

Documentation UML

Diagramme UML Utilisation :

Les éléments du diagramme :

Acteur :

Les différents acteurs dans ce diagramme sont les suivants :

- Utilisateur
- Compagnie
- Agence

Use-Case : Le diagramme de cas d'utilisation décrit les interactions possibles des acteurs avec le système en question

Cas d'Utilisation Décrits :

Créer un utilisateur :

- Cela permet à un nouvel utilisateur de s'inscrire dans le système. Il peut s'agir d'un processus d'inscription où l'utilisateur fournit ses données personnelles.

Gérer les données clients :

- il implique la gestion des informations des clients par la compagnie (savoir qui à réserver que ce soit du client ou le passager même par exemple)

S'identifier :

- Obligation des utilisateurs de se connecter pour interagir avec le système .

Choix passager :

- Suite à la connexion et à une réservation l'utilisateur choisi ou créer son passager pour l'identifier pour le vol

Réservation vol :

- La réservation permet de choisir son passager et le vol .
- De plus l'utilisateur peut annuler la réservation et la confirmer suite à une requête de validation

Vol :

- Représente la sélection des différents vol sur l' Application

Ferme la réservation :

- Permet de finaliser et clore la réservation

Annulation Vol :

- L'utilisateur a la possibilité d'annulé son vol que ce soit le client ou la compagnie

Identification Vol :

- Permet d'identifier le vol avec différente spécificité tel quel le numéro vol ou critère de recherche

Ouvre la réservation :

- Permet d'ouvrir ou Ré-ouvrir les réservation

Création vol :

- Permet l'ajout de vol par les compagnie vers l'agence

Affichage vol :

- Permet au utilisateur de voir les différents details d'un vol spécifique

Gérer la gestion des vols :

- Permet la gestion des vols par l'agence comprenant l'affichage , l'ajout des vols sur l'application ...

Règles de Gestion Intégrées

Le modèle intègre également des règles de gestion, comme indiqué par les relations d'extension (extends) et d'inclusion (include) entre les différents cas d'utilisation.

- "Confirmation réservation" et "Annulation réservation" sont conditionnées par le cas d'utilisation "Réservation vol ».

- "Ouvre la réservation" et "Ferme la réservation" sont des extensions du cas "Vol", indiquant que ces actions ne peuvent être entreprises que dans le contexte d'un vol sélectionné.

- « Ajout du vol" et "Affichage du vol" sont inclus dans "Création du vol", ce qui signifie qu'ils sont des étapes nécessaires dans le processus de création d'un vol.

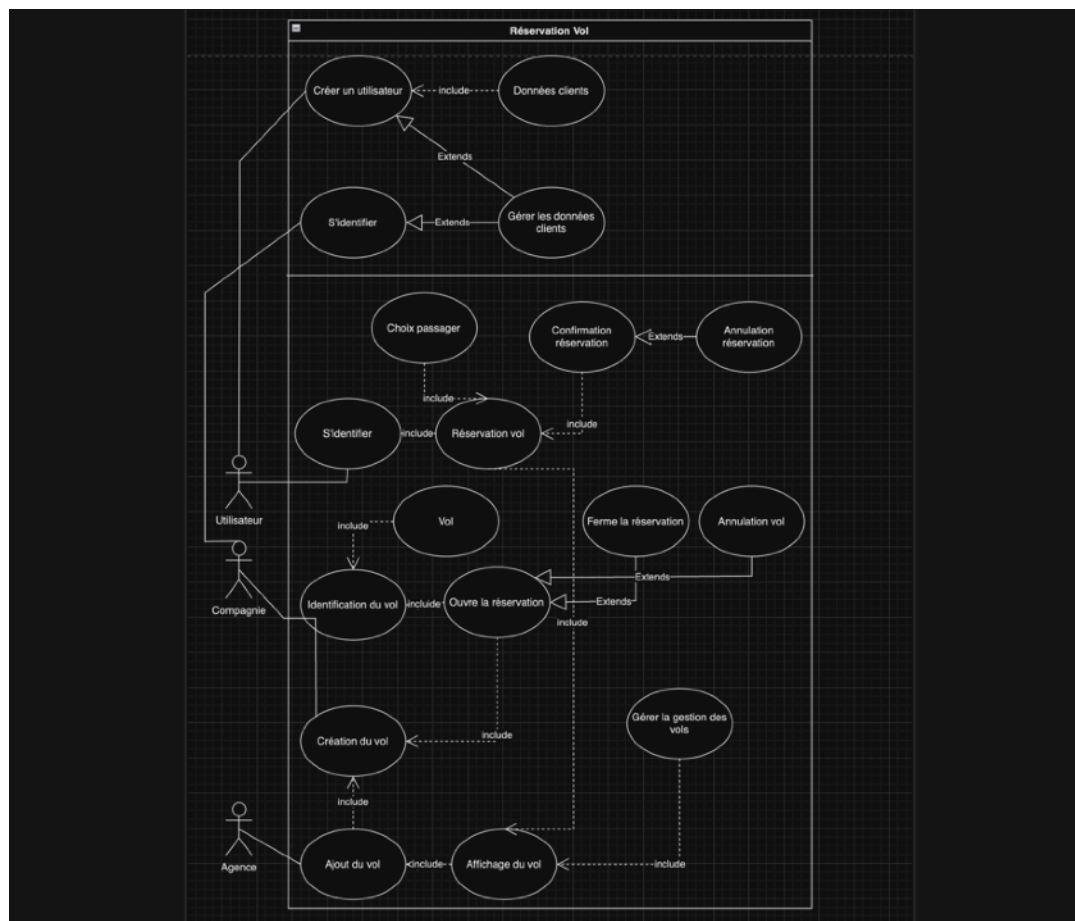


Diagramme UML Classes :

- **User :**
 - Représente un utilisateur du système.
 - Attributs : id, mail, password, role.
 - Méthodes : login(), logOut(), createUser(), modifyUser(), deleteUser().
 - Relation : Un utilisateur peut être un client (Customer).
- **Customer :**
 - Spécialisation de User pour les clients.
 - Attributs : id, item (attribut spécifique à Customer).
 - Méthodes : buyReservation(), cancelReservation(), seeReservationHistory().
 - Relation : Un client peut avoir plusieurs réservations.
- **Reservation :**
 - Représente une réservation.
 - Attributs : id, confirmed (booléen).
 - Méthodes : createPassenger(), recupPassenger() (récupérer des informations sur le passager), ShowPassengerInfo().
 - Relations : Liée à un passager et peut-être liée à une ou plusieurs instances de Customer.
- **Passenger :**
 - Représente une personne voyageant.
 - Attributs : id, nom, prénom, date de naissance, adresse, nationalité, état, ville, téléphone.
 - Relation : Associé à une réservation.
- **Flight :**
 - Représente un vol.
 - Attributs : id, numéro, heure de départ, heure d'arrivée, escale (booléen), retard (booléen).
 - Méthodes : isStopover() (retourne une instance Stopover si une escale est prévue), showFlightInfos().
 - Relation : Peut avoir un ou plusieurs arrêts (Stopover) et est créé par une compagnie (Company).
- **Stopover :**
 - Représente une escale d'un vol.
 - Attributs : id, heure de départ, heure d'arrivée.
 - Relation : Associé à un vol.
-

- **Agency**
 - Représente une agence de voyages (nous).
 - Attributs : id, agency (nom de l'agence).
 - Méthodes : addFlight(), addCompany(), closeFlight(), closeCompany().
 - Relation : Peut avoir plusieurs compagnies et vols associés.
- **Company :**
 - Représente une compagnie aérienne.
 - Attributs : id, nom.
 - Méthodes : createFlight(), deleteFlight().
 - Relation : Crée des vols et peut être associée à une agence.
- **Airport :**
 - Représente un aéroport.
 - Attributs : id, nom.
 - Méthodes : ShowAirportInfo().
 - Relation : Peut être associé à plusieurs vols.

Diagramme UML Sequence :

Diagramme Réservation :

- **Acteurs** : User, Agency, Customer, Centre d'autorisation, Passenger
- **Processus de réservation** : Commence par la recherche de vol par l'utilisateur et l'affichage des vols disponibles.
- **Sélection et informations du vol** : L'utilisateur sélectionne un vol et affiche ses informations.
- **Alternatives (ALT)** : Si l'utilisateur veut réserver, il doit être inscrit et connecté. Sinon, il est redirigé vers une connexion/inscription.
- **Références (REF)** : Des processus de connexion/inscription séparés sont référencés ici.
- **Formulaire de réservation** : L'utilisateur remplit et envoie le formulaire de réservation.
- **Optimisation (OPT)** : L'utilisateur peut choisir d'enregistrer ses données pour une utilisation future.
-

- **Boucle de validation** : Si les données sont incorrectes, l'utilisateur est invité à les corriger jusqu'à ce qu'elles soient validées.
- **Processus de paiement** : L'utilisateur utilise une carte bancaire (CB) pour le paiement, qui doit être autorisé par le centre.
- **Confirmation** : Après le paiement, la réservation est confirmée et les informations sont envoyées.

Diagramme Connexion / Inscription & Modification :

- **Connexion :**
 - L'utilisateur initie la connexion et entre les informations requises dans une boucle jusqu'à succès.
 - Les informations sont vérifiées par la base de données.
 - En cas d'échec, un message d'erreur s'affiche; en cas de succès, l'utilisateur est redirigé vers l'accueil.
- **Inscription :**
 - L'utilisateur remplit un formulaire d'inscription.
 - La base de données vérifie l'unicité des données soumises.
 - Si les données ne sont pas uniques, l'utilisateur est invité à modifier le formulaire; si elles le sont, le compte est créé.
- **Modification de compte :**
 - L'utilisateur doit être connecté pour modifier son compte.
 - Le formulaire de modification s'affiche dans une boucle jusqu'à ce que les données valides soient soumises.
 - La base de données valide les modifications.

Diagramme Creation vol / Ajout Vol :

- **Créer un vol de compagnie :**
 - L'utilisateur doit être connecté ou inscrit.
 - Un formulaire de création de vol est envoyé par la compagnie, rempli par l'utilisateur, et retourné.
 - L'agence vérifie les données.
 - Si les données sont validées, le vol est créé et enregistré; sinon, une boucle d'erreur permet de corriger et renvoyer le formulaire.
-
-
-

- **Ajouter un vol :**
- - La procédure commence par une référence à la connexion ou inscription.
 - L'agence envoie un formulaire pour ajouter des informations sur le vol.
 - L'utilisateur remplit et retourne le formulaire, qui est alors vérifié.
 - Si les données sont validées, le vol est ajouté et enregistré; sinon, une boucle d'erreur permet de corriger et renvoyer le formulaire.
-