CERICar

Application de covoiturage

Rendu UML – Étape 1

21/11/2025

GHABI Malek

L3 Informatique — Semestre 5

Année scolaire 2025–2026

Sommaire

1. Introduction

Cette section présente brièvement le projet CERICar ainsi que l'objectif du document. Elle explique le contexte du travail demandé et introduit la modélisation UML réalisée dans le cadre de l'étape 1 du projet AMS.

2. Diagrammes UML réalisés

Cette partie regroupe l'ensemble des diagrammes UML nécessaires à la conception du système. Chaque diagramme apporte une vision différente : fonctionnalités, acteurs, structure des données et déroulement des interactions

2.1 Description des cas d'utilisation (Use Cases)

Use cases – Visiteur (non connecté)

➤ **S'inscrire**

- **Objectif :** Permettre au visiteur de créer un compte afin d'accéder à toutes les fonctionnalités du système CERICar.

- **Scénario :**
  1. Le visiteur clique sur « *S'inscrire* ».
  2. Il saisit ses informations personnelles (nom, prénom, pseudo, e-mail, mot de passe).
  3. Il peut renseigner un numéro de permis (optionnel).
  4. Il valide le formulaire.
  5. Le système vérifie la conformité des champs et l'unicité du pseudo.
  6. Si les données sont correctes, le système crée le compte et affiche un message de confirmation.

- **Post-condition :** Le compte du visiteur est créé dans la base, et il peut désormais se connecter.

➤ **Se connecter**

- **Objectif :** Permettre au visiteur de s'authentifier pour accéder à son espace personnel.

- **Scénario :**
  1. Le visiteur clique sur « *Se connecter* ».
  2. Il saisit son identifiant et son mot de passe.
  3. Le système compare les informations saisies avec celles enregistrées.
  4. Si elles sont valides, la session s'ouvre et l'utilisateur devient "connecté".
  5. Sinon, le système affiche un message d'erreur.

- **Post-condition :** L'utilisateur est authentifié et redirigé vers son tableau de bord.

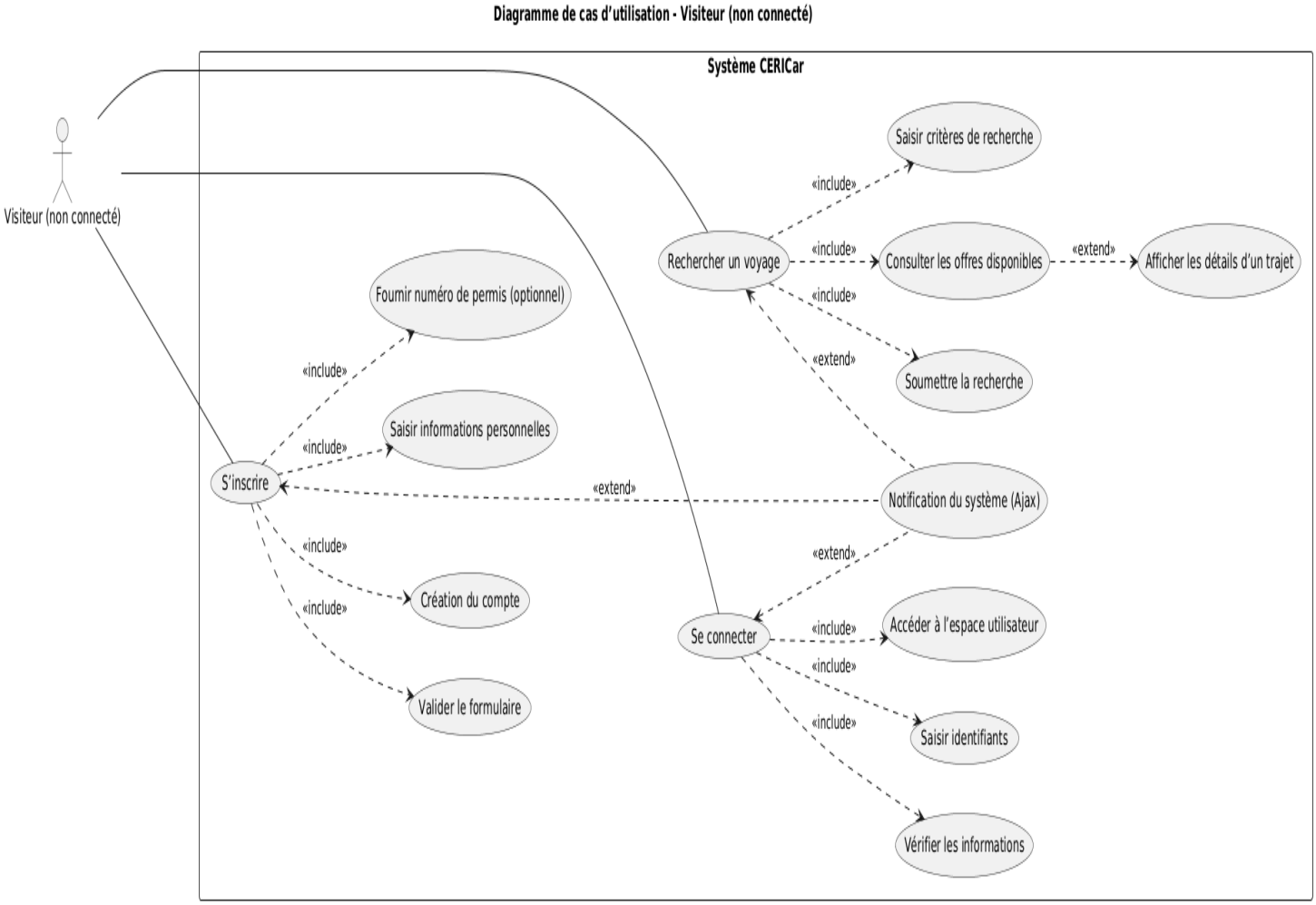
➤ Rechercher un voyage

• **Objectif :** Permettre au visiteur d'effectuer une recherche de trajets avant d'être inscrit.

• **Scénario :**

1. Le visiteur saisit la ville de départ, d'arrivée et le nombre de passagers.
2. Il valide la recherche.
3. Le système interroge la base de données.
4. Les résultats correspondants sont affichés à l'écran.
5. Le visiteur peut consulter les détails d'un trajet mais pas le réserver.

• **Post-condition :** Les trajets trouvés sont affichés, mais la réservation nécessite une connexion.



Use cases – Utilisateur connecté (Conducteur)

➤ Proposer un voyage

• **Objectif :** Permettre au conducteur de publier une offre de covoiturage en précisant les informations du trajet et du véhicule.

• **Scénario :**

1. Le conducteur accède à la section *« Proposer un voyage »*.
2. Il saisit les informations du trajet : départ, arrivée, date, heure, tarif, nombre de places.
3. Il renseigne les informations du véhicule : modèle, type, bagages, contraintes.
4. Il valide la proposition.
5. Le système enregistre le trajet et affiche une confirmation.

• **Post-condition :** Le trajet est ajouté à la base et visible dans la liste des offres de covoiturage.

➤ Consulter ses voyages proposés

• **Objectif :** Permettre au conducteur de visualiser et gérer les trajets qu'il a publiés.

• **Scénario :**

1. Le conducteur sélectionne *« Mes trajets proposés »*.
2. Le système affiche la liste de ses trajets.
3. Il peut consulter les détails d'un trajet.
4. Il peut modifier ou supprimer une offre existante.
5. Le système met à jour les informations en conséquence.

• **Post-condition :** Les trajets proposés sont actualisés selon les modifications effectuées.

➤ Voir les réservations reçues

• **Objectif :** Permettre au conducteur de consulter et gérer les réservations faites par les voyageurs sur ses trajets.

• **Scénario :**

1. Le conducteur accède à la section *« Réservations reçues »*.
2. Le système affiche la liste des réservations associées à ses trajets.
3. Il peut consulter le détail (nom du voyageur, nombre de places, message).
4. Il confirme ou refuse la réservation.
5. Le système met à jour le statut de la réservation et notifie le voyageur.

• **Post-condition :** Les réservations validées sont confirmées et enregistrées dans la base.

➤ Consulter profil / informations personnelles

• **Objectif :** Permettre au conducteur de consulter et modifier ses informations personnelles.

• **Scénario :**

1. Le conducteur ouvre la page *« Mon profil »*.
2. Le système affiche ses données (nom, email, permis, véhicule).
3. Il peut modifier ou mettre à jour ses informations.
4. Le système enregistre les changements et affiche une confirmation.

• **Post-condition :** Les informations du conducteur sont mises à jour dans la base de données.

➤ Se déconnecter

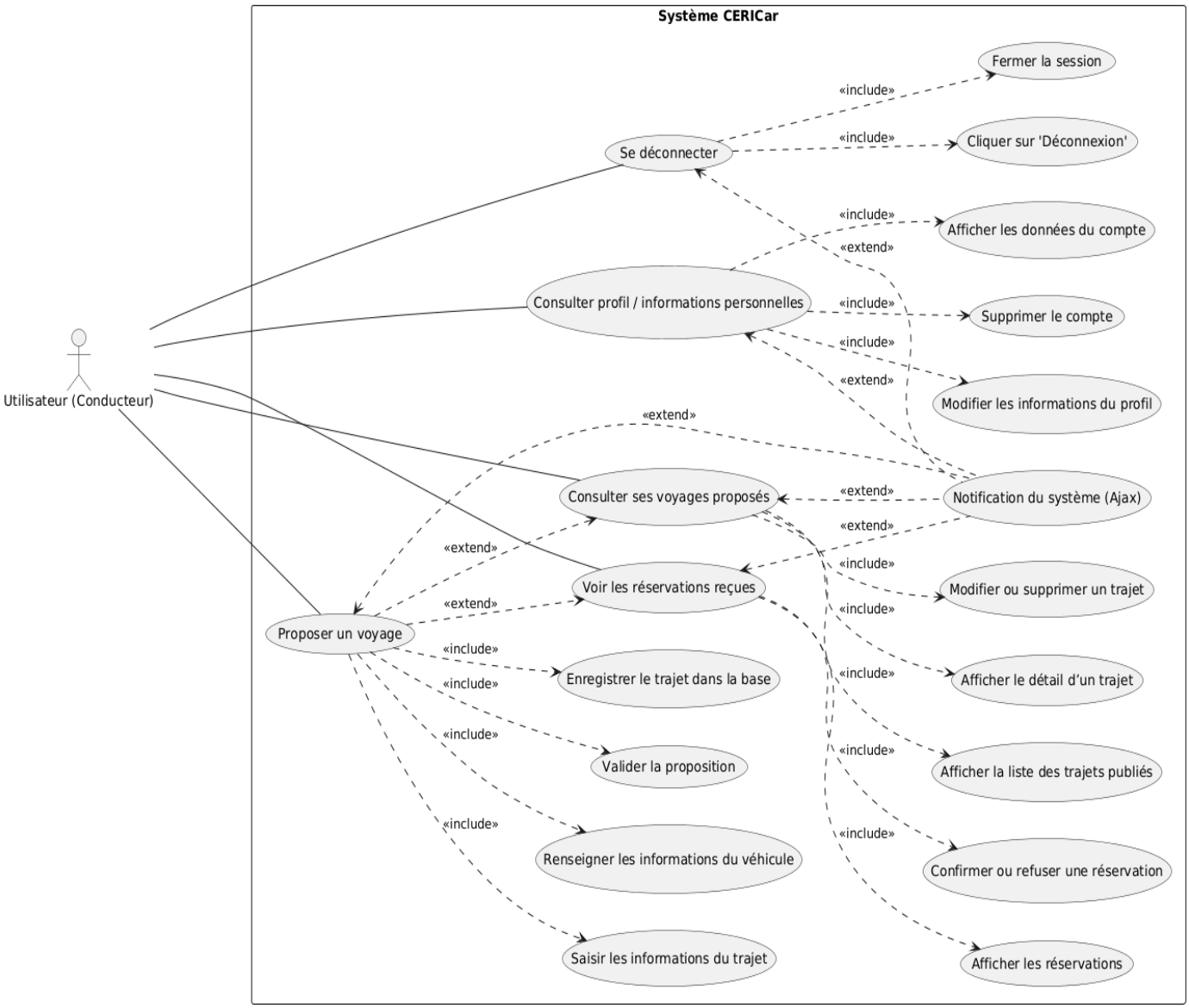
• **Objectif :** Permettre au conducteur de quitter sa session utilisateur.

• **Scénario :**

1. Le conducteur clique sur *« Déconnexion »*.
2. Le système ferme la session et supprime les données de connexion.
3. L'utilisateur revient sur la page d'accueil.

• **Post-condition :** La session est fermée et le conducteur redevient visiteur.

Diagramme de cas d'utilisation - Utilisateur connecté (Conducteur)



Use cases : Utilisateur connecté (Voyageur

➤ Rechercher un voyage

• **Objectif :** Permettre au voyageur de rechercher un trajet selon ses critères.

• **Scénario :**

1. L'utilisateur saisit les villes de départ et d'arrivée, la date et le nombre de voyageurs.
2. Il valide la recherche.
3. Le système affiche la liste des trajets correspondants.
4. Il peut consulter les détails avant réservation.

• **Post-condition :** Les offres disponibles sont affichées pour consultation ou réservation.

➤ Réserver un voyage

• **Objectif :** Permettre au voyageur de réserver une ou plusieurs places sur un trajet.

• **Scénario :**

1. Le voyageur sélectionne un trajet dans la liste des résultats.
2. Il indique le nombre de places à réserver.
3. Il confirme sa réservation.
4. Le système vérifie la disponibilité et enregistre la réservation.
5. Un message de confirmation s'affiche.

• **Post-condition :** La réservation est enregistrée et visible dans le profil du voyageur.

➤ Voir les réservations reçues

• **Objectif :** Permettre au conducteur de consulter et gérer les réservations faites par les voyageurs sur ses trajets.

• **Scénario :**

1. Le conducteur accède à la section *« Réservations reçues »*.
2. Le système affiche la liste des réservations associées à ses trajets.
3. Il peut consulter le détail (nom du voyageur, nombre de places, message).
4. Il confirme ou refuse la réservation.
5. Le système met à jour le statut de la réservation et notifie le voyageur.

• **Post-condition :** Les réservations validées sont confirmées et enregistrées dans la base.

➤ Consulter profil / réservations

• **Objectif :** Permettre au voyageur de visualiser ses informations personnelles et ses réservations.

• **Scénario :**

1. L'utilisateur accède à *« Mon profil »*.
2. Le système affiche ses données et la liste de ses réservations.
3. Il peut consulter les détails d'un trajet ou annuler une réservation.
4. Le système met à jour la base de données.

• **Post-condition :** Le profil utilisateur et les réservations sont actualisés.

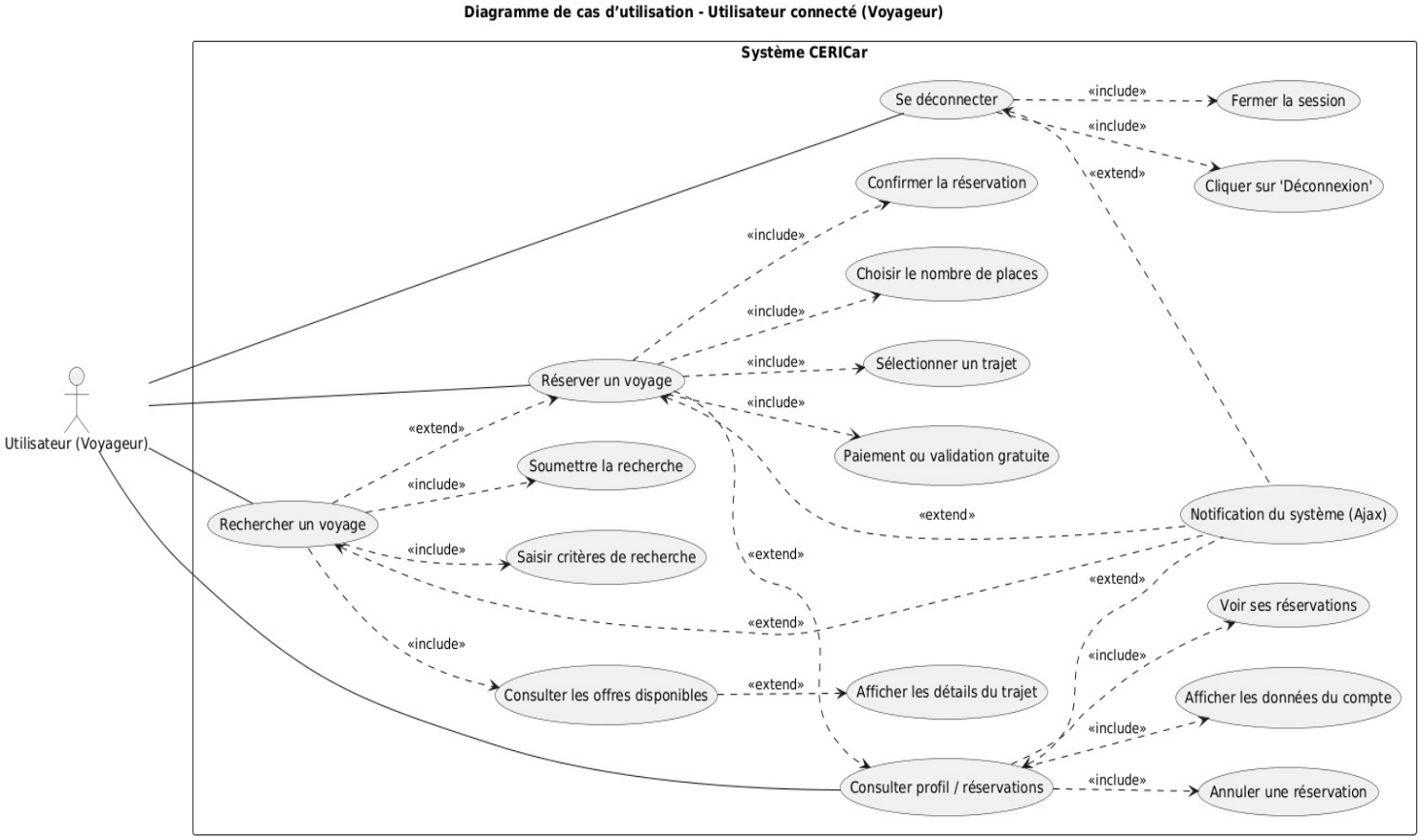
➤ Se déconnecter

• **Objectif :** Permettre au voyageur de quitter la session et revenir à l'état visiteur.

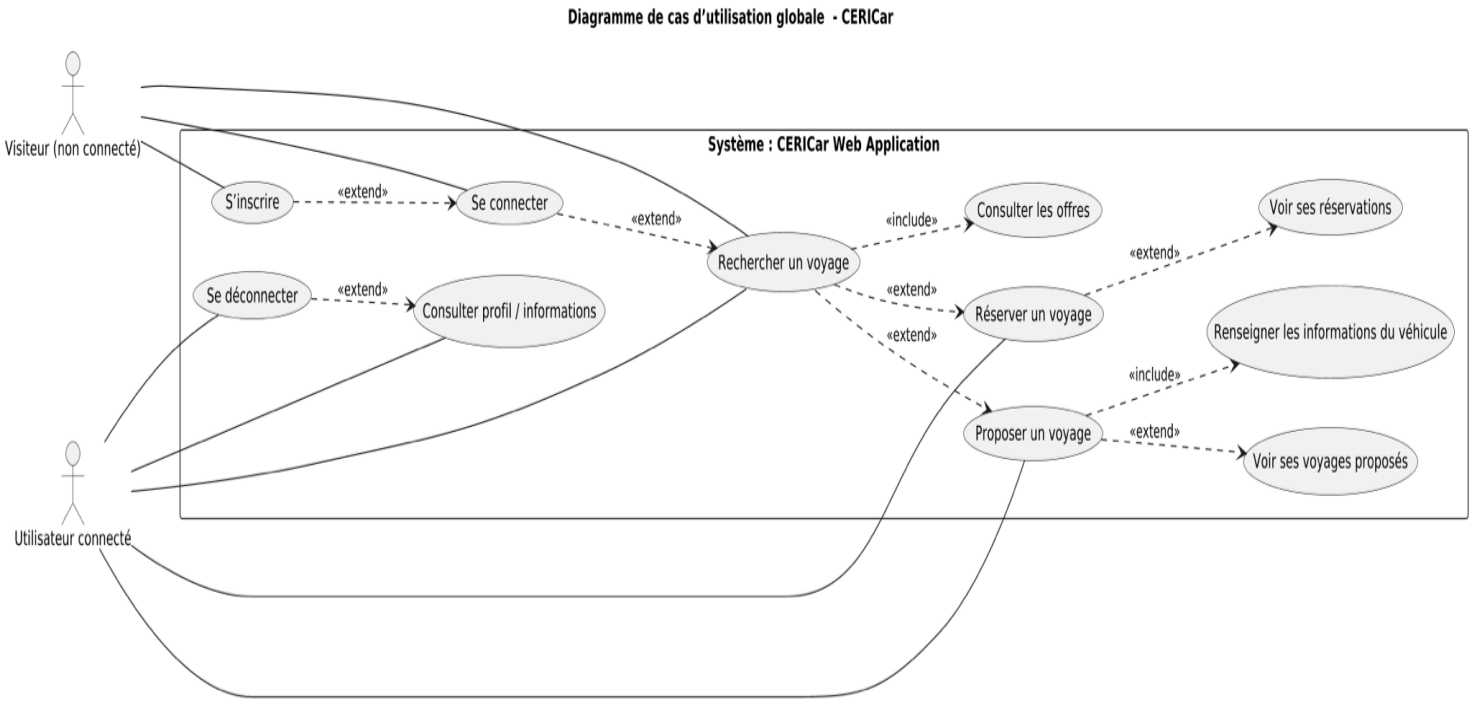
• **Scénario :**

1. L'utilisateur clique sur *« Déconnexion »*.
2. Le système détruit la session active.
3. Il est redirigé vers la page d'accueil publique.

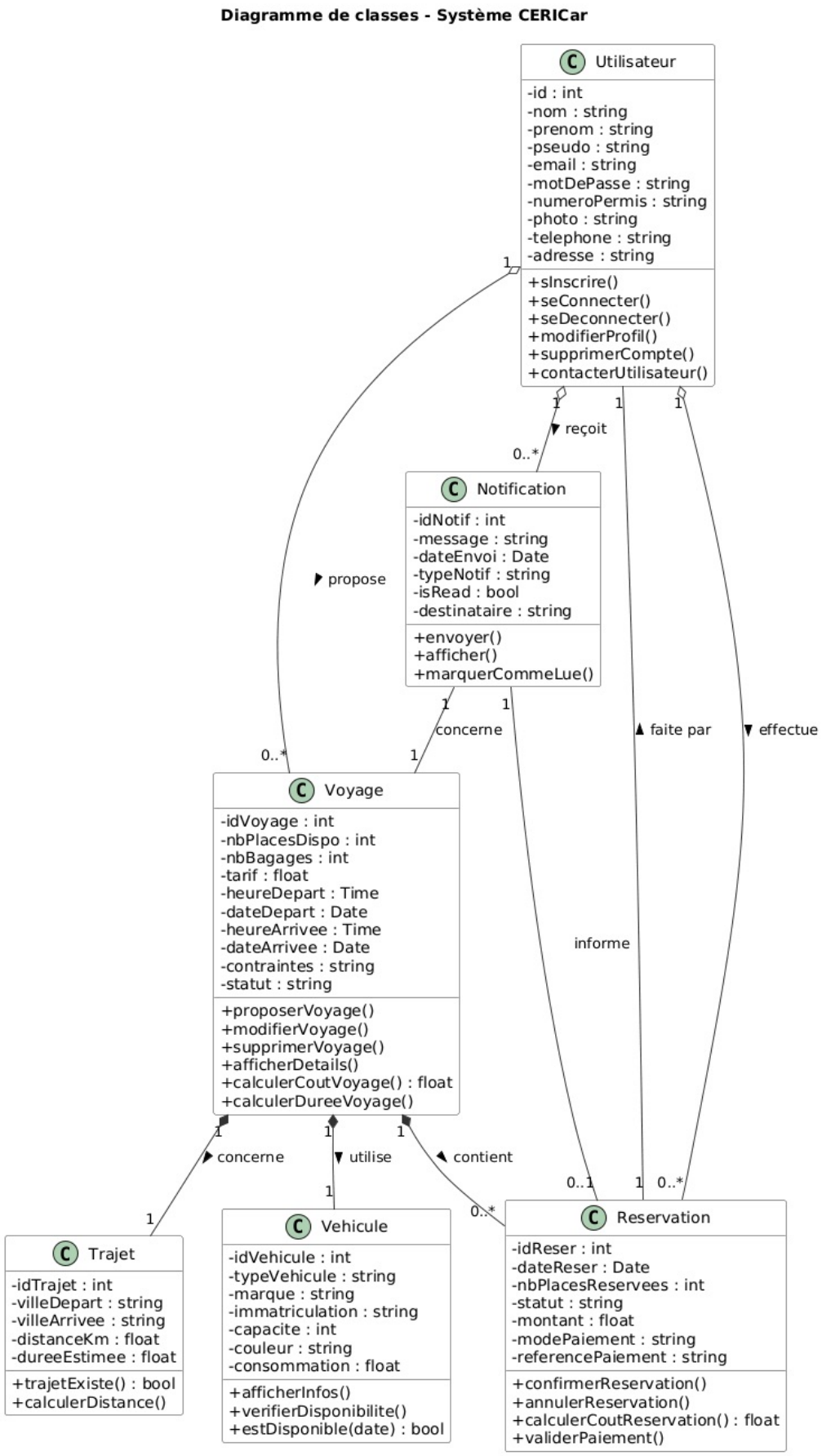
• **Post-condition :** La session est fermée et le conducteur redevient visiteur.



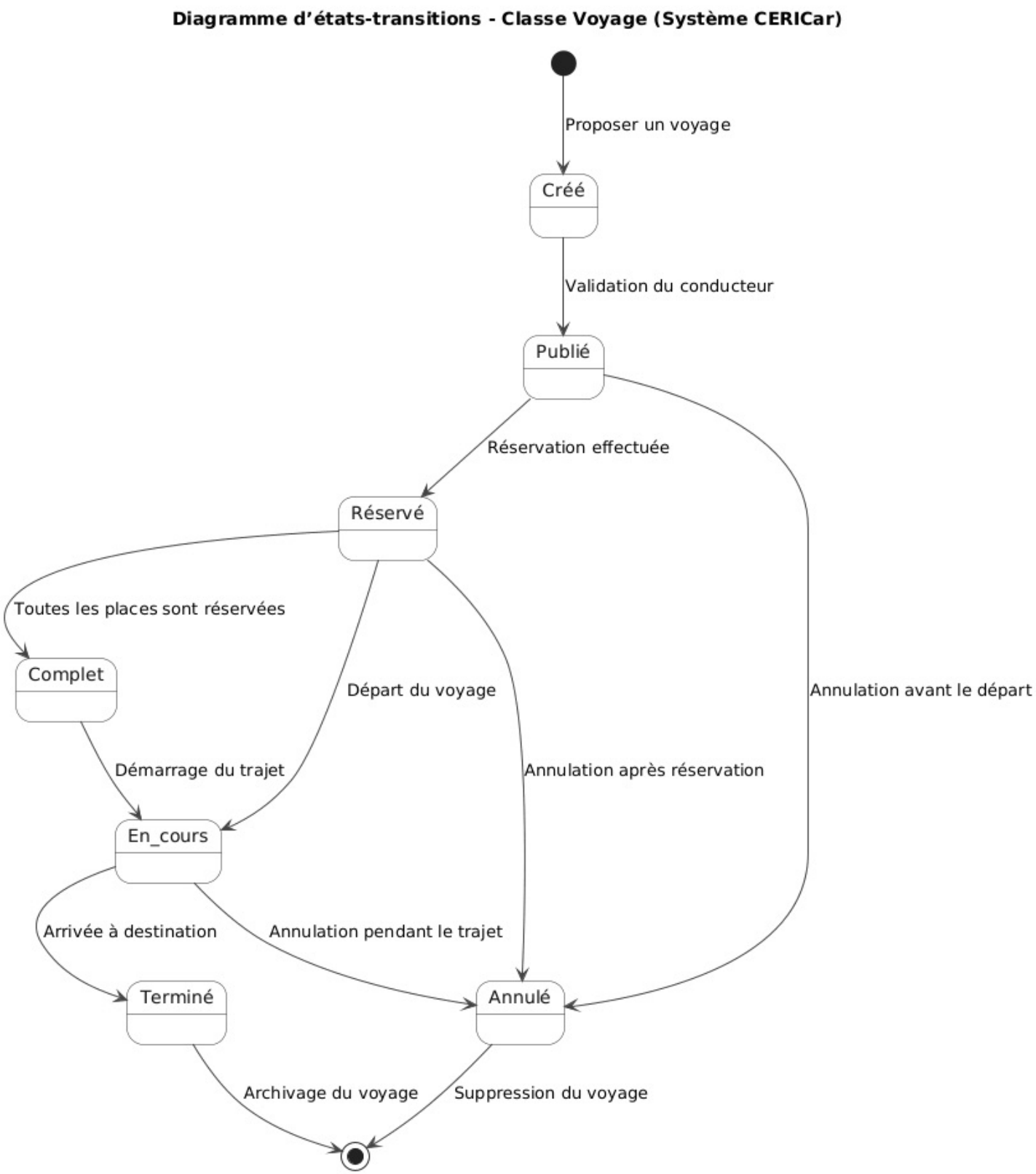
2.2 Diagramme de cas d'utilisation global



2.3 Diagramme de classes



2.4 Diagramme d'états-transitions



2.4 Diagramme de séquence

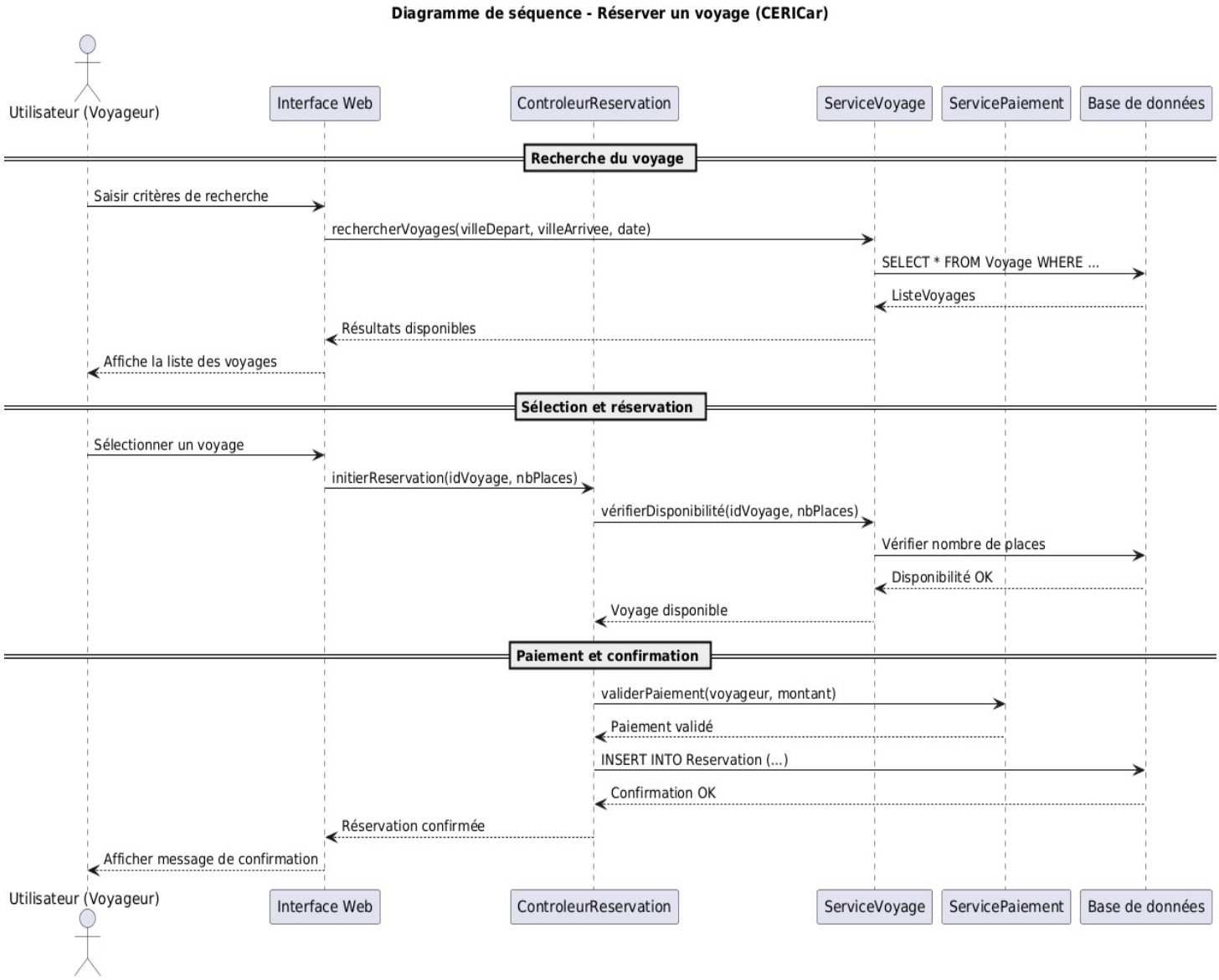
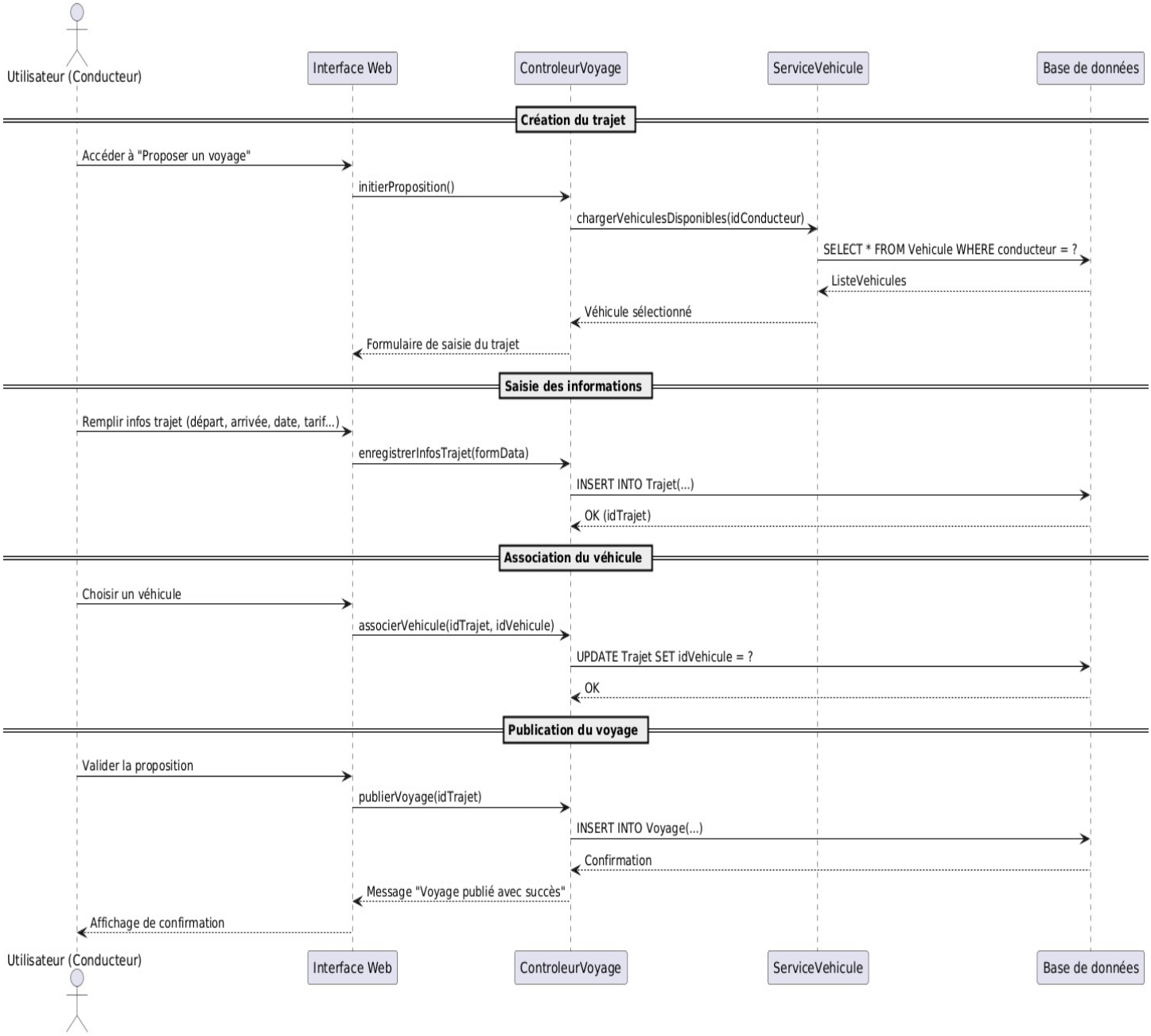


Diagramme de séquence - Proposer un voyage (CERICar)



3.Conclusion

Les diagrammes UML du projet CERICar montrent clairement le fonctionnement du système.

Les cas d'utilisation présentent les actions possibles des utilisateurs.

Le diagramme de classes illustre les principales entités et leurs relations.

Le diagramme d'états-transitions décrit les différentes étapes du voyage.

Enfin, les diagrammes de séquence expliquent comment les utilisateurs interagissent avec le système.


CERICar