

**REPUBLIQUE DE GUINNEE**

***TRAVAIL-JUSTICE-SOLIDARITE***

**Année universitaire : 2022 - 2023**

**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR, DE LA RECHERCHE SCIENTIFIQUE ET DE L’INOVATION**

**UNIVESITE DE GAMAL ABDEL NASSER DE CONAKRY (UGANC)**

**CENTRE INFORMATIQUE**

**DEPARTEMENT DEVELOPPEMENT ET FORMATION**

**Rapport de projet au choix**

Pour l’obtention du diplôme de Licence Professionnelle au Centre Informatique

**THÈME**

Conception et réalisation d’un système informatisé d’une application Web **GESTION AUTO-ECOLE**

Etudiantes : **MAKERA KABA** et

**MARLIATOU DIALLO**

**Encadreur : Dr Ibrahim Khalil Touré**

# Dédicace

A ma maman (Hadja Fanta Kaba)

Pour tout ce qui tu fais pour moi, tout amour que tu m’as donnée, des risques que tu as pris à ma cause, des prières quotidiennes et des dépenses, je ferai mon mieux de rester une question de fierté à tes yeux en ne jamais te décevoir « Maman Jolie » Que ce modeste travail soit l’exaucement de tes vœux tant formulés.

A mon très cher oncle (Cheick Koutoubou Kaba)

Je vous remercie de tout votre appui inconditionnel et je vous consacre ce mémoire dans l’espoir d’un avenir meilleur.

A mes très chères sœurs (Diacka Kaba, Djénebou Kaba,Aicha Kaba)

Je vous dédie ce travail, merci pour tout ce que vous avez fait pour moi, conseils, des dépenses scolaires, que le tout puissant vous récompense et je vous souhaite un bel avenir chez vous.

A mon très cher frère (Abdoulaye Kaba)

Merci d’avoir été avec moi dans ce domaine très vaste de l’informatique, d’être mon confident et ma complice de vie il m’a été facile de commencer avec vous, et je vous donne cette thèse pour toute votre aide au démarrage jusqu’à maintenant.

A mes amis les plus chers (Mamoudou faro, Lacine Camara, lonceny keita, lancine keita, Aboubacar sidiki keita,Aminata diané) Je vous dédie ce travail, il me touche au cœur, sachant que vous êtes spécial pour moi.

A ma très chères meilleure amie (Mamadamah Soumah) Merci pour tout ton appui, alors je te dédie ce fameux travail chérie ainsi qu’à notre deuxième maman(Yarie Camara). Et je vous aime de tout mon cœur.

**Remerciements**

Alhamdoulilaye Tout d’abord, je remercie et loue le Seigneur le Créateur, Allah, qui a voulu que je sois humain et qui m’a donné l’opportunité d’être l’un de ceux qui reconnaissent ses bienfaits Allah est certes miséricordieux.

Ensuite, il serait très étrange de commencer ce travail sans pour autant remercier des personnes qui se tiennent debout pour m’accompagner dans ce long travail, aux premiers rangs, je remercie mes deux géniteurs, Lamine et Fanta Kaba, qui ont été la raison de ma naissance, ils m’ont aidé pendant ces moments difficiles.

Je suis reconnaissant à mon oncle Koutoubou Kaba, qui m’a fourni un logement et a été mon tuteur pendant la fac. Et qui a été un père pour moi.

Je remercie mes sœurs qui ont toujours voulu que je réussisse et s’en sont souciées, les sœurs qui m’ont fait passer devant tout le monde, les sœurs dont je n’ai jamais manqué de rien pendant mes études, je vous apprécie vraiment Diacka Kaba et Djenabou Kaba, je remercie également mes tantes Domani Kaba, Oumou Kaba, Diacka Kaba et Halimatou Savané, je me souviens de vous depuis l’enfance.

Je serais ingrate de ne pas remercier mes tuteurs du primaire à l’université, ce sont : Madame Mariame Touré, qui m’a a pris à écrire bien et à faire face au public lors des exposés à l’école Yassafè Koura, M. Therna Mara et M. Odilon Famany Sagnon de primaire, de ‘Amitié’, M. Mamadi Keita, M. Souleymane Tairé Diallo et M. Thierno Mouguita Bah, sans oublier mes encadreurs du SEMYG, à l’Université ici, M.Mohamed conté , M. Moucatar Barry, Dr koukouma. et Dr Ibrahime kallil Touré.

Je voudrais également remercier les personnes formidables qui m’ont fait découvrir la programmation et qui m’ont si bien guidé, un grand merci à Mon frère Abdoulaye kaba

Enfin, je voudrais remercier tous mes amis Yasmine jawad kouyaté, Lonceny keita, emaky.

**Glossaire**

|  |  |
| --- | --- |
| Reporting | Les rapports |
| PHPUnit | Est un Framework open source de tests unitaires dédié au PHP |
| Digitalisation | Numérisation, transformation |
| Appareils Electroniques | Projecteurs, cameras …… |
| Recouvrement | Suivi des locations en cours |
| Planning poker | Est unetechnique d’estimation basé sur le consensus. Les équipes Agile du monde entier utilisent le Planning Poker pour estimer leurs backlogs produits. |
| Sprint back log | Le Sprint Backlog est unplan par et pour les développeurs. |

**Abréviations**

|  |  |
| --- | --- |
| SGBD /BD | Système de gestion de base données |
| PHP | HyperText Preprocessor |
| MVC | Model Vue Controller |
| UGANC | Université Gamal Abdel Nasser de Conakry |
| UML | Modèle de Langage Unifié |

**Résumé**

Dans le cadre de ce projet, en tant que projet de fin d’études, il s’agit de construire une application web pour DEBO, Auto-école spécialisée dans la formation des apprenants(candidat) pour l’obtention d’un permis de conduite, dont le Fondateur souhaite voir toutes les activités connexes et être informé chaque fois qu’une action est effectuée, puis de recevoir des rapports quotidiens.

Par conséquent, pour y parvenir, nous avons choisi les étapes suivantes :

* Identifier les exigences fonctionnelles et non fonctionnelles de Fondateur,
* Etudes techniques et conception détaillée,
* Mise en œuvre/réalisation du projet

Pour faire ce fameux travail nous avons utilisé PHP comme langage de programmation, nous avons utilisé donc les Framework et bibliothèques suivants : Laravel, Livewire et Bootstrap, nous avons utilisé le langage UML et Access pour modéliser le système sans oublié la méthode de travail utilisée, la méthode dite agile et une de ses sous-méthodes (Scrum).

Le test de l’application est fait avec PHPUnit

**Abstract**

As part of this project, as a graduation project, it is a question of building a web application for DEBO, a company specializing in the rental of cars, apartments, audiovisual equipment, ceremonial equipment and many other finished products, of which the Founder wishes to see all the related activities and be informed each time an action is carried out, and then receive daily reports.

Therefore, to achieve this, we have chosen the following steps:

* Identify the functional and non-functional requirements of Founder,
* Technical studies and detailed design,
* Implementation/realization of the project

To do this famous work we used PHP as a programming language, so we used the following Framework and libraries: Laravel, Livewire and Bootstrap, we used the UML language to model the system without forgetting the working method used, the so-called agile method and one of its sub-methods (Scrum).

The test of the application is done with PHPUnit

**LISTE DES FIGURES**

[Figure 1 : Logo de L’entreprise 13](#_Toc108171181)

[Figure 2 : Schéma de fonctionnement 18](#_Toc108171182)

[Figure 3 : cas d’utilisation de super admin 27](#_Toc108171183)

[Figure 4 : cas d’utilisation d’Admin 28](#_Toc108171184)

[Figure 5 : cas d’utilisation d’Employé 29](#_Toc108171185)

[Figure 6 : Diagramme de séquence de Super d’Admin 30](#_Toc108171186)

[Figure 7 : diagramme de séquence d’Admin 31](#_Toc108171188)

[Figure 8 : diagramme de séquence d’Employé 32](#_Toc108171189)

[Figure 9 : diagramme de classe 34](#_Toc108171191)

[Figure 10 : Interface de connexion 38](#_Toc108171192)

[Figure 11 : Interface de page d’accueil 39](#_Toc108171193)

[Figure 12 : Interface de liste des utilisateurs 40](#_Toc108171195)

[Figure 13: Interface d’ajout des utilisateurs 41](#_Toc108171197)

[Figure 14 : Interface de mis à jour des utilisateurs 42](#_Toc108171199)

[Figure 15 : Interface de liste des types d’articles 43](#_Toc108171201)

[Figure 16 : Interface d’ajout d’un type d’article. 44](#_Toc108171202)

[Figure 17 : Interface d’ajout des propriétés de types d’articles et liste 44](#_Toc108171204)

[Figure 18 : Interface de liste des articles. 45](#_Toc108171206)

[Figure 19 : Interface d’ajout d’articles 46](#_Toc108171207)

[Figure 20 : Interface de mis à jour d’un article. 47](#_Toc108171208)

[Figure 21 : Interface de liste des clients. 48](#_Toc108171209)

[Figure 22 : Interface d’ajout d’un nouvel client. 49](#_Toc108171210)

[Figure 23 : Interface de modification d’un client. 50](#_Toc108171211)

[Figure 24 : Interface d’effectuer une location. 51](#_Toc108171212)

[Figure 25 : Interface de formulaire d’une location. 52](#_Toc108171213)

[Figure 26 : Interface de liste des locations en cours. 53](#_Toc108171214)

[Figure 27 : Interface du paiement effectué d’une location 54](#_Toc108171215)

[Figure 28 : Interface d’enregistrer le paiement d’une location. 54](#_Toc108171216)

**Table des matières**

[1 Chapitre 1 : Généralité 12](#_Toc107254400)

[1.1 Cadre général 13](#_Toc107254401)

[1.1.1 Présentation de l’entreprise 13](#_Toc107254402)

[1.1.2 Services 13](#_Toc107254403)

[1.1.3 Conditions pour le service de formation 14](#_Toc107254404)

[1.1.4 Service de paiement 14](#_Toc107254405)

[1.2 Problématique 14](#_Toc107254406)

[1.3 Objectifs 15](#_Toc107254407)

[1.3.1 Objectif général 15](#_Toc107254408)

[1.3.2 Objectif spécifique 15](#_Toc107254409)

[1.4 Méthodologie 16](#_Toc107254410)

[1.4.1 Introduction 16](#_Toc107254411)

[1.4.2 Approche d’agile 16](#_Toc107254412)

[1.4.3 Fonctionnement de Scrum 17](#_Toc107254413)

[1.4.4 Les avantages et inconvenants de la méthode d’agile 18](#_Toc107254414)

[1.5 ETUDE DE L’EXISTANT ET ANALYSE DES BESOINS 20](#_Toc107254415)

[1.5.1 Introduction : 20](#_Toc107254416)

[1.5.2 Domaine de gestions 20](#_Toc107254417)

[1.6 Spécification des besoins 21](#_Toc107254418)

[1.6.1 Spécification des besoins fonctionnels 21](#_Toc107254419)

[1.6.2 Spécification des besoins non fonctionnels 21](#_Toc107254420)

[2 Chapitre 2 : Etude détaillée - Modélisation 23](#_Toc107254421)

[2.1 Introduction 24](#_Toc107254422)

[2.2 Diagrammes de cas d’utilisation 24](#_Toc107254423)

[2.2.1 Identification des acteurs 24](#_Toc107254424)

[2.2.2 Identification des cas d’utilisation 24](#_Toc107254425)

[2.2.3 Diagramme de cas d’utilisation des tâches des utilisateurs 25](#_Toc107254426)

[2.3 Diagrammes de séquences 29](#_Toc107254427)

[Pour Admin 30](#_Toc107254430)

[Pour Utilisateur 31](#_Toc107254431)

[31](#_Toc107254432)

[2.4 Diagramme de classe 31](#_Toc107254433)

[3 Chapitre 3 : Conception et développement / Etude technique 33](#_Toc107254434)

[3.1 Introduction 34](#_Toc107254435)

[3.2 Environnements de travail : 34](#_Toc107254436)

[3.3 QUELQUES INTERFFACES DE L’APPLICATION 36](#_Toc107254437)

[3.4 Les Tests 53](#_Toc107254438)

**Introduction**

Dans ce monde des startups où l’évolution des entreprises ne fait que s’accélérer du jour au lendemain, les entrepreneurs font face à de nombreux défis au sein de l’entreprise en termes de fonctionnement et de gestion, face à ce sujet, les développeurs d’applications (informaticiens) se mettent au service des entrepreneurs pour répondre à leurs besoins et leur font savoir à quel point la digitalisation en informatique peut être efficace.

Dans ce cas précis, comme notre cible, il s’agit d’une entreprise de formation (auto-école) qui forme tout candidat(apprenant) voulant obtenir une catégorie de permis de conduite. Le responsable d’auto-école « Debo » conscient des différentes difficultés rencontrées dans sa gestion et conscient aussi de l’importance des applications informatique, nous a fait appel afin de répondre à tous ses besoins d’automatisation

La structure de mise en œuvre de notre projet est la suivante : le chapitre 1 présente la généralité comme un objectif général et un objectif spécifique pour préciser les différents besoins d’une application (fonctionnels, non fonctionnels et techniques), le chapitre 2 présente des recherches/modélisations détaillées, le chapitre 3 présente la mise en œuvre et le développement, présente une solution finale, telle que des résultats, puis tire une conclusion générale, résumant notre travail, et envisageant ses perspectives.

**Chapitre 1 : Généralité**

## Cadre général

### Présentation de l’entreprise



Figure 1 : Logo de L’entreprise

L’entreprise Debo est une entreprise spécialisée dans le domaine de formation des candidats susceptible de passer un examen afin d’obtenir un permis de conduite, créée en 2007, dont le chef de l’entreprise se nomme Mohamed Barry,

Cette entreprise existe il y a longtemps et est responsable de la formation de nombreux citoyens guinéens détenteur de permis de conduite dans les circulations.

### Conditions pour le service de formation

Dans auto-école Debo, voici les conditions que Mr Barry exige pour effectuer une formation

* Les renseignements de chaque candidats(apprenant) sont enregistrés ayant suivis une formation ;
* Les informations de chaque Utilisateur sont enregistrées ayant effectué l’inscription ;
* Les informations de chaque moniteur sont enregistrées ayant donné un cours (théorique ou pratique)
* Le paiement d’une formation est fait en des tranches ;

**Service de paiement**

Les paiements des formations peuvent être effectués sur :

* Un compte Bancaire UBA
* Orange money
* Mobile money ou bien
* En Liquidité

## Problématique

La gestion dans une entreprise est actuellement l’un des problèmes auxquels chaque entreprise est confrontée, de sorte que les entrepreneurs se demandent toujours cette question, comment organiser les choses dans leurs entreprises, afin d’assurer une bonne gestion au sein.

La suivie des inscriptions, de la formation des apprenants (candidats) est réaliser manuellement, en particulier utilisation des fiches de renseignement des candidats ainsi que des registres.

La planification des séances de formation se fait verbalement, en cas de changement de la date ou de l’horaire de la séance suite par exemple à l’absence du moniteur ou un autre évènement la secrétaire doit contacter les apprenants (candidats) par téléphone pour signaler le changement.

En cas d’intempérie les apprenants (candidats) contacteront souvent les moniteurs pour confirmer les séances de conduite.

Après un examen, le candidat (l’apprenant) doit se déplacer pour contacter à l’auto-école ou contacter la secrétaire par téléphone pour connaître le résultat de l’examen, de même pour un moniteur s’il veut connaître le résultat de ces apprenants (candidats).

Tous ces problèmes nous ont motivé pour développer une application web en laravel qui permet d’un côté d’aider la secrétaire et le gérant de l’auto-école à la saisie, l’enregistrement et la diffusion de l’information sur les apprenants (candidats) et les moniteurs, planifier les séances de formation et les examens, et d’un autre côté offrir aux apprenants (candidats) et aux moniteurs un moyen pour consulter les informations en ligne sans se déplacer à l’auto-école.

**Objectifs**

Dans cette section, nous vous présenterons les deux objectifs liés à notre projet en cours dont :

* Objectif général
* Objectif spécifique

**Objectif général**

L’objectif global de ce projet consiste à concevoir une application web qui automatise toute la gestion d’auto-école Debo qui sont :

* Gestion des apprenants(candidats) ;
* Gestion des moniteurs ;
* Gestion des cours ;
* Gestion des véhicules ;
* Gestion des examens ;
* Gestion des payements ;
* Gestion des dépenses ;
* Gestion Tarifs ;

**Objectif Spécifique**

Dans cette partie, la solution doit d’abord assurer l’ensemble de fonctionnalités ci-après pour les utilisateurs de l’application.

**Admin**

* Enregistrer les fonctionnalités ;
* Faire la mise à jour des fonctionnalités ;

Supprimer fonctionnalités ;

**Utilisateur(employés)**

* Enregistrer les informations des candidats qui effectuent des formations ;
* Mentionner les paiements effectués par un candidats ;

## Méthodologie

**Introduction**

Le développement des logiciels est devenu un moteur majeur des innovations technologiques et son rôle s’est accru pour constituer une pièce maîtresse du puzzle du développement économique. Tout cela signifie qu’il faut réfléchir davantage aux besoins et à la méthodologie de développement idéale pour minimiser les coûts et les risques afin d’obtenir un produit de qualité qui répond aux besoins et aux attentes des utilisateurs

Parmi ces besoins à la méthodologie de développement les plus connues sont :

Méthodologie en Cascade et celle d’Agile

D’après l’étude complète sur notre projet, ce dernier répond à suivre celle de la méthodologie agile plus précisément la méthode Scrum.

**Méthode Scrum**

Les méthodes agiles sont des méthodologies essentiellement dédiées à la gestion de projets informatiques. Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent notamment d’impliquer l’ensemble des collaborateurs ainsi que le client dans le développement du projet.

**Approche d’agile**

Depuis une quinzaine d’années, la majorité des développements des logiciels s’appuie sur des méthodes dites “agiles”. Sous cette bannière se regroupent plusieurs méthodes basées sur un développement itératif et incrémental, dans lequel la recherche de solutions aux problèmes rencontrés s’appuie sur la collaboration de pair à pair. Elle promeut des réponses rapides et flexibles, une planification des tâches adaptatives dans des laps de temps très courts permettant une très grande réactivité. Les approches plus classiques tels que cycle en V ou le modèle en cascade sont souvent mises en œuvre pour les projets répondant à un imposant cahier des charges. Le client et le prestataire s’entendent alors sur un contrat et si le projet prend du retard ou ne répond pas à tous les besoins à la date butoir, des pénalités sont alors facturées au prestataire. De la même manière si le client s’aperçoit en cours de route que certains besoins ont été omis dans le cahier des charges, il devra alors renégocier avec le prestataire le contrat. Les premières méthodes agiles apparues sont EVO (Evolutionary Project Management) (1976), RAD (développement rapide d’applications) (1991), puis DSDM, la version anglaise du RAD (1995). Les trois méthodes agiles désormais les plus utilisées sont : la méthode Kanban, issue de la méthode industrielle Lean, la méthode Scrum publiée en 2001 par Ken Schwaber et Jeff Sutherland, et la méthode XP (Extreme programming) publiée en 1999 par Kent Beck.

**Pourquoi Scrum**

Scrum est l’une des méthodologies la plus utilisée parmi les méthodes Agile existantes.

Elle est la plus documentée et supportée.

Le bon nombre de ces ressources sont accessibles gratuitement, on pourrait pratiquement parler d’un standard Agile.

Un autre atout important : Scrum est simple à comprendre.

Les experts de Scrum et ses fondateurs, le décrivent comme un « cadre de travail permettant de répondre à des problèmes complexes et changeants tout en livrant de manière productive et créative des produits de la plus grande valeur possible ».

Il s’appuie sur trois piliers :

* **Transparence** : Les aspects importants du processus doivent être visibles à ceux qui sont responsables des retombées. La transparence requiert la définition d’un standard commun pour ces aspects afin que les observateurs partagent une compréhension commune de ce qui est observé.
* **L’inspection** : Les utilisateurs de Scrum doivent fréquemment inspecter les artéfacts Scrum et l’état d’avancement par rapport à un objectif de Sprint (Sprint Goal) afin de détecter les écarts indésirables.
* **L’adaptation** : Si un inspecteur détermine qu’un ou plusieurs aspects du processus dérivent hors des limites acceptables, et que le produit qui en résulte sera inacceptable, le processus ou le matériel utilisé par le processus doit être ajusté. Un ajustement doit être fait dès que possible afin de minimiser le risque d’autres dérives.

### Fonctionnement de Scrum

La méthode Scrum s’appuie sur le découpage des projets en itérations encore nommées « sprints ». Un sprint peut avoir une durée qui varie généralement entre deux semaines et un mois. Avant chaque sprint, les tâches sont estimées en temps et en complexité à l’aide de certaines pratiques comme le « planning poker », une manière ludique de chiffrer la complexité des tâches ou évolutions à l’aide de cartes à l’instar du célèbre jeu dont le nom est repris. Ces estimations permettent à la fois de planifier les livraisons, mais aussi d’estimer le coût de ces tâches auprès du client. Les fonctionnalités (encore appelées « user stories ») qui font l’objet d’un sprint constituent ce que l’on appelle un « sprint back log » du produit éventuellement livrable à la fin du sprint. Il est nécessaire de distinguer le sprint back log du « Product back log » qui lui correspond à l’ensemble des fonctionnalités attendues pour le produit sur l’ensemble des sprints. La méthode Scrum est aussi caractérisée par une « mêlée » quotidienne, encore appelée « morning » ou « stand-up », dans laquelle les collaborateurs (chefs de projets, développeurs et responsables fonctionnels) indiquent tour à tour les tâches qu’ils ont effectuées la veille, les difficultés rencontrées et enfin ce sur quoi ils vont poursuivre leur travail le jour suivant. Cela permet d’évaluer l’avancement du projet, de mobiliser des ressources là où cela est le plus nécessaire, mais aussi de venir en aide aux collaborateurs rencontrant des difficultés lorsque celles-ci ont déjà été rencontrées auparavant par d’autres membres de l’équipe.

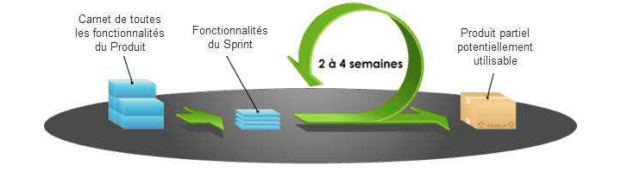


Figure 2 : Schéma de fonctionnement

### Les avantages et inconvenants de la méthode d’agile

**Les Avantages :**

La méthode Agile présente de nombreux avantages :

**Flexibilité :** Avec l’approche Agile, vous travaillez en flux tendu. Vous vous occupez des tâches qui sont importantes à un instant T. Vous êtes donc capable de vous adapter rapidement à votre environnement.

**Feedback :** La notion de feedback est au cœur de la méthodologie Agile. À la fin de chaque sprint, vous livrez votre travail et récupérez les feedbacks de vos clients ou de vos utilisateurs puis vous les incorporez dans le sprint suivant ce qui permet de coller au plus près à leurs besoins.

**Compétitivité :** Quand vous travaillez en mode Agile, vous livrez régulièrement de nouvelles fonctionnalités, de nouveaux produits et de nouvelles améliorations ce qui vous rend compétitif et attractif auprès de vos clients et utilisateurs.

**Qualité :** La livraison fréquente de projets, les tests et l’intégration permanente de feedbacks permettent de développer un produit final de qualité. Pendant tout le cycle de production, le produit est testé et confronté à la réalité.

**Les Inconvenants :**

L’approche Agile a aussi quelques inconvénients :

Manque de documentation : Étant donné que le scope des projets change au gré des besoins et des feedbacks, il est difficile de tenir une documentation à jour. La documentation d’un produit Agile est donc généralement moins travaillée et détaillée.

**Gestion des demandes :** Intégrer continuellement les demandes des clients et utilisateurs vous rend plus réactif, mais cela implique aussi de devoir gérer un plus grand volume de demandes entrantes ce qui peut compliquer la gestion de projet.

**Manque de prévisibilité :** Quand vous travaillez en mode Agile, vous ne savez pas toujours à quoi ressemblera le résultat final. Difficile donc de prévoir le coût, le temps et les ressources nécessaires.

**Conclusion :**

La méthode Agile est une méthode de gestion de projet qui consiste à décomposer vos projets en une suite de petits objectifs atteignables et sur lesquels vous travaillez lors de sprints.

**Voici les étapes à suivre pour appliquer la méthode Agile Scrum :**

***Étape 1*** : Définissez le cadre du projet en lisant les objectifs et exigences

***Étape 2*** : Préparez le backlog en listant toutes les demandes de vos clients/utilisateurs

***Étape 3 :*** Travaillez sur les tâches de votre sprint

***Étape 4*** : Récoltez les feedbacks de vos clients/utilisateur

***Étape 5 :*** Recommence

## ETUDE DE L’EXISTANT ET ANALYSE DES BESOINS

### Introduction :

La phase d’analyse vise à analyser le système existant, à identifier les acteurs qui interagissent avec le système et à définir les besoins fonctionnels et non fonctionnels du système, alors cette partie se présente comme étant charge de développer une application web.

Nous commencerons par une étude préliminaire des exigences de l’application (Spécifications) qui constitue toute la gestion de la société., puis nous détaillerons les besoins fonctionnels et non fonctionnels de ce système

### Domaine de gestions

**Gestion apprenants(candidats)**

* Constituer une base de données pour les candidats (enregistrer leurs informations dans une table) ;

Améliorer l’offre des candidats (Réduction de prix pour les candidats favoris) ;

**Gestion des moniteurs**

Faire enregistrer les moniteurs selon le type de cours enseigner et pendant un intervalle de temps bien précis.

**Gestion des véhicules**

* Constituer une base de données pour les véhicules (Les différent types, model, marque, couleur et autre) ;

Gérer le stock (disponibilité des véhicules ou l’indisponibilité dans le stock) ;

**Gestion de payements**

* Selon la catégorie de permis des payements sont effectuer en des tranches de trois (3) ;

Des tarifs aussi seront payer en fonction du type de cours ainsi que du nombre de jours d’apprentissage ;

**Gestion des dépenses**

Gérer les dépenses selon son type (Personnel, véhicule, école et autre)

**Gestion des examens**

* Pour obtenir son permis après la formation, candidat va devoir effectuer un texte.

Gestion des cours

Gérer les cours selon leurs types

## Spécification des besoins

### Spécification des besoins fonctionnels

Les besoins fonctionnels expriment un ensemble d’actions que doit effectuer le système en réponse à une demande.

* Authentification :

C’est l’étape essentielle de toutes les fonctionnalités, il s’agit de fournir le courrier et d’entrer le mot de passe pour pouvoir utiliser les services concernés, Cela signifie que chaque utilisateur doit d’abord s’authentifier avant de pouvoir accéder à l’application pour effectuer les tâches.

* Enregistrement :

Les informations de chaque utilisateur seront enregistrées,

Les informations de chaque candidat seront enregistrées ayant demandé une formation,

Chaque cour effectuée sera enregistrée et chaque délai dépasser sera mentionné, tarifié puis enregistré

* Tableau de bord :

Notre application doit fournir un espace, qui permettra d’afficher les statiques à savoir un diagramme qui permet de visualiser chaque tâche à effectuer au sein.

### Spécification des besoins non fonctionnels

Ce sont des besoins qui permettent d’améliorer la qualité de service de l’application comme :

**Besoin d’utilisabilité :** le système doit être simple à utiliser

**Besoin de performances :** le temps de traitement doit être optimal

**Besoin de disponibilité :** l’application doit être disponible lors de chaque opération à moins que l’application soit en cours de maintenance

**Convivialité et Ergonomie des interfaces** :

L’application doit fournir une interface simple et élégante pour l’utilisateur afin de faciliter l’exploitation des services de l’application

**Portabilité, maintenance et évolution** :

L’application doit être multiplateforme c’est à-dire fonctionnelle sur n’importe quel système d’exploitation. Le code doit être bien lisible, compréhensible et modulaire pour garantir la souplesse, l’évolution et la maintenance de la solution et ainsi répondre aux changements du marché.

**Rapidité et robustesse** :

L’application doit assurer la rapidité de réponse et la fiabilité du résultat procuré.

**Sécurité** :

L’application doit tenir compte de confidentialité des données des utilisateurs.

* L’organisation et la gestion des taches
* **Emploi du temps**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Noms des  tâches | Durée | Début | Fin |
|  | Obtention d’un ordre de mission | 7jours |  |  |
|  | Visite sur le terrain pour recueillir des informations | 2jours |  |  |
|  | Mise en place  De l’interface  graphique | 6  Jours |  |  |
|  | Analyse et conception de la Base de données | 7  jours |  |  |
|  | Programmer la phase d’authentification  Et les utilisateurs  De l’application (ajout,  Modification suppression) | 8  jours |  |  |
|  | Programmer les menus candidats, les moniteurs(ajout, modification, suppression) | 14jours |  |  |
|  | Programmer les menues dépenses, paiements, cours  (ajout, modification, suppression) | 16  jours |  |  |
|  | Ajout d’image, d’icône, pour embellir l’application, | 3  jours |  |  |
|  | Rédaction du  rapport | 15  jours |  |  |

**Chapitre 2 : Etude détaillée - Modélisation**

**Introduction**

Dans ce chapitre, nous présenterons la conception détaillée du projet, qui constitue à indiquer d’abord les cas d’utilisations, les diagrammes de séquences, le diagramme de classe, qui a pour objectif de détailler la conception du système, partant par l’architecture logique de l’application puis la conception détaillée (diagrammes).

**Diagrammes de cas d’utilisation**

Le diagramme de cas d’utilisation est un diagramme UML utilisé pour donner une vision statique et globale du comportement fonctionnel d’un système logiciel.

**Identification des acteurs**

Un acteur représente un élément externe qui interagit avec le système.

Cet élément peut être un utilisateur ou un système tiers (autre ordinateur, autre programme, base de données). En observant les utilisateurs directs du système ainsi que les autres systèmes qui interagissent avec le système que nous tentons de définir, nous trouvons ces acteurs qui opérant avec le système : admin, utilisateur.

**Identification des cas d’utilisation**

Un cas d’utilisation est une unité cohérente représentant une fonctionnalité visible de l’extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l’acteur qui l’initie. Un cas d’utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

**Cas d’utilisation « Authentification »** :

|  |  |
| --- | --- |
| **Titre** | **Authentification** |
| Acteurs | Admin et utilisateur(employé) |
| Préconditions | L’utilisateur de l’application (admin, employé) fourni le courrier et mot de passe |
| Scenarios | Identification :  L’utilisateur saisit ses droits d’accès  (Courrier et mot de passe)  Le Système vérifie si les informations sont valides, si erreur alors Exception.  Le système redirige l’acteur vers page d’accueil selon son rôle |
| Post-conditions | L’accès au système(application) |

**Cas d’utilisation « Effectuer les tâches »**

|  |  |
| --- | --- |
| **Titre** | **Effectuer des tâches** |
| Acteurs | Admin, employé (Utilisateurs) |
| Préconditions | L’utilisateur doit avoir le droit d’accès  L’utilisateur doit s’authentifier |
| Scenarios | Etapes  L’utilisateur soumet une demande d’ajouter un nouvel enregistrement  Le système envoie le formulaire d’ajout d’un nouvel enregistrement  L’utilisateur remplit le formulaire et clique sur le bouton ajouter (Valider)  Le système vérifie si les champs obligatoires sont complets.  Le système vérifie si les champs obligatoires et optionnels sont valides.  Effectue l’enregistrement |
| Post-conditions | Information ajoutée ou modifiée |

**Diagramme de cas d’utilisation des tâches des utilisateurs**

Les diagrammes suivants (figure 3, figure 4 et figure 5) présentent les cas d’utilisations des modules de gestion de tout ce qui est lié au projet, les deux acteurs principaux sont admin et utilisateur(employé). Dans ce cas admin est l’acteur principale de ce système donc peut ajouter, modifier et supprimer toutes les fonctionnalités présentes ; par contre, des utilisateurs(employés) ne peut interagir qu’avec quelqu’un.

Toutes ces activités nécessitent l’authentification de l’utilisateur.

#### Cas d’utilisation pour Admin

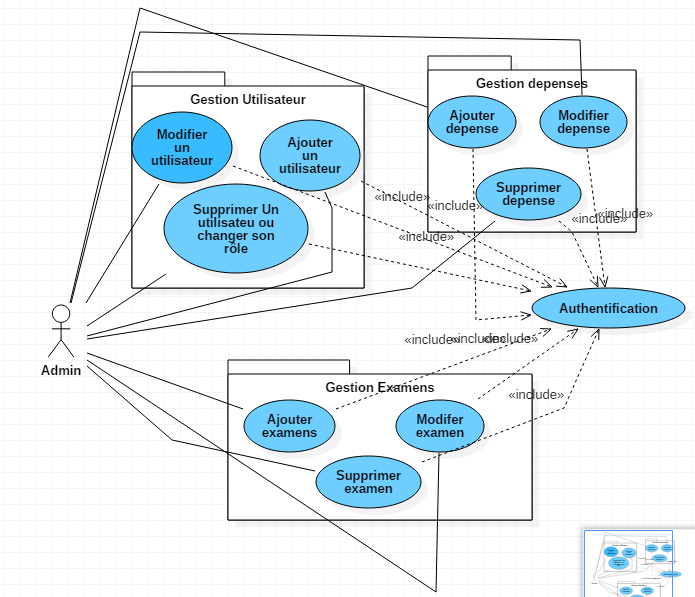


Figure 3 : cas d’utilisation de super admin

#### Cas d’utilisation pour Utilisateur

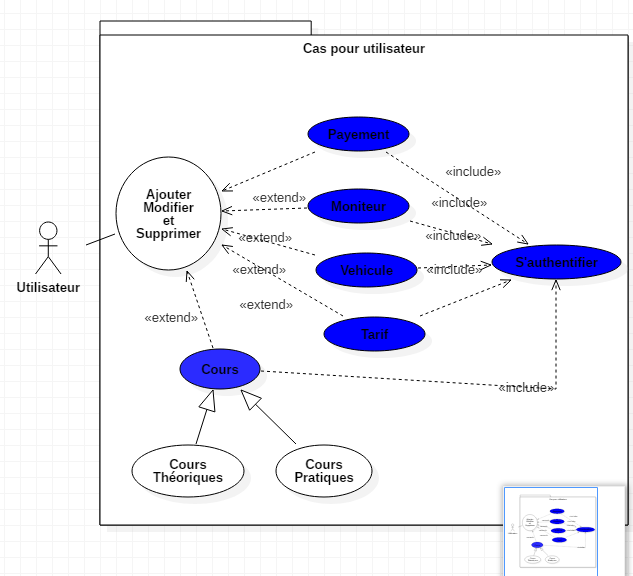


Figure 4 : cas d’utilisation d’Admin

Cas d’utilisation pour admin et Employés

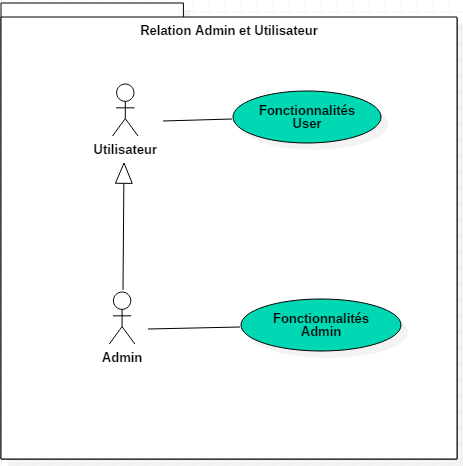


Figure 5 : cas d’utilisation d’Employé

**Diagrammes de séquences**

Le diagramme de séquence présente la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux. Le diagramme de séquence permet de cacher les interactions d’objets dans le cadre d’un scenario d’un diagramme des cas d’utilisation. Dans un souci de simplification. Dont le but est de décrire comment se déroulent les actions entre les acteurs ou objets. Après la description des cas d’utilisation, nous allons élaborer le modèle dynamique dans lequel nous allons décrire les scenarios de quelques cas d’utilisation, sous forme de diagrammes de séquence.

Alors on représente le diagramme de séquence de chaque utilisateur de l’application

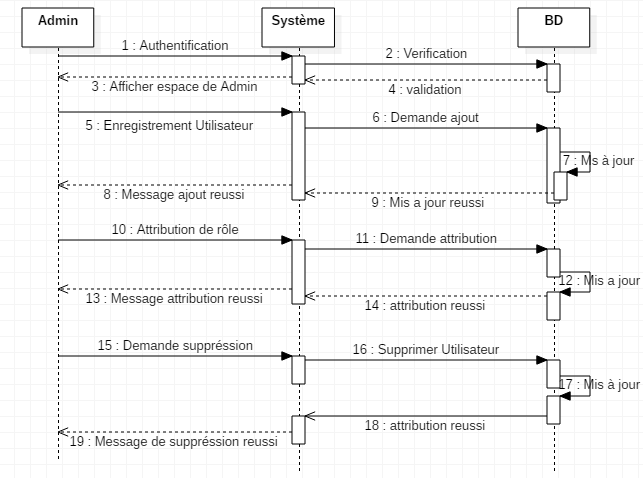
Pour Admin

Figure 6 : Diagramme de séquence d’Admin

### Pour Utilisateur

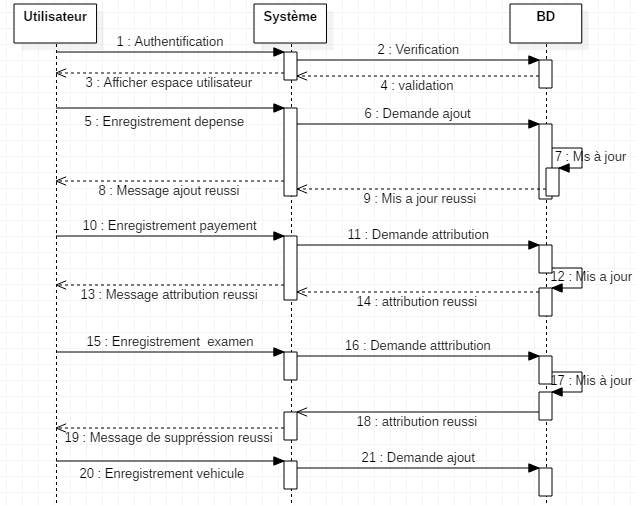


Figure 7 : diagramme de séquence utilisateur

**Diagramme de classe**

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d’une telle modélisation. Le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d’utilisation. A partir de l’analyse lors de la partie analyse du projet et des diagrammes de séquence présente, nous avons extrait un ensemble d’entités et de dépendances, cela a été traduit par UML en un diagramme de classes (figure) qui sera ensuite décrit par un tableau détaillant le rôle de chaque entité

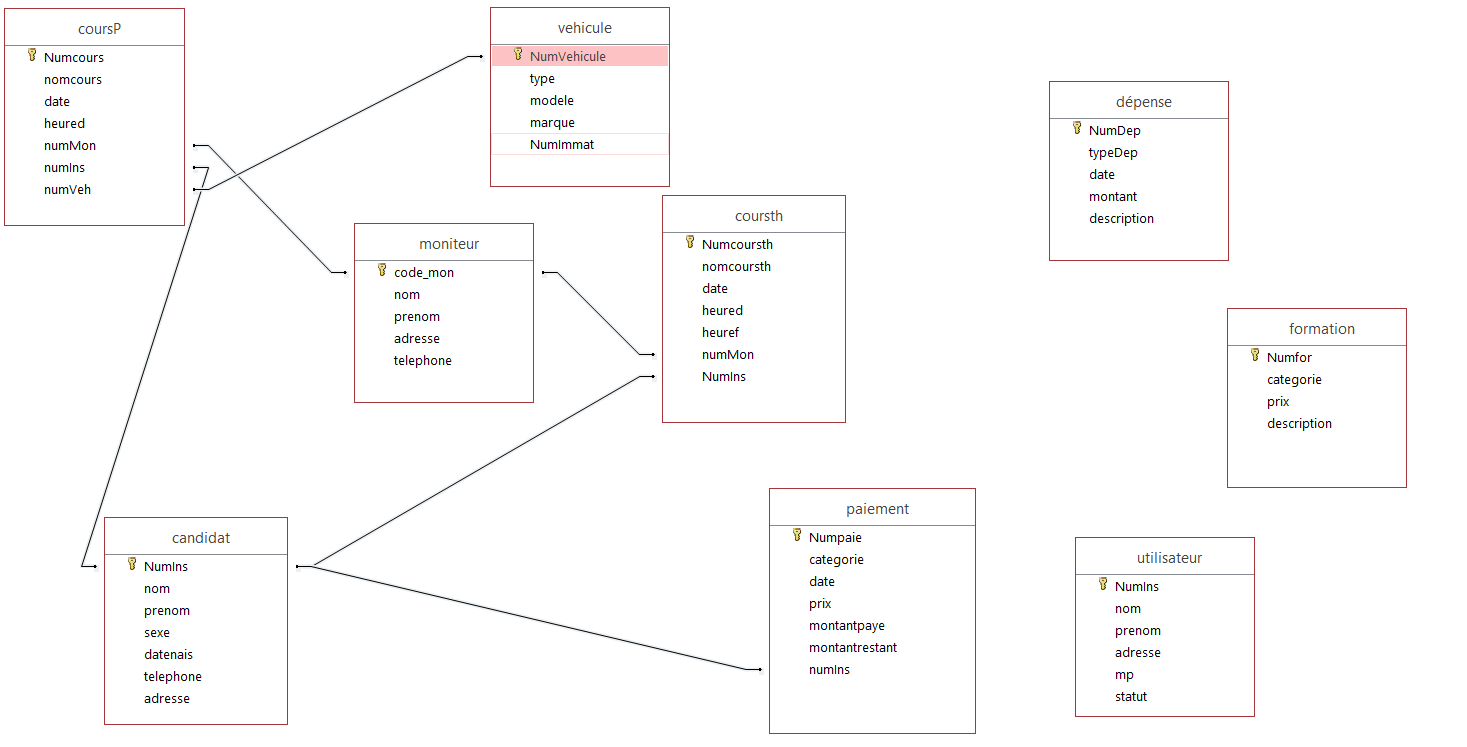


Figure 9 : diagramme de classe

**Conclusion**

Ce chapitre a permis de donner une image claire et nette du système. Nous avons spécifié les cas d’utilisations du système, les diagrammes de séquence et un diagramme de classe.

**Chapitre 3 : Conception et développement / Etude technique**

**Introduction**

Dans cette partie, nous présenterons les outils de développement utilisés, le logiciel et l’environnement matériel en premier, après nous vous montrerons quelques interfaces importantes de notre application.

## Environnements de travail :

Dans cette partie, nous montrerons l’environnement matériel mis à la disposition du présent projet, ainsi que l’environnement logiciel utilisé pour le développement et la mise en place de notre application web

***Environnements matériel***

Pour la réalisation de ce projet, nous avons disposé d’un ordinateur Lenovo caractérisé par :  
• Processeur : Intel® Core™ i5-4300M CPU @ 2.60 GHz 2.59 GHz  
• Mémoire : 8 Go de RAM.  
• Disque dur : 1 Téra .  
• Système d’exploitation : Windows 10 professionnel & office news édition 2019.

***Environnement logiciel :***

Voici les logiciels qui nous avons utilisé pour mettre en place cette application

* **Langage de programmation**

****

PHP: HyperText Préprocesseur, plus connu sous son sigle PHP (sigle autoréférentiel), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur http, c’est un langage orienté objet.

* + ***Framework :***

***Laravel :***

Laravel est un Framework PHP Open Source, un cadre d’application PHP à code source ouvert, qui utilise le patron de conception MVC.

Il a été créé par [Taylor Otwel](http://taylorotwell.com/) en 2011.

* **Logiciels de conception**

StarUML

StarUML est un logiciel de modelage UML qui est entré récemment dans le monde de l’open source. Ecrit en Delphi, il est modulaire et propose plusieurs générateurs de code.****

Access

Access est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

**Visual Studio Code :**

Visual studio code est un éditeur de texte générique qui est disponible sous Windows, Mac et Linux.

**Système de gestion de base de données**

**MySQL**

****MySQL est un système de gestion de base de données (SGBD). C’est un serveur de bases de données relationnelles Open Source. Il stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l’ensemble. Il est très utilisé dans les projets libres et dans le milieu industriel. MySQL est très souvent utilisées avec PHP.

**Laragon**

Laragon est outil puissant pour les applications modernes : Django, Rails, Laravel, Node, Go, Java. Laragon rend le codage plus amusant et plus facile

## QUELQUES INTERFFACES DE L’APPLICATION

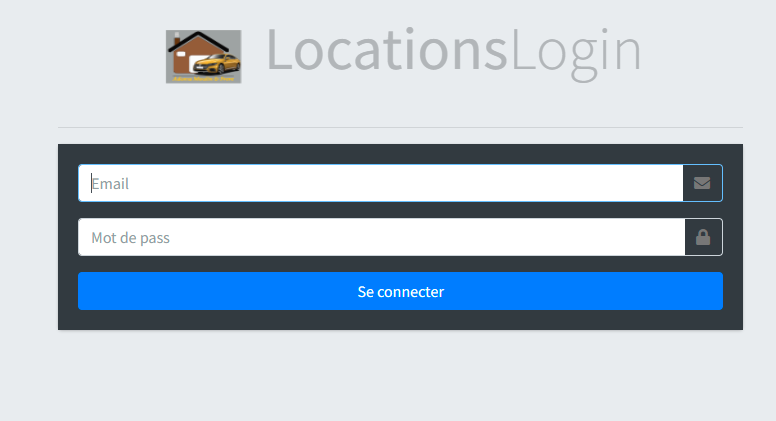


Figure 10 : Interface de connexion

Cette figure montre l’identification des utilisateurs pour se connecter et effectuer les tâches.

Ici chaque employé rempli ces deux champs puis clique sur le bouton Se connecter et il sera redirigé vers son profil.

La page suivante montre le profil de l’employé connecté.

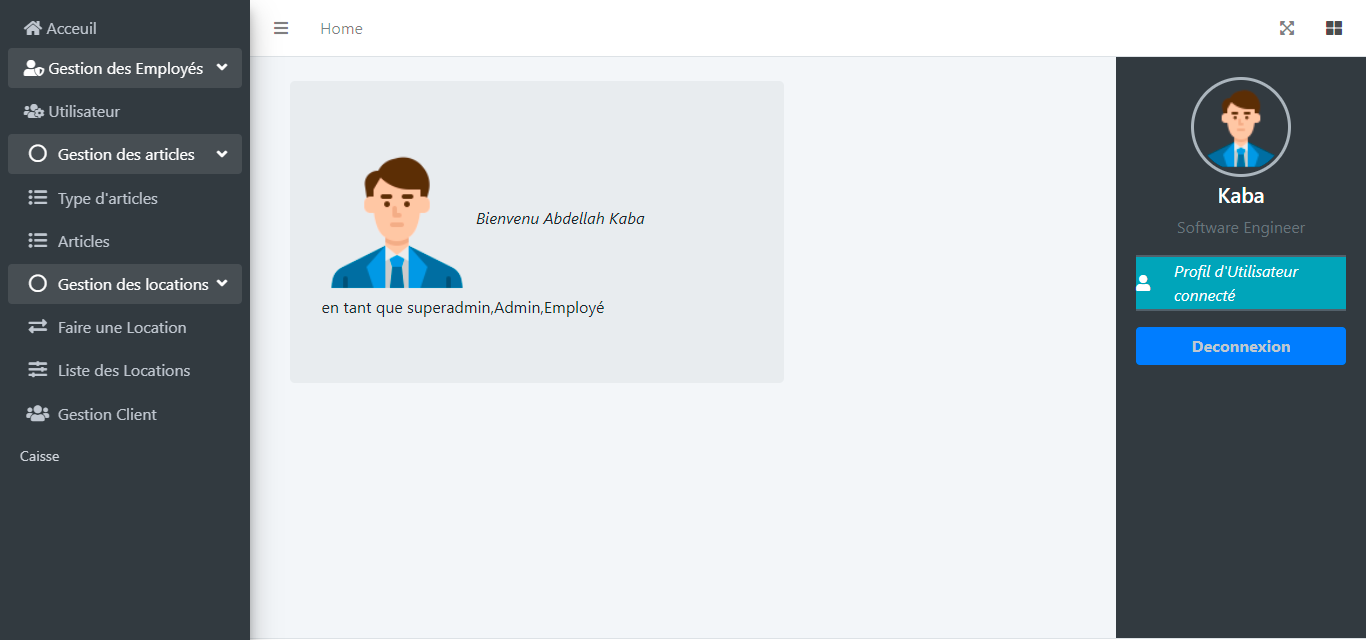
s

Figure 11 : Interface de page d’accueil

Cette interface montre le profil de l’Utilisateur connecté.

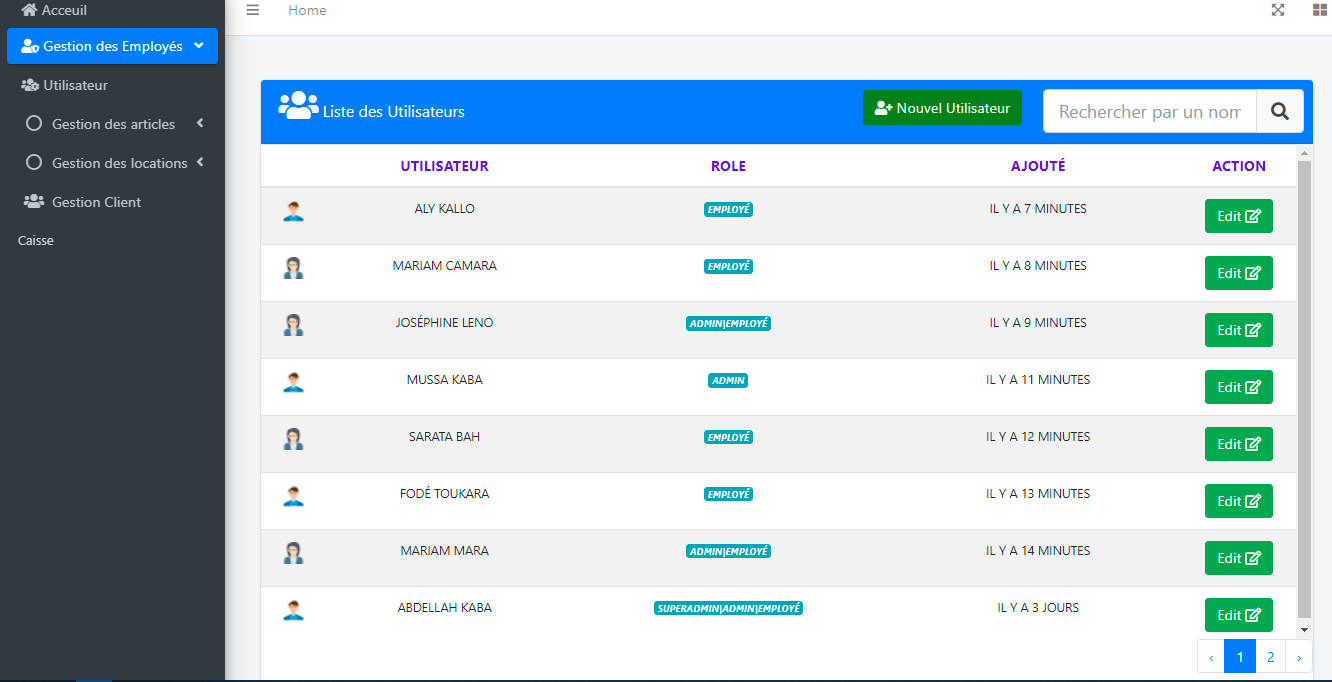
Il peut se déconnecter en cliquant sur le bouton Déconnexion à droite, il peut également voir tous les tâches qu’il peut effectuer dans le tableau de bord à gauche.

Figure 12 : Interface de liste des utilisateurs

Cette interface permet de montrer la liste de tous les Utilisateurs enregistrés dans la base de donnée avec les rôles qui les sont attribués.

Ici le Super admin peut cliquer sur le bouton en haut pour ajouter un nouvel Utilisateur.

Passons maintenant à la création d’un nouvel utilisateur. Le Super admin remplit le formulaire dans la page suivante.

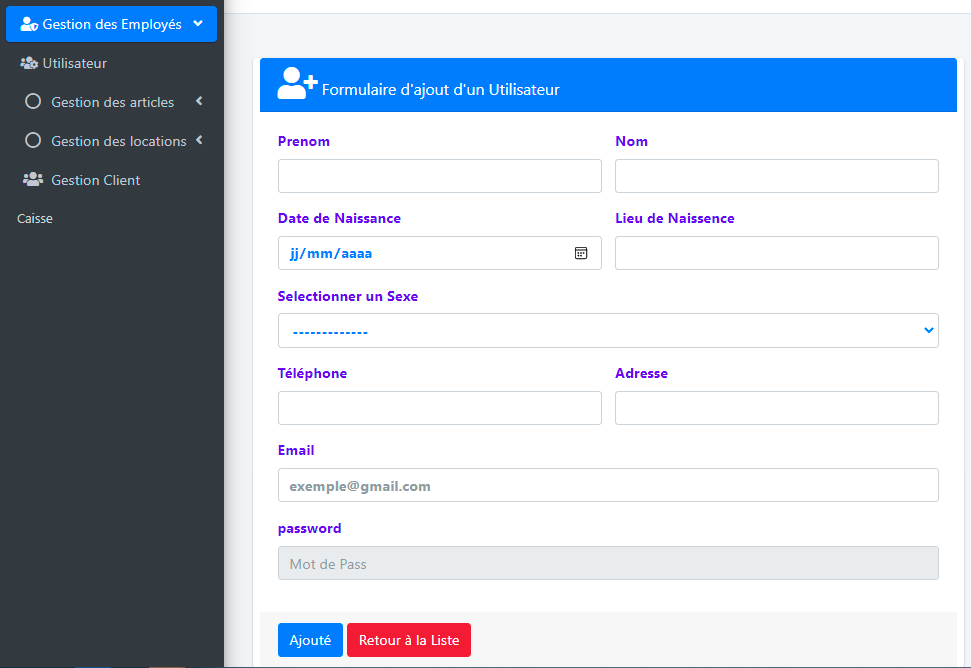
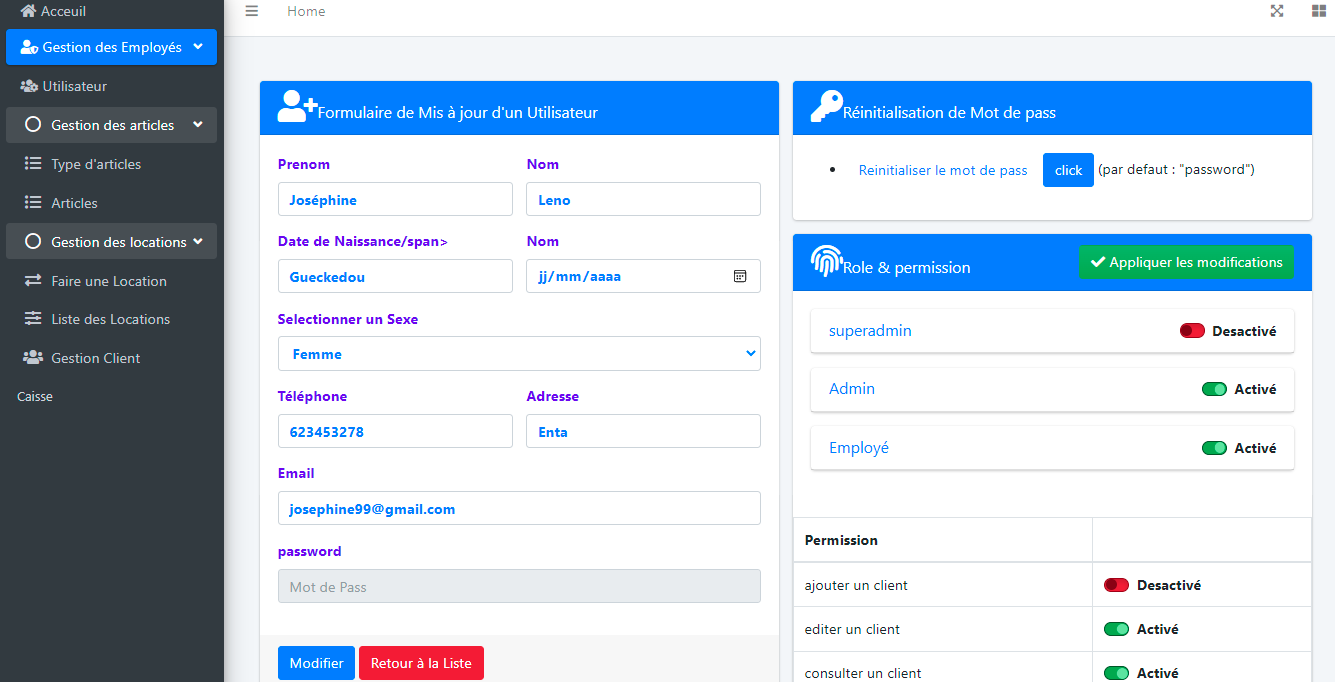


Figure 13: Interface d’ajout des utilisateurs

Cette interface permet de montrer le formulaire d’ajout d’un nouvel Utilisateur ou un Employé.

Ici le Super admin remplit les champs ci-dessous puis il clique sur le bouton ajouter pour enregistrer un nouvel Utilisateur.

***LOGIN***

Figure 14 : Interface de mis à jour des utilisateurs

Cette interface montre le formulaire de modification d’un utilisateur.

Elle montre aussi l’attribution du rôle à l’Utilisateur édité.

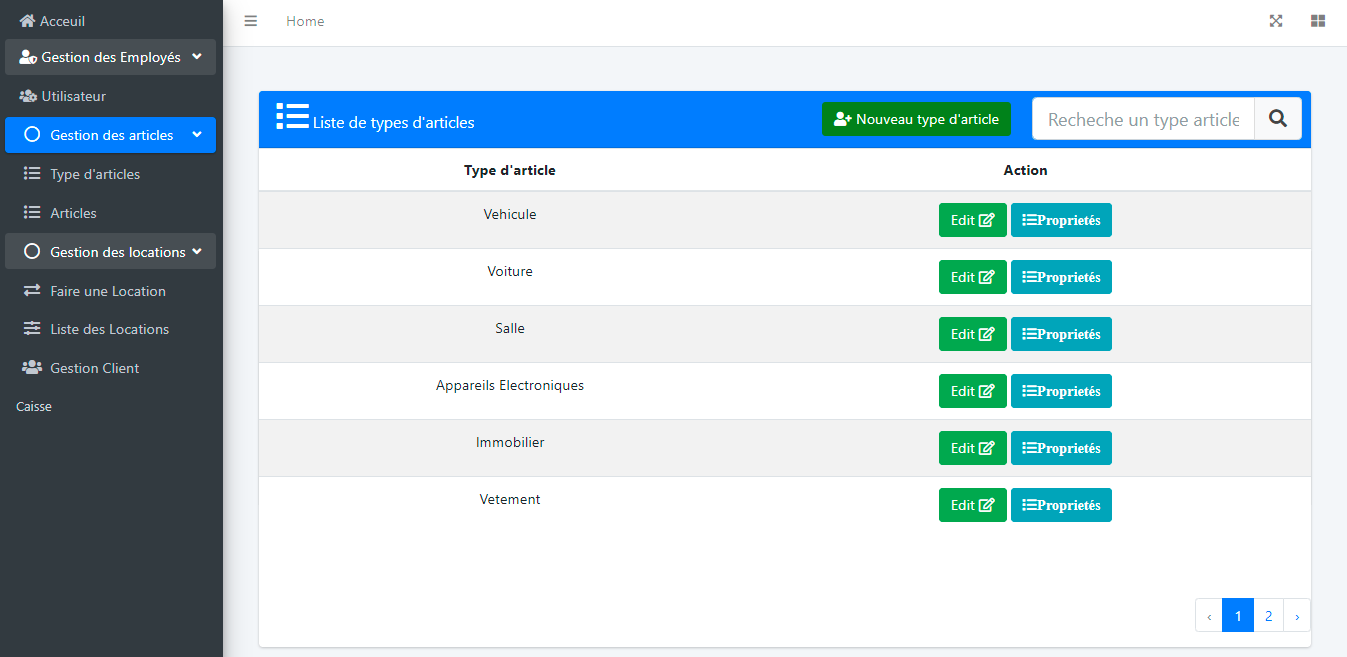
Ici le Super admin peut modifier les informations qui sont affichées dans les champs de saisi puis clique sur le bouton Modifier pour faire la modification, il peut aussi attribuer des rôles à un utilisateur à droite.

Figure 15 : Interface de liste des types d’articles

Cette interface permet de montrer la liste de tous les types d’articles.

Ici l’admin peut cliquer sur le bouton en haut pour enregistrer un type d’article, il peut ajouter des propriétés à chaque type d’article en cliquant sur le bouton propriétés.

Passons à l’ajout d’un type et comment ajouter des propriétés ces types d’article dans la page suivante.

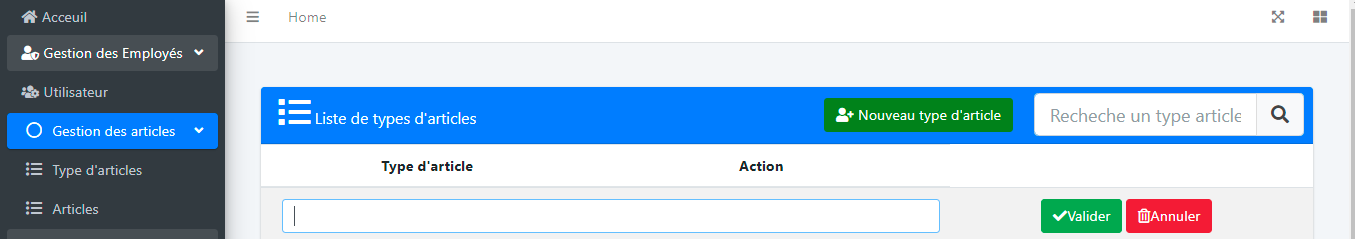


Figure 16 : Interface d’ajout d’un type d’article.

Cette interface permet de montrer le formulaire d’ajout d’un type d’article.

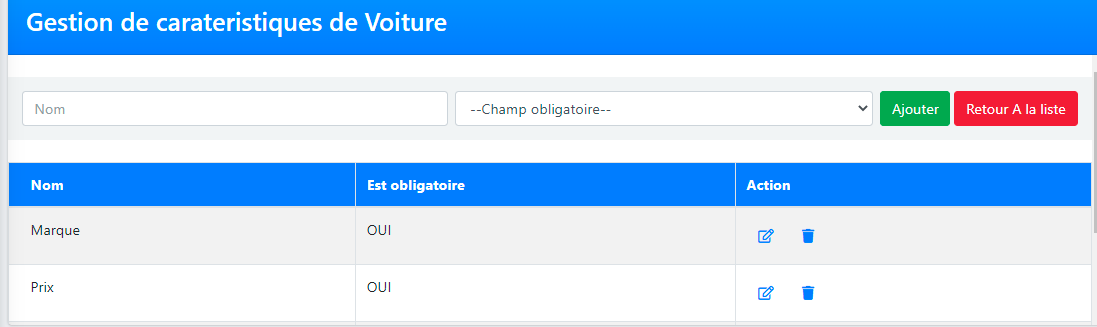
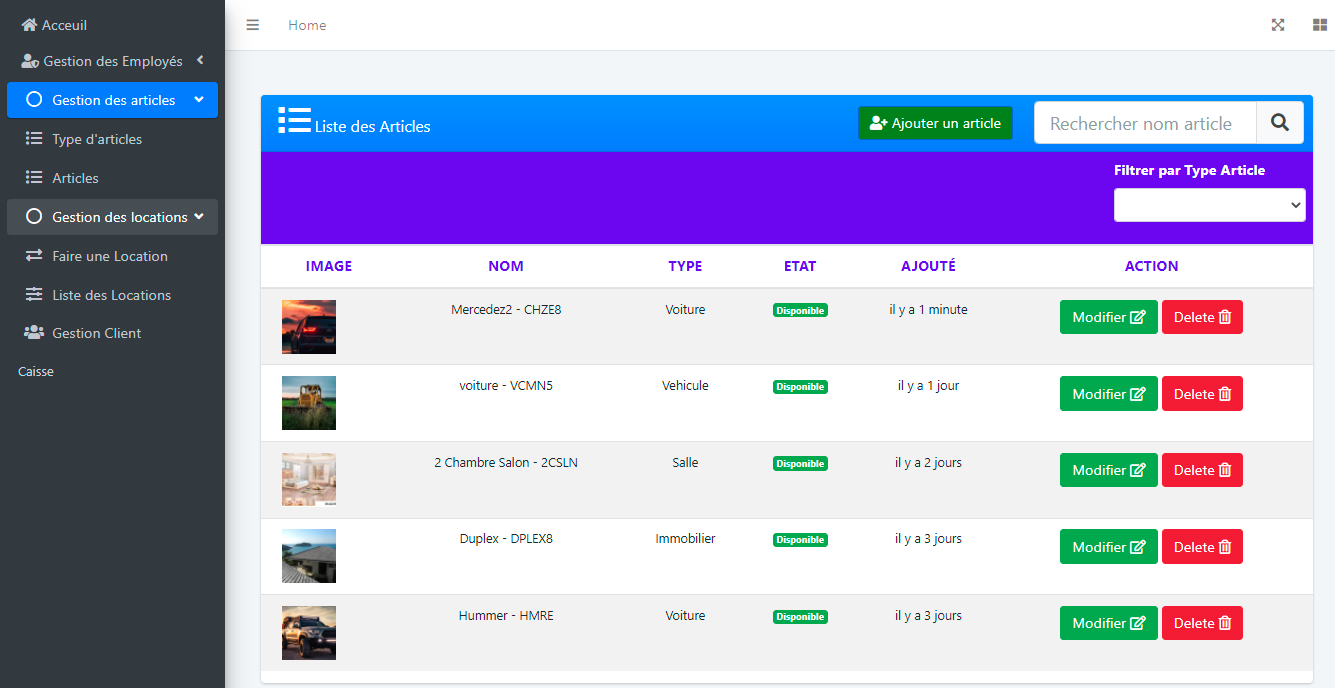
Ici l’Admin peut ajouter un type article.

Figure 17 : Interface d’ajout des propriétés de types d’articles et liste

Cette interface permet de montrer le formulaire d’ajout et la liste des propriétés de types d’articles.

Ici l’Admin peut ajouter un type d’article en remplissant les deux champs, il peut également modifier et supprimer les propriétés de chaque type d’article.

Les deux icônes montrent la suppression et modification de ces propriétés de types d’articles.



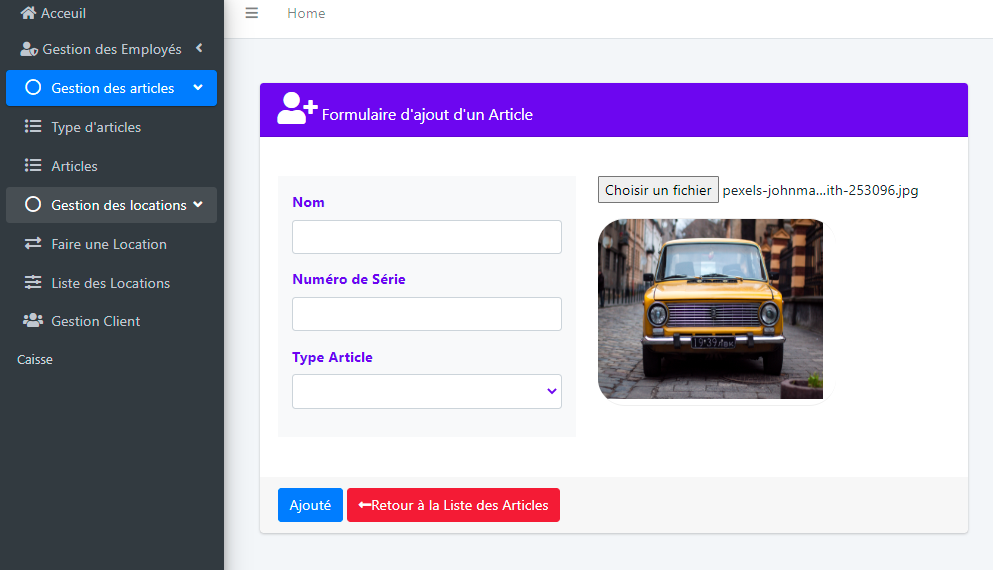


Figure 18 : Interface de liste des articles.

Cette interface permet de montrer la liste de tous les articles existant dans la base de donnée.

Ici l’Admin peut voir tous les articles enregistrés, il peut cliquer sur les trois boutons Modifier, Delete et Ajouter un article pour modifier, supprimer et ajouter un article.

Passons aux formulaires de modification et ajout d’un article dans les pages suivantes.

Figure 19 : Interface d’ajout d’articles

Cette interface permet de montrer le formulaire d’ajout d’un nouvel article.

Ici l’Admin peut remplir les champs de saisi et choisir la photo de cet article puis cliquer sur Ajouter pour l’enregistrer.

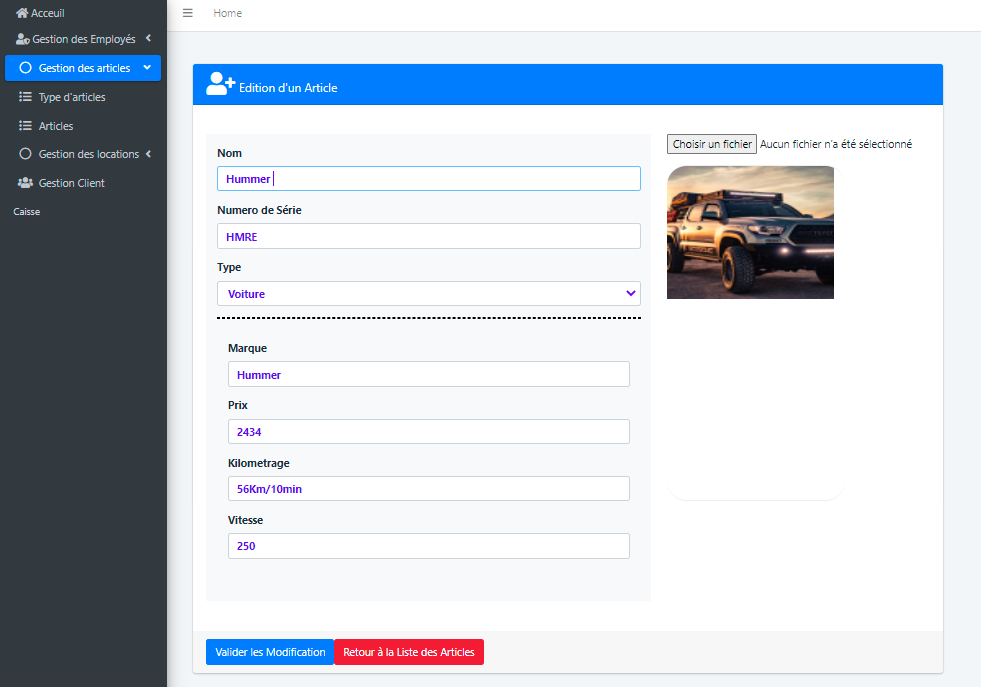


Figure 20 : Interface de mis à jour d’un article.

Cette interface permet de montrer le formulaire de modification d’un article.

Ici l’Admin peut modifier les informations qui sont écrites dans les champs et modifier l’image d’article puis cliquer sur le bouton Valider les Modifications.

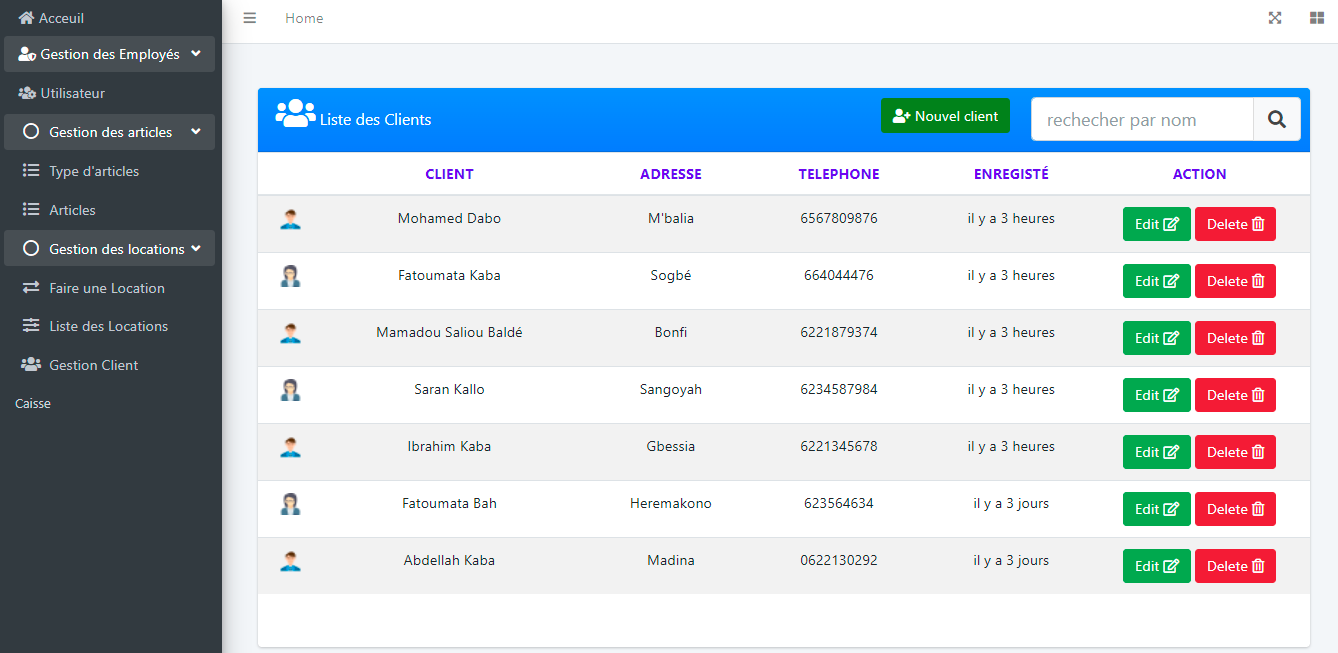


Figure 21 : Interface de liste des clients.

Cette interface permet de montrer la liste des clients qui sont venus pour louer les articles.

Ici l’Employé peut voir toutes les informations concernant aux clients, il peut modifier un client, le supprimer et il peut cliquer sur le bouton nouvel client pour ajouter.

Passons maintenant aux formulaires de modification et d’ajout d’un nouvel client dans les pages suivantes.

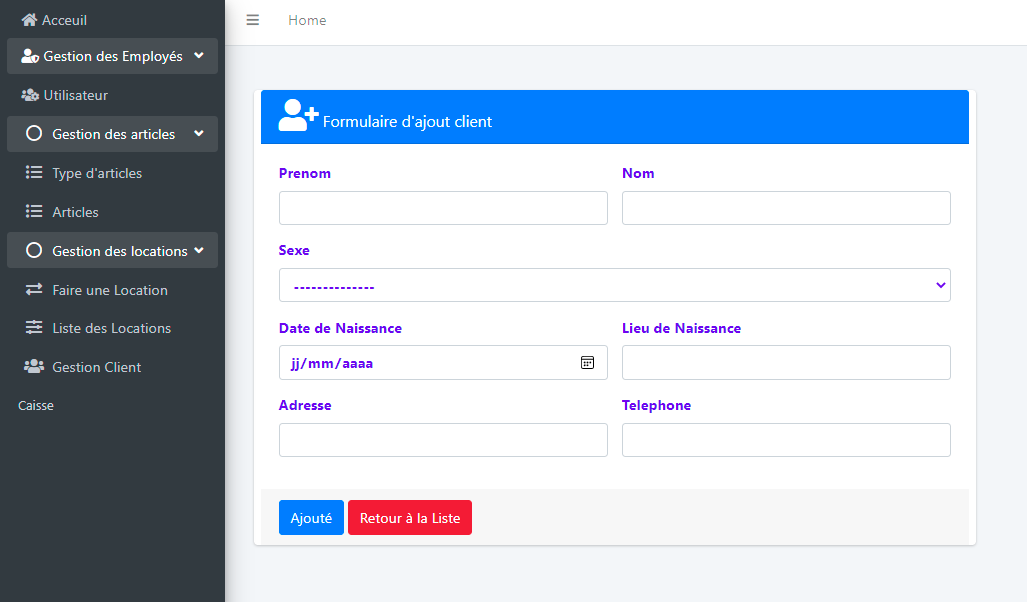


Figure 22 : Interface d’ajout d’un nouvel client.

Cette interface permet de montrer le formulaire d’ajout d’un nouvel Client.

Ici l’Employé peut remplir les champs de saisi puis cliquer sur le bouton Ajouter pour ajouter un client.

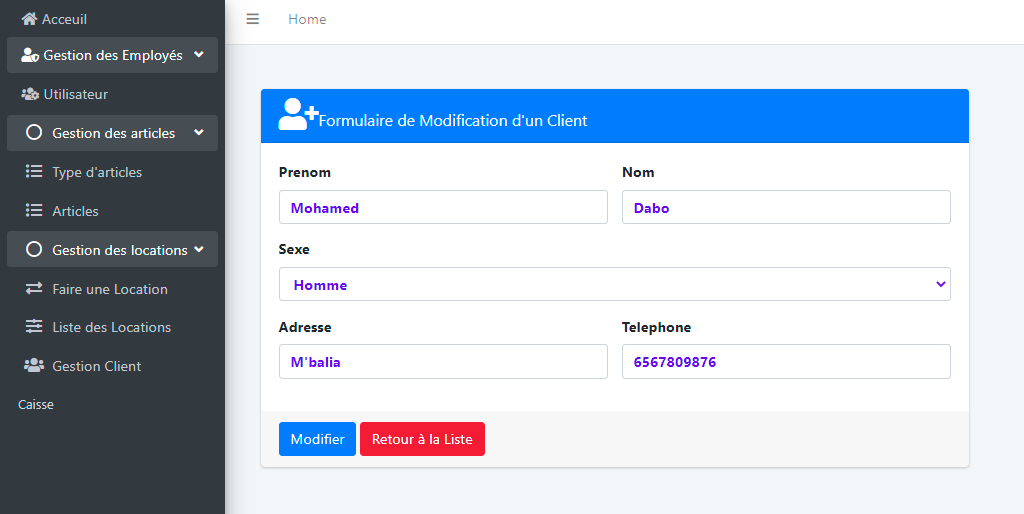


Figure 23 : Interface de modification d’un client.

Cette interface permet de montrer le formulaire de modification d’un client.

Ici l’Employé peut modifier les informations qui sont écrites dans les champs et cliquer sur le bouton Modifier.

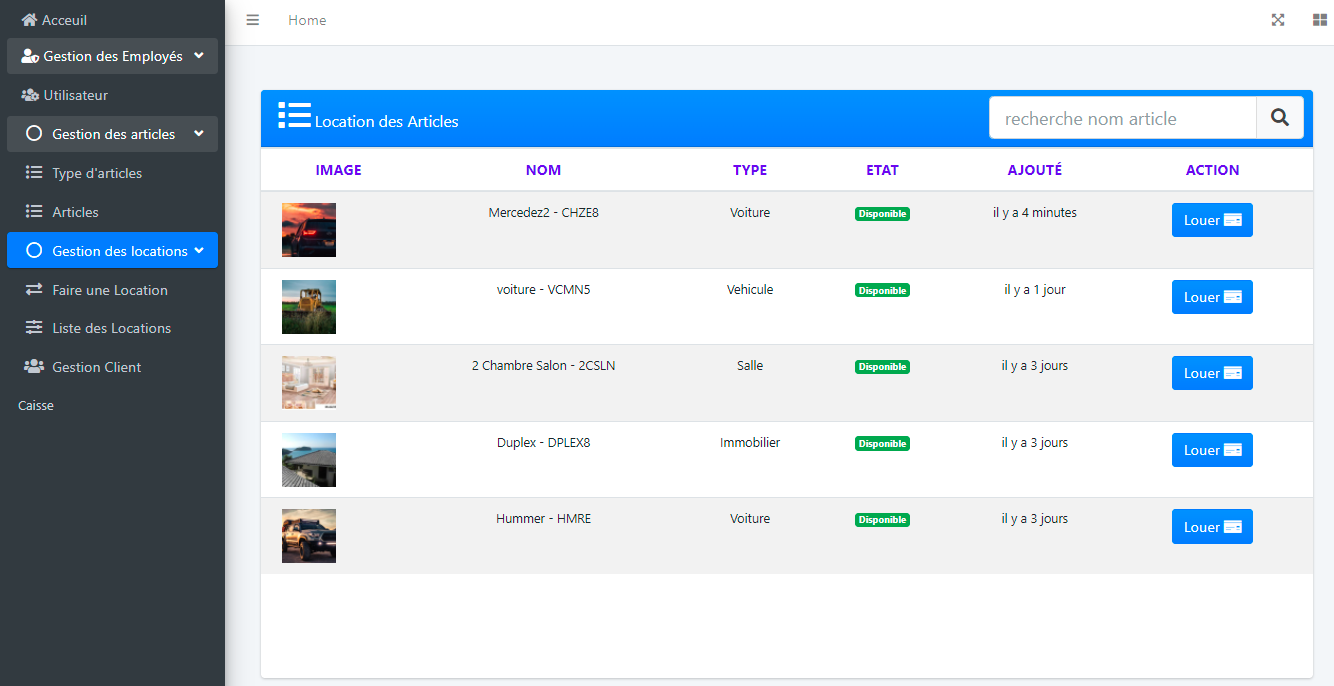


Figure 24 : Interface d’effectuer une location.

Cette interface permet de montrer à l’employé d’effectuer une location.

Ici l’Employé peut cliquer sur le bouton Louer, il sera redirigé au formulaire d’enregistrement d’une location à moins que l’article n’est pas en cours de location et il y’a un bouton dont qu’il va cliquer pour enregistrer.

Passons à la liste et au formulaire de location d’un article dans la page suivante.

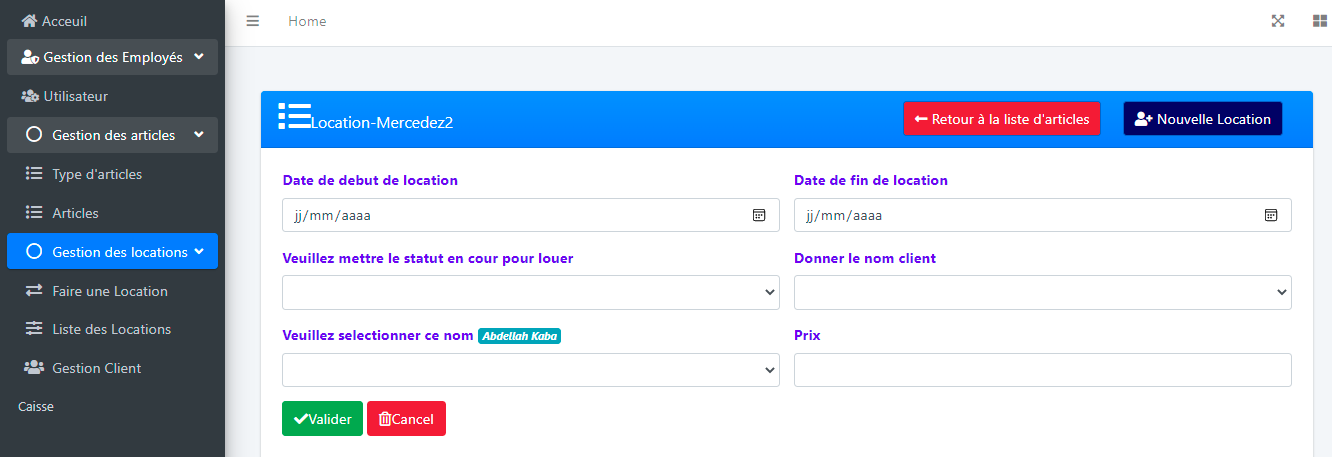


Figure 25 : Interface de formulaire d’une location.

Cette interface permet de montrer le formulaire pour effectuer une nouvelle location d’un article.

Ici l’Employé rempli les champs de saisi puis il clique sur le bouton Valider pour enregistrer la location.

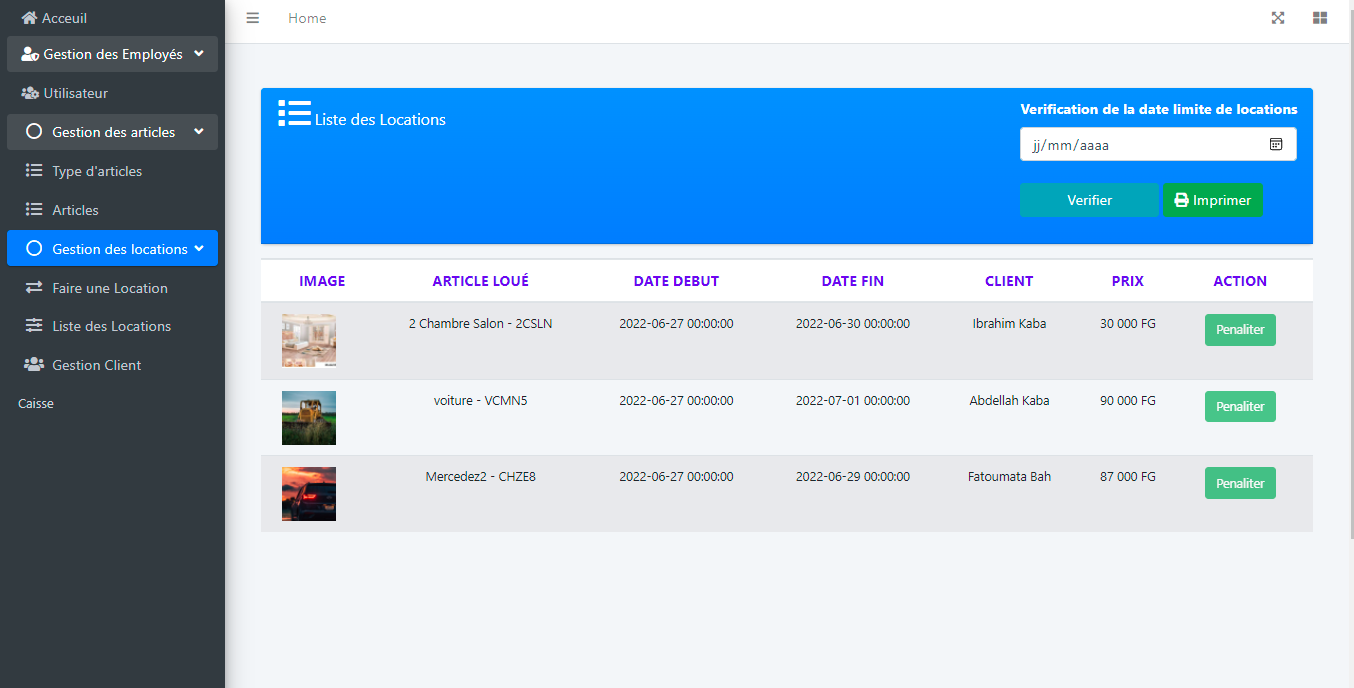


Figure 26 : Interface de liste des locations en cours.

Cette interface permet de montrer la liste des locations qui sont en cours.

Ici l’Employé peut voir la liste de toutes les locations d’articles qui sont en cours.

Il peut également imprimer cette liste et vérifie la date des locations qui est dépassée.

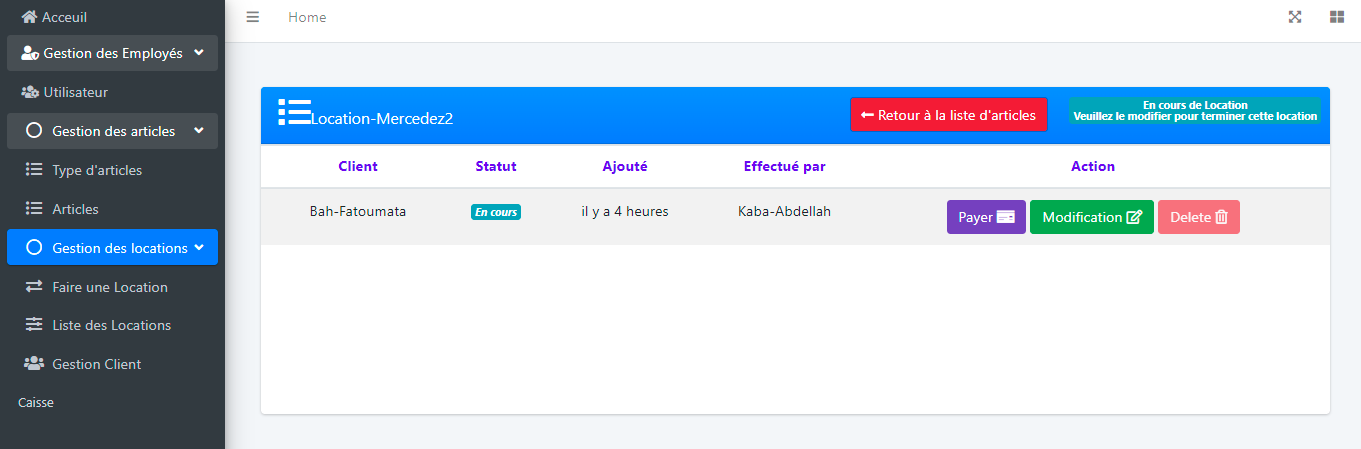


Figure 27 : Interface du paiement effectué d’une location

Cette interface permet de montrer le paiement effectué d’une location en cours.

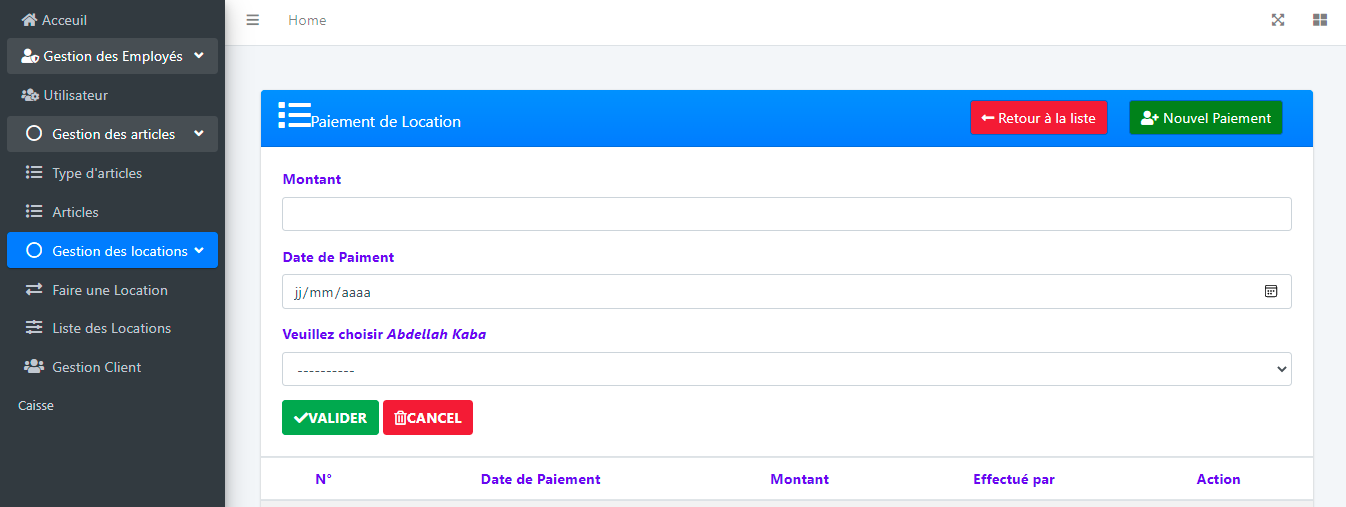
Ici l’Employé clique sur le bouton payer et il sera redirigé vers le formulaire de paiement, il peut aussi modifier une location c’est-à-dire terminé une location en cliquant sur le bouton Modifier et supprimer un paiement qui est effectué en cliquant sur le bouton Delete.

Figure 28 : Interface d’enregistrer le paiement d’une location.

Cette interface permet de montrer le formulaire d’enregistrement de nouvel paiement d’une location. Ici l’Employé rempli les champs et clique sur valider.

**Les Tests**

En informatique, un test désigne une procédure de vérification partielle d’un système. Son objectif principal est d’identifier un nombre maximal de comportements problématiques du logiciel. Il permet ainsi, dès lors que les problèmes identifiés seront corrigés, d’en augmenter la qualité. D’une manière plus générale, le test désigne toutes les activités qui consistent à rechercher des informations quant à la qualité du système afin de permettre la prise de décisions.  
Pour la réalisation de notre application, nous avons recours à deux type de Test

* **Test Unitaire** : Pour tester de façon indépendante chaque bout de partie de l’application
* **Test d’intégration** : consiste la vérification de bon fonctionnement de chacun des parties indépendantes de l’application.

**Conclusion générale**

Notre projet de fin d’étude a été réalisé au profit de l’entreprise Adama Moulin & Frère. Durant ce projet la mission suivante nous a été confiée : la conception et la réalisation d’une application de gestion des Locations de l’entreprise afin d’aider à la décision. A ce stade de réalisation, on a pu développer une solution informatique couvrant totalement les besoins métiers fonctionnels et techniques qui répondent totalement aux exigences des utilisateurs finaux. Notre choix de la plateforme a porté ses fruits en poussant les performances de l’application à des records surprenants. Pour ce faire nous avons commencé dans un premier lieu par comprendre le contexte général de notre application et identifier les différentes exigences de notre futur système. Ensuite une étude détaillée du cahier de charges afin de comprendre les besoins des futurs utilisateurs du système, et par la suite nous avons commencé à développer les différents vues et objets, en respectant les différentes étapes du cycle de développement Scrum.

**Perspectives**

Les perspectives possibles à la suite du présent projet sont multiples et couvrent plusieurs aspects, tels que l’émigration des données a une base de données Heidi SQL, cette expérience était opportunité que nous avons apprise en travaillant avec des technologies récentes et sollicitées sur le marché, plus particulièrement, le développement des applications avec la spécification PHP Web et en utilisant un Framework très populaire Laravel, comme elle nous a offert l’opportunité d’intégrer dans l’environnement de l’entreprise et d’améliorer nos capacités dans notre vie professionnelle tout en enrichissant nos connaissances et notre formation théorique et pratique acquise durant nos études UGANC.

**Références et Bibliographie**

[Réf 1] : mémoire online [www.memoireonline.com](http://www.memoireonline.com) consulté le 10/mai/2022

[Réf 2] : <https://laravel.com/docs/9.x> consulté le 13/juin/2022

[Réf 3] : <https://www.coderspassion.com/> consulté le 13/juin/2022

[Réf 4] : <https://laravel-livewire.com/docs/2.x/properties> 12/Juin/2022

UML pratique de la modélisation- Yann THIERRY-MIEG, édition Pearson