

I - DESCRIPTION COMPLÈTE DU SYSTÈME —

TAKA TAKA (Version 1)

Taka Taka est une plateforme de mobilité composée d'une application mobile, d'une application web et d'un tableau de bord administrateur. Elle a pour objectif principal de faciliter la mise en relation entre des passagers qui souhaitent se déplacer et des chauffeurs disponibles dans leur zone. La plateforme permet aux passagers de réserver un trajet en quelques secondes, d'être géo localisés, de suivre l'arrivée du chauffeur en temps réel et de gérer le paiement du service. Du côté des chauffeurs, le système leur offre la possibilité de recevoir des demandes de courses, d'accepter ou de refuser ces demandes, de suivre la navigation sur carte et de gérer leurs revenus au quotidien. Enfin, un administrateur supervise l'ensemble du système, valide les chauffeurs, contrôle les opérations et assure la qualité du service.

Taka Taka s'articule autour de trois acteurs principaux : le passager, le chauffeur et l'administrateur.

Le passager est une personne disposant d'un smartphone, capable de créer un compte personnel dans l'application. Il est caractérisé par des informations d'identification telles que son nom, son numéro de téléphone, une photo de profil et son historique d'utilisation. Le passager a la possibilité de réserver un trajet en choisissant un point de départ et une destination. Lorsqu'il effectue une réservation, le système estime automatiquement le prix du trajet en fonction de la distance et lui affiche les chauffeurs disponibles à proximité. Après confirmation, le passager peut suivre en temps réel l'arrivée du chauffeur grâce au GPS intégré. Une fois la course réalisée, il peut consulter son historique, payer en espèces ou via le mobile money et laisser une note ou un avis sur le chauffeur.

Le chauffeur, quant à lui, est un utilisateur professionnel du système. Il possède un compte chauffeur distinct, validé au préalable par l'administrateur grâce à ses documents (permis, pièce d'identité, immatriculation du véhicule). Le chauffeur est caractérisé par un statut de disponibilité (en ligne ou hors ligne), un véhicule associé, ses documents et ses statistiques de performance. Lorsque le passager envoie une demande de trajet, le chauffeur reçoit une notification instantanée sur son téléphone. Il peut alors accepter ou refuser la demande. En cas d'acceptation, il est guidé vers le point de départ du passager grâce au système de géolocalisation. Durant la course, il peut suivre la navigation, puis signaler la fin du trajet à l'application. Le système calcule alors le montant final et enregistre la course dans son historique. Le chauffeur peut également consulter ses gains au jour le jour.

L'administrateur joue un rôle central dans la gouvernance du système. Il a pour mission de gérer les utilisateurs, particulièrement les chauffeurs dont il doit valider les documents, vérifier l'authenticité et assurer la conformité aux règles du transport. Il supervise les trajets en cours, les paiements, les litiges, et peut intervenir en cas de problème entre un passager et un chauffeur. L'administrateur accède à un tableau de bord web lui permettant de suivre l'activité globale de la plateforme, consulter les statistiques, bloquer un utilisateur si nécessaire, ou encore analyser les performances de la flotte de chauffeurs.

L'ensemble de ces interactions repose sur un processus bien défini : le passager crée un trajet, le système recherche un chauffeur disponible, le chauffeur confirme ou rejette, la course est réalisée, le paiement est effectué et l'expérience se termine par une évaluation. Ce cycle constitue le cœur du fonctionnement Taka Taka.

Le système intègre également un module GPS permettant de suivre en temps réel la position du chauffeur et du passager. La localisation joue un rôle crucial dans le matching, l'estimation du prix, la navigation et le suivi en direct. Cette dimension temps réel est assurée par des services tels que Firebase pour la synchronisation et Google Maps pour l'affichage.

Au niveau structurel, la plateforme repose sur plusieurs entités fondamentales. L'utilisateur représente la base du système et peut être soit un passager, soit un chauffeur, soit un administrateur. Le chauffeur possède un profil étendu contenant les informations de validation et un véhicule associé. Le trajet constitue l'unité centrale du système : il relie un passager à un chauffeur, possède un statut, une localisation, des timestamps, une distance et un prix. Le paiement est rattaché à un trajet et enregistre la méthode et le montant. Le système enregistre également les notifications envoyées à chaque utilisateur ainsi que les évaluations sous forme de notes ou commentaires.

Toutes ces entités interagissent de manière cohérente pour produire un écosystème fluide et sécurisé. Grâce à elles, on peut modéliser facilement le système en diagrammes UML (cas d'utilisation, classes, séquences) et en schémas de bases de données.

En résumé, Taka Taka est une plateforme complète combinant mobilité, géolocalisation, interaction en temps réel et gestion opérationnelle. Elle offre aux passagers un moyen simple de se déplacer, aux chauffeurs une activité rentable et encadrée, et aux administrateurs une interface professionnelle pour superviser l'ensemble des opérations. Cette description narrative permet de comprendre clairement les besoins, les processus, les entités et les interactions afin de produire une analyse fonctionnelle et une modélisation technique solide.

II - Analyse du Cahier de Charges V1 (Model)

ÉTAPE 1 — DESCRIPTION FONCTIONNELLE DE LA V1

Taka Taka V1 est une plateforme de mise en relation entre passagers et chauffeurs.

Elle permet à un passager de réserver un trajet depuis son téléphone, et à un chauffeur de recevoir, accepter et exécuter ce trajet.

L'application assure un suivi GPS en temps réel, une gestion des paiements de base, une gestion des profils utilisateurs, un historique des courses, et une interface d'administration pour contrôler le fonctionnement global du système.

La Version1 doit être Fonctionnelle, Complète, Stable et Conforme aux exigences minimales de la plateforme de Taka Taka

Elle constitue la base sur laquelle seront développées les versions suivantes (V2, V3).

ÉTAPE 2 — LES ACTEURS CLES ET PROCESSUS DU SYSTEME DE LA VERSION V1

● 1. Passager

Le passager est un utilisateur classique souhaitant se déplacer.

Il possède un compte personnel et peut réserver des trajets.

Il est caractérisé par :

- ❖ Nom.
- ❖ Prénom.
- ❖ Numéro de téléphone.
- ❖ Email.
- ❖ Photo.

Il peut :

- ❖ Créer un compte.
- ❖ Se connecter / Se déconnecter.

- ❖ Rechercher un Trajet (Point Départ, Point Arrivé) = Automatiquement Estimation Tarif + Distance + Prix + Voitures Disponibles + Chauffeur plus Proches.
- ❖ Réserver le Trajet (immédiat).
- ❖ Suivre le Chauffeur.
- ❖ Voir sa position (GPS)
- ❖ Payer la course (début, pendant, fin – choix).
- ❖ Noter le chauffeur.
- ❖ Consulter son historique (listes des trajets, transactions, profil chauffeur)
- ❖ Bouton Urgence

● 2. Chauffeur

Le chauffeur est un utilisateur professionnel validé par l'administrateur.

Il transporte les passagers et gère les trajets.

Il est caractérisé par :

- ❖ Identité.
- ❖ Permis de conduire.
- ❖ Documents.
- ❖ Statut de disponibilité.
- ❖ Véhicule associé.
- ❖ Position GPS.
- ❖ Historiques des courses.
- ❖ Revenus du jour/Semaines/Mois

Il peut :

- ❖ S'inscrire (avec informations Personnelles) + Modal (Pour l'ajout des documents) + VérificationOtp (sms, email – au choix).
- ❖ Être validé par Admin (en vérifiant les documents soumis).
- ❖ Se connecter (email/tel, mot de passe).

- ❖ Activer “En ligne / Hors ligne.
- ❖ Voir les Trajets.
- ❖ Recevoir des demandes.
- ❖ Accepter ou Refuser.
- ❖ Calcul Automatique du Tarif.
- ❖ Contacte (Passager / Chauffeur).
- ❖ Suivre la localisation du passager.
- ❖ Démarrer / Pause / Terminer la course.

● 3. Administrateur

- ❖ Connexion ().
- ❖ Il supervise le système via le Dashboard (Trajets en Cours, Chauffeurs Actifs, Passagers Connectés, Statistiques).

Il peut :

- ❖ Gérer passagers (Lister, Activer ou Réactiver un Compte).
- ❖ Gérer chauffeurs (Lister, Activer ou Réactiver un Compte).
- ❖ Gérer les Trajets (suivre les trajets en temps réel, gérer les tarifs et promotions).
- ❖ Vérifier et Valider les Documents.
- ❖ Gérer les Litiges (Problèmes Signalés).
- ❖ Gérer les Paiements (Suivre les transactions – OM, MM, Cash).
- ❖ Gérer Rapports (Bilan Mensuel / Annuel).
- ❖ Afficher les Trajets les plus fréquents, les plus récents.
- ❖ Consulter statistiques

Cas Pratiques – Simulation – Workflow

Cas 1 — Demande du Trajet par un Passager

- a. Passager ouvre l'application

L'application détecte automatiquement :

La position GPS (pour localiser le lieu de départ)

- b. Il saisit la destination

Dans un champ (Où Aller – Vous ?)

Dès que la destination est connue :

Le Système détecte automatiquement : (la distance, le prix estimé, le temps estimé, les chauffeurs disponibles à proximité avec leurs voitures disponibles (nombre de place libre))

- c. Envoie de demande pour la réservation

Bouton (Réserver)

A ce moment :

Une requête est envoyée au Back end

Le trajet passe en Statut en Attente

Cas 2 — Réception de la Demande par les Chauffeurs les plus Proches

Le système interroge :

La base des chauffeurs en ligne.

Leurs Positions GPS

Leurs Statut (Disponible ou Occupé)

- a. Réception des Notifications

Notification vient (« nouvelle course disponible ») : qui contient

Point de Départ

Point D'Arrivée

Prix Estimé + Distance + Nom du Passager

b. Visualisation par les Chauffeurs via leurs écrans

Le Trajet Demander

Distance depuis Eux, Distance du Trajet

Prix Estimé

Bouton (Accepter ou Refuser)

Cas 3 : Validation de la Demande par le Chauffeur

Le Premier chauffeur qui Clique « Accepter »

Dévié le Chauffeur Officiel du Trajet

Les Autres reçoivent une Notifications « Course déjà acceptée »

Le Trajet passe en Statut « Accepté »

Le Système enregistre : Le Chauffeur, Heure d'Acceptation, Position au moment de l'acceptation

Cas 4 : Retour de la Demande de Notification

Le Passager reçoit une notification :

Votre Chauffeur arrive avec : nom + photos de chauffeur + modèle et plaque de véhicule + temps estimé d'arrivée, numéro de téléphone (pour le contact)

Il peut suivre le Chauffeur en temps sur la Carte

Cas 5 : Déplacement du Chauffeur vers le Passager

Le système notifie le passager

Le chauffeur suit le passager sur la carte – Pareil pour le Passager

Puis lui Contacte une fois arriver – Pareil aussi pour le Passager

Cas 6 : Début de la Course

- a. Une fois le passager a bord :
 - Le chauffeur clique sur « Démarrer »
 - Automatiquement le GPS est activé pour les deux
 - Le Trajet passe en Statut « En Attente »
- b. Pendant la course le passager peut voir la progression
 - Le chauffeur suit le GPS intégré
 - En cas d'urgence les deux peuvent signaler

Cas 7 : Fin de la Course

- a. Quand le chauffeur arrive à la destination
 - Il clique sur « Terminer »
- b. Le système :
 - Arrête le Suivi GPS
 - Enregistre la distance finale parcourue
 - Passe le Trajet en Statut « Terminé »

Cas 8 : Paiements

- a. Mode Cash
 - Le Passager remet le montant au Chauffeur.
 - Le Chauffeur Confirme sur l'Application : « Paiement Reçu »
 - Puis remet une commission à l'Administrateur
- b. Mode Electronique (OM, MM)
 - Le Passager choisit le mode de paiement
 - La demande de Paiement est envoyée
 - L'opérateur Confirme par Webhook
 - Le Trajet passe à « Payé »
- c. Toutes les Transactions sont visibles dans :
 - Historique Passager
 - Revenus Chauffeur
 - Dashboard Admin

Cas 9: Notations – Avis – Evaluation

Une fois le Trajet terminé

- a. Le Passager note le Chauffeur
 - 5 Etoiles (*****)
 - Commentaires
 - Qualité de véhicules
 - Sécurité
- b. Le Chauffeur peut aussi noter le Passager
 - Problème avec le Passager
- c. Visibilité Admin

Toutes les notes sont visibles chez l'Administrateurs

Résumer du Processus

Le passager entre une destination

Le système calcule prix + distance + chauffeurs proches

Il appuie sur "Réserver"

Tous les chauffeurs proches reçoivent la demande

Le premier chauffeur qui accepte devient officiel

Le passager est notifié et voit le chauffeur arriver

La course démarre

La course se termine

Paiement (cash, mobile money, wallet)

Notation chauffeur/passager

ÉTAPE 3 — LES ENTITÉS/MODÈLES DE LA V1

Après analyse, voici les objets réels du système.

● 1. User

Représente un utilisateur général : passager, chauffeur ou admin.

Champs : id, nom, prénom, téléphone, genre, email, mpasse, role, photo, date_creation, status (actif, inactif).

● 2. ChauffeurProfile (Chauffeur)

Profil dédié au chauffeur.

Champs :

id, user_id, numéro_permis, nationalité, statut_validation (en attente, validé, réjété) , disponibilité (online / offline / occupé), date_validation.

● 3. Véhicule

Champs :

id, chauffeur_id, marque, modèle, type (moto, taxi, voiture_privé), couleur, immatriculation, capacité, photos, status (actif, suspendu).

● 4. Trip (Trajet - Course)

Représente une course c'est le cœur du système.

Champs :

id,passager_id, chauffeur_id, depart (latitude_depart, longitude_depart), arrivee (latitude_arrivee, longitude_arrivee), distance, durée_estime, prix_estime, prix_final, statut (pending, accepted, driver_arrived, in_progress, completed, paid), date_demande, date_acceptation (heure_dacceptation), date_debut, date_fin, mode_paiement(cash, OM, MM), motif_annuler(si la course est annulé)

● 5. Payment

Champs : id, trajet_id, montant, mode_paiement(cash, OM_MM, wallet), statut (pending, payé, nonPayé), date_paiement, référence_transaction

● 6. Notification

Champs : id, user_id, type (trajet, paiement, système), message (contenu), statut (envoyer, lu), date.

● 7. Note (Evaluation)

Champs :

id, trajet_id, auteur_id, destinataire_id, note (5 étoile *****), commentaire, date.

● 8. Géolocalisation

Champs :

Id, user_id, role (passager / chauffeur), latitude, longitude, etat_reseau (online, offline), timestamp, trajet_id

NB : Les positions GPS sont transmises en temps réel via Firebase/WebSocket pour la navigation et le suivi sur l'application, mais toutes ne sont pas stockées en base.

Seules les positions clés (début, fin, éventuellement points intermédiaires pour analyses) sont conservées dans la table.

● 9. Historiques (facultatif pour un début de V1)

Champs :

id_hist, id_users, actions (connexion, déconnexion, modif_profil, annulation_trajet, urgence), date_heure, détails.

NB : Cette table ne duplique pas les données de Trajet ou Paiement.

Elle sert uniquement à la traçabilité des actions système et utilisateur, offrant une vue globale pour audit et reporting.

● 10. Paramètres

Champs :

id, cle (distance, pourcentage_comission, prix), valeur, description, date_dernières_modif, id_Admin

● 11. Rapports (Bilan)

Champs :

(id_rap, type, contenu, date_génération)

● 11. Litiges (Problèmes)

Champs :

id_lit, id_users, description, type, status (régler, nonRégler), date_ouverture, date_résolution, id_trajet, id_paiement.

● 12. DocumentsChauffeurs

Champs :

idt, id_chauffeur, type_doc (permis, assurance, carte_grise, contrôle_technique), url, status (valide, nonValide), date_upload, date_validation

● 13. Réservations (Association Porteuse)

Champs :

idt, id_passager, id_trajet, date_heure_reservation, status (en attente, valide, expiré)

● 14. Commisssions

Champs :

idcommission, id_paiement, idTarifCommision, montant_com, montant_chauffeur (revenu chauffeur), id_chauffeur, status (en attente, payé, nonpayé)

● 15. TarifCommission

Champs :

idTarif, taux, date_enregistrement

Etapes 4 - Analyse du Cahier de Charges V1 (Rélation)

● 1. Relation : User ↔ ChauffeurProfile (Chauffeur)

Un utilisateur peut être un simple passager ou un chauffeur.

S'il est chauffeur, il possède un profil chauffeur.

Relation textuelle :

Un utilisateur peut avoir zéro ou un profil chauffeur.

Un profil chauffeur appartient obligatoirement à un utilisateur.

Cardinalité :

User (1) — (0,1) ChauffeurProfile

● 2. Relation : ChauffeurProfile ↔ Véhicule

Chaque chauffeur possède exactement un véhicule dans la version V1.

Chaque véhicule appartient à un seul chauffeur.

Relation textuelle :

Un chauffeur possède un seul véhicule.

Un véhicule appartient à un seul chauffeur.

Cardinalité :

ChauffeurProfile (1) — (1, 1) Véhicule

● 3. Relation : User (passager) ↔ Trip (Course – Trajet)

Un passager peut effectuer plusieurs trajets.

Chaque trajet peut couvrir un ou plusieurs passagers.

Relation textuelle :

Un passager peut avoir plusieurs trajets.

Un trajet est effectué par un seul passager.

Cardinalité :

User (1) — (0, N) Trajet (en tant que passager)

● 4. Relation : ChauffeurProfile ↔ Trajet

Un chauffeur peut effectuer plusieurs trajets.

Chaque trajet est assuré par un seul chauffeur une fois acceptée.

Relation textuelle :

Un chauffeur effectue plusieurs trajets.

Un trajet est pris en charge par un seul chauffeur.

Cardinalité :

ChauffeurProfile (1) — (0, N) Trajet

● 5. Relation : Trajet ↔ Paiement

Chaque trajet possède un paiement associé.

Un paiement correspond à un trajet unique.

Relation textuelle :

Un trajet génère un seul ou plusieurs paiements.

Un paiement est lié à un seul trajet.

Cardinalité :

Trajet (1) — (0, N) Paiement

● 6. Relation : User ↔ Notification

Un utilisateur peut recevoir plusieurs notifications.

Une notification appartient à un seul utilisateur.

Relation textuelle :

Un utilisateur reçoit plusieurs notifications.

Une notification est destinée à un seul utilisateur.

Cardinalité :

User (1) — (0, N) Notification

● **7. Relation** : Trajet \leftrightarrow Rating (Note – Evaluation)

Après un trajet, le passager peut laisser une note.

Un trajet peut avoir une ou zéro note.

Relation textuelle :

Un trajet peut recevoir zéro ou une note.

Une note appartient à un seul trajet.

Cardinalité :

Trip (1) — (0,1) Note

● **8. Relation** : User \leftrightarrow Note

Un utilisateur peut écrire plusieurs notes (en tant que passager).

Une note est rédigée par un seul utilisateur.

Relation textuelle :

Un utilisateur peut laisser plusieurs avis.

Chaque avis appartient à un seul utilisateur.

Cardinalité :

User (1) — (0, N) Notes

● **9. Relation** : User (chauffeur) \leftrightarrow Note (note reçue)

Un chauffeur peut recevoir plusieurs notes via les trajets.

Une note a un seul destinataire.

Relation textuelle :

Un chauffeur peut recevoir plusieurs notes.

Une note est destinée à un seul chauffeur.

Cardinalité :

ChauffeurProfile (1) — (0, N) Notes

● **Relation 10 : Paiement ↔ Commission**

Chaque paiement génère une commission pour Taka Taka.

Une commission correspond toujours à un seul paiement.

Relation textuelle :

Un paiement peut avoir zéro ou une commission (car certains paiements cash n'auront pas de commission dans certains cas particuliers).

Toute commission doit obligatoirement venir d'un paiement.

Cardinalité

Commission (1) — (1) Paiement

● **Relation 11 : Commission ↔ TarifCommission**

Chaque commission utilise un tarif (pourcentage) défini dans la table TarifCommission.

Un tarif peut être utilisé pour plusieurs commissions.

Rélation Textuelle :

Un tarif peut être appliqué à plusieurs commissions.

Une commission utilise un seul tarif.

Cardinalité :

TarifCommission (1) — (0,N) Commission

● Relation 12 : Litige ↔ User

Un litige est créé par un seul utilisateur (passager ou chauffeur).

Un utilisateur peut avoir plusieurs litiges ouverts ou historiques.

Rélation Textuelle :

Chaque litige appartient à un seul utilisateur (celui qui déclare le problème).

Un utilisateur peut ouvrir plusieurs litiges.

Cardinalité

User (1) — (0,N) Litige

Litige (1) — (1) User

● Relation 13 : Litige ↔ Trip (Trajet)

Un litige est forcément lié à un trajet (ex : refus de paiement, fraude, retard, insultes...).

Un trajet peut avoir zéro, un ou plusieurs litiges.

Rélation

Un trajet peut générer plusieurs litiges.

Chaque litige se rapporte à un seul trajet.

Cardinalité

Trajet (1) — (0,N) Litige

Litige (1) — (1) Trajet

● Relation 14 — Paiement ↔ Litiges (Facultatif)

Description fonctionnelle

Un litige peut être lié à un paiement lorsqu'il y a un problème :

- Chauffeur qui garde l'argent
- Passager qui refuse de payer
- Erreur de montant
- Paiement OM/MM non confirmé
- Annulation après paiement
- Fraude détectée

Donc :

Un paiement peut générer plusieurs litiges

Mais chaque litige ne concerne qu'un seul paiement

Relation textuelle

Un paiement peut avoir zéro, un ou plusieurs litiges.
(ex : paiement frauduleux, contestation du montant, double paiement)

Un litige appartient obligatoirement à un seul paiement.

Cardinalité

Paiement (1) — (0, N) Litiges

Litige (1) — (1) Paiement

● Relation 15 : Paramètres ↔ User (Admin)

Rélation Textuelle :

Les paramètres du système (tarifs, commissions, distance minimale, configuration générale)

ne peuvent être modifiés que par un administrateur.

À chaque modification d'un paramètre, on enregistre quel admin a effectué la modification.

Cardinalité

User (Admin) (1) — (0, N) Paramètres (modifications)

Paramètres (1) — (1) Admin

« Le module Paramètres est relié à l'administrateur, car seul un admin peut modifier les paramètres système. Chaque modification est traçable (id_admin, date_modif). »

● Relation 16 : Rapports ↔ User (Admin)

Rélation Textuelle :

Les rapports (bilan journalier, hebdomadaire, mensuel, statistiques...) sont générés par :

L'administration manuellement (ex : bouton "Générer rapport")

ou automatiquement par le système (CRON)

Donc un rapport peut avoir :

Soit un admin créateur

Soit "SYSTEME" comme créateur

Cardinalité

User (Admin) (1) — (0, N) Rapports

Rapport (1) — (0,1) Admin
(puisqu'il peut être généré automatiquement)

Etapas 4 – Conception et Modélisation (UML)

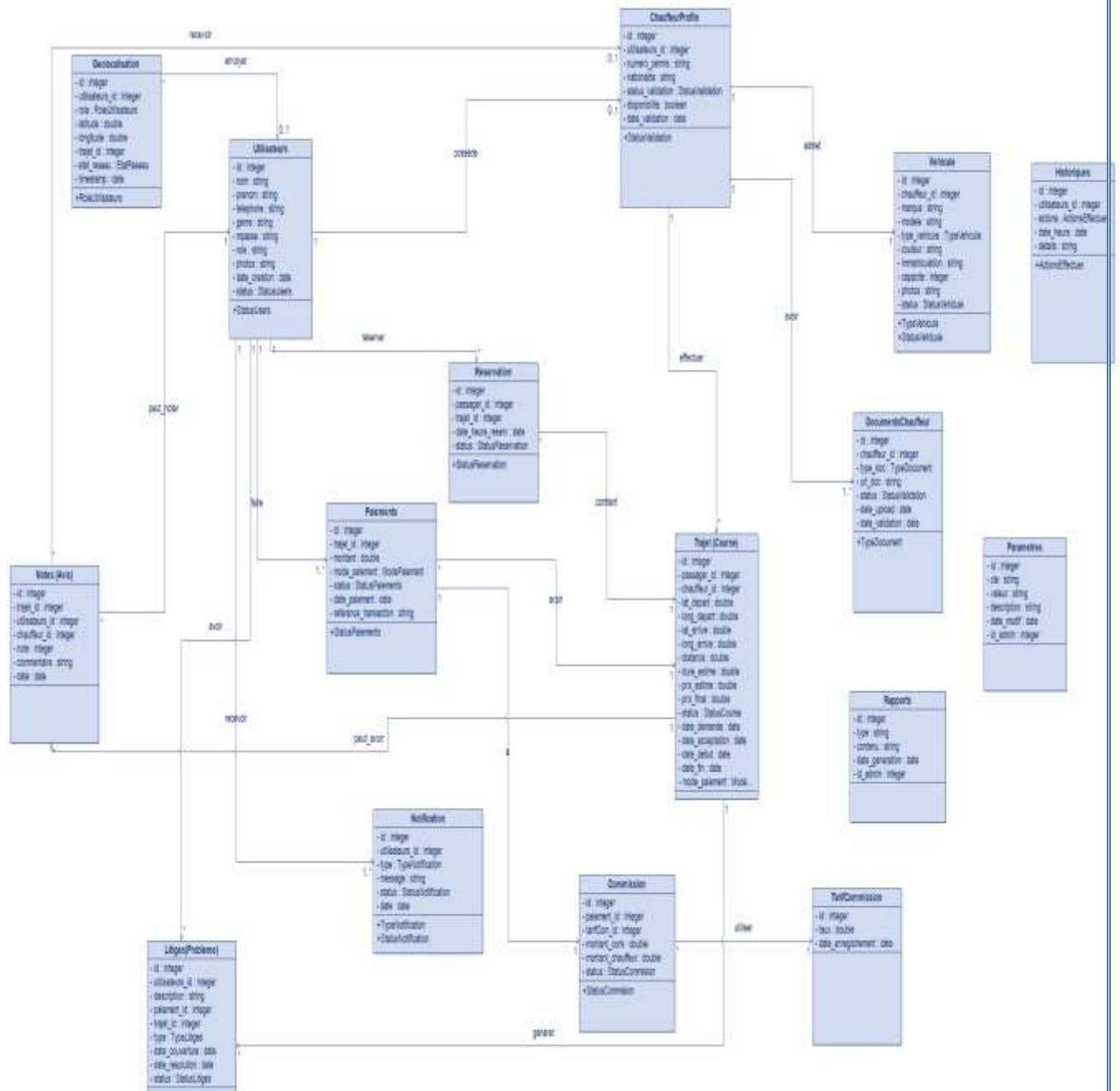
1 – Diagramme de Cas d'Utilisation

2 – Diagramme de Communication

3 – Diagramme de Séquence

4 – Diagramme d'Activité

5 – Diagramme de Classe



Recommandations :

- ❖ Multilingue : partiellement implémenté dans le MVP, complet dans les versions suivantes.
- ❖ USSD/SMS : prévu pour V2/V3, pas bloquant pour le lancement initial.

Fonctionnalités MVP Proposés par Entreprise EINEx

8.1 Version 1 : Lancement Initial (MVP - Minimum Viable Product)

- **Objectif :**

- Lancer une application fonctionnelle avec les fonctionnalités essentielles pour répondre aux besoins de base des utilisateurs et tester le marché.

- **Fonctionnalités prioritaires :**

1. Réservation de trajets en temps réel et planifiée.
2. Suivi en temps réel des véhicules via GPS.
3. Options de transport : moto-taxis, taxis partagés, voitures privées.
4. Paiements flexibles : espèces, Mobile Money, portefeuille électronique.
5. Vérification des chauffeurs et des véhicules.
6. Bouton d'urgence pour signaler des incidents.
7. Interface multilingue (français, langues locales).
8. Mode USSD/SMS pour les utilisateurs sans smartphones.