l'efficacité, la sécurité et l'accessibilité.

3.2.7 Rapport entre concept et facteurs CPX

Le concept développé pour l'assistant d'IA prend en compte plusieurs facteurs CPX notamment:

1. **Accessibilité:** L'application est accessible aux personnes atteintes d'un handicap, répondant aux normes établies par la loi canadienne sur l'accessibilité concernant les sites Web.
2. **Support (Réconfort):** L'assistant est sensible aux besoins psychologiques de l'utilisateur.
3. **Sécurité:** L'application est sécuritaire, empêchant la fuite de données sensibles.
4. **Vitesse:** L'assistant donne une réponse selon un délai raisonnable, avec une réponse à la question posée en moins de 10 secondes.
5. **Facilité d'utilisation:** L'assistant permet d'être utilisé en posant des questions par voix et par écrit ainsi que répondre par audio ou par écrit.

Ces considérations montrent que le concept est aligné avec les attentes et les besoins identifiés dans le cadre des facteurs CPX, visant à créer une solution efficace et sensible aux besoins des utilisateurs.

3.3 Plan de projet

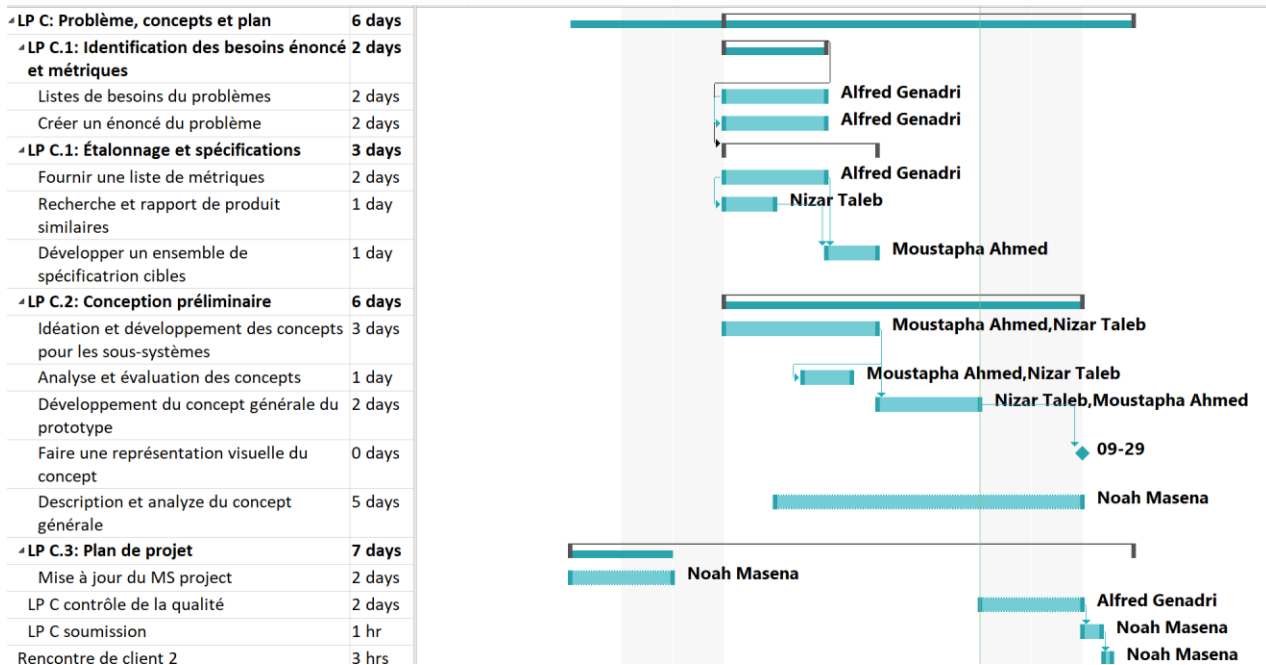


Figure 3 : Photo du plan de projet qui démontre les tâches à compléter et leurs dépendances du livrables C ainsi que la ou les personnes responsables de l'achèvement des tâches.



Figure 4 : Photo du plan du Livrable B mise à jour selon l'achèvement de la tâche

4 Conception détaillé et NDM

Notre deuxième rencontre avec le client a eu lieu le 30 octobre 2024. Le but de cette rencontre était de lui présenter les sous-concepts développés dans la livrable C afin de recevoir ses commentaires à propos de notre conception. De plus, nous avons profité de cette rencontre pour lui poser des questions sur les spécifications souhaitées du concept.

D'abord, selon les commentaires du client, un frontend similaire au 4ème exemple (*figure 5*) que nous lui avons présenté serait idéal pour le concept. Le client a exprimé une préférence pour un frontend simple, comportant plusieurs raccourcis vers des ressources spécifiques. De plus, il est de l'avis qu'inclure une fonctionnalité permettant de connecter un profil utilisateur, contenant des informations personnalisées, serait apprécié tant et aussi longtemps que nous pouvons garantir que l'information reste sécurisée. Par la suite, le client nous a fourni de l'information sur les couleurs appropriées (*figure 6*) pour notre concept. Enfin, il souhaite que l'application soit accessible à tous, quel que soit leur handicap physique. Par exemple, même une personne ayant un handicap visuel peut utiliser l'application sans difficulté.

En résumé, les rétroactions du client ont confirmé les facteurs que nous avons établis pour nos CPX qui influencent nos sous-concepts. Cette rencontre nous a permis de constater que notre conception est en ligne avec la vision du client, étant donné qu'il ne nous a pas fait une remarque négative à propos de notre présentation et de nos concepts. Nous pouvons donc avancer avec notre plan de conception sans apporter de modifications majeures à notre plan de conception.

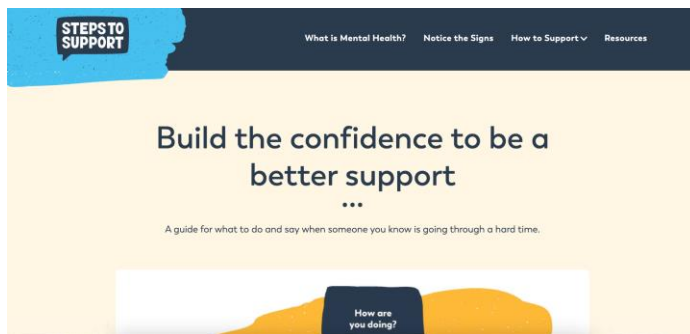


Figure 5 : 4ème exemple de frontend

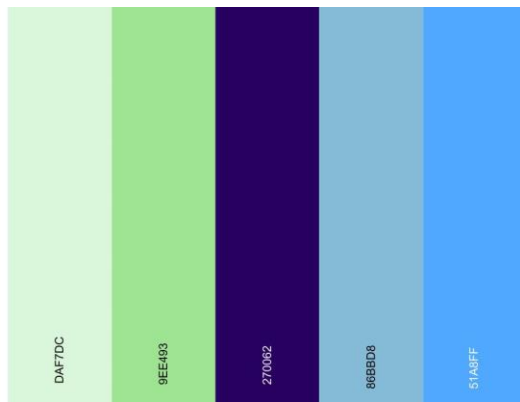


Figure 6 : Figure de la palette de couleur donnée par le client

4.1 Facteurs CPX

Pour respecter nos facteurs CPX identifiés (accessibilité, sécurité, scalabilité, et facilité d'utilisation), voici les principaux éléments à prendre en compte lors de la conception et du développement de notre assistant IA

- **Accessibilité**

Nous devons garantir que l'interface utilisateur est accessible à tous, y compris aux utilisateurs ayant des handicaps. Cela implique de suivre les directives WCAG 2.1, d'assurer une compatibilité avec les lecteurs d'écran et de proposer une navigation intuitive via clavier. La conception de l'interface doit intégrer ces aspects dès le début du projet pour éviter des refontes coûteuses plus tard. Nous inclurons une option de communication orale avec l'assistant IA (par microphone). Cela permet soit la communication orale ou écrite.

Importance

Ce facteur est crucial car l'assistant IA vise un public pas mal large (n'importe qui aura accès), y compris les personnes en situation de handicap. Si ce critère n'est pas respecté, cela pourrait limiter l'adoption du produit.

- **Sécurité des données**

Les informations partagées par les utilisateurs, notamment dans des situations d'urgence ou de besoin d'assistance, doivent être protégées avec les normes de sécurité actuelles. Nous devons intégrer dès le départ des mécanismes comme le cryptage des données sensibles (AES-256) et l'utilisation de protocoles sécurisés (TLS 1.2+) pour toutes les communications. Ces mécanismes de sécurité sont robustes et garantiront la sécurité des données (chiffrement, authentification).

Importance

C'est un facteur de haute priorité car un manquement à la sécurité pourrait avoir des conséquences graves, affectant non seulement la confidentialité des utilisateurs, mais aussi la confiance dans l'outil.

- **Scalabilité et performance**

Étant donné que notre assistant IA sera potentiellement utilisé par de nombreux utilisateurs, il est essentiel que le système soit capable de s'adapter aux fluctuations du trafic. Nous devons nous assurer que le cloud (AWS/Azure) est configuré pour supporter cette montée en charge sans perte de performance.

Importance

Bien que moins critique que l'accessibilité et la sécurité, la scalabilité garantit que le système puisse gérer un nombre croissant d'utilisateurs sans affecter la performance.

- **Facilité d'utilisation**

L'interface doit être intuitive et facile à prendre en main. Nous devons concevoir une interface claire, avec des instructions simples, pour que même les utilisateurs ayant peu d'expérience avec les technologies puissent interagir avec l'assistant sans difficulté. Il est important d'effectuer des tests utilisateurs dès les premières étapes de développement.

Importance

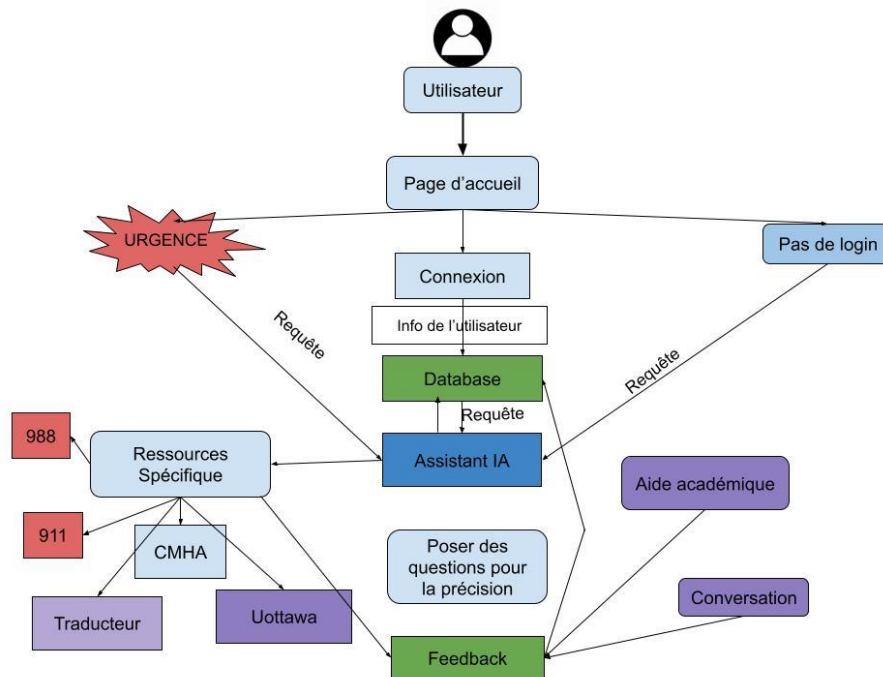
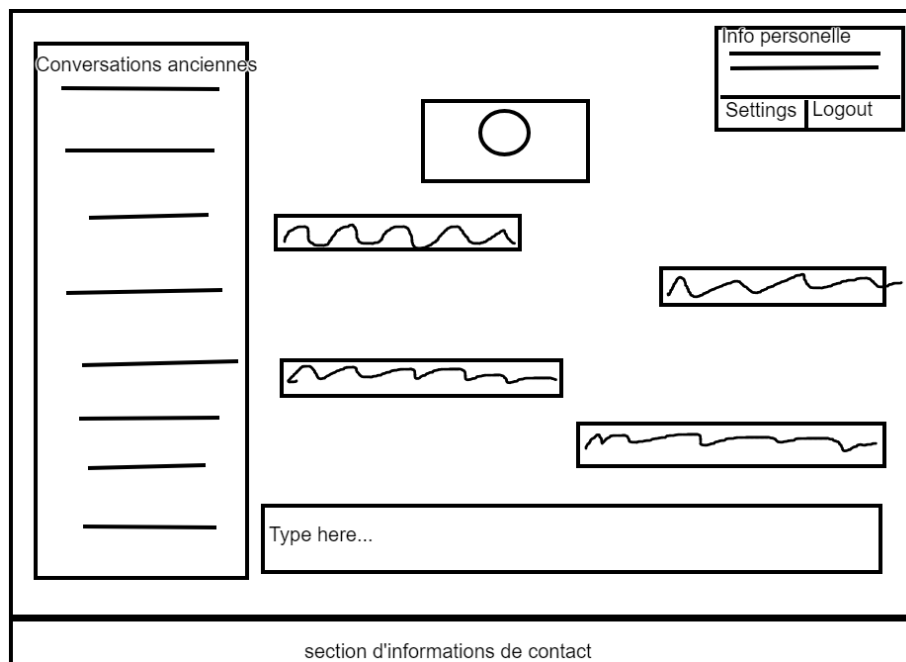
Bien que ce facteur soit important pour améliorer l'expérience utilisateur, il est légèrement moins prioritaire que l'accessibilité et la sécurité dans le contexte de notre projet. Cependant, il reste indispensable pour garantir l'adoption de l'outil.

Certains facteurs sont-ils plus importants que d'autres ? Pourquoi ?

Oui, certains facteurs sont plus importants que d'autres. **L'accessibilité et la sécurité** sont les plus critiques, car elles touchent directement à l'inclusivité de notre public cible et à la protection des informations sensibles des utilisateurs. Si ces deux éléments ne sont pas bien pris en compte, l'assistant IA risquerait de ne pas remplir son rôle ou de compromettre la confiance des utilisateurs.

La scalabilité et la facilité d'utilisation, bien qu'importantes pour la satisfaction et l'expérience utilisateur, viennent après. L'assistant doit d'abord être accessible et sécurisé avant d'être optimisé pour de grandes échelles ou une interface plus intuitive.

4.2 Conception détaillé



Le premier diagramme montre l'interface utilisateur, où les utilisateurs peuvent consulter leurs anciennes conversations, taper des messages et accéder aux paramètres personnels. Le second

organigramme illustre l'interaction entre les différents composants du système : de la page d'accueil à l'assistant IA, avec des liens vers des ressources externes comme le 911 ou le CMHA.

L'application sera divisée en plusieurs parties, chacune ayant une fonction majeure.

L'interface utilisateur sera développée avec Webflow et React Mantine afin de garantir une expérience utilisateur fluide tout en respectant les normes WCAG. Ces outils sont efficaces dans le contexte de notre projet puisqu'ils s'agissent d'un bon équilibre entre complexité et personnalisation.

Django est utilisé pour l'arrière-plan (backend) et contient des mécanismes internes (middleware) pour la sécurité, la gestion d'utilisateurs. Cet outil rend le développement d'un API intuitif, fluide et facile avec son architecture Model-View-Template.

L'assistant IA (chatbot) sera une aggrégation de Rasa et de ChatGPT (CustomGPT) afin d'avoir le meilleur des deux mondes (gestion de session et de contexte de Rasa et le corpus énorme de ChatGPT).

PostgreSQL serait la base de données utilisée dans le projet grâce à son intégration facile avec Django et sa nature SQL qui permet d'avoir un schéma définit des données stockées,

4.3 Compétences et ressources disponibles

- **Compétences disponibles**

Développement Web (Frontend et Backend)

Nous sommes familiers avec les technologies de développement telles que **React JS** pour le frontend, et **Django** pour le backend. Bien que nous ne soyons pas des experts, nous avons une bonne base. Webflow est une application de développement de site web avec un mécanisme Drag-and-Drop qui est facile à utiliser.

API et Modèles d'IA

Nous n'avons pas une base de compréhension de l'intégration des modèles d'IA via des **API** comme **GPT-4 et Rasa**, que nous comptons utiliser pour le traitement du langage naturel. On va lire la documentation d'OpenAPI pour l'intégration d'un API dans notre application. Des exemples sont fournis en ligne et ceux-là nous permettrons de concrétiser cet aspect d'intégration IA.

Base de données

Nous commençons à nous familiariser avec **PostgreSQL**, une base de données SQL flexible, que nous avons choisi pour stocker les données des utilisateurs, les conversations avec l'assistant IA et la rétroaction des utilisateurs.

Déploiement Cloud

Nous nous apprêtons à créer un compte **AWS** pour déployer notre projet sur le cloud. Bien que nous ne soyons pas encore familiers avec cette plateforme, nous avons prévu d'apprendre à l'utiliser pour garantir l'évolutivité et la disponibilité de notre application. AWS offre plusieurs cours gratuits et une documentation riche qui permettra une acquisition de connaissances rapide et fluide.

- **Ressources disponibles**

Accès à l'API GPT-4

Nous avons accès à l'API **GPT-4** via OpenAI, que nous intégrerons au backend pour générer des réponses en temps réel.

Infrastructure AWS

Nous allons créer un compte **AWS** et utiliser cette infrastructure pour héberger l'application, avec des instances pour le backend et un service de stockage pour la base de données.

- **Compétences manquantes**

PostgreSQL, AWS, Rasa

Nous devons nous familiariser davantage avec PostgreSQL et **AWS**. Cela comprend l'apprentissage des bases de données SQL et des pratiques de déploiement sur AWS. Rasa offre une multitude de fonctionnalités pour leur assistant IA et une exploration de ceux-ci est nécessaire afin d'avoir un chatbot efficace et personnalisé.

Accessibilité

Bien que nous comprenions l'importance des normes **WCAG 2.1**, nous ne sommes pas encore experts en accessibilité et pourrions avoir besoin de ressources supplémentaires pour garantir une conformité totale. Un membre du groupe lira les normes WCAG 2.1 et actera comme juge d'accessibilité lors du développement de l'interface utilisateur.

Sécurité des Données

Nous avons une bonne base pour mettre en place des mesures de sécurité comme le cryptage, mais un audit de sécurité complet pourrait nécessiter une expertise supplémentaire.

4.4 Estimation du temps et disponibilité

Nous avons ajusté notre plan pour terminer à temps. La phase de conception commence maintenant et se terminera d'ici la première semaine d'octobre. Pendant ce temps, nous allons définir les architectures du frontend et du backend, et planifier l'intégration de l'API GPT-4 et Rasa.

Ensuite, du milieu à la fin octobre, nous développerons en parallèle le backend (Django et PostgreSQL) et l'interface utilisateur (React JS et Webflow), tout en intégrant les algorithmes d'IA (API OpenAI et Rasa). Les tests unitaires seront faits au fur et à mesure pour assurer la qualité du code.

Début novembre, nous sécuriserons l'application en intégrant le chiffrement et l'authentification, puis passerons aux tests fonctionnels et d'intégration pendant deux semaines. Cela nous permettra d'identifier et de corriger d'éventuels bugs.

Enfin, la mise en production est prévue pour la troisième semaine de novembre, avec un peu de marge pour régler les derniers détails avant la date limite. Chaque membre de l'équipe pourra y consacrer 10 à 15 heures par semaine, avec des réunions hebdomadaires pour suivre l'avancement et ajuster si besoin.

4.5 Hypothèses critiques

Voici plusieurs hypothèses clés qui pourraient influencer notre capacité à livrer l'assistant IA

- **Disponibilité de l'API GPT-4**

Nous partons du principe que l'API GPT-4 restera accessible tout au long du projet. Si jamais l'API devenait inaccessible ou si des modifications imprévues survenaient, cela pourrait impacter le traitement des requêtes IA en temps réel.

- **Disponibilité de l'infrastructure AWS**

Nous comptons sur les services AWS pour l'hébergement et le stockage des données. Si des interruptions de service ou des changements tarifaires se produisaient, cela pourrait affecter la mise en production du projet.

- **Conformité aux normes de sécurité et d'accessibilité**

Nous supposons que nous serons en mesure de respecter les normes de sécurité (comme TLS et AES-256) et les directives WCAG 2.1 pour l'accessibilité. Si leur implémentation s'avère plus complexe que prévu, cela pourrait ralentir notre progression.

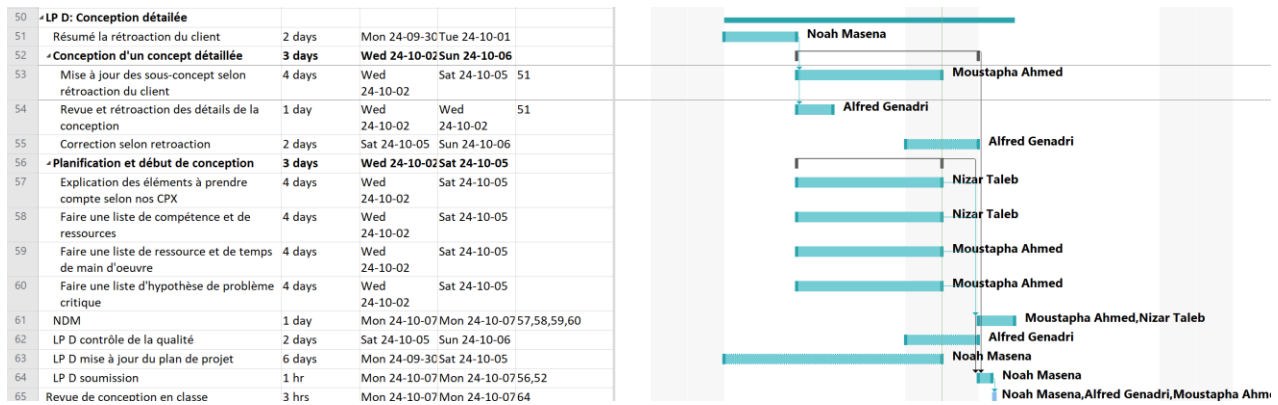
- **Compétences de l'équipe**

Nous prévoyons d'apprendre rapidement à utiliser **PostgreSQL**, **AWS**, **Rasa** et **l'intégration avec OpenAPI**. Si nous rencontrons des difficultés dans l'apprentissage ou la mise en place de ces technologies, cela pourrait affecter notre calendrier.

4.6 NDM

Composant	Description	Prix	Lien
API GPT-4	L'Assistant et NLP	20\$/mois USD	Openai.com/api/
Instance AWS EC2	Serveur backend	0\$	AWS EC2
PostgreSQL	Pour le stockage de données	0\$	PostgreSQL
React JS	Interface frontend	0\$	React JS
JWT Library	Authentification	0\$	JWT Library
Chiffrement AES-256	Sécurisation données	0\$	AES-25
Django	Gestion requêtes	0\$	Django
Python	Langage de programmation	0\$	Python

4.7 Plan de Projet



5 Conclusions

Au terme du développement de notre projet d'assistant d'IA, plusieurs leçons importantes ont été tirées. Tout d'abord, nous avons appris que la conception d'une application destinée à un public vulnérable, tel que des individus en situation de handicap ou en détresse psychologique, nécessite une attention particulière à l'accessibilité et à la sécurité des données. Ces aspects ont dominé nos décisions de conception, tout en répondant aux attentes des clients en matière de personnalisation et de performance.

Nous avons également constaté que l'intégration de technologies avancées telles que Rasa et GPT-4 dans un environnement cloud (AWS) impose des défis en termes de gestion des ressources et d'optimisation de la consommation énergétique, comme souligné dans notre analyse de développement durable. Néanmoins, la flexibilité offerte par ces outils nous permet de concevoir une solution scalable et adaptable à des besoins variés.

En termes de conception pour l'accessibilité, nous avons pu nous conformer aux directives WCAG 2.1 niveau AA, garantissant que notre produit soit inclusif et puisse être utilisé par des personnes ayant divers handicaps. De plus, nous avons développé une interface intuitive et multilingue pour favoriser l'engagement des utilisateurs tout en respectant la confidentialité de leurs données grâce à des mesures de sécurité robustes comme le chiffrement AES-256.

En conclusion, nous avons relevé les défis posés par le développement de cette application tout en respectant les contraintes CPX identifiées. Cependant, il reste quelques questions ouvertes, notamment sur l'évolutivité à grande échelle et la gestion à long terme des coûts énergétiques liés aux infrastructures cloud. Ces éléments devront être surveillés et ajustés au fur et à mesure de l'évolution de l'application pour garantir une solution durable et accessible à tous.

6 Bibliographie

Insérer votre liste de références ici.