**PFA**

Présenté par:

**Intégration de la Technologie dans la Travel Booking Web App:**

**Developpement d'une Solution sur Mesure**

**Encadré par :**

**Année universitaire:**

2024/2025



**Dédicace**

**Remerciement**

**Sommaire**

**Liste des figures**

**Introduction Générale**

Le secteur du voyage et du tourisme est confronté à de nombreux défis, notamment en termes de gestion des réservations et de coordination des différents services comme les vols, les hôtels, et les locations de voitures. Dans ce contexte, le projet de développement d'une application web de réservation de voyages vise à remédier à ces lacunes en offrant une plateforme centralisée et intuitive qui permet aux utilisateurs de réserver facilement et rapidement tous les aspects de leur voyage.

Le système actuel de réservation de voyages est souvent fragmenté, avec des plateformes distinctes pour les vols, les hôtels, et les voitures de location. Cette fragmentation peut entraîner une mauvaise expérience utilisateur, des erreurs de réservation, et une inefficacité dans la gestion des itinéraires de voyage. Ces défis peuvent diminuer la satisfaction des clients et entraver la croissance des entreprises dans le secteur du tourisme.

L'objectif principal du projet d'application web de réservation de voyages est de concevoir et de mettre en œuvre une plateforme centralisée qui simplifie le processus de réservation pour les utilisateurs tout en intégrant plusieurs services de voyage. Cette application vise à améliorer l'expérience utilisateur en offrant une interface conviviale, une gestion efficace des réservations, et une communication fluide entre les utilisateurs et les prestataires de services. Le projet ambitionne également d'optimiser l'efficacité opérationnelle des entreprises partenaires en centralisant les données de réservation et en automatisant les processus.

La méthodologie du projet implique d'abord une analyse approfondie des besoins des utilisateurs et des lacunes existantes dans les systèmes de réservation actuels. Ensuite, une conception et un développement informatique sont réalisés pour créer une application intégrée répondant aux exigences identifiées. Une fois l'application mise en œuvre, des tests et des ajustements sont effectués pour garantir son bon fonctionnement et son adaptation aux besoins spécifiques des utilisateurs.

L'application web de réservation de voyages a permis de simplifier les processus de réservation, d'améliorer la coordination entre les différents services de voyage, et de faciliter la gestion des itinéraires par les utilisateurs. Grâce à son interface intuitive et à ses fonctionnalités avancées, elle a également contribué à une amélioration de la satisfaction des clients et à une augmentation des réservations.

En conclusion, le projet d'application web de réservation de voyages a démontré son efficacité en améliorant la gestion des réservations et en optimisant l'expérience utilisateur. Il constitue un exemple concret de l'impact positif que les technologies de l'information peuvent avoir sur le secteur du tourisme. À l'avenir, une expansion et une évolution continues de cette application pourraient encore renforcer son potentiel à transformer et à améliorer les pratiques de réservation dans l'industrie du voyage.

Le contenu de ce rapport est organisé comme suit:

**Chapitre 1**

**Presentation generale:**

**“** L'étape de présentation du projet englobe la définition claire des objectifs visés, établis à travers un mind mapping pour visualiser la structure et les liens du projet. Un planning prévisionnel de réalisation, soutenu par un diagramme de Gantt, offre une vue chronologique des tâches à accomplir. La méthodologie adoptée guide l'approche de travail, tandis que les technologies sélectionnées définissent les outils et les plateformes utilisés. Une étude comparative des technologies justifie ces choix en examinant leurs avantages et inconvénients, garantissant ainsi une mise en œuvre appropriée et efficace du projet ”

**1.1 Introduction:**

Le premier chapitre du rapport débute par une présentation détaillée, suivie de la définition claire des objectifs. Un mind mapping est élaboré pour visualiser les différentes composantes du projet de manière hoistique. Un planning prévisionnel est établi, puis représenté sous forme de diagramme de Gantt pour une gestion efficace du temps. La méthodologie de travail est exposée, accompagnée de la spécification des technologies utilisées. Une étude comparative des technologies est réalisée pour justifier les choix opérés.

* 1. **Présentation du projet**:

Cette étape consiste à fournir une vue d'ensemble du projet d'application web de réservation de voyages, en décrivant le contexte, les acteurs impliqués, et les motivations derrière sa mise en place. Elle comprend également une analyse des besoins et des problématiques auxquels le système doit répondre, ainsi qu'une présentation des principales fonctionnalités envisagées.

* 1. **Objectifs du projet:**
     1. **Problimatique:**

La gestion des données est un critère essentiel pour toute entreprise, notamment dans le secteur du tourisme, où il peut s'agir de la gestion des réservations, des disponibilités de vols, d'hôtels, ou de voitures de location. Pour ce projet, nous avons opté pour le développement d'une application web de réservation de voyages qui permettra de gérer l'ensemble des réservations et des services associés à travers une interface simple et pratique. Cette gestion sera bien sûr traitée à travers une base de données stockée localement ou sur un serveur sécurisé.

Le problème se pose principalement au niveau de la gestion des réservations, surtout lorsque le nombre d'utilisateurs, de services disponibles, et de transactions augmente. Cela devient de plus en plus délicat de traiter toutes ces données manuellement et de les stocker de manière efficace. C'est ici que l'informatisation des données devient un élément nécessaire. Le traitement automatique via une application web améliore énormément l'efficacité opérationnelle des entreprises, que ce soit en termes de gain de temps ou de gestion des ressources humaines choisies pour effectuer les tâches.

* + 1. **Objectifs et Solution:**

On va réaliser une application qui permet de gérer les services suivants :

* Service d’Inscription (Gestion des comptes avec rôles)
* Service de Gestion des Vols.
* Service de Gestion Voitures.
* Service de Gestion des Hôtels.
* Service de Gestion des reservation Hôtels.
* ervice de Gestion des reservation Voitures.
* Service de Gestion des reservation Vols.
* Service de paiement via Stripe
  1. **Carte mentale :**

La carte mentale établit les rôles distincts de chaque acteur (client, responsable, administrateur) au sein du système de réservation en ligne, en détaillant leurs responsabilités et les fonctionnalités auxquelles ils ont accès.

* **Rôle de client:**

Réservation : Les Clients peuvent rechercher et réserver des vols, des hôtels et des voitures via l'application. Ils ont la possibilité de consulter et de gérer leurs propres réservations, y compris de faire des modifications ou des annulations. o Confirmation de Paiement :

Les Clients peuvent confirmer leurs réservations via un processus de paiement sécurisé intégré avec Stripe. Cela garantit que les transactions sont traitées de manière efficace et sécurisée.

Pages Authentifiées : Les Clients ont accès à des pages authentifiées où ils peuvent consulter des listes détaillées et des informations, y compris :

Page Détails : Informations détaillées sur chaque voiture, hôtel et vol.

Page À Propos : Informations sur l'application et ses fonctionnalités.

Page Contact : Informations de contact pour le support et les demandes.

Page Équipe : Informations sur l'équipe derrière l'application.

Gestion des Réservations : Les Clients peuvent gérer leurs réservations avec des opérations CRUD (Création, Lecture, Mise à jour, Suppression) pour leurs propres réservations.

* **Rôle du responsable:**

Création et Gestion des Données : Les Responsables peuvent créer et gérer les données qu'ils ont saisies (vols, hôtels, voitures). Ils peuvent effectuer des opérations CRUD (Création, Lecture, Mise à jour, Suppression) sur leurs propres soumissions.

Consultation des Données : Les Responsables peuvent consulter toutes les données pertinentes associées à leurs propres entrées, mais ne peuvent pas modifier ou supprimer les données créées par d'autres utilisateurs.

Consultation des Réservations : Les Responsables peuvent consulter les réservations associées aux vols, hôtels et voitures qu'ils gèrent. Cette fonctionnalité leur permet de suivre et de superviser les réservations liées à leurs données.

* **Rôle de l’administrateur:**

Opérations CRUD pour Responsables : L'Admin peut effectuer des opérations de Création, Lecture, Mise à jour et Suppression (CRUD) pour les Responsables, y compris la gestion et la modification des comptes Responsables.

Gestion des Données : Les Admins ont l'autorité pour consulter (Lire) toutes les entrées de données sur la plateforme, y compris les vols, les hôtels et les voitures. Ils peuvent superviser toutes les données mais n'effectuent pas directement des opérations CRUD sur les entrées de vols, d'hôtels ou de voitures.

Consultation des Réservations : Les Admins peuvent consulter les réservations sur la plateforme, y compris celles gérées par les Responsables. Cela permet une vue d'ensemble complète de toutes les réservations.

* 1. **Planning prévisionnel de réalisation:**

**1.5.1Definition du diagramme de Gantt:**

Diagramme de Gantt: est une représentation graphique des tâches d'un projet, affichée sur une ligne de temps. Chaque tâche est représentée par une barre horizontale dont la longueur correspond à sa durée. Les tâches sont listées verticalement, tandis que l'échelle de temps est affichée horizontalement.

**1.5.2 Fonctionnalités:**

**Visualisation claire du planning:** Il offre une représentation visuelle simple et compréhensible de la chronologie des tâches, ce qui facilite la compréhension de l'ordre des activités et des délais associés.

**Gestion du temps:** Il permet de planifier et de suivre l'avancement des tâches dans le temps, en identifiant les périodes de chevauchement ou les goulets d'étranglement.

**Identification des dépendances:** Les dépendances entre les tâches sont clairement indiquées, ce qui permet de comprendre quelles tâches doivent être terminées avant que d'autres ne puissent commencer.

**Allocation des ressources:** En visualisant le calendrier des tâches, les chefs de projet peuvent mieux allouer les ressources disponibles en fonction des besoins de chaque tâche.

**Communication:** Le diagramme de Gantt est un outil de communication efficace. Il permet aux membres de l'équipe et aux parties prenantes de comprendre rapidement l'état d'avancement du projet et les prochaines étapes.

**Gestion des risques:** En identifiant les tâches critiques et les potentielles zones de retard, le diagramme de Gantt aide à anticiper les problèmes et à prendre des mesures correctives.

**Flexibilité:** Bien que le diagramme de Gantt fournisse une structure temporelle initiale pour le projet, il est également flexible. Les modifications peuvent être apportées au planning en fonction des changements de priorités, des retards ou des nouveaux besoins.

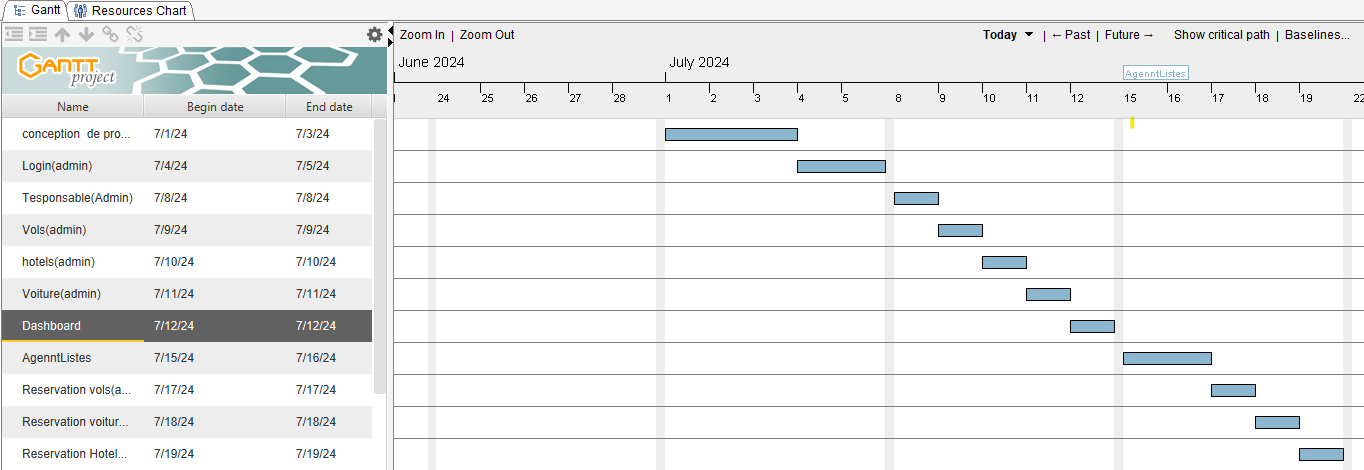
Voici une planification de projet pour le développement de l'application web de réservation de voyages, utilisant un diagramme de Gantt, mettant en évidence ldurées des tâches et les tâches importantes :

|  |  |  |
| --- | --- | --- |
| Taches | P(x) | Durée en jour |
| Conception de project | -- | 3 |
| Login(admin) | Conception de project | 2 |
| Responsables (admin) | login | 1 |
| Vols(admin) | Login+responsable+Vols | 2 |
| Voiture(admin) | Login+responsable+Voiture | 2 |
| Dashboard(Admin) | Login+ Vols+Voiture+Hotels | 1 |
| Hotels(Admin) | Login+responsable+hoteles | 1 |
| AgentListes(Admin) | Login+responsable Vols+Voiture+Hotels | 2 |
| Reservation Vols (Admin) | Login+responsable+client Vols+Voiture+Hotels | 1 |
| Reservation Hotels(Admin) | Login+responsable+client Vols+Voiture+Hotels | 1 |
| Reservation Voiture(Admin) | Login+responsable+client Vols+Voiture+Hotels | 1 |
| Login(Responsable) | Conception de project | 1 |
| Dashboard(Responsable) | Login+Vols+Voiture+Hotels | 1 |
| Voiture (Responsable) | Login | 3 |
| Vols (Responsable) | Login | 3 |
| Hotels(Responsable) | login | 3 |
| Reservation Vols (Responsable) | Login+client+vols | 3 |
| Reservation Voiture(Responsable) | Login+client+voiture | 3 |
| Reservation Hotels(Responsable) | Login+client+hotels | 3 |
| Login(client) | Conception de project | 1 |
| Page vols + details(client) | Login + vols | 2 |
| Page voiture+ details (client) | Login + voiture | 2 |
| Page Hotels+ details (client) | Login + Hotels | 2 |
| Reservation hotels | login | 2 |
| Reservation Voiture | login | 2 |
| Reservation vols | login | 2 |
| Autre page(about contact+team+error404)  +Stripe | -- | 6 |

**1.5.3 Utilisation de Gantt Project:**

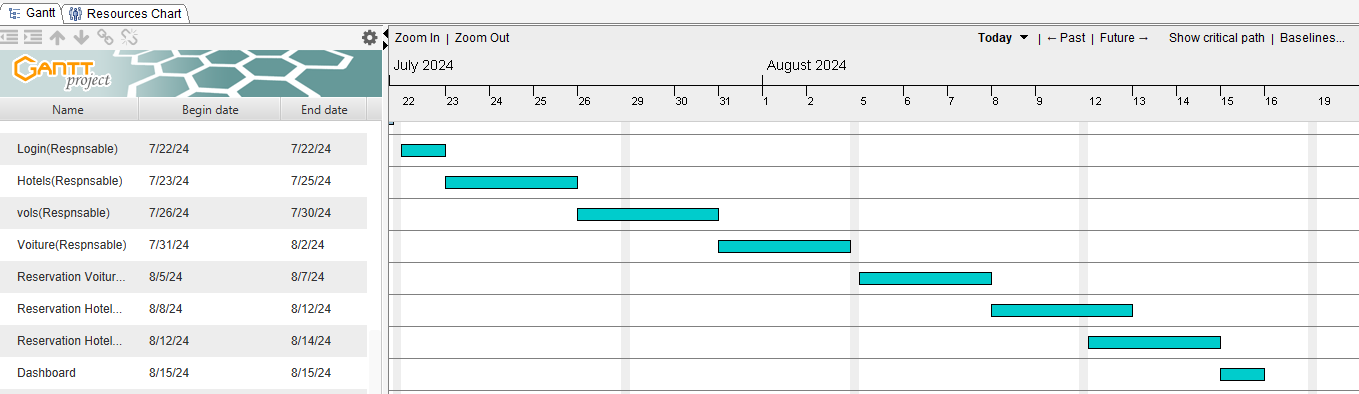
GanttProject est un logiciel de gestion de projet open source qui permet de créer des diagrammes de Gantt, de suivre les tâches, de gérer les ressources et de générer des rapports. C'est une excellente option pour ceux qui recherchent un outil simple mais puissant pour planifier et suivre leurs projets.

**Les pages admin:**



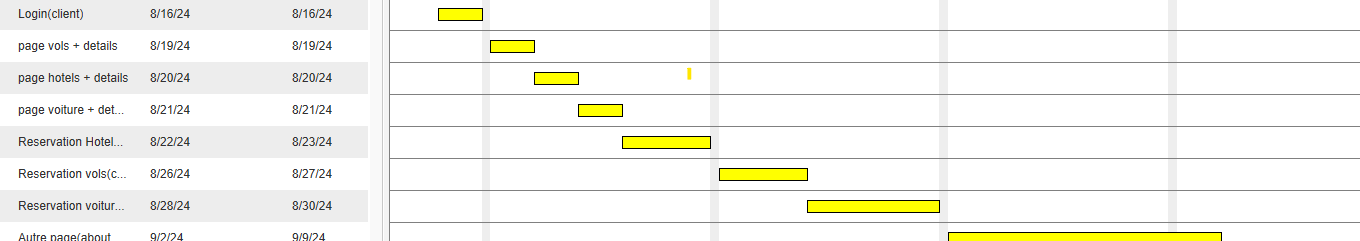
*Figure 1.1: Les taches de l’admin*

**Les pages Responsable:**

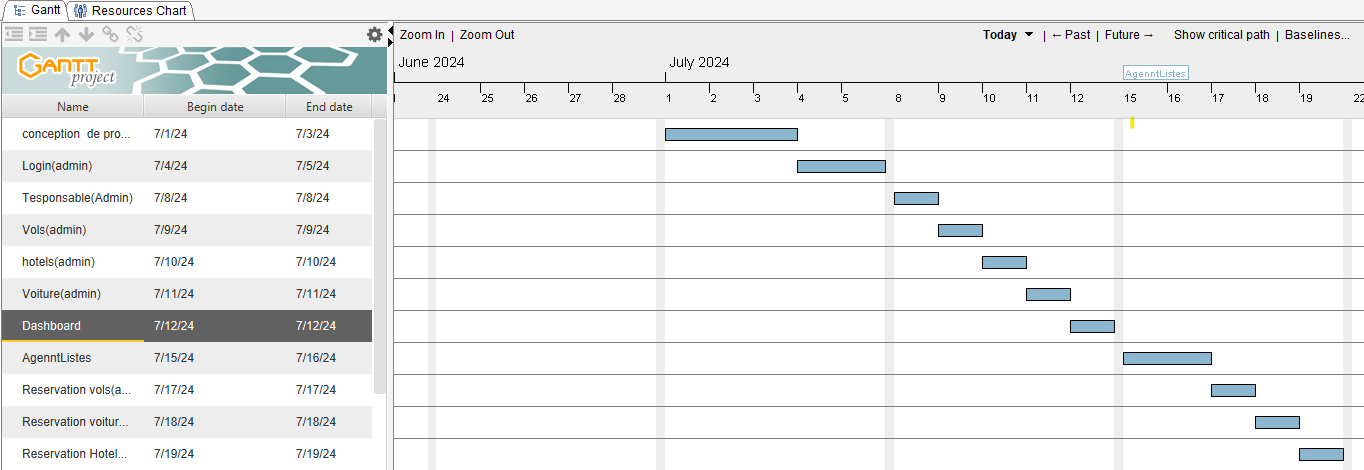
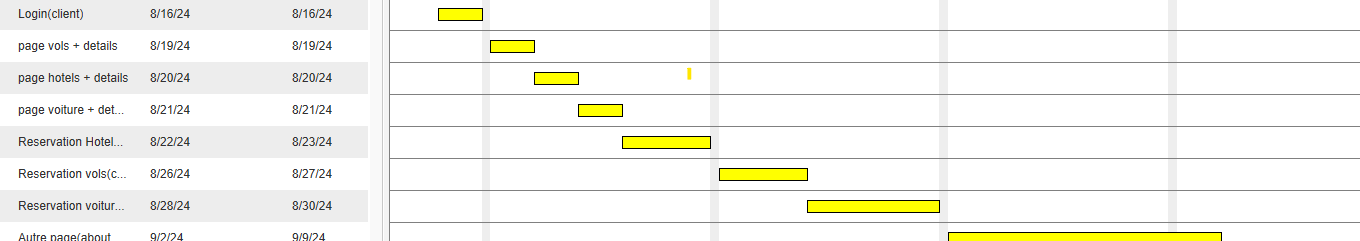
****

*Figure 1.2: Les taches de responsable*

**Les pages client:**



*Figure 1.3: Les taches de client*

**Capture En Générale:** ****

*Figure 1.4: Les taches de Chaque Acteurt*

* 1. **Méthodologie adoptee:** 
     1. **Méthode en cascade:**

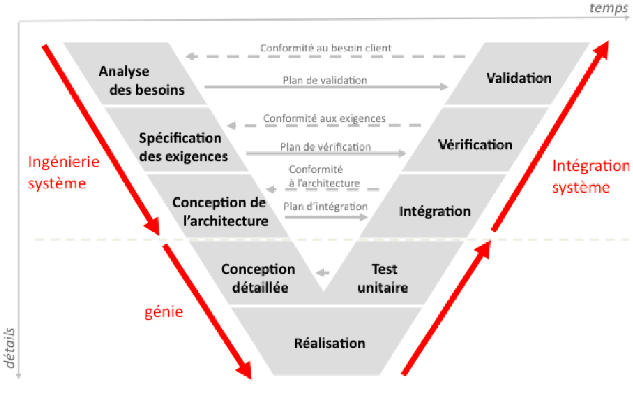
La méthode en cascade est un modèle linéaire de développement de logiciels, où chaque phase doit être complétée avant que la suivante puisse commencer. Par exemple, on termine d'abord l'analyse des besoins, puis la conception, puis le développement, puis les tests, etc. C'est moins flexible que l'approche Agile, car les changements sont difficiles à intégrer une fois qu'une phase est terminée.



*Figure 1.5: La Methodologie Cascade*

* + 1. **Méthode Cycle en V:**

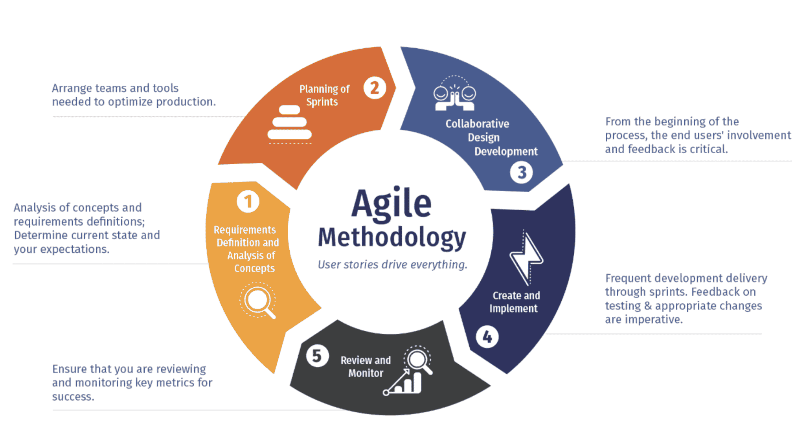
Ce modèle de développement logiciel est caractérisé par une approche séquentielle. Il comprend des phases distinctes, comme l'analyse des exigences, la conception, la réalisation des tests et la maintenance. Chaque phase est planifiée en avance et doit être complétée avant que la suivante puisse commencer. C'est souvent utilisé dans les projets où les exigences sont bien comprises et relativement stables.

****

*Figure 1.6: La Methodologie Cycle en V*

* + 1. **Approche agile:**

L'approche Agile est un ensemble de valeurs et de principes qui met l'accent sur la collaboration, l'adaptabilité et la livraison de valeur client continue. Contrairement au modèle en V, l'approche Agile est itérative et incite à des livraisons fréquentes et incrémentales. Les équipes Agile s'adaptent rapidement aux changements et accueillent le feedback continu des clients pour améliorer le produit.

****

*Figure 1.7: L'approche Agile*

En Choisis l'approche Agile peut être bénéfique pour plusieurs raisons, en fonction des besoins spécifiques de votre projet et de votre organisation:

**Adaptabilité aux changements:** L'Agilité met l'accent sur la flexibilité et la capacité à s'adapter rapidement aux changements. Dans un environnement où les exigences du projet évoluent fréquemment ou où l'incertitude est élevée, l'approche Agile permet de répondre efficacement aux nouvelles priorités et aux retours d'information des parties prenantes.

**Livraisons incrémentales de valeur:** Avec l'approche Agile, les fonctionnalités sont développées et livrées de manière itérative, permettant ainsi une mise en production plus rapide et une rétroaction plus précoce des utilisateurs finaux. Cela garantit que le produit final répond aux besoins réels des utilisateurs et génère de la valeur dès les premières étapes du projet.

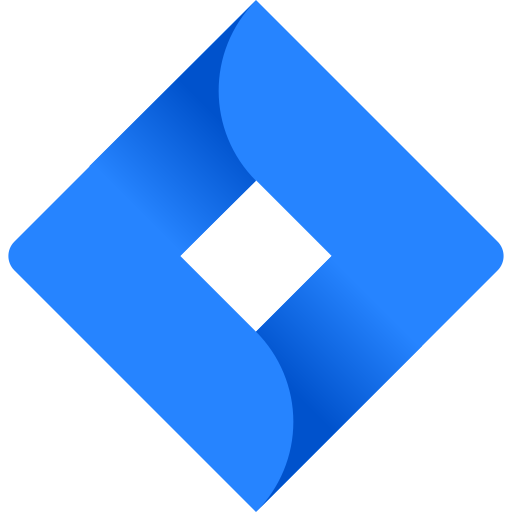
**Collaboration accrue:** L'Agilité favorise une collaboration étroite entre les membres de l'équipe de développement, les parties prenantes et les clients. Les cycles de développement courts encouragent la communication régulière et la transparence, ce qui conduit à une meilleure compréhension des besoins et des défis du projet.

**Motivation et engagement de l’équipe:** Les équipes Agile sont encouragées à s'auto-organiser et à prendre des décisions collectives, ce qui favorise un sentiment de responsabilité et d'engagement envers le projet. Les cycles de travail courts et la possibilité de voir les résultats de leur travail rapidement peuvent également renforcer la motivation des membres de l'équipe.

**Réduction des risques:** En adoptant une approche itérative et incrémentale, les risques du projet sont identifiés et atténués plus tôt dans le processus de développement. Les échecs potentiels sont découverts rapidement, ce qui permet à l'équipe de prendre des mesures correctives avant qu'ils ne deviennent des problèmes majeurs.

* 1. **Organisation du Travail:**
     1. **Platform jira:**

JIRA est une plateforme logicielle populaire utilisée pour la gestion de projets, le suivi des problèmes et le suivi du travail d'équipe. Il est principalement utilisé dans un contexte Agile, mais peut également être adapté à d'autres méthodologies de gestion de projet.



*Figure 1.8: Logo du Platform jira*

voici les raison pourquoi en utilisant la plateforme jira:

**Gestion des tâches:** JIRA permet de créer, attribuer et suivre les tickets de travail. Ces tickets peuvent représenter des tâches, des bogues, des améliorations ou d'autres éléments du projet.

**Tableaux Kanban et Scrum:** JIRA propose des tableaux Kanban et Scrum intégrés pour visualiser et gérer le flux de travail. Les tableaux Kanban permettent de suivre visuellement les tâches à travers différentes étapes, tandis que les tableaux Scrum sont conçus pour les itérations planifiées.

**Personnalisation:** JIRA est hautement personnalisable pour s'adapter aux besoins spécifiques de votre équipe et de votre projet. Vous pouvez définir des flux de travail, des champs de ticket, des permissions d'accès et plus encore.

**Intégrations:** JIRA offre une large gamme d'intégrations avec d'autres outils et services, tels que des outils de développement logiciel (comme Bitbucket et GitHub), des outils de communication (comme Slack) et des outils de gestion de portefeuille de projets.

**Reporting et suivi des performances:** JIRA fournit des fonctionnalités de reporting pour suivre les performances du projet, y compris les délais, les problèmes ouverts, la vélocité de l'équipe, et plus encore. Ces rapports aident à identifier les tendances, les goulots d'étranglement et les domaines à améliorer.

* + 1. **Fréquence de reunion:**

**Fréquence de réunion et canaux de communication:** Des réunions quotidiennes auront lieu en début de journée pour mettre à jour l'état d'avancement et identifier les obstacles. En outre, des réunions de planification de sprint auront lieu à la fin de chaque sprint. Nous utiliserons principalement Slack pour la communication asynchrone et Zoom pour les réunions synchrones.

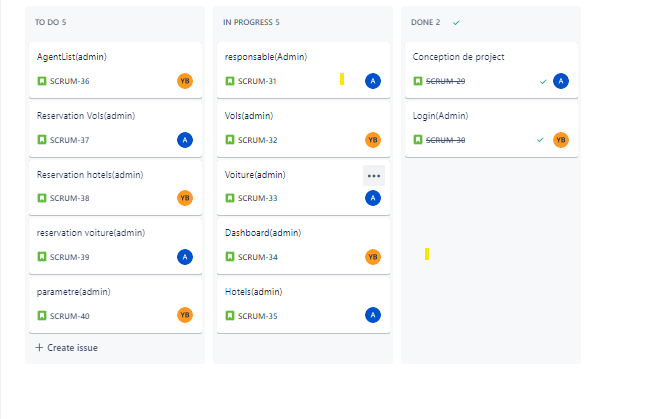
* + 1. **Test:**

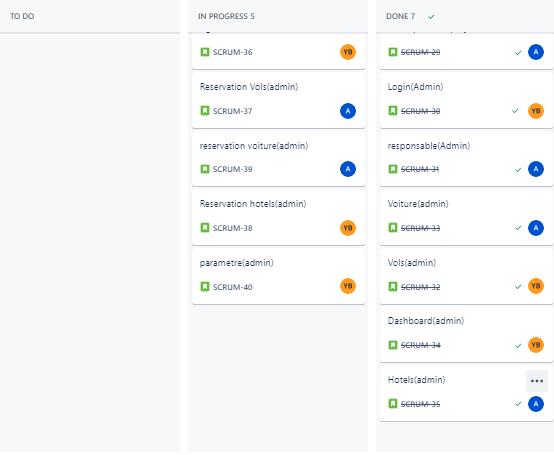
Nous adopterons une approche de développement pilotée par les tests (TDD) pour garantir la qualité du code. Des tests unitaires, d'intégration et de validation seront écrits pour chaque fonctionnalité développée.

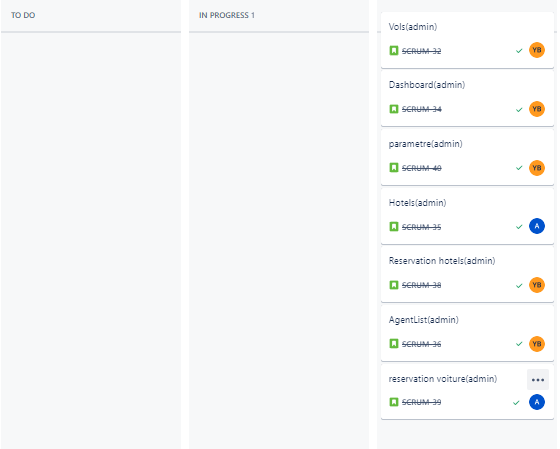
* + 1. **TAF Division des taches:**

Pour diviser les tâches du projet par équipe, en commencer par identifier les différentes phases du projet (analyse, conception, développement, test, etc.). Ensuite, assignez chaque phase à une personne en fonction de ses compétences et spécialisations. Assurez-vous également de répartir équitablement la charge de travail et de clarifier les responsabilités de chaque équipe. Utilisez des outils de gestion de projet pour suivre la progression et faciliter la communication entre les équipes

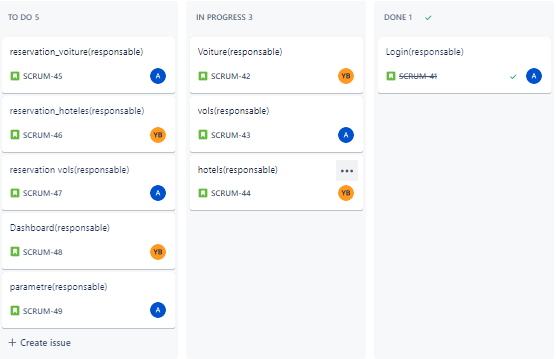
**les taches Admin :**

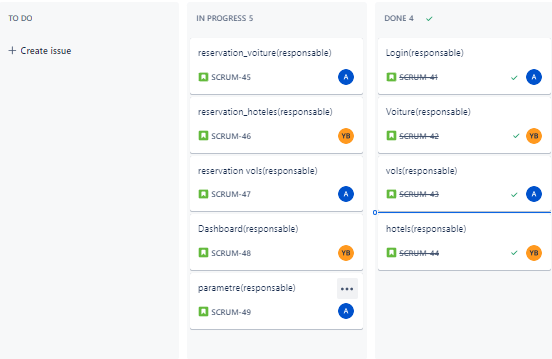


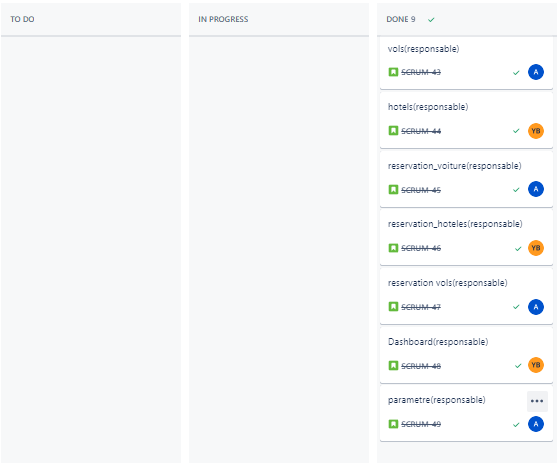




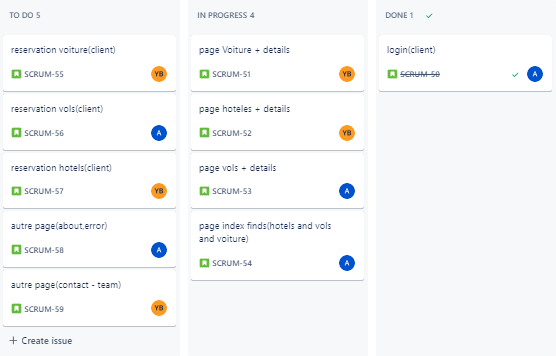
**les taches Responsable:**

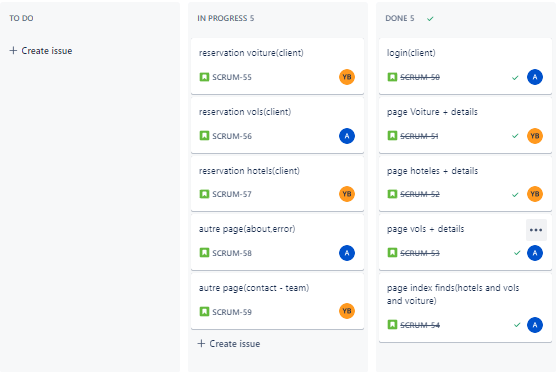


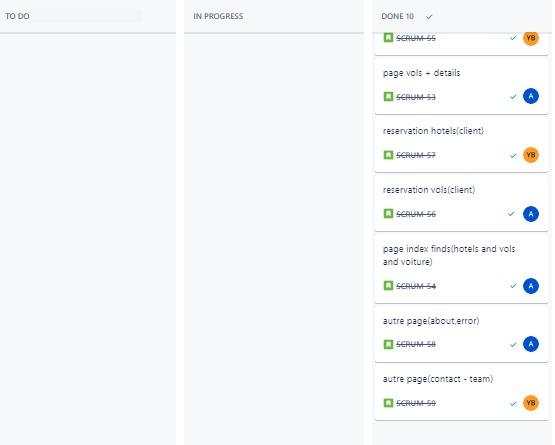




**les taches client:**







**Chapitre 2**

**Presentation generale:**

**“L’étape** "État de l’art" implique une revue exhaustive des connaissances actuelles et des technologies disponibles dans le domaine du projet. Cela comprend une recherche bibliographique approfondie, une évaluation des technologies existantes, et l'analyse d'études de cas. Les conclusions sont synthétisées dans un rapport détaillé présentant les technologies recommandées. Les résultats sont ensuite validés et ajustés en fonction des retours des parties prenantes, fournissant une base solide pour les phases suivantes du projet.”

**2.1 Etude de l’existent:**

L'étude de l'existant pour notre projet visant à développer une application web de réservation de voyages implique une analyse approfondie et méthodique des systèmes et plateformes de réservation actuels. Cette phase commence par une évaluation des outils et des solutions en place pour la gestion des réservations de vols, d'hôtels et de voitures.

Pour les utilisateurs, l'étude se concentre sur les outils utilisés pour rechercher et réserver des services de voyage, consulter les détails des réservations, accéder aux informations sur les services disponibles, et gérer leurs profils. Pour les gestionnaires de services, l'analyse examine les systèmes utilisés pour gérer les disponibilités des services, suivre les réservations, coordonner les offres, et gérer les tarifs.

Les administrateurs, quant à eux, utilisent des outils pour superviser l'ensemble du système de réservation. Cela inclut la génération de rapports sur les performances des services, la gestion des comptes utilisateurs (clients et fournisseurs), et la surveillance des transactions et des paiements.

Cette phase initiale est suivie par une évaluation détaillée des processus actuels de gestion des réservations. Cela implique de documenter comment les utilisateurs accèdent actuellement aux informations de réservation, comment les gestionnaires coordonnent les services, et comment les administrateurs supervisent les opérations. L'identification des besoins des utilisateurs se fait à travers des consultations, des entretiens, et des questionnaires pour comprendre leurs attentes et les défis rencontrés avec les systèmes en place.

L'analyse des flux de données et des infrastructures techniques est également cruciale. Elle consiste à examiner comment les données sont collectées, stockées, sécurisées, et partagées. Cette évaluation inclut également les capacités des serveurs, la bande passante internet, et les protocoles de sauvegarde des données pour garantir la fiabilité et la sécurité du système.

**2.1.1 Avantages:**

* **Compréhension approfondie :** L'analyse détaillée des systèmes et plateformes de réservation actuels permet une compréhension complète des besoins et des défis. Cela facilite la conception d'une solution sur mesure qui répond précisément aux attentes des utilisateurs et des gestionnaires.
* **Optimisation des ressources :** En identifiant les inefficacités et les redondances dans les systèmes existants, l'étude permet d'optimiser l'utilisation des ressources disponibles et d'améliorer les processus opérationnels.
* **Amélioration de l'expérience utilisateur :** Impliquer les utilisateurs dans l'identification de leurs besoins garantit que le nouveau système de réservation répondra à leurs attentes. Cela contribue à améliorer leur satisfaction et favorise une adoption plus rapide et efficace de la nouvelle plateforme.

**2.1.2 Inconvénients:**

* **Temps et coût élevés :** La réalisation d'une étude exhaustive des systèmes existants demande du temps et peut entraîner des coûts significatifs, ce qui peut impacter le budget et le calendrier du projet.
* **Résistance au changement :** Les utilisateurs peuvent être réticents à adopter un nouveau système de réservation, même s'il est amélioré, ce qui peut compliquer la transition et nécessiter des efforts supplémentaires pour assurer une adoption réussie.
* **Complexité de la transition :** La mise en œuvre progressive des nouvelles fonctionnalités nécessite une gestion minutieuse pour éviter les perturbations dans les opérations de réservation et garantir une transition fluide sans impact négatif sur les utilisateurs ou les opérations.

**2.2 Solution Proposées:**

Pour notre projet voici des solutions proposées pour chaque groupe d'utilisateurs (client, Responsable, administrateurs) en se basant sur l'analyse de l'existant et les besoins identifiés :

**2.2.1 Solutions pour les client:**

* **Accès aux Informations Personnelles :** Un tableau de bord personnalisé permettant aux clients de consulter et de mettre à jour leurs informations personnelles.
* **Gestion des Réservations :** Une interface pour consulter les options disponibles (vols, hôtels, voitures) et gérer leurs réservations.
* **Historique des Réservations :** Un système sécurisé pour que les clients puissent consulter leurs réservations passées et suivre leurs réservations actuelles.
* **Calendrier des Réservations :** Un calendrier interactif affichant les dates de réservation, les voyages à venir et d'autres événements importants.
* **Profil Client :** Une section où les clients peuvent voir et éditer leurs informations personnelles, de réservation et de paiement

**2.2.2 Solutions pour les professeurs:**

* Gestion des Réservations : Une interface pour gérer et voir les réservations qu'ils ont créées, avec des outils pour modifier ou annuler les réservations.
* Tableau de Bord de Performance : Un outil pour suivre les métriques de performance liées aux réservations qu'ils gèrent.
* Accès aux Informations des Clients : Accès aux détails des clients ayant effectué des réservations via leurs services.
* Gestion des Inventaires : Outils pour gérer la disponibilité des vols, hôtels et voitures, y compris l'ajout, la mise à jour et la suppression des annonces.
* Commentaires et Avis : Une section pour gérer et répondre aux commentaires et avis des clients.

**2.2.3 Solutions pour les administrateurs:**

* **Gestion des Comptes Responsables :**
* **CRUD (Créer, Lire, Mettre à Jour, Supprimer) :** L'administrateur peut créer, consulter, modifier ou supprimer les comptes des Responsables.
* **Affichage en Lecture Seule :**
* **Données Générales :** L'administrateur peut consulter toutes les données (y compris la liste des agents, les avis, les réservations) en mode lecture seule.
* **Réservations :** Consultation des réservations de vols, hôtels et voitures en mode lecture seule.
* **Hôtels et Vols :** Visualisation des informations concernant les hôtels et les vols en mode lecture seule.
* **Modification des Comptes Administrateurs :**
* **Gestion des Comptes Admin :** L'administrateur peut modifier les comptes des autres administrateurs.

**2.3 Les besoins fonctionnels:**

**2.3.1 la base de données:**

La base de données du système de gestion scolaire assure l'authentification sécurisée des utilisateurs autorisés, leur permettant d'accéder aux fonctionnalités en toute sécurité. Elle gère également la lecture, l'écriture et la mise à jour des données conformément aux autorisations définies, garantissant ainsi l'intégrité et la confidentialité des informations stockées. De plus, elle est capable d'exécuter des requêtes complexes pour récupérer des données spécifiques, répondant ainsi aux besoins variés des utilisateurs. En cas d'erreurs ou d'exceptions liées à l'accès à la base de données, des mécanismes de gestion appropriés sont en place pour assurer la stabilité et la continuité du système.

**2.3.2 Back-end:**

Dans le système de notre projet, la gestion des requêtes HTTP entrantes est essentielle pour diriger chaque acteur vers les fonctionnalités qui leur sont pertinentes :

Pour les **administrateurs**, des routes spécifiques sont définies :

* **Connexion Administrateur :**
* GET /admin : Permet à l'administrateur de se connecter au système.
* POST /admin : Authentifie l'administrateur.
* **Tableau de Bord :**
* GET /dashboard : Affiche un tableau de bord complet avec des options de gestion.
* **Gestion des Responsables :**
* GET /responsables : Affiche la liste des responsables.
* GET /add-responsable : Affiche le formulaire pour ajouter un responsable.
* POST /add-responsable : Ajoute un nouveau responsable.
* GET /edit-responsable/:id : Affiche le formulaire pour modifier un responsable.
* PUT /edit-responsable/:id : Modifie les détails d'un responsable.
* DELETE /edit-responsable/:id : Supprime un responsable.
* **Gestion des Clients :**
* GET /admin-clients : Affiche la liste des clients.
* **Gestion des Hôtels :**
* GET /admin-Hoteles : Affiche la liste des hôtels.
* GET /hotelBooking : Affiche les réservations d'hôtels.
* **Gestion des Vols :**
* GET /admin\_flights : Affiche la liste des vols.
* GET /admin\_flights : Affiche les réservations de vols.
* **Gestion des Voitures :**
* GET /admin-cars : Affiche la liste des voitures.
* GET /carBooking : Affiche les réservations de voitures.
* **Liste des Agents :**
* GET /agent-list : Affiche la liste des agents.
* GET /agent-list/:id : Affiche les détails d'un agent spécifique.
* **Gestion des Avis :**
* GET /reviws : Affiche les avis.
* DELETE /admin\_delete\_review/:id : Supprime un avis.
* **Déconnexion :**
* GET /logout : Déconnecte l'administrateur.

Les **Responsible** ont également leurs propres routes :

* **Connexion Responsable :**
* GET /responsable : Permet au responsable de se connecter au système.
* POST /responsable : Authentifie le responsable.
* **Tableau de Bord :**
* GET /dashboard-Responsable : Affiche le tableau de bord du responsable avec des informations sur les réservations.
* **Gestion des Réservations :**
* **Vols :**
  + GET /tickets-flights : Affiche les réservations de vols.
  + GET /add-flightReservation : Affiche le formulaire pour ajouter une réservation de vol.
  + POST /add-flightReservation : Ajoute une réservation de vol.
  + GET /edit\_flightReservation/:id : Affiche le formulaire pour modifier une réservation de vol.
  + PUT /edit\_flightReservation/:id : Modifie une réservation de vol.
  + DELETE /edit\_flightReservation/:id : Supprime une réservation de vol.
* **Hôtels :**
  + GET /Hotels : Affiche la liste des hôtels.
  + GET /add-Hotel : Affiche le formulaire pour ajouter un hôtel.
  + POST /add-Hotel : Ajoute un hôtel.
  + GET /edit-Hotel/:id : Affiche le formulaire pour modifier un hôtel.
  + PUT /edit-Hotel/:id : Modifie un hôtel.
  + DELETE /edit-Hotel/:id : Supprime un hôtel.
  + GET /reservations : Affiche les réservations d'hôtels.
  + DELETE /edit-HotelReservation/:id : Supprime une réservation d'hôtel.
* **Voitures :**
  + GET /add-car : Affiche le formulaire pour ajouter une voiture.
  + POST /add-car : Ajoute une voiture.
  + GET /cardcar : Affiche la liste des voitures.
  + GET /edite-car/:id : Affiche le formulaire pour modifier une voiture.
  + PUT /edite-car/:id : Modifie une voiture.
  + DELETE /edite-car/:id : Supprime une voiture.
  + GET /carReservationList : Affiche les réservations de voitures.
  + DELETE /edite\_carReservation/:id : Supprime une réservation de voiture.
* **Gestion des Avis :**
* GET /responsable\_reviws : Affiche les avis.
* DELETE /responsable\_delete\_review/:id : Supprime un avis.
* **Déconnexion :**
* GET /logout-responsable : Déconnecte le responsable.

Quant aux **client**, leurs routes spécifiques incluent :

* **Inscription et Connexion :**
* GET /Signup : Affiche le formulaire d'inscription.
* POST /Signup : Inscrit un nouveau client.
* GET /login : Affiche le formulaire de connexion.
* POST /login : Authentifie le client.
* **Réservations :**
* **Vols :**
  + GET /Home : Affiche les options de réservation de vols.
  + POST /flight-list : Obtient la liste des vols.
  + GET /flight-detail/:id : Affiche les détails d'un vol.
  + POST /flight-detail : Ajoute une réservation de vol.
* **Hôtels :**
  + GET /client-Hotel : Affiche la liste des hôtels.
  + POST /client-Hotel : Recherche d'hôtels.
  + GET /reserver-Hotel : Affiche le formulaire pour réserver un hôtel.
  + POST /add-ReservationClient : Ajoute une réservation d'hôtel.
* **Voitures :**
  + GET /index-car : Affiche la liste des voitures.
  + POST /client-cars : Recherche de voitures.
  + GET /add-carReservation : Affiche le formulaire pour réserver une voiture.
  + POST /add-ReservationCar : Ajoute une réservation de voiture.
* **Gestion des Réservations :**
* **Hôtels :**
  + GET /Cards-reservation : Affiche toutes les réservations d'hôtel.
  + GET /edit-HotelReservation-cleint/:id : Affiche le formulaire pour modifier une réservation d'hôtel.
  + PUT /edit-HotelReservation-cleint/:id : Modifie une réservation d'hôtel.
  + DELETE /edit-HotelReservation-cleint/:id : Supprime une réservation d'hôtel.
* **Voitures :**
  + GET /Cards-reservationCar : Affiche toutes les réservations de voitures.
  + GET /edite\_Reservation\_car/:id : Affiche le formulaire pour modifier une réservation de voiture.
  + PUT /edite\_Reservation\_car/:id : Modifie une réservation de voiture.
  + DELETE /edite\_Reservation\_car/:id : Supprime une réservation de voiture.
* **Autres Fonctionnalités :**
* GET /get-allReview : Affiche tous les avis.
* POST /add-review : Ajoute un avis.
* DELETE /edite-review/:id : Supprime un avis.
* GET /Contact : Affiche le formulaire de contact.
* POST /add\_contact : Ajoute une demande de contact.
* GET /About : Affiche des informations sur le service.
* GET /Team : Affiche les membres de l'équipe.
* GET /client\_settings : Permet de consulter et de mettre à jour les paramètres du client.
* GET /myBooking : Affiche les réservations du client.
* GET /delete : Affiche la confirmation de suppression.
* GET /deleteProfile : Supprime le profil du client.
* GET /Listing\_Hoteles : Affiche la liste de tous les hôtels.
* GET /Listing\_cars : Affiche la liste de toutes les voitures.
* GET /Hotel\_detail/:id : Affiche les détails d'un hôtel.
* GET /Car\_detail/:id : Affiche les détails d'une voiture.
* GET /logout-client : Déconnecte le client.

**2.3.3 Front-end:**

* Interface utilisateur intuitive et réactive.
* Affichage des données récupérées de la base de données de manière organisée et conviviale.
* Possibilité pour les utilisateurs d'interagir avec les données via des formulaires ou des interfaces interactives.
* Prise en charge de différentes tailles d'écran et de dispositifs pour assurer une expérience utilisateur cohérente.

**2.4 Les besoins non fonctionnels:**

**2.4.1 Performance:**

Le système est conçu avec une architecture optimisée pour des temps de réponse rapides, grâce à l'utilisation de technologies de base de données performantes et à des mécanismes d'optimisation des requêtes. Les interactions avec le back-end sont efficacement gérées grâce à des protocoles de communication rapides et fiables, garantissant une réactivité maximale des applications. De plus, sa capacité à gérer un volume élevé de transactions simultanées repose sur des mécanismes de mise à l'échelle horizontale et verticale, ainsi que sur des stratégies de gestion de la charge de travail dynamique. En ce qui concerne les bases de données NoSQL, des techniques avancées d'optimisation sont mises en œuvre pour maximiser l'efficacité de l'accès aux données, telles que l'indexation efficace, la distribution des données et l'utilisation de caches intelligents. Cette approche garantit que même dans des environnements à forte charge, le système maintient des performances élevées tout en assurant une disponibilité continue des données et une expérience utilisateur optimale.

**2.4.2 Sécurité:**

Le système intègre des mécanismes robustes d'authentification et d'autorisation afin de sécuriser les données sensibles. Des techniques avancées telles que l'authentification à deux facteurs, la gestion fine des droits d'accès et l'audit des activités sont mises en place pour garantir que seules les personnes autorisées puissent accéder aux informations sensibles. De plus, toutes les données sont chiffrées lors de leur transmission sur le réseau ainsi que lors de leur stockage, assurant ainsi une confidentialité totale des données en transit et au repos. Pour renforcer davantage la sécurité, le système est constamment surveillé pour détecter et répondre rapidement aux vulnérabilités et aux failles de sécurité. Des mises à jour régulières sont effectuées pour maintenir les défenses contre les menaces émergentes, et des audits de sécurité périodiques sont réalisés pour évaluer et renforcer la posture de sécurité globale du système. En combinant ces mesures, le système garantit la protection des données sensibles contre les menaces internes et externes, tout en offrant une tranquillité d'esprit aux utilisateurs quant à la sécurité de leurs informations.

**2.4.3 Fiabilité:**

**Le système est conçu pour offrir une disponibilité élevée, avec un temps d'arrêt minimal prévu pour la maintenance ou en cas de pannes imprévues. Des mécanismes sophistiqués de redondance et de basculement automatique sont en place pour garantir une continuité de service sans interruption. De plus, des stratégies de sauvegarde régulières et des procédures de restauration des données sont mises en œuvre pour prévenir la perte de données en cas de catastrophe ou de défaillance du système. La surveillance proactive du système est assurée par des outils avancés qui surveillent en permanence les performances, la disponibilité et la sécurité. Ces outils permettent de détecter et de résoudre rapidement les problèmes potentiels avant qu'ils n'affectent les utilisateurs, assurant ainsi une expérience utilisateur optimale et une fiabilité inébranlable du système.**

**2.4.4 Évolutivité:**

Le système est conçu avec une architecture flexible et évolutive, prête à répondre à la croissance future du nombre d'utilisateurs et du volume de données. Son infrastructure est conçue pour être facilement extensible, permettant ainsi d'ajuster les capacités en fonction des besoins changeants. Grâce à une architecture modulaire, de nouvelles fonctionnalités peuvent être intégrées de manière transparente, offrant ainsi une adaptabilité aux exigences métier émergentes. Cette approche permet au système de rester agile et de s'adapter rapidement aux évolutions du marché et aux demandes des utilisateurs, assurant ainsi sa pertinence et sa compétitivité à long terme.

**2.4.5 Convivialité:**

L'interface utilisateur est soigneusement conçue pour offrir une expérience intuitive et conviviale, avec des indications claires sur la manière d'interagir avec le système. Des principes de conception centrés sur l'utilisateur sont appliqués pour garantir une navigation fluide et une compréhension facile des fonctionnalités. De plus, le système prend en charge le multilinguisme et s'adapte à différentes cultures ou contextes d'utilisation, offrant ainsi une expérience personnalisée à chaque utilisateur, quel que soit son emplacement ou sa langue préférée. Cette approche garantit que le système est accessible et utilisable par un large éventail d'utilisateurs, favorisant ainsi son adoption et sa satisfaction globale.

**2.4.6 Compatibilité:**

Le système est conçu pour prendre en charge les navigateurs web courants ainsi que les dispositifs mobiles populaires, garantissant ainsi une accessibilité maximale à travers une variété de plateformes. Que les utilisateurs accèdent au système depuis un ordinateur de bureau, une tablette ou un smartphone, l'interface s'adapte de manière dynamique pour offrir une expérience optimale sur chaque appareil. Cette approche assure que les utilisateurs peuvent interagir avec le système de manière transparente, où qu'ils soient et quel que soit le dispositif qu'ils utilisent, favorisant ainsi une utilisation généralisée et une satisfaction utilisateur accrue.

**Chapitre 3**

**Conception et modélisation:**

**“L’étape** de conception et modélisation, nous avons défini l'architecture du système en tenant compte des exigences fonctionnelles et non fonctionnelles. Les schémas de base de données ont été élaborés pour une organisation efficace et un accès rapide aux données, avec une optimisation des modèles NoSQL. L'interface utilisateur a été soigneusement conçue à l'aide de maquettes et prototypes interactifs pour garantir ergonomie et convivialité. De plus, des diagrammes UML ont modélisé les processus métier, les flux de données et les interactions entre composants, assurant une compréhension claire et une communication efficace entre les membres de l'équipe de développement..”

**3.1 Processus de développement:**

Le processus de développement, en suivant la méthode Agile, est structuré en plusieurs phases itératives et incrémentales, permettant une adaptation continue et une amélioration progressive du produit. Voici un aperçu détaillé des différentes étapes:

**3.1.1 Planification Initiale et Définition des Exigences:**

L'équipe se réunit pour comprendre les objectifs généraux du projet et définir les exigences de base. Afin d’organiser des rendez-vous, nous utilisons l’application Discord.

**Discord**: est une plateforme de communication gratuite conçue à l'origine pour les communautés de gamers, mais qui s'est étendue pour accueillir divers types de groupes et d'organisations. Elle offre une combinaison de messagerie instantanée, de discussion vocale et vidéo, et de partage de fichiers, le tout dans une interface conviviale. Les utilisateurs peuvent rejoindre des serveurs, qui sont des espaces dédiés à des communautés spécifiques, et à l'intérieur de ces serveurs, ils peuvent accéder à différents canaux textuels et vocaux pour organiser les discussions.



*Figure 1.9: Logo Discord*

Les raisons qui nous ont poussé à choisir Discord

* **Communication en Temps Réel**: Discord offre une messagerie instantanée permettant des conversations textuelles rapides et efficaces, complétée par des appels vocaux et vidéo faciles à utiliser, que ce soit individuellement ou en groupe. De plus, le partage d'écran intégré est particulièrement utile pour les présentations, les démonstrations ou l'assistance technique, enrichissant ainsi les interactions et la collaboration au sein de la plateforme.
* **Organisation Structurée**: Discord permet des conversations textuelles rapides et efficaces via sa messagerie instantanée, et offre également la possibilité de passer des appels vocaux et vidéo, que ce soit individuellement ou en groupe. De plus, la fonctionnalité de partage d'écran est particulièrement utile pour les présentations, les démonstrations ou l'assistance technique, enrichissant ainsi les interactions et facilitant la collaboration sur la plateforme.
* **Accessibilité et Flexibilité**: Discord facilite des conversations textuelles rapides et efficaces avec sa messagerie instantanée et permet de passer des appels vocaux et vidéo, que ce soit individuellement ou en groupe. De plus, la fonctionnalité de partage d'écran est idéale pour les présentations, les démonstrations ou l'assistance technique, enrichissant ainsi les interactions et favorisant une collaboration fluide sur la plateforme.
* **Communauté et Collaboration**: Discord est utilisé par des communautés diverses, allant des gamers aux groupes d'étudiants, en passant par les équipes professionnelles, les clubs de lecture, et bien d'autres. Les outils de communication et de partage de fichiers de Discord facilitent la collaboration sur des projets, que ce soit pour des études, des travaux professionnels ou des loisirs.

**3.1.2 Livraison et Intégration:**

La méthode de livraison incrémentale consiste à intégrer et déployer progressivement les fonctionnalités développées, offrant ainsi aux utilisateurs finaux une utilisation anticipée et la possibilité de fournir un feedback précieux. En parallèle, des tests continus, à la fois automatisés et manuels, sont effectués en continu pour garantir la qualité et la fiabilité du produit. Tout dysfonctionnement est rapidement identifié et corrigé, permettant ainsi d'assurer une expérience utilisateur optimale et une progression fluide du projet.

**3.1.3 Adaptation et Evolution:**

Dans le processus Agile de développement, les retours des utilisateurs et des parties prenantes sont essentiels. Ces retours sont intégrés dans les cycles de développement suivants, assurant ainsi une évolution du produit en fonction des besoins réels et des préférences des utilisateurs. De plus, le backlog est régulièrement réévalué et mis à jour pour refléter les nouvelles priorités et exigences, garantissant que l'équipe se concentre toujours sur les fonctionnalités les plus importantes et répond aux besoins changeants du projet.

**3.1.4 Documentation et Support:**

Dans le cadre du développement, une documentation continue est maintenue tout au long du processus pour consigner toutes les décisions, modifications et fonctionnalités, garantissant ainsi une traçabilité complète. Cette documentation fournit une référence précieuse pour l'équipe de développement et assure une cohérence dans le projet. De plus, une fois le produit livré, un support est fourni pour aider les utilisateurs à résoudre les problèmes rencontrés. Des mises à jour régulières sont planifiées pour améliorer le produit et répondre aux besoins évolutifs des utilisateurs, assurant ainsi une expérience utilisateur optimale à long terme.

**3.2 Définition des acteurs:**

**Oussama Itou:** est le chef de projet chargé de la planification et de l'organisation d'un projet de développement. Ses principales responsabilités incluent l'élaboration de plans détaillés, la définition des objectifs et des échéances, ainsi que la répartition des tâches et des responsabilités au sein de l'équipe. Il coordonne également les activités entre les différentes équipes fonctionnelles, alloue les ressources nécessaires pour le projet et surveille régulièrement son avancement pour s'assurer qu'il reste sur la bonne voie.

**Amine el aarouchi et Youness Benbakka:** sont responsables de la communication pour ce projet. Leur rôle consiste à développer une stratégie de communication alignée sur les objectifs du projet, à assurer une communication transparente au sein de l'équipe et avec les parties prenantes externes, à documenter les informations importantes et à gérer les crises de communication.

**Amine el aarouchi et Youness Benbakka:** sont responsables de la qualité du code. Leur mission est de définir les normes de qualité de codage, d'organiser des revues de code régulières, de mettre en place des tests automatisés et d'analyser les métriques de qualité pour garantir un code robuste et fiable.

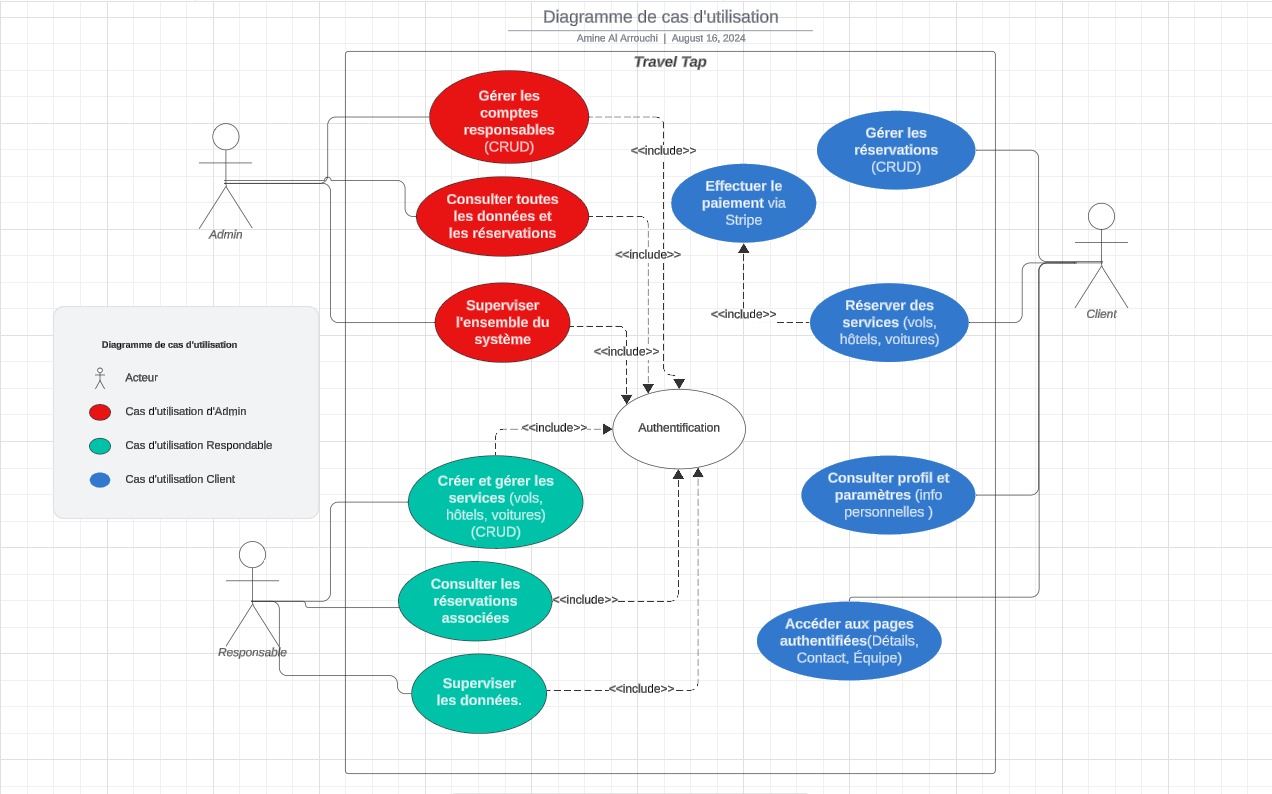
Amine el aarouchi et Youness Benbakka: responsable des livrables du projet. Ses tâches comprennent la compréhension des besoins des utilisateurs, la définition des spécifications fonctionnelles, la planification et la gestion des livrables, la communication des progrès aux parties prenantes, la gestion des risques et l'assurance qualité, ainsi que la formation et le support des utilisateurs finaux.

**3.3 Les cas d’utilisation:**

Le cas d'utilisation "admin" est crucial pour assurer la sécurité et l'accès contrôlé au système de gestion scolaire. L'administrateur, en tant qu'utilisateur principal avec des privilèges étendus, accède à la page de connexion où il saisit son nom d'utilisateur et son mot de passe. Ces informations sont ensuite vérifiées par le système, qui utilise des techniques de cryptage pour assurer la confidentialité des données. En cas de succès, l'administrateur est dirigé vers le tableau de bord, offrant un aperçu complet des fonctionnalités de gestion disponibles. Cela inclut la possibilité de gérer les informations des étudiants, des professeurs, des cours, des classes, de planifier les horaires et de générer des rapports détaillés. Si les informations d'identification sont incorrectes, le système affiche un message d'erreur précisant que les identifiants sont invalides. De plus, pour renforcer la sécurité, le cas d'utilisation peut inclure des fonctionnalités supplémentaires telles que la vérification en deux étapes ou la récupération de mot de passe pour assurer un accès sécurisé même en cas d'oubli du mot de passe.

Le cas d'utilisation "professor" est essentiel pour permettre au professeur d'accéder au système de gestion scolaire en fournissant ses identifiants valides, tels que son nom d'utilisateur et son mot de passe. Une fois authentifié, le professeur est redirigé vers le tableau de bord qui offre une vue d'ensemble complète de ses cours et des informations sur les étudiants inscrits. Sur ce tableau de bord, le professeur peut naviguer vers différentes fonctionnalités, telles que la liste des cours qu'il enseigne, où il peut consulter les détails de chaque cours, y compris le programme et les horaires. De plus, il peut accéder aux informations détaillées sur les étudiants inscrits dans ses cours, lui permettant de suivre leur progression académique et de communiquer avec eux si nécessaire. Le professeur peut saisis et gérer les notes des étudiants, en garantissant ainsi une évaluation précise et opportune de leur travail. Enfin, le cas d'utilisation "professor" lui permet de consulter son calendrier de cours, assurant une gestion efficace de son emploi du temps. Ensemble, ces fonctionnalités offrent au professeur un accès complet et pratique à toutes les informations et outils nécessaires pour assurer une expérience d'enseignement et d'apprentissage optimale dans le système de gestion scolaire.

Le cas d'utilisation pour “étudiants” permettant à l'étudiant de se connecter au système en saisissant ses identifiants, qui sont ensuite vérifiés pour assurer une authentification sécurisée. Une fois authentifié, l'étudiant est redirigé vers un tableau de bord personnalisé affichant un aperçu complet de ses informations académiques, y compris les cours actuels et les notes obtenues. À partir de là, l'étudiant peut accéder à une liste détaillée sue les cours aux quels il est inscrit, facilitant ainsi la gestion de ses activités académiques et permet à l'étudiant de consulter ses notes, offrant une vue détaillée de ses performances dans chaque cours. Pour une gestion efficace du temps il affiche le calendrier des cours de l'étudiant, incluant les horaires et les salles de classe. Enfin, la fonctionnalité profile permet à l'étudiant de consulter et de mettre à jour ses informations personnelles, telles que l'adresse, le numéro de téléphone et d'autres détails pertinents. Toutes ces fonctionnalités sont soutenues par une logique métier complexe qui traite les demandes en temps réel et interagit de manière sécurisée avec la base de données, assurant ainsi la sécurité et l'intégrité des transactions de données et des informations personnelles de l'étudiant.



*Figure 2.1: Diagramme de cas d’ utilisation*

Ce diagramme de cas d'utilisation pour l'application "Travel Tap" illustre les interactions entre trois rôles clés : Admin, Responsable, et Client. Chaque rôle est défini par des opérations spécifiques qu'il peut effectuer au sein du système.

Les Admins gèrent le système global en supervisant les comptes des Responsables, les données du système, et en accédant à toutes les réservations. Les Responsables se concentrent sur la création et la gestion des services tels que les vols, hôtels, et voitures, tout en supervisant les réservations et les données associées. Les Clients sont principalement impliqués dans la réservation de services, la gestion de leur profil, et les paiements via Stripe.

L'authentification est une exigence commune à tous les rôles, garantissant un accès sécurisé au système. Cette intégration met en évidence les opérations coordonnées nécessaires pour une expérience utilisateur fluide.

**3.4 Conception UML:**

**3.4.1 Definition UML:**

UML (Unified Modeling Language): est un langage de modélisation visuel standardisé utilisé pour spécifier, visualiser, construire et documenter les artefacts d'un système logiciel. Il est largement adopté dans l'industrie du développement logiciel pour faciliter la conception et la communication des systèmes complexes.

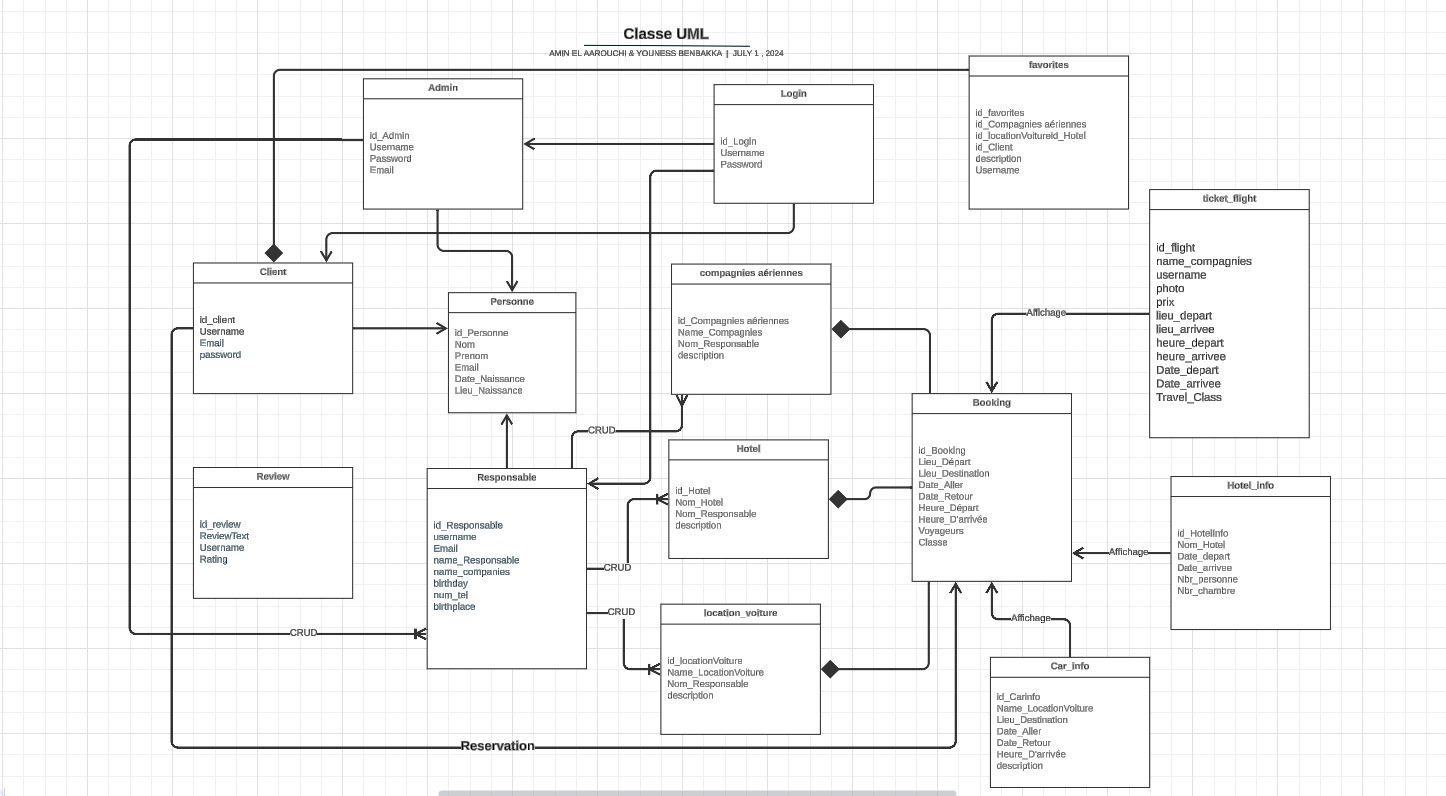
Principales caractéristiques de l’UML:

* Standardisation: UML est un langage standardisé par l'Object Management Group (OMG), garantissant une compréhension commune et une cohérence dans la modélisation à travers différents projets et organisations.
* Visualisation: UML permet de créer des diagrammes visuels qui aident à représenter les différentes aspects du système, ce qui facilite la compréhension et la communication entre les membres de l'équipe de développement et les parties prenantes.
* Spécification: UML est utilisé pour définir précisément les exigences fonctionnelles et non fonctionnelles du système. Il permet de capturer les détails sur les composants, les interactions et les comportements du système.
* Construction: En fournissant une représentation détaillée de l'architecture et des interactions du système, UML guide le processus de développement, assurant que les implémentations respectent les spécifications définies.
* Documentation: UML fournit une documentation complète et structurée du système qui peut être utilisée pour la maintenance future, la mise à jour et l'extension du système.

Types des diagrammes UML utiliser:

**Le diagramme de cas d’utilisation:** est un outil essentiel de modélisation dans le développement logiciel, offrant une vue d'ensemble des interactions entre les acteurs externes et le système lui-même. Les acteurs représentent les différents utilisateurs ou systèmes impliqués, tandis que les cas d'utilisation décrivent les fonctionnalités ou services que le système offre à ces utilisateurs. Les relations telles que l'association, l'inclusion et l'extension permettent de représenter les dépendances et les interactions entre les différents cas d'utilisation. En résumé, ce diagramme fournit une représentation graphique claire et concise des besoins et des fonctionnalités du système du point de vue des utilisateurs, facilitant ainsi la communication et la compréhension entre les membres de l'équipe de développement et les parties prenantes. *(Voir la figure 2.1)*

**Le diagramme de classes:** est une représentation essentielle de la structure statique d'un système logiciel, mettant en lumière les classes du système, leurs attributs et leurs méthodes, ainsi que les relations entre elles. Ce diagramme offre une vision claire des concepts et des entités fondamentaux du système, ainsi que des associations, agrégations, héritages et dépendances entre les différentes classes. Il fournit une base solide pour la conception et la compréhension du système, permettant aux développeurs de modéliser efficacement les interactions et les fonctionnalités du logiciel à développer.

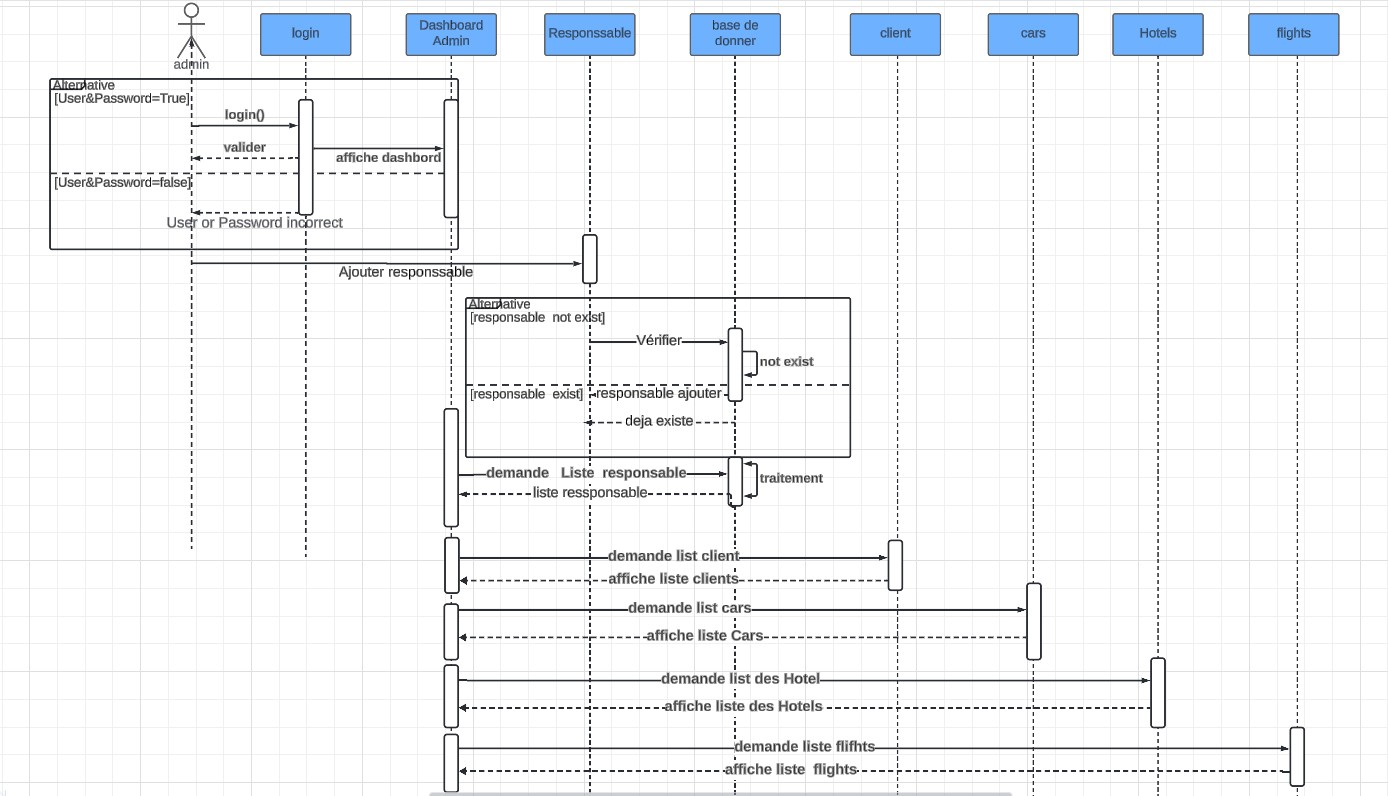


*Figure 2.2: Diagramme de Classes*

Le diagramme UML illustre un système complexe de réservation de voyages qui englobe la gestion des vols, hôtels, et locations de voitures. La classe "Responsable" occupe une position stratégique dans cette architecture en étant connectée aux principales entités du système, telles que les compagnies aériennes, les hôtels, et les voitures de location. Cette classe a pour mission d'effectuer des opérations CRUD (Create, Read, Update, Delete) sur ces entités, ce qui lui permet de gérer efficacement les services associés.

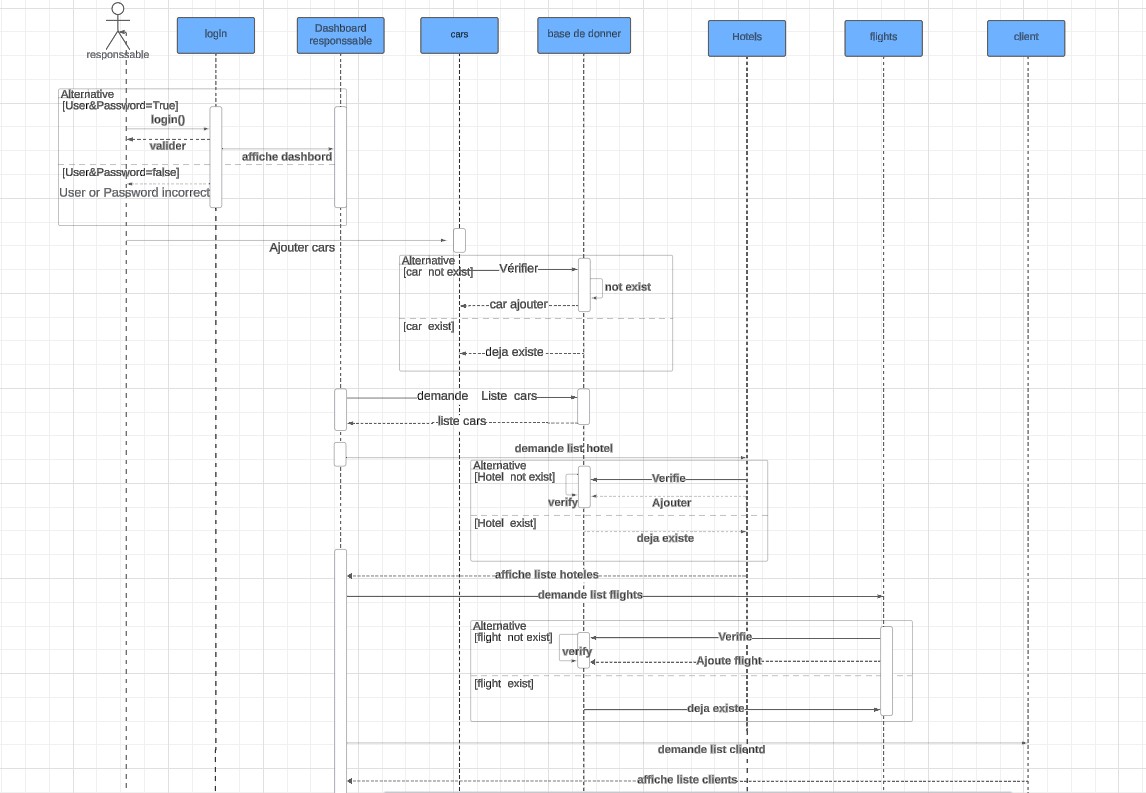
Cette centralisation des opérations sous la responsabilité de cette classe garantit une administration homogène et fluide des différents aspects du système. En conférant au Responsable la capacité de gérer les ressources essentielles, le système assure une coordination optimale des processus de réservation, tout en offrant une répartition claire et précise des responsabilités. Ainsi, la structure présentée favorise une gestion efficace des services et une réactivité accrue face aux demandes des clients, tout en maintenant un contrôle rigoureux des opérations réalisées par les Responsables.

**Le diagramme de sequence** : est une représentation graphique qui décrit comment les objets dans un système interagissent entre eux dans une séquence chronologique. Il montre l'ordre des messages échangés entre les différents objets pour accomplir une certaine fonctionnalité. Les éléments principaux d'un diagramme de séquence incluent les objets (représentés par des rectangles verticaux), les messages (représentés par des flèches entre les objets), et la séquence des événements (représentée horizontalement). Ces diagrammes sont utilisés pour modéliser le comportement dynamique d'un système et pour visualiser les scénarios d'utilisation. Ils sont utiles pour comprendre le flux d'exécution d'une fonctionnalité, identifier les interactions entre les objets, et vérifier la logique de communication entre les composants d'un système logiciel.

**Admin:** 

*Figure 2.3: Diagramme de Sequence (Admin)*

Le diagramme de séquence représente les interactions entre un administrateur et le système pour gérer les utilisateurs et les données. L'administrateur commence par se connecter, et le système valide les informations avant d'afficher le tableau de bord en cas de succès. Ensuite, l'administrateur peut ajouter un responsable après vérification de son existence dans la base de données. Si le responsable existe déjà, une alerte est déclenchée, sinon, il est ajouté avec succès. Le diagramme montre également comment l'administrateur peut demander des listes, telles que celles des clients, des voitures, des hôtels, et des vols, qui sont ensuite récupérées et affichées par le système. Ces actions garantissent une gestion efficace des données au sein de la plateforme.

* 

**Responsable:**

*Figure 2.4: Diagramme de Sequence (responsable)*

Le diagramme de séquence représente de manière séquentielle les interactions entre le professeur et le système pour accomplir diverses tâches liées à son rôle dans l'enseignement. Initialement, le professeur accède à l'interface de connexion ("/login") où il fournit ses informations d'identification pour accéder au système. Une fois connecté avec succès, il est dirigé vers le tableau de bord du professeur ("/dashboard"), qui affiche des informations essentielles sur les cours enseignés et les étudiants inscrits. Ensuite, le professeur peut accéder à la liste des cours qu'il enseigne ("/courses") pour voir les détails de chaque cours. Il peut également consulter les informations des étudiants inscrits dans ses cours ("/students") pour suivre leur progression académique. Lorsqu'il est temps de saisir ou de gérer les notes des étudiants, le professeur utilise la fonctionnalité "/professor/grades". Enfin, le professeur peut consulter son emploi du temps pour les cours qu'il enseigne en utilisant la fonctionnalité "/professor/schedule". À chaque étape, le diagramme de séquence illustre les messages échangés entre le professeur et le système, montrant ainsi le flux d'exécution des actions effectuées par le professeur dans le cadre de son rôle d'enseignant.

**client:**



*Figure 2.5: Diagramme de Sequence (client)*

Le diagramme de séquence commence avec le client qui entre son nom d'utilisateur et son mot de passe, lesquels sont validés par le système. En fonction de la validité des identifiants, l'utilisateur est soit connecté avec succès, soit invité à les saisir à nouveau. Une fois authentifié, le client peut initier des demandes de réservation pour des vols, des locations de voitures, et des hôtels. Chaque demande de réservation incite le système à vérifier la disponibilité et, si elle est confirmée, la réservation correspondante est effectuée. De plus, l'utilisateur peut ajouter les services réservés à ses favoris et récupérer les détails de la réservation. Ces actions impliquent la vérification de la disponibilité et le stockage des résultats dans le profil de l'utilisateur.

**Intégration de Stripe pour le paiement :**

Pour intégrer la fonctionnalité de paiement Stripe, une séquence supplémentaire peut être ajoutée après chaque confirmation de réservation (vol, location de voiture ou hôtel). Une fois la disponibilité du service demandé confirmée par le système, une demande de paiement est envoyée à Stripe. L'API Stripe traite le paiement et, une fois le paiement réussi, Stripe notifie la plateforme, qui confirme alors la réservation. Ce processus de paiement garantit que seules les réservations associées à des paiements validés sont confirmées, offrant ainsi une expérience transactionnelle sécurisée et fluide.

**Chapitre 4**

**Réalisation et tests:**

**“Nous avons conçu une application de travel booking web app, Le front-end utilise EJS pour le rendu dynamique, Bootstrap pour le style, et du CSS personnalisé. Le back-end, construit avec Node.js et Express, communique avec MongoDB pour la gestion des données. Git a été utilisé pour la gestion de version, avec Jest pour les tests unitaires et Postman pour les tests d'API. L'application est intuitive, modulaire et sécurisée, offrant une gestion complète des ressources travel booking.”**

**4.1 Environnement de Développement:**

Technologies Utilisées:

* Front-end: CSS, EJS, BOOTSRAP pour l'interface utilisateur.
* Back-end: Node.js avec Express pour gérer les requêtes et la logique serveur.
* Base de données: MongoDB pour sa flexibilité et son adaptation aux besoins de l'application.
* Gestion de versions: Git et GitHub pour le contrôle des versions et la collaboration.
* Environement de travail: Discord, Visual Studio Code, Cloudinary (pour enregister images)
* Conception: StarUML, Lucidchart

**4.2 Présentation de l’Application:**

**Description Générale:**

**Notre site permet aux administrateurs de gérer les informations relatives aux réservations de voyages, aux hôtels, aux voitures, et aux vols. Les fonctionnalités clés incluent:**

* **Gestion des utilisateur**: Ajouter, modifier, et supprimer responsables, et client(only read).
* **Gestion des réservations** : Création et gestion des réservations de vols, hôtels, et voitures.
* **Gestion des paiements** : Traitement des paiements sécurisés via Stripe pour les réservations.
* **Gestion des ressources** : Planification et affichage des disponibilités des vols, hôtels, et voitures.

**Interfaces Front-end:**



*Figure 2.6: Login administrateur*

Description:

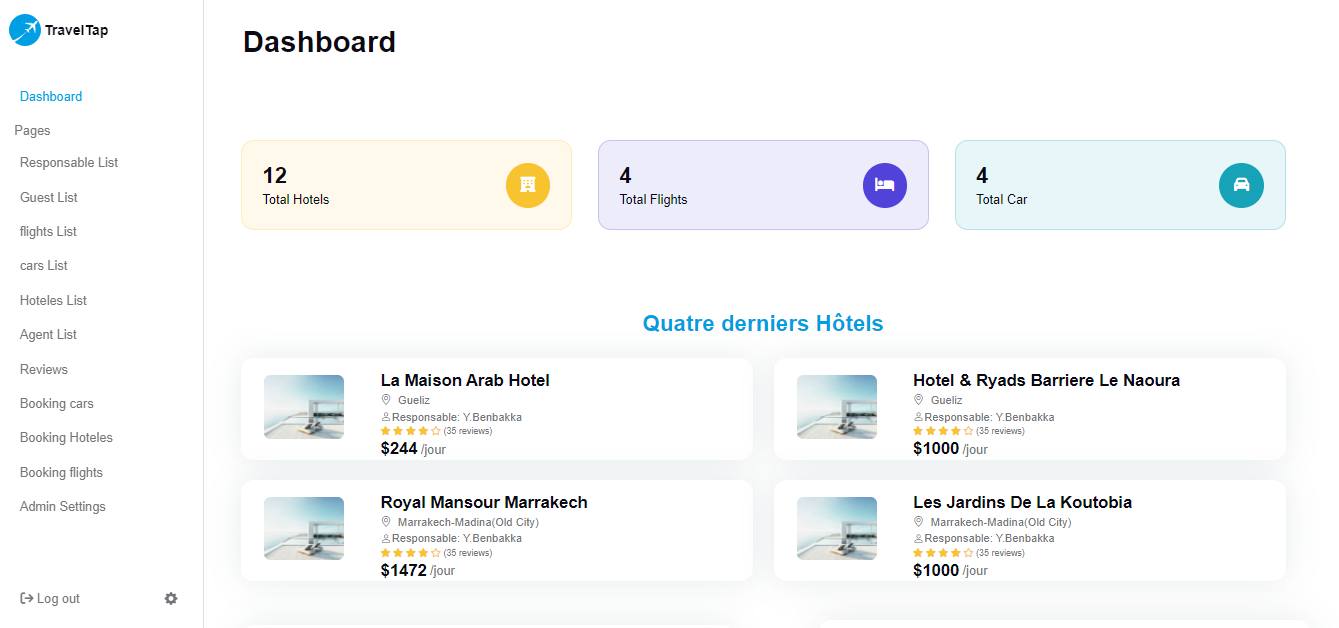
Cette page permet aux administrateur de se connecter à l'application avec leurs identifiants.

Éléments:

* Champs de saisie pour le nom et le prenom /email et le mot de passe.
* Bouton de log In.

Validation en temps réel: Les champs de saisie valident les données en temps réel, fournissant des.

Rederige autre fois pour saisir authentification



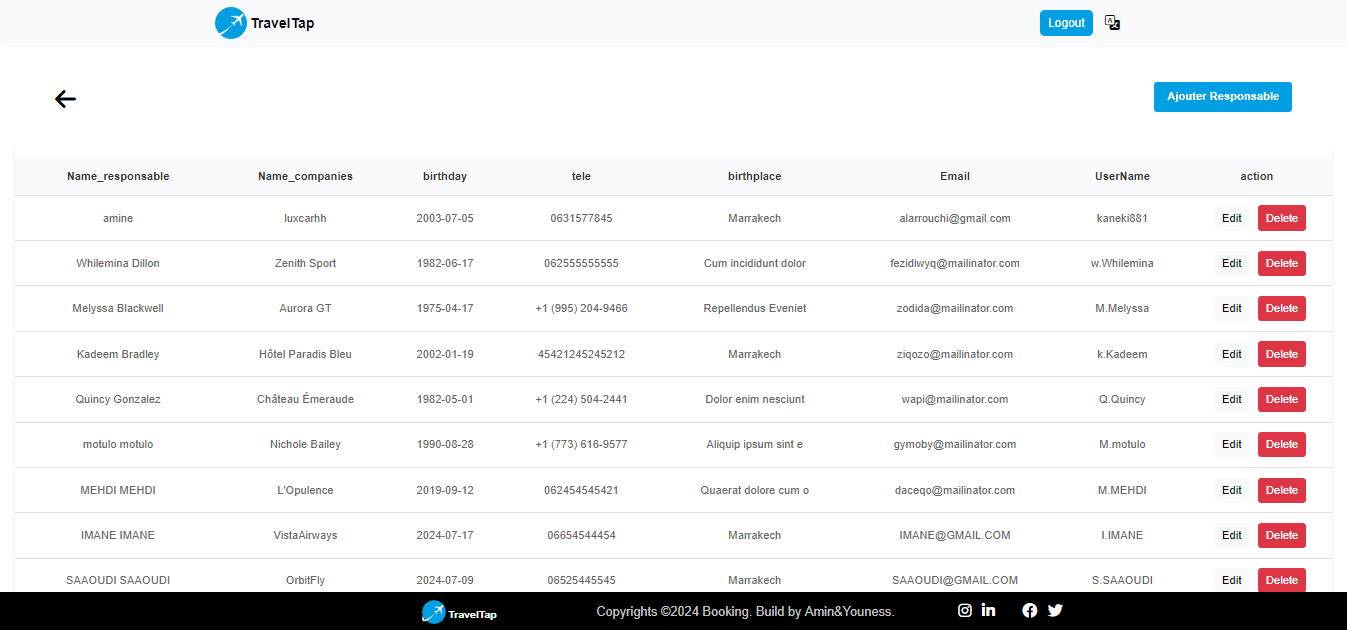
*Figure 2.7: Tableau de Bord Administrateur*

Description:

Cette page d'accueil affiche une vue d'ensemble des informations importantes sue les Hoteles,vols,Voiture. statistique

Éléments:

* Graphiques de performance.
* Menu de navigation latéral pour accéder à différentes parties de l'application.



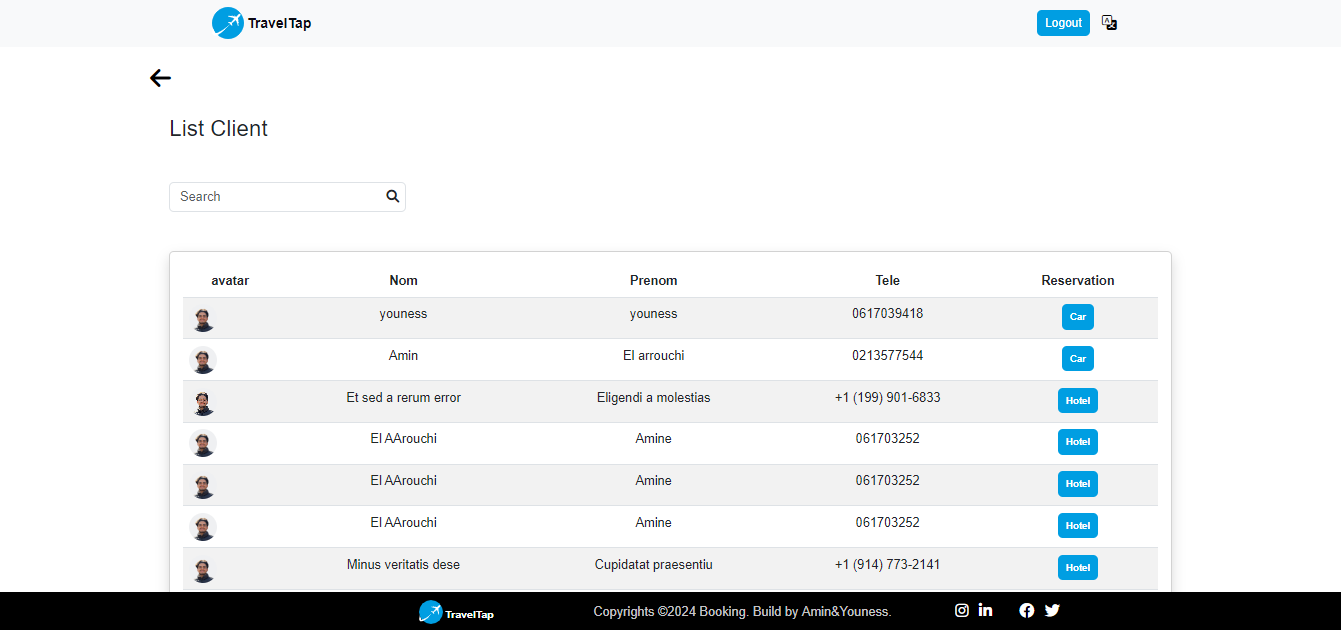
*Figure 2.8: Gestion des Responsable*

Description:

Cette page permet aux administrateurs de gérer les informations des responsible inscrits, y compris ajouter, modifier. effacer

Éléments:

* Boutons pour modifier un responsable.
* Boutons pour modifier un responsable
* Fleche de sortie



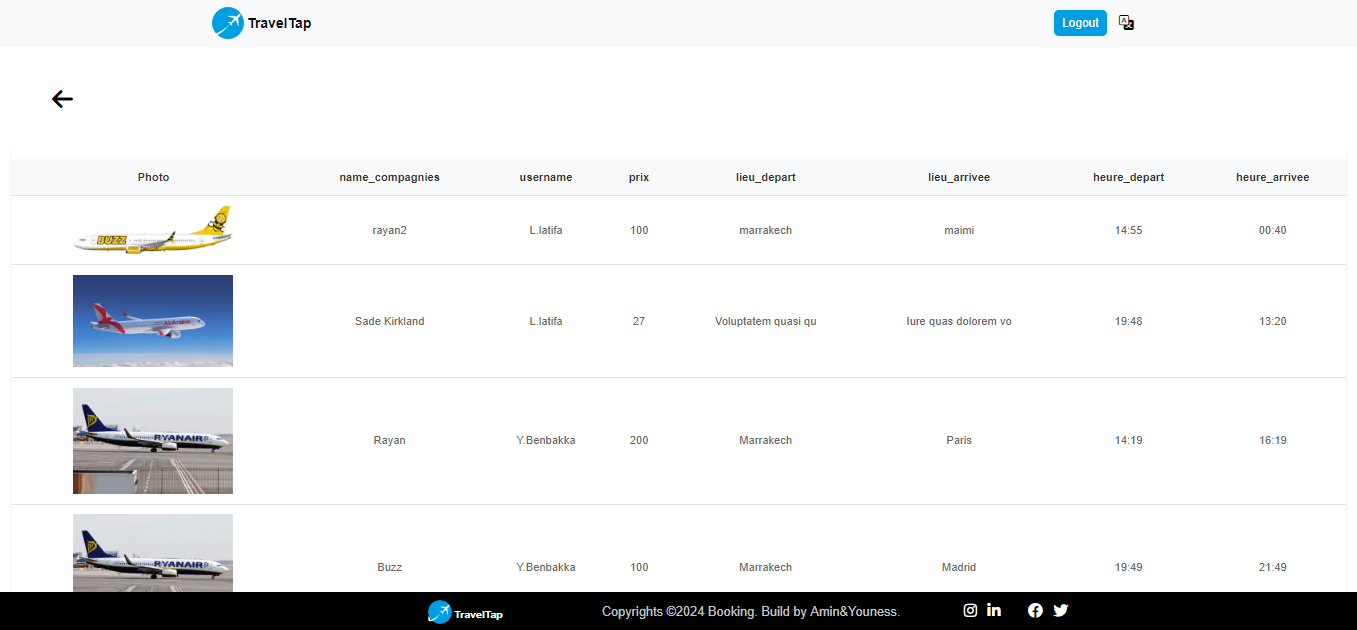
*Figure 2.9: liste clients*

Description:

Cette page est similaire à celle des clients affiche les information

Éléments:

* Fleche de sortie
* Table avec les information



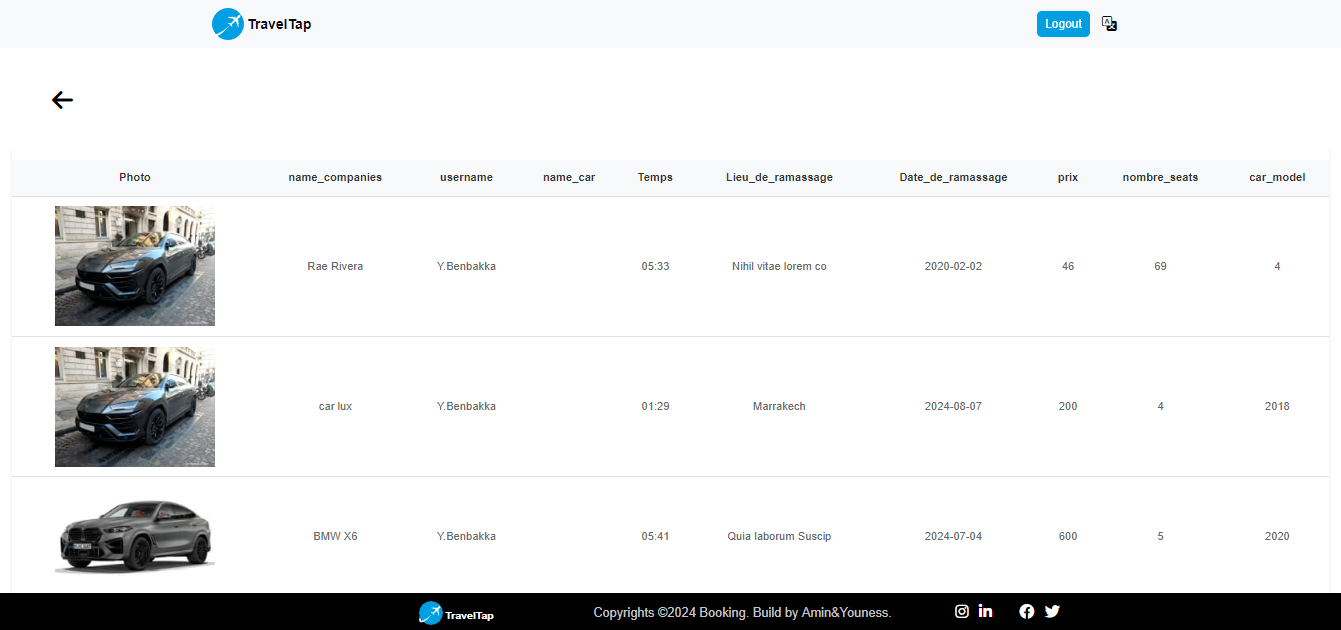
*Figure 3.1: list vols*

Description:

Cette page permet la affichage liste vols. Avec plus information

Éléments:

* Liste des vols.
* Fleche de sortie
* Détails des vols (dates,price….).



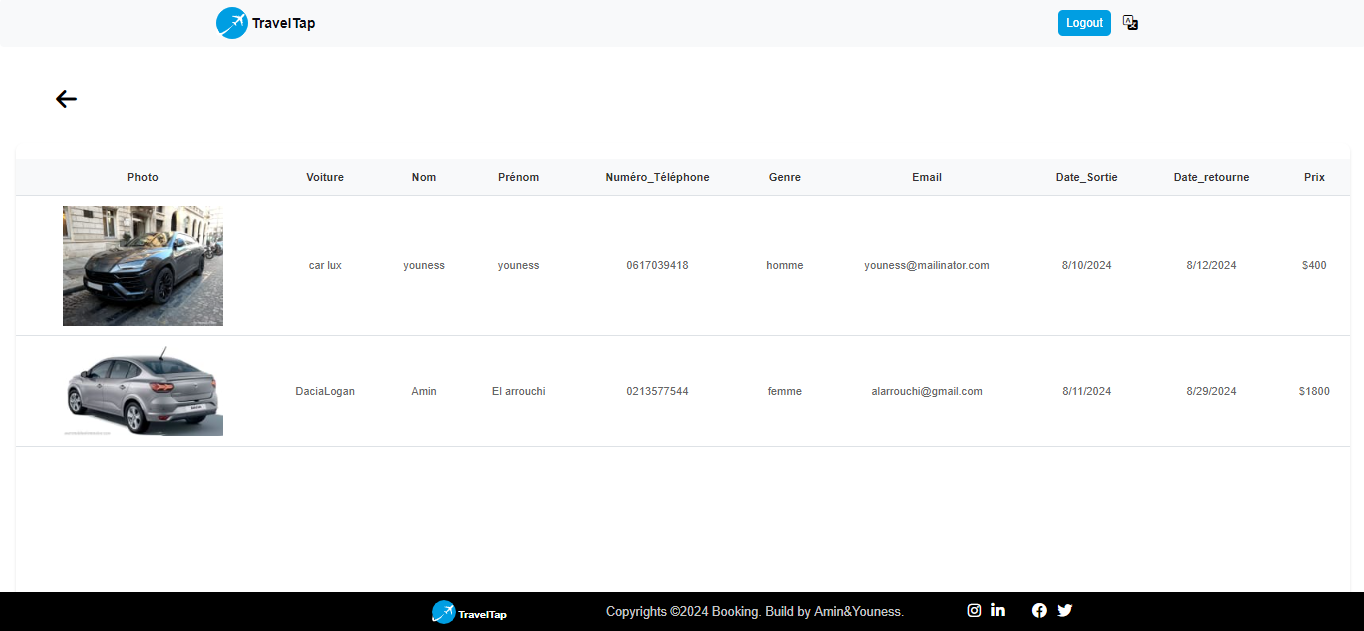
*Figure 3.2: affichage liste Voitures*

Description:

Cette page permet de planifier et de visualiser les liste voitures

Éléments:

* Les details chaque voiture cree par defferente responsabble.



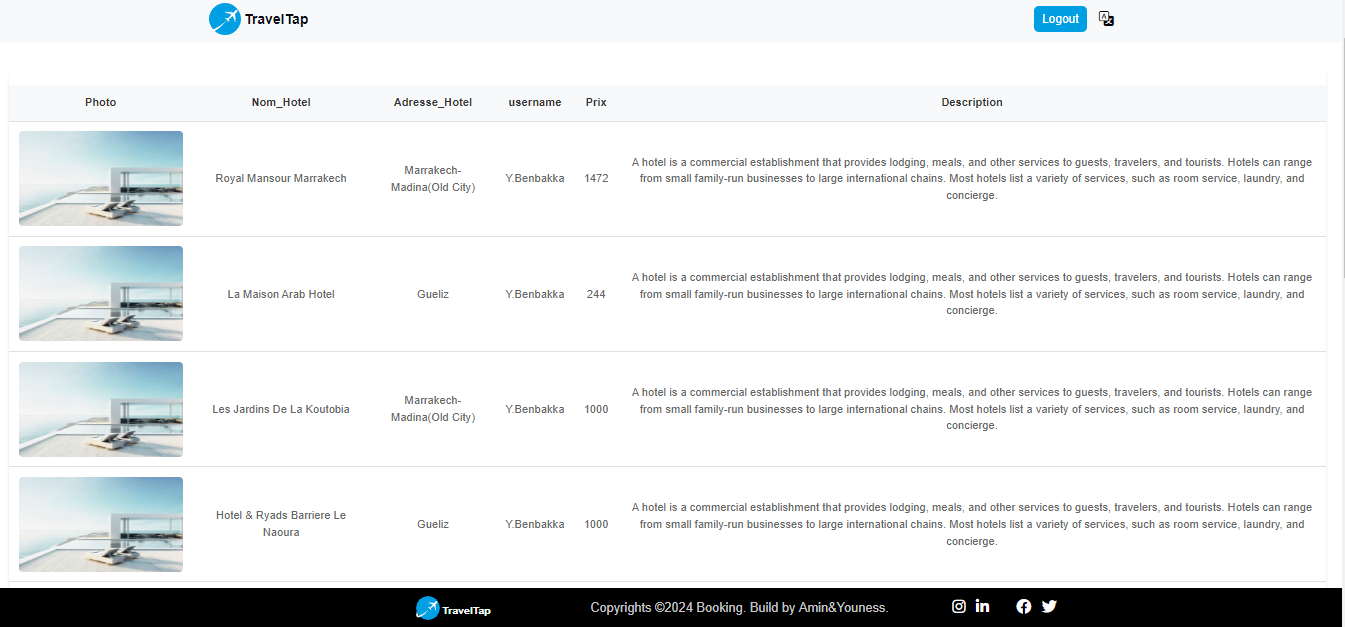
*Figure 3.3: Voiture Reservé par chaque client*

Description:

Cette page permet aux consulter les Voiture Reservé par chaque client

Éléments:

* Tableau des Voiture Reservé avec des information clients.
* Fleche de sortie



*Figure 3.4: Affichage Liste hotels*

Description:

Cette page permet aux consulter les Hotels par chaque responsable

Éléments:

* Tableau des les Hotels avec des information(details).
* Fleche de sortie



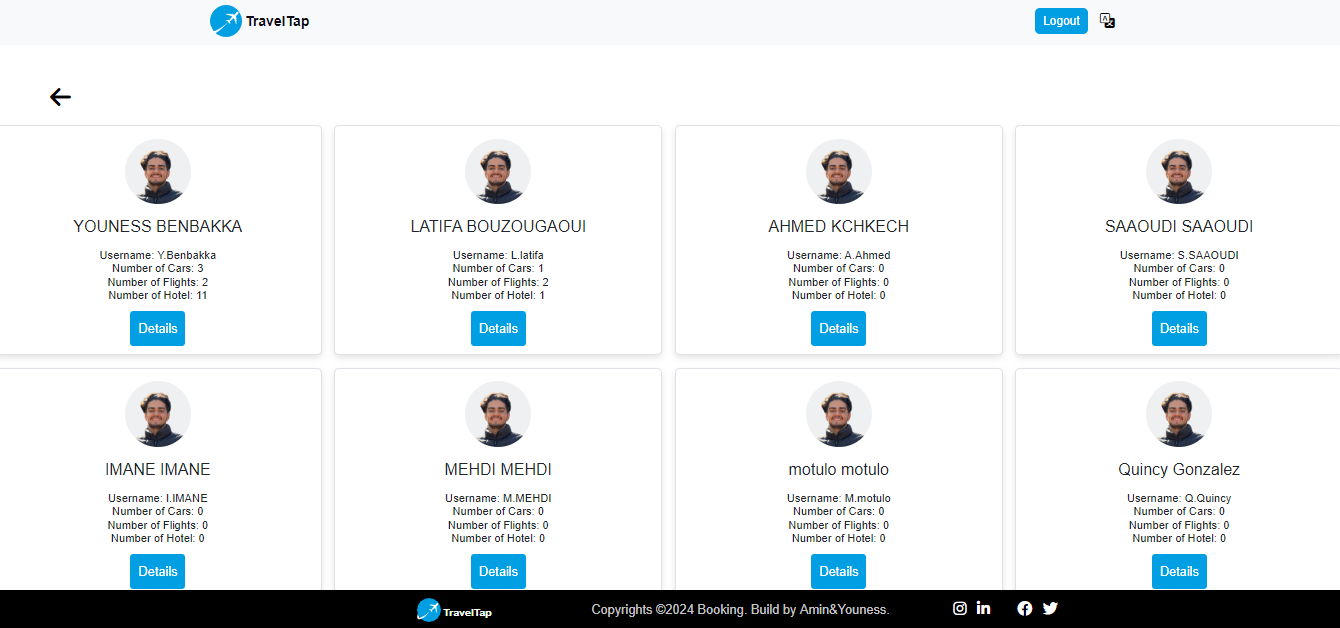
*Figure 3.5: liste hotels reserve par les client*

Description:

Cette page permet aux consulter les Hotels Reservé par chaque client

Éléments:

* Tableau des Hotels Reservé avec des information clients.
* Fleche de sortie

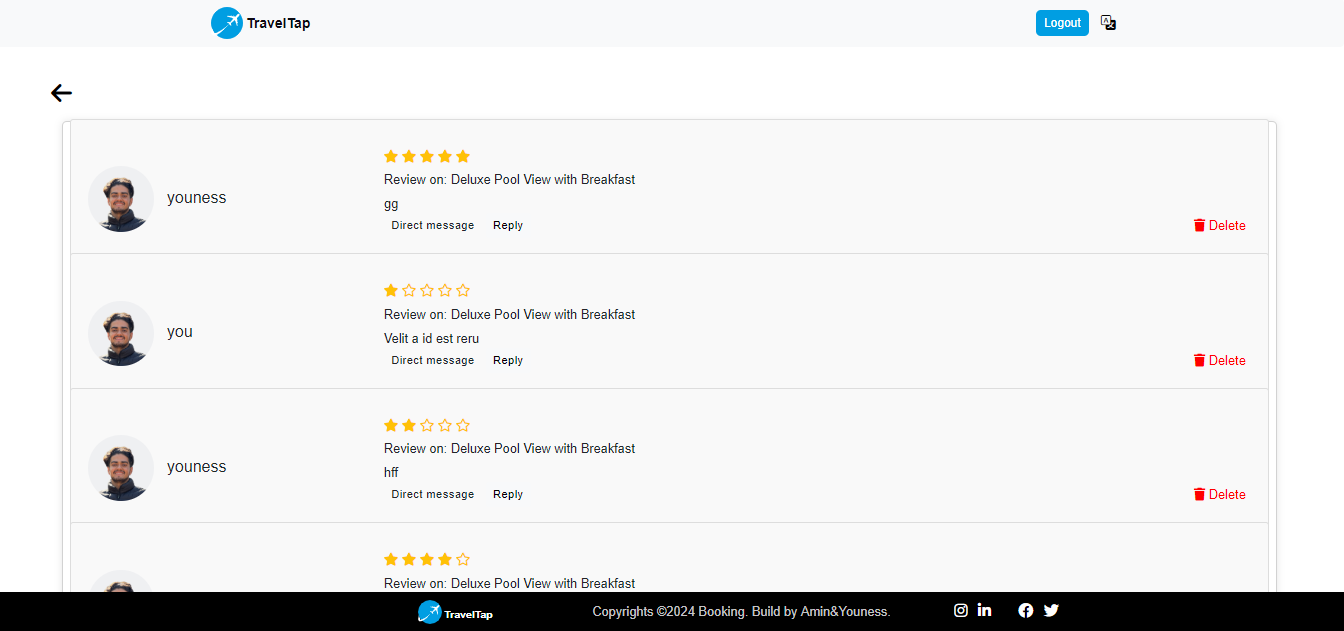


*Figure 3.6: cards Agents list*

Permet aux administrateurs affiche cards tous les responsible créé acec les information combine les vols et hotels et voiture créer

**Éléments Clés:**

* **Cards des responsable** : Affiche tous les responsible, avec des information des resource (vols,voiture,hotels)
* **Bouton pour details**: affiche plus information des responsable

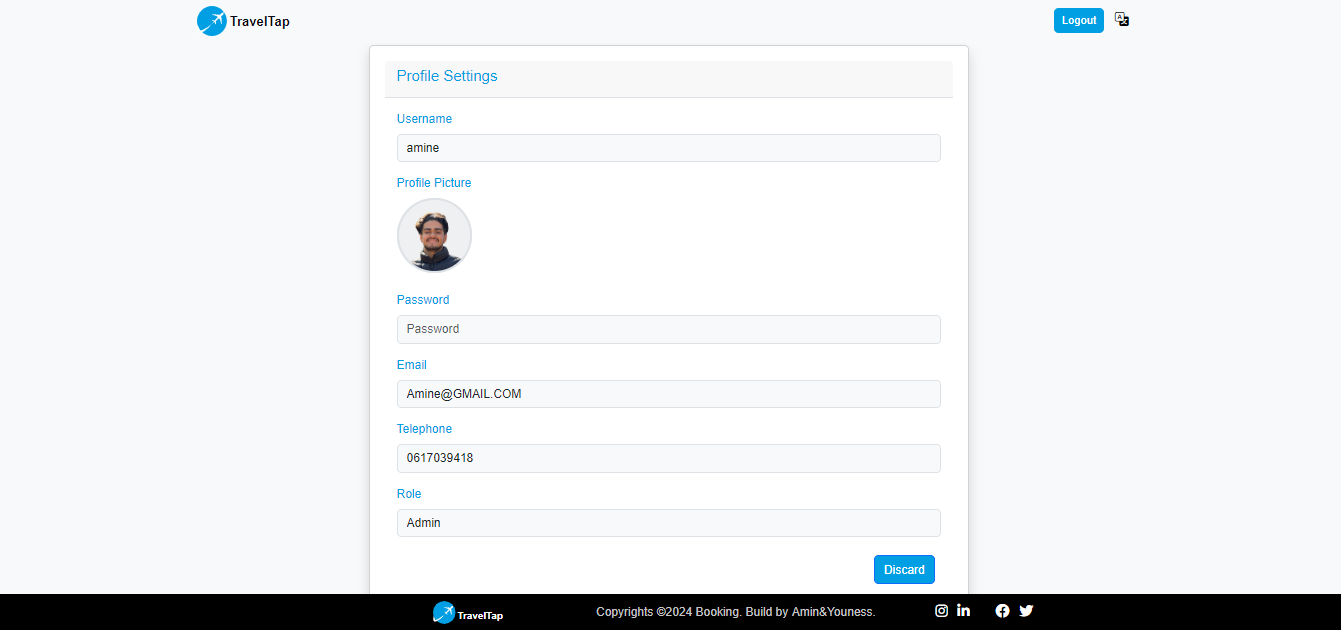


*Figure 3.7: afffiche commetaire des client*

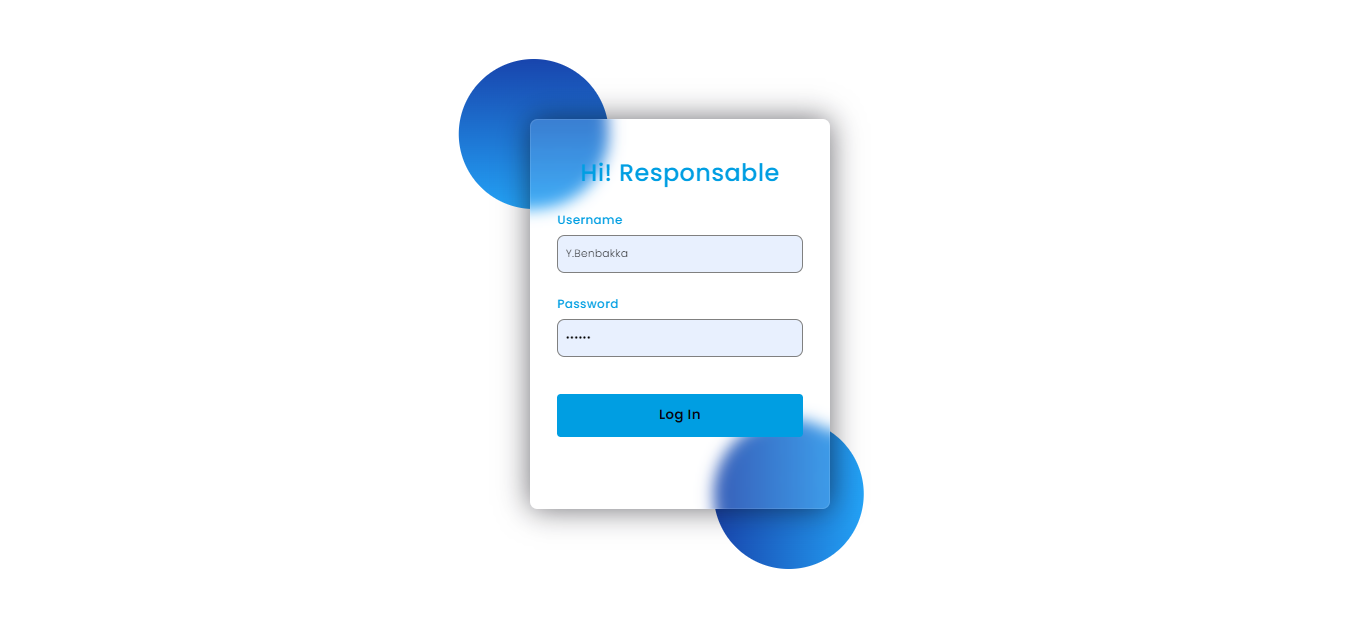
Gère les commentaire des clients, y compris la suppression comment .

**Éléments Clés:**

* **Liste des commentaire**:affiche tous les commentaire des client avec evaluation des performances.
* **Options de suppression**: suppression des commentaire.



*Figure 3.8: parameter Administrateur*



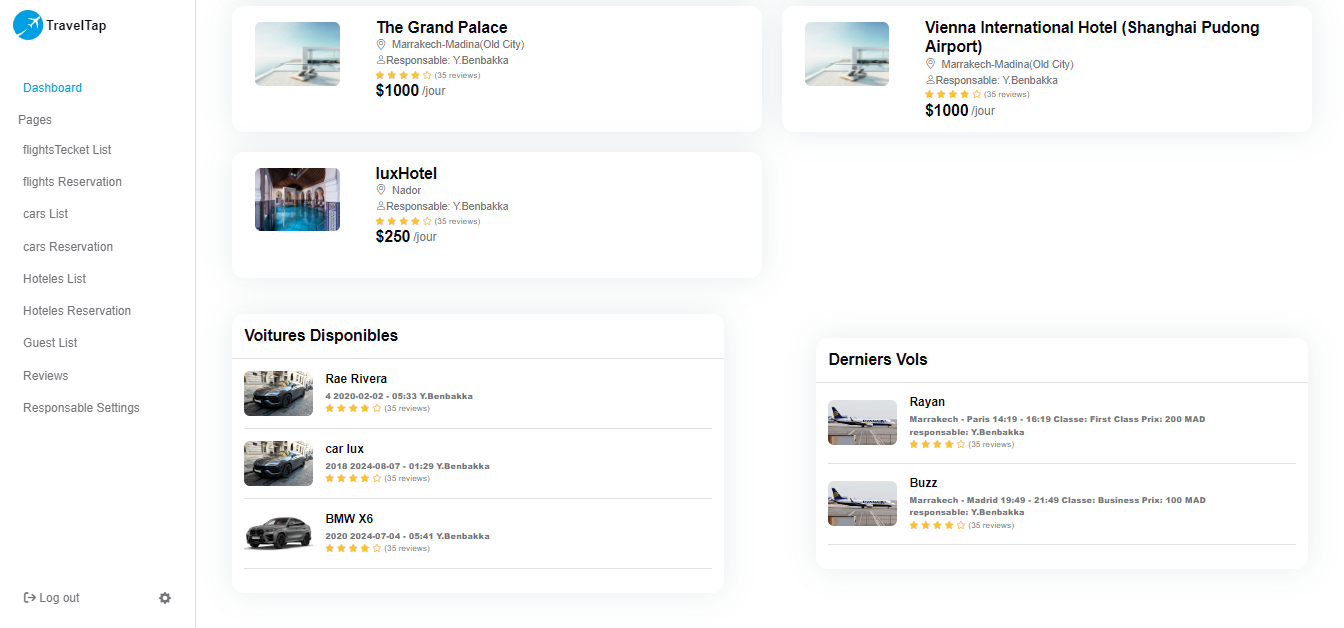
*Figure 3.9: Login Responsable*

Permet aux Responsible de se connecter.

**Éléments Clés:**

* **Champs de saisie pour le nom Responsable/ et le mot de passe** : Ces champs sont positionnés de manière centrale pour un accès facile. Des labels clairs et des placeholders guident l'utilisateur sur les informations à entrer.
* **Bouton de connexion**: Un bouton bien visible, souvent stylisé avec une couleur contrastante pour attirer l'attention. Le bouton inclut des mécanismes pour désactiver l'état de soumission multiple
* **Lien pour réinitialiser le mot de passe**: Placé sous les champs de saisie, permettant aux utilisateurs de récupérer leur accès en cas d'oubli.

Validation en temps réel: Les champs de saisie valident les données en temps réel, fournissant des redirége autre fois entrer authentification

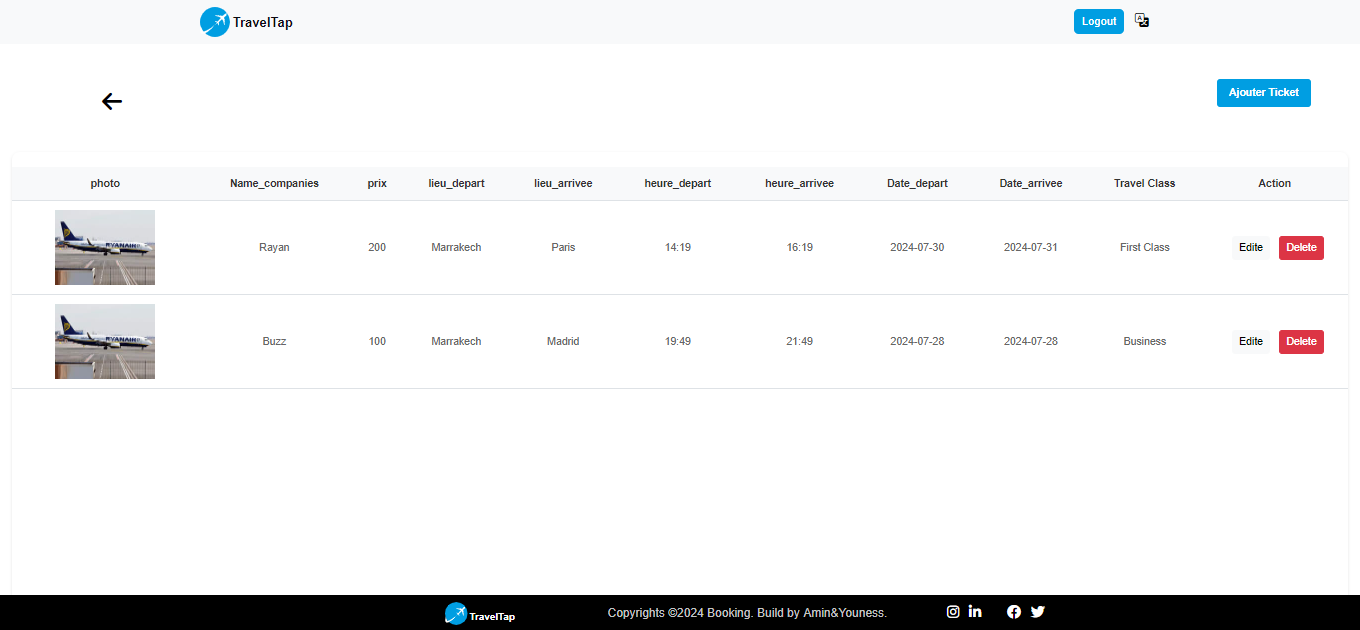


*Figure 4.1: Tableau de Bord responsable*

Offre une vue d'ensemble des informations critiques et des accès rapides aux différentes fonctionnalités de l'application.

**Éléments Clés:**

* **Liens rapides**: Des raccourcis vers les sections souvent utilisées, comme la gestion des cours ou absences, notes et Emploi du temps, facilitent l'accès aux principales fonctionnalités.
* **Menu de navigation latéral**: Offre une navigation persistante vers toutes les sections principales de l'application, avec des icônes et des labels pour une reconnaissance rapide.
* **Cards hotels,vols,Voiture : affichage des cards hotels and vols and voiture avec les statistique**

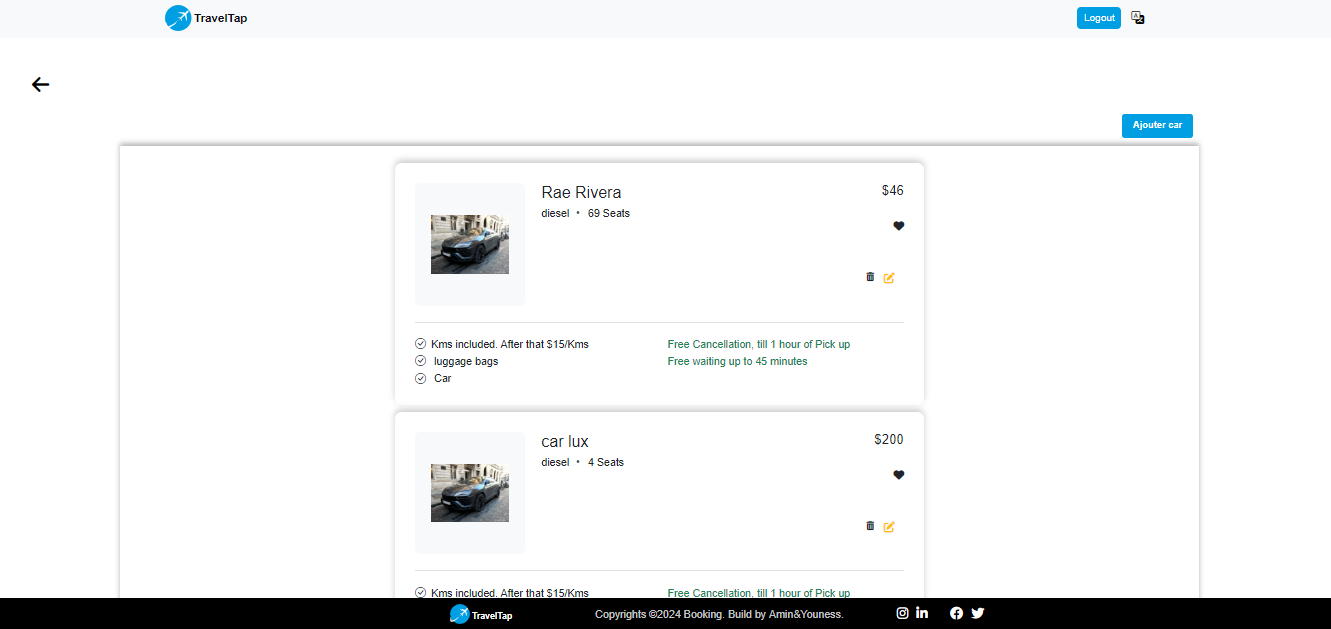


*Figure 4.2: Gestion des flights*

Gère flights des par chaque responsable, y compris l'ajout, la modification, et la suppression.

**Éléments Clés:**

**Liste de flights**: presente tous les flights qu’il a créé chaque responsible .

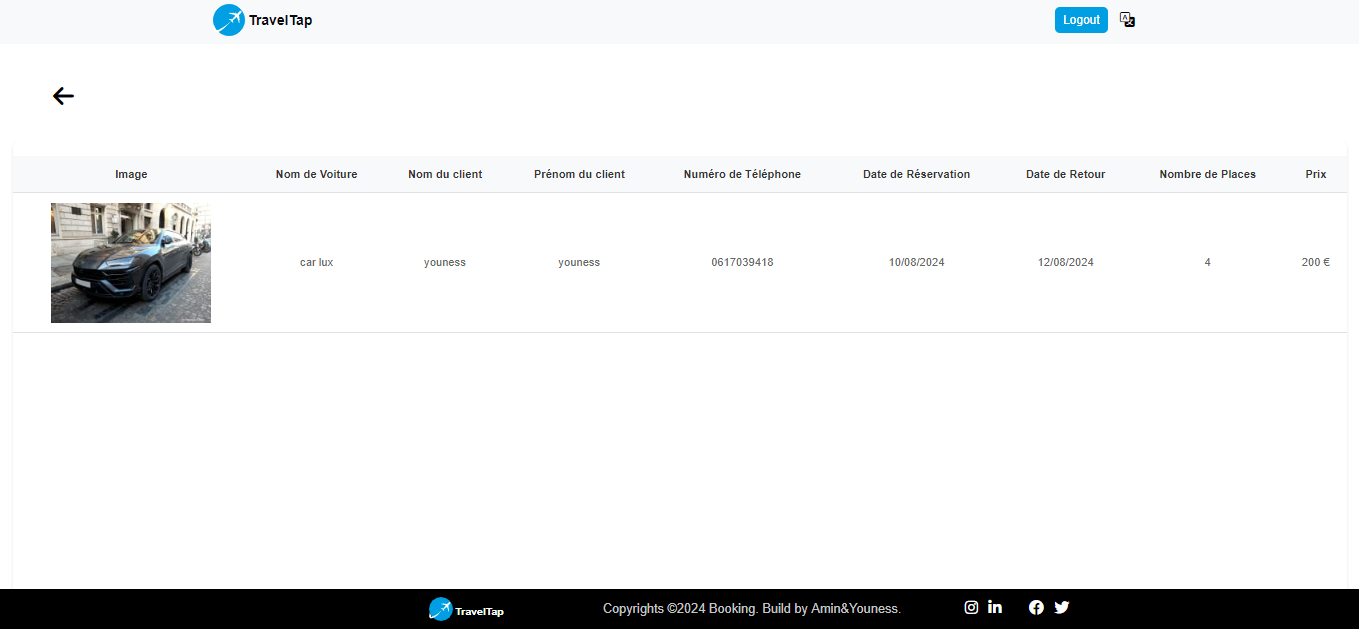


*Figure 4.3: Gestion Cards Voitures*

Gère voiture des par chaque responsable, y compris l'ajout, la modification, et la suppression.

**Éléments Clés:**

* **Cards de voiture**: presente tous les voitures qu’il a créé chaque responsible .

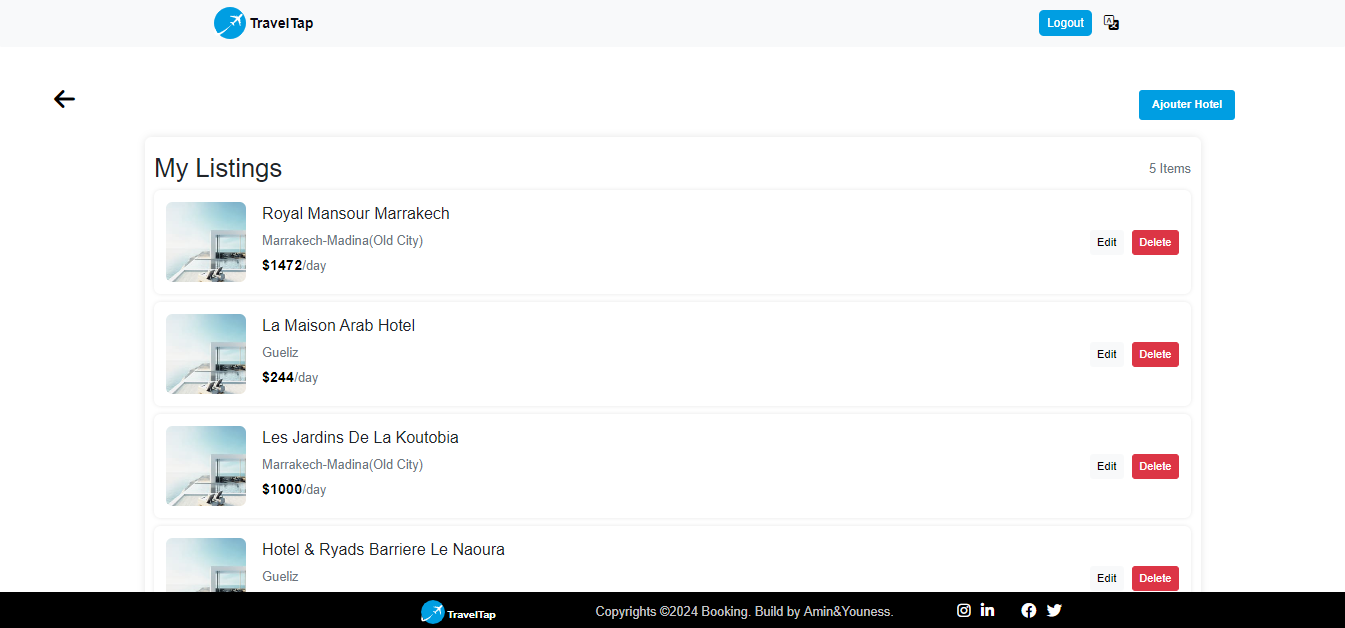


*Figure 4.4: Gestion de reservation de voiture pour client*

Permet aux responsable de consulter les Voiture reservation des client. selement ce qui appatient a son auteur

**Éléments Clés:**

* **Tableau Reservation des voiture** : Présente les voiture reserve par les client avec des information.

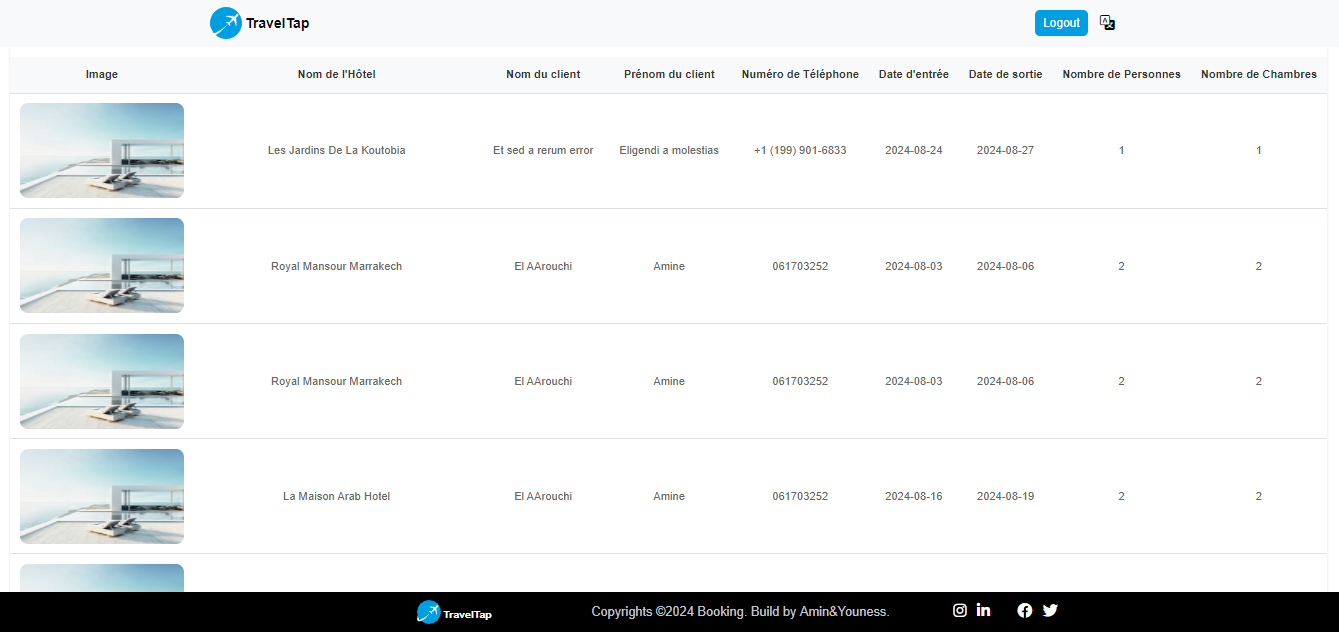


*Figure 4.5: list Hoteles*

Gère Hotels des par chaque responsable, y compris l'ajout, la modification, et la suppression

**Éléments Clés:**

* **Tableau des Hotels** : Présente les Hotels avec des information.(prix address….)

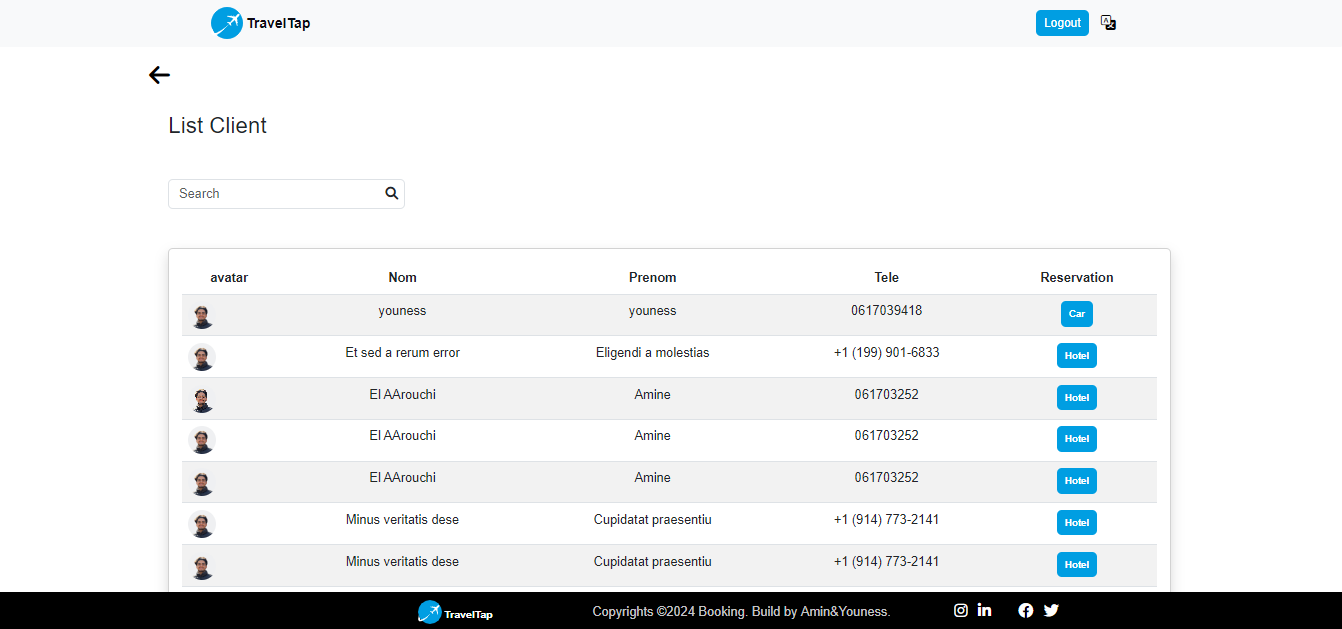


*Figure 4.6: liste gestion reservation des hotels pour les client*

Permet aux responsable de consulter les Hotels reservation des client. selement ce qui appatient a son auteur

**Éléments Clés:**

* **Tableau Reservation des Hotels**: Présente les voiture reserve par les client avec des information.

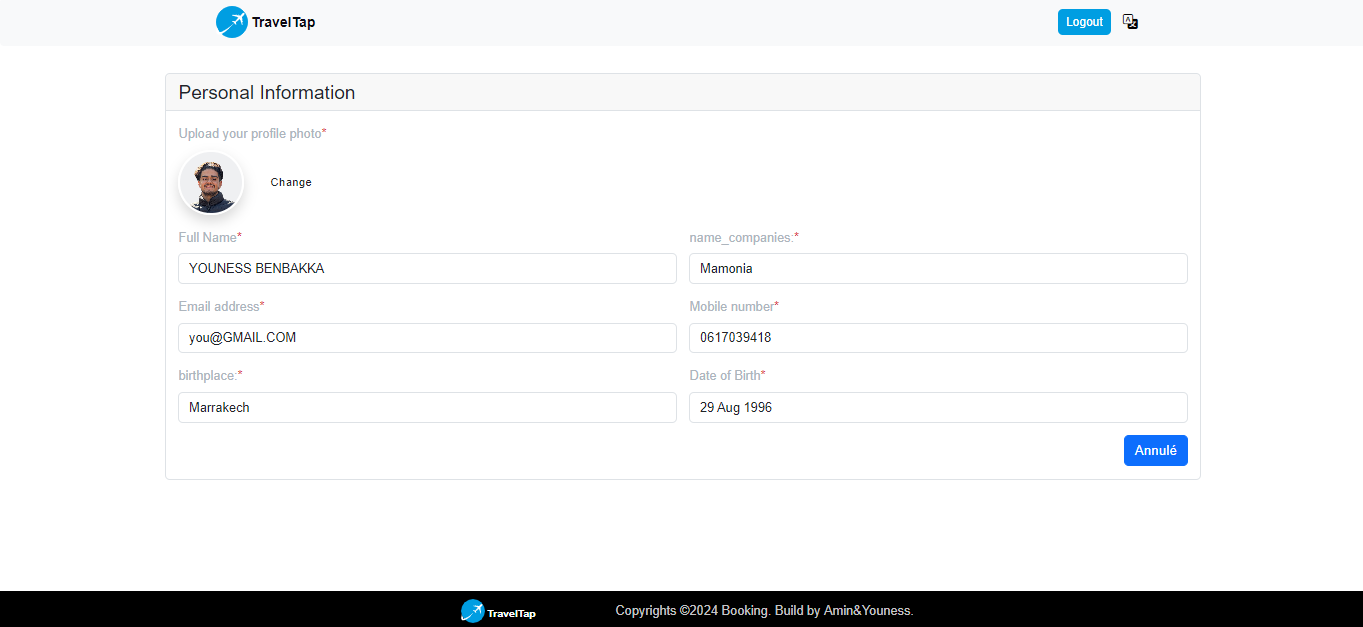


*Figure 4.7: list clients reservé des resource*

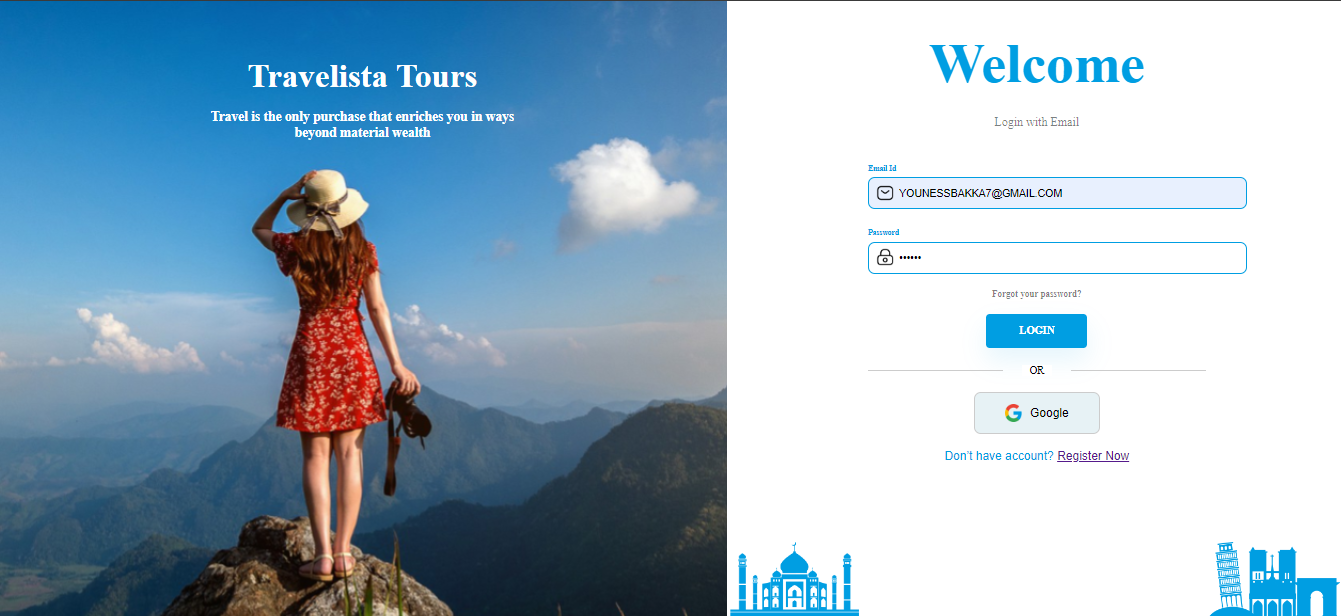
Permet aux les client reserves des resource a chaque responsible.

**Éléments Clés:**

* **Tableau Reservation des clinets**: Présente les clients reserve des resource a chaque responsible.



*Figure 4.8: Parmetre responsable*

****

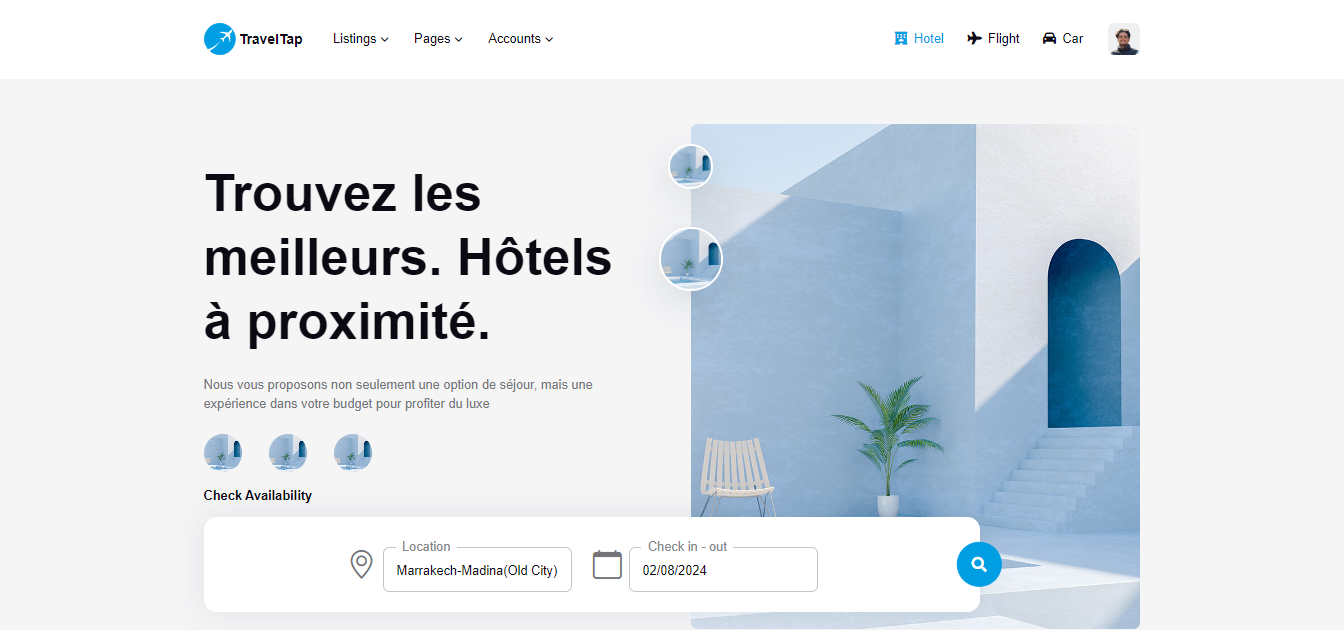
*Figure 4.9: login client*

Permet aux client de se connecter.

**Éléments Clés:**

* **Champs de saisie pour le Email client/ et le mot de passe** : Ces champs sont positionnés de manière centrale pour un accès facile. Des labels clairs et des placeholders guident l'utilisateur sur les informations à entrer.
* **Bouton de connexion**: Un bouton bien visible, souvent stylisé avec une couleur contrastante pour attirer l'attention. Le bouton inclut des mécanismes pour désactiver l'état de soumission multiple
* **Lien pour réinitialiser le mot de passe**: Placé sous les champs de saisie, permettant aux utilisateurs de récupérer leur accès en cas d'oubli.

Validation en temps réel: Les champs de saisie valident les données en temps réel, fournissant des redirége autre fois entrer authentification

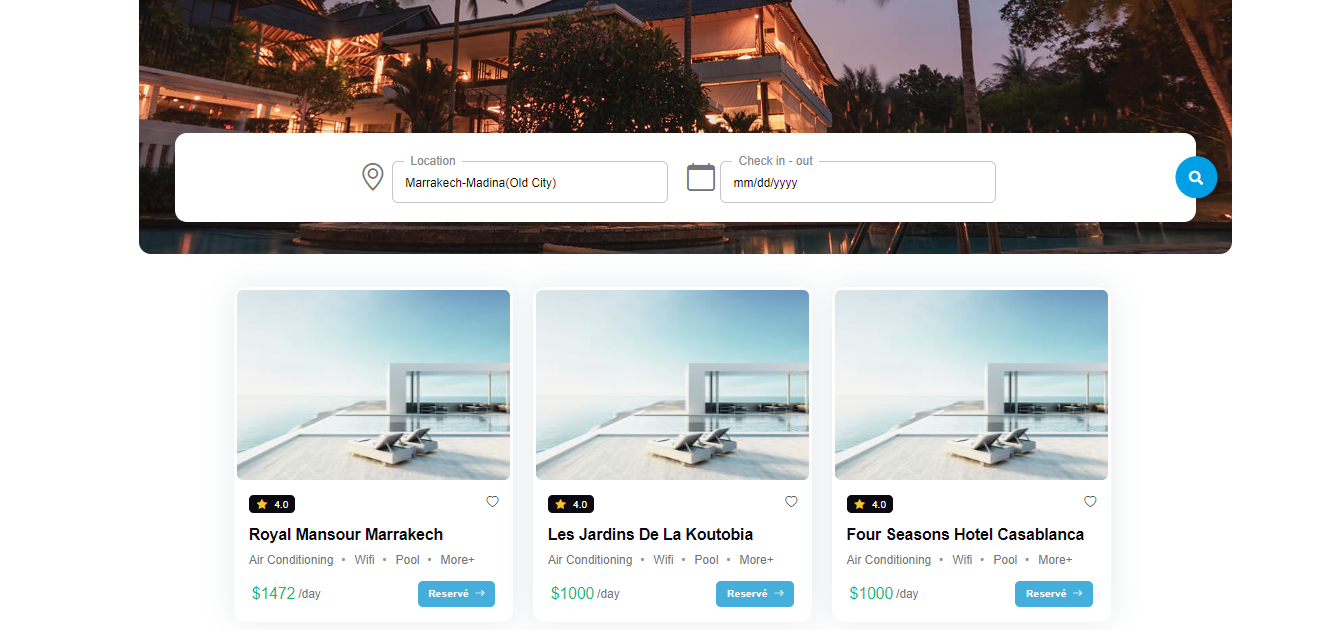
**

*Figure 5.1: index page hotels*

Pour la recherche d'hôtels, l'utilisateur doit remplir les champs de localisation et de date dans un formulaire. Si des hôtels correspondent aux critères de recherche, l'utilisateur sera redirigé vers une page qui affiche une liste de tous les hôtels disponibles correspondant à la localisation et aux dates spécifiées.

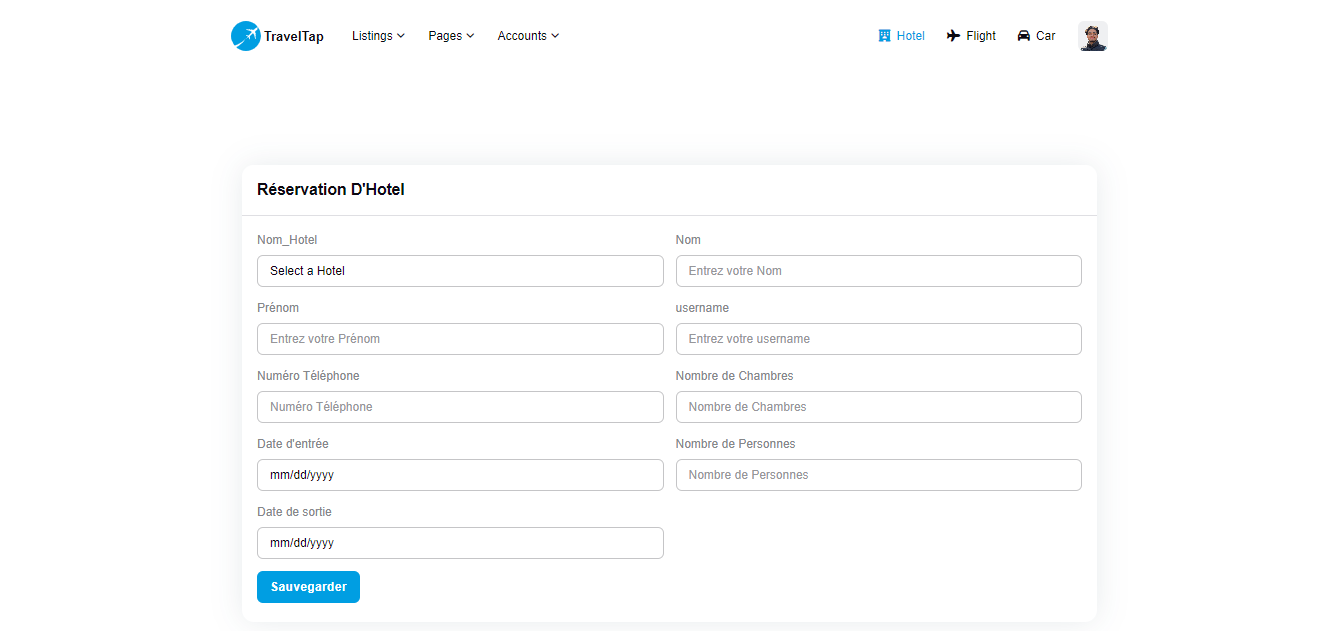
### Éléments Clés :

* **Formulaire de Recherche** :
  + Un formulaire simple qui inclut deux champs : un pour la localisation (ville, région) et un pour la date d'arrivée.
  + Le formulaire doit inclure un bouton de soumission pour déclencher la recherche.
* **Vérification des Données** :
  + Les champs de localisation et de date sont obligatoires.
  + Une validation de base doit être effectuée pour s'assurer que les valeurs sont bien remplies.



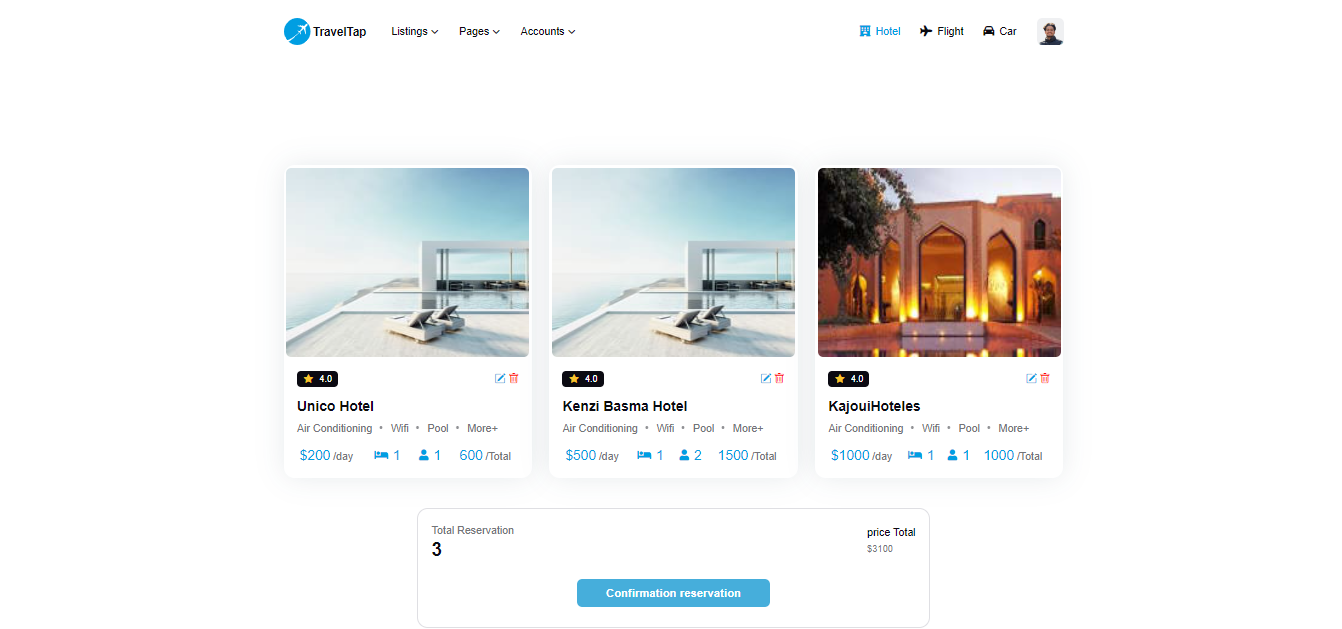
*Figure 5.2: resultat de recherche*

* **Redirection et Résultats** :
  + Une fois le formulaire soumis, l'application vérifie s'il existe des hôtels disponibles correspondant aux critères.
  + Si des résultats sont trouvés, l'utilisateur est redirigé vers une page qui affiche la liste des hôtels disponibles.
  + Sinon, un message d'erreur ou d'absence de résultats peut être affiché, avec la possibilité de modifier la recherche.



*Figure 5.3: client reserve hotel*

* Le formulaire de réservation d'hôtel permet aux utilisateurs de choisir un hôtel via un menu déroulant, de saisir leurs informations personnelles (prénom, nom, nom d'utilisateur) et de fournir leur numéro de téléphone. Ils peuvent également indiquer le nombre de chambres et de personnes, ainsi que sélectionner les dates d'arrivée et de départ. Pour finaliser la réservation, il suffit de cliquer sur le bouton « Enregistrer ».



*Figure 5.4: cards reservation client*

### Redirection après enregistrement

* Après que l'utilisateur ait rempli les champs et cliqué sur le bouton "Enregistrer", il est redirigé automatiquement vers la page "Crad Reservations", où les réservations sont gérées.

### Calcul du total pour chaque carte de reservation

* Sur la page de réservation, chaque carte de réservation affiche un total calculé en fonction des éléments ou services ajoutés à cette réservation. Ce total est mis à jour automatiquement en fonction des changements apportés.

### Calcul du total général

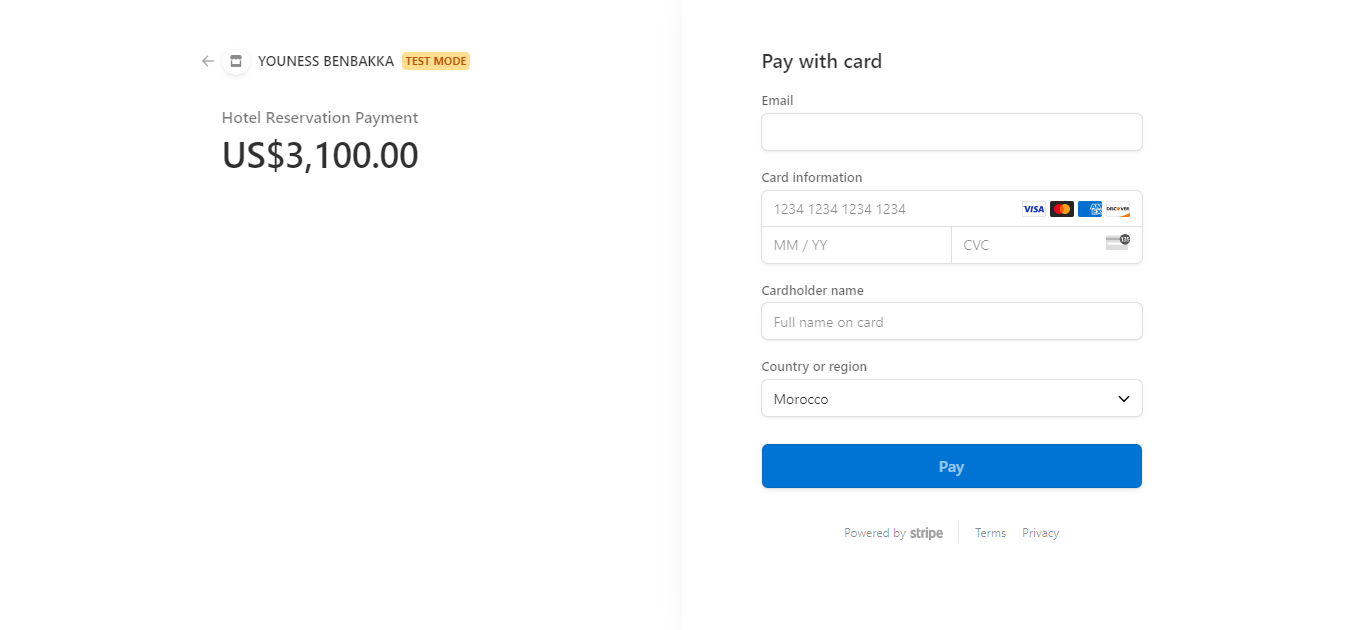
* En plus du total pour chaque carte, un total général combinant toutes les réservations est affiché. Cela permet à l'utilisateur de voir le montant global de ses réservations.

### Fonctionnalités de suppression et de modification

* Chaque carte de réservation dispose d'options pour **modifier** ou **supprimer** la réservation
* **Modifier** : L'utilisateur peut cliquer sur un bouton pour modifier les détails de la réservation, comme les services ajoutés ou les quantités. Une fois les modifications apportées, les totaux sont recalculés automatiquement.
* **Supprimer** : L'utilisateur a la possibilité de supprimer une carte de réservation, ce qui met à jour le total général en conséquence.

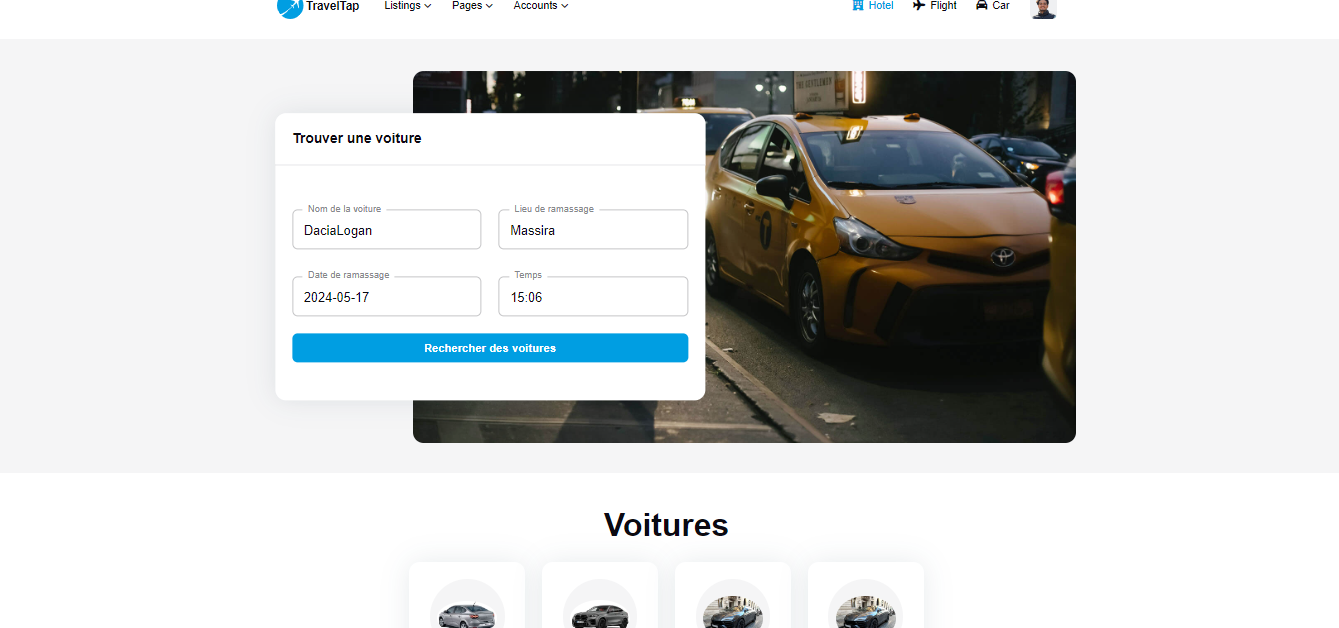
### Confirmation de la réservation

* Enfin, après avoir vérifié et éventuellement modifié ou supprimé certaines réservations, l'utilisateur peut confirmer l'ensemble de ses réservations via un bouton de confirmation.

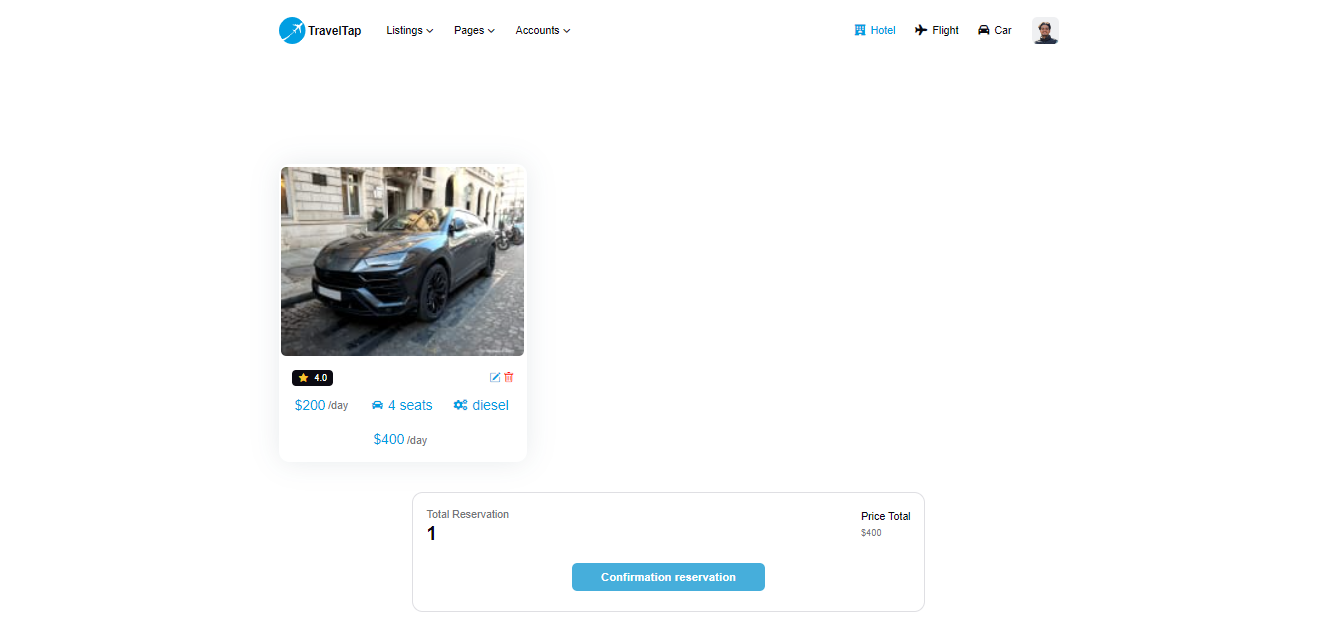


*Figure 5.5: paiement Stripe*

* page de paiement pour une réservation d'hôtel. Après avoir cliqué sur "Confirmation de la réservation", l'utilisateur est redirigé vers cette page où il peut finaliser le paiement de 3 100 USD. Il doit saisir son email, les détails de sa carte bancaire, et le pays ou la région, puis cliquer sur "Pay" pour valider la transaction. Le paiement est sécurisé et géré par Stripe.



*Figure 5.6: page index voiture pour recherche*



*Figure 5.7: card reservation voiture*

### Redirection après enregistrement

* Après que l'utilisateur ait rempli les champs et cliqué sur le bouton "Enregistrer", il est redirigé automatiquement vers la page "Cars Reservations", où les réservations sont gérées.

### Calcul du total pour chaque carte de reservation

* Sur la page de réservation, chaque carte de réservation affiche un total calculé en fonction des éléments ou services ajoutés à cette réservation. Ce total est mis à jour automatiquement en fonction des changements apportés.

### Calcul du total général

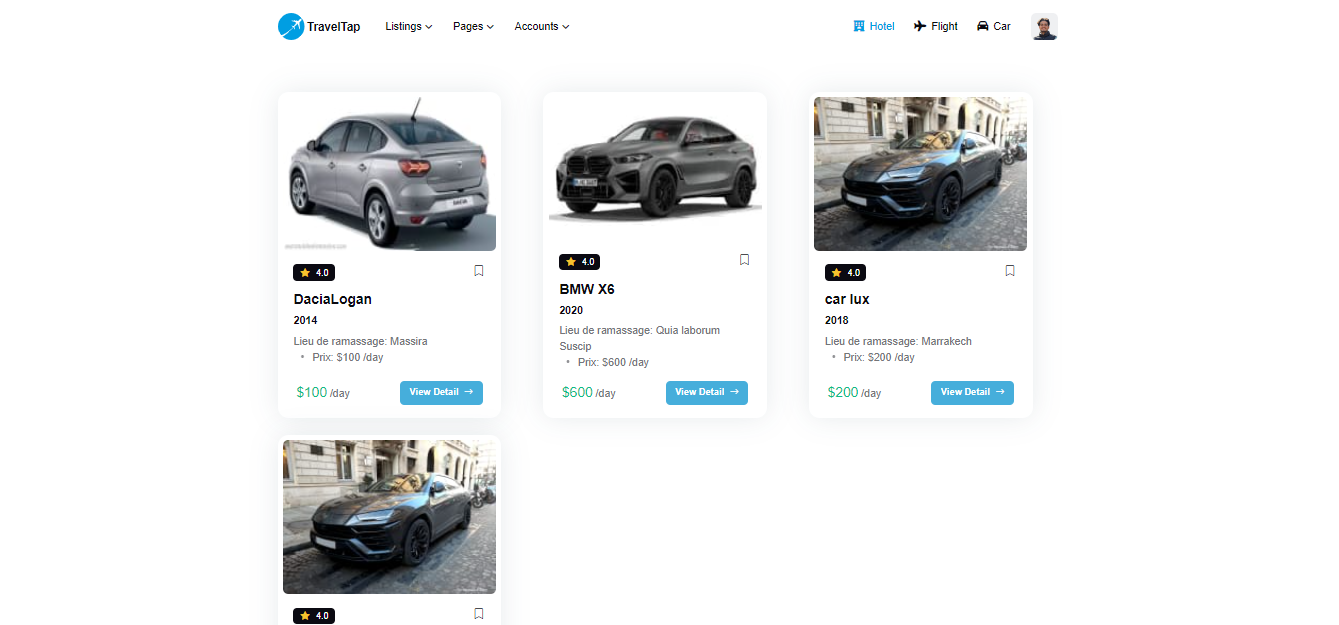
* En plus du total pour chaque carte, un total général combinant toutes les réservations est affiché. Cela permet à l'utilisateur de voir le montant global de ses réservations.

### Fonctionnalités de suppression et de modification

* Chaque carte de réservation dispose d'options pour **modifier** ou **supprimer** la réservation
* **Modifier** : L'utilisateur peut cliquer sur un bouton pour modifier les détails de la réservation, comme les services ajoutés ou les quantités. Une fois les modifications apportées, les totaux sont recalculés automatiquement.
* **Supprimer** : L'utilisateur a la possibilité de supprimer une carte de réservation, ce qui met à jour le total général en conséquence.

### Confirmation de la réservation

* Enfin, après avoir vérifié et éventuellement modifié ou supprimé certaines réservations, l'utilisateur peut confirmer l'ensemble de ses réservations via un bouton de confirmation.



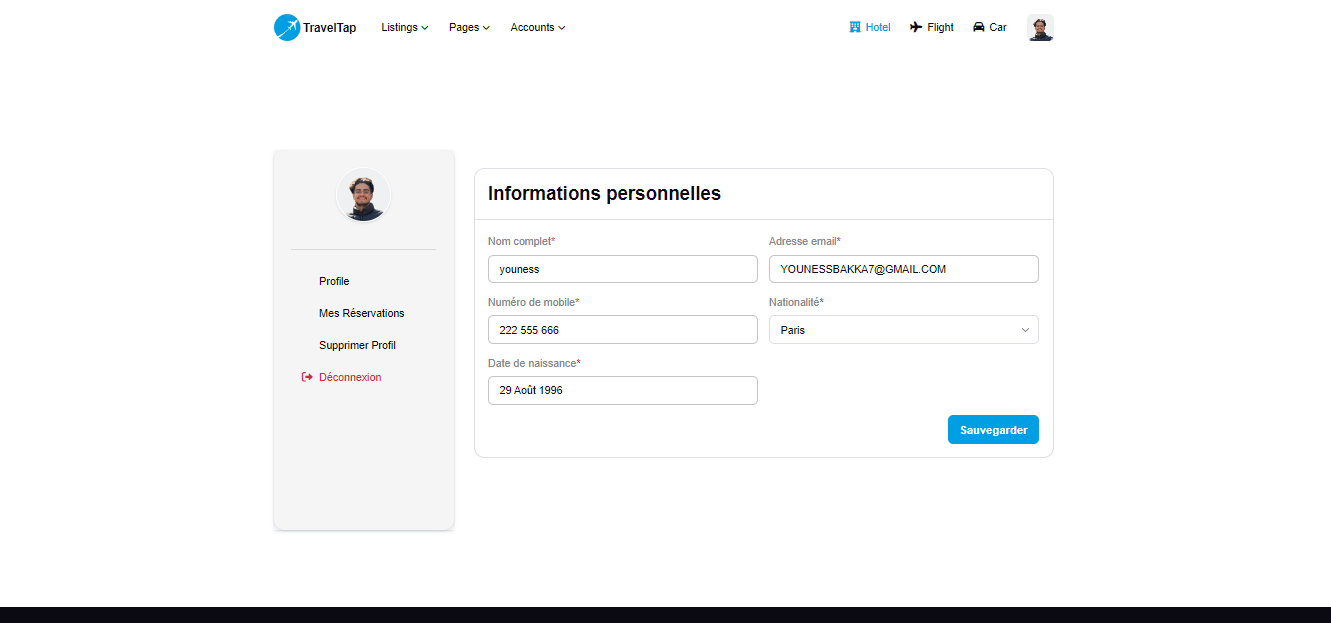
*Figure 5.8: liste des voiture*

Cette Page affiche tous les voiture avec button details pour affiche autre information.



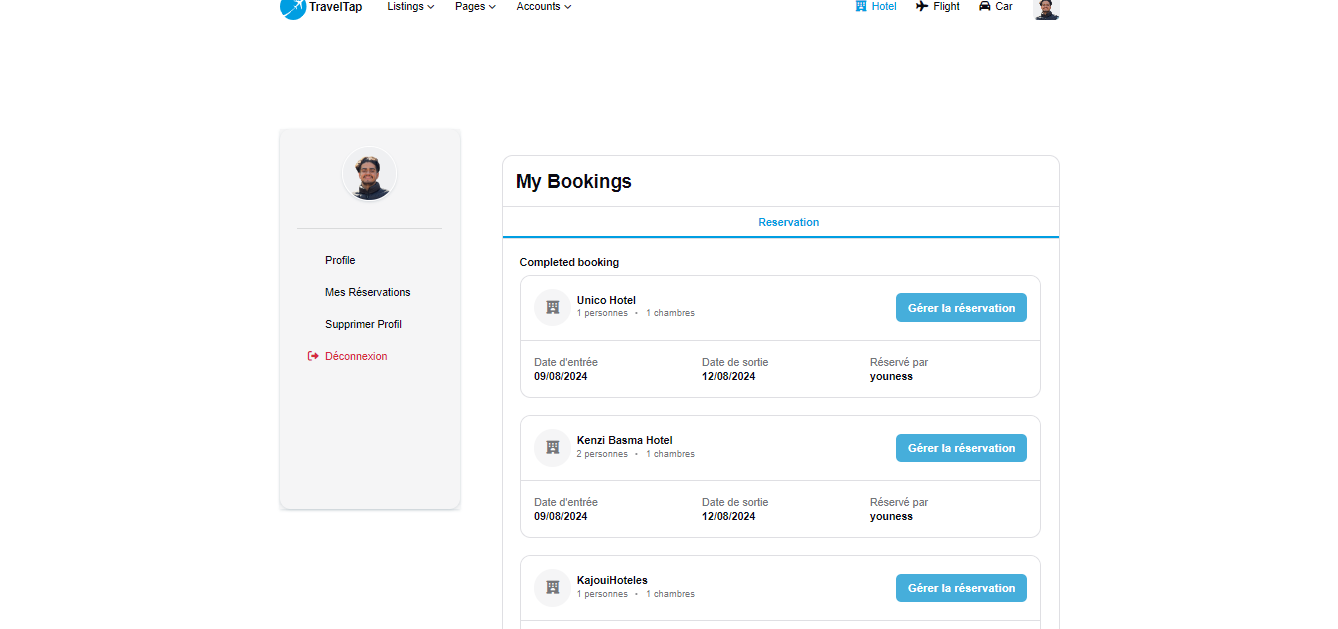
*Figure 5.9: details Voiture*

* Cliquez boutton details pour affiche autre information. Et pour effecter reservation.



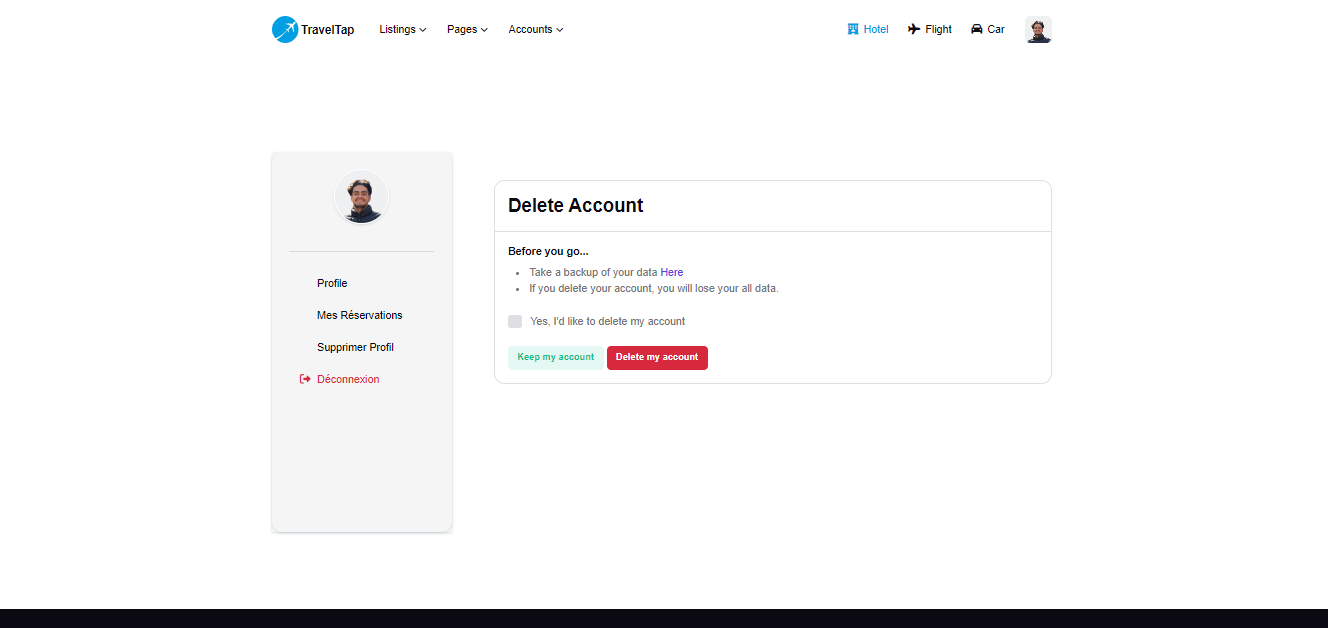
*Figure 6.1: parameter client*

* Cette partie affiche le profil du client, incluant ses informations personnelles telles que le nom, l'email, le numéro de téléphone, et la date de naissance.



*Figure 6.2: reservation client*

Cette section affiche toutes les réservations du client, que ce soit pour des hôtels, des vols ou des voitures, avec un bouton pour gérer chaque réservation.



*Figure 6.3: Supprimer le compte*

* Cette section, si vous cliquez sur le bouton "Supprimer le compte", affiche une carte de confirmation demandant à l'utilisateur de vérifier s'il souhaite sérieusement effacer ce compte.

**Conclusion**

En conclusion, notre projet de gestion des réservations de voyages a été réalisé avec succès en utilisant des technologies avancées pour développer un système intégré de gestion des réservations de vols, d’hôtels, et de voitures. Ce système offre une interface conviviale permettant aux clients de consulter et gérer leurs réservations, tandis que les responsables peuvent facilement administrer les ressources disponibles. L'interface administrateur facilite la gestion globale des utilisateurs, des réservations, et des paiements, tout en offrant des rapports analytiques sur les performances des services proposés.

Nous avons dû relever certains défis au cours de la réalisation, notamment l’intégration fluide de l’interface utilisateur avec la logique côté serveur et la gestion adéquate de la base de données pour assurer la cohérence et la sécurité des transactions. Grâce à une planification minutieuse, une analyse approfondie des besoins, et une collaboration étroite entre les membres de l’équipe, nous avons surmonté ces obstacles pour aboutir à un système fonctionnel qui répond aux exigences des utilisateurs finaux.

En somme, ce projet de gestion des réservations de voyages met en évidence l’application concrète des technologies choisies pour créer une plateforme de réservation complète, soulignant à la fois les réussites et les solutions adoptées pour faire face aux défis rencontrés. Ce projet illustre l’impact positif que les technologies de l’information peuvent avoir sur l’efficacité administrative et l’expérience utilisateur, et constitue un modèle pour d’autres entreprises souhaitant moderniser leurs services de réservation.

**Biographie et Webographie**

* W3Schools - Tutoriels HTML, CSS, et JavaScript
* Gestion de bases de données avec Mongodb:
* Développement d'applications Web avec HTML, CSS et NodeJS, JavaScript .
* Bootstrap Documentation: Bootstrap Documentation .
* Cloudinary : pour enregester images