L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ M'HAMED BOUGARA - BOUMERDES



Faculté des sciences

Département d'informatique

Mémoire de fin de cycle en vue de l'obtention du diplôme de licence

Théme:

Conception et Réalisation d'une Application Web pour la gestion d'une auto-école

Encadré par:

M. Baroudi Mouhammed Yassine

Examiné par:

M. Meftah Mustapha

Présenté par:

Aici Lynda

Ghoraf Sabrina

Kahli Rayene

Mana Fatima Zahra

Anné universitaire: 2025-2026

Table des matières:

Chapitre 01: Introduction Générale. 4
1.1 Introduction: 4
1.2 context du projet: 4
1.3 Problématique du projet: 4
1.4.Objectif du projet: 5
1.5 Solution proposée: 5
1.6 Méthodologie de travail: 5
Chapitre 02: Etude et analyse 6
2.1 Introduction: 6
2.2 Cahier descharges: 6
2.2.1 Identification des acteurs: 6
2.2.2 Besoins fonctionnels: 6
2.2.3 Besoins non fonctionnels: 6
2.2.4 Constraintes du systéme: 7
2.3 Diagramme de Cas d'utulisation: 7
2.3.1 Acteurs du système: 7
2.3.2 Description des cas d'utilisation: 7
2.3.3 Représentation du diagramme de cas d'utulisation: 8
Chapitre 03: conception. 10
3.1 introduction:....10
3.2 Diagramme de classe:....10
3.2.1 Représentation du diagramme de Classe: 10
3.3. Diagramme de séquence:....12
3.3.1 Représentation du Diagramme de Séquence: 12
3.3.1.1 Diagramme de Séquence N°1: planifier séance 12
a) Acteurs du système: 12
b) Description du scenario principal: ….....12
3.3.1.2 Diagramme de Séquence N°2: Paiement….....13
a) Acteurs du Système: 13
b) Description du scénario principal: 14
3.4 Diagramme de d'activité: 15
3.4.1 Représentation du Diagramme d'activité: 15
a) Acteurs du systéme: 15

b) Description des étapes (Le Flux): 15
3.5 Architecture logique et technique: 17
3.5.1 Architecture logique: 17
3.5.2 Architecture technique: 17
3.5.3 Transition vers la phase de Développement: 17
Chapitre 04: Développement 18
4.1 Introduction: 18
4.2 Environnement de développement: 18
4.3 Présentation générale de l'application: 19
4.4 Représentation des interfaces del'application: ….....19
4.4.1 Page d'accueil: 19
4.4.2 Interfaces d'Authentification: 19
4.4.3 Interfaces Admin: 20
4.4.3.1 Interfaces Tableau de Bord: 20
4.4.3.2 Gestion des candidats: 21
4.4.3.3 Gestion moniteurs: 22
4.4.3.4 Gestion véhicules: 22
4.4.3.6 Gestion des Paiements: 24
4.4.3.7 Gestion des Examens: 25
4.4.2.8 Consulter des Rapports d'examens: 25
4.4.4 Interfaces Moniteur: 26
4.4.4.1 Accueil Moniteur: 26
4.4.4.2 Emplois du temps (Planning): 26
4.4.4.3 Consultations des candidats: 27
4.4.4.4 Messagerie: 27
4.4.4.5 valider séance: 27
4.4.5 Interfaces Candidat: 28
4.4.5.1 Accueil Candidat: 28
4.4.5.2 Notifications: 28
4.4.5.3 Consultation des séances: 29
4.4.5.4 Consulter Paiements: 29
4.4.5.5 Consulter Mes Résults: ….....29

Chapitre 05: Test et Validations 30
5.1 Introduction: 30
5.2 Objectif des tests: 30
5.3 Tests réalisés: 30
5.4 Difficultés rencontrées: 30
5.5 Solutions apportées: 30
5.6 Conclusion: 30
Conclusion Générale: 31
Bibliographie &Webographie: 32

Chapitre 01: Introduction Générale:

• Introduction:

Dans le cadre de la formation universitaire en Licence Informatique, ce projet de fin d'études consiste à concevoir et réaliser une application de gestion d'auto-école.

Ce travail a pour objectif de faciliter la gestion des élèves, des moniteurs, des inscriptions ainsi que le suivi des examens théoriques et pratiques. L'application vise également à automatiser certaines tâches administratives afin de gagner du temps et d'améliorer l'efficacité du système de gestion. Ce projet permet de mettre en pratique les connaissances acquises durant le cursus universitaire, notamment en développement web, en bases de données et en conception d'applications.

Enfin, ce mémoire est structuré en plusieurs parties: une partie théorique, une partie analyse et conception, et une partie réalisation du projet. **[2] [3]**

• 1.2 context du projet:

Dans un contexte où la digitalisation devient indispensable, les auto-écoles cherchent à améliorer leurs méthodes de gestion afin de gagner en efficacité et en organisation. Cependant, plusieurs d'entre elles utilisent encore des méthodes traditionnelles basées sur le papier ou des outils non adaptés.

Dans ce cadre, notre projet consiste à concevoir et développer une application de gestion d'autoécole permettant de centraliser et d'automatiser les différentes opérations administratives et pédagogiques. **[3]**

• 1.3 Problématique du projet:

La gestion d'une auto-école reste souvent complexe et peu organisée**.** malgré son importance dans la formation des conducteurs. Les méthodes traditionnelles utilisées rendent difficile le suivi des différentes activités et peuvent nuire à l'efficacité globale.

Problèmes rencontrés:

- Dispersion des informations (élèves, séances, examens, paiements)
- Risque d'erreurs dans la gestion des données

- Difficulté de suivi des élèves et de leur progression Manque d'automatisation des taches administratives. **[3]**
- Perte ou mauvaise organisation des informations

• 1.4 Objectif du projet:

Ce projet a pour objectif principal de concevoir et développer une application permettant d'améliorer l'organisation et le suivi des activités d'une auto-école**,** en rendant les opérations plus simples, rapides et efficaces.

Objectifs spécifiques:

- Gérer les informations des élèves (inscription, suivi, dossier)
- Organiser les séances de conduite et les plannings
- Gérer les moniteurs et leurs disponibilités
- Assurer le suivi des examens (théorique et pratique)
- Gérer les paiements et les dossiers administratifs
- Automatiser les tâches pour réduire les erreurs Améliorer l'organisation et la qualité des services. **[3]**

• 1.5 Solution proposée:

Afin de répondre aux différents problèmes rencontrés dans la gestion traditionnelle des auto-écoles, nous proposons une application web permettant d'automatiser plusieurs tâches administratives et pédagogiques. Cette solution facilite la gestion des candidats, des moniteurs, des paiements, des examens et des séances de conduite à travers une interface simple, intuitive et accessible. **[2][3]**

• 1.6 Méthodologie de travail:

Pour la réalisation de ce projet, nous avons suivi plusieurs étapes principales. Nous avons commencé par l'analyse des besoins et l'étude du système, puis nous avons réalisé la conception UML de l'application. Ensuite, nous avons développé l'application web en utilisant plusieurs technologies comme PHP, MySQL, HTML, CSS et JavaScript. Enfin, nous avons effectué des tests afin de vérifier le bon fonctionnement du système. **[1][2][3][4]**

Chapitre 02: Etude et analyse:

2.1 introduction:

Une auto-école est un établissement spécialisé dans la formation des candidats à la conduite. Elle assure des cours théoriques (code de la route) ainsi que des séances pratiques de conduite. Elle permet également de préparer les candidats aux examens théoriques et pratiques afin d'obtenir le permis de conduire.

Dans la majorité des cas, la gestion des auto-écoles repose encore sur des méthodes traditionnelles, ce qui peut rendre le suivi des candidats et l'organisation des activités difficile et peu efficace. **[3]**

2.2 Cahier des charges:

2.2.1 Identification des acteurs:

Le système de gestion d'une auto-école repose principalement sur trois acteurs:

- ✓ **Administrateur (Admin)**: responsable de la gestion globale du système (candidats, moniteurs, véhicules, paiements, examens).
- ✓ **Moniteur**: assure les séances de conduite et suit la progression des candidats.
- ✓ **Candidat**: s'inscrit à l'auto-école et participe aux formations théoriques et pratiques. **[1]**

2.2.2 Besoins fonctionnels:

Le système doit permettre les fonctionnalités suivantes:

- Gestion des candidats (ajout, modification, suppression, suivi)
- Gestion des moniteurs
- Gestion des véhicules
- Planification des leçons de conduite
- Gestion des séances de code
- Gestion des paiements avec génération de reçus
- Gestion des examens théoriques et pratiques
- Consultation des statistiques et rapports. **[3]**

2.2.3 Besoins non fonctionnels:

Le système doit respecter plusieurs contraintes de qualité:

- Sécurité des données et protection des informations

- Accessibilité via une application web
- Interface simple, intuitive et facile à utiliser
- Performance et rapidité d'exécution
- Sauvegarde régulière de la base de données
- Historisation des actions des utilisateurs
- Gestion sécurisée des accès
- Protection des données utilisateurs **[3]**

➢ 2.2.4 Constraintes du systéme:

- **Développement d'une application web**
- **Compatibilité avec différents navigateurs web**
- **utulisation d'une base de données Relationnelle (MySQL).[2]**

• 2,3 Diagramme de Cas d'utulisation:

Le diagramme de cas d'utilisation permet de représenter les interactions entre les différents acteurs du système et les fonctionnalités offertes par l'application de gestion d'auto-école. Il aide à comprendre le comportement global du système du point de vue des utilisateurs. **[1] [16]**

➢ 2.3.1 Acteurs du système:

✓ Le système comporte trois acteurs principaux:

✓ Administrateur (Admin): gère l'ensemble du système.

✓ Moniteur: assure le suivi des séances de conduite.

✓ Candidat: suit sa formation et consulte ses informations. **[1]**

➢ 2.3.2 Description des cas d'utilisation:

L'application offre plusieurs fonctionnalités aux différents acteurs:

➢ Cas d'utilisation de l'Administrateur:

➢ L'administrateur a les droits suivants:

✓ Gérer les candidats (ajouter, modifier, supprimer)

✓ Gérer les moniteurs

✓ Gérer les véhicules

✓ Planifier les séances (code, conduite)

✓ Gérer les paiements (ajout et suivi)

✓ Consulter les statistiques.

- ✓ Gérer les examens (théorique et pratique)

➢ Cas d'utilisation du Moniteur:

Le moniteur peut:

- ✓ Consulter son planning
- ✓ Suivre les candidats
- ✓ Valider les séances de conduite

➢ Cas d'utilisation du Candidat:

Le candidat peut:

- ✓ Consulter son planning
- ✓ Suivre ses séances
- ✓ Consulter ses résultats
- ✓ Consulter son état de paiement. **[1]**

➢ 2.3.3 Représentation du diagramme de cas d'utulisation:

Le diagramme de cas d'utilisation est représenté graphiquement afin de montrer clairement les interactions entre les différents acteurs du système et les fonctionnalités offertes par l'application de gestion d'auto-école. Il permet de visualiser de manière globale le fonctionnement du système. **[1]**

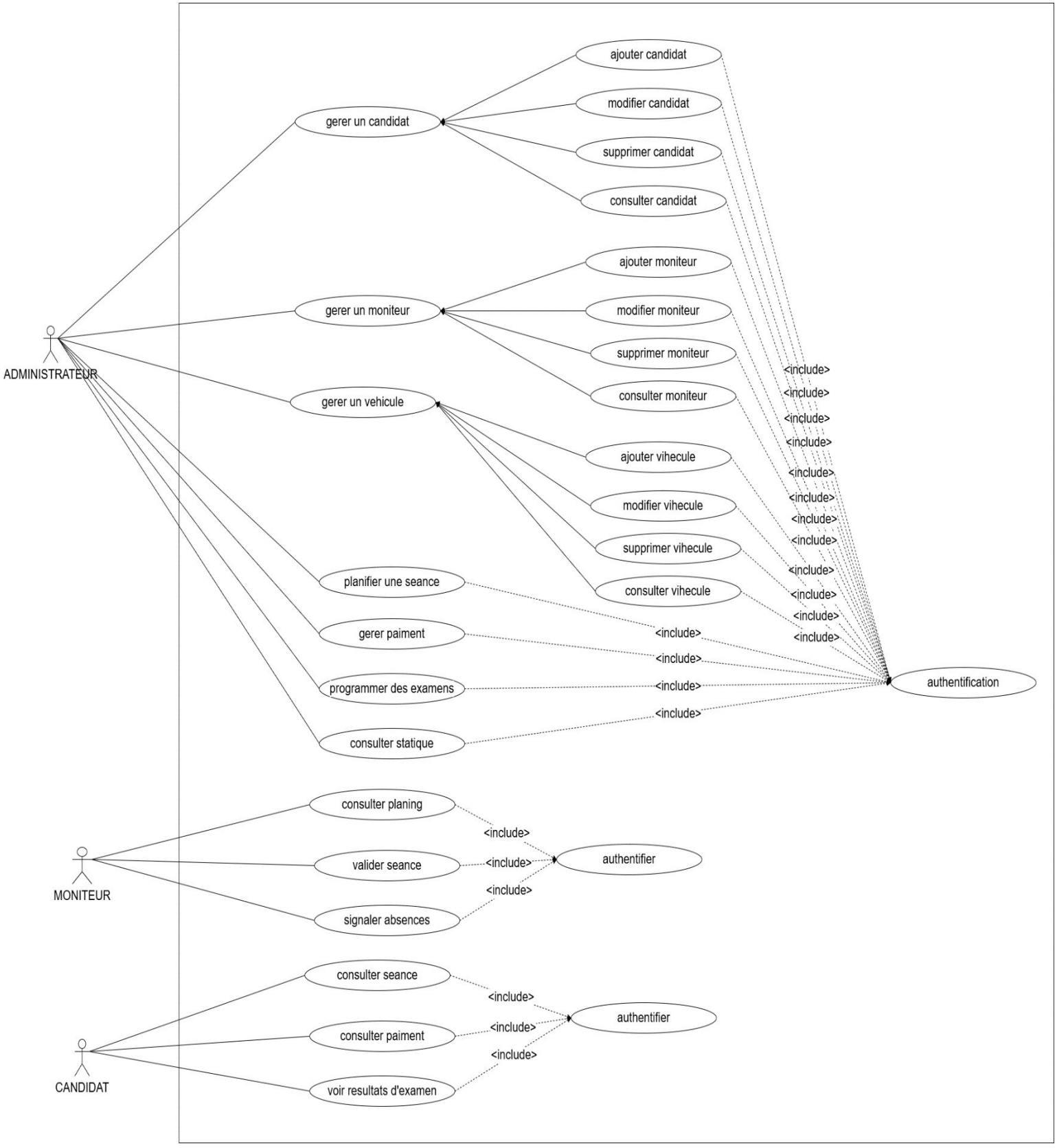


Figure2.1: Diagramme de Cas d'utulisation.

Chapitre 03: conception:

- ## 3.1 introduction:

Dans ce chapitre, nous présentons la phase de conception de notre application de gestion d'autoécole. Cette étape permet de définir la structure du système et d'organiser ses différentes parties avant le développement. Nous allons présenter le diagramme de classes ainsi que le diagramme de séquence afin de mieux comprendre le fonctionnement et l'organisation du système. **[1]**

- ## 3.2 Diagramme de classe:

Le diagramme de classes représente l'organisation statique du système ainsi que les relations entre les différentes entités en montrant les différentes classes, leurs attributs ainsi que les relations entre elles. Dans notre application de gestion d'auto-école, ce diagramme permet de définir les principales entités du système et leurs interactions afin de mieux organiser la base de données et le fonctionnement global de l'application. **[1] [7] [16]**

➢ 3.2.1 Représentation du diagramme de Classe:

Le diagramme ci-dessous illustre les différentes classes du système ainsi que leurs relations, permettant de représenter l'organisation de l'application de gestion d'auto-école.

Le système est composé de plusieurs entités principales:

- **Utilisateur**: représente les utilisateurs du système avec un identifiant, un nom, un email et un rôle.
- **Candidat**: contient les informations personnelles du candidat ainsi que les données liées au permis.
- **Moniteur**: représente les moniteurs avec leurs diplômes et disponibilités.
- Véhicule: décrit les véhicules utilisés dans les séances de conduite.
- **SéanceConduite**: représente les séances de conduite avec date, heure, moniteur, candidat, véhicule
- **SéanceCode**: représente les séances de code avec date, heure et salle.
- **Paiement**: gère les paiements effectués par les candidats.
- **Examen**: contient les informations sur les examens (code ou pratique) et leurs résultats.**[1]**

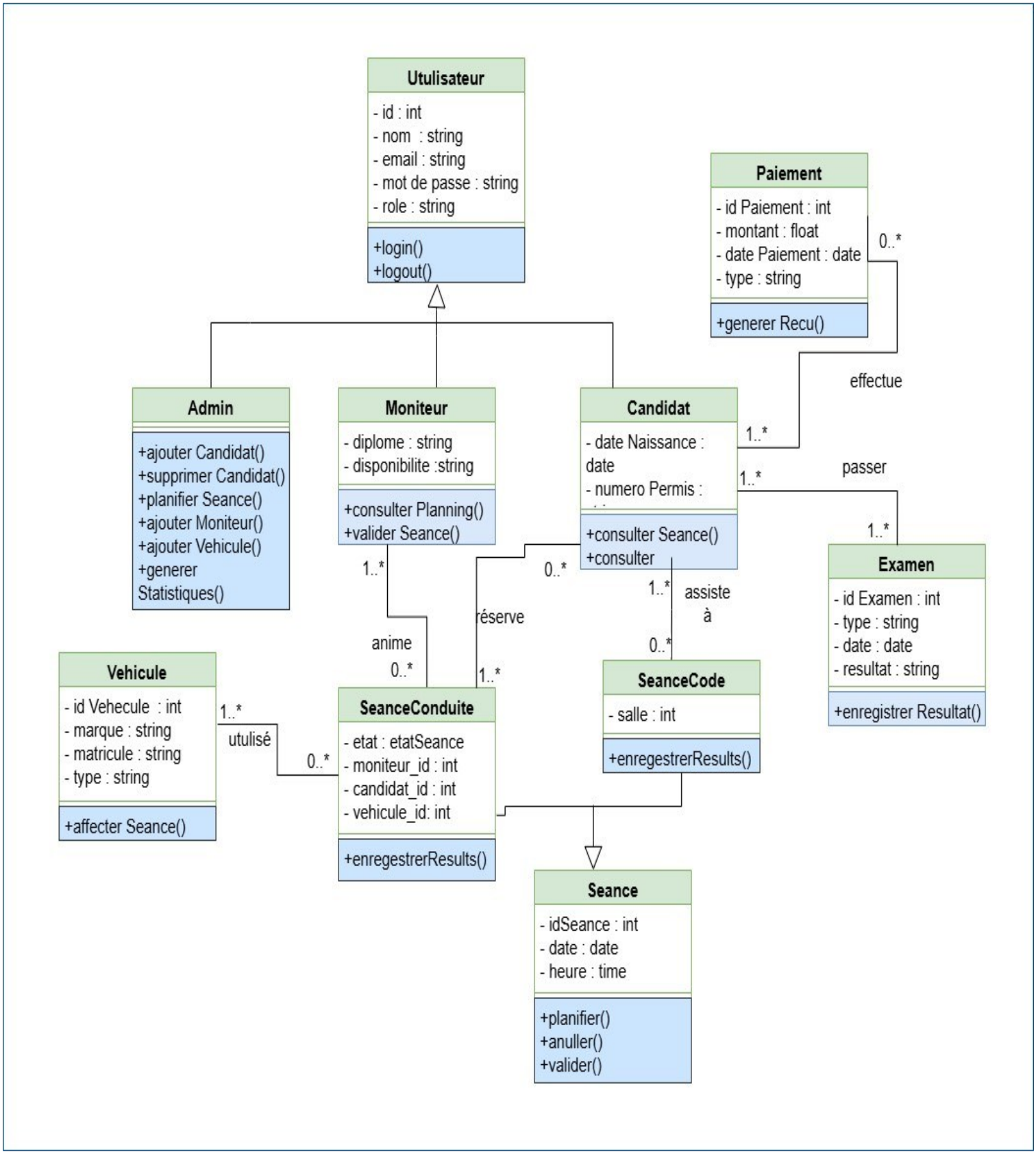


Figure3.1: Diagrammede Classse

- ## 3.3. Diagramme de séquence:

Le diagramme de séquence illustre les échanges de messages entre les différents acteurs et composants du système dans un ordre chronologique. Il met en évidence les échanges de messages entre les acteurs et les composants du système afin de décrire le fonctionnement dynamique de l'application de gestion d'auto-école.

Il offre également une vision claire des différentes étapes d'exécution d'un scénario donné, ce qui facilite la compréhension du comportement du système et permet d'identifier les interactions essentielles entre les différentes entités. **[1] [16]**

➢ 3.3.1 Représentation du Diagramme de Séquence:

3.3.1.1 Diagramme de Séquence N°1: planifier séance

a) Acteurs du système:

Ce scénario implique les éléments suivants:

- ✓ Administrateur: lance le processus de création d'une séance.
- ✓ Système: interface principale qui gère les opérations.
- ✓ Base de données: stocke et vérifie les informations.
- ✓ Moniteur: reçoit la notification de la séance.
- ✓ Candidat: reçoit également la notification.

b) Description du scénario principal:

Dans ce scénario, l'administrateur commence par cliquer sur l'option « Transférer une séance », ce qui déclenche l'affichage du formulaire de création. Il saisit ensuite les informations nécessaires, notamment la date et le type d'heure, ainsi que le type de séance (code ou conduite). Par la suite, il sélectionne les participants concernés, à savoir un moniteur et un candidat, et éventuellement un véhicule selon le type de séance choisi. Le système vérifie alors la disponibilité des ressources dans la base de données. Si la disponibilité est confirmée, la séance est automatiquement créée et enregistrée. Enfin, des notifications sont envoyées au moniteur et au candidat, et une confirmation de réussite est affichée à l'administrateur

- ➢ À la fin de ce scénario la séance est correctement planifiée, les données sont enregistrées dans la base de données, les participants sont informés automatiquement

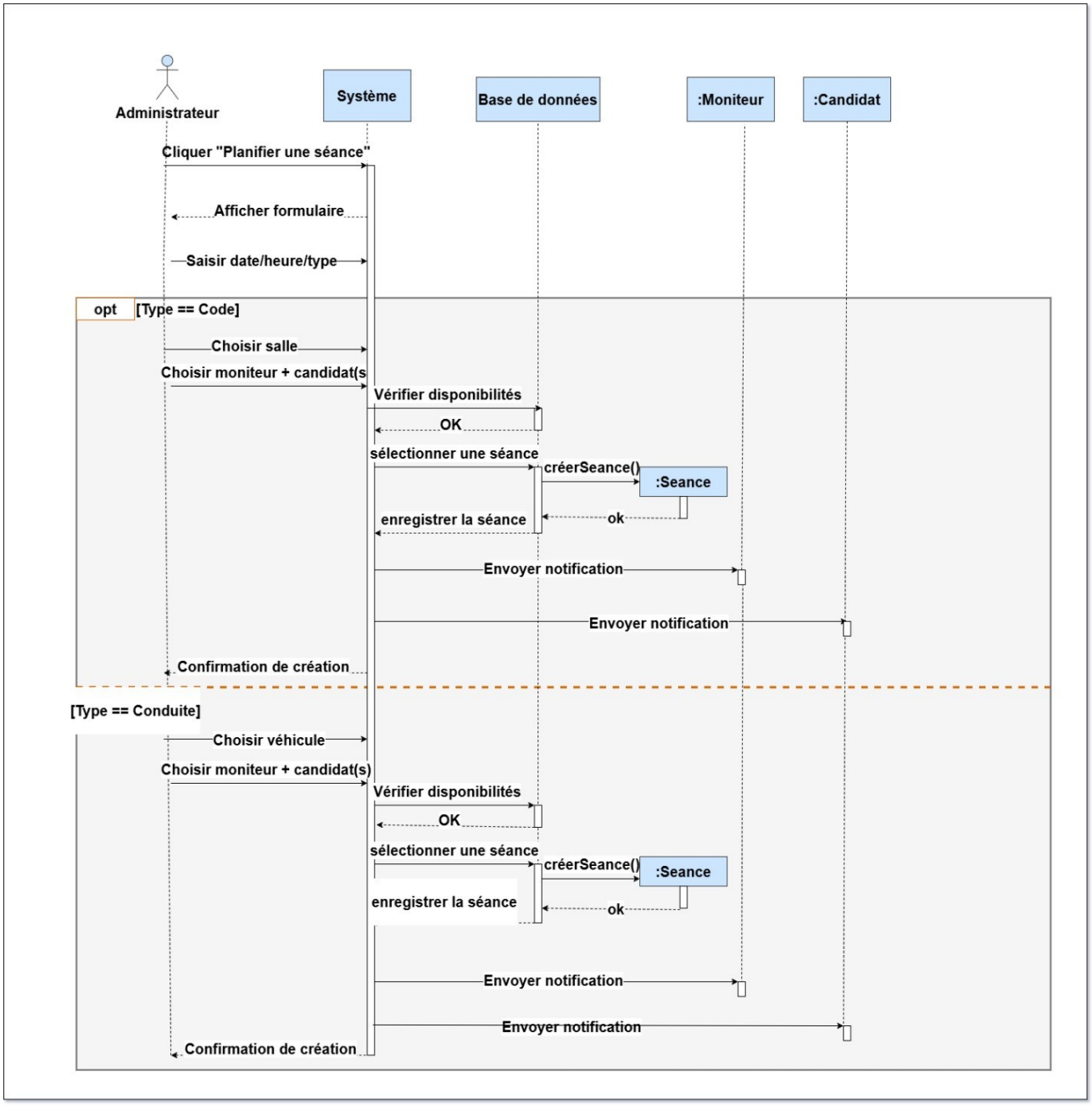


Figure3.2: Diagramme de Séquence du cas d'utulisation planifier séance.

3.3.1.2 Diagramme de Séquence N°2: Paiement

a) Acteurs du Système:

- ✓ Candidat: Consulte son solde, choisit le mode de paiement et règle sa séance.
- ✓ Administrateur: Supervise l'opération, enregistre le cash ou valide la preuve de virement.
- ✓ Système: Gère la logique métier, traite les requêtes et orchestre les flux.
- ✓ Base de données: Stocke les transactions et met à jour les données financières.
- ✓ ServicePaiement: Service externe qui valide les transactions par carte.

b) <u>Description du scénario principal:</u>

Le processus **<u>s'enclenche</u>** par la consultation du montant restant à payer par **l'utilisateur,** ce qui pousse **le système** à récupérer et afficher les données financières depuis **la base de données**. Une fois le mode de règlement sélectionné, **le système** s'adapte selon trois scénarios : pour un paiement en cash, **l'administrateur** valide la réception des fonds et déclenche l'enregistrement ; pour un paiement par carte, le flux transite par un service externe pour validation ; et pour un virement postal**, le candidat** soumet une image de sa preuve de paiement que **l'administrateur** doit vérifier. Quel que soit le chemin choisi**, le système** insère la transaction en **base de données**, actualise le montant restant, génère le reçu final, puis vérifie le solde pour mettre automatiquement à jour le statut du dossier du **candidat** <u>si ce dernier est totalement soldé.</u>

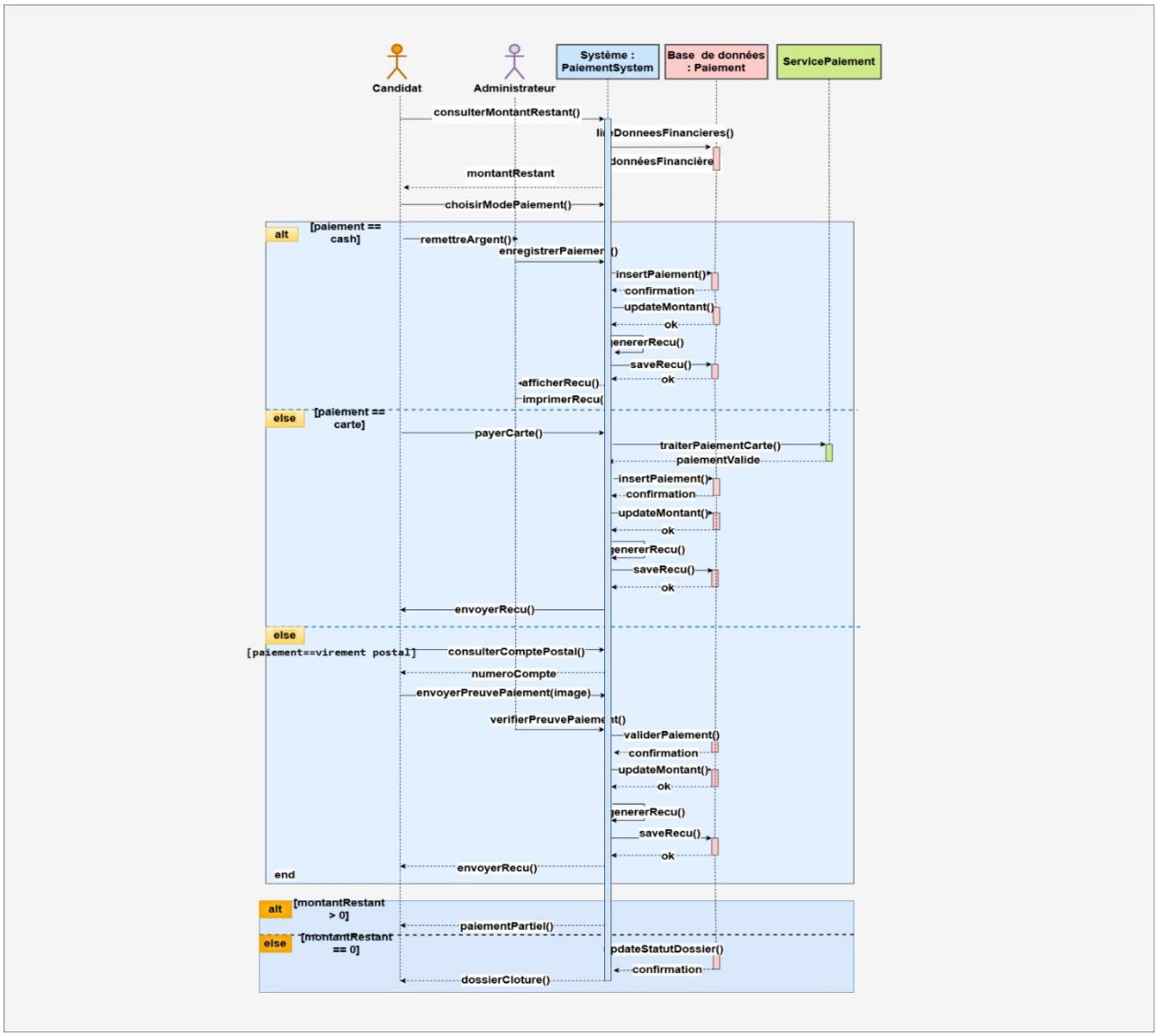


Figure3.3: Diagramme de Séquence du cas d'utulisation Paiement.

• 3.4 Diagramme de d'activité:

diagramme d'activité permet de représenter le déroulement des différentes activités d'un système sous forme d'un enchaînement logique des actions et décisions. Il met en évidence les étapes successives ainsi que les décisions prises au cours d'un processus. Ce type de diagramme facilite la compréhension du fonctionnement global du système de gestion d'auto-école. **[1] [16]**

➢ 3.4.1 Représentation du Diagramme d'activité:

a) Acteurs du systéme:

Le processus est divisé en deux lignes d'eau distinctes pour clarifier les responsabilités:

- ✓ Administrateur: Déclenche le processus, saisit les informations requises et effectue les choix nécessaires.
- ✓ Système: Gère l'affichage des formulaires, effectue les vérifications logiques, enregistre les données et génère les notifications/alertes.

b) Description des étapes (Le Flux):

1) **Initialisation et Authentification:** Le processus commence par le nœud initial (Start), suivi de l'authentification de l'administrateur sur la plateforme.
2) **Demande de planification**: L'administrateur clique sur l'option "Planifier une séance". En réponse, le Système affiche le formulaire de planification.
3) **Saisie et Branchement Conditionnel (Decision Node)**: L'administrateur remplit les données de la séance. Le système évalue ensuite le Type de séance:
4) **Si [Code] (Séance théorique):** L'administrateur doit obligatoirement "Choisir une salle".
5) **Si [Conduite] (Séance pratique)**: L'administrateur doit obligatoirement Choisir véhicule".
6) **Synchronisation et Rôles (Fork & Join)**: Une fois le lieu ou le véhicule défini, le flux se sépare pour permettre l'attribution des ressources humaines indispensables: "Choisir un moniteur" et "Choisir le(s) candidat(s)".
7) **Vérification et Validation (Le Système)**: Le système exécute l'action "Vérifier disponibilités" (vérification des conflits d'horaires pour le moniteur, le candidat, la salle ou le véhicule).
8) **Alternative [Non] (Disponibilité invalide)**: Le système affiche un message d'erreur à l'administrateur pour modifier les créneaux.
9) Alternative [Oui] (Disponibilité valide): Le système procède à l'enregistrement de la séance et crée l'objet "Séance" en base de données.

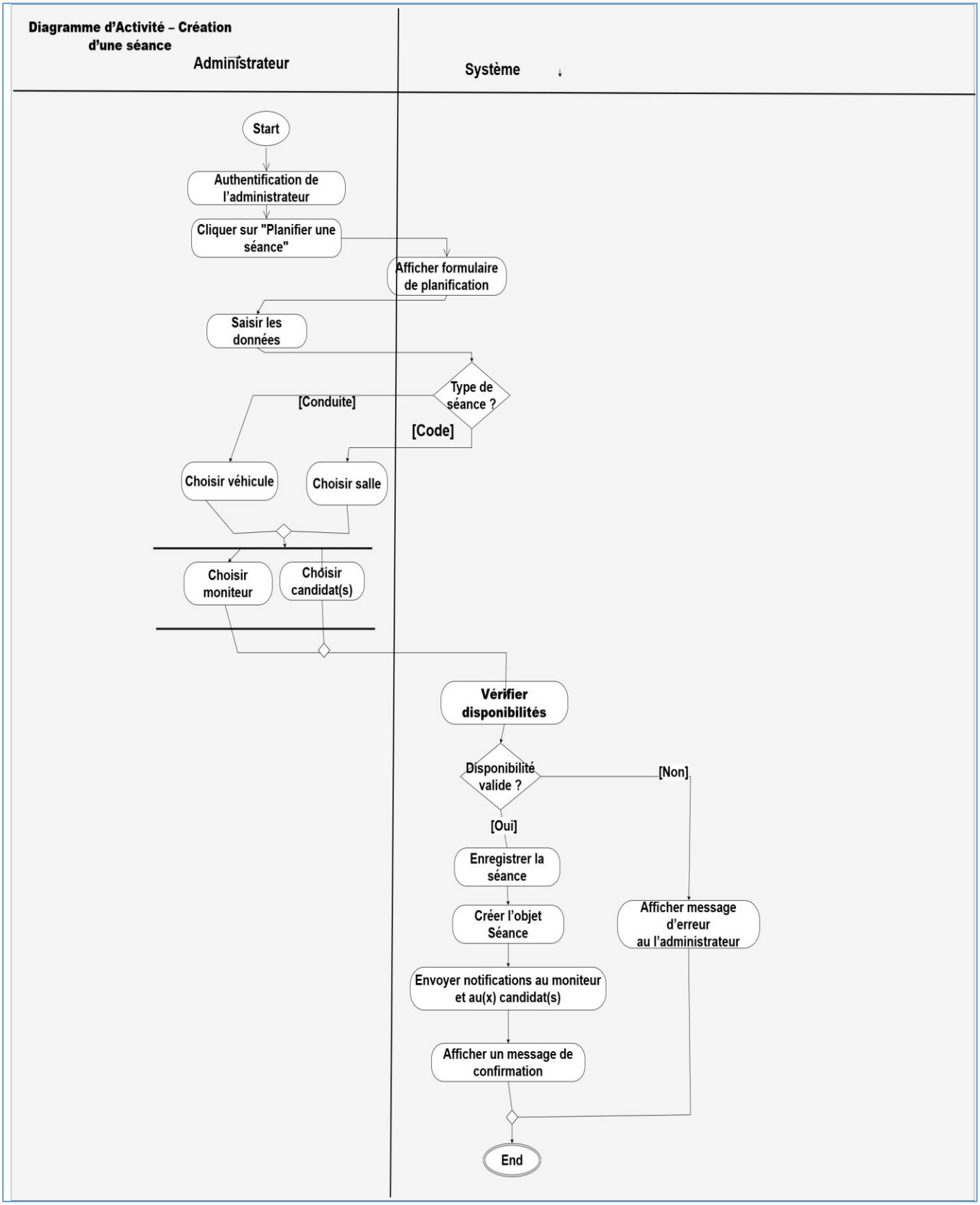


Figure3.4: Diagramme d'activité

3.5 Architecture logique et technique:

L'architecture logique et technique de l'application de gestion d'auto-école permet de définir l'organisation globale du système ainsi que les technologies utilisées pour son développement. **[3]**

3.5.1 Architecture logique:

Le système est composé de plusieurs modules fonctionnels interconnectés tels que la gestion des candidats, des moniteurs, des véhicules, des séances, des paiements et des examens. Chaque module permet d'assurer une fonctionnalité précise au sein de l'application.

3.5.2 Architecture technique:

L'application est développée en utilisant le langage de programmation PHP pour le traitement côté serveur et backend, ainsi que HTML, CSS, javascript pour l'interface utilisateur. La gestion des données est assurée par une base de données MySQL permettant le stockage, la modification et la récupération des informations de manière structurée et sécurisée. Bootstrap a été utilisé afin d'assurer un design responsive et moderne et **infinity free** assurer accès ligne de l'application.

[2] [7] [8] [9] [10] [11] [12] [13]

Le système utilise une architecture client/serveur permettant une meilleure organisation des traitements

3.5.3 Transition vers la phase de Développement:

Dans ce chapitre, nous avons présenté la conception de l'application de gestion d'auto-école à travers le diagramme de classes, le diagramme de séquence ainsi que l'architecture du système. Ces éléments permettent de structurer et de comprendre le fonctionnement global de l'application avant passer à la phase de Développement. **[1]**

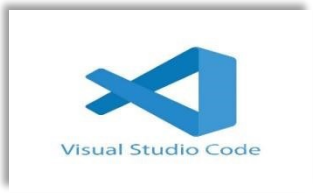
Chapitre 04: Développement:

4.1 Introduction:

Dans ce chapitre, nous présentons la phase de réalisation de notre application web dédiée à la gestion d'une auto-école. Cette phase consiste à transformer la conception réalisée en une application fonctionnelle et interactive. Nous allons détailler les outils utilisés ainsi que les principaux modules développés. **[3] [6]**

4.2 Environnement de développement:

Pour la réalisation de ce projet, nous avons utilisé les outils suivants:



Visual Studio Code: éditeur de code **[14]**



Bootstrap: pour un design responsive et moderne.**[9]**



MySQL: système de gestion de base de données **[11]**



XAMPP: serveur local pour exécuter l'application **[15]**





HTML / CSS / javasc ript /Php: développement de l'interface utilisateur|**[12] [10]**



InfinityFree: pour l'hébergement de notre application web et pour assurer son accès ligne**[13]**

4.3 Présentation générale de l'application:

Avant de détailler les différents modules, il est important de présenter l'interface générale de l'application. Cette dernière est composée d'une page d'accueil permettant aux utilisateurs de accéder aux principales fonctionnalités du système.

4.4 Représentation des interfaces de l'application:

➢ 4.4.1 Page d'accueil:

La page d'accueil constitue la première interface visible par l'utilisateur. Elle présente l'auto-école ainsi que ses services. Elle contient également deux boutons principaux: « Connexion » pour accéder à un compte existant et « S'inscrire » pour créer un nouveau compte.

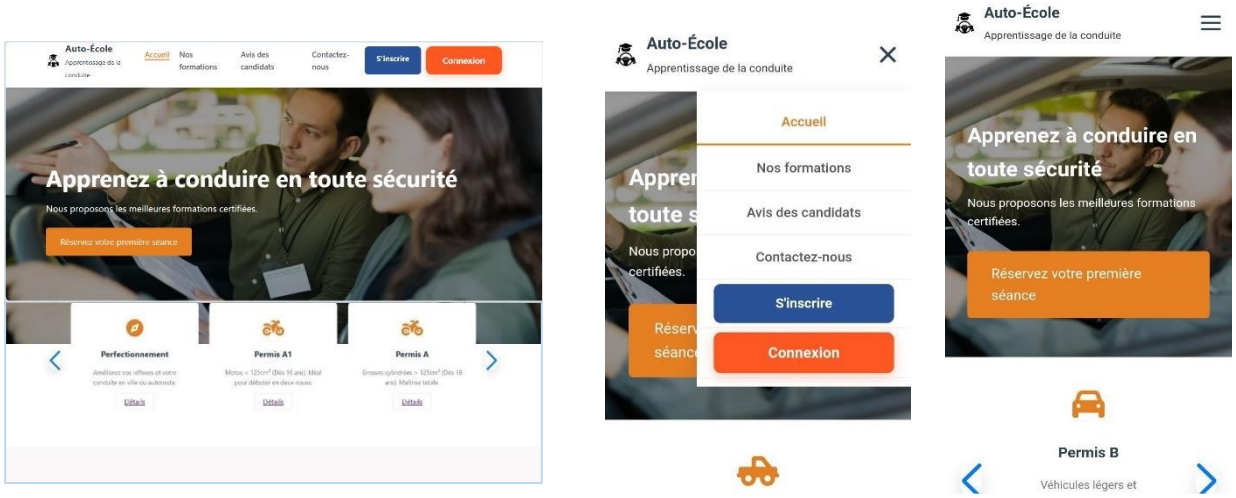
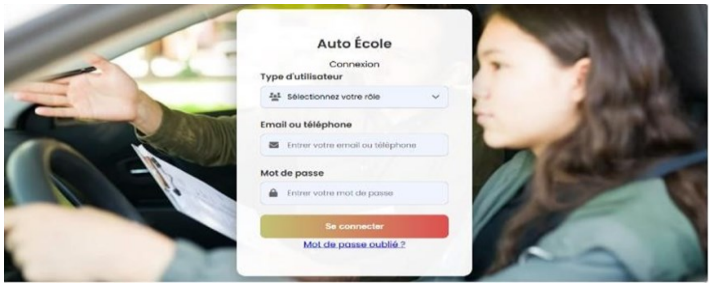


Figure 4.1: Présentation globale de la page d'accueil de l'application auto-école

➢ 4.4.2 Interfaces d'Authentification :(connexion) ou (S'inscrire)

a) **connexion**:cette interface intuitive et ergonomique permet aux utilisateurs (admin, moniteur,candidat)de se connecter selon leurs rôles.



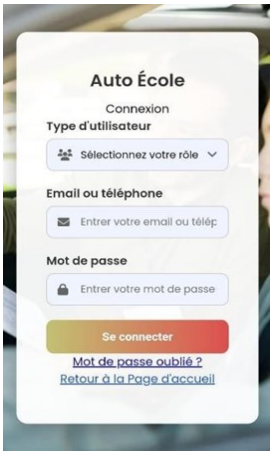


Figure 4. 2: Interface de connexion

b) **S'inscrire:** Cette interface permet aux utilisateurs de créer un nouveau compte en saisissant leurs informations personnelles afin d'accéder à la plateforme en toute sécurité.



Figure 4.3: Interface de S'inscrire

➢ 4.4.3 Interfaces Admin:

L'interface administrateur est la partie principale de l'application. Elle permet à l'administrateur de gérer toutes les fonctionnalités de l'auto-école telles que la gestion des **candidats**, des **moniteurs**, des **véhicules**, des séances et des **paiements** Ces interfaces permettent une gestion rapide et centralisée des informations.

4.4.3.1 Interfaces Tableau de Bord:

Cette interface représente la page principale de l'administrateur. Elle permet d'accéder rapidement aux différentes fonctionnalités du système et consulter les informations importantes.



Figure 4.4: Interface de Tableau de Bord Administrateur

4.4.3.2 Gestion des candidats:

Cette interface permet de gérer les informations des candidats, notamment l'ajout, la modification et le suivi des données administratives.

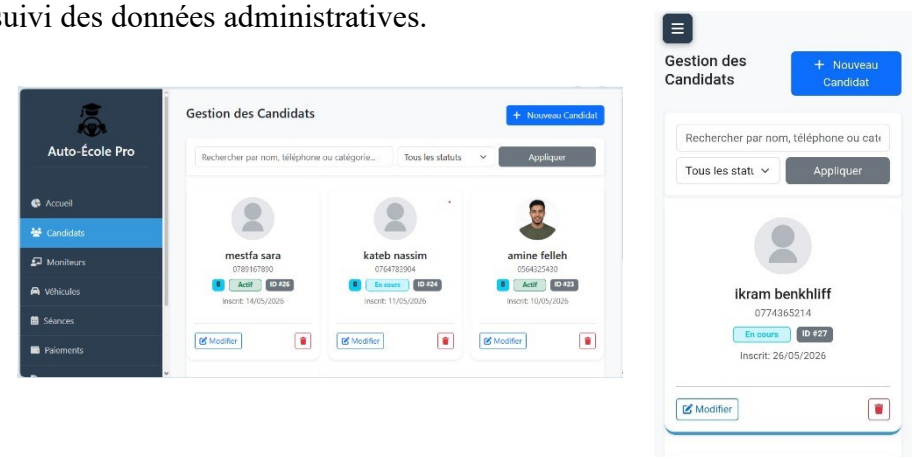


Figure 4.5: Interface de Gestion des candidat

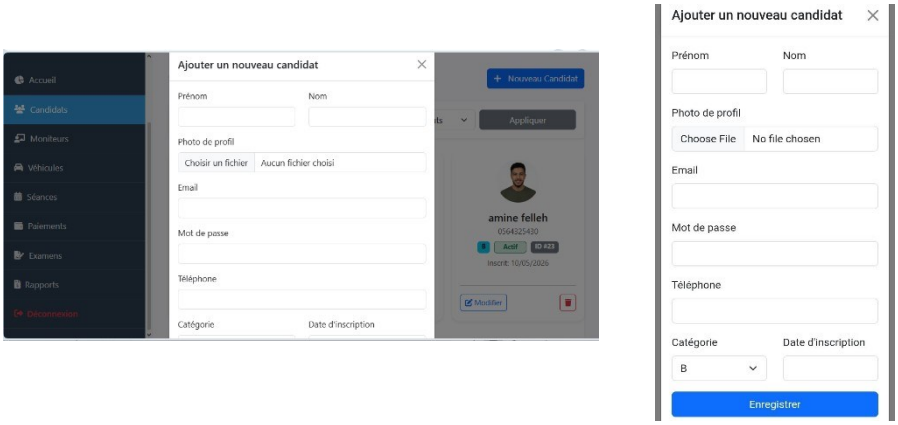


Figure 4.6: Formulaire d'ajout d'un candidat

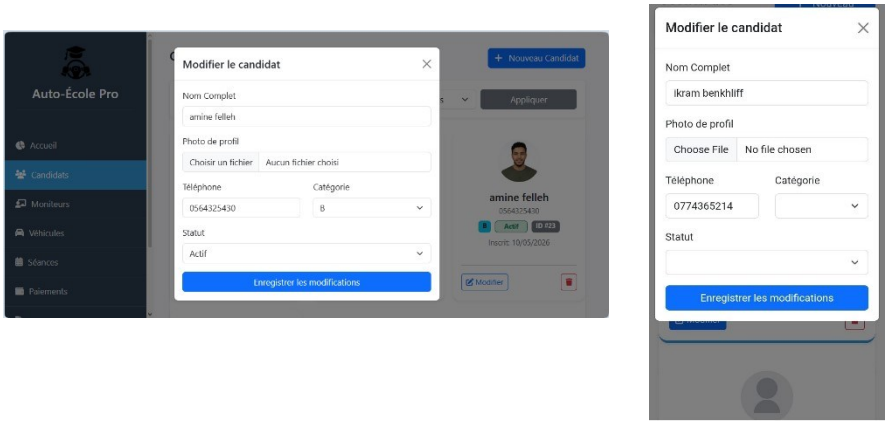


Figure 4.7: Formulaire Modifier d'un candidat

4.4.3.3 Gestion moniteurs:

Cette interface permet à l'administrateur de gérer les informations des moniteurs, ainsi que le suivi de leurs activités et plannings.

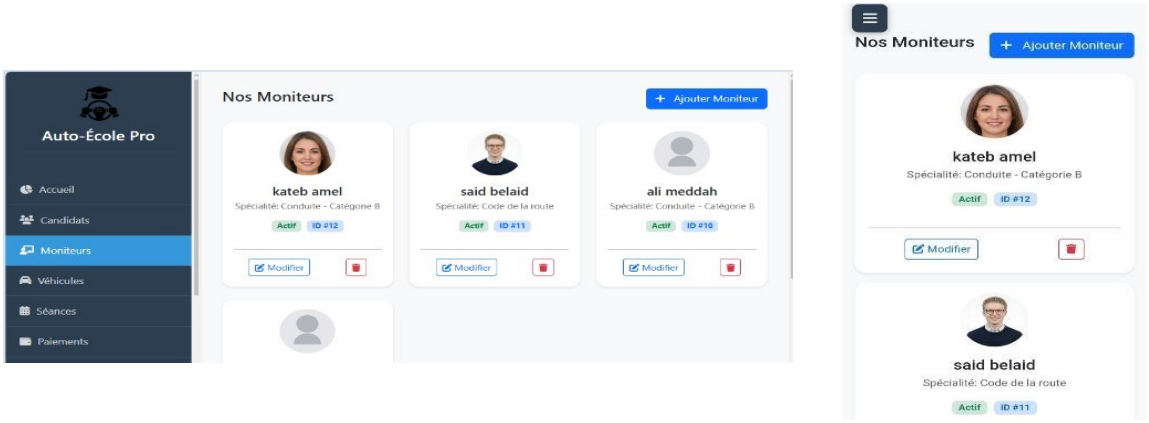


Figure 4.8: Interface de gestion des moniteurs

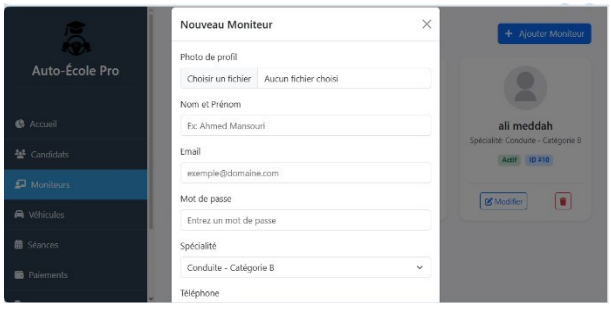


Figure 4.9: Formulaire d'ajout d'un moniteur

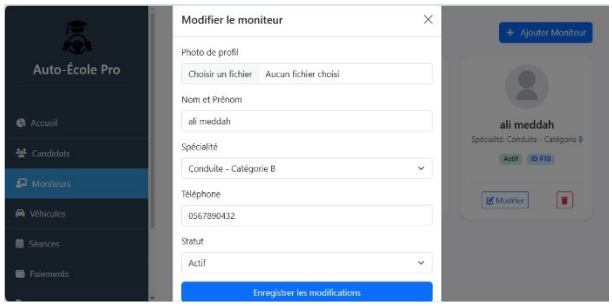


Figure 4.10: Formulaire Modifier d'un moniteur

4.4.3.4 Gestion véhicules:

Cette interface permet de gérer les véhicules utilisés dans les séances de conduite ainsi que leurs

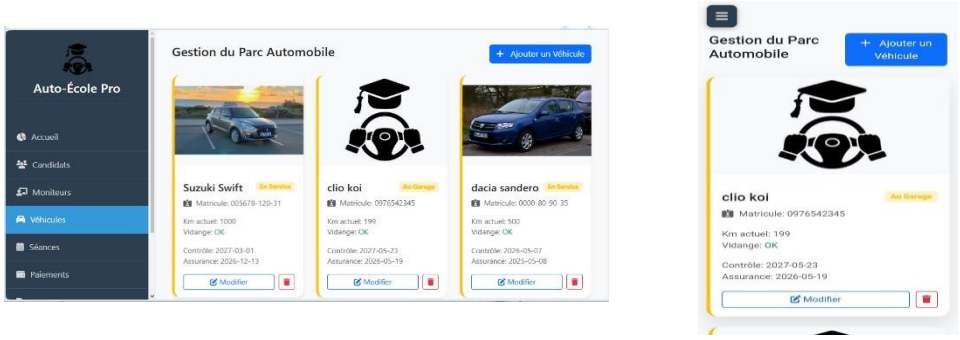


Figure 4.11: Interface de gestion des véhicules informations.

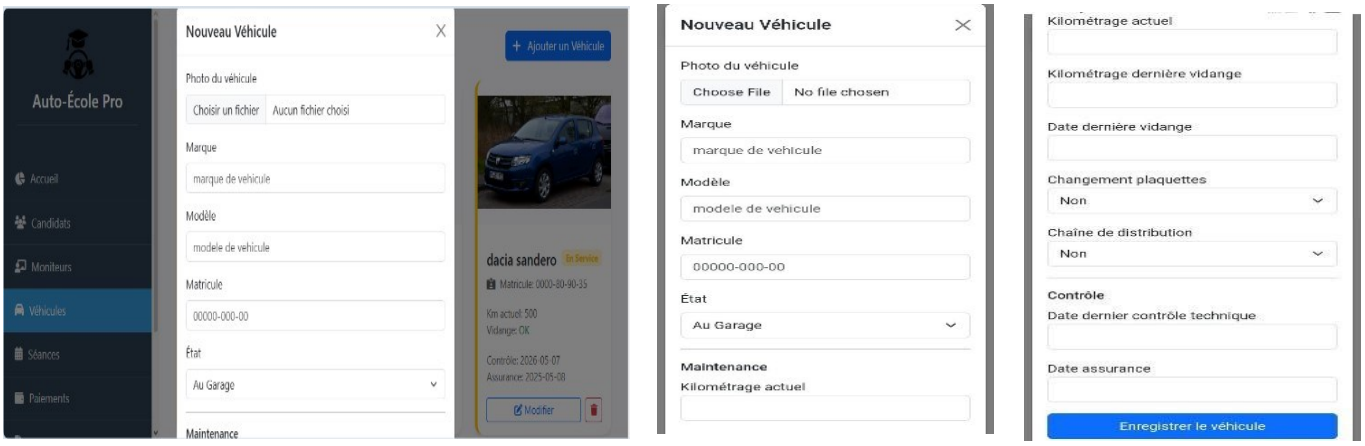


Figure 4.12: Formulaire d'ajout d'un nouvelle véhicule

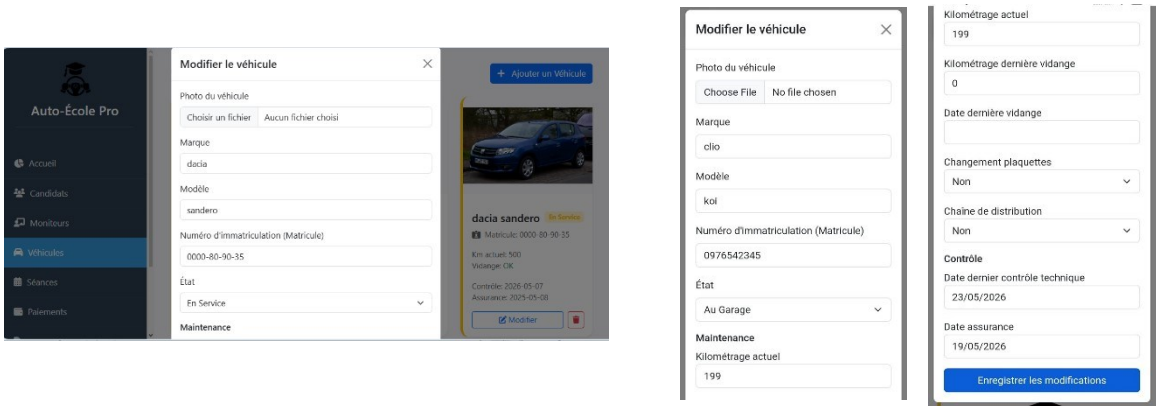


Figure 4.13: Formulaire Modifier d'une véhicule

4.4.3.5 Gestion des séances:

Cette interface permet d'organiser les séances de conduite et gérer les plannings des candidats et des moniteurs.

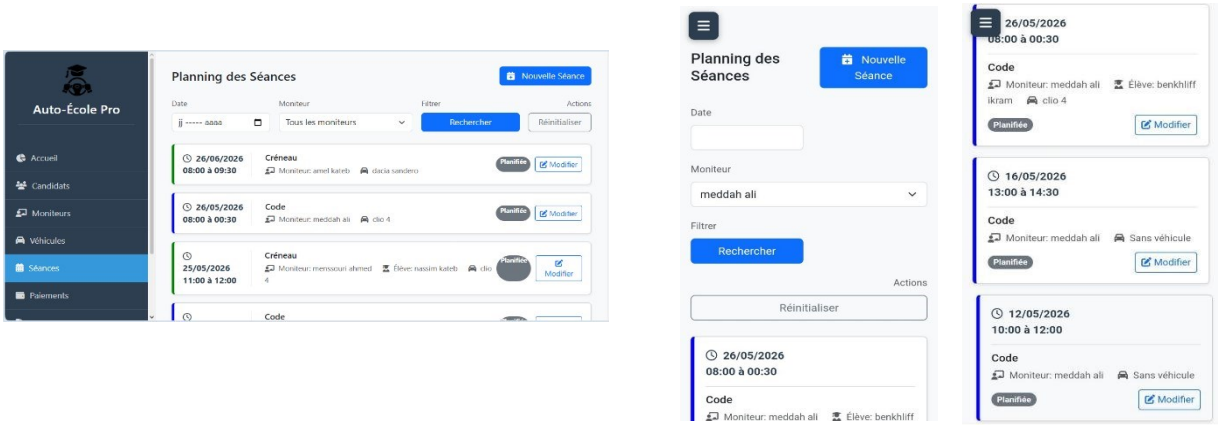


Figure 4.14: Interface de Gestion des séances

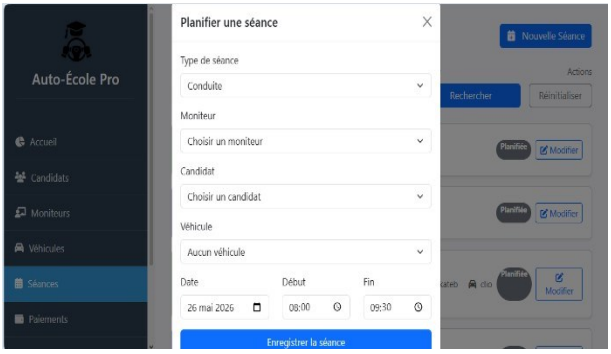


Figure 4.15: Interface de planifier une séance

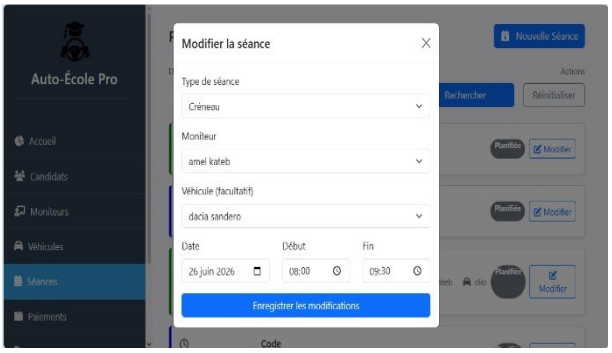
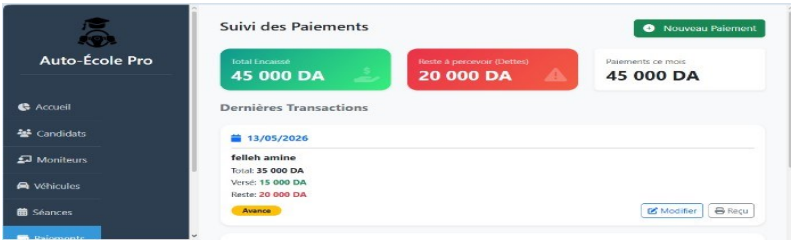


Figure 4.16: Interface de Modifier une séance

4.4.3.6 Gestion des Paiements:

Cette interface permet d'assurer le suivi des paiements effectués par les candidats et consulter les informations financières.



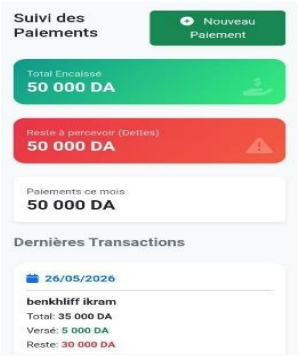


Figure 4.17: Interface de gestion des Paiements

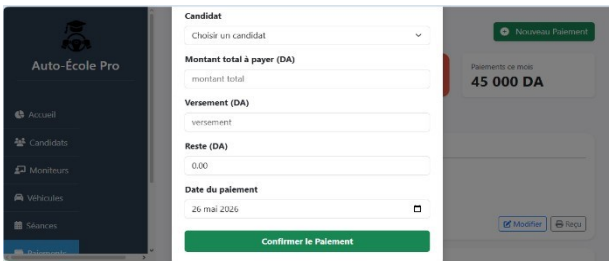








Figure 4.18: Interface d'ajout nouveau paiement **Figure 4.19**: Interface de modifier paiement

4.4.3.7 Gestion des Examens:

Cette interface permet de gérer les examens et consulter les résultats des candidats.

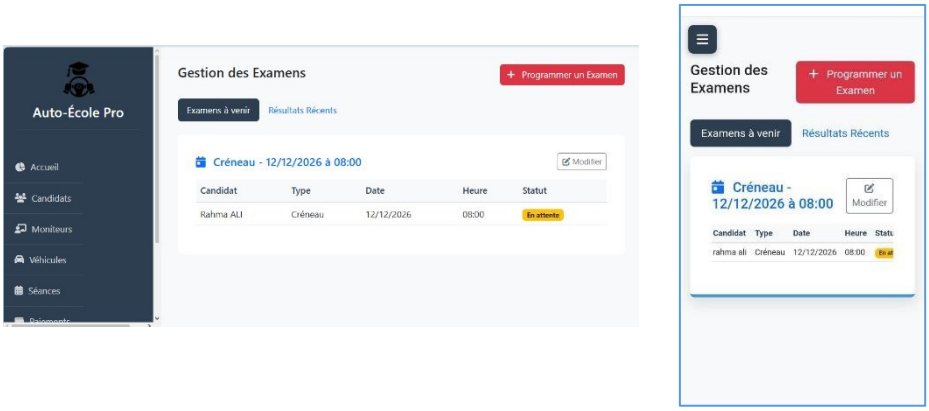


Figure 4.20: Interface de gestion des Examens

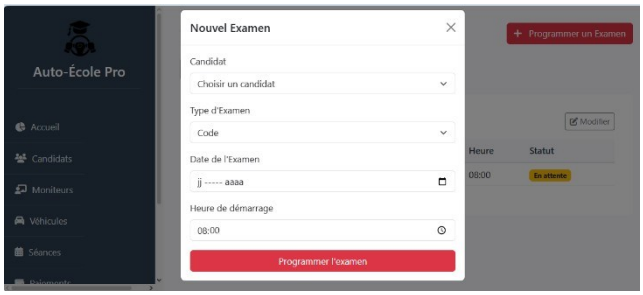


Figure 4.21: Interface de programmer Examen

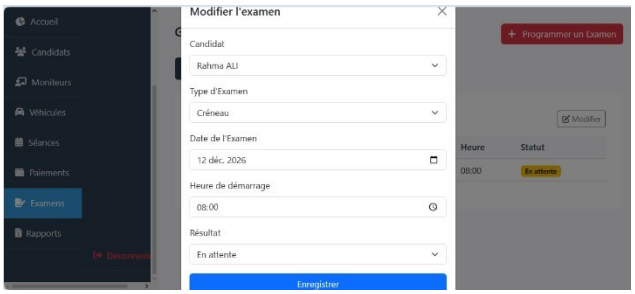


Figure 4.22: Interface de modifier Examen

4.4.3.8 Consulter des Rapports d'examens:

Cette interface permet de consulter différentes statistiques et rapports liés aux activités.



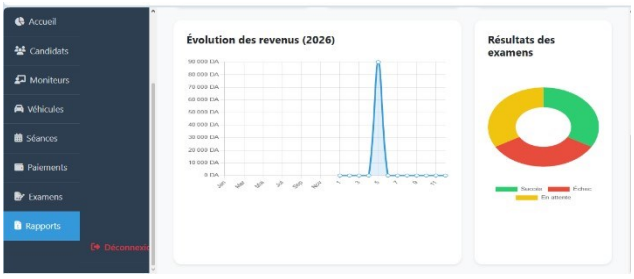


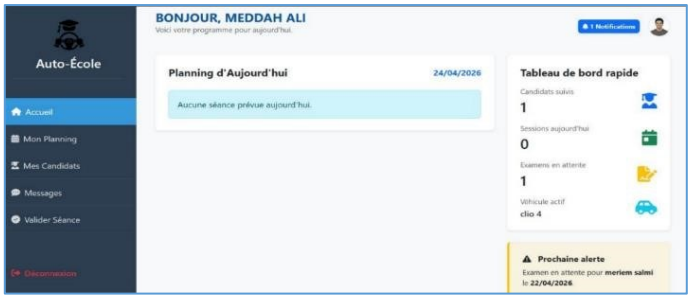
Figure 4.23: Interface de Rapports

➢ 4.4.4 Interfaces Moniteur:

Cette interface est destinée au moniteur de l'auto-école. Elle lui permet de consulter les séances de conduite planifiées ainsi que les informations relatives aux candidats. Elle offre également la possibilité de suivre son planning et de gérer les différentes activités qui lui sont attribuées dans le système.

4.4.4.1 Accueil Moniteur:

Cette interface représente la page d'accueil du moniteur. Elle permet d'accéder rapidement aux principales fonctionnalités du système.



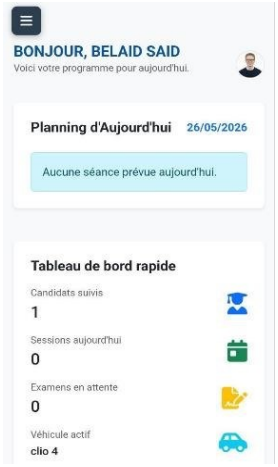
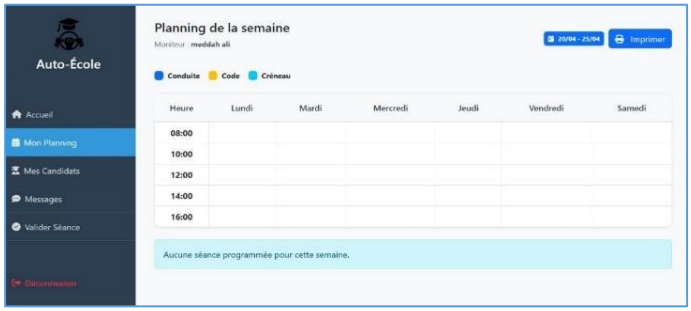


Figure 4.24: Interface Accueil Moniteur

4.4.4.2 Emplois du temps (Planning):

Cette interface permet au moniteur de consulter les séances de conduite programmées ainsi que son emploi du temps



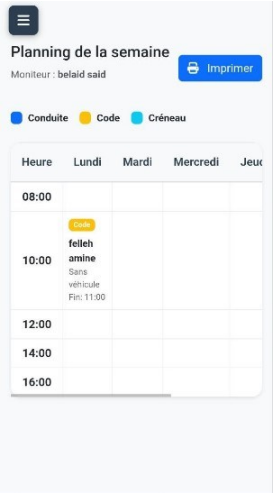


Figure 4.25: Interface planning des séances

4.4.4.3 Consultations des candidats:

Cette interface permet au moniteur de consulter les informations des candidats et suivre leur progression.

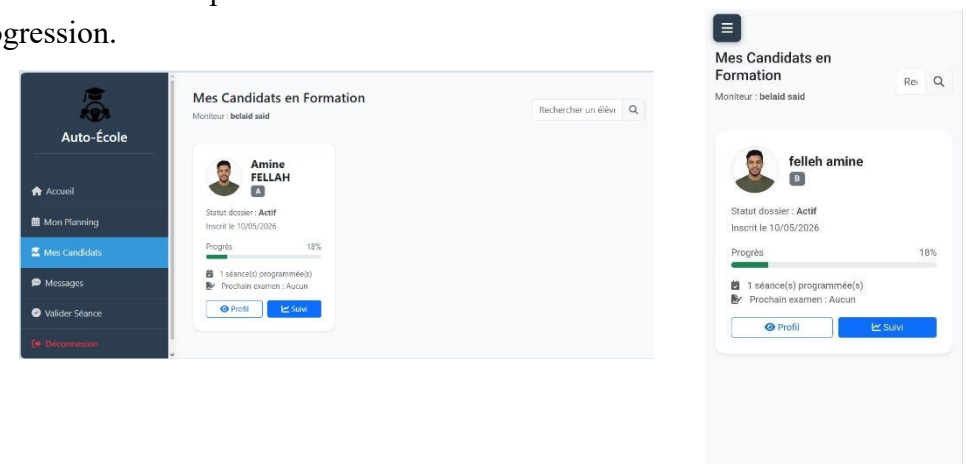


Figure 4.26: Interface des Candidats

4.4.4.4 Messagerie:

Cette interface permet au moniteur de communiquer avec les candidats et l'administration à travers le système de messagerie.

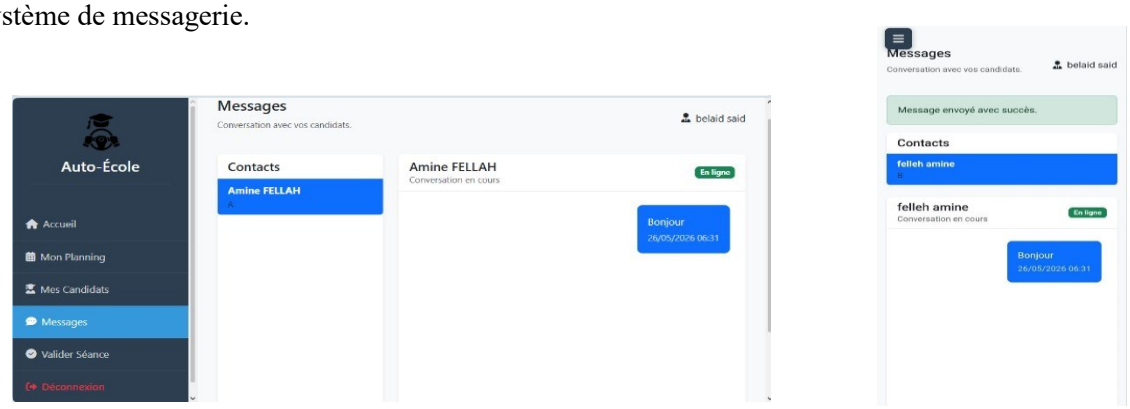


Figure 4.27: Interface des message

4.4.4.5 valider séance:

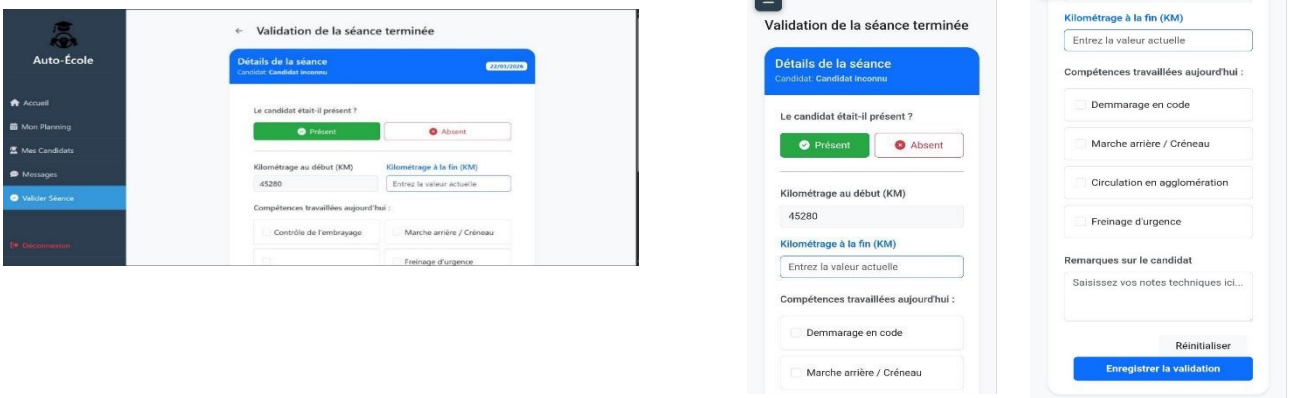


Figure 4.28: Interface de validation d'une séance

4.4.5 Interfaces Candidat:

Cette interface est destinée au candidat de l'auto-école. Elle lui permet de consulter ses informations personnelles, ses séances de conduite planifiées ainsi que son état de paiement. Elle offre également un accès aux différentes fonctionnalités liées à son suivi au sein du système.

4.4.5.1 Accueil Candidat:

Cette interface représente la page d'accueil du candidat. Elle permet d'accéder rapidement aux principales fonctionnalités de l'application.

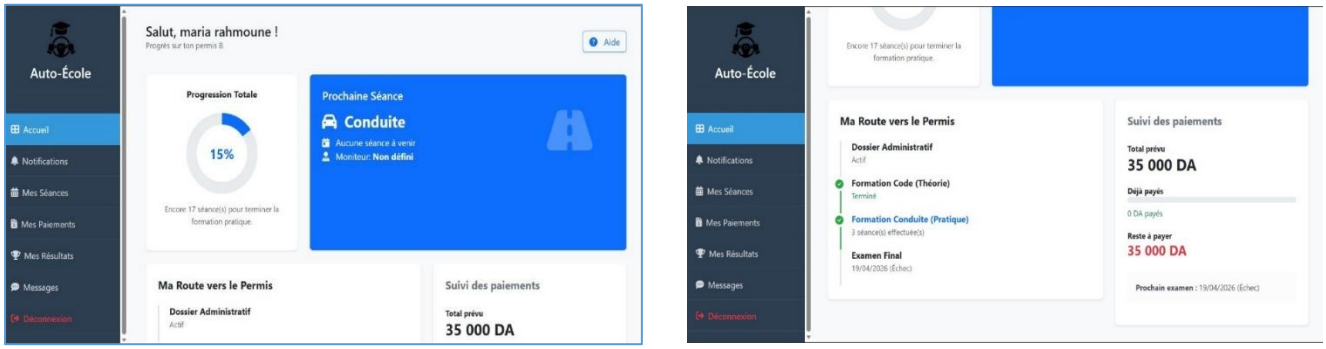


Figure 4.29: InterfaceAccueil Candidat

4.4.5.2 Notifications

Cette interface permet au candidat de consulter les différentes notifications et informations envoyées par l'administration.

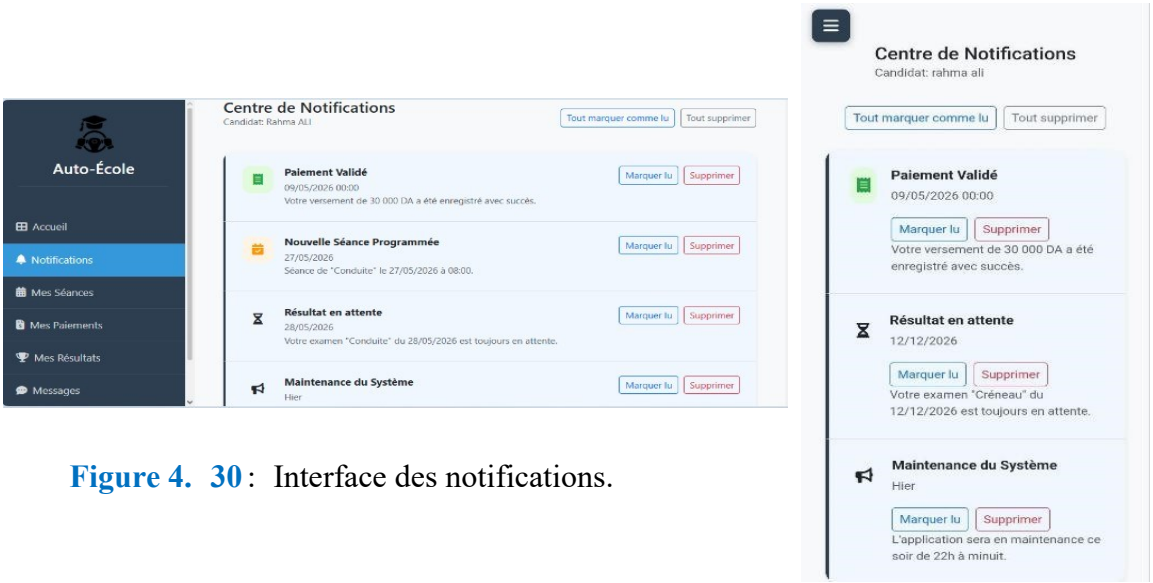


Figure 4. 30 : Interface des notifications.

4.4.5.3 Consultation des séances:

Cette interface permet au candidat de consulter les séances de conduite programmées ainsi que leurs détails.



Figure 4.31: Interface des mes séances

4.4.5.4 Consulter Paiements

Cette interface permet au candidat de consulter les paiements effectués et suivre sa situation financière.





Figure 4.32: Interface des Paiements

4.4.5.5 Consulter Mes Résults:





Figure 4.33: Interface Mes Résultas

Chapitre 05: Test et Validations:

• 5.1 Introduction:

Cette phase représente une étape importante dans le développement de l'application. Elle permet de vérifier le bon fonctionnement des différentes fonctionnalités du système avant sa mise en service. Les tests réalisés ont pour objectif de détecter les erreurs et garantir la fiabilité de l'application développée **[5].**

• 5.2 Objectif des tests:

Les tests effectués permettent de vérifier que les différentes fonctionnalités de l'application répondent correctement aux besoins des utilisateurs. Ils assurent également la stabilité du système ainsi que le bon déroulement des opérations réalisées au sein de l'application.**[5]**

• 5.3 Tests réalisés:

Fonction testée	Résultat attendu	Résultat obtenu
Connexion	Accès autorisé au système	Réussi
Ajout candidat	Enregistrement du candidat	Réussi
Gestion des paiements	Paiement enregistré	Réussi
Consultation planning	Affichage correct des séances	Réussi
Gestion des examens	Enregistrement des résultats	Réussi

• 5.4 Difficultés rencontrées:

Durant la réalisation du projet, plusieurs difficultés ont été rencontrées, notamment l'organisation de la base de données, la gestion des relations entre les différentes tables ainsi que certaines erreurs liées au développement des interfaces et du code PHP.

• 5.5 Solutions apportées:

Afin de résoudre les problèmes rencontrés, plusieurs solutions ont été mises en place. Nous avons amélioré la structure de la base de données, corrigé les erreurs du code source et optimisé l'organisation des interfaces afin d'assurer un meilleur fonctionnement du système. **[6]**

• 5.6 Conclusion:

Les différents tests réalisés ont permis de vérifier le bon fonctionnement de l'application et de valider les principales fonctionnalités du système. Les résultats obtenus montrent que l'application répond correctement aux besoins de gestion d'une auto-école.

•

- ## Conclusion Générale:

Au terme de ce projet, nous avons réussi à concevoir et développer une application web de gestion d'auto-école permettant d'améliorer l'organisation et le suivi des différentes activités administratives et pédagogiques. Cette application offre plusieurs fonctionnalités importantes telles que la gestion des candidats, des moniteurs, des véhicules, des séances de conduite, des paiements ainsi que des examens.

La réalisation de ce projet nous a permis de mettre en pratique les connaissances acquises durant notre parcours universitaire, notamment dans les domaines du développement web, des bases de données et de la conception UML. Nous avons également appris à analyser les besoins des utilisateurs, concevoir une solution adaptée et résoudre plusieurs problèmes techniques rencontrés durant le développement.

Les différents tests effectués ont montré que l'application répond correctement aux besoins de gestion d'une auto-école et assure un fonctionnement fiable et efficace. Ce projet représente ainsi une expérience enrichissante aussi bien sur le plan technique que méthodologique.

- Comme perspectives, cette application pourrait être améliorée en ajoutant de nouvelles fonctionnalités telles qu'un système de paiement électronique en ligne, des notifications automatiques par e-mail ou SMS, ainsi qu'une version mobile afin de faciliter davantage l'accès aux services proposés. [2][3][5]

Bibliographie & Webographie:

[1] Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, 2nd Edition, Addison-Wesley, 2005.

[2] Luke Welling, Laura Thomson, PHP and MySQL Web Development, 5th Edition, AddisonWesley Professional, 2016.

[3] Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015.

[4] Philippe Kruchten, The Rational Unified Process: An Introduction, Addison-Wesley, 2004.

[5] Roger S. Pressman, Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 8th Edition, McGraw-Hill Education, 2014.

[6] Paul Deitel, Harvey Deitel, Internet & World Wide Web: How to Program, 5th Edition, Pearson, 2011.

[7] Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2016.

[8] Silberschatz Abraham, Henry F. Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill, 2011.

[9] Bootstrap Team, "Bootstrap Documentation": getbootstrap.com.consulté le 17/03/2026 à 9:38

[10] PHP Documentation Team, "PHP Official Documentation": php.net. consulté le 10/04/2026

[11] MySQL Documentation Team, "MySQL Reference Manual", disponible sur: dev.mysql.com

[12] Mozilla Developer Network (MDN), "HTML, CSS and JavaScript Documentation", disponible sur: developer.mozilla.org. consulté le 11/04/2026 à 13:29

[13] InfinityFree, "Free Web Hosting Services": infinityfree.com. consulté le 11/05/2026 à 21:34

[14] Visual Studio Code Team, "Visual Studio Code Documentation", disponible sur code.visualstudio.com. consulté le 11/04/2026 à 18:19

[15] Apache Friends, "XAMPP Documentation": apachefriends.org. consulté le 22/04/2026 à 7:28

[16] OMG (Object Management Group), "UML Specification".: omg.org. consulté le 08/03/2026