



République Tunisienne
Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique
Université de Kairouan
Institut Supérieur d'Informatique et de Gestion
de Kairouan



Projet de Fin d'Études

En vue de l'obtention du Diplôme de
Licence en Informatique de Gestion
Parcours : Business Intelligence

Application de planification de voyage alimentée par l'IA

Elaboré par :

Mohamed Massous & Seif Eddine Thairi

Encadré par :

M. Houssem Eddine Enouri

Encadrant Académique

Mme. Oumayma Ben Afia

Encadrant Professionnel

Année Universitaire 2024-2025

Dédicace

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,

À Mes Très Chers Parents

Que ce travail soit l'expression de ma reconnaissance pour vos sacrifices consentis, votre soutien moral et matériel que vous n'avez cesse de prodiguer. Vous avez tout fait pour mon bonheur et ma réussite.

Que Dieu vous préserve en bonne santé et vous accorde une longue vie.

À Mon Frère et Mes sœurs

Vous étiez toujours présents pour m'aider et m'encourager. Je vous souhaite une vie pleine de bonheur et de succès et que Dieu vous protège et vous garde.

À Tous mes amis...

Seif Eddine Thairi

J'aimerai dédier ce rapport,

À mes Parents,

qui m'ont comblé de leur soutien et m'ont voué un amour inconditionnel. Vous êtes pour moi un exemple de courage et de sacrifice continu.

Que cet humble travail témoigne mon affection, mon éternel attachement et qu'il appelle sur moi vos continuelles bénédictions.

À mes frères et sœurs,

pour leurs compréhensions, leurs soutiens, leurs tendresses... Qu'ils trouvent ici l'expression de ma reconnaissance et le témoignage de ma gratitude ressentie.

À ceux qui m'aiment,

Que ce modeste travail vous honore et vous témoigne mes reconnaissances.

Puisse cette étude apporter la pleine satisfaction à tous ceux qui la lisent...

Mille Merci.

Mohamed Massous

Remerciements

À la fin de ce travail, nous souhaitons exprimer notre profonde gratitude à toutes les personnes qui nous ont soutenus et accompagnés tout au long de la réalisation de ce projet de fin d'études.

Nous tenons tout d'abord à remercier Monsieur Houssem Eddine Enouri pour ses précieuses idées, ses conseils avisés ainsi que pour ses explications claires et pertinentes, qui ont grandement contribué à l'avancement de notre projet.

Nous adressons également nos sincères remerciements à toute l'équipe d'Elyos Digital pour leur accueil chaleureux, leur disponibilité et les excellentes conditions de travail qu'ils nous ont offertes durant notre stage. Un remerciement tout particulier à Madame Oumayma Ben Afia pour son encadrement, ses encouragements constants, ses remarques constructives et ses conseils judicieux qui nous ont été d'une grande aide.

Nous remercions également les membres du jury pour le temps qu'ils ont consacré à lire, analyser et évaluer notre travail. Nous espérons que ce projet saura répondre à leurs attentes et refléter notre sérieux et notre implication.

Enfin, nous exprimons toute notre reconnaissance à nos enseignants pour la qualité de la formation qu'ils nous ont dispensée tout au long de notre parcours universitaire, ainsi que pour leur engagement à partager leurs savoirs avec passion et rigueur.

Table des matières

Introduction générale	1
1 Cadre du projet	4
1.1 Introduction	4
1.2 Présentation de l'organisme d'accueil	4
1.2.1 Description de l'entreprise	4
1.2.2 Missions effectuées	6
1.3 Présentation du sujet	6
1.3.1 Problématique	7
1.3.2 Étude de l'existant	7
1.3.3 Critique de l'existant	8
1.3.4 Solution proposée	8
1.4 Méthodologie Adoptée	9
1.4.1 Méthodologie Agile	9
1.4.2 Choix et justification de méthodologie	10
1.4.3 Démarche scrum	11
1.5 Conclusion	13
2 Sprint 0 : Spécification des besoins	14
2.1 Introduction	14
2.2 Collecte des besoins	14
2.2.1 Besoins fonctionnels	14
2.2.2 Besoins non fonctionnels	15
2.3 Diagramme de cas d'utilisation global	15
2.3.1 Identification des acteurs	15
2.3.2 Modélisation du diagramme	16
2.4 Diagramme de classes	16
2.5 Product Backlog	18
2.5.1 Définition des sprints	18
2.5.2 Planification des sprints	19

TABLE DES MATIÈRES

2.5.3	Diagramme de Gantt	20
2.6	Environnement de développement	20
2.6.1	Environnement matériel	21
2.6.2	Outil logiciel	21
2.6.3	Frameworks et langage de programmation	22
2.7	Architecture	23
2.7.1	Architecture technique	23
2.7.2	Architecture logique	24
2.7.3	Diagramme de déploiement	24
2.8	Conclusion	25
3	Sprint 1 : Authentification et gestion des utilisateurs	26
3.1	Introduction	26
3.2	Analyse	27
3.2.1	Raffinement de cas d'utilisation « S'inscrire »	27
3.2.2	Raffinement de cas d'utilisation « S'authentifier »	28
3.2.3	Raffinement des cas d'utilisations «Gérer les clients»	30
3.3	Conception	32
3.3.1	Diagramme d'activité «Gérer les clients»	32
3.3.2	Diagramme de séquence «S'authentifier»	33
3.3.3	Diagramme de séquence «Modifier client»	33
3.3.4	Diagramme de séquence «Annuler client»	34
3.3.5	Schéma relationnel	34
3.4	Réalisation	35
3.4.1	Les interface de bienvenue	35
3.4.2	Les interfaces de s'inscrire et de s'authentifier	36
3.5	Conclusion	36
4	Sprint 2 : Recherche de voyage alimentée par l'IA	38
4.1	Introduction	38
4.2	Sprint Backlog	38
4.3	Analyse	39
4.3.1	Raffinement de cas d'utilisation «Gérer voyage»	39
4.4	Conception	41
4.4.1	Diagramme d'activité «Chercher voyage»	41
4.4.2	Diagramme d'activité «Réserver voyage»	42
4.4.3	Diagramme de séquence «Chercher voyage»	43
4.4.4	Diagramme de séquence «Réserver voyage»	44
4.4.5	Schéma relationnel	44
4.5	Réalisation	45
4.5.1	Les interface d'accueil	45

TABLE DES MATIÈRES

4.5.2	Les interfaces d'organisation de voyage	46
4.5.3	Les interfaces du budget	48
4.5.4	Les interfaces de la Omra et du Hajj	49
4.6	Conclusion	52
5	Sprint 3 : Validation des voyages et consultation du tableau de bord	53
5.1	Introduction	53
5.2	Sprint Backlog	53
5.3	Analyse	54
5.3.1	Raffinement de cas d'utilisation « Valider les Voyages »	54
5.3.2	Raffinement de cas d'utilisation « Consulter les tableaux de bord »	56
5.4	Conception	57
5.4.1	Diagramme d'activité « Consulter les tableaux de bord»	57
5.4.2	Diagramme de séquence « Consulter les tableaux de bord»	58
5.4.3	Diagramme de séquence « Accepter voyage »	58
5.4.4	Schéma relationnel	58
5.5	Réalisation	59
5.5.1	Interface de page d'accueil du tableau de bord administrateur	59
5.5.2	Interface de gestion des clients	59
5.6	Conclusion	59
Conclusion générale		60
Bibliographie		61
Annexe		63

Table des figures

1.1	Structure de l'agence	5
1.2	L'équipe de la société	5
1.3	Client en Tunisie	5
1.4	Client en France - Afrique	6
1.5	Logo WonderPlan IA	7
1.6	Logo TripPlanner IA	8
1.7	Solution proposée	9
1.8	Méthodologie Agile	10
1.9	Présente un résumé du processus de la méthodologie Scrum	13
2.1	Diagramme de cas d'utilisation global	16
2.2	Diagramme de classes	17
2.3	Planning prévisionnel détaillé	20
2.4	Architecture technique	24
2.5	Architecture logique	24
2.6	Diagramme de déploiement	25
3.1	Raffinement de cas d'utilisation « S'inscrire »	27
3.2	Raffinement de cas d'utilisation « S'authentifier »	28
3.3	Raffinement de cas d'utilisation «Gérer les clients»	30
3.4	Diagramme d'activité «Gérer les clients»	32
3.5	Diagramme de séquence «S'authentifier»	33
3.6	Diagramme de séquence «Modifier client»	33
3.7	Diagramme de séquence «Annuler client»	34
3.8	Interface splash	35
3.9	Interface de bienvenue	35
3.10	Interface création de compte	36
3.11	Interface de s'authentifier	36
4.1	Raffinement de cas d'utilisation «Gérer voyage»	39
4.2	Diagramme d'activité «Chercher voyage»	41

TABLE DES FIGURES

4.3	Diagramme d'activité «Chercher voyage»	42
4.4	Diagramme de séquence «Chercher voyage»	43
4.5	Diagramme de séquence «Réserver voyage»	44
4.6	Interface de page d'accueil	45
4.7	Interface de chercher par message libre	45
4.8	Interface de chercher par l'agent llama	45
4.9	Interface de consultation des destinations	46
4.10	Interface de réservation d'hôtel	47
4.11	Interface de réservation d'un vol	47
4.12	Interface d'estimation du budget	48
4.13	Interface de suivi des dépenses	48
4.14	Interface de planification du Hajj et Omra	49
4.15	organisation de voyage Omra	50
4.16	Interface de confirmation de réservation	51
4.17	Interface de suivi budgétaire	51
4.18	Interface de profile	52
5.1	Raffinement de cas d'utilisation «Valider voyage»	54
5.2	Raffinement de cas d'utilisation «Consulter les tableaux de bord»	56
5.3	Diagramme d'activité «Consulter les tableaux de bord»	57
5.4	Diagramme de séquence « Consulter les tableaux de bord»	58
5.5	Diagramme de séquence « Accepter voyage »	58
5.6	Interface de page d'accueil du tableau de bord administrateur	59
5.7	Interface de gestion des clients	59
8	Chargement et Préparation des Données	65
9	Détermination de la saison en fonction de la date	65
10	Exemple de Fonction principale ' Génération de Recommandations '	65
11	Exemple de dataset Restaurant	66
12	Exemple de dataset Hotel	66

Liste des tableaux

1.1	Comparaison entre différentes méthodologies de développement logiciel	10
1.2	Répartition des rôles SCRUM au sein de l'équipe Easy Travel	12
2.1	Description des classes dans le diagramme de classes	18
2.2	Product Backlog	19
2.4	Outils de développement	22
2.5	Frameworks and development languages	23
3.1	Sprint Backlog détaillé – Sprint 1	26
3.2	Description Textuelle de Cas d'utilisation « S'inscrire »	28
3.3	Raffinement de cas d'utilisation « S'authentifier »	29
3.4	Description Textuelle de Cas d'utilisation « Gérer les clients »	31
4.1	Sprint Backlog détaillé – Sprint 2	39
4.2	Description du cas d'utilisation <i>Gérer voyage</i>	39
5.1	Sprint Backlog détaillé – Sprint 3	53
5.2	Description Textuelle de Cas d'utilisation « Valider les Voyages »	55
5.3	Description Textuelle de Cas d'utilisation « Consulter les tableaux de bord » . .	57

Introduction générale

L'informatique joue aujourd'hui un rôle central dans notre quotidien : elle facilite la communication, le travail, l'apprentissage, les déplacements, et bien plus encore. Grâce à des technologies toujours plus performantes, une multitude d'outils numériques sont désormais accessibles, permettant de planifier nos voyages, qu'ils soient professionnels ou personnels. Dans ce contexte d'évolution numérique, le web et les applications mobiles offrent de nouvelles façons d'inter agir, de découvrir des destinations, d'organiser ses séjours et de partager ses expériences avec d'autres utilisateurs.

Cette transformation numérique modifie profondément notre manière de voyager. Les voyageurs d'aujourd'hui recherchent des solutions simples, rapides et personnalisées pour organiser leurs déplacements. Ils souhaitent accéder à des informations fiables, découvrir des lieux authentiques, et bénéficier de conseils adaptés à leurs goûts. C'est dans cette optique que les applications mobiles deviennent des outils indispensables, en combinant accessibilité, interactivité et richesse de contenu.

Dans le cadre de ce projet, nous avons conçu et développé une application mobile intitulée Easy Travel. Cette application permettra aux utilisateurs de rechercher des destinations, de consulter des guides de voyage, d'enregistrer leurs itinéraires et de partager leurs expériences avec la communauté. Parmi ses fonctionnalités clés, on retrouve la gestion des profils utilisateurs, la recommandation de lieux touristiques ainsi que la planification d'itinéraires personnalisés.

— Chapitre 1 : Introduction générale

Nous exposons le contexte général du sujet proposé en présentant en détail la problématique principale ainsi que les solutions envisagées. Pour conclure, ce chapitre décrira la méthodologie de travail adoptée, justifiée par les raisons ayant motivé ce choix.

— Sprint 0 : Spécification des besoins

Dans ce sprint, nous spécifions les besoins, planifions les tâches, définissons l'environnement de développement et posons l'architecture pour préparer l'analyse et la conception détaillées.

— Sprint 1 : Authentification et gestion des utilisateurs

Dans ce sprint, nous concevons et réalisons l'authentification, la gestion des clients, les

interfaces utilisateur et les schémas de données pour établir les fondations de l'application.

— **Sprint 2 : Recherche de voyage alimentée par l'IA**

Dans ce sprint, nous concevons et réalisons des interfaces intuitives de recherche, réservation et gestion budgétaire pour permettre aux utilisateurs d'organiser facilement leurs voyages via l'IA.

— **Sprint 3 : Validation des voyages et consultation du tableau de bord**

Dans ce sprint, nous développons une interface administrateur pour gérer les utilisateurs et les réservations. Grâce à notre travail, nous validons les voyages et consultons facilement le tableau de bord.

Enfin, nous présenterons une synthèse des éléments abordés dans les chapitres précédents, mettant en lumière les perspectives et les orientations futures de notre projet.

Objectif du projet

Dans le cadre de ce projet, nous concevrons et développerons une application mobile intitulée **Easy Travel**. Cette application vise à accompagner les utilisateurs dans la planification complète de leurs voyages, de manière simple, rapide et personnalisée.

Le parcours utilisateur typique dans l'application comprend les étapes suivantes :

- Sélection de ses **préférences de voyage** (type de destination, budget, centres d'intérêt, etc.),
- Choix d'un **forfait de voyage** adapté,
- Réservation de l'**hébergement**, des **restaurants** et des **vols**,
- Consultation des **activités disponibles** sur place,
- Estimation et affichage du **budget global** pour le séjour.

Parmi les autres fonctionnalités clés, on retrouve :

- La **gestion des profils utilisateurs**,
- La **recommandation intelligente** de lieux touristiques selon les préférences,

Enjeux techniques

Un des défis majeurs de ce projet réside dans l'élaboration d'une **navigation fluide et intuitive**, tout en assurant la **sécurité des données personnelles**. Il est fondamental de protéger les informations sensibles des utilisateurs contre toute utilisation abusive.

Objectif global

Ce projet a pour but de démontrer comment une application mobile peut répondre efficacement aux besoins des voyageurs, en leur proposant un outil à la fois **pratique, interactif et sécurisé**. Il mettra également en évidence :

- Les **bonnes pratiques** de développement mobile
- L'importance de l'**expérience utilisateur**
- La **protection des données personnelles**.

Cadre du projet

1.1 Introduction

Ce rapport présente un projet de fin d'études. Ce projet a été réalisé dans l'entreprise Elyos Digital, une agence web qui travaille en Tunisie, en France et en Afrique.

Le but du projet est de créer une application mobile pour aider les gens à organiser leurs voyages. L'application propose des destinations, des activités, un budget, et aide à faire des réservations. Elle utilise des technologies modernes et l'intelligence artificielle.

Dans ce rapport, nous allons présenter l'entreprise, le sujet du projet, le problème à résoudre, les solutions existantes, la solution proposée, les objectifs et la méthode utilisée pour réaliser ce travail.

1.2 Présentation de l'organisme d'accueil

Dans cette partie, nous allons présenter l'entreprise qui nous a accueilli pour notre stage, ainsi que les missions réalisées pendant cette période.

1.2.1 Description de l'entreprise

Ce projet a été réalisé dans l'entreprise **Elyos Digital**, une agence web fondée en 2011. Elle est présente en Tunisie et en France. Depuis sa création, Elyos Digital a développé plusieurs services comme le design, le développement web, le marketing digital, le référencement et le conseil.

L'agence travaille avec des clients dans différents domaines : industrie, santé, mode, tourisme, éducation, finance, gouvernement, art, etc. Elle propose des solutions globales pour aider ses clients à réussir leur transformation numérique.

Elyos Digital est composée de trois sociétés et deux marques. Son équipe est jeune, dynamique et bien organisée. Elle collabore avec des clients en Tunisie, mais aussi en France et en Afrique.



FIGURE 1.1 – Structure de l’agence



FIGURE 1.2 – L’équipe de la société



FIGURE 1.3 – Client en Tunisie



FIGURE 1.4 – Client en France - Afrique

1.2.2 Missions effectuées

Nous avons choisi d'effectuer notre stage au sein de l'agence **Elyos Digital**, un choix motivé par sa solide réputation dans le domaine du développement web, du marketing digital et de la transformation numérique, en Tunisie et à l'étranger.

Depuis sa création, Elyos Digital propose des services modernes et variés, comme le conseil en stratégie digitale, le design, le développement web et mobile, le référencement, la gestion des réseaux sociaux et le marketing digital. Grâce à son expertise, l'agence accompagne plusieurs clients dans différents secteurs et leur propose des solutions innovantes et sur mesure.

Pendant notre stage, nous avons eu l'opportunité de participer à plusieurs missions intéressantes. Les échanges avec les membres des différentes équipes nous ont permis de mieux comprendre l'organisation de l'agence. Nous avons d'abord découvert le fonctionnement général, puis nous avons intégré l'équipe technique, où nous avons contribué au développement d'un projet web.

1.3 Présentation du sujet

Dans ce premier chapitre, nous avons présenté notre projet de fin d'études ainsi que le contexte de sa création. Ce projet consiste en le développement d'une application mobile innovante dédiée à la planification de voyages.

L'application permettra aux utilisateurs de découvrir des destinations adaptées à leurs centres d'intérêt, de générer des itinéraires optimisés, de recevoir des suggestions d'activités locales et de gérer leur budget de voyage de manière efficace. Elle intégrera également un outil facilitant la réservation de vols, d'hébergements et de transports en comparant les offres pour proposer les meilleures options.

Pour réaliser ce projet, nous utiliserons des technologies modernes telles que **Flutter** pour le développement mobile, **VueJS** pour l'interface web et **Laravel** pour la gestion du backend.

Ce projet vise à simplifier la planification des voyages en offrant une solution complète et intuitive. Dans les prochains chapitres, nous détaillerons les différentes étapes de son développement ainsi que ses fonctionnalités principales.

1.3.1 Problématique

Planifier un voyage peut être une tâche complexe et chronophage, nécessitant la recherche de destinations, l'optimisation des itinéraires, la gestion du budget et la réservation de divers services. Chaque voyageur a des attentes spécifiques en fonction de ses centres d'intérêt, de son budget et de la durée de son séjour, rendant la planification encore plus difficile.

Aujourd'hui, bien que de nombreuses plateformes offrent des solutions partielles (réservation d'hôtels, billets d'avion, guides touristiques), il manque une solution centralisée et intelligente qui prenne en compte les préférences de l'utilisateur pour proposer un itinéraire personnalisé et optimisé.

Dès lors, comment développer une application mobile qui utilise l'intelligence artificielle pour automatiser et personnaliser la planification des voyages en tenant compte des préférences, des contraintes budgétaires et des meilleures offres disponibles ?

Cette question guidera notre travail et nous permettra de concevoir une solution innovante répondant aux besoins des voyageurs.

1.3.2 Étude de l'existant

L'étude de l'existant est une phase importante pour comprendre les applications actuelles sur le marché, et déterminer les points forts pour terminer après par une critique. Nous avons trouvé plusieurs applications répondant aux besoins de planification des voyages. Parmi ces applications, nous citons les deux meilleures solutions du marché.

WonderPlan AI

WonderPlan est un outil gratuit qui aide à planifier des voyages. Il utilise l'intelligence artificielle pour créer des itinéraires personnalisés selon vos goûts et votre budget. Vous pouvez modifier votre plan, ajouter ou supprimer des étapes. Il est aussi possible de télécharger l'itinéraire en PDF pour une utilisation hors ligne.

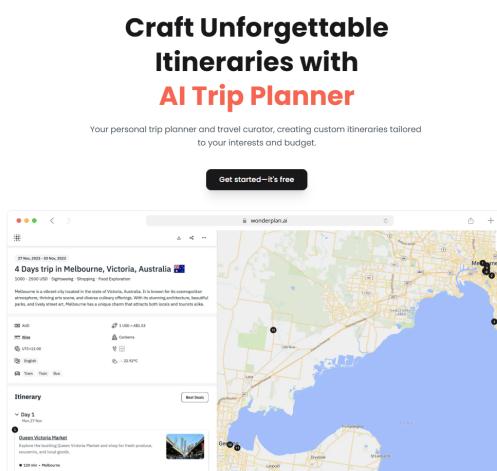


FIGURE 1.5 – Logo WonderPlan IA

TripPlanner.AI

TripPlanner.ai est un site gratuit qui aide à organiser des voyages. Il crée des itinéraires personnalisés selon vos goûts, budget et durée du séjour. L'outil propose des activités, restaurants et hébergements adaptés. Il optimise les trajets et permet de modifier facilement votre programme. De plus, il s'inspire des réseaux sociaux pour suggérer des lieux populaires. C'est un assistant pratique pour planifier vos vacances sans stress.

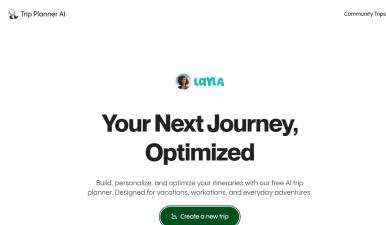


FIGURE 1.6 – Logo TripPlanner IA

1.3.3 Critique de l'existant

Après une analyse approfondie des plateformes **Wonderplan.ai** et **Tripplanner.ai**, nous avons identifié les forces et faiblesses suivantes :

Points forts

- **Interface intuitive** : Conception ergonomique facilitant la prise en main.
- **Rapidité de planification** : Génération d'itinéraires en temps réel.
- **Suggestions automatisées** : Recommandations pertinentes basées sur des algorithmes.

Limites actuelles

- **Coûts élevés** :
 - Fonctionnalités premium payantes
 - Frein à l'adoption pour les utilisateurs occasionnels
- **Personnalisation limitée** :
 - Options de customisation insuffisantes
 - Difficulté à adapter les propositions aux besoins spécifiques
- **Adaptabilité réduite** :
 - Prise en compte imparfaite des contraintes budgétaires
 - Suggestions peu évolutives selon les préférences individuelles

1.3.4 Solution proposée

Pour répondre aux défis liés à la planification de voyages, nous proposons une application mobile intelligente intégrant l'intelligence artificielle. Elle offre une expérience utilisateur

personnalisée, depuis la planification jusqu'à la réalisation du séjour.



FIGURE 1.7 – Solution proposée

Ce projet a pour objectif de comparer différentes approches de recommandation (basées sur l'intelligence artificielle) pour sélectionner celle qui offre les suggestions les plus pertinentes. Le résultat attendu est le développement d'un système capable d'aider les voyageurs à organiser leurs séjours de manière efficace et personnalisée. La qualité des données sur les destinations, les préférences des utilisateurs et les offres disponibles sera un élément essentiel pour garantir la performance et la fiabilité de la solution finale.

1.4 Méthodologie Adoptée

Afin de réussir un projet en respectant les délais, le budget et les ressources, il faut être organisé et efficace, en suivant une méthodologie de management adaptée qui aide à accomplir chaque étape de projet de façon structurée.

1.4.1 Méthodologie Agile

La méthode Agile est une approche de développement logiciel fondée sur des cycles courts et répétés, permettant de livrer régulièrement des versions fonctionnelles du produit.

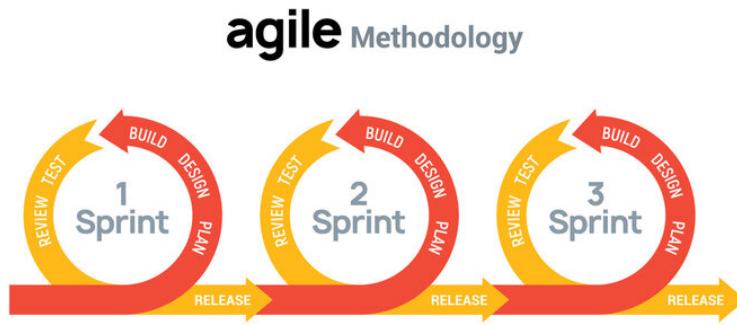


FIGURE 1.8 – Méthodologie Agile

1.4.2 Choix et justification de méthodologie

Pour choisir la méthodologie la plus appropriée, nous effectuons une comparaison des approches principales de développement. Un tableau comparatif est donc présenté pour identifier celle qui correspond le mieux aux exigences de notre projet

Méthodologie	Principes clés	Avantages	Inconvénients
Waterfall (Cycle en V)	Approche linéaire et séquentielle. Chaque phase doit être complétée avant de passer à la suivante.	Facile à gérer, bien adaptée aux projets à exigences stables.	Rigidité, peu de flexibilité aux changements tardifs.
V-Model	Extension du modèle Waterfall avec une forte emphase sur la validation et la vérification.	Meilleur contrôle qualité avec phases de test planifiées.	Même rigidité que Waterfall, difficile à adapter aux changements.
Spiral	Approche itérative combinant conception et prototypage avec gestion des risques.	Bonne gestion des risques, adaptée aux projets complexes.	Complexe à gérer, coûteux et nécessite une expertise importante.
Agile Scrum	Approche itérative, divisée en sprints. Collaboration forte et adaptation continue.	Grande flexibilité, forte implication client, livraisons rapides.	Moins adapté aux projets très réglementés ou avec exigences figées.

TABLE 1.1 – Comparaison entre différentes méthodologies de développement logiciel

Pour le projet **Easy Travel**, la méthodologie **Agile Scrum** a été choisie en raison de sa flexibilité et de son approche itérative, qui s'adaptent parfaitement à un projet web évolutif. En effet, Easy Travel vise à développer une application intelligente basée sur les préférences des utilisateurs, ce qui nécessitent des ajustements réguliers et une interaction constante avec

les parties prenantes.

Scrum permet de diviser le projet en *sprints*, c'est-à-dire des cycles de développement courts, facilitant l'intégration continue de nouvelles fonctionnalités et la prise en compte des retours utilisateurs. De plus, cette méthode favorise la collaboration au sein de l'équipe de développement et permet une meilleure réactivité face aux imprévus ou aux changements de priorités.

Contrairement aux modèles plus rigides comme le Waterfall ou le V-Model, Agile Scrum offre la souplesse nécessaire pour expérimenter, tester rapidement des prototypes, et améliorer progressivement la solution. Cela en fait une méthode parfaitement adaptée aux besoins du projet Easy Travel, qui repose sur l'innovation, la personnalisation et l'amélioration continue.

1.4.3 Démarche scrum

Scrum est un cadre de travail utilisé pour gérer le développement des produits complexes. Il est défini comme un processus itératif et collaboratif, centré sur des objectifs communs, visant à livrer des produits de haute valeur de manière efficace et innovante.[1]

Rôles des membres d'équipe

La bonne gestion des ressources humaines est essentielle à la réussite d'une équipe. Dans la méthode Scrum, trois rôles principaux organisent la structure et assurent le bon déroulement du projet. Le tableau ci-dessous présente les membres de notre équipe Scrum ainsi que leurs responsabilités.

Rôle	Missions	Acteurs
Product Owner	<ul style="list-style-type: none"> — Identifier les besoins et définir les fonctionnalités à développer. — Prioriser et organiser les éléments du Product Backlog . — Valider les fonctionnalités complétées et reclasser celles à corriger. 	Mohamed Massous
Scrum Master	<ul style="list-style-type: none"> — Animer les réunions de planification et les stand-ups quotidiens. — Aider à lever les obstacles rencontrés par l'équipe. — Veiller au respect des bonnes pratiques Scrum. 	Seif Eddine Thairi
Équipe de développement	<ul style="list-style-type: none"> — Concevoir et développer les différentes fonctionnalités de l'application. — Réaliser les tests et assurer la qualité du produit. 	Mohamed Massous Seif Eddine Thairi

TABLE 1.2 – Répartition des rôles SCRUM au sein de l'équipe Easy Travel

Outils utilisés

Dans le cadre de la méthodologie Scrum, nous utilisons le Product Backlog et le backlog de sprint pour structurer et planifier les tâches à réaliser.

Product Backlog : Il s'agit d'une liste des fonctionnalités à développer pour le produit ou service final, appelées également histoires utilisateur (user stories).

Backlog de sprint : Ce backlog contient les éléments du Product Backlog qui ont été sélectionnés pour être développés lors d'une itération spécifique.

Etapes d'une itération

Le schéma ci-dessus illustre le processus Scrum, utilisé pour gérer un projet en plusieurs sprints (cycles courts de 2 à 4 semaines).

Le projet commence par une vision, traduite en user stories (besoins utilisateurs).

Ces stories sont sélectionnées dans le sprint backlog lors de la planification.

L'équipe développe les fonctionnalités pendant le sprint.

Un Daily Scrum (réunion de 15 minutes chaque jour) permet de suivre l'avancement.

À la fin du sprint, un incrément du produit est livré.

Une revue et une rétrospective permettent d'améliorer les prochains sprints.

Le Scrum Master veille au bon déroulement du processus.

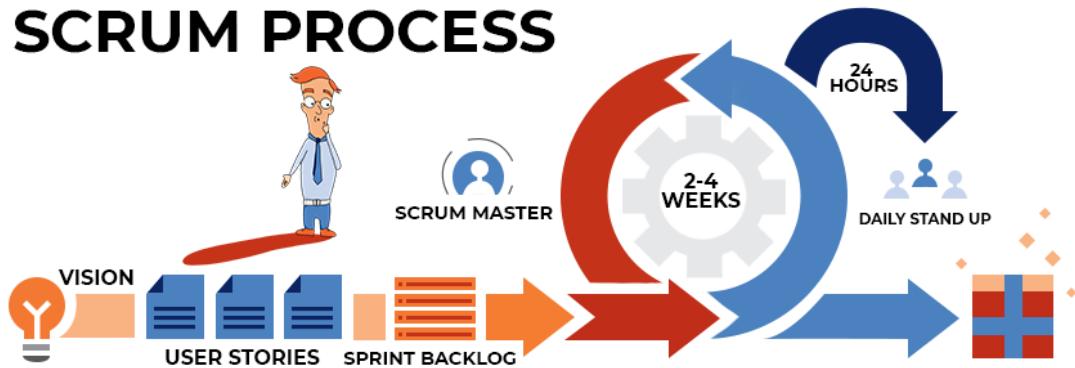


FIGURE 1.9 – Présente un résumé du processus de la méthodologie Scrum

1.5 Conclusion

Dans ce chapitre, nous présentons le projet Easy Travel. Nous parlons de l'entreprise, du sujet du projet et des problèmes que nous voulons résoudre. Ensuite, nous étudions des applications similaires comme WonderPlan AI et TripPlanner.AI, et nous voyons leurs limites.

Pour cela, nous proposons une nouvelle solution : Easy Travel. C'est une application simple et pratique pour aider les gens à organiser leurs voyages. Nous choisissons la méthode Agile, et plus précisément Scrum, pour bien organiser le travail en équipe.

Ce projet a des objectifs clairs et une bonne méthode. Cela nous aide à réussir.

Sprint 0 : Spécification des besoins

2.1 Introduction

Dans ce chapitre, nous nous concentrons sur la spécification des besoins fonctionnels et non fonctionnels. Tout d'abord, nous procémons à la collecte des besoins des utilisateurs. Ensuite, nous présentons le diagramme global des cas d'utilisation ainsi que les différents acteurs impliqués dans notre système.

2.2 Collecte des besoins

Dans cette section, nous allons définir les acteurs de notre système ainsi que les besoins fonctionnels et non fonctionnels.

2.2.1 Besoins fonctionnels

Pour l'utilisateur :

- Créer un compte et se connecter.
- Gérer Profil.
- Définir préférences de voyage.
- Recevoir suggestions personnalisées.
- Accéder aux activités touristiques.
- Planifier et gérer budget.
- Réserver services (vols, hôtels, restaurants ..).

Pour l'utilisateur :

- Gérer les clients.
- Gérer les Voyages.
- Surveiller performances par les Tableaux de bord.

Pour l'administrateur :

- Gérer comptes utilisateurs.

2.2.2 Besoins non fonctionnels

- **Performance** : Le système doit garantir une réponse rapide (inférieure à 2 secondes) aux requêtes des utilisateurs, même en cas de forte charge. L'optimisation des temps de chargement et la fluidité de navigation sont essentielles.
- **Accessibilité** : L'interface utilisateur doit être intuitive, ergonomique et conforme aux normes d'accessibilité . Le site doit être responsive, c'est-à-dire accessible et fonctionnel sur tous types d'appareils (smartphones, tablettes).
- **Sécurité** : Le système doit assurer la confidentialité, l'intégrité et la disponibilité des données. Cela inclut l'authentification sécurisée, la gestion des droits d'accès, le chiffrement des données sensibles, et la protection contre les attaques courantes.
- **Fiabilité** : Le système doit être disponible au moins 99,9% du temps. Il doit pouvoir fonctionner de manière continue sans défaillance majeure, avec des mécanismes de sauvegarde et de reprise après incident.
- **Évolutivité** : L'architecture du système doit permettre l'ajout facile de nouvelles fonctionnalités sans devoir refondre l'ensemble. Elle doit également permettre la montée en charge en cas d'augmentation du nombre d'utilisateurs ou du volume de données.
- **Maintenabilité** : Le code source doit être structuré, documenté et modulaire pour faciliter les corrections de bugs, les mises à jour et les évolutions.

2.3 Diagramme de cas d'utilisation global

Après avoir achevé la description des besoins de notre système, cette section s'intéresse à la conception de la solution. Nous abordons d'abord sa conception globale, avant de passer aux aspects plus détaillés et spécifiques.

2.3.1 Identification des acteurs

L'identification des acteurs constitue une étape clé pour définir le périmètre du système. Dans cette section, nous présentons les différents acteurs qui interagissent directement avec notre système, en détaillant leurs rôles et leurs exigences. Les principaux acteurs identifiés sont les suivants :

- Client
- Administrateur
- Utilisateur

2.3.2 Modélisation du diagramme

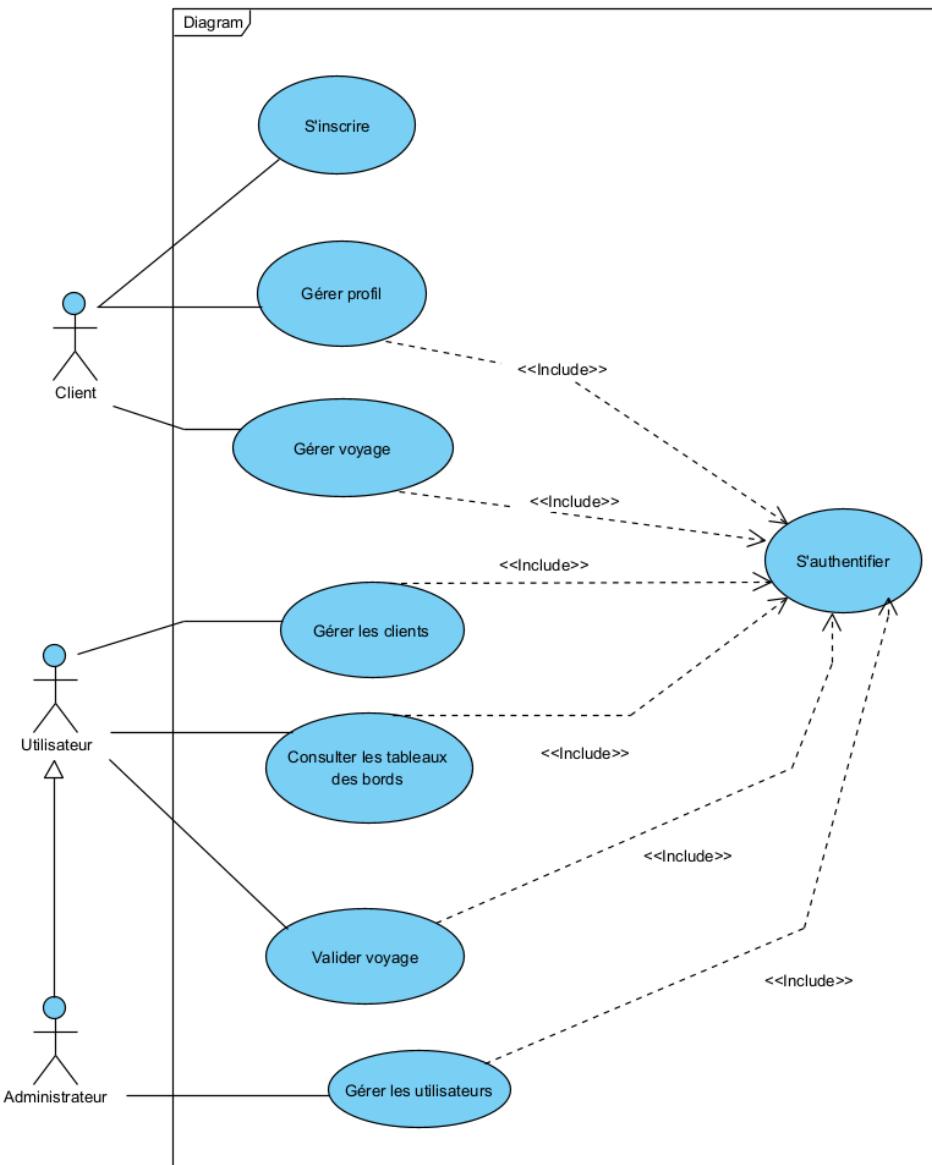


FIGURE 2.1 – Diagramme de cas d'utilisation global

2.4 Diagramme de classes

Dans le langage UML, le diagramme de classes permet de représenter la structure d'un système ou d'un sous-système. Il peut être utilisé de manière globale pour modéliser l'ensemble du système, ou divisé en plusieurs diagrammes pour illustrer différentes parties. En phase de conception, ce type de diagramme reflète l'architecture du code dans une approche orientée objet.

- **Classe** : Représentation abstraite d'un ensemble d'objets ayant les mêmes attributs et comportements.
- **Objet** : Entité bien définie, dotée d'une identité propre, d'un état et d'un comportement. Une *instance* est un objet concret issu d'une classe.

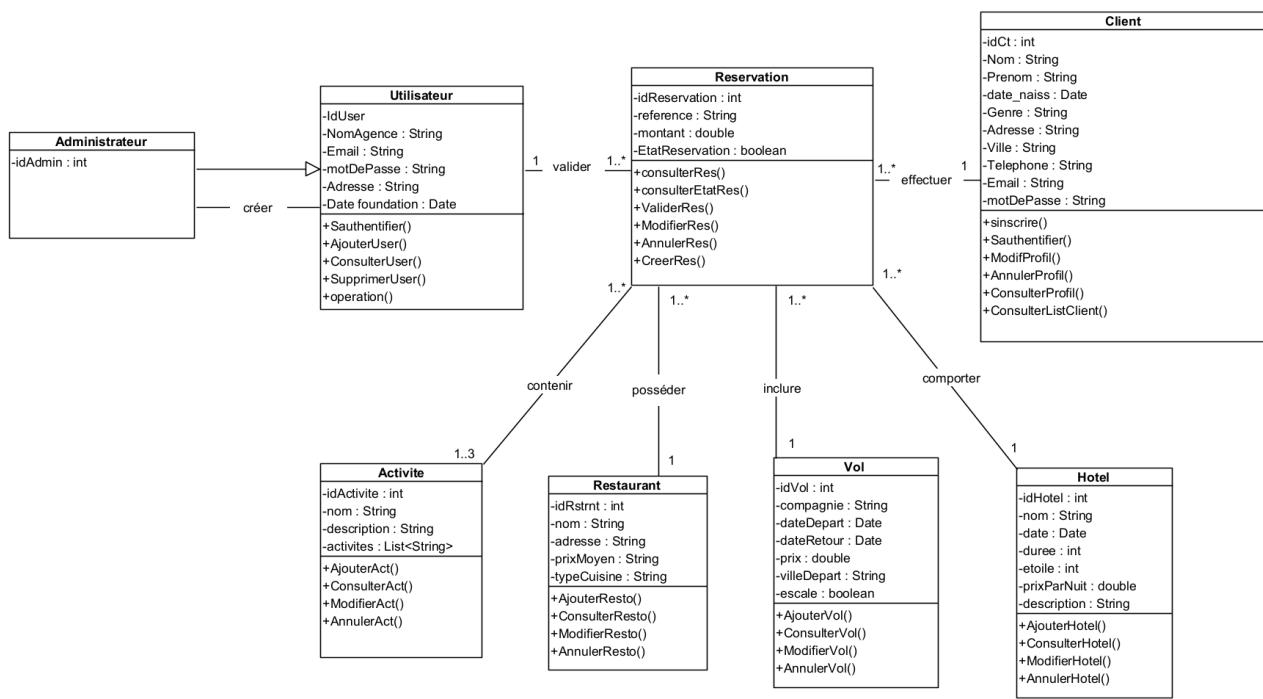


FIGURE 2.2 – Diagramme de classes

- **Attribut** : Information caractéristique décrivant une propriété d'une classe.
- **Opération** : Fonction ou service que la classe peut réaliser.
- **Association** : Lien sémantique durable entre deux classes, représentant leur relation.

Description du diagramme de classes

Classe	Description
Administre(ElyosDigital)	ElyosDigital supposée pour gérer les droits administratifs .
Utilisateur	Représente un utilisateur du système avec des informations comme l'email, le mot de passe, le nom de l'agence, et l'adresse. Offre des méthodes pour l'authentification et la gestion des utilisateurs.
Reservation	Gère les réservations avec des attributs comme la référence, le montant, et l'état. Permet de consulter, valider, modifier, annuler, et créer des réservations.
Client	Contient les informations personnelles d'un client (nom, prénom, adresse, etc.). Permet l'inscription, l'authentification, et la gestion du profil.
Tableau de bord	Représente un tableau de bord avec un identifiant, un type, et un nom. Permet de créer, consulter, et modifier les tableaux de bord.

Classe	Description
Activité	Gère les activités avec un nom, une description, et une liste d'activités. Permet d'ajouter, consulter, modifier, et annuler des activités.
Restaurant	Décrit un restaurant avec des attributs comme le nom, l'adresse, le prix moyen, et le type de cuisine. Permet d'ajouter, consulter, modifier, et annuler des restaurants.
Vol	Représente un vol avec des informations comme la compagnie, les dates de départ et retour, le prix, etc. Permet d'ajouter, consulter, modifier, et annuler des vols.
Hotel	Décrit un hôtel avec des attributs comme le nom, la date, la durée, le prix par nuit, etc. Permet d'ajouter, consulter, modifier, et annuler des hôtels.

TABLE 2.1 – Description des classes dans le diagramme de classes

2.5 Product Backlog

2.5.1 Définition des sprints

Le **Product Backlog** est un artefact fondamental du cadre Scrum. Il s'agit d'une liste ordonnée des besoins et attentes des clients, exprimés sous forme d'exigences fonctionnelles. Le *Product Owner* est chargé de prioriser ces éléments afin de planifier l'ordre de mise en œuvre des fonctionnalités dans le développement de l'application.

Chaque *User Story* est analysée en détail lors des sprints et doit être entièrement réalisée. Le Product Backlog , représenté dans le tableau ci-dessous, comprend les champs suivants :

- **IDF** : Identifiant unique attribué à chaque fonctionnalité.
- **Fonctionnalité** : Service ou fonction spécifique et cohérente du produit.
- **User Story** : Description de la fonctionnalité, formulée du point de vue de l'utilisateur.
- **User Story ID** : Identifiant numérique unique et auto-incrémenté attribué à chaque User Story.
- **Priority** : Niveau de priorité défini par le *Product Owner*, allant de la plus élevée à la moins élevée.[2]

Fonctionnalité	ID User Story	User Story	Priorité
Inscription	US01	En tant que client, je veux créer un compte pour accéder aux fonctionnalités personnalisées.	Haute
Authentification	US02	En tant que client, je veux me connecter avec mes identifiants pour accéder à mon espace personnel.	Haute
Recherche de destinations	US03	En tant que client, je veux rechercher des destinations par mot-clé ou catégorie.	Haute
Visualiseur Destination	US04	En tant que client, je veux voir les détails d'une destination (description, photos).	Moyenne
Filtre Destinations	US09	En tant que client, je veux filtrer les destinations selon le prix ou la région.	Moyenne
Réservations	US05	En tant que client, je veux réserver un voyage en ligne.	Haute
Gestion des destinations	US07	En tant qu'utilisateur, je veux consulter ,valider , modifier ou annuler des voyages.	Moyenne
Statistiques	US10	En tant qu'utilisateur, je veux consulter les statistiques des réservations.	Basse

TABLE 2.2 – Product Backlog

2.5.2 Planification des sprints

Afin d'assurer le bon déroulement du projet, le travail a été réparti en plusieurs sprints. Ces sprints ont été définis à partir du backlog Product, en tenant compte des priorités des différents modules. Nous avons structuré notre travail en 3 sprints afin de mieux équilibrer la charge de travail, comme présenté dans le tableau ci-dessous.

Sprint	Modules	Durée estimée
Sprint 1	— Authentification et gestion des utilisateurs	3 semaine
Sprint 2	— Recherche de voyage alimentée par l'IA	3 semaines
Sprint 3	— Interface administrateur et finalisation du projet	3 semaine

2.5.3 Diagramme de Gantt

Pour organiser les tâches et suivre l'évolution des différentes activités du projet, il est essentiel de créer un diagramme de Gantt. Cette méthode, largement adoptée en gestion de projet, permet de visualiser le flux de travail. Voici notre diagramme de Gantt :

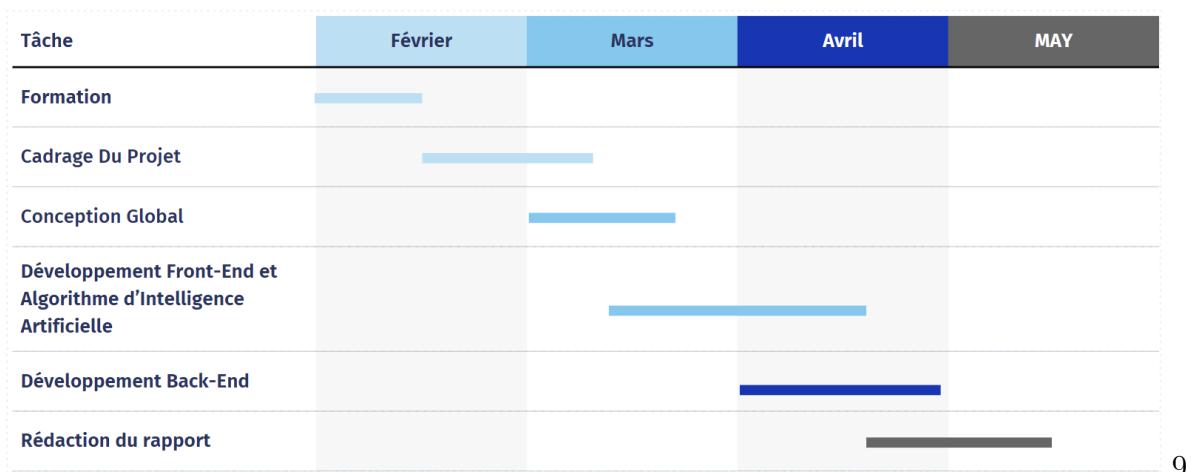


FIGURE 2.3 – Planning prévisionnel détaillé

2.6 Environnement de développement

Dans cette section, nous présentons l'ensemble des ressources matérielles, logiciels, outils et frameworks utilisés pour la conception, le développement, la modélisation, le test et la mise en œuvre de notre application.

2.6.1 Environnement matériel

Les ressources matérielles utilisées pour mener à bien le projet sont les suivantes :

Équipement	Caractéristiques
Ordinateur de développement	<ul style="list-style-type: none">— Mémoire RAM : 16 Go— Processeur : Intel Core i5 (12^e génération)— Stockage : 512 Go SSD— Système d'exploitation : Windows 10 64 bits
Smartphone de test (Samsung Galaxy S23)	<ul style="list-style-type: none">— Processeur : Qualcomm Snapdragon 8 Gen 2— RAM : 8 Go— Stockage : 256 Go— Système d'exploitation : Android 13

2.6.2 Outil logiciel

L'environnement logiciel regroupe les outils utilisés pour la modélisation, le développement, la conception UI, les tests, la gestion de version, et la structuration des données.

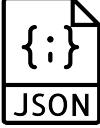
Outil	Logo	Utilisation
Visual Paradigm	 Visual Paradigm	Modélisation des diagrammes UML nécessaires à la conception, en mettant l'accent sur la réutilisabilité, la modularité et l'extensibilité.[6]
Android Studio		Environnement officiel pour le développement Android, utilisé pour coder, tester et exécuter l'application mobile.[7]
Figma		Outil collaboratif pour concevoir les wireframes, maquettes et prototypes interactifs d'interface utilisateur. [8]
Visual Studio Code		Éditeur léger pour développer le front-end et le back-end, compatible avec de nombreuses extensions.[9]
Postman		Permet de tester les API en envoyant des requêtes HTTP et en analysant les réponses.[10]
Git		Gestion de version pour suivre les modifications, collaborer et revenir à des versions antérieures du projet.[11]
JSON		Format léger pour structurer et échanger les données entre client et serveur.[12]

TABLE 2.4 – Outils de développement

2.6.3 Frameworks et langage de programmation

Nous avons fait appel à plusieurs technologies modernes et efficaces pour assurer le bon fonctionnement de notre application, tant côté client que serveur.

Technology	Description	Languages	Logo
Vue.js	JavaScript framework for building dynamic user interfaces, ideal for single-page applications.[13]	JavaScript, HTML	
Laravel	PHP framework following the MVC pattern, offering elegant syntax and powerful tools for backend development.[14]	PHP	
Flutter	Google's framework for cross-platform mobile development (iOS and Android) from a single codebase.[15]	Dart	
Python	Versatile language, primarily used for data processing and application business logic.[16]	Python	
LLaMA	A family of open-source large language models developed by Meta, used in natural language processing tasks.[17]	Python	

TABLE 2.5 – Frameworks and development languages

2.7 Architecture

2.7.1 Architecture technique

L'architecture en trois tiers (3-tier) est un modèle d'architecture logicielle qui sépare une application en trois couches principales :

- **La couche présentation** : elle représente l'interface utilisateur. Elle permet à l'utilisateur d'interagir avec l'application.
- **La couche métier** : elle contient la logique du traitement des données, les règles métiers, et gère la communication entre la présentation et les données.
- **La couche donnée** : elle s'occupe de la gestion, du stockage et de l'accès aux données. Ce modèle facilite la maintenance, l'évolution et la réutilisation de l'application en séparant

clairement les responsabilités.[3]

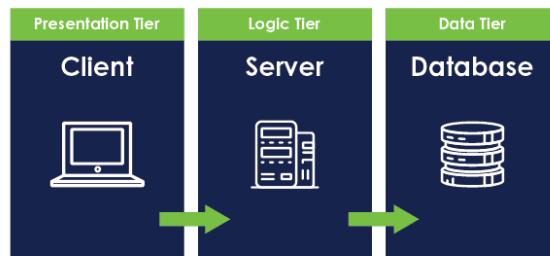


FIGURE 2.4 – Architecture technique

2.7.2 Architecture logique

L'architecture MVC (Modèle/Vue/Contrôleur) est une façon d'organiser une interface graphique d'un programme. Elle consiste à différencier trois entités distinctes, qui sont le modèle ; la vue et le contrôleur ayant chacun un rôle précis dans l'interface.[4]

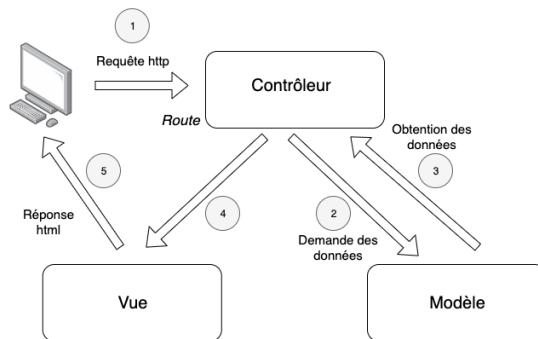


FIGURE 2.5 – Architecture logique

2.7.3 Diagramme de déploiement

Le diagramme de déploiement est une vue statique qui sert à représenter l'utilisateur de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. [5]

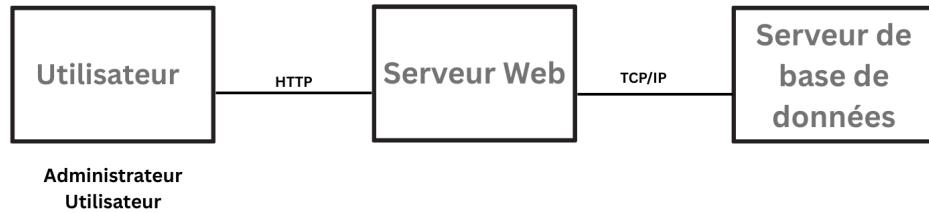


FIGURE 2.6 – Diagramme de déploiement

2.8 Conclusion

Dans ce chapitre, nous avons analysé les besoins fonctionnels et non fonctionnels de notre solution, tout en présentant le déroulement général du projet à travers un diagramme de cas d'utilisation globale. Nous passons à présent à une phase d'analyse approfondie et de conception détaillée de notre système.

Chapitre 3

Sprint 1 : Authentification et gestion des utilisateurs

3.1 Introduction

L'objectif de ce chapitre est d'exposer le premier sprint relatif au module «Mise en place de la structure de base et authentification » où nous allons détailler les différentes Fonctionnalités du sprint Backlog. Par la suite, nous procéderons à une analyse et une conception, avant de présenter les techniques et technologies mises en œuvre pour la réalisation de ce sprint.

Tâche ID	Description	Responsable	Estimation (jours)
T1	Initialisation du projet (structure frontend et backend)	Backend Developer	3
T2	Configuration du routage et des pages de base	Frontend Developer	3
T3	Création de la base de données et des tables utilisateurs	Backend Developer	2
T4	Développement du formulaire d'inscription	Frontend Developer	1
T5	Développement de l'API d'inscription	Backend Developer	3
T6	Développement du formulaire de connexion	Frontend Developer	2
T7	Développement de l'API de connexion et gestion des utilisateur	Backend Developer	2

TABLE 3.1 – Sprint Backlog détaillé – Sprint 1

3.2 Analyse

Cette section s'analyse sur l'examen du module « Authentification et gestion des comptes » représenté ci-dessous par un diagramme de cas d'utilisation qui décrit les actions possibles pour un client non connecté.

Cette section se s'analyse sur l'examen du module « Mise en place de la structure de base et authentification» représentées ci-dessous par un diagramme de cas d'utilisation qui décrit les actions possibles pour un client non connecté.

3.2.1 Raffinement de cas d'utilisation « S'inscrire »

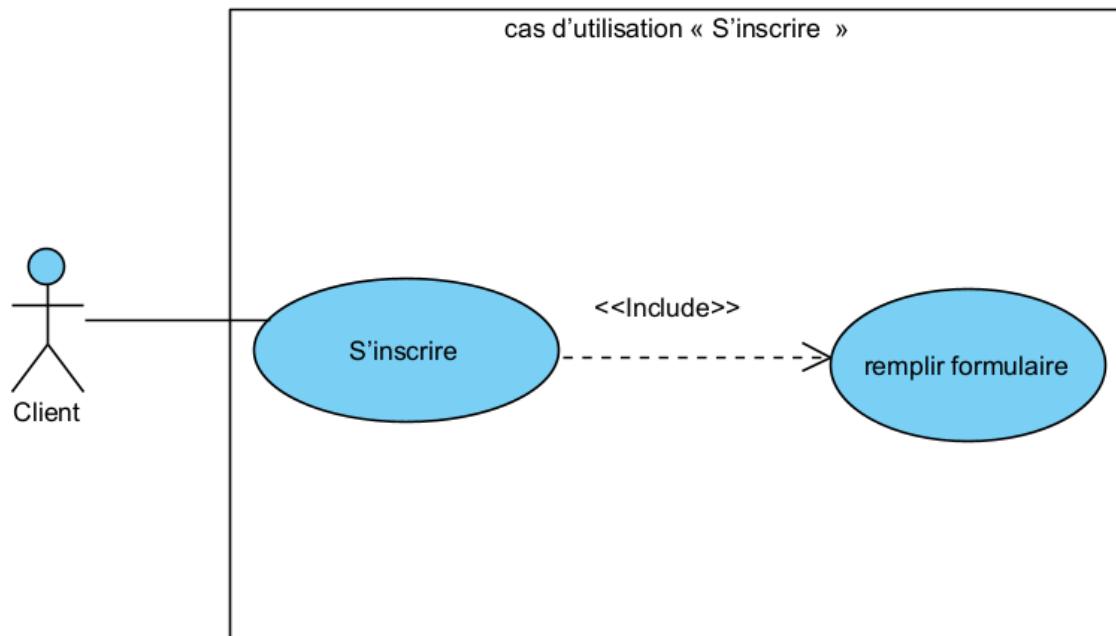


FIGURE 3.1 – Raffinement de cas d'utilisation « S'inscrire »

Acteur	Client
Objectif	Le client permet de créer un compte pour accéder à l'application.
Précondition	Tous les champs doivent être remplis.

Scénario Nominal	<ol style="list-style-type: none"> 1. Le client accède à l'application . 2. Le client sélectionne l'option « S'inscrire ». 3. Le système affiche un formulaire. 4. Le client doit remplir tous les champs avec ses informations personnelles. 5. Le système vérifie les informations. 6. Le système enregistre les données.
Postcondition	L'inscription a été créée avec succès.
Scénario Alternatif	Le système affiche un message d'erreur indiquant qu'il y a un champ vide.

TABLE 3.2 – Description Textuelle de Cas d'utilisation « S'inscrire »

3.2.2 Raffinement de cas d'utilisation « S'authentifier »

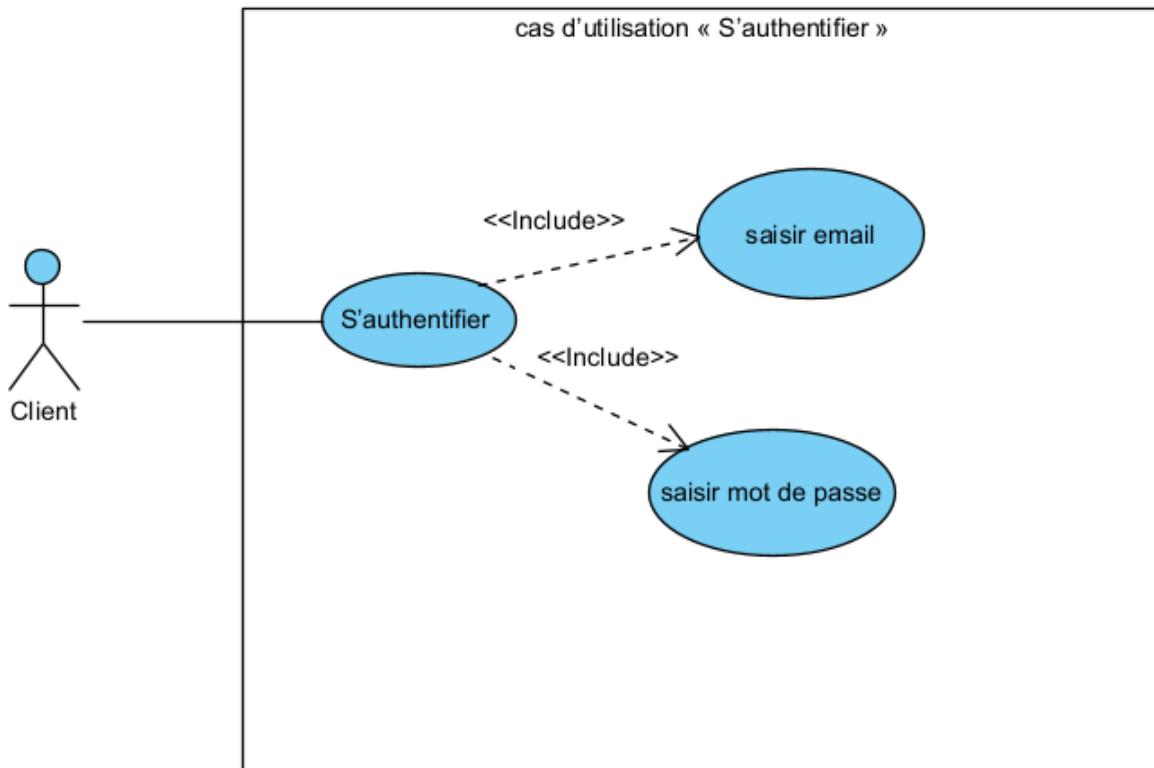


FIGURE 3.2 – Raffinement de cas d'utilisation « S'authentifier »

Acteur	Client
Objectif	Le Client permet de s'authentifier via login et mot de passe avant d'accéder à l'application.
Précondition	Le Client doit disposer d'un compte pour pouvoir s'authentifier sur l'application.
Scénario No-nominal	<ol style="list-style-type: none"> 1. Le client remplit les champs nécessaires (login et mot de passe) pour se connecter à l'application. 2. Le client clique sur le bouton se connecter. 3. Le système vérifie la cohérence des données saisies par l'utilisateur. 4. Le système redirige l'utilisateur vers la page d'accueil.
Postcondition	L'utilisateur est connecté à son compte et peut accéder aux fonctionnalités de l'application.
Scénario Alternatif	<ul style="list-style-type: none"> — Login ou mot de passe incorrect. — Champ vide. — Le système affiche un message d'erreur.

TABLE 3.3 – Raffinement de cas d'utilisation « S'authentifier »

3.2.3 Raffinement des cas d'utilisations «Gérer les clients»

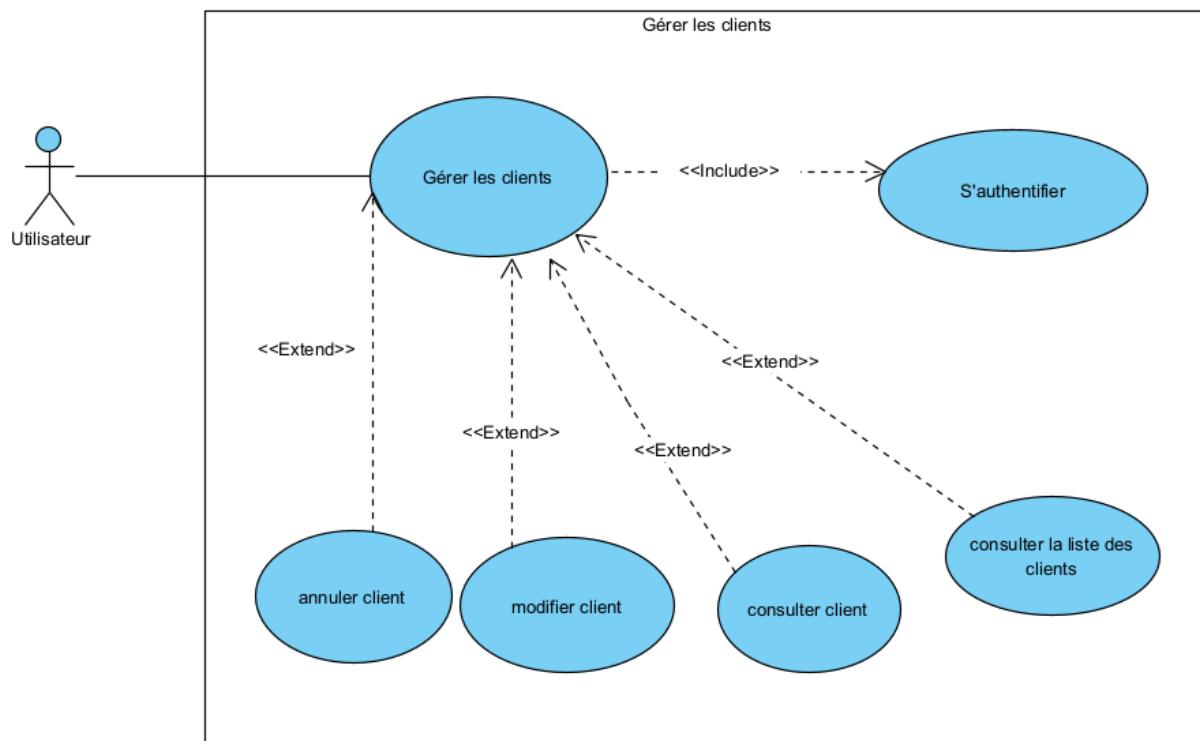


FIGURE 3.3 – Raffinement de cas d'utilisation «Gérer les clients»

Acteur	Utilisateur
Objectif	Permettre à l'administrateur de gérer les comptes clients via différentes opérations.
Précondition	L'utilisateur doit être authentifié.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'utilisateur s'authentifie via l'application web . 2. Il accède à l'interface de gestion des clients. 3. Le système affiche les options disponibles : <ul style="list-style-type: none"> — Consulter la liste des clients — Consulter client — Modifier client — Annuler client 4. L'utilisateur choisit une des actions proposées. 5. Le système exécute l'action sélectionnée et affiche un message de confirmation.

Postcondition	L'action de gestion client (consultation, modification ou suspension) est effectuée avec succès.
Scénario Alternatif	<ul style="list-style-type: none">— Si l'identification échoue, l'accès est refusé.— Si la suspension échoue, un message d'échec est affiché.

TABLE 3.4 – Description Textuelle de Cas d'utilisation « Gérer les clients »

3.3 Conception

3.3.1 Diagramme d'activité «Gérer les clients»

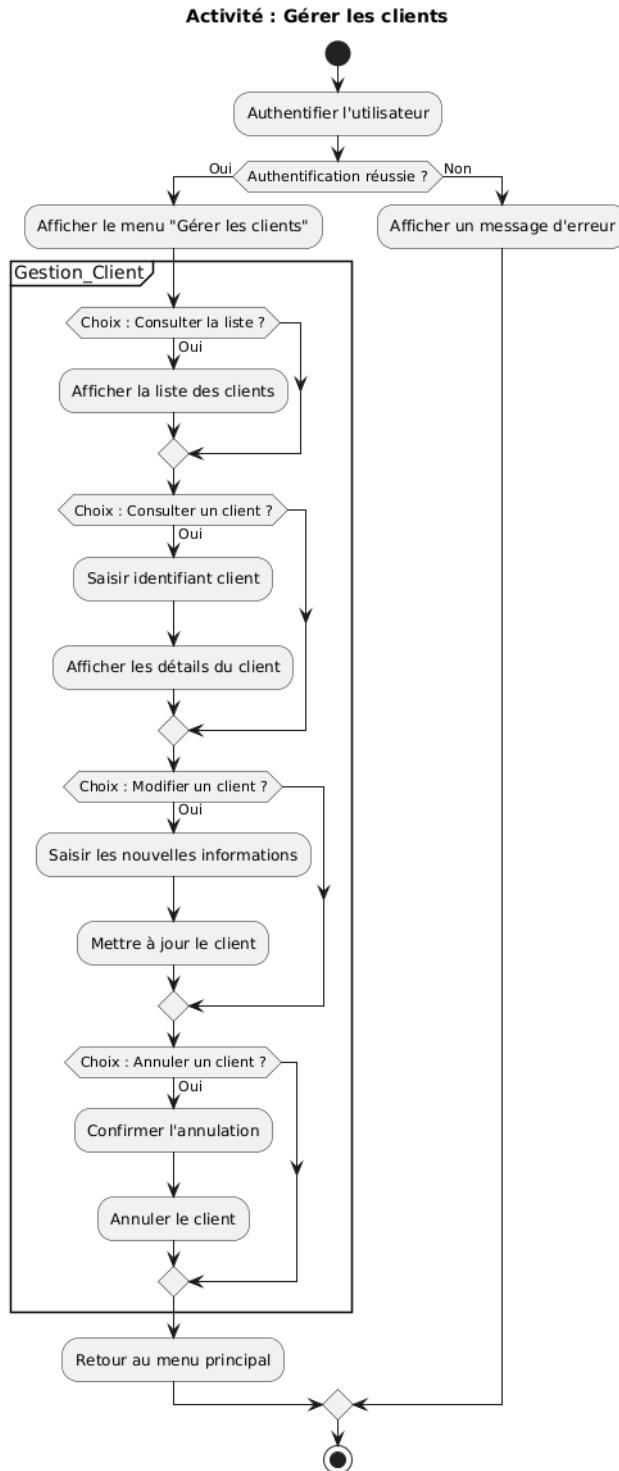


FIGURE 3.4 – Diagramme d'activité «Gérer les clients»

3.3.2 Diagramme de séquence «S'authentifier»

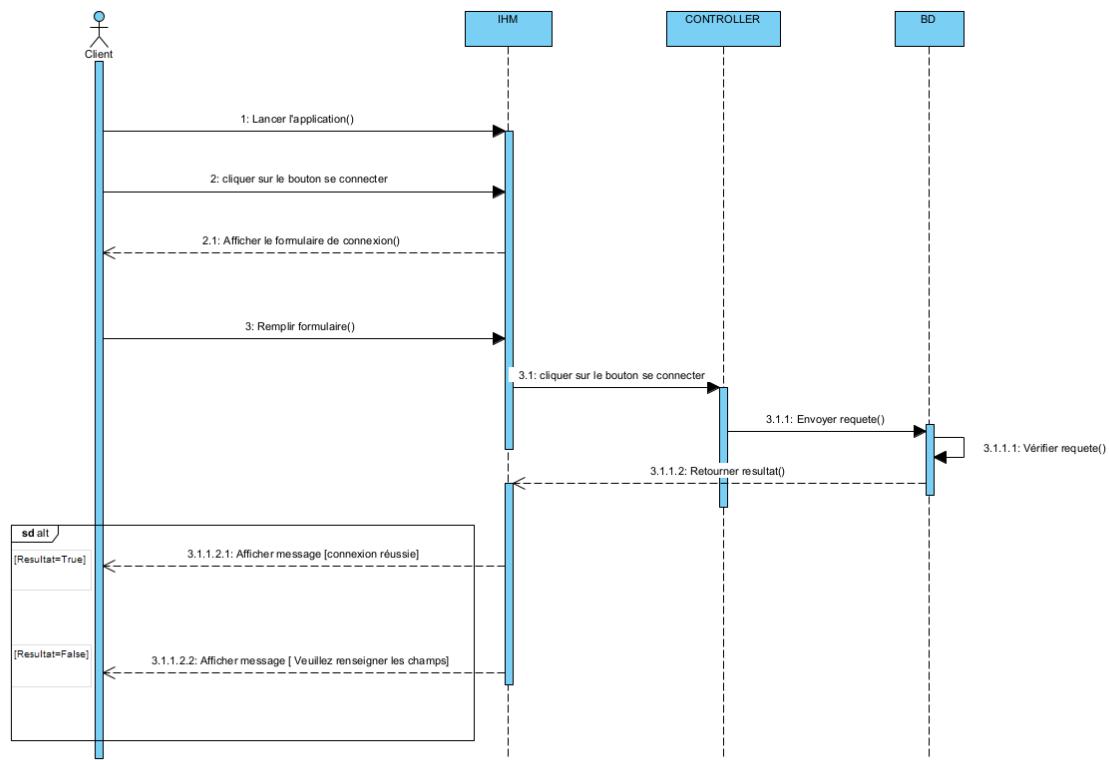


FIGURE 3.5 – Diagramme de séquence «S'authentifier»

3.3.3 Diagramme de séquence «Modifier client»

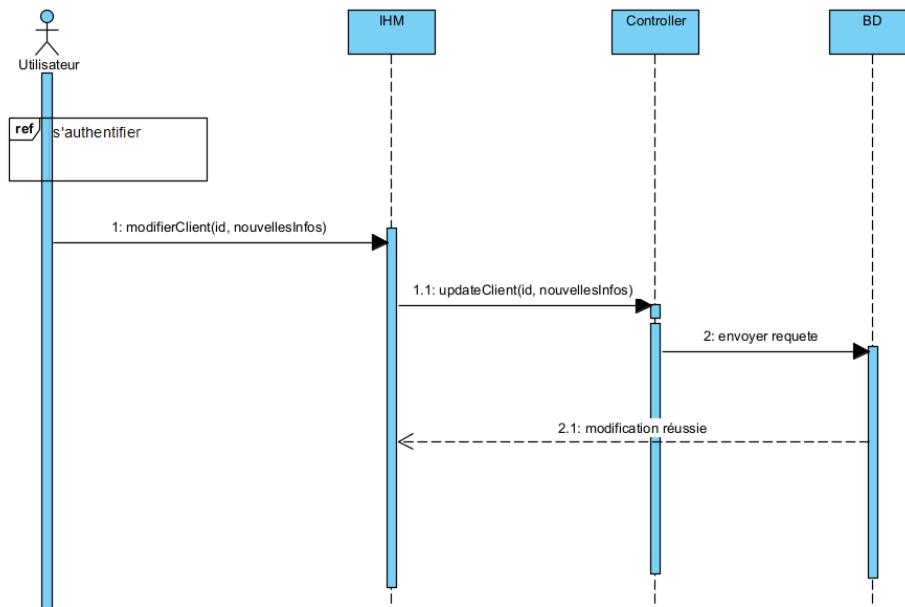


FIGURE 3.6 – Diagramme de séquence «Modifier client»

3.3.4 Diagramme de séquence «Annuler client»

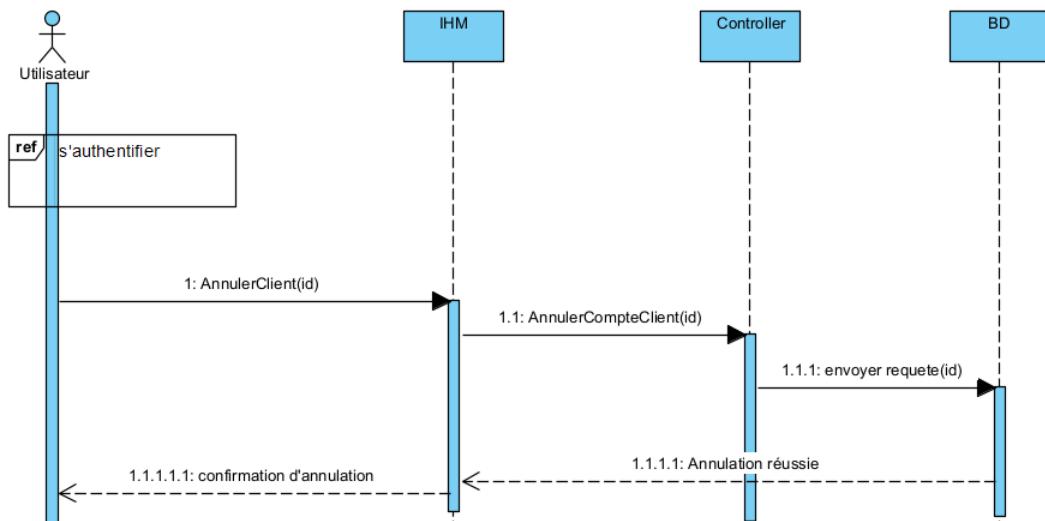


FIGURE 3.7 – Diagramme de séquence «Annuler client»

3.3.5 Schéma relationnel

Voici les tables des bases de données qui concernent cette sprint :

- Client (idCt, Nom, Prenom, dateNaiss, Genre, Adresse, Ville, Telephone, Email, motDePasse)
- Utilisateur (IdUser, Email, motDePasse, Adresse, Date foundation)

3.4 Réalisation

3.4.1 Les interfaces de bienvenue



Bienvenue

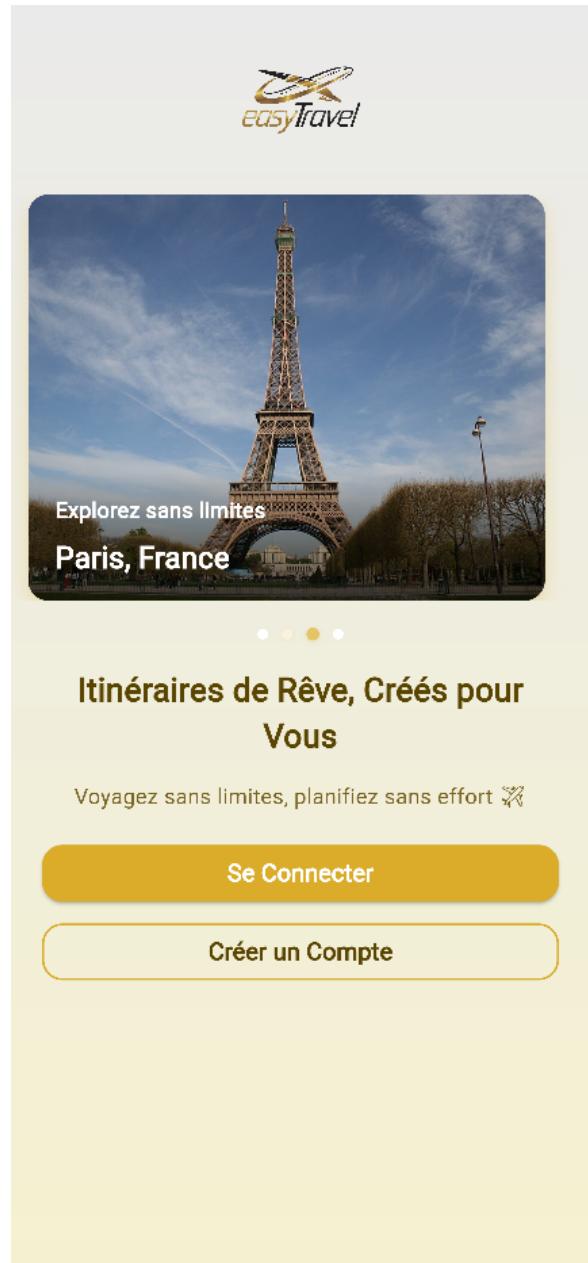


FIGURE 3.8 – Interface splash

FIGURE 3.9 – Interface de bienvenue

- L'écran splash s'affiche au lancement pour assurer une première impression professionnelle.
- L'interface de bienvenue permet la connexion ou l'inscription en toute convivialité.

3.4.2 Les interfaces de s'inscrire et de s'authentifier

The screenshot shows the 'Créer un compte' (Create account) screen for EasyTravel. At the top is the EasyTravel logo. Below it, the text 'Créer un compte' and 'Commencez votre aventure avec nous'. There are three input fields: 'Nom complet' (Name), 'Email', and 'Mot de passe' (Password). Each field has a placeholder icon (person, envelope, lock) and a small eye icon to the right for password visibility. A large yellow 'S'inscrire' (Sign Up) button is at the bottom, and a link 'Déjà un compte ? Se connecter' (Already have an account? Log in) is below it.

FIGURE 3.10 – Interface création de compte

The screenshot shows the 'Bienvenue à EasyTravel' (Welcome to EasyTravel) screen. At the top is the EasyTravel logo. Below it, the text 'Bienvenue à EasyTravel' and 'Connectez-vous pour continuer'. There are two input fields: 'Email' and 'Mot de passe' (Password). Each field has a placeholder icon (envelope, lock) and a small eye icon to the right for password visibility. A large yellow 'Se connecter' (Log in) button is at the bottom, and a link 'Pas encore de compte ? Créer un compte' (Don't have an account? Create one) is below it.

FIGURE 3.11 – Interface de s'authentifier

- L'interface de s'inscrire permet une inscription simple et intuitive.
- L'interface de s'authentifier permet aux utilisateurs existants de se connecter aisément.

3.5 Conclusion

Ce premier sprint a permis de poser les fondations essentielles de l'application en assurant l'initialisation du projet, la structuration du backend et frontend, ainsi que l'intégration des fonctionnalités d'inscription et d'authentification. Grâce à une analyse rigoureuse, une conception structurée et une réalisation soignée, les utilisateurs peuvent désormais créer un compte, se

connecter en toute sécurité et accéder à l'application. Ce sprint marque ainsi une étape cruciale dans la mise en place de l'environnement fonctionnel de ce projet.

Sprint 2 : Recherche de voyage alimentée par l'IA

4.1 Introduction

L'objectif de ce sprint est de développer les premières fonctionnalités orientées utilisateur pour améliorer l'expérience de navigation et d'interaction avec l'application. Il s'agit notamment de permettre aux utilisateurs de consulter les offres disponibles, de rechercher une destination, et d'accéder à des informations détaillées.

4.2 Sprint Backlog

Tâche ID	Description	Responsable	Estimation (jours)
T9	Création de la page d'accueil dynamique avec les offres	Frontend Developer	3
T10	Développement de l'API pour récupérer les offres depuis les dataset	Backend Developer	3
T11	Intégration de la recherche par destination (UI)	Frontend Developer	2
T12	Développement de l'API de recherche des offres	Backend Developer	3
T13	Affichage des détails d'une offre (interface utilisateur)	Frontend Developer	2
T14	Développement de l'AI pour les détails d'une offre	Backend Developer	7

T15	Mise en place des filtres (prix, catégorie, etc.)	Frontend Developer	1
T16	Optimisation des requêtes côté serveur pour les recherches filtrées	Backend Developer	1

TABLE 4.1 – Sprint Backlog détaillé – Sprint 2

4.3 Analyse

4.3.1 Raffinement de cas d'utilisation «Gérer voyage»

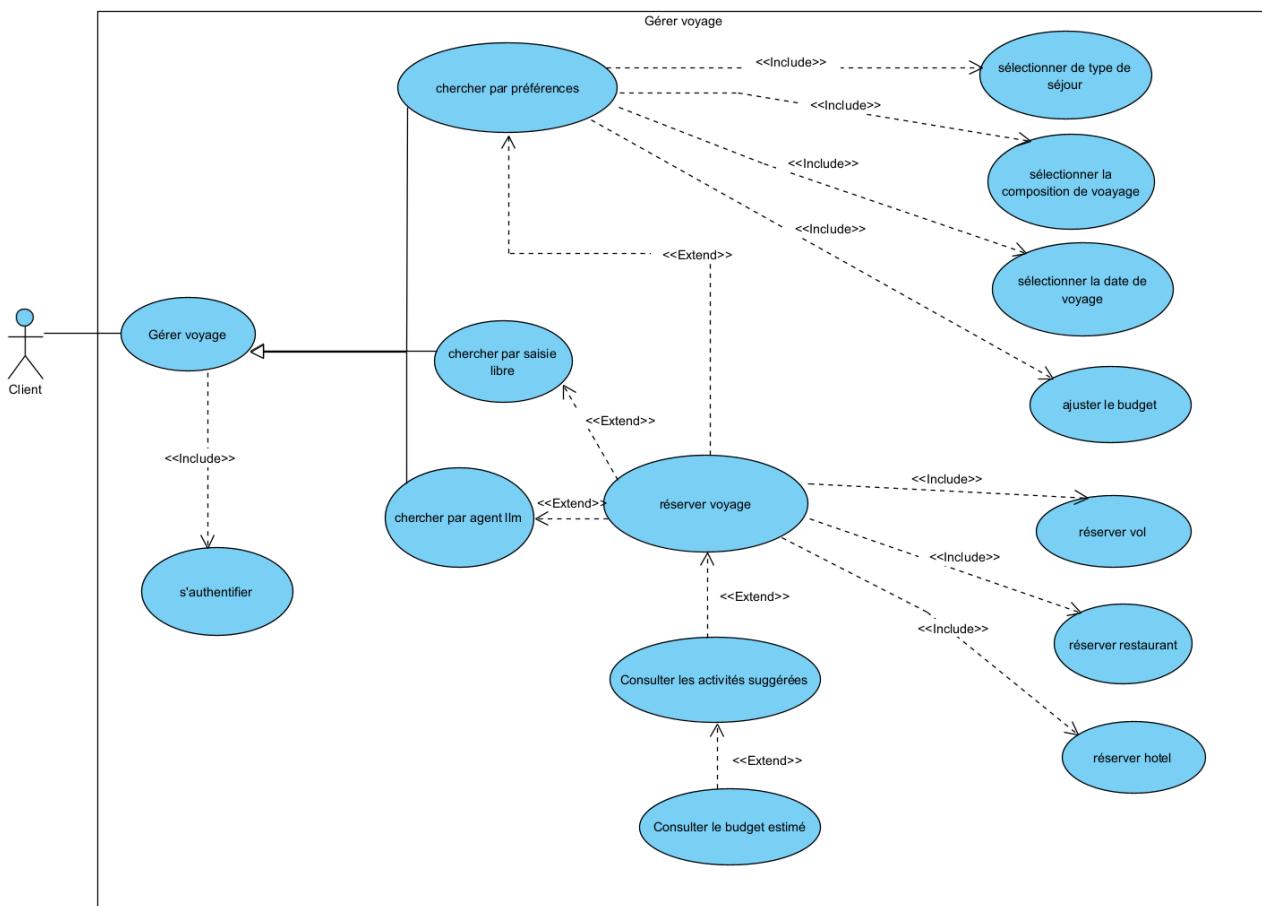


FIGURE 4.1 – Raffinement de cas d'utilisation «Gérer voyage»

TABLE 4.2 – Description du cas d'utilisation *Gérer voyage*

Élément	Description
Cas d'utilisation	Gérer voyage
Acteur	Client

Objectif	Permettre au client de rechercher, planifier et réserver un voyage personnalisé selon plusieurs modalités (préférences, saisie libre, ou via un agent LLM).
Précondition	Le client doit être authentifié sur la plateforme.
Scénario nominal	<ul style="list-style-type: none"> — Le client s'authentifie. — Il accède à la fonctionnalité <i>Gérer voyage</i>. — Il peut choisir entre trois méthodes de recherche : <ul style="list-style-type: none"> — Chercher par préférences (type de séjour, composition du voyage, date, budget), — Chercher par saisie libre, — Chercher par agent LLM. — Le système propose des résultats adaptés. — Le client peut : <ul style="list-style-type: none"> — Consulter les activités suggérées, — Consulter le budget estimé, — Réserver les services suivants : vol, hôtel, restaurant.
Postcondition	Un ou plusieurs voyages sont proposés, avec possibilité de réservation. Le client a accès aux activités et au budget estimé.
Extensions	<ul style="list-style-type: none"> — Consultation des activités suggérées — Consultation du budget estimé
Inclus	<ul style="list-style-type: none"> — Sélection du type de séjour — Composition du voyage — Date de voyage — Ajustement du budget — Réservation de vol, hôtel et restaurant

4.4 Conception

4.4.1 Diagramme d'activité «Chercher voyage»

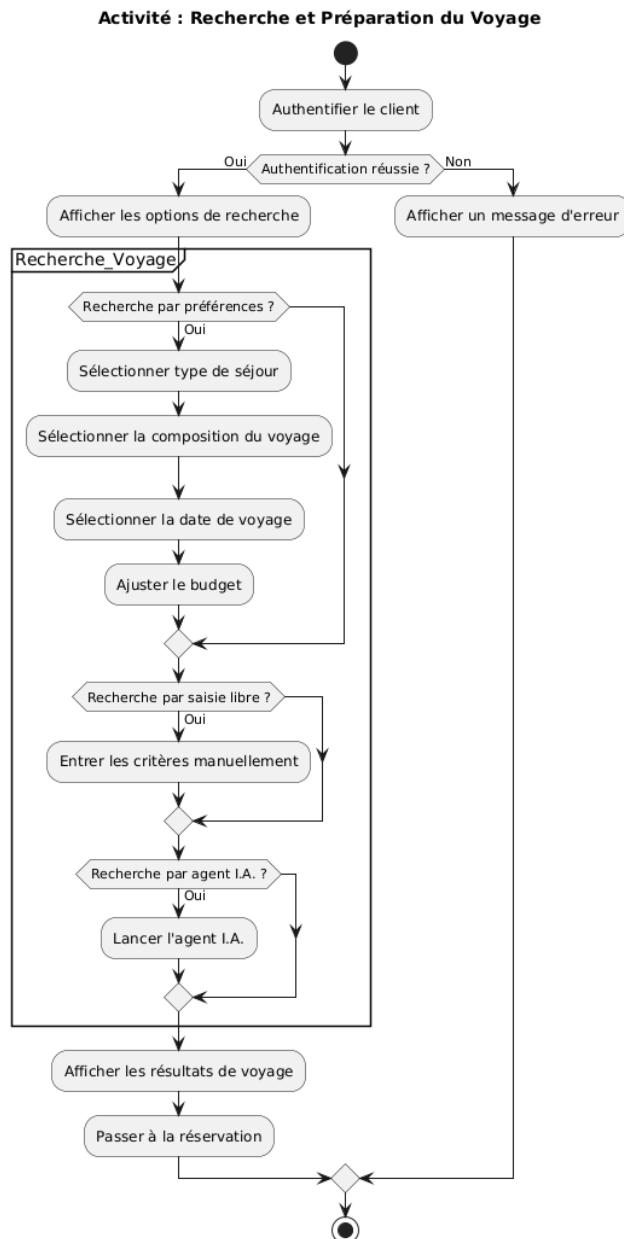


FIGURE 4.2 – Diagramme d'activité «Chercher voyage»

4.4.2 Diagramme d'activité «Réserver voyage»

Activité : Réservation et Services Complémentaires

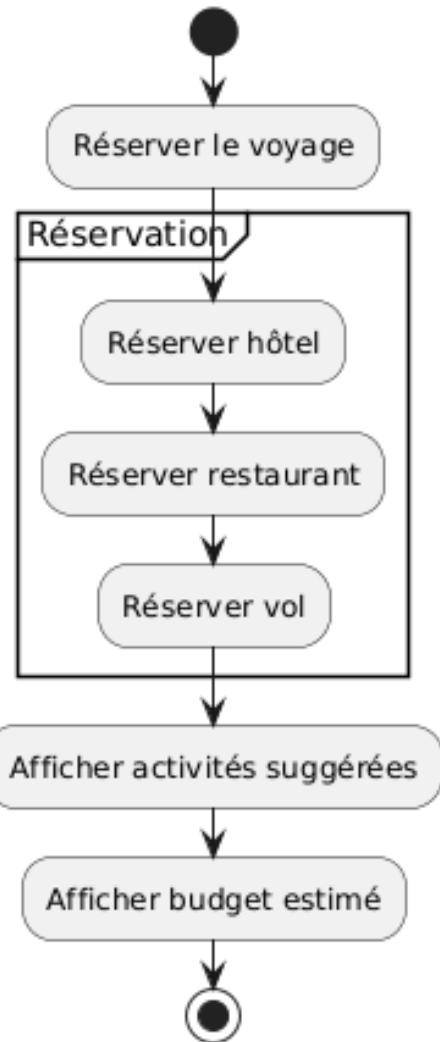


FIGURE 4.3 – Diagramme d'activité «Chercher voyage»

4.4.3 Diagramme de séquence «Chercher voyage»

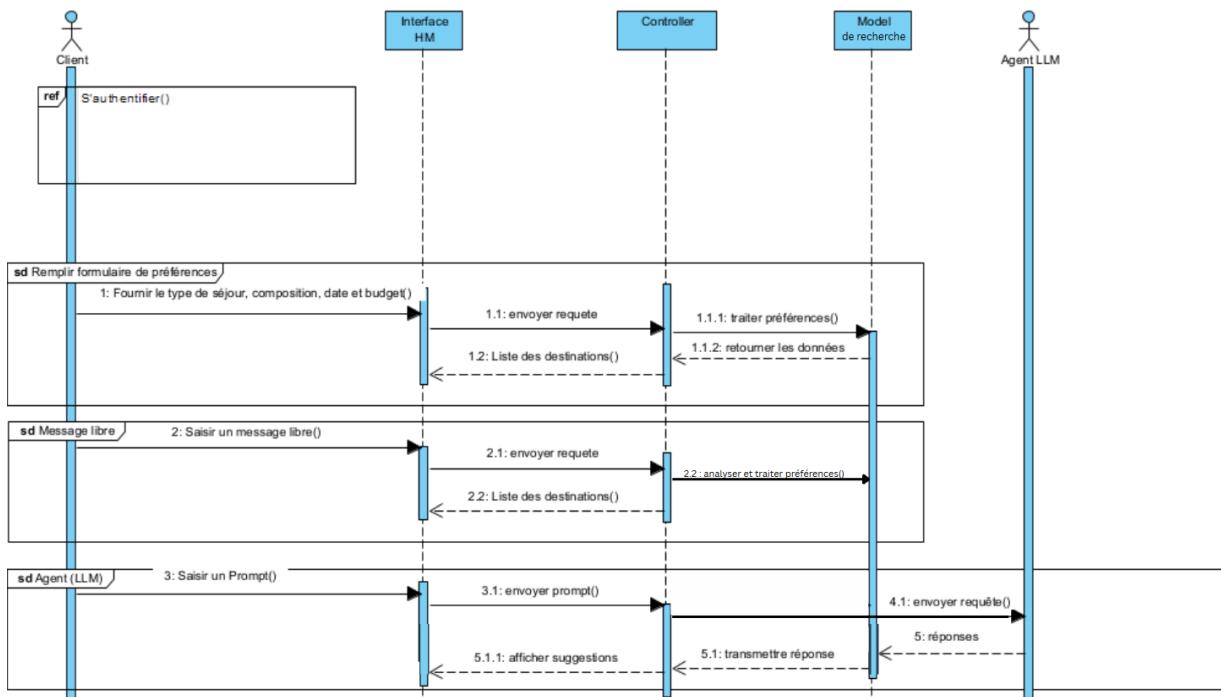


FIGURE 4.4 – Diagramme de séquence «Chercher voyage»

4.4.4 Diagramme de séquence «Réserver voyage»

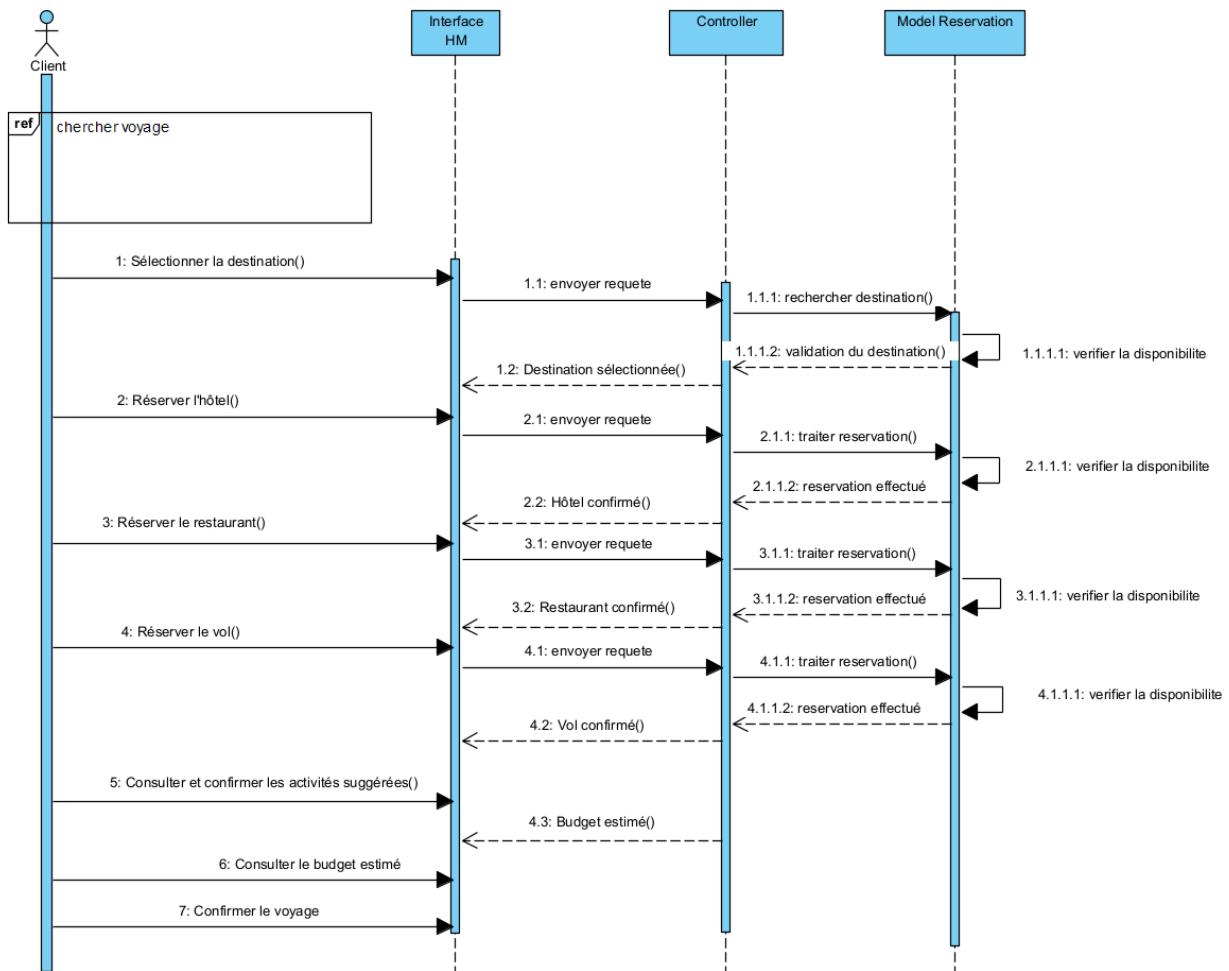


FIGURE 4.5 – Diagramme de séquence «Réserver voyage»

4.4.5 Schéma relationnel

Voici les tables des bases de données qui concernent cette sprint :

- Hotel (idHotel, nom, date, duree, etoile, prixParNuit,description)
- Restaurant (idRstrnt, nom, adresse, prixMoyen, typeCuisine)
- Vol (idVol, compagnie, dateDepart, dateRetour, prix, villeDepart, escale)
- Activité (idActivite, nom, description, List<String>)

4.5 Réalisation

4.5.1 Les interfaces d'accueil

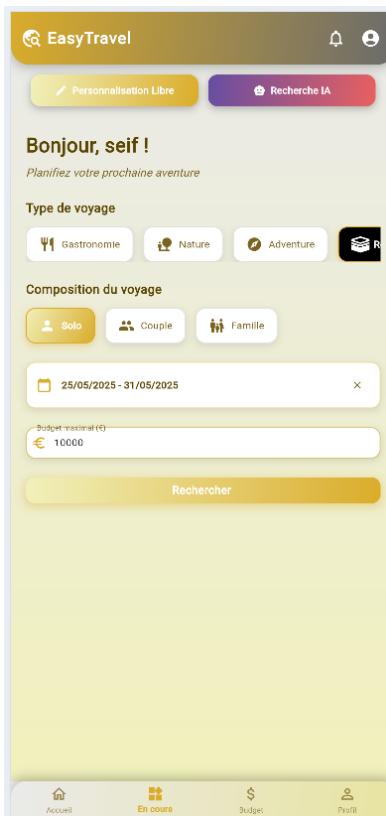


FIGURE 4.6 – Interface de page d'accueil

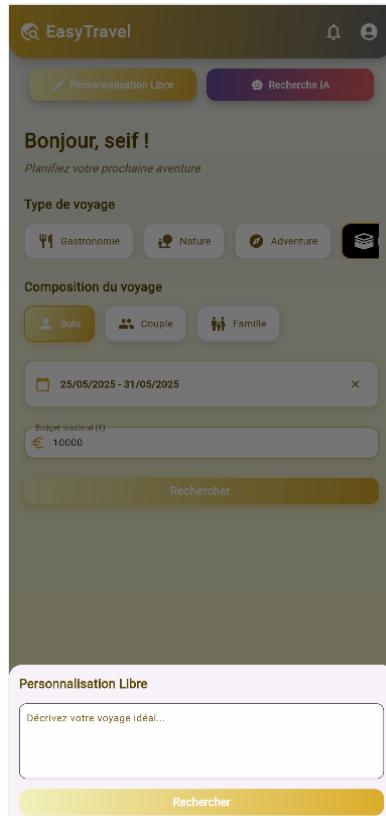


FIGURE 4.7 – Interface de chercher par message libre

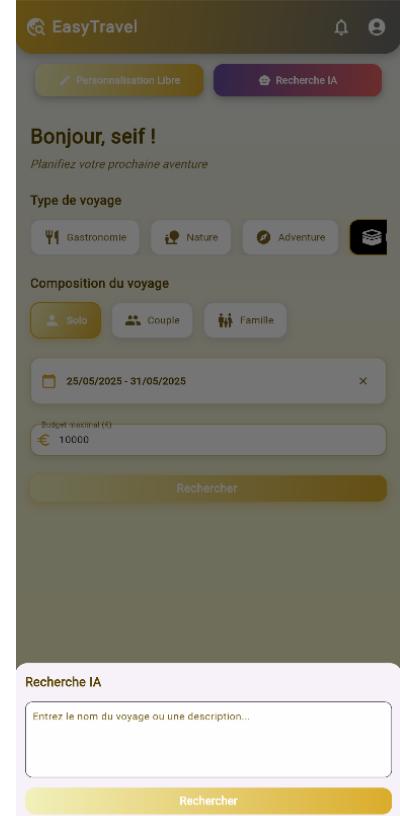


FIGURE 4.8 – Interface de chercher par l'agent llama

- Ceci est la page d'accueil de l'application EasyTravel. Le client peut choisir le type de voyage (gastronomie, nature, aventure), la composition du groupe (solo, couple, famille), les dates du voyage, et le budget maximum.
- Le client peut aussi utiliser un agent IA (Ilma Prompt) ou bien écrire un message libre pour organiser son voyage.

4.5.2 Les interfaces d'organisation de voyage

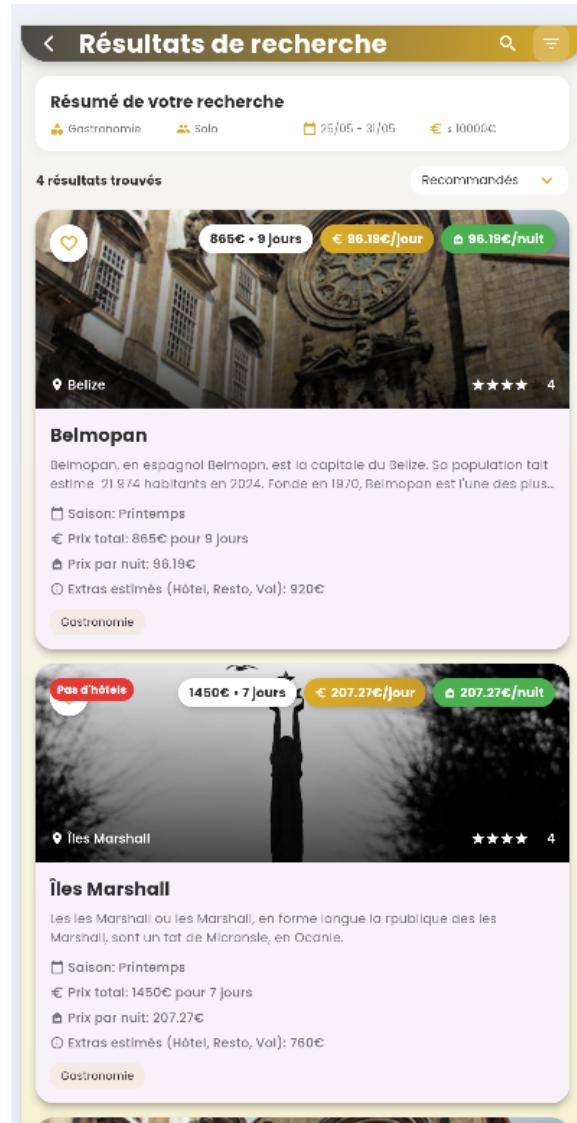


FIGURE 4.9 – Interface de consultation des destinations

- L'interface affiche les résultats du voyage avec une liste de destinations, les prix, la durée, les notes et un bouton pour voir plus d'informations comme les hôtels, les activités et le budget

The screenshot displays a list of three hotel options in Beirut:

- Hôtel Beyrouth**: \$143.00 par nuit. Includes a night view of a modern hotel complex and a "Réserver" button.
- Inn Beyrouth**: \$270.00 par nuit. Includes a daytime exterior view of a building with teal-framed windows and a "Réserver" button.
- Auberge Beyrouth**: \$75.00 par nuit. Includes a decorative sign for "AUBERGE ALSACIENNE HOTEL RESTAURANT" and a "Réserver" button.

FIGURE 4.10 – Interface de réservation d'hôtel

The screenshot displays a list of flight options to Beirut:

- Tunisair**: Direct flight, €687.00. Includes a photo of a Tunisair airplane in flight and a "Réserver" button.

FIGURE 4.11 – Interface de réservation d'un vol

- Les interfaces permettent au client de réserver un hôtel, un restaurant et un vol pour sa destination de voyage.

4.5.3 Les interfaces du budget



FIGURE 4.12 – Interface d'estimation du budget

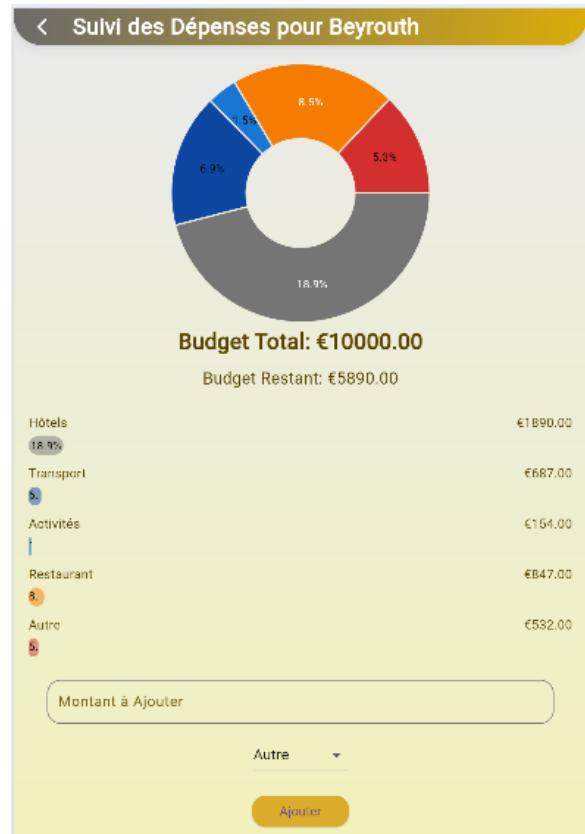


FIGURE 4.13 – Interface de suivi des dépenses

- L'interface **Estimation du budget** affiche une estimation du budget pour un séjour, avec une répartition des dépenses par catégorie et le montant restant disponible.
- L'interface **Suivi des dépenses** permet de suivre les dépenses effectuées, de visualiser leur répartition par catégorie et d'ajouter des nouvelles dépenses au budget.

4.5.4 Les interfaces de la Omra et du Hajj

The interface is a mobile application for pilgrimage planning. It features a yellow header bar with the title "Planifier un Pèlerinage". Below the header, there are several input fields and buttons:

- Type de Pèlerinage:** A dropdown menu set to "Umrah".
- Type d'Umrah:** A dropdown menu set to "Confort".
- Visiter Madinah:** A radio button group where "Oui" (Yes) is selected.
- Durée à Madinah (jours):** A date picker showing "5".
- Durée à Makkah (jours):** A date picker showing "7".
- Durée totale du voyage (jours):** A text input field showing "12 jours".
- Rechercher:** A large yellow button at the bottom.

FIGURE 4.14 – Interface de planification du Hajj et Omra

- Interface de planification de pèlerinage permettant de configurer le type d'Omra, la visite de Madinah et les durées de séjour à la Mecque et la Madinah.

Sélection des Hôtels et Vols

Saison
Basse saison

Type d'Omrah
Confort

Hôtels à Makkah

- Swissôtel Al Maqam
Prix par nuit: 400 €
- Pullman ZamZam
Prix par nuit: 250 €
- Hilton Suites Makkah
Prix par nuit: 300 €
- Raffles Makkah Palace
Prix par nuit: 450 €
- Jabal Omar Hyatt Regency
Prix par nuit: 280 €
- Makkah Marriott Hotel
Prix par nuit: 320 €
- Conrad Makkah
Prix par nuit: 380 €
- InterContinental Dar Al Tawhid
Prix par nuit: 410 €
- Al Rajhi Towers
Prix par nuit: 150 €
- Elaf Kinda Hotel
Prix par nuit: 230 €

Vols

- Tunisair - vol 1
Prix: 450 €
- Saudi Airlines - Vol 2
Prix: 600 €
- Turkish Airlines - Vol 3
Prix: 550 €
- Qatar Airways - Vol 4
Prix: 650 €
- EgyptAir - Vol 5
Prix: 500 €
- Emirates - Vol 6
Prix: 700 €
- Royal Jordanian - Vol 7
Prix: 620 €
- Lufthansa - Vol 8
Prix: 580 €
- Air France - Vol 9
Prix: 610 €
- Etihad Airways - Vol 10
Prix: 670 €
- British Airways - Vol 11
Prix: 640 €
- Alitalia - Vol 12
Prix: 590 €

Activités suggérées

- Tawaf autour de la Kaaba**
Coût : 0 €
Circonvolution autour de la Kaaba
- Sa'i entre Safa et Marwah**
Coût : 0 €
Marche entre les collines
- Visite de la Grotte de Hira**
Coût : 10 €
Site historique
- Prière à la Mosquée du Prophète**
Coût : 0 €
Prière dans la mosquée sacrée
- Visite du Mont Uhud**
Coût : 5 €
Lieu de la bataille historique
- Visite du cimetière Al-Baqi**
Coût : 0 €
Cimetière des compagnons

Valider la Sélection

Voir l'Estimation du Budget

FIGURE 4.15 – organisation de voyage Omra

— Interface de sélection de voyage Omra avec activités, vols et hôtels à La Mecque.

Confirmation de la Réservation

Visa

Oui

Agence de Visa

Makkah Tours

Mode de Visa

Express

Coût du visa : 170.00 € Délai : 2 jours

Confirmer la Réservation

FIGURE 4.16 – Interface de confirmation de réservation

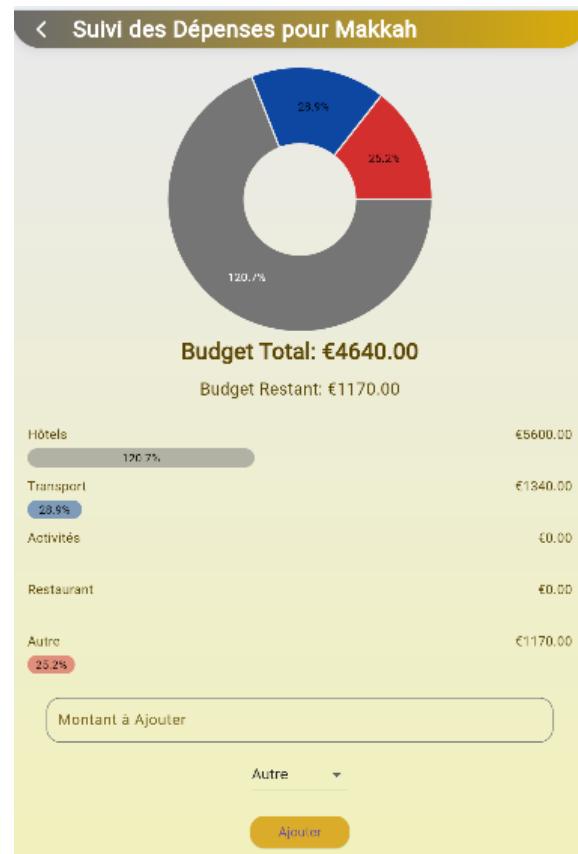


FIGURE 4.17 – Interface de suivi budgétaire

- L'interface **confirmation de réservation** finale avec sélection des services visa, choix de l'agence partenaire et validation du mode de traitement .
- L'interface **suivi budgétaire** détaillé pour le voyage à La Mecque avec graphique circulaire des dépenses et répartition par catégories (hôtels, transport, activités).

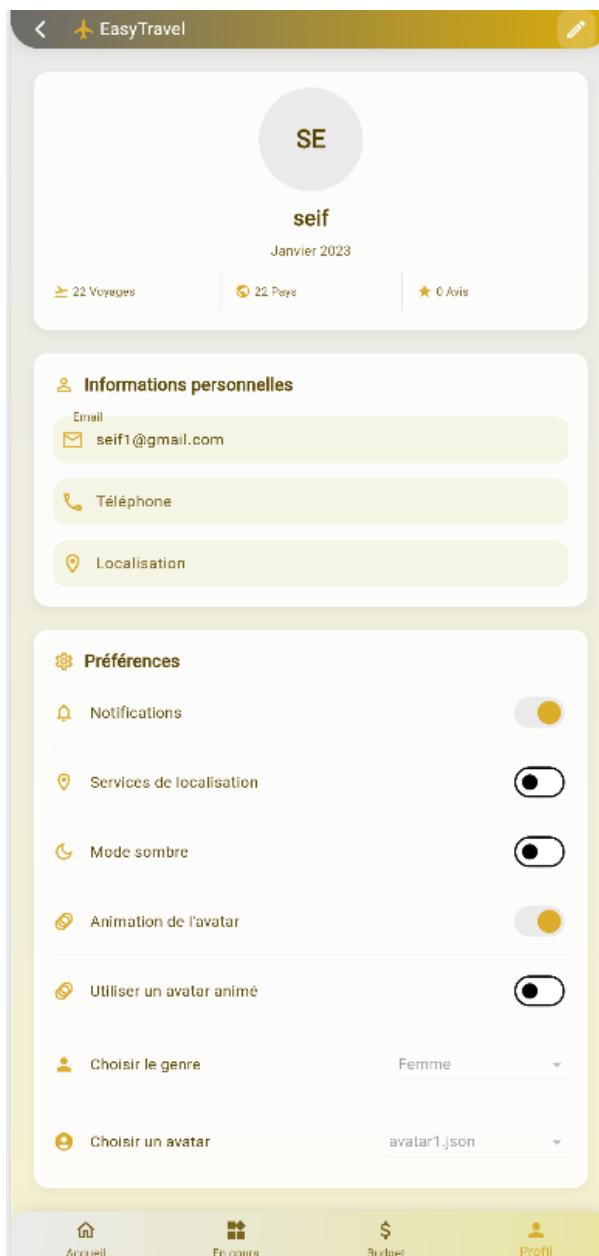


FIGURE 4.18 – Interface de profil

- Interface de profil client affichant les informations personnelles, statistiques de voyage et paramètres de préférences avec options de notifications et personnalisation.

4.6 Conclusion

Ce sprint nous a permis de mettre en place les premières fonctionnalités principales pour les utilisateurs. Nous avons conçu et réalisé des interfaces simples et efficaces pour la consultation des offres, la recherche par destination et l'affichage des détails.

Grâce à ce travail, l'application devient plus intuitive et commence à offrir une expérience utilisateur agréable. Les schémas, diagrammes et l'API développés nous aident à structurer le projet pour les prochaines étapes.

Sprint 3 : Validation des voyages et consultation du tableau de bord

5.1 Introduction

Ce chapitre présente le troisième sprint du projet Easy Travel, qui se concentre sur la mise en place de l'interface administrateur et la finalisation des fonctionnalités du système. Nous détaillerons le Sprint Backlog, suivie de l'analyse et de la conception à travers des diagrammes d'activités et de séquence. Enfin, nous aborderons les étapes de réalisation et la conclusion de ce sprint.

5.2 Sprint Backlog

Tâche ID	Description	Responsable	Estimation (jours)
T1	Mise en place de l'interface administrateur	Frontend Developer	2
T2	Implémentation des fonctionnalités de gestion des utilisateurs	Backend Developer	3
T3	Développement des outils de gestion des réservations	Backend Developer	4
T4	Finalisation des tests de l'interface	Frontend Developer	2
T5	Documentation de l'interface administrateur	Frontend Developer	1

TABLE 5.1 – Sprint Backlog détaillé – Sprint 3

5.3 Analyse

5.3.1 Raffinement de cas d'utilisation « Valider les Voyages »

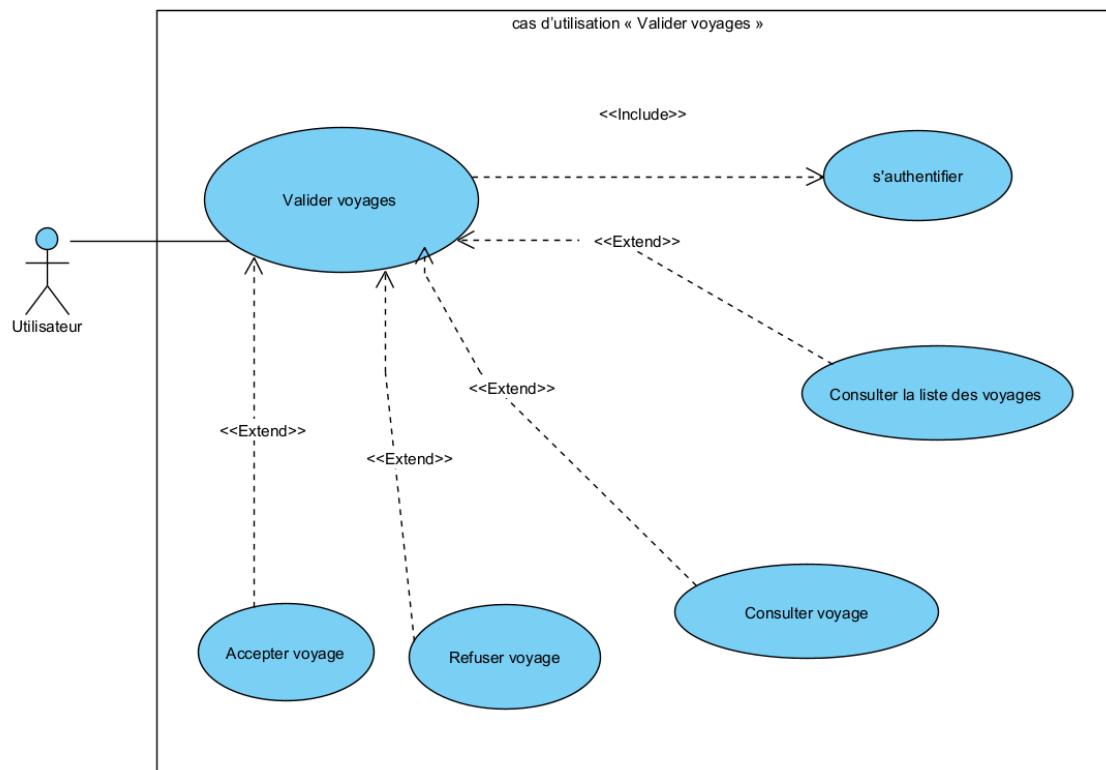


FIGURE 5.1 – Raffinement de cas d'utilisation «Valider voyage»

Acteur	Utilisateur
Objectif	Permettre à l'utilisateur de valider les voyages via différentes opérations (consulter, valider, refuser).
Précondition	L'utilisateur doit être authentifié.

Scénario No-nominal	<ol style="list-style-type: none"> 1. L'utilisateur s'authentifie via l'application. 2. Il accède à l'interface de gestion des voyages. 3. Le système affiche les options disponibles : <ul style="list-style-type: none"> — Consulter la liste des voyages — Consulter un voyage — Valider un voyage — Refuser un voyage 4. L'utilisateur sélectionne une des actions. 5. Le système exécute l'action choisie et affiche un message de confirmation.
Postcondition	L'action de gestion de voyage (consultation, validation, modification ou annulation) est réalisée avec succès.
Scénario Alternatif	<ul style="list-style-type: none"> — Si l'authentification échoue, l'accès est refusé. — Si l'action de gestion échoue, un message d'erreur est affiché.

TABLE 5.2 – Description Textuelle de Cas d'utilisation « Valider les Voyages »

5.3.2 Raffinement de cas d'utilisation « Consulter les tableaux de bord »

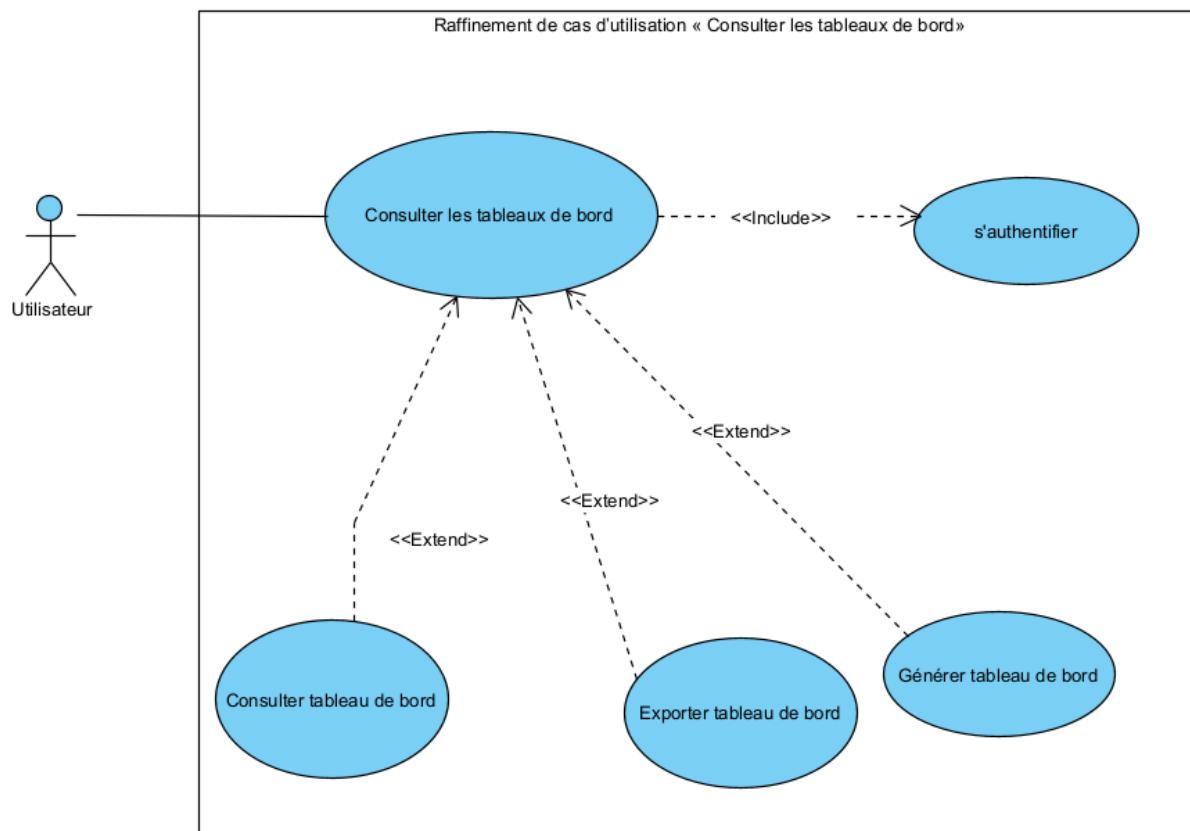


FIGURE 5.2 – Raffinement de cas d'utilisation «Consulter les tableaux de bord»

Acteur	Utilisateur
Objectif	Permettre à l'utilisateur de consulter les tableaux de bord, les générer et les exporter.
Précondition	L'utilisateur doit être authentifié.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'utilisateur s'authentifie via l'application. 2. Il accède à l'interface de consultation des tableaux de bord. 3. Le système propose les options suivantes : <ul style="list-style-type: none"> — Générer un tableau de bord — Exporter un tableau de bord — Consulter un tableau de bord 4. L'utilisateur sélectionne une option. 5. Le système exécute l'action sélectionnée et affiche un retour correspondant (visualisation ou confirmation).

Postcondition	L'utilisateur a pu consulter, générer ou exporter un tableau de bord selon son choix.
Scénario Alternatif	<ul style="list-style-type: none"> — Si l'authentification échoue, l'accès est refusé. — Si l'exportation échoue, un message d'erreur est affiché. — Si la génération échoue, un message d'erreur est affiché.

TABLE 5.3 – Description Textuelle de Cas d'utilisation « Consulter les tableaux de bord »

5.4 Conception

5.4.1 Diagramme d'activité «Consulter les tableaux de bord»

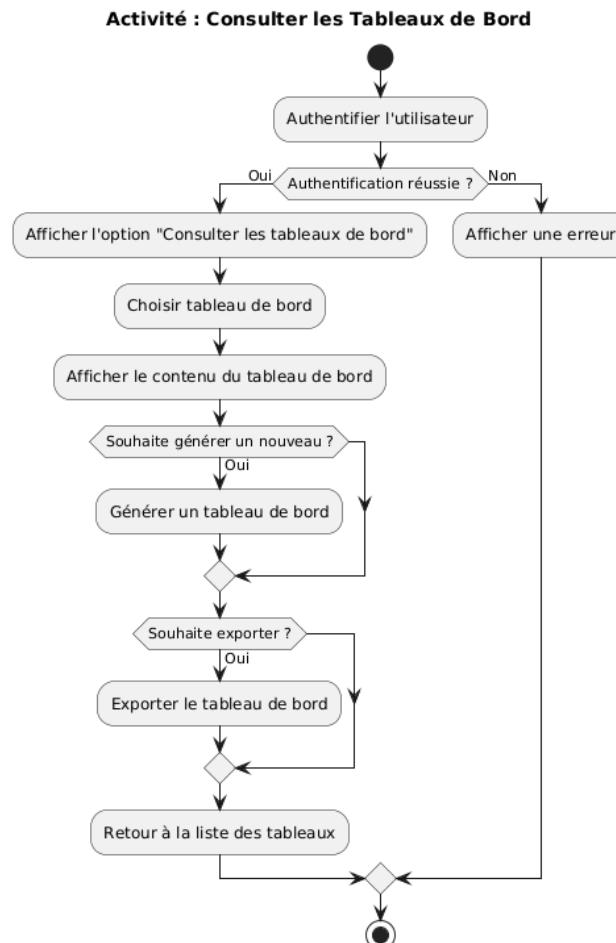


FIGURE 5.3 – Diagramme d'activité «Consulter les tableaux de bord»

5.4.2 Diagramme de séquence « Consulter les tableaux de bord »

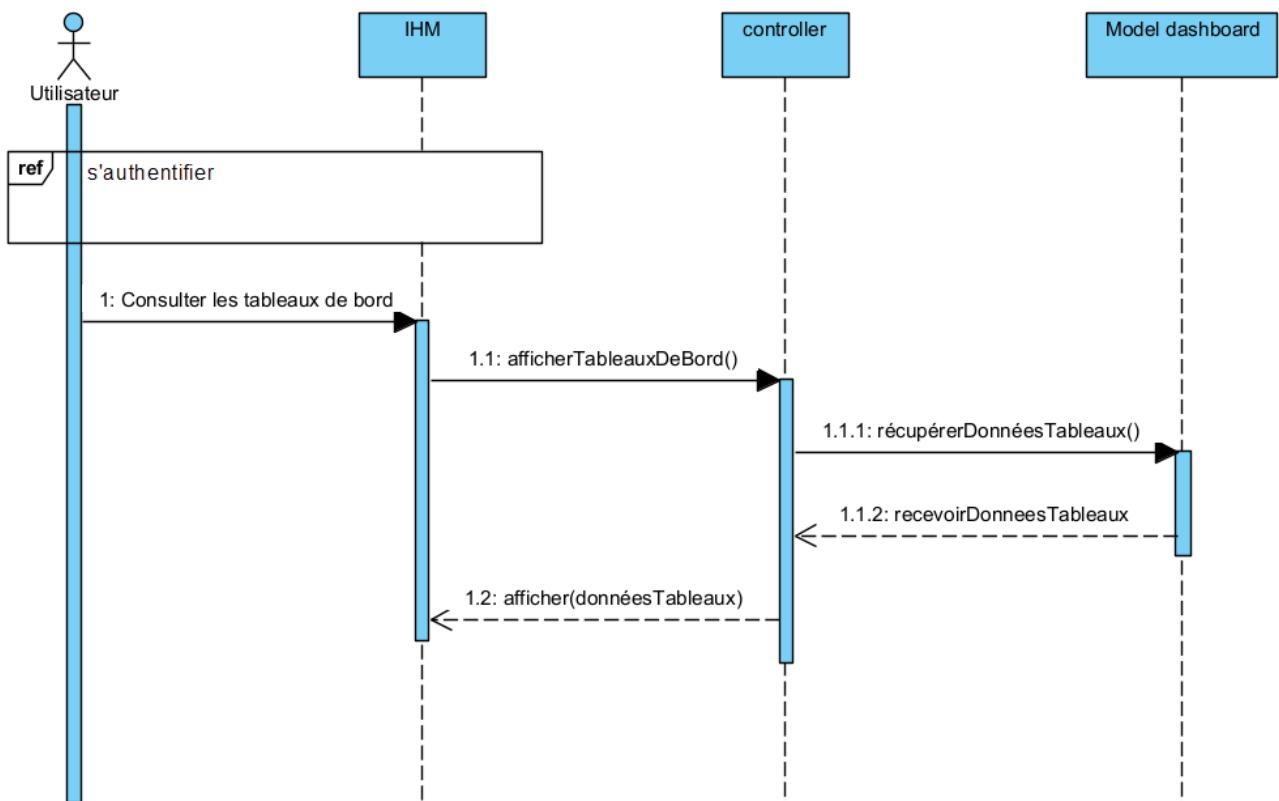


FIGURE 5.4 – Diagramme de séquence « Consulter les tableaux de bord »

5.4.3 Diagramme de séquence « Accepter voyage »

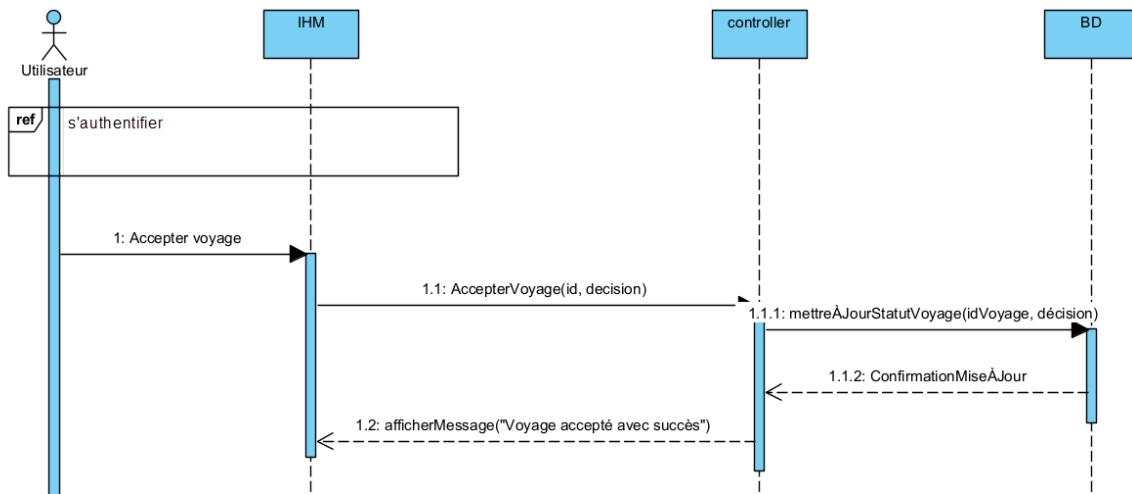


FIGURE 5.5 – Diagramme de séquence « Accepter voyage »

5.4.4 Schéma relationnel

Voici les tables des bases de données qui concerne cette sprint :

- Administrateur (Email,motDePasse,idAdmin)

5.5 Réalisation

5.5.1 Interface de page d'accueil du tableau de bord administrateur



FIGURE 5.6 – Interface de page d'accueil du tableau de bord administrateur

- Cette interface sert de page d'accueil du tableau de bord administrateur. Elle présente des statistiques essentielles telles que le nombre de destinations disponibles, les réservations effectuées et les voyages en cours, ainsi qu'un graphique illustrant les tendances des réservations. Elle offre également un accès à la gestion des utilisateurs et d'autres sections administratives via le menu latéral.

5.5.2 Interface de gestion des clients

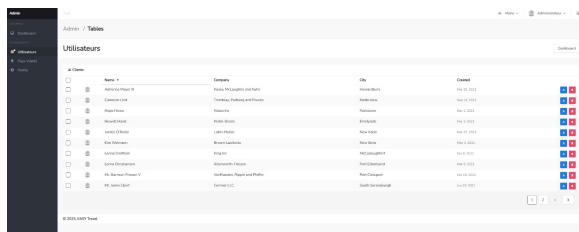


FIGURE 5.7 – Interface de gestion des clients

- Cette interface présente une liste des utilisateurs avec leurs informations principales, permettant de consulter, modifier ou supprimer chaque profil.

5.6 Conclusion

En conclusion, ce projet a permis de développer une interface administrateur fonctionnelle pour la gestion des utilisateurs et des réservations. Les tâches du sprint ont été réalisées avec succès, et l'interface a été testée et documentée correctement. Grâce à une approche structurée et à un travail d'équipe, nous avons pu atteindre les objectifs fixés pour cette phase du projet.

Conclusion générale

La fidélisation des utilisateurs dans le secteur du tourisme est essentielle face à une concurrence croissante. Dans ce contexte, Easy Travel a été conçu pour offrir une plateforme de gestion de réservations et de recommandations de destinations, intégrant un algorithme intelligent basé sur les données des utilisateurs pour suggérer des voyages personnalisés.

Ce projet a permis de développer une solution complète, de l'analyse des besoins à la mise en œuvre d'une interface administrateur permettant de gérer les utilisateurs. L'algorithme de recommandation, intégré dans le deuxième sprint, personnalise l'expérience en fonction des préférences et de l'historique de réservation des utilisateurs.

Le projet a rencontré plusieurs défis techniques, notamment lors de l'implémentation de l'algorithme et des tests de performance. Toutefois, nous avons abouti à une plateforme fonctionnelle qui facilite la gestion des réservations et améliore l'expérience utilisateur.

Perspectives d'Amélioration :

Plusieurs évolutions peuvent être envisagées pour enrichir la plateforme :

- **Système de paiement en ligne sécurisé** : Permettre la finalisation des réservations directement sur la plateforme.
- **Personnalisation avancée des recommandations** : Intégrer des critères supplémentaires, comme les avis des utilisateurs.
- **Application mobile** : Offrir une expérience pratique sur mobile.
- **Ajout d'une interface conversationnelle** : Utiliser un chatbot avec reconnaissance vocale pour améliorer l'interaction.

En conclusion, ce projet a permis de mettre en œuvre une solution innovante et complète pour le secteur du tourisme, tout en ouvrant la voie à des futures améliorations pour répondre aux besoins des utilisateurs et aux évolutions du marché.

Bibliographie

- [1] La Méthode SCRUM. *C'est quoi la méthode SCRUM?* Disponible en ligne : <https://ignition-program.com/>. Consulté le 3 mai 2025.
- [2] Product backlog - What is it how to create one. *A healthy product backlog is much like a healthy human.* Disponible en ligne : <https://www.atlassian.com/agile/scrum/backlogs>. Consulté le 5 mai 2025.
- [3] Architecture trois tiers. *Architecture logicielle en trois couches.* Disponible en ligne : https://fr.wikipedia.org/wiki/Architecture_trois_tiers. Consulté le 7 mai 2025.
- [4] Modèle/Vue/Contrôleur (MVC). *Architecture logicielle MVC en Java Swing.* Disponible en ligne : <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html>. Consulté le 9 mai 2025.
- [5] Diagramme de déploiement. *Représentation des composants logiciels dans l'infrastructure physique.* Disponible en ligne : https://fr.wikipedia.org/wiki/Diagramme_de_d%C3%A9ploiement. Consulté le 11 mai 2025.
- [6] Visual Paradigm. *Modélisation UML et conception logicielle.* Disponible en ligne : <https://www.visual-paradigm.com/>. Consulté le 13 mai 2025.
- [7] Android Studio. *Environnement de développement intégré pour Android.* Disponible en ligne : <https://developer.android.com/studio>. Consulté le 14 mai 2025.
- [8] Figma. *Outil de design collaboratif pour interfaces utilisateurs.* Disponible en ligne : <https://www.figma.com/>. Consulté le 16 mai 2025.
- [9] Visual Studio Code. *Éditeur de code source léger et puissant.* Disponible en ligne : <https://code.visualstudio.com/>. Consulté le 17 mai 2025.
- [10] Postman. *Plateforme pour tester et gérer les APIs.* Disponible en ligne : <https://www.postman.com/>. Consulté le 19 mai 2025.
- [11] Git. *Système de gestion de versions distribué.* Disponible en ligne : <https://git-scm.com/>. Consulté le 20 mai 2025.
- [12] JSON (JavaScript Object Notation). *Format léger d'échange de données facile à lire et écrire.* Disponible en ligne : <https://www.json.org/json-fr.html>. Consulté le 21 mai 2025.

- [13] Vue.js. *Framework JavaScript progressif pour construire des interfaces utilisateur*. Disponible en ligne : <https://vuejs.org/>. Consulté le 23 mai 2025.
- [14] Laravel. *Framework PHP pour le développement d'applications web*. Disponible en ligne : <https://laravel.com/>. Consulté le 24 mai 2025.
- [15] Flutter. *Framework UI open source de Google pour construire des applications natives*. Disponible en ligne : <https://flutter.dev/>. Consulté le 26 mai 2025.
- [16] Python. *Langage de programmation interprété, polyvalent et facile à apprendre*. Disponible en ligne : <https://www.python.org/>. Consulté le 28 mai 2025.
- [17] LLaMA (Large Language Model Meta AI). *Modèle de langage développé par Meta*. Disponible en ligne : <https://ai.facebook.com/blog/large-language-model-llama-meta-ai/>. Consulté le 30 mai 2025.

Annexe

Dans le cadre de ce projet, un algorithme a été conçu pour automatiser le traitement de requêtes API liées à l'extraction et à la mise en forme de données. L'objectif principal était de faciliter l'interaction avec une base de données externe via une interface RESTful. L'approche adoptée permet à la fois une gestion efficace des requêtes entrantes, un contrôle des erreurs, et une réponse structurée selon les besoins de l'utilisateur.

L'algorithme repose sur une architecture modulaire permettant d'assurer une scalabilité et une maintenance aisées. L'utilisation de bibliothèques modernes de traitement des API et de structures de données assurent également robustesse et performance.

Description de l'Algorithme de Recommandation de Voyage

Présentation Générale

Dans le cadre de ce projet, nous avons développé une API intelligente pour la planification de voyages personnalisés. L'algorithme repose sur l'analyse des préférences utilisateur (budget, type de voyage, activités souhaitées, période, etc.) pour générer automatiquement des recommandations de destinations, hôtels, restaurants, vols et activités.

Ce système intègre plusieurs technologies de traitement de données, de machine learning et d'analyse sémantique. Il fonctionne de manière modulaire pour permettre une extension simple et une maintenance efficace.

Fonctionnement Général de l'Algorithme

Le fonctionnement de l'algorithme peut être résumé en plusieurs étapes principales :

1. **Réception des données utilisateur** via une interface API (budget, préférences, dates, etc.).
2. **Nettoyage et traitement des données** (normalisation, vérification de la cohérence des dates, orthographe, etc.).
3. **Calcul des coûts estimés** par poste (hôtel, vol, restaurant, activités).

4. **Filtrage intelligent** du dataset de destinations en fonction du budget, des préférences et des contraintes temporelles.
5. **Génération de recommandations** hiérarchisées selon un score de pertinence.
6. **Sélection des services associés** (hôtel, restaurant, vol) compatibles avec le budget restant.
7. **Création automatique d'un itinéraire quotidien** optimisé selon les activités sélectionnées.
8. **Récapitulatif final** avec tous les éléments du voyage et possibilité de confirmation.

Méthodes et Modules Utilisés

- **Pandas** : traitement des datasets (fichiers CSV) contenant les informations sur les destinations, hôtels, restaurants, vols et activités.
- **Scikit-learn (LinearRegression)** : utilisé pour l'estimation de coûts futurs ou la prédiction de tendances si nécessaire.
- **FuzzyWuzzy** : traitement de la similarité textuelle pour la reconnaissance des préférences ou des destinations proches.
- **PySpellChecker** : correction automatique des fautes d'orthographe dans les textes entrés.
- **SpaCy (fr_core_news_sm)** : traitement du langage naturel pour extraire les préférences depuis des phrases complètes.
- **Flask + Flask-CORS** : création de l'API Web RESTful, permettant d'interagir avec le système de recommandation.
- **JSON** : gestion de l'état du voyage (budget, choix, itinéraire) à travers un fichier dynamique `current_trip.json`.

Exemple de Traitement

Lorsque l'utilisateur soumet une requête avec les paramètres :

- Budget : 2000 €
- Type de voyage : Couple
- Préférence : Nature
- Dates : 1er au 10 août 2025

L'algorithme :

1. Détermine la saison (Été), la durée (10 jours), et le nombre de personnes (2).
2. Calcule le budget par poste (hébergement, repas, transport, activités).
3. Filtre les destinations disponibles avec des coûts compatibles.
4. Classe les résultats par pertinence (coût ajusté, affinité avec les préférences, saisonnalité).
5. Retourne jusqu'à 5 recommandations personnalisées

Exemple des parties de l'algorithme et de dataset

```
# Charger les datasets
dataset = pd.read_csv("voyage_dataset_full.csv")
hotels = pd.read_csv("hotels.csv")
restaurants = pd.read_csv("restaurants.csv")
flights = pd.read_csv("flights.csv", keep_default_na=False)
activities = pd.read_csv("activities_dataset.csv")
```

FIGURE 8 – Chargement et Préparation des Données

```
def get_saison(date_str):
    try:
        date = datetime.strptime(date_str, "%Y-%m-%d")
        month = date.month
        if 3 <= month <= 5:
            return "Printemps"
        elif 6 <= month <= 8:
            return "Été"
        elif 9 <= month <= 11:
            return "Automne"
        else:
            return "Hiver"
    except ValueError:
        return "Été"
```

FIGURE 9 – Détermination de la saison en fonction de la date

```
def generer_recommandations(budget, preference, type_voyage, saison, duree, start_date=None, end_date=None, destination=None):
    personnes = get_nombre_personnes(type_voyage)
    budget_total = budget
    facteur_saison = {"Printemps": 1.1, "Été": 1.2, "Automne": 0.9, "Hiver": 1.0}

    cout_hotel_moyen_par_nuit = hotels["PrixParNuit"].mean() if not hotels.empty else 70
    cout_resto_moyen_par_jour = restaurants["PrixMoyen"].mean() if not restaurants.empty else 30
    cout_vol_moyen_par_personne = flights["PrixVol"].mean() if not flights.empty else 200

    cout_hotel_estime = cout_hotel_moyen_par_nuit * duree * personnes
    cout_resto_estime = cout_resto_moyen_par_jour * duree * personnes
    cout_vol_estime = cout_vol_moyen_par_personne * personnes
    cout_extras_estime = cout_hotel_estime + cout_resto_estime + cout_vol_estime

    budget_destination_max = budget_total - cout_extras_estime
    if budget_destination_max < 0:
        budget_destination_max = budget_total * 0.8

    print(f"Extras - Hôtel: {cout_hotel_estime}, Resto: {cout_resto_estime}, Vol: {cout_vol_estime}, Total: {cout_extras_estime}")
    print(f"Budget destination max: {budget_destination_max}")

    dataset["Coût Ajusté"] = dataset["Coût"].apply(lambda x: x * facteur_saison[saison])
    dataset[["DateDebut"]] = pd.to_datetime(dataset[["DateDebut"]])
    dataset[["DateFin"]] = pd.to_datetime(dataset[["DateFin"]])

    if start_date and end_date:
        start_date_dt = datetime.strptime(start_date, "%Y-%m-%d")
        end_date_dt = datetime.strptime(end_date, "%Y-%m-%d")
        date_filter = (
            (dataset["DateDebut"] <= end_date_dt) & (dataset["DateFin"] >= start_date_dt)
        )
    else:
        date_filter = pd.Series([True] * len(dataset))
```

FIGURE 10 – Exemple de Fonction principale 'Génération de Recommandations'

```

1 Destination,Restaurant,TypeCuisine,PrixMoyen,Étoiles
2 Kaboul,Café Kaboul 24,Méditerranéenne,109,1
3 Kaboul,Chez Kaboul 64,Française,58,2
4 Kaboul,Chez Kaboul 29,Indienne,107,2
5 Kaboul,Café Kaboul 34,Locale,95,4
6 Kaboul,Bistro Kaboul 38,Indienne,41,5
7 Le Cap,Restaurant Le Cap 62,Française,33,3
8 Le Cap,Café Le Cap 69,Méditerranéenne,51,3
9 Le Cap,Restaurant Le Cap 63,Asiatique,120,2
10 Johannesburg,Restaurant Johannesburg 51,Méditerranéenne,42,5
11 Johannesburg,Bistro Johannesburg 97,Indienne,58,5
12 Durban,Café Durban 80,Gastronomique,135,2
13 Durban,Café Durban 1,Locale,126,5

```

FIGURE 11 – Exemple de dataset Restaurant

```

Destination,Hôtel,Étoiles,PrixParNuit,Catégorie
Kaboul,Hôtel Kaboul 10,5,301,Économique
Kaboul,Palace Kaboul 99,1,54,Économique
Kaboul,Inn Kaboul 80,1,222,Confort
Kaboul,Palace Kaboul 90,2,274,Confort
Kaboul,Auberge Kaboul 26,4,134,Économique
Le Cap,Hôtel Le Cap 2,4,403,Luxe
Le Cap,Résidence Le Cap 7,1,500,Confort
Le Cap,Résidence Le Cap 43,1,327,Économique
Le Cap,Résidence Le Cap 9,4,142,Luxe
Johannesburg,Résidence Johannesburg 51,1,144,Luxe
Johannesburg,Hôtel Johannesburg 14,5,363,Luxe
Johannesburg,Hôtel Johannesburg 87,2,388,Luxe
Johannesburg,Hôtel Johannesburg 85,3,113,Confort
Johannesburg,Auberge Johannesburg 29,4,341,Luxe

```

FIGURE 12 – Exemple de dataset Hotel

Conclusion

Ce système constitue un moteur de recommandation multi-critères pour la planification de voyages personnalisés. Il exploite des données variées et adapte dynamiquement les suggestions en fonction du profil et des contraintes de l'utilisateur.