**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
**Université M'hamed Bougara - Boumerdès**




**Faculté des Sciences Département d'Informatique**
**Domaine :** Mathématiques Informatique
**Filière :** Informatique
**Spécialité :** Ingénierie des systèmes d'information et du logiciel

**Mémoire de fin d'études en vu de l'obtention du Diplôme de Licence Académique**

# Thème

Application web et mobile pour la gestion des absences et le suivi de transport scolaire

## Présenté par :

- **CHERABI Zineddine**
- **BETTAYEB Abdelkader**
- **NEGGACHE Mohammed amine**
- **SEBAA Abderrahim**

# Remerciements

**Nous tenons tout d'abord à remercier Dieu, le Tout-Puissant et Miséricordieux, qui nous a donné la force, la volonté et la patience pour accomplir ce modeste travail.**

**Nous adressons nos sincères remerciements à notre encadreur, M. Faycel Abbas, pour son accompagnement, ses conseils précieux, sa disponibilité ainsi que son soutien tout au long de la réalisation de ce projet.**

**Nous exprimons également notre profonde gratitude à Mme Sabba-Chehbour Fairouz, examinatrice de ce travail, pour l'honneur qu'elle nous fait en acceptant d'évaluer notre mémoire et pour l'intérêt porté à notre recherche.**

**Nos remerciements s'adressent aussi à tous nos enseignants qui nous ont accompagnés durant notre parcours universitaire et qui ont contribué à notre formation.**

**Enfin, nous remercions chaleureusement toutes les personnes ayant participé, de près ou de loin, à la réalisation de ce travail.**

# Dédicace

**Nous dédions ce modeste travail à nos chers parents, pour leur amour inconditionnel, leurs sacrifices, leur soutien moral et matériel, ainsi que leurs précieux conseils tout au long de notre parcours**

**À nos familles, à nos amis, à nos camarades de promotion, ainsi qu'à toutes les personnes qui nous ont encouragés et soutenus de près ou de loin pour la réalisation de ce travail**

**Je dédie ce travail à ma très chère mère MIDOU Hayat, source d'amour, de tendresse et de courage, qui a toujours cru en moi et m'a soutenu dans les moments difficiles**

**À mon père BETTAYEB Kamal, qui a toujours été un exemple de sacrifice et de persévérance, et grâce à qui je suis arrivé à ce niveau**

**BETTAYEB Abdelkader**

**Je dédie ce travail à mes très chers parents BECHIM Akila et CHERABI Ali, pour leur amour, leur patience et leur soutien inestimable tout au long de mon parcours**

**À ma famille, mes amis et tous ceux qui m'ont encouragé et accompagné jusqu'à la réalisation de ce travail**

**CHERABI Zinedine**

# Table des matières

Introduction générale ..... 3
Problématique : ..... 3
Objectif du projet : ..... 3
Organisation du mémoire : ..... 4
Chapitre 1 : Étude de l'existant ..... 5
I.1 Introduction : ..... 5
I.2 Présentation du domaine : ..... 5
I.3 Étude de l'existant : ..... 5
I.3.1 Gestion des absences ..... 5
I.3.2 Suivi du transport scolaire ..... 6
I.4 Critique de l'existant : ..... 6
I.4.1 Problèmes liés à la gestion des absences ..... 6
I.4.2 Problèmes liés au transport scolaire ..... 6
I.5 Solution proposée: ..... 6
I.5.1 Automatisation et centralisation de la gestion des absences ..... 6
I.5.2 Suivi intelligent du transport scolaire : ..... 7
I.5.3 Architecture et choix techniques ..... 7
Conclusion du chapitre ..... 7
Chapitre 2 : Analyse et conception du système ..... 8
II.1Introduction. ..... 8
II.2 Identification des acteurs ..... 8
II.2.1 L'Administrateur ..... 8
II.2.2 L'Enseignant : ..... 8
II.2.3Le Parent ..... 8
II.2.4 Le Chauffeur ..... 9
II.3 Besoins fonctionnels ..... 9
II.3.1 Gestion des comptes et des accès ..... 9
II.3.2 Gestion des élèves et des classes ..... 9
II.3.3 Gestion des absences ..... 9
II.3.4Suivi du transport scolaire ..... 10
II.4 Besoins non fonctionnels ..... 10
II.5 Diagramme de cas d'utilisation ..... 10
II.5.1 Description du diagramme et de ses relation***s*** ..... 12

II.6 Diagramme de Classes ..... 12
II.6.1DICTIONNAIRE DE DONNÉES: ..... 12
II.7 Passage au schéma relationnelPassage du diagramme de classes au schéma relationnel ..... 15
II.8. Diagrammes de séquence ..... 15
II.8 .2 Description du scénario : ..... 15
II.8.3 Diagramme de séquence du cas d'utilisation "Suivi du bus (Côté Parent)" ..... 16
Chapitre 3 : Réalisation et Implémentation ..... 18
III.1 Introduction ..... 18
III.2 Environnement de développement ..... 18
III.2.1 Environnement matériel ..... 18
III.2.2 Environnement logiciel ..... 18
III.3 Outils et Technologies utilisés ..... 19
III.3.1 Langages de programmation ..... 19
III.3.2 Frameworks et Bibliothèques ..... 20
III.3.3 Sécurité et Utilitaires : ..... 21
III.4.2 Logique de l'application ..... 23
III.4.3 Schéma relationnel ..... 23
III.5 Présentation des interfaces ..... 23
III.5.2 Interface Administrateur — Gestion des enseignants ..... 25
III.5.3 Interface Parent — Suivi des absences ..... 26
III.5.4 Interface Parent — Suivi du bus en temps réel ..... 26
III.5.5 Interface Enseignant — Emploi du temps ..... 27
III.5.6 Interface Enseignant — Prise d'absence ..... 28
III.5.7 Interface Chauffeur — Tableau de bord ..... 29
Conclusion du chapitre ..... 31
Conclusion générale ..... 31

# Introduction générale

L'évolution des technologies de l'information a profondément transformé plusieurs secteurs, notamment celui de l'éducation. Aujourd'hui, les établissements scolaires s'appuient de plus en plus sur des systèmes informatiques afin d'améliorer la gestion administrative et faciliter la communication entre les différents acteurs (administration, enseignants et parents).

Cependant, certaines tâches comme la gestion des absences des élèves et le suivi du transport scolaire restent souvent réalisées de manière manuelle ou traditionnelle. Cette situation peut entraîner des erreurs, une perte de temps et un manque de visibilité pour les parents concernant le trajet scolaire de leurs enfants.

## Problématique :

La gestion manuelle des absences des élèves et du transport scolaire présente plusieurs insuffisances. En effet, l'enregistrement et l'organisation des données peuvent devenir complexes et sources d'erreurs. De plus, l'accès aux informations n'est pas toujours rapide ni pratique pour les différents acteurs de l'établissement. L'absence d'un système de suivi en temps réel du transport scolaire empêche les parents de connaître la position du bus, ce qui peut engendrer des retards et des inquiétudes concernant la sécurité des élèves. Ces limites montrent la nécessité d'adopter une solution informatique moderne et efficace.

## Objectif du projet :

L'objectif de ce travail est la conception et la réalisation d'une application web et mobile dédiée à la gestion des absences des élèves ainsi qu'au suivi du transport scolaire en temps réel. Cette application a pour finalité d'améliorer l'organisation administrative au sein des établissements scolaires, de faciliter la communication entre les enseignants, les parents et les chauffeurs de transport scolaire, et de garantir un accès rapide, fiable et sécurisé aux informations.

Pour assurer une bonne conception du système, nous avons utilisé le langage UML, qui permet de modéliser les différentes fonctionnalités de l'application à travers des diagrammes adaptés.

## Organisation du mémoire :

Notre mémoire est organisé en trois chapitres principaux :

Chapitre 1 : Étude de l'existant

Dans ce premier chapitre, nous commençons par présenter le domaine de l'éducation et l'importance de la numérisation des services scolaires. Nous décrivons ensuite le fonctionnement actuel de la gestion des absences et du transport scolaire au sein des établissements, en mettant en évidence les insuffisances du système manuel utilisé. Sur la base de cette analyse, nous proposons une solution informatique moderne et adaptée aux besoins identifiés.

Chapitre 2 : Analyse et conception du système

Le deuxième chapitre est consacré à l'analyse et à la modélisation du système. Nous y identifions les quatre acteurs principaux — l'administrateur, l'enseignant, le parent et le chauffeur — et définissons leurs besoins fonctionnels et non fonctionnels. Nous modélisons ensuite le système à l'aide du langage UML à travers les diagrammes de cas d'utilisation, de classes et de séquence, avant d'établir le schéma relationnel de la base de données.

Chapitre 3 : Réalisation et implémentation

Le troisième chapitre présente la phase de développement de l'application. Nous y décrivons l'environnement matériel et logiciel utilisé, ainsi que les technologies et frameworks adoptés tels que Flutter, Node.js, Express.js et MySQL. Nous exposons ensuite l'architecture générale du système, les mécanismes de sécurité mis en place, et nous présentons les différentes interfaces réalisées pour chaque acteur de l'application.

Enfin, nous clôturons ce mémoire par une conclusion générale qui récapitule les résultats obtenus et les compétences acquises tout au long de ce projet.

# Chapitre 1 : Étude de l'existant

## I.1 Introduction :

Dans ce chapitre, nous allons présenter l'étude de l'existant de notre système. Nous allons décrire le fonctionnement actuel de la gestion des absences des élèves et du transport scolaire, puis identifier les différents problèmes rencontrés.

Ensuite nous allons proposer une solution informatique adaptée

## I.2 Présentation du domaine :

Le domaine de l'éducation constitue un pilier fondamental dans toute société moderne. Avec l'évolution des technologies de l'information, les établissements scolaires sont appelés à adopter des solutions numériques afin d'améliorer l'organisation et la gestion de leurs services administratifs et pédagogiques. Parmi les aspects essentiels du fonctionnement d'un établissement scolaire figure la gestion des absences des élèves. Cette tâche ne se limite pas uniquement à enregistrer la présence ou l'absence, mais elle représente un élément clé du suivi pédagogique et de la discipline. Une gestion efficace des absences permet d'assurer une communication rapide entre l'administration et les parents, contribuant ainsi à la sécurité et à l'encadrement des élèves. D'autre part, le transport scolaire constitue un service logistique important garantissant le déplacement sécurisé des élèves entre leur domicile et l'établissement. Ce service nécessite une organisation rigoureuse, une planification précise des trajets ainsi qu'un suivi régulier afin de respecter les horaires et d'assurer la sécurité durant les déplacements. Ainsi, la gestion des absences et le suivi du transport scolaire représentent deux composantes essentielles du système éducatif. L'intégration d'une solution numérique centralisée permettrait d'optimiser ces services, d'automatiser le traitement des informations et d'améliorer la communication entre les différents acteurs de l'établissement scolaire.

## I.3 Étude de l'existant :

### I.3.1 Gestion des absences

Actuellement, la gestion des absences au sein d'établissement repose sur un système entièrement manuel et traditionnel. À chaque début de séance, l'enseignant procède à l'appel et note les élèves absents sur des fiches papier ou dans un registre de classe. Ces informations sont ensuite collectées physiquement et transmises au bureau de l'administration afin d'être archivées ou saisies ultérieurement. Ce circuit d'information s'avère relativement lent et peu efficace. Il empêche l'alerte des parents en temps réel et augmente le risque d'erreurs lors de la manipulation des documents papier. De plus, la recherche ou la consultation des absences nécessite un effort supplémentaire, ce qui complique le suivi régulier des élèves.

### I.3.2 Suivi du transport scolaire

Le service de transport scolaire fonctionne actuellement sans outil technologique de suivi en temps réel. Les parents ne disposent d'aucun moyen leur permettant de connaître la position exacte du bus ou l'heure précise de son arrivée. Dans la pratique, le chauffeur gère son itinéraire de manière autonome, ce qui peut entraîner des retards ou un non-respect des temps d'attente aux points de ramassage. L'absence de communication directe entre le conducteur, l'administration et les parents peut générer un sentiment d'insécurité et une certaine désorganisation quotidienne.

## I.4 Critique de l'existant :

Bien que le système actuel permette d'assurer un suivi minimal des absences et du transport scolaire, notre analyse met en évidence plusieurs insuffisances majeures qui limitent son efficacité, sa fiabilité et son niveau de sécurité

### I.4.1 Problèmes liés à la gestion des absences

**Perte de temps considérable** : L'enregistrement manuel des absences sur support papier mobilise un temps important durant les séances pédagogiques.

**Manque de fiabilité :** La saisie manuelle expose le système à un risque élevé d'erreurs humaines lors du calcul ou du transfert des informations

**Absence de communication instantanée :** Les parents ne sont pas informés en temps réel des absences ou retards de leurs enfants

**Fragilité de l'archivage** : Les documents physiques sont vulnérables à la perte, à la détérioration ou à un classement inefficace, rendant la consultation des historiques complexe.

### I.4.2 Problèmes liés au transport scolaire

**Absence de suivi en temps réel** : Le manque d'un système de géolocalisation empêche les parents et l'administration de connaître la position exacte des bus scolaires

**Difficulté de contrôle :** L'administration ne dispose pas d'outils efficaces pour vérifier le respect des itinéraires et des horaires.

**Sentiment d'insécurité :** L'absence de visibilité sur les trajets génère une inquiétude constante chez les parents.

## I.5 Solution proposée :

Afin de remédier aux insuffisances identifiées dans le système actuel, nous proposons la conception et le développement d'une solution informatique intégrée, composée d'une application mobile destinée aux parents et aux chauffeurs, ainsi que d'une interface web dédiée à l'administration et aux enseignants. Cette solution repose sur les axes suivants :

### I.5.1 Automatisation et centralisation de la gestion des absences

**Numérisation du processus :** Remplacement des registres papier par une interface numérique intuitive permettant aux enseignants d'enregistrer les absences en quelques secondes.

**Centralisation des données :** Toutes les informations seront stockées dans une base de données sécurisée (MySQL), garantissant un archivage fiable et une consultation rapide.

**Notifications instantanées :** Envoi automatique d'alertes aux parents dès la validation d'une absence ou d'un retard

### I.5.2 Suivi intelligent du transport scolaire :

**Géolocalisation en temps réel :** Intégration de la bibliothèque **Leaflet.js** permettant aux parents de visualiser la position exacte du bus sur une carte interactive. La partie Mobile communique avec des APIs développées avec Node.js et Express.js permettant de gérer les opérations liées à la géolocalisation, au suivi GPS des bus scolaires et à l'envoi des notifications.

**Contrôle des trajets et des horaires :** L'administration pourra superviser le respect des itinéraires et des points d'arrêt prédéfinis.

**Renforcement de la sécurité :** La transparence du suivi réduit l'incertitude des parents et améliore la ponctualité du service

### I.5.3 Architecture et choix techniques

**Frontend Mobile :** Développement avec le Framework Flutter afin d'assurer une compatibilité multiplateforme (Android et iOS)

**Backend :** Mise en place d'une API développée avec Node.js et Express.js assurant la communication sécurisée entre les applications (web et mobile) et la base de données. Cette architecture permet de traiter les requêtes HTTP, gérer les utilisateurs et assurer le suivi en temps réel du transport scolaire.

**Base de données :** Utilisation de MySQL pour garantir la fiabilité et la performance du stockage des données

En définitive, cette solution vise à moderniser la gestion administrative, à améliorer la communication entre l'établissement et les familles, et à garantir un environnement scolaire plus sécurisé et mieux organisé

## Conclusion du chapitre

Dans ce premier chapitre, nous avons présenté l'étude de l'existant relative à la gestion des absences des élèves et au suivi du transport scolaire. Cette analyse nous a permis d'identifier les différentes limites du système actuel, notamment le caractère manuel des procédures, le manque de communication en temps réel et l'absence d'un outil technologique adapté

Dans le chapitre suivant, nous allons identifier les acteurs du système et définir les besoins fonctionnels et non fonctionnels, puis modéliser le système à l'aide des diagrammes UML

# Chapitre 2 : Analyse et conception du système

## II.1Introduction

Dans ce chapitre, nous présentons l'analyse et la conception du système proposé pour la gestion des absences des élèves et le suivi du transport scolaire. Cette étape permet d'identifier les besoins du système ainsi que les interactions entre les différents acteurs. Pour modéliser le fonctionnement du système, nous avons utilisé le langage UML à travers différents diagrammes permettant de représenter les fonctionnalités, les interactions et la structure des données. Ce chapitre constitue une étape essentielle avant la phase de réalisation du système.

## II.2 Identification des acteurs

Dans le cadre de notre système de gestion des absences et de suivi du transport scolaire, quatre acteurs principaux interagissent avec l'application : l'Administrateur, l'Enseignant, le Parent et le Chauffeur.

### II.2.1 L'Administrateur

L'administrateur est responsable de la gestion globale du système. Il assure la création et la gestion des comptes utilisateurs (enseignants, parents et chauffeurs), l'enregistrement des élèves ainsi que leur affectation aux classes.

Il organise également les lignes de transport scolaire, attribue les bus aux élèves et supervise l'ensemble des données relatives aux absences et aux trajets. Il dispose d'un accès aux rapports et aux statistiques permettant d'assurer un contrôle efficace du système.

### II.2.2 L'Enseignant :

L'enseignant accède au système via une interface sécurisée. Grâce à l'emploi du temps préalablement enregistré, le système affiche automatiquement, selon le jour et l'heure, la classe concernée ainsi que la liste des élèves correspondants.

Il est chargé d'enregistrer la présence ou l'absence des élèves durant chaque séance. Les informations sont ensuite sauvegardées et mises à jour dans la base de données afin d'assurer un suivi pédagogique structuré.

### II.2.3Le Parent

Le parent utilise l'application mobile pour consulter les informations relatives à son ou ses enfants.

Dans le cas où le parent a plusieurs enfants inscrits dans l'établissement, le système lui offre la possibilité de sélectionner l'enfant concerné à travers une interface dédiée afin d'accéder aux informations spécifiques à chacun.

Dans la partie dédiée aux absences, il peut visualiser l'emploi du temps ainsi que le détail des séances où l'élève était présent ou absent.

Concernant le transport scolaire, il peut consulter le numéro du bus affecté à son enfant, vérifier si celui-ci est monté dans le bus et suivre la position du véhicule en temps réel. Il reçoit également une notification lorsque le bus approche du point d'arrêt.

### II.2.4 Le Chauffeur

Le chauffeur dispose d'une interface mobile affichant l'itinéraire à suivre ainsi que les différentes stations prévues. Pour chaque arrêt, la liste des élèves concernés est indiquée.

Lors de l'arrêt à une station, il confirme la montée des élèves dans le bus. Un temps d'attente prédéfini est respecté avant la reprise du trajet. Les informations sont mises à jour automatiquement afin d'assurer un suivi précis du transport scolaire

## II.3 Besoins fonctionnels

Les besoins fonctionnels décrivent les fonctionnalités que le système doit assurer afin de répondre aux exigences des utilisateurs. Ils sont classés comme suit

### II.3.1 Gestion des comptes et des accès

**F1 : Authentification sécurisée :**

Le système doit permettre à chaque utilisateur (Administrateur, Enseignant, Parent et Chauffeur) d'accéder à son espace personnel via un login et un mot de passe sécurisés

**F2 : Gestion des profils :**

L'administrateur doit pouvoir créer, modifier ou supprimer les comptes utilisateurs et gérer leurs droits d'accès

### II.3.2 Gestion des élèves et des classes

**F3 : Enregistrement des élèves**

Le système doit permettre l'enregistrement des informations des élèves et leur affectation aux classes correspondantes

**F4: Association Parent - Élève**

le système doit permettre d'établir la relation entre chaque élève et son parent Dans le cas où un parent possède plusieurs enfants, il doit pouvoir sélectionner l'enfant concerné pour consulter ses informations

### II.3.3 Gestion des absences

**F5: Affichage automatique des classes**

Le système doit permettre à l'enseignant de sélectionner le jour et la plage horaire afin d'afficher la classe et la liste des élèves correspondante**.**

**F6: Saisie des présences**

L'enseignant doit pouvoir marquer chaque élève comme présent ou absent en temps réel

**F7 : Consultation de l'historique des absences**

Le parent doit pouvoir consulter l'historique des absences de son enfant.

### II.3.4Suivi du transport scolaire

**F8 : Géolocalisation du bus en temps réel**

Le système doit permettre le suivi GPS du bus scolaire via via une carte interactive **(Leaflet.js)**

**F9 : Gestion du transport scolaire**

Le chauffeur doit pouvoir consulter l'itinéraire à suivre ainsi que la liste des élèves à chaque station

**F10 : Validation de la montée des élèves**

Le chauffeur doit confirmer si l'élève est monté dans le bus à chaque arrêt.

**F11 : Alertes de proximité**

Le système doit envoyer une notification au parent lorsque le bus s'approche du point d'arrêt

## II.4 Besoins non fonctionnels

Les besoins non fonctionnels définissent les contraintes techniques et les critères de qualité que le système doit respecter.

- **Sécurité :** assurer la protection des données et un accès sécurisé via authentification.
- **Performance :** garantir une rapidité d'exécution pour le suivi GPS et la gestion des absences.
- **Fiabilité :** assurer la stabilité du système et la cohérence des données.
- **Disponibilité :** permettre l'accès à l'application à tout moment.
- **Ergonomie :** proposer une interface simple et facile à utiliser.

## II.5 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un outil de modélisation qui permet de représenter les interactions entre les différents acteurs et le système, ainsi que les fonctionnalités offertes par celui-ci. Il offre une vue globale du fonctionnement du système et facilite la compréhension des services proposés [1].

Le diagramme de cas d'utilisation ci-dessous présente de manière synthétique les interactions entre les différents acteurs et les principales fonctionnalités du système. Il permet également de visualiser l'organisation des services ainsi que les relations entre les cas d'utilisation, notamment les relations « include » et « extend » (Booch et al., 2005).

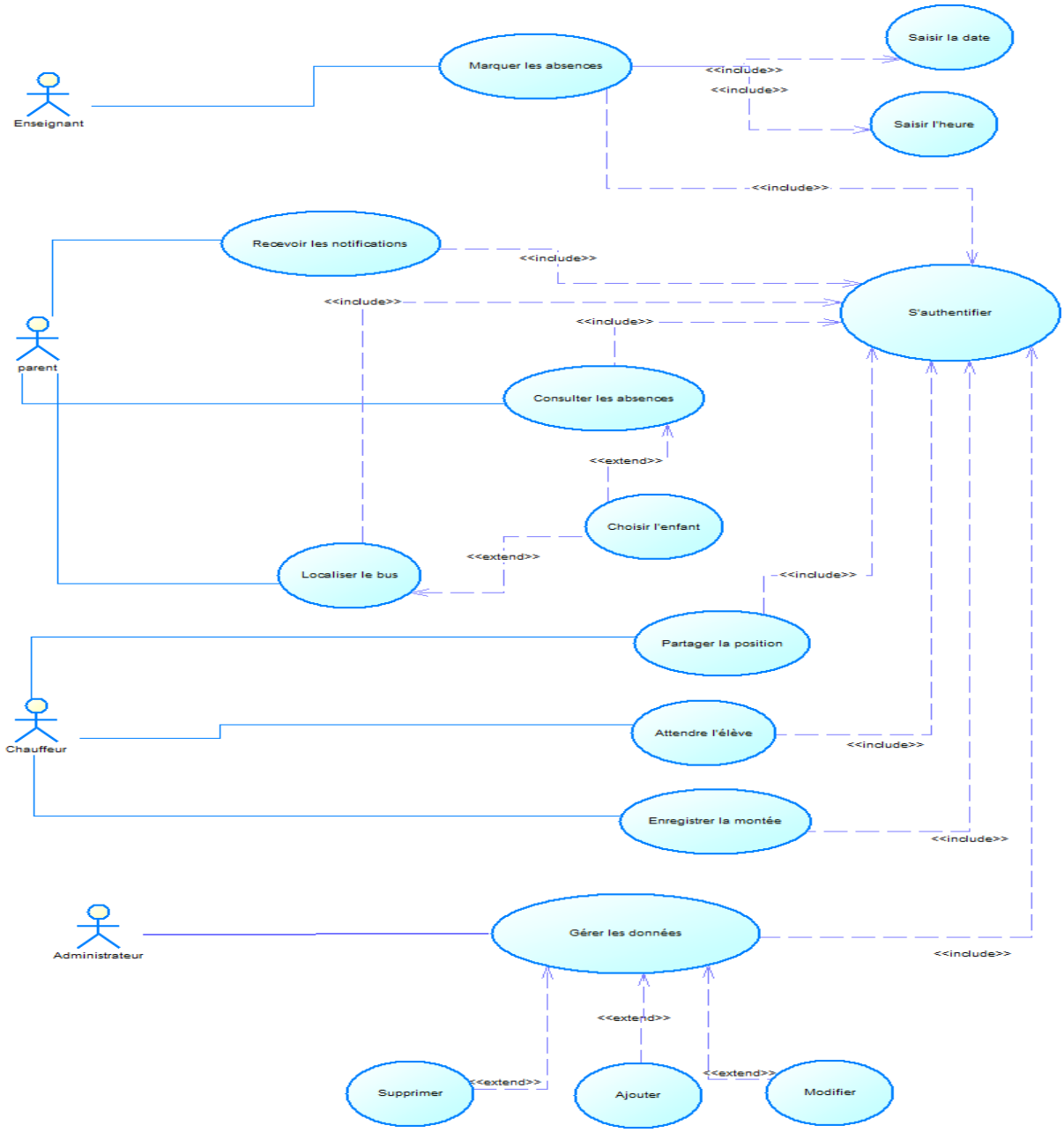



**Figure II.1.1 : Diagramme de cas d'utilisation du système**

### II.5.1 Description du diagramme et de ses relation*s*

Ce diagramme synthétise les interactions entre les quatre acteurs et les fonctionnalités clés du système. L'organisation des services repose sur les processus suivants :

- **L'Administrateur :** Gère globalement les données (ajouter, modifier et supprimer).
- **L'Enseignant :** Marque les absences en saisissant la date et l'heure.
- **Le Parent :** Consulte les absences et suit le bus. Le cas **« Choisir l'enfant »** est une extension activée en cas de fratrie.
- **Le Chauffeur :** Partage sa position GPS et confirme la montée des élèves.

**Analyse technique des relations :**

Selon les standards du langage UML **[2]**, le diagramme met en évidence deux types de relations fondamentales **[3]** :

- **La relation d'inclusion (« include ») :** Elle représente une étape obligatoire. Dans notre cas, l'authentification est incluse dans chaque service pour sécuriser l'accès.
- **La relation d'extension (« extend ») :** Elle représente des fonctionnalités optionnelles, comme la sélection d'un enfant spécifique ou la gestion des données.

## II.6 Diagramme de Classes

Le diagramme de classes constitue la vue statique du système en représentant sa structure interne. Il permet de modéliser les classes, leurs attributs ainsi que les associations qui les lient, **indépendamment du langage de programmation [4]**. La figure suivante présente la structure de données de notre application :

### II.6.1DICTIONNAIRE DE DONNÉES:

La réalisation du diagramme de classe se base sur le dictionnaire de données et les règles de gestion. L'analyse sémantique des données du dictionnaire permet de les regrou-per dans des entités à part. Les liens qui les relient tiennent compte des règles de gestion.

A =Alphabétique, N= Numérique AN =Alpha Numérique D=Date

| Nom de classe | codification | libllé | type | longueur |
|---|---|---|---|---|
| Parent | idParent | L'identifiant du parent | N | 10 |
| | nomComplet | Nom complet du parent | A | 50 |
| | motDePasse | Mot de passe du parent | AN | 30 |
| | email | Adresse email du parent | AN | 40 |
| Élève | idEleve | L'identifiant de l'étudiant | N | 10 |
| | nomComplet | Nom complet de l'étudiant | A | 50 |
| | class | la classe de l étudiant | N | 10 |
| | dansBus | État de présence dans le bus | AN | 5 |
| Chauffeur | idChauffeur | L'identifiant du chauffeur | N | 10 |
| | bus | Identifiant du bus assigné | AN | 20 |
| | position | Position géographique actuelle (GPS) | AN | 50 |
| | motDePasse | Mot de passe du chauffeur | AN | 30 |
| | nomComplet | Nom complet du chauffeur | A | 50 |
| Station | idStation | L'identifiant de la station | N | 10 |
| | position | Position géographique | AN | 50 |
| | nomStation | Nom de la station | A | 30 |
| | attente | État d'attente à la station | A | 10 |
| Notification | idNotification | L'identifiant de la notification | N | 10 |
| | idParent | L'identifiant du parent concerné | N | 10 |
| | nomEleve | Nom complet de l'étudiant concerné | A | 15 |
| | titre | Titre ou désignation de la notification | A | 15 |
| | message | Le texte ou contenu de l'id_notification | AN | 50 |
| | type | Le type d'alerte | A | 10 |
| | estLu | Etat de lecture de la notification | A | 10 |
| Trajet | idTrajet | L'identifiant de la voyage | N | 10 |
| | nomTrajet | Nom ou désignation du trajet | A | 30 |
| | heureDepart | Heure de début du voyage | D | 8 |
| Enseignant | idEnseignant | L'identifiant de l'enseignant | N | 10 |
| | motDePasse | Mot de passe de l'enseignant | AN | 30 |
| | nomComplet | Nom complet del'enseignant | A | 50 |
| Emploi du temps | idEmploi | L'identifiant de la séance | N | 10 |
| | classe | Nom de la classe concernée | AN | 20 |
| | jour | Date de la séance | D | 10 |
| | heure | L'heure de la séance | D | 8 |
| Absence | idAbsence | L'identifiant de l'absence | N | 10 |
| | jour | La date de l'absence | D | 10 |
| | heure | L'heure de l'absence | D | 8 |
| | statut | L'état de l'absence (Justifié ou non) | A | 10 |

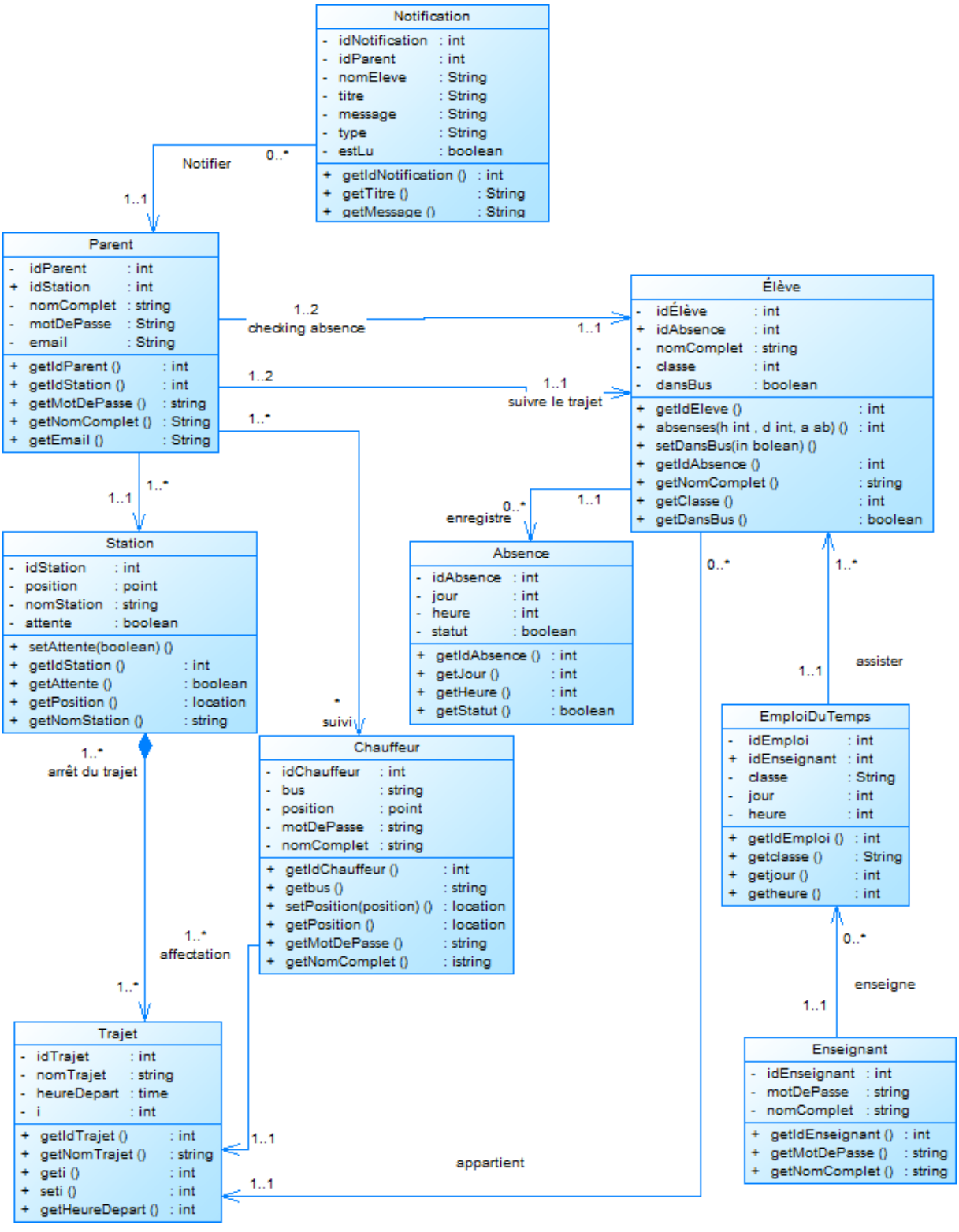



**Figure II.2.1 Diagramme de classe générale**

## II.7 Passage au schéma relationnel Passage du diagramme de classes au schéma relationnel

Le passage du diagramme de classes UML vers le modèle relationnel consiste à transformer les différentes classes en tables relationnelles. Les attributs deviennent des champs de la table, tandis que les identifiants sont définis comme clés primaires. Les relations entre les classes sont représentées à travers des clés étrangères selon les cardinalités définies [5].

Après application de ces règles, nous avons obtenu le schéma relationnel suivant :

Parent (#id_parent, nomComplet, motDePasse, email, #id_station)

Station (#id_station, nomStation, position, attente)

Élève (#id_eleve, nomComplet, classe, dansBus, #id_parent)

Chauffeur (#id_chauffeur, nomComplet, bus, position, motDePasse)

Absence (#id_absence, jour, heure, statut, #id_eleve)

Enseignant (#id_enseignant, nomComplet, motDePasse)

Emploi_du_temps (#id_emploi, classe, date, heure, #id_enseignant)

Trajet (#id_trajet, nomTrajet, heureDepart, indice)
Notification (#id_notification, titre, message, type, estLu, #id_parent)

## II.8. Diagrammes de séquence

### II.8.1 Introduction :

Ce diagramme de séquence illustre le processus d'interaction entre l'enseignant et le système afin de récupérer la liste des élèves selon l'heure et le jour de la séance. Il met en évidence les échanges entre les différents composants du système, notamment l'interface utilisateur, la base de données et les objets impliqués dans le traitement des informations.

### II.8 .2 Description du scénario :

Le scénario débute lorsque l'enseignant saisit l'heure et le jour de la séance dans l'interface du système. Une requête **getClasse(h, j)** est envoyée à la base de données afin de récupérer la classe correspondante. Une fois la classe identifiée, le système exécute la requête **getEleve(classe, h, j)** pour obtenir la liste des élèves concernés. Enfin, les informations récupérées sont affichées à l'enseignant afin de lui permettre de consulter ou d'enregistrer les absences.

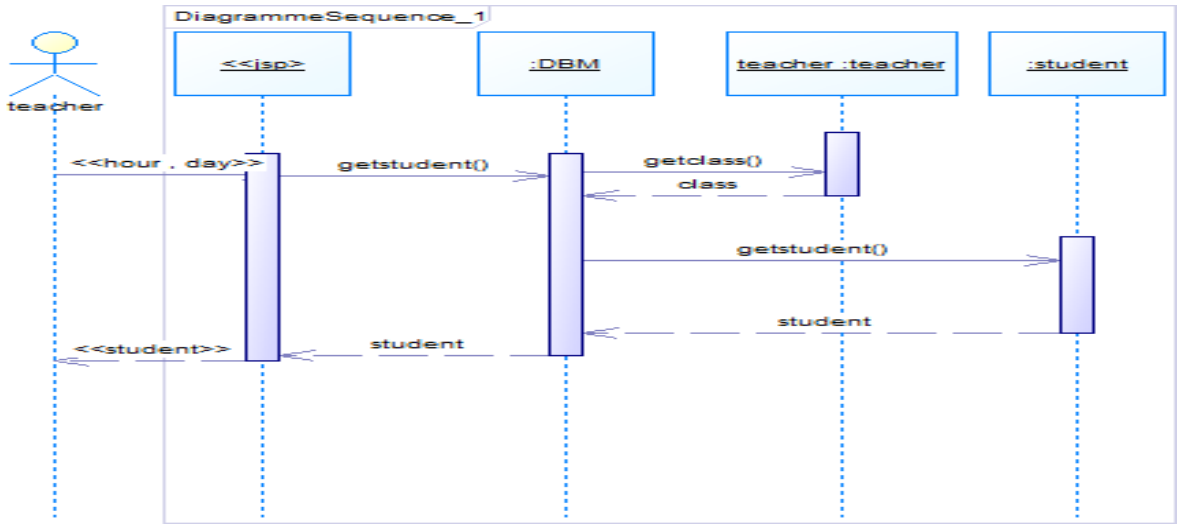



***Figure II 3.1: Diagramme de séquence d'affichage des étudiants***

## II.8.3 Diagramme de séquence du cas d'utilisation "Suivi du bus (Côté Parent)"

Le diagramme de séquence présenté décrit les interactions dynamiques entre l'acteur Parent, le serveur Express et la base de données MySQL.

Le processus commence par l'initialisation de l'interface, où le parent envoie la requête **GET /parenttrip.html** pour charger la page, et reçoit en retour la **Réponse (HTML, CSS, JS)**. Dès que la page s'affiche, le navigateur demande le profil du parent via **GET /api/parent/profile**. Le serveur Express va alors chercher les informations dans la base de données avec la commande **SELECT parent, enfants, stations**. Une fois les **Données récupérées (MySQL)**, le serveur les renvoie au parent sous forme d'une **Réponse JSON (Nom, Tableau d'enfants)**, ce qui permet au navigateur de faire l'action interne **Affichage du nom et de la liste des enfants** sur l'écran.

Ensuite, le système entre dans une boucle de suivi automatique (**loop**), qui se déclenche **[Chaque 15s OU Changement d'enfant]**. Dans cette boucle, le navigateur demande l'état du bus via **GET /api/parent/bus-status?studentId=X**. Le serveur exécute d'abord la sécurité avec **requireParent()**, puis cherche les détails du transport avec **SELECT étudiant, station, trajet, chauffeur**. Il utilise ensuite la fonction interne **getTrajetActuel(idChauffeur)** pour vérifier la course, et demande la position du bus via **SELECT position_chauffeur, état_trajet**. Après avoir reçu la **Position actuelle et état du trajet**, le serveur fait le **Calcul de distance et vérification du passage des stations** et envoie le résultat final sous forme d'une **Réponse JSON (Temps chauffeur, Position, État enfant, Distance)**. Enfin, le navigateur fait une **Mise à jour (Cards, Distance, État enfant)** pour changer les textes et exécute l'action **Redessiner la carte : Itinéraire (Leaflet/OSRM)** afin de visualiser le déplacement du bus en temps réel sur la carte.

Enfin, comme l'indique la note du diagramme, le système repose sur un mécanisme asynchrone de mise à jour de la position du bus. L'application du chauffeur envoie périodiquement ses coordonnées au serveur via la route **POST /api/driver/location**. Le serveur Express met ensuite à jour les informations dans la base de données MySQL, ce qui garantit au parent l'accès à des données de géolocalisation continuellement actualisées.

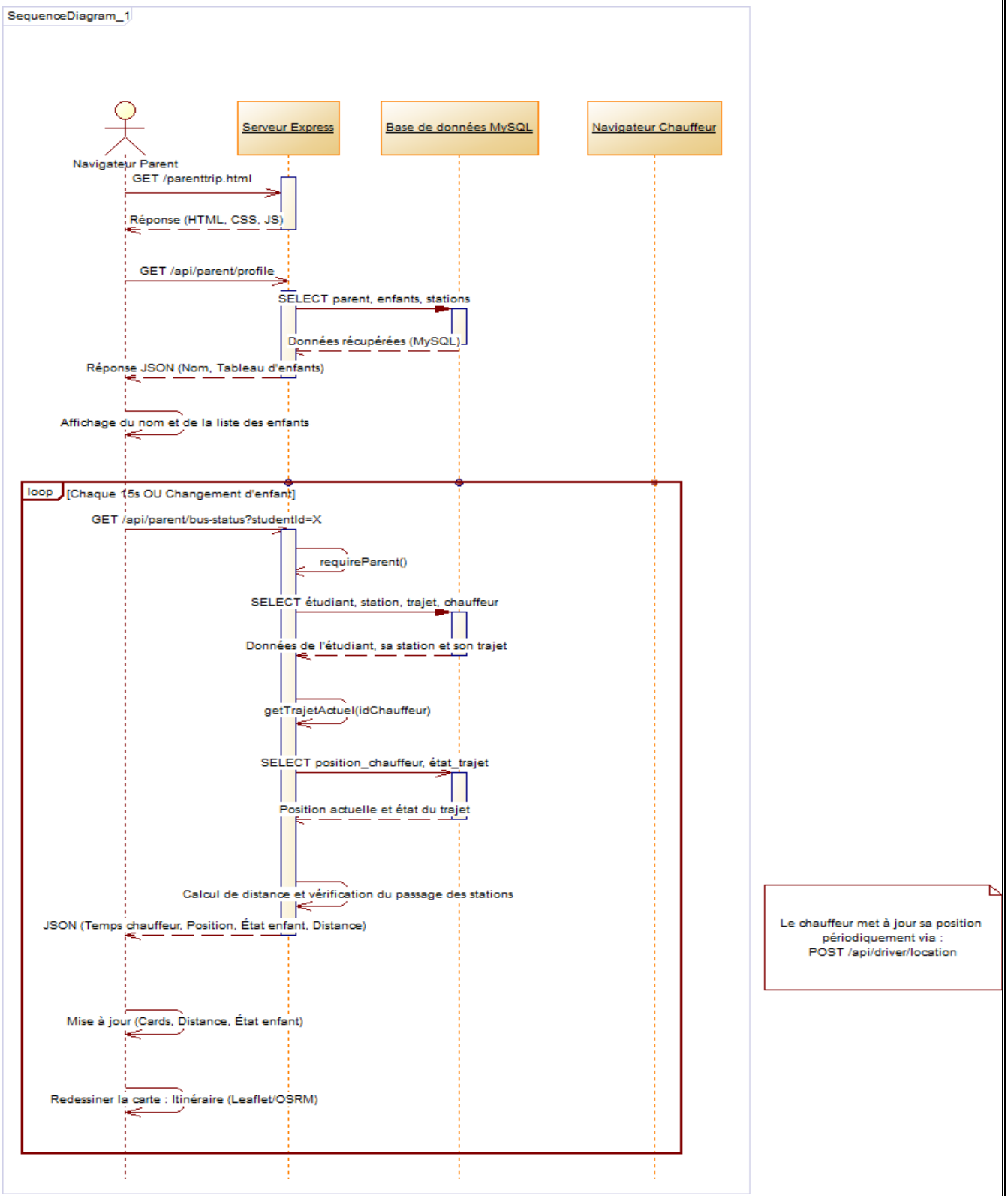



***Figure II 3.2 : Diagramme de séquence du cas d'utilisation "Suivi du bus Côté Parent"***

# Chapitre 3 : Réalisation et Implémentation

## III.1 Introduction

Dans ce chapitre, nous présentons la phase de réalisation et d'implémentation de notre application web et mobile dédiée à la gestion des absences et au suivi du transport scolaire. Nous allons d'abord décrire l'environnement matériel et logiciel utilisé durant le développement, puis présenter les différents outils, technologies, frameworks et bibliothèques adoptés pour la conception du système.

Ensuite, nous détaillerons l'architecture générale de l'application ainsi que les mécanismes de sécurité mis en place afin d'assurer la protection des données et la fiabilité du système. Nous présenterons également les principales fonctionnalités implémentées, notamment le suivi GPS en temps réel, le système de notifications et la gestion des utilisateurs.

Enfin, nous exposerons les différentes interfaces réalisées pour les acteurs du système (administrateur, enseignant, parent et chauffeur), tout en reliant chaque interface aux besoins fonctionnels définis dans le chapitre d'analyse et de conception.

## III.2 Environnement de développement

### III.2.1 Environnement matériel

Le développement ainsi que les différents tests de notre application ont été réalisés sur un ordinateur disposant des caractéristiques matérielles suivantes :

- **Processeur :** AMD Ryzen 4650G
- **Mémoire vive (RAM) :** 16Go
- **Stockage :** SSD 512 Go
- **Architecture du système :** 64 bits

Ces caractéristiques matérielles ont permis d'assurer le bon fonctionnement des outils de développement, l'exécution des différentes composantes du système ainsi que la réalisation des tests dans de bonnes conditions.

### III.2.2 Environnement logiciel

Pour la réalisation de notre projet, nous avons utilisé plusieurs outils logiciels permettant le développement, les tests et l'exécution des différentes parties de l'application.

**Système d'exploitation : Windows 11**

a été utilisé comme environnement principal de développement et d'exécution du projet, grâce à sa stabilité, sa compatibilité avec les technologies employées ainsi que sa prise en charge des différents outils nécessaires au développement.

**Visual Studio Code**

a été utilisé comme éditeur de code principal pour le développement de l'application Web et des différentes fonctionnalités du système. Cet environnement a été choisi grâce à sa légèreté, sa simplicité d'utilisation ainsi qu'à ses nombreuses extensions facilitant le développement avec les technologies Web et Node.js.

**Android Studio**

a été utilisé pour le développement, l'exécution et les tests de l'application mobile. Il permet également d'utiliser un émulateur Android afin de tester les différentes fonctionnalités de l'application dans un environnement proche des conditions réelles d'utilisation.

**MySQL Workbench**

a été utilisé pour la gestion et l'administration de la base de données. Cet outil a permis de créer les tables, gérer les relations entre les données et exécuter différentes requêtes SQL nécessaires au fonctionnement du système.

**Navigateur Web (Google Chrome)**

Le navigateur **Google Chrome** a été utilisé pour exécuter et tester l'application Web. Il a également permis de vérifier le bon fonctionnement des interfaces utilisateur, des formulaires ainsi que des fonctionnalités interactives du système.

**Postman**

**Postman** a été utilisé pour tester certaines requêtes HTTP et vérifier le bon fonctionnement des APIs et des échanges de données entre les différentes parties du système.

L'ensemble de ces outils logiciels a contribué au développement, aux tests et à la validation des différentes fonctionnalités de notre système dans un environnement stable et adapté.

## III.3 Outils et Technologies utilisés

Dans le cadre du développement de notre application web et mobile, nous avons utilisé plusieurs technologies, frameworks et bibliothèques permettant d'assurer la réalisation des interfaces, le traitement des données, la communication avec le serveur ainsi que le suivi du transport scolaire en temps réel.

### III.3.1 Langages de programmation

**Front-end Web**

- **HTML5 :** utilisé pour la structuration des pages du site web.
- **CSS3 :** utilisé pour la mise en forme et le design des interfaces utilisateur.
- **JavaScript :** utilisé pour gérer les interactions utilisateur, les événements ainsi que certaines fonctionnalités dynamiques.

**Développement Mobile**

- **Dart :** langage de programmation utilisé avec Flutter pour le développement de l'application mobile.

**Back-end**

- **JavaScript (Node.js) :** utilisé pour le développement de la partie serveur et le traitement des requêtes côté Back-end.

### III.3.2 Frameworks et Bibliothèques

- **Node.js :** environnement d'exécution JavaScript utilisé pour développer le Back-end de l'application et gérer les requêtes côté serveur.
- **Express.js :** framework utilisé avec Node.js afin de faciliter la gestion des routes (Routes), des requêtes HTTP ainsi que des différentes fonctionnalités du serveur.
- **Flutter :** framework utilisé pour le développement de l'application mobile multiplateforme destinée aux parents et aux chauffeurs.
- **Leaflet.js :** bibliothèque utilisée pour afficher des cartes interactives et assurer le suivi des bus scolaires en temps réel.
- **Mysql2 :** bibliothèque utilisée pour assurer la connexion entre Node.js et la base de données MySQL ainsi que l'exécution des requêtes SQL.
- **flutter_map :** bibliothèque Flutter utilisée pour afficher les cartes et les trajets des bus scolaires dans l'application mobile.
- **latlong2 :** bibliothèque utilisée pour manipuler les coordonnées géographiques et calculer les distances entre les positions GPS.
- **geolocator :** bibliothèque utilisée pour récupérer la position GPS réelle du chauffeur via le téléphone mobile.
- **http :** Bibliothèque utilisée pour envoyer des requêtes HTTP (GET, POST) au format JSON entre l'application Flutter et le serveur **Node.js**
- **Provider :** bibliothèque utilisée pour la gestion de l'état (State Management) de l'application Flutter.
- **Shared_preferences :** bibliothèque utilisée pour sauvegarder certaines données localement, comme le thème de l'application.
- **Google_fonts :** bibliothèque utilisée pour améliorer l'apparence des textes et intégrer des polices Google dans l'application mobile.

### III.3.3 Sécurité et Utilitaires :

Afin d'assurer la sécurité et le bon fonctionnement de notre application, plusieurs bibliothèques et outils utilitaires ont été utilisés durant le développement.

- Bcrypt : bibliothèque utilisée pour le chiffrement des mots de passe des utilisateurs avant leur stockage dans la base de données. Cette technique de chiffrement à sens unique (One-way encryption) permet d'améliorer la sécurité des comptes utilisateurs.
- Nodemailer : bibliothèque utilisée pour l'envoi des emails et des notifications aux parents, notamment concernant les absences et certaines informations liées au transport scolaire.
- Body-parser : middleware utilisé pour lire et traiter les données envoyées par les utilisateurs à travers les formulaires et les requêtes HTTP.
- Session : utilisée pour gérer les sessions utilisateurs et conserver les informations de connexion durant l'utilisation de l'application.
- HTTPS : utilisé afin de sécuriser les échanges entre le client et le serveur grâce au chiffrement des données transmises.
- FS (File System) : module utilisé pour lire, écrire et gérer les fichiers au niveau du serveur.
- Path : module utilisé pour manipuler les chemins des fichiers et assurer une gestion correcte et sécurisée des ressources du système.

En outre, notre application est protégée contre les attaques SQL Injection grâce à l'utilisation des requêtes préparées (Prepared Statements). Cette méthode permet de séparer les commandes SQL des données saisies par l'utilisateur afin d'empêcher l'exécution de code malveillant dans la base de données.

Exemple :

SELECT id_sta FROM station WHERE S_Name = ? LIMIT 1

Dans cette requête, la valeur saisie par l'utilisateur est traitée comme une donnée contrôlée et non comme une commande SQL, ce qui améliore considérablement la sécurité du système.

## III.4 Architecture et Implémentation

### III.4.1 Architecture générale du système

L'architecture de notre système repose sur une structure client-serveur composée d'une application web, d'une application mobile ainsi que d'un serveur central chargé du traitement des données et de la communication avec la base de données.

Le système est organisé selon une architecture en trois couches principales :

**La couche présentation**

Cette couche représente les interfaces utilisées par les différents acteurs du système.

- L'application web développée avec HTML, CSS et JavaScript est principalement destinée à l'administrateur et à l'enseignant pour la gestion des utilisateurs, des absences et des informations scolaires.
- L'application mobile développée avec Flutter est destinée aux parents et aux chauffeurs afin d'assurer le suivi du transport scolaire, la consultation des informations et la réception des notifications.

Cette couche communique avec la couche logique à travers des requêtes HTTP.

**La couche logique**

Cette couche contient les traitements principaux du système ainsi que les différentes fonctionnalités métier.

- La partie Web utilise un serveur développé avec Node.js et Express pour traiter les requêtes, gérer les utilisateurs, les absences ainsi que la communication avec la base de données.
- La partie Mobile communique avec des APIs développées avec Node.js et Express.js permettant de gérer les opérations liées à la géolocalisation, au suivi GPS des bus scolaires et à l'envoi des notifications**.**

Cette couche assure également la validation des données, le traitement des informations ainsi que la sécurisation des échanges.

**La couche données**

Cette couche représente la base de données MySQL utilisée pour stocker toutes les informations du système telles que :

- Les utilisateurs,
- Les élèves,
- Les absences,
- Les trajets,
- Les stations,
- Les positions GPS,
- ainsi que les notifications.

La communication entre les différentes couches permet d'assurer un fonctionnement dynamique et temps réel de l'application.

### III.4.2 Logique de l'application

Cette section présente les principales fonctions implémentées dans notre système.

**Fonction getActiveTripForDriver**

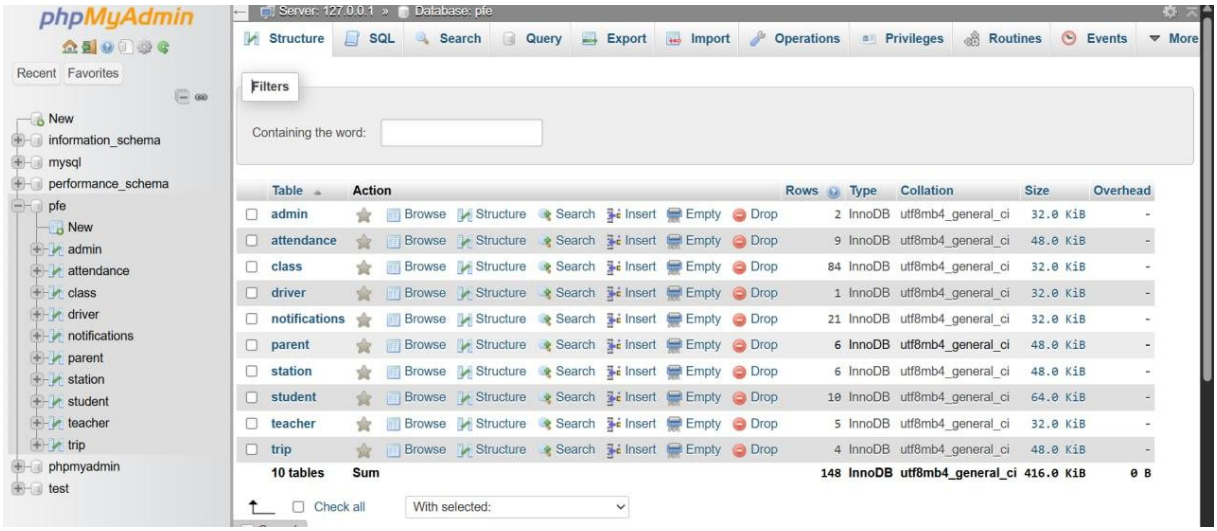
Cette fonction prend l'identifiant du chauffeur en entrée, puis récupère la position GPS du chauffeur toutes les 5 secondes. Elle met à jour la carte en temps réel et affiche le trajet de la station actuelle vers la destination finale. Cette fonctionnalité permet un suivi dynamique du bus scolaire.

**Fonction sendEmail**

Cette fonction permet d'envoyer des emails aux utilisateurs. Elle prend en paramètres l'adresse email du destinataire ainsi que le message à envoyer. Elle est utilisée principalement pour notifier les parents concernant les absences ou les événements liés au transport scolaire.

### III.4.3 Schéma relationnel

Structure de la Base de Données (MySQL / phpMyAdmin)




## III.5 Présentation des interfaces

### III.5.1 Interface Administrateur — Gestion des parents

**Description :** Cette interface permet à l'administrateur d'enregistrer un nouveau parent via un formulaire dédié et de visualiser la liste de tous les comptes existants dans un tableau pour les modifier ou les supprimer

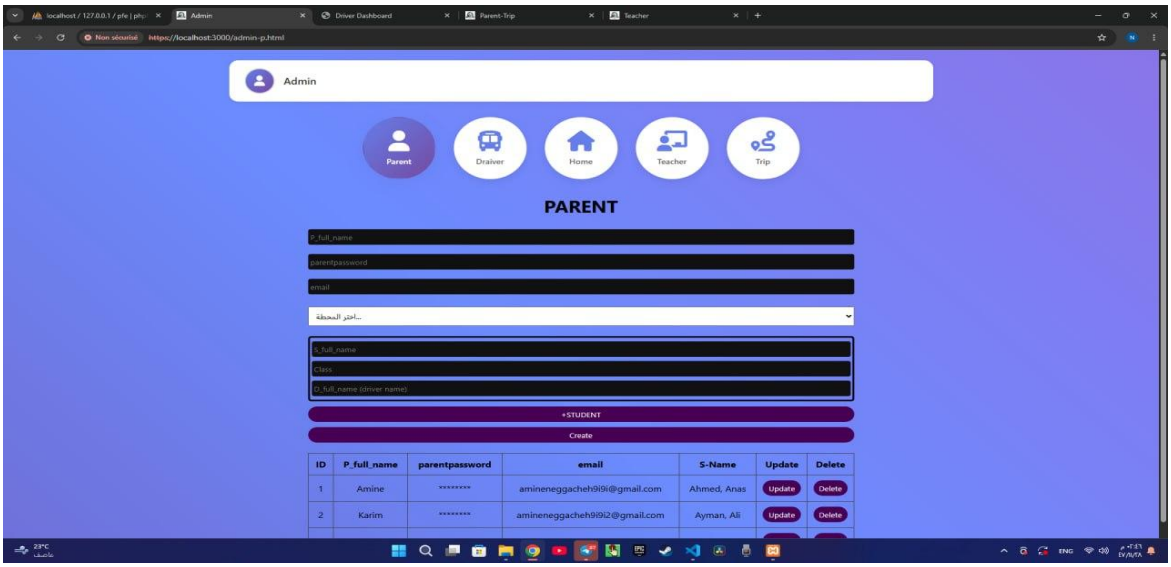



***Figure* III.4.1 : Interface Administrateur — Gestion des parents web**

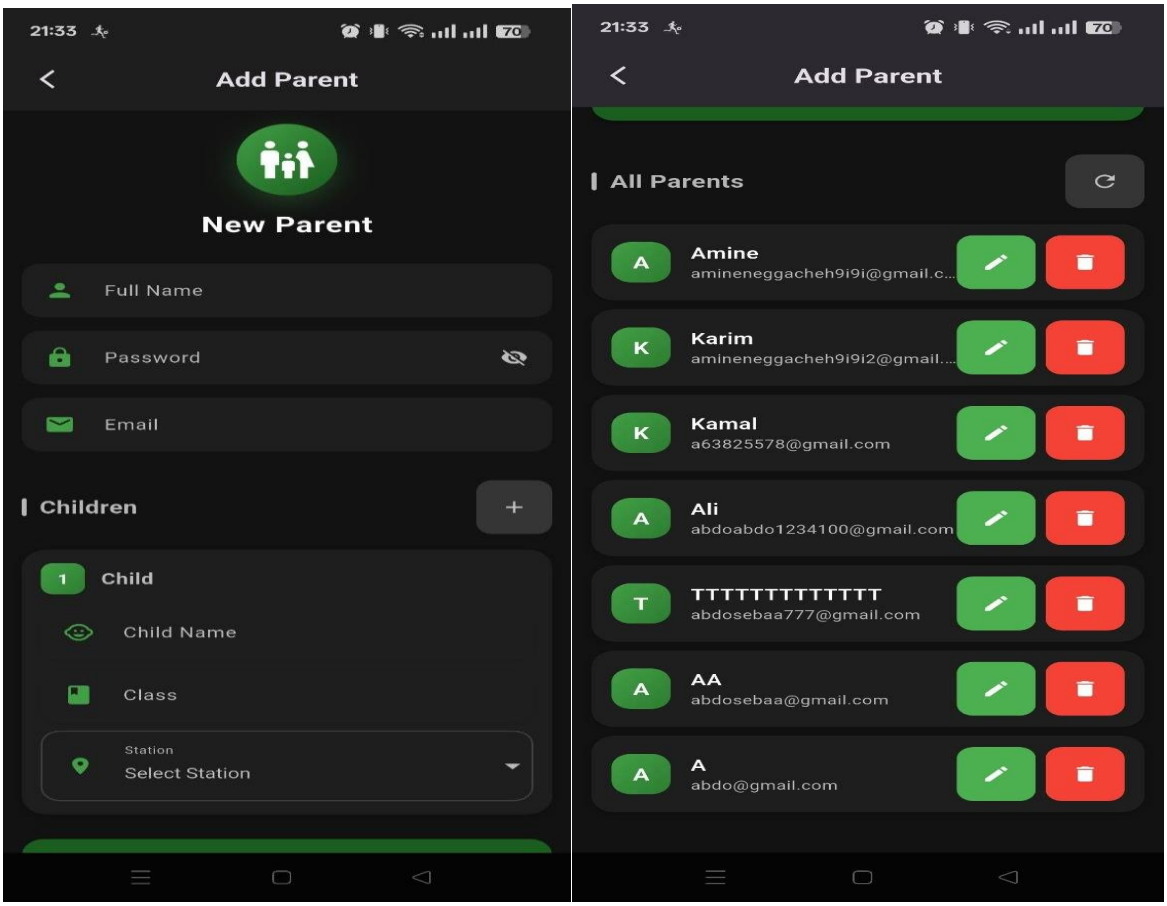



***Figure* III.*4.2* Interface Administrateur — Gestion des parents App**

## III.5.2 Interface Administrateur — Gestion des enseignants

**Description :** Cet écran permet à l'administrateur de créer les comptes des enseignants et de configurer leur emploi du temps hebdomadaire avant de valider l'enregistrement via le bouton "**Create**"

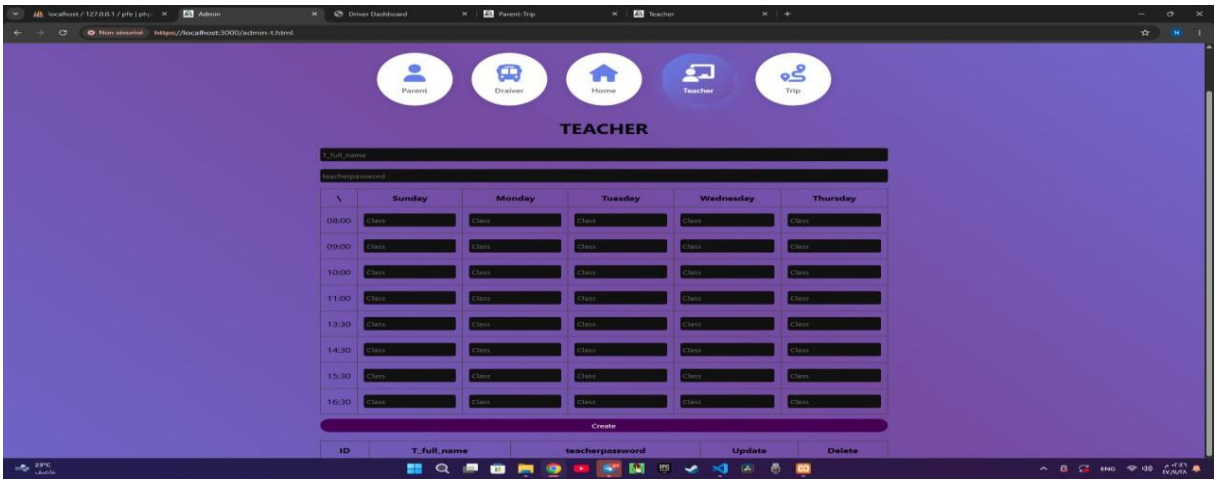



***Figure* III *4.3* Interface Administrateur — Gestion des enseignants web**

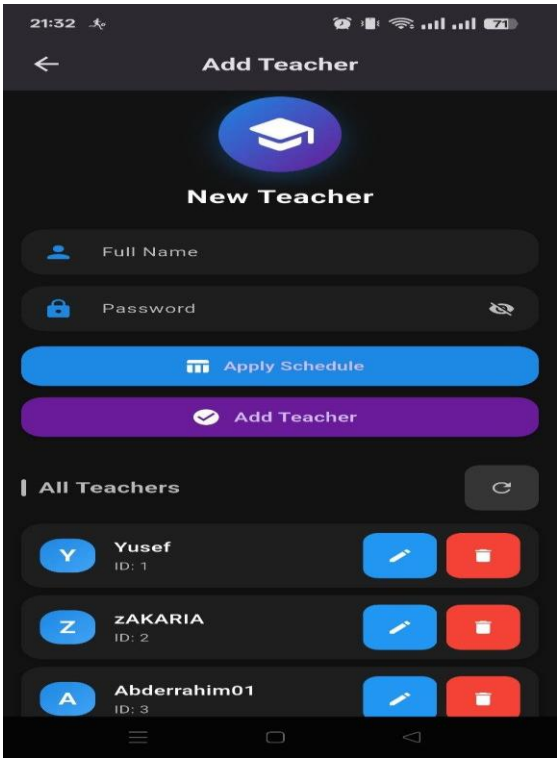



***Figure* III *4.4* Interface Administrateur — Gestion des enseignants App**

## III.5.3 Interface Parent — Suivi des absences

**Description :** Cette interface permet au parent connecté de suivre l'assiduité de ses enfants. En accédant à l'onglet "**Attendance**", le système génère un historique détaillé des absences avec les dates, les plages horaires et le statut de chaque enregistrement.

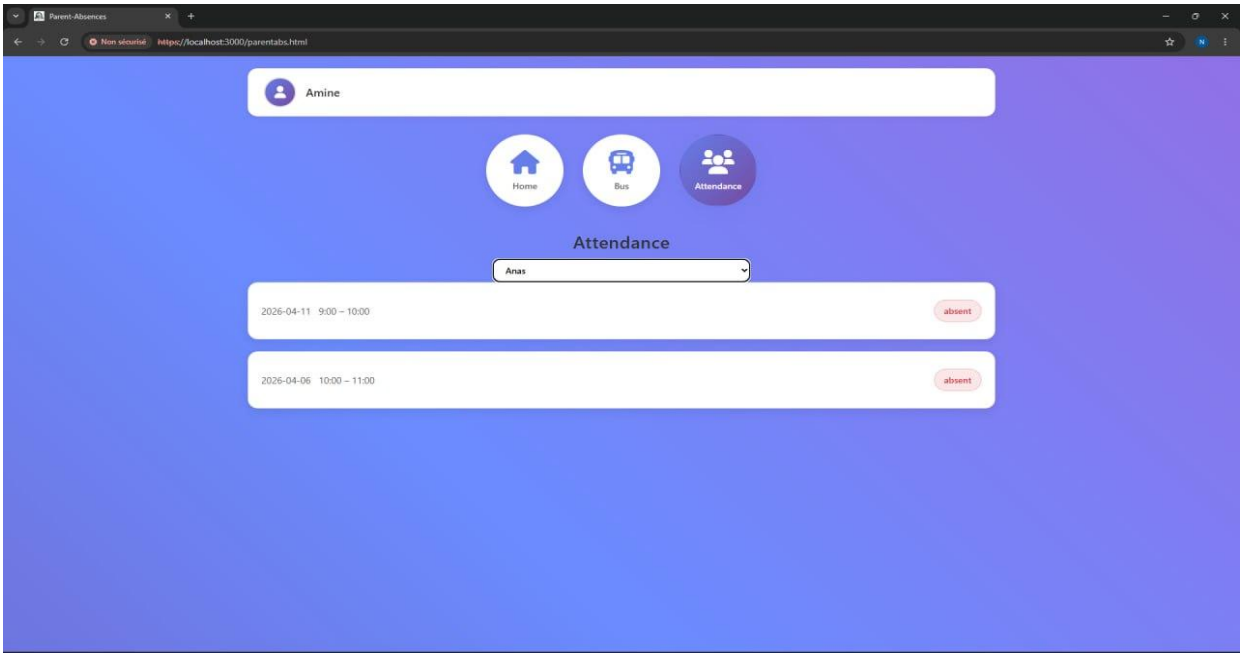



***Figure* III *5.1:* Interface Parent — Suivi des absences web**

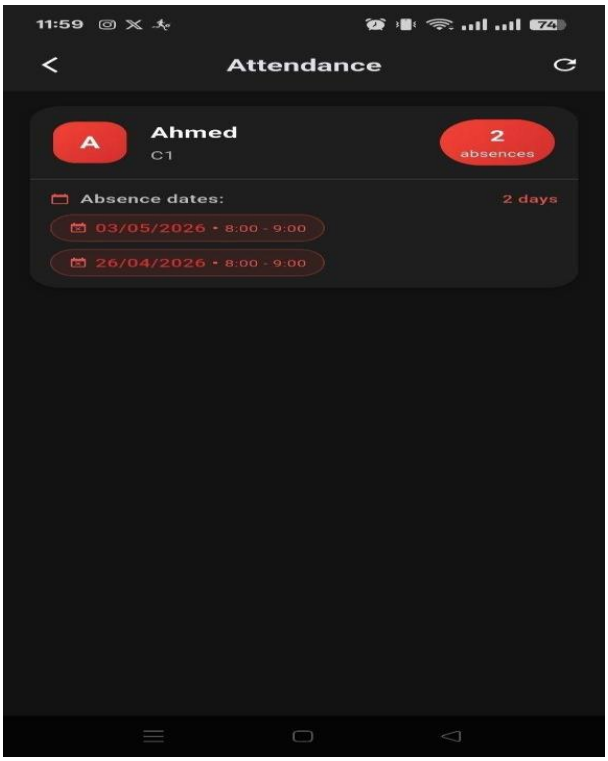



***Figure* III *5.2*Interface Parent — Suivi des absences App**

## III.5.4 Interface Parent — Suivi du bus en temps réel

**Description :** Cette interface permet au parent connecté de géolocaliser en temps réel le bus de son enfant sur une carte interactive, tout en affichant des indicateurs clés comme le nom du chauffeur, la distance restante et le statut d'embarquement de l'élève.

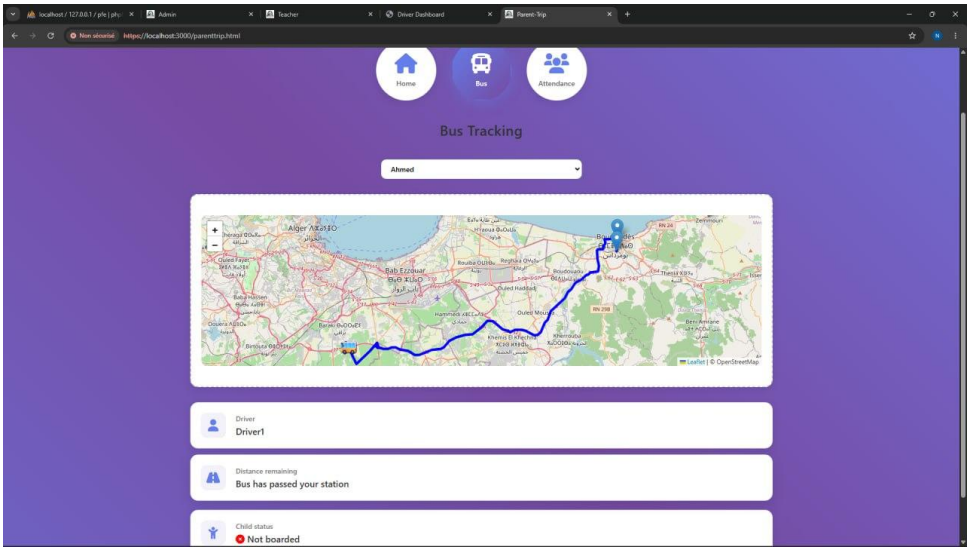



***Figure* III *5.3:* Interface Parent — Suivi du bus en temps réel Web**




***Figure* III *5.4:* Interface Parent — Suivi du bus en temps réel App**

## III.5.5 Interface Enseignant — Emploi du temps

**Description :** Cette interface présente l'emploi du temps hebdomadaire de l'enseignant connecté (**Ahmed Mohamed**). En accédant à l'onglet "**Schedule**", l'enseignant peut visualiser la répartition de ses séances de cours du Dimanche au Jeudi. Le tableau affiche de manière organisée les créneaux horaires (périodes) et indique la classe affectée à chaque séance (par exemple : **C1**, **C2**, **C3**)

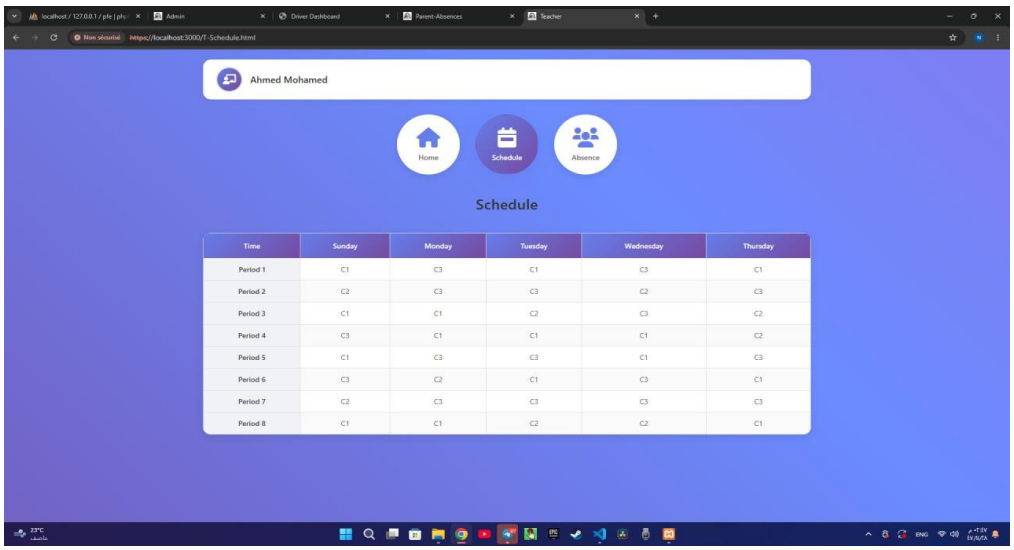



***Figure* III.*6.1:* Interface Enseignant — Emploi du temps web**

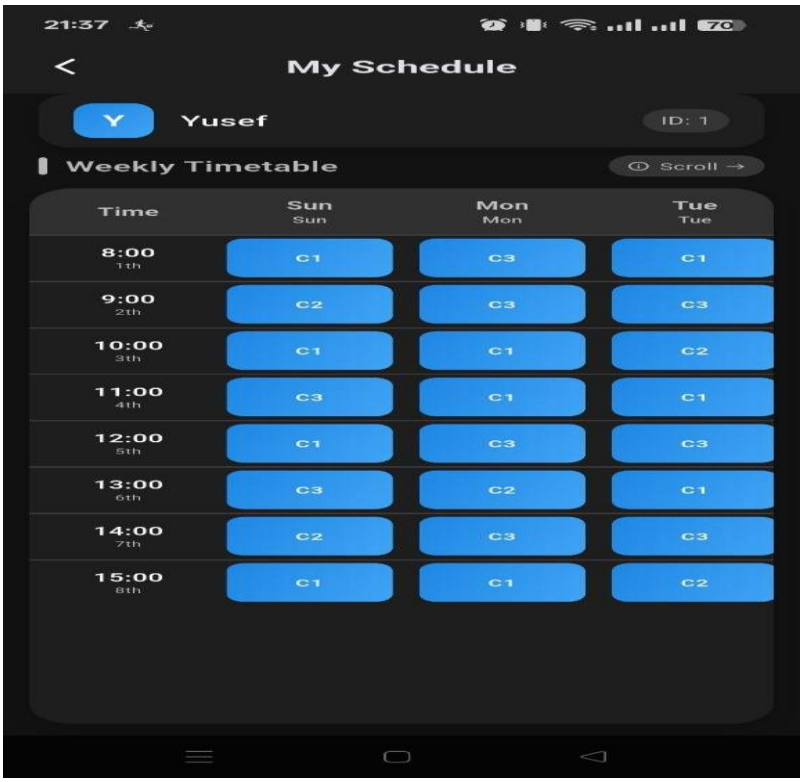



***Figure* III *6.2 :* Interface Enseignant — Emploi du temps App**

## III.5.6 Interface Enseignant — Prise d'absence

**Description :** Cette interface permet à l'enseignant de réaliser l'appel numérique en classe. Après avoir sélectionné la date et la plage horaire correspondante, l'enseignant clique sur le bouton "**Show Class**" pour afficher la liste des élèves de la classe concernée (ici la classe **C3**). Il peut ensuite marquer chaque élève comme présent via le bouton vert "**Present**" ou absent via le bouton rouge "**Absent**".

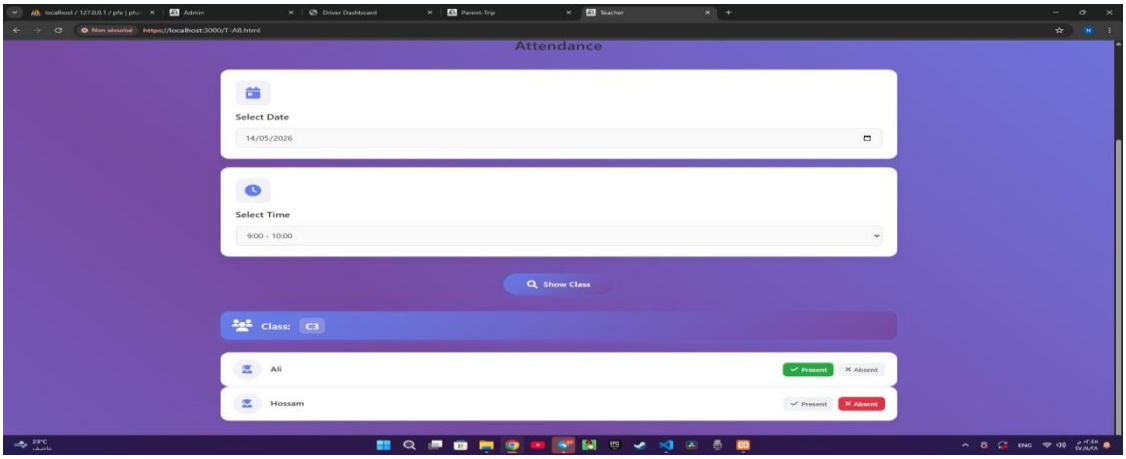



***Figure* III *6.3 :* Interface Enseignant — Prise d'absence Web**

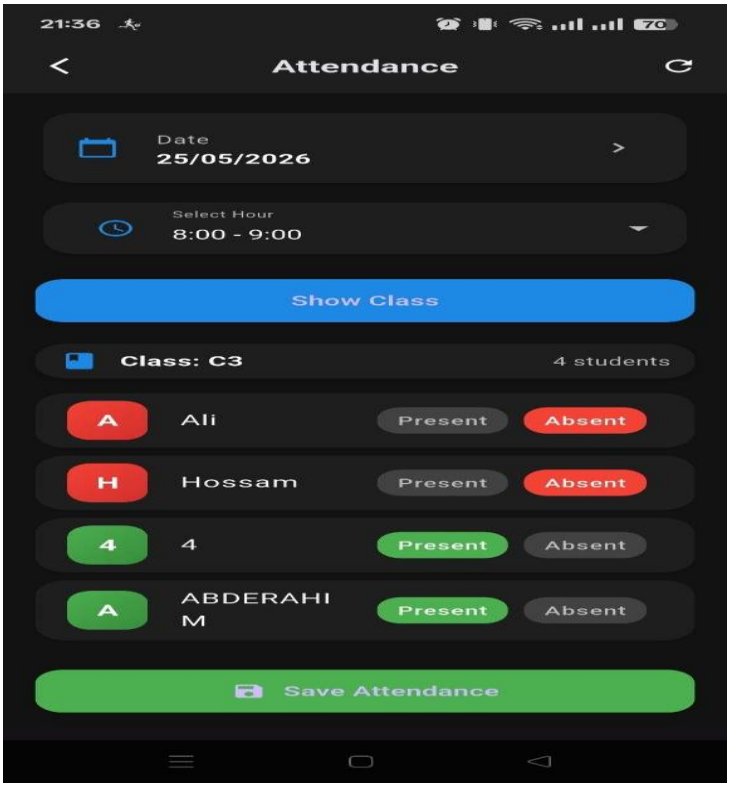



***Figure* III.*6.4:* Interface Enseignant — Prise d'absence App**

## III.5.7 Interface Chauffeur — Tableau de bord

**Description :** Cette interface est dédiée au conducteur du bus (**Driver1**) pour piloter sa tournée active (ici une tournée aller : **Outbound**). L'écran affiche une carte interactive avec l'itinéraire tracé et des instructions de navigation détaillées sur le côté droit. En bas, l'interface fournit des outils de gestion en temps réel : la station actuelle (**Current Stop**), le nombre d'élèves attendus, et un compte à rebours pour le temps d'attente (**Waiting Time**). De plus, le chauffeur dispose de la liste des élèves à récupérer (**Ayman** et **Ali**) avec des boutons pour valider leur embarquement (**Board**) ou signaler leur absence (**No**), avant de passer à l'arrêt suivant via le bouton "**Next Stop**".

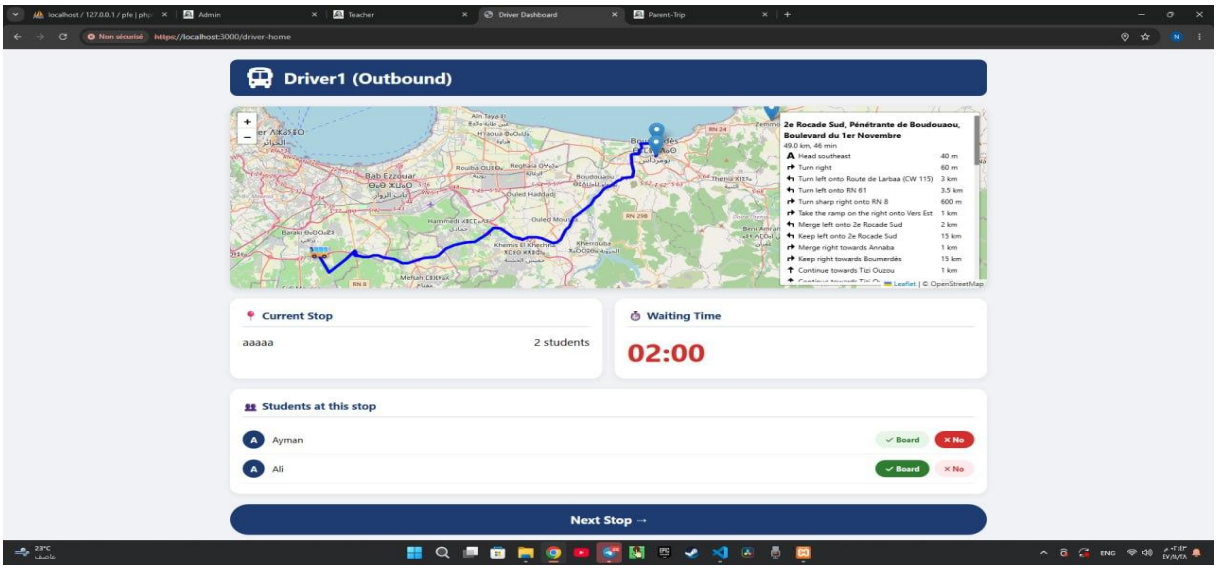



***Figure* III *7.1* Interface Chauffeur — Tableau de bord Web**

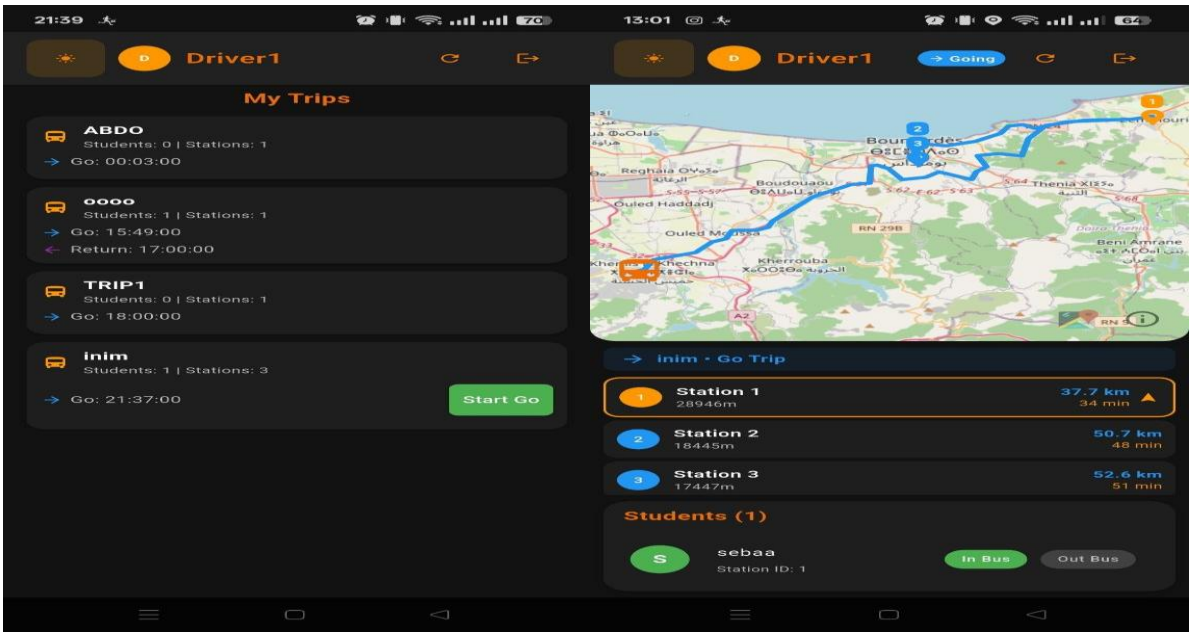



***Figure* III.*7.2:* Interface Chauffeur — Tableau de bord App**

## Conclusion du chapitre

Dans ce chapitre, nous avons présenté la phase de réalisation de notre application. Nous avons décrit l'environnement matériel et logiciel utilisé, les technologies et frameworks adoptés, ainsi que l'architecture générale du système. Nous avons également exposé les différentes interfaces développées pour chaque acteur de l'application, illustrant ainsi la concrétisation des besoins identifiés lors des chapitres précédents.

# Conclusion générale

Ce travail nous a permis de concevoir et de développer une application web et mobile dédiée à la gestion des absences des élèves et au suivi du transport scolaire en temps réel.

À travers les différentes étapes de ce projet, nous avons d'abord analysé les limites du système manuel existant, puis proposé une solution numérique moderne reposant sur des technologies actuelles telles que Flutter, Node.js, Express.js et MySQL. Cette solution intègre des fonctionnalités clés comme la géolocalisation GPS des bus via Leaflet.js, la gestion automatisée des absences, ainsi qu'un système de notifications permettant d'alerter les parents en temps réel.

Les résultats obtenus montrent que l'application répond aux besoins identifiés : elle améliore la communication entre les différents acteurs, renforce la sécurité des élèves et optimise la gestion administrative au sein de l'établissement.

Ce projet nous a également permis d'approfondir nos compétences en développement mobile multiplateforme, en conception de bases de données relationnelles et en modélisation UML.

## *Bibliographies*

[1] Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide. 2nd Edition, Addison-Wesley, 2005.

[2] Pascal ROQUES. UML 2 par la pratique: Études de cas et exercices corrigés. 7ème édition, Groupe Eyrolles, 2011

[3] Laurent AUDIBERT. UML 2: De l'apprentissage à la pratique. Edition de l'Université de Limoges, 2013 )

[4] GILLES Roy, Conception de bases de données avec UML, Presses de l'Univer-sité de Québec, 2009, lere édition.

[5] Stéphane Crozat. Cours: Passage UML - Relationnel: classes et associations. Université de technologie de Compiègne. 2016.

Flutter Team. *Flutter Documentation*. Google, Available at: https://docs.flutter.dev

Node.js Foundation. *Node.js Documentation*. Available at: https://nodejs.org/docs

Express.js. *Express Web Framework Documentation*. Available at: https://expressjs.com

Oracle Corporation. *MySQL Reference Manual*. Available at: https://dev.mysql.com/doc