



**CENTRE INFORMATIQUE
ET DE CALCUL (CIC)**



ALL SOLUTIONS

N° d'ordre : 2019/GL-012

MEMOIRE DE FIN DE FORMATION

pour l'obtention du diplôme de

LICENCE PROFESSIONNELLE

Domaine : **Sciences et technologies**

Mention : **Informatique**

Spécialité: **Génie Logiciel**

Thème :

**CONCEPTION D'UN SYSTÈME DE PAIEMENT AUTOMATIQUE
DE PÉAGE ROUTIER**

Présenté et soutenu publiquement le 16/10/2019 par :

VEYATO AKPAKU Koami Hervé

Devant le Jury composé de :

Président du Jury : M. ATCHONOU GLO Kossi, Maître de conférences, Directeur au CIC

Examineur : M. ABOLO-SEWOVI Komi, Ingénieur, Enseignant au CIC

Directeur de mémoire : M. AKAKPO Folly, Informaticien, Enseignant au CIC

Maître de stage : M. DOFONNOU M. K., Agent technique des impôts, Directeur à ALL Solutions

DÉDICACE

À ma famille,
pour tous les efforts consentis en faveur de mon éducation.

REMERCIEMENTS

Gloire au créateur, **DIEU**, qui par sa grâce, nous a toujours insufflé le souffle vital et l'abnégation nécessaire tout le long de notre formation.

Nous exprimons notre gratitude à :

- M. Kossi ATCHONOU GLO, Directeur du CIC, pour sa foi en cet institut et pour tous les efforts qu'il y déploie ;
- M. Tchamye Tcha-Esso BOROZE, Directeur-Adjoint du CIC, pour son dévouement pour la cause des étudiants, sa culture de l'excellence et aussi pour nous avoir rendu verveux afin de réaliser des projets de promotion ;
- M. Folly AKAKPO, notre Directeur de mémoire, pour avoir accepté de nous accorder un peu de son temps précieux pour diriger ce travail, pour ses conseils et pour tous les efforts consentis pour la réussite de notre cursus scolaire ;
- M. Modeste Kévin Mawouéna DOFONNOU, notre Maître de stage, pour nous avoir accepté au sein de son entreprise et pour son monitoring durant toute la période du stage ;
- M. Anani ATADEGNON, Responsable de la division Développement et Enseignant au CIC, pour la transmission de ses connaissances et le partage de son expérience ;
- M. Kodjo ASSIDENU, Responsable du service des examens et Enseignant au CIC, en sa qualité de Parrain de notre promotion ;
- M. Yaovi HOETOWOU, Développeur au CIC, qui par son esprit d'analyse et son savoir-faire, nous a guidé au cours de notre formation pour la réalisation du projet tutoré ;
- M. Toyo Koffi EDARH BOSSOU, Enseignant en Mathématiques, pour ses conseils ;
- M. Badamessim ADEWUI, Secrétaire pédagogique du CIC, pour son encadrement ;
- M. Kossi MADZRA DZOKA, Chargé de la logistique du CIC, pour la bonne gestion de notre cadre de formation ;
- Mes très chers parents, pour leur affection, leur soutien et leur confiance ;
- Tout le corps enseignant, le personnel administratif et technique du CIC, pour la qualité de la formation et l'encadrement ;
- Tous mes amis, camarades étudiants, devanciers et promotionnels du CIC, pour leur soutien et leur confiance sans condition.

RÉSUMÉ

Les activités de transport de personnes et de biens augmentent sans cesse dans le monde et en particulier dans les pays en voie de développement. Afin d'améliorer la fluidité et réduire les risques d'encombrements, une des solutions consiste à rendre les péages plus performants. L'entreprise ALL Solutions, dans l'atteinte de l'un de ses objectifs, être un acteur du Plan National de Développement (PND) du Togo, s'est donné pour mission la mise sur pied d'un système de péage moderne qui permettra de réduire les dysfonctionnements observés dans nos péages actuels. Nous avons ainsi conçu un système qui, par le truchement d'un identificateur de véhicules, automatise les paiements au péage. Il permet un paiement automatique des péages avec un temps d'arrêt réduit. Ce système fournit des statistiques détaillées et permet de faire un suivi des passages au péage. Un système gère en back office les abonnements de passages des conducteurs et communique avec un système d'identification de véhicules qui, détecte et vérifie l'éligibilité des véhicules via une carte RFID (Radio Frequency Identification). Le système mis en place permettra donc d'accélérer et de rendre plus efficaces les différents processus liés à la gestion des péages.

Mots clés : péage, identification de véhicules, back office, RFID.

ABSTRACT

The activities of transport of people and goods are increasing in the world, especially in the developing countries. In order to improve fluidity and reduce the risks of congestion, one of the solutions is to make tolls more efficient. The company ALL Solutions, in the achievement of one of its objectives, been an actor of the National Development Plan (NDP) of Togo, has gave itself the mission to set up a modern toll system which will reduce the malfunctions observed in our current tolls. We have then designed a system that, through a vehicle identification system, automates toll payments. It allows automatic payment without stopping vehicles. This system manages detailed statistics and allows tolls to be tracked. A back office system manages pass subscriptions and communicates with a vehicle identification system that detects and verifies the eligibility of vehicles via RFID (Radio Frequency Identification) card. The system established would therefore make it possible to speed up and make more efficient the various processes related to toll management.

Key words: toll, vehicle identification, back office, PND, RFID.

SOMMAIRE

LISTE DES PARTICIPANTS DU PROJET	vii
LISTE DES TABLEAUX.....	viii
LISTE DES FIGURES	ix
GLOSSAIRE.....	xii
INTRODUCTION	1
CHAPITRE 1 PRÉSENTATION DES CADRES DE FORMATION ET DE STAGE.....	3
1.1. Présentation du cadre de formation : CIC-UL [1] [2]	4
1.2. Présentation du cadre de stage : Entreprise ALL Solutions	7
CHAPITRE 2 ÉTUDE PRÉALABLE DU PROJET	12
2.1. Présentation du sujet.....	13
2.2. Problématique du sujet.....	13
2.3. Intérêt du sujet.....	14
2.4. Etude de l'existant.....	15
2.5. Critique de l'existant	16
2.6. Proposition de solutions	17
2.7. Choix de la solution.....	18
2.8. Spécification de la solution	20
2.9. Evaluation technique des livrables.....	20
2.10. Choix des livrables	27
2.11. Etude et choix de la base de données	28
2.12. Evaluation financière des modules de la solution	31
2.13. Evaluation du coût total de la solution	32
2.14. Planning de réalisation.....	32
CHAPITRE 3 ANALYSE ET CONCEPTION	34

3.1. Présentation de la méthode d'analyse	35
3.2. Présentation de l'outil de modélisation [16]	39
3.3. Etude détaillée de la solution	40
CHAPITRE 4 RÉALISATION ET MISE EN ŒUVRE.....	58
4.1. Mise en œuvre	59
4.2. Présentation du système	73
CONCLUSION.....	85
REFERENCES BIBLIOGRAPHIQUES.....	86
TABLE DES MATIERES	89

LISTE DES PARTICIPANTS DU PROJET

Ci-dessous un tableau présentant l'équipe technique du projet.

Tableau 1 Liste des participants du projet

Noms et prénoms	Poste	Rôle
M. VEYATO AKPAKU Koami Hervé	Etudiant en 3ème année Génie Logiciel au Centre Informatique et de Calcul	Concepteur et réalisateur
M. AKAKPO Folly	Enseignant au Centre Informatique et de Calcul	Directeur de mémoire
M. DOFONNOU Mawouéna Kévin Modeste	Directeur de ALL Solutions	Maître de stage

LISTE DES TABLEAUX

Tableau 1 Liste des participants du projet	vii
Tableau 2 Volume de péage et précision des technologies existantes de paiement de péage [7].	19
Tableau 3 Tableau comparatif d'Oracle Database, MySQL, Maria DB et PostgreSQL.....	30
Tableau 4 Estimation financière des modules la solution.....	31
Tableau 5 Evaluation financière de la solution.....	32
Tableau 6 Résumé du planning de réalisation	33
Tableau 7 Liste descriptive des acteurs du système.....	42
Tableau 8 Description textuelle du cas « S'authentifier ».....	49
Tableau 9 Description textuelle du cas « Enregistrer un abonnement ».....	50
Tableau 10 Description textuelle du cas « Consulter l'état d'un abonnement ».....	51
Tableau 11 Récapitulatif des systèmes et leurs modules	59
Tableau 12 Matériel de développement.....	69

LISTE DES FIGURES

Figure 1 Organigramme administratif du CIC.....	6
Figure 2 Organigramme hiérarchique de ALL Solutions	10
Figure 3 Présentation du cycle itératif et incrémental	36
Figure 4 Logo UML.....	37
Figure 5 Logo Draw.io.....	40
Figure 6 Diagramme de contexte statique	41
Figure 7 Diagramme de déploiement.....	43
Figure 8 Diagramme des composants	44
Figure 9 Diagramme de paquetages.....	45
Figure 10 D0 d'utilisation du paquet « Administration ».....	46
Figure 11 Diagramme des cas d'utilisation du paquet « Partenaire »	47
Figure 12 Diagramme des cas d'utilisation du paquet « Client ».....	48
Figure 13 Diagramme des classes.....	52
Figure 14 Diagramme d'activités « Gestion des abonnements »	53
Figure 15 Diagramme d'activités « Interaction Client – Partenaire »	54
Figure 16 Diagramme d'activités « S'authentifier ».....	55
Figure 17 Diagramme des séquences représentant la consultation d'un abonnement	56
Figure 18 Diagramme des séquences de l'affichage des statistiques	57
Figure 19 Logo Visual Studio Code	60
Figure 20 Logo PHP	61
Figure 21 Logo Golang.....	61
Figure 22 Logo JavaScript.....	62
Figure 23 Logo HTML	62
Figure 24 Logo CSS.....	63

Figure 25 Logo Vue JS	64
Figure 26 Logo Laravel	64
Figure 27 Architecture MVC	65
Figure 28 Logo Bootstrap	65
Figure 29 Logo Ajax.....	66
Figure 30 Carte RFID sans contact.....	67
Figure 31 Logo Apache	68
Figure 32 Logo MySQL.....	68
Figure 33 Raspberry Pi 3	69
Figure 34 Architecture 3 tiers	70
Figure 35 Processus d'authentification avec JWT Token	72
Figure 36 Structure de l'espace d'administration	73
Figure 37 Structure de l'espace des partenaires	74
Figure 38 Structure de l'espace Client	74
Figure 39 Page d'authentification de l'espace d'administration	76
Figure 40 Tableau de bord de l'espace d'administration	76
Figure 41 Affichage de la liste des partenaires	77
Figure 42 Interface de vente d'abonnements à un partenaire.....	77
Figure 43 Interface de création d'abonnement	78
Figure 44 Interface d'ajout d'une carte RFID	78
Figure 45 Page des historiques de passages.....	79
Figure 46 Tableau de bord de l'espace des partenaires	80
Figure 47 Page d'enregistrement d'un client	80
Figure 48 Page présentant la liste des clients.....	81
Figure 49 Interface de vente d'abonnement	81

Figure 50 Interface des abonnements disponibles	82
Figure 51 Interface d'accueil du client	83
Figure 52 Interface présentant les offres d'abonnements	83
Figure 53 Interface des abonnements d'un client	84
Figure 54 Interface de paiement en ligne	84

GLOSSAIRE

Terme	Définition
API	En informatique, une interface de programmation applicative (souvent désignée par le terme API pour Application Programming Interface) est un ensemble de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels [10].
Back office	Une application de back office est la partie d'un système informatique qui n'est pas accessible aux utilisateurs finaux ou aux clients.
JWT	Le JWT (JSON Web Token) est un standard qui permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. [22]
ORM	Un mapping objet-relationnel est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par « correspondance entre monde objet et monde relationnel » [24].

Middleware	Un middleware ou un intergiciel est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques.
Péage	Un péage est une redevance pour circuler sur une route ou un pont, il est encore défini comme étant le lieu où l'on paye le droit de passage.
PIB	Le Produit Intérieur Brut est l'un des agrégats majeurs des comptes nationaux. Il reflète l'activité économique interne d'un pays et sa variation est censée mesurer le taux de croissance économique [9].
PND	Plan National de Développement adopté par le Togo le 03 Août 2018.
RFID	La radio-identification, le plus souvent désignée par le sigle RFID (de l'anglais Radio Frequency Identification), est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes » (« RFID tag » ou « RFID transponder » en anglais) [26].
SGBD	L'acronyme SGBD, pour Système de Gestion de Base de Données, désigne un logiciel informatique permettant le stockage, la consultation, la mise à jour, la structuration ou encore le partage d'informations dans une base de données [8].
TIC	Technologies de l'Information et de Communication.

INTRODUCTION

Au cours des deux dernières décennies, l'Afrique a doublé son PIB (Produit Intérieur Brut) et mis en œuvre des projets d'infrastructures d'envergure. Les pays de la sous-région Ouest Africaine ont su suivre le rythme de tels changements en obtenant, grâce à des plans stratégiques, des résultats significatifs en matière de progrès économique et social, au-dessus de la moyenne continentale sur de nombreux indicateurs [3]. Cette croissance du PIB peut être associée à la croissance du secteur des transports [3]. Les Etats investissent considérablement dans les infrastructures routières, conscients de leur importance. Des stratégies de financement sont alors adoptées pour la réhabilitation et l'entretien des réseaux routiers. L'une de ces stratégies est la mise en place des péages entraînant ainsi le développement des systèmes de paiement de péage.

Une Afrique qui fait face à des évolutions importantes en matière de démographie, de technologie, nécessite des politiques d'accompagnement ciblées. Le Togo a récemment adopté le Plan National de Développement (PND) qui ne fait aucun doute sur l'importance du secteur privé dans sa mise en œuvre. L'engagement de tous les acteurs étant la clé de l'atteinte des objectifs du PND, l'entreprise ALL Solutions s'engage dans la participation de l'axe stratégique 1 du PND. Cet axe est la mise en place d'un hub logistique d'excellence et un centre d'affaires de premier ordre dans la sous-région à travers une amélioration des infrastructures existantes et de la connectivité multimodale et des TIC [3]. Il met l'accent sur l'importance du développement des infrastructures de transport parce qu'elles fournissent le socle essentiel sur lequel se bâtissent le développement et la compétitivité de l'économie. Le défi majeur est la mobilisation des ressources nécessaires pour investir progressivement dans le réseau routier national. Il serait alors intéressant de renforcer l'efficacité du réseau de péages. Cependant, les systèmes de péage actuels ont un frein dans leur efficacité et rapidité d'exécution : des encombrements aux stations de péage, des accidents, les risques d'erreur humaine dans la fourniture des données de statistiques fiables.

Afin d'améliorer le système actuel de péage routier, ALL Solutions a jugé nécessaire de lancer le projet : « **Conception d'un système de paiement automatique de péage routier** ».

La réalisation du projet nous a été confiée dans le cadre de notre stage de fin de formation pour l'obtention de la Licence Professionnelle en Génie Logiciel. La formation reçue au CIC (Centre Informatique et de Calcul), nous permettra de réaliser le projet avec pour objectif principal l'automatisation des paiements au péage et le passage sans arrêt des véhicules.

Ce document est un résumé des travaux effectués au cours de nos dix (10) semaines de stage de fin de formation. Il est organisé en quatre (04) chapitres principaux à savoir :

- **Présentation des cadres de formation et de stage** : présente le cadre de formation et l'entreprise d'accueil ;
- **Etude préalable du projet** : permet de faire une présentation détaillée du thème de stage, de présenter sa problématique et enfin de comprendre l'intérêt d'apporter une solution à cette dernière ;
- **Analyse et conception** : présente l'analyse et la conception de la solution. Après avoir présenté la méthode d'analyse et l'outil de modélisation, une étude détaillée de la solution sera faite ;
- **Réalisation et mise en œuvre** : les moyens et outils de mise en œuvre ainsi que les résultats obtenus feront l'objet de ce chapitre.

CHAPITRE 1 PRÉSENTATION DES CADRES DE FORMATION ET DE STAGE

Il sera présenté dans cette section, la structure de formation et ensuite l'entreprise d'accueil.

1.1. Présentation du cadre de formation : CIC-UL [1] [2]

Le CIC –UL qui a servi de cadre pour notre formation sera présenté dans cette partie.

1.1.1. Historique

Le Centre Informatique et de Calcul (CIC) est un établissement et un centre de ressources de l'Université de Lomé créés par arrêté N°67/MENRS du 26 Septembre 1988. Il est situé dans l'enceinte de l'Université de Lomé (UL). Sa mission est d'apporter un appui logistique et technique en informatique aux établissements et à l'administration de l'Université de Lomé dans la mise en place de programmes informatiques et d'assurer la diffusion de l'informatique. A ce titre, le CIC réalise les applications informatiques au profit des institutions de la communauté universitaire et offre un cadre pour la formation en informatique à l'UL.

Le CIC dépend de la présidence de l'UL et a pour objectifs principaux :

- Développer des applications informatiques ;
- Résoudre les problèmes technologiques particulièrement dans le domaine de la maintenance informatique à l'UL ;
- Former des étudiants compétents en informatique.

Depuis quelques années le centre héberge en son sein l'académie Cisco qui forme à la préparation du certificat « Cisco Certified Network Academy (CCNA) », ainsi que le Laboratoire des logiciels libres (**Labtlic**) offert par l'Institut francophone des Nouvelles Technologies de l'Information et de la Formation (INTIF), à l'Association Togolaise des Utilisateurs de Logiciels Libres (ATULL) pour la promotion du Libre.

1.1.2. Les objectifs

Le CIC est un support en informatique aux établissements de l'Université de Lomé en ce sens qu'il organise :

- Des cours d'initiation ;
- Des cours de recyclages ;
- L'assistance aux chercheurs de l'université ;

- Des séminaires de formation ;
- L'assistance technique aux instituts (mise en place des systèmes d'information) ;
- L'élaboration et développement de projets.

En outre, le CIC assure la formation académique de ses étudiants.

1.1.3. Les activités

Au sein du CIC, nous avons cinq (5) grandes activités qui sont : la formation académique des étudiants, la formation du CISCO NETWORKING ACADEMY, l'ATULL, le développement d'applications et la maintenance des équipements informatiques.

- **La formation académique des étudiants**

Le recrutement au CIC s'effectue sur concours pour les étudiants nationaux et sur étude de dossier pour les étudiants étrangers. La durée de la formation est de six (6) semestres, pour l'obtention d'une Licence Professionnelle en Génie Logiciel ou en Systèmes et Réseaux avec une soutenance d'un mémoire de stage en fin de formation.

- **La formation CISCO NETWORKING ACADEMY**

Le Cisco Networking Academy Program est un cours de quatre (4) semestres ou modules de cours sur la technologie des réseaux, la configuration, la maintenance et la conception de réseaux. Il prépare à la certification CCNA (Cisco Certified Network Associate).

- **Le cycle Master**

L'ouverture du cycle Master a été effectuée dans le mois de mars 2016, en partenariat avec l'Université de Technologie de Belfort-Montbéliard (UTBM) en France et l'Institut Africain d'Informatique (IAI-TOGO).

- **Le développement d'applications et la maintenance des équipements informatiques**

Le développement d'applications assuré par la Division Développement est composé de développeurs et d'enseignants chercheurs.

La Division Maintenance s'occupe du maintien en état de bon fonctionnement des équipements informatiques.

1.1.4. Les perspectives

Le CIC projette de créer, un Laboratoire de Recherche en Informatique Appliquée (LRIA). Le LRIA devrait répondre à deux objectifs:

- Développer à l'Université de Lomé la formation et la recherche dans le domaine de l'informatique appliquée ;
- Promouvoir le travail des chercheurs du laboratoire et valoriser leurs travaux auprès des instances universitaires africaines et internationales.

1.1.5. Organisation administrative

Le Centre Informatique et de Calcul (CIC) est dirigé par un directeur assisté d'un directeur adjoint. La structure administrative du CIC est structurée comme ci :

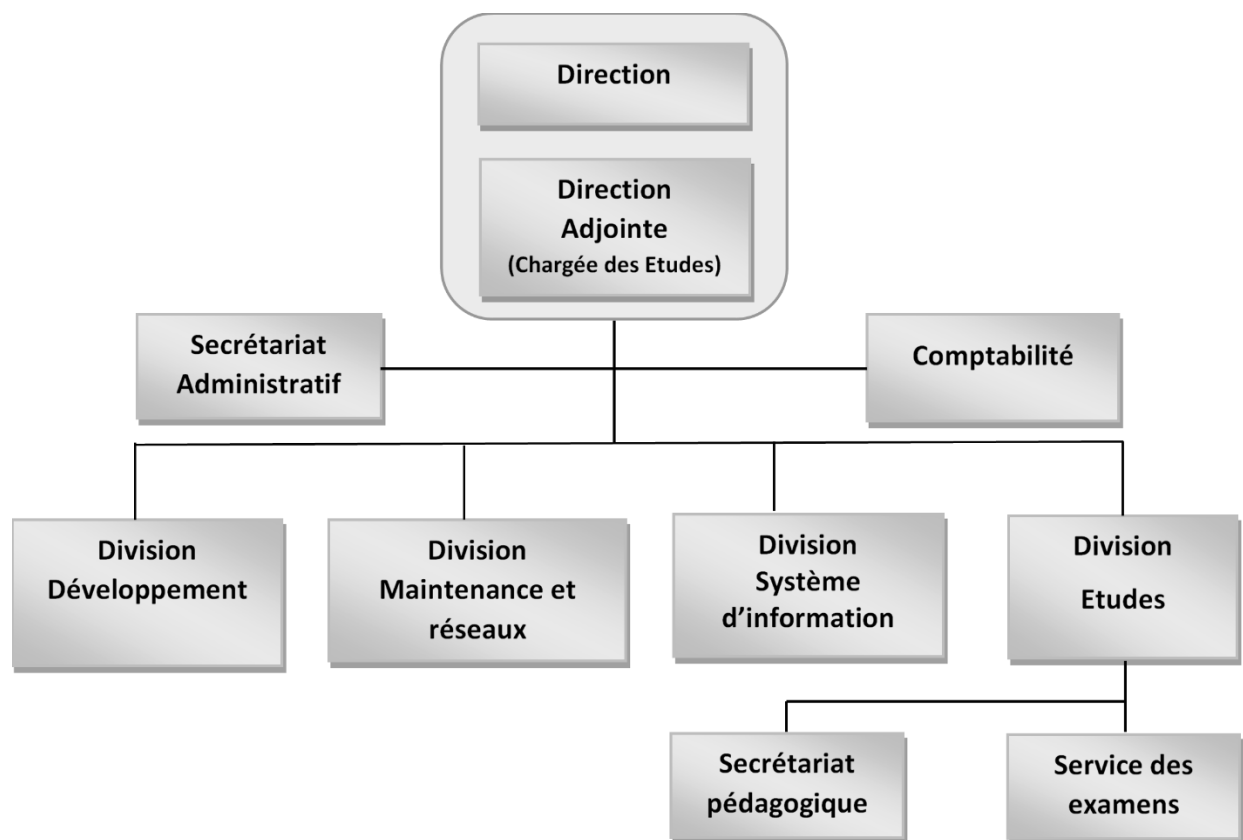


Figure 1 Organigramme administratif du CIC

Le Centre Informatique et de Calcul (CIC) dispose actuellement de quatre (4) divisions :

- **Division Etudes** en charge de la formation universitaire au grade Licence et Master.
- **Division Système d'information** chargée de l'étude, de la conception, de la mise en œuvre de la gestion du système d'information global de l'Université de Lomé.
- **Division Développement** chargée des études informatiques nécessaires à l'informatisation des différentes entités de l'Université de Lomé, de la conception et du développement des applications informatiques et de la maintenance des logiciels développés.
- **Division Maintenance et réseaux**, chargée de la maintenance des équipements informatiques de l'Université de Lomé, de la mise en œuvre des programmes d'assemblage de micro-ordinateurs initiés par le CIC, de la réalisation des différentes études nécessaires à l'informatisation de toutes les entités de l'Université de Lomé en collaboration avec les autres divisions.

1.2. Présentation du cadre de stage : Entreprise ALL Solutions

Il sera présenté l'entreprise ALL Solutions qui nous a servi de cadre de stage.

1.2.1. Description de l'entreprise

ALL Solutions est une société togolaise spécialisée dans le domaine des sciences et technologies. Fondée par un jeune entrepreneur, elle a orienté ses activités, en ne se contentant pas de gérer des projets classiques mais en misant sur des collaborateurs aux compétences et domaines d'activités variés. Elle regroupe des jeunes ingénieurs qui veulent utiliser leur savoir-faire pour participer au développement des pays de la sous-région en proposant des solutions et en offrant des formations. Il en résulte aujourd'hui, une grande expérience et une véritable ouverture d'esprit qui permet aux collaborateurs de ALL Solutions de proposer des solutions d'une grande qualité.

ALL Solutions est située à Adidogomé non loin de la Station T-Oil 1 d'Adidogomé, Lomé-Togo.

ALL Solutions dispose de jeunes ingénieurs et diplômés en :

- **Génie civil** : représente l'ensemble des techniques de constructions civiles. Les ingénieurs civils ou ingénieurs en génie civil s'occupent de la conception, la réalisation, l'exploitation et la réhabilitation d'ouvrages de construction et d'infrastructures dont ils assurent la gestion afin de répondre aux besoins de la société, tout en assurant la sécurité du public et

la protection de l'environnement. Très variées, leurs réalisations se répartissent principalement dans cinq grands domaines d'intervention : structures, géotechnique, hydraulique, transport, et environnement.

- **Génie électrique** : les ingénieurs électriques contribuent à la conception industrielle, la gestion et la mise sur pied de projets. Ils utilisent les mathématiques, des notions d'informatique, ainsi que leur habileté à comprendre des concepts technologiques pour parvenir à leurs fins.
- **Génie mécanique** : désigne l'ensemble des connaissances liées à la mécanique, au sens physique (sciences des mouvements) et au sens technique (étude des mécanismes). Ce champ de connaissances va de la conception d'un produit mécanique au recyclage de ce dernier en passant par la fabrication, la maintenance, etc.
- **Informatique** : est essentiellement concernée par la science de génie industriel qui étudie les méthodes de travail et les bonnes pratiques des ingénieurs qui développent des logiciels, le génie logiciel. Le génie logiciel s'intéresse en particulier aux procédures systématiques qui permettent d'arriver à ce que des logiciels de grande taille correspondent aux attentes du client, soient fiables, aient un coût d'entretien réduit et de bonnes performances tout en respectant les délais et les coûts de construction.

Dans un domaine en perpétuel mouvement, ALL Solutions anticipe les évolutions technologiques et innovations afin de proposer des produits de qualité. Convaincu de la vocation économique des nouveaux métiers et de la bonne stratégie d'entreprise, elle adopte une approche commerciale dès la conception des projets et propose un accompagnement au développement des projets.

L'entreprise projette de s'associer à des architectes, des commerciaux, des experts comptables, des juristes et des psychologues afin de créer un empire capable de trouver des solutions dans la plupart des domaines.

1.2.2. Mission de l'entreprise

ALL Solutions a pour mission de fournir aux entreprises et organismes publics, les solutions pour rendre plus efficace la gestion de leurs processus.

Les missions de ALL Solutions sont :

- Développer grâce au numérique les entités administratives ;
- Réaliser un audit informatique pour les entreprises ;
- Piloter les éléments opérationnels et stratégiques d'un système d'informations ;
- Gérer la sécurité informatique.

En se basant sur les résultats d'un audit bien détaillé ainsi que sur le cahier des charges imposé par le client, ALL Solutions s'occupe aussi de l'installation et de la maintenance des réseaux informatiques.

Elle trouve des solutions pour accroître la productivité des entreprises et améliorer la performance des équipes sans créer de rupture dans les processus de travail.

1.2.3. Les activités

L'entreprise ALL Solutions a réalisé les projets suivants :

- **Flash Document**

Le projet tire son origine de l'identification des difficultés rencontrées dans l'acquisition des casiers judiciaires dans un bref délai. Une plateforme a été développée pour mettre en relation les demandeurs de casier judiciaire, les agents informels et les sociétés de transport. Le projet permet à toute personne quel que soit son lieu de résidence de faire et de suivre une demande de casier judiciaire.

- **Diaspora Togo**

La difficulté de s'informer sur les études supérieures dans les différents pays a poussé les ingénieurs de ALL Solutions à créer une application d'interaction entre les étudiants togolais et ceux de la diaspora.

- **FSOS Violence**

Il s'agit d'une application mobile développée pour les femmes battues. Elle permet aux femmes de signaler des agressions et des tentatives de viol.

- **Conseil du citoyen**

ALL Solutions conseille, hebdomadairement via les réseaux sociaux, les citoyens sur le domaine fiscal et juridique.

1.2.4. Organigramme structurel de l'entreprise

L'organigramme de l'entreprise ALL Solutions est le suivant :

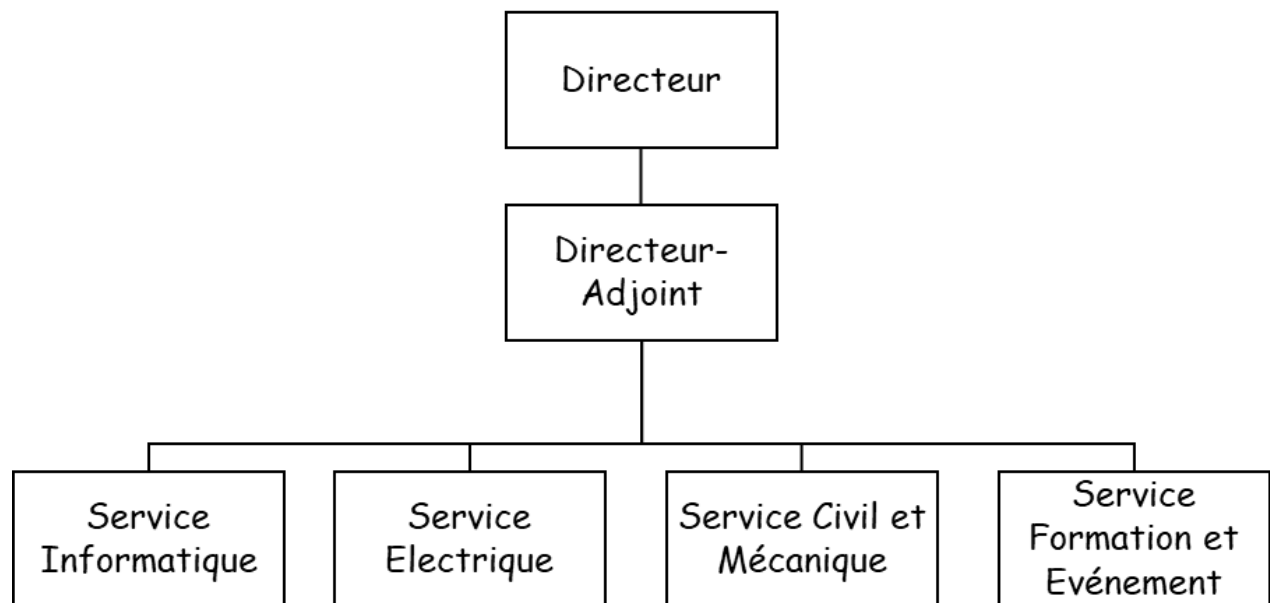


Figure 2 Organigramme hiérarchique de ALL Solutions

ALL Solutions est gérée par un directeur et son adjoint. Chaque service a un responsable. Elle dispose actuellement de quatre (4) services :

- **Service Informatique** en charge des projets informatiques. Ce service regroupe principalement des développeurs d'application. Ils sont chargés de la réalisation technique d'une application, basée sur un cahier de charge précis. Ils calculent et conçoivent des programmes informatiques pour le traitement des données ;
- **Service Electrique** chargé de l'étude, de la conception, de la mise en œuvre et de la gestion des systèmes électriques. Les ingénieurs de ce service mettent au point les équipements et les composants destinés au développement de nouveaux réseaux électriques ou au

remplacement des réseaux existants. Ils conçoivent les constructions de postes électriques et de systèmes de surveillance des réseaux, ainsi que les raccordements aux producteurs décentralisés d'énergie renouvelable ;

- **Service Civil et Mécanique** en charge des projets de constructions. Ce service a pour mission de concevoir et dimensionner de manière optimale des systèmes mécatroniques ou des unités mécaniques automatisées répondant au problème posé, en s'appuyant sur l'état de la technique, une étude ou une modélisation ; les évaluer compte tenu des différents paramètres du cahier des charges ;
- **Service Formation et Evénement** chargé de faire les formations et de gérer les événements. Il gère la logistique et la communication autour des projets et événements. Ce service exerce également dans la recherche de marchés, l'élaboration et la négociation des contrats.

Ces deux cadres, le CIC et ALL Solutions, ont été des facteurs importants dans la réalisation du projet. Le CIC a offert les compétences nécessaires à la réalisation du système et ALL Solutions a fourni un bon cadre de travail et un suivi.

CHAPITRE 2 ÉTUDE PRÉALABLE DU PROJET

Ce chapitre permet de faire une présentation détaillée du thème de stage, de présenter sa problématique et enfin de comprendre l'intérêt d'apporter une solution à cette dernière.

2.1. Présentation du sujet

Les routes jouent un rôle prépondérant dans le développement économique d'un pays. Leur importance, tant sur le plan de la densité que de la qualité, constitue le support préalable de tout décollage économique. C'est dans cette optique que de nombreux pays, aussi bien développés qu'en voie de développement, sont amenés à faire montre d'abnégation afin d'investir considérablement pour s'équiper en infrastructures routières. Le péage routier a été ainsi mis en place comme stratégie de financement de réhabilitation et d'entretien du réseau routier. Malheureusement, le système de péage instauré dans la plupart des pays ouest africains, est essentiellement manuel, vétuste, en déphasage avec les exigences de la modernité telles qu'appliquées dans certains pays et présente d'importantes irrégularités [4]. Ainsi, vu la contreperformance du système de péage actuel, nous avons pensé mettre sur pied le péage routier automatique : un système moderne qui permettra de réduire les dysfonctionnements observés dans nos péages actuels.

2.2. Problématique du sujet

Le système traditionnel de paiement de péage utilisé dans la sous-région ouest africaine est le « stop-and-go » [5]. Il utilise des moyens de paiement manuel et des barrières. Avec ce système, le conducteur doit s'arrêter devant la barrière pour payer en espèce puis retirer un ticket de passage. Par ailleurs, plus les véhicules arrivent en même temps, plus le temps d'arrêt pour effectuer le paiement augmente et donc les congestions arrivent avec les pics de trafic.

Outre ces problèmes qui freinent le processus de passage des conducteurs au péage, le système de paiement de péage « stop-and-go » engendre plusieurs autres problèmes parmi lesquels on peut noter [6] :

- Le retard accusé pour le retrait de monnaie ;

En moyenne 30 secondes par voiture [6], or le réseau routier représente 5,7 % du PIB Togolais (en 2017) [3].

- La difficulté de faire des comptes ;

Il arrive que les sommes réelles soient différentes des sommes enregistrées dans le système [6].

- Le risque d'accidents et de perte en vie humaine des agents de péage dû à la perte de contrôle des conducteurs ;
- La non existence d'un moyen de passage sans arrêt.

Face à cette situation, il est nécessaire de se poser les questions suivantes :

- Comment réduire le temps d'arrêt au péage ?
- Comment identifier chaque véhicule qui passe au péage ?
- Comment automatiser le paiement des péages ?
- Comment améliorer les performances du système actuel afin de fournir des historiques plus fiables de passages ?

L'enjeu du projet est donc d'apporter des solutions à ces différentes questions.

2.3. Intérêt du sujet

Afin de mieux comprendre l'intérêt du sujet, une description des objectifs (généraux et spécifiques) et des résultats attendus à la fin s'avère nécessaire.

2.3.1. Objectifs

Nous présenterons les objectifs généraux et spécifiques du projet.

2.3.1.1. Objectifs généraux

Le projet a pour but principal d'automatiser les paiements au péage via un système d'identification de véhicules.

2.3.1.2. Objectifs spécifiques

Le projet devra permettre à la fin de son exécution d'atteindre les objectifs spécifiques suivants :

- Les véhicules doivent pouvoir passer au péage avec un temps d'arrêt réduit ;
- Un véhicule doit pouvoir être identifié et associé à une personne en sa charge ;
- Un conducteur doit pouvoir avoir un abonnement de passages en fonction de ses besoins ;

- Le système devra permettre de gérer les partenaires de l'institution officielle de gestion des péages ; partenaires désireux d'offrir des services d'abonnements aux conducteurs ;
- Le système devra fournir des informations en temps réel sur les passages au péage.

2.3.2. Résultats attendus

A la fin du processus de développement, l'ensemble des solutions devra permettre :

- De réduire le temps du processus de passage au péage ;
- De faciliter le suivi des passages ;
- D'avoir des statistiques de paiement en temps réel ;
- D'avoir un système stable et fiable.

2.4. Etude de l'existant

Il sera parcouru dans cette partie l'essentiel du système de péage utilisé dans la sous-région sans y apporter une critique.

Deux systèmes seront étudiés :

- Le système de gestion de véhicules ;
- Le système de back office.

2.4.1. Le système de gestion des véhicules

Le système de paiement de péage instauré utilise des moyens de paiement manuel et des barrières. Ce système a besoin d'employés de péage. Le conducteur s'arrête devant la barrière, paye son ticket de péage et l'employé de péage fait lever la barre mécaniquement pour lui permettre de passer. L'historique général des passages est enregistré au système de back office.

2.4.2. Le système de back office

Un système gère en back office la génération des tickets de péage et permet d'enregistrer à chaque caisse la date et l'heure de passage, le type de véhicules et le montant payé. Ce système fournit la somme totale des passages par caisse de manière incrémentale.

Les processus aux péages actuels sont gérés et permettent d'atteindre l'objectif des péages routiers tant bien que mal.

Cette situation est loin d'être idéale et des améliorations peuvent y être apportées pour plus d'efficacité.

2.5. Critique de l'existant

La section précédente a permis de décrire le système existant. Toutefois, cette solution présente des insuffisances que cette nouvelle section aborde sous forme de critique constructive.

Le système de gestion de véhicules et le système de back office feront l'objet de notre critique.

2.5.1. Le système de gestion des véhicules

L'un des points qui pose le plus de problèmes est la non identification des véhicules. Vu le passage régulier de certains véhicules, il serait préférable de les identifier afin de leur permettre d'avoir des abonnements. Le système actuel ne permet pas de passer sans arrêt, ce qui pourrait être très utile pour les ambulances et les voitures d'Etat. Par ailleurs, il est fréquent d'avoir des encombrements au péage pour des retraits de monnaie. Cette situation pourrait être évitée si les paiements étaient rassemblés en abonnements et effectués d'avance avec le système de back office.

2.5.2. Le système de back office

Des milliers de tickets de passage sont générés par jour avec le système de back office. Les tickets de péage sont en petites coupures et donc difficiles à collecter et à recycler ; alors que la fabrication d'une tonne de papiers neufs nécessite en moyenne deux (2) à trois (3) tonnes de bois [4]. Cette situation est un frein au développement du pays et en conflit avec les exigences environnementales. Quant à la fiabilité des rapports fournis par les agents de péage, elle peut être mise en cause tant le risque d'erreur humaine est élevé. Il arrive en effet, que la somme totale enregistrée dans le système soit différente de la somme en caisse [6]. Plus les véhicules sont nombreux, plus la marge d'erreur est élevée.

Cette critique de l'existant a permis de relever les insuffisances et défaillances des principaux processus du système actuel de péage dans la sous-région, le « stop-and-go ». Face à ces limites énumérées, quelles sont les solutions envisageables pour améliorer l'existant ?

2.6. Proposition de solutions

Suite à l'étude de l'existant et la problématique soulevée précédemment, plusieurs solutions sont envisageables pour rendre plus efficace le système de péage routier.

Ces solutions sont entre autres :

- Le système de paiement de péage « stop-and-go » amélioré ;
- Le système de paiement automatique ou télépéage par reconnaissance de plaque d'immatriculation par caméra ;
- Le système global de navigation par satellite ;
- Le système de téléphone cellulaire ;
- Le système de paiement automatique par radio fréquence.

Une description de ces systèmes s'avère nécessaire.

2.6.1. Le système de paiement de péage « stop-and-go » amélioré

Ce système utilise aussi des moyens de paiement manuel mais ne se limite pas au paiement d'encaisse. Les paiements peuvent se faire par crédit ou carte intelligente ou par des coupons. L'historique de passages des véhicules est enregistré au système de back office. Le conducteur s'arrête pour effectuer son paiement mais il n'y a pas de retrait de monnaie.

2.6.2. Le système de paiement automatique ou système de télépéage

Afin de réduire les temps de paiement et minimiser les risques d'erreurs humaines, les moyens de paiement automatique ont vu le jour. Ils permettent d'effectuer des paiements de péage sans arrêt.

2.6.2.1. Système de télépéage par reconnaissance de plaque d'immatriculation par caméra

Au passage d'un véhicule, la plaque d'immatriculation est détectée par une caméra. L'immatriculation permet de prélever les frais de péage sur les comptes des conducteurs. Le système de back office permet d'avoir des statistiques sur les véhicules. Ce système est plus utilisé dans les pays où l'utilisation des comptes en banque est fréquente.

2.6.2.2. Système Global de Navigation par Satellite (GNS)

Dans ce système, chaque véhicule doit avoir un badge, ce badge comporte un module de GPS (Global Positioning System) et une donnée sur la classification du véhicule. Le véhicule se localise grâce au module GPS du badge et il est enregistré sur un serveur. Quand le véhicule passe la station de péage, la donnée de classification du véhicule est déclarée ; c'est un signal pour que le paiement soit envoyé.

2.6.2.3. Système de téléphone cellulaire

Un téléphone cellulaire est installé dans chaque véhicule, le véhicule est localisé grâce à la communication fréquente entre le cellulaire du véhicule et la station de base (point d'accès) du réseau. Le moindre coût et la facilité d'installation du téléphone cellulaire sur le véhicule sont les avantages de ce système. Mais chaque station de base gère les transmissions de nombreux terminaux cellulaires dans sa zone d'opération donc le fonctionnement en zone mal desservie est discutable.

2.6.2.4. Système de paiement automatique par radio fréquence

Ce système se compose d'un badge (ou étiquette) qui est fixé sur le pare-brise du véhicule, d'un lecteur qui est fixé sur un portique au-dessus de la route. Toutes les informations du véhicule : le type, la plaque d'immatriculation, l'adresse et un compte du propriétaire de véhicule sont stockés dans le badge. Quand le véhicule arrive vers la zone d'opération du lecteur, il est détecté puis il communique au lecteur les informations stockées sur le badge. Les informations de la transaction sont enregistrées au système de back office et participe à l'historique du véhicule pour cette station afin d'augmenter la sécurité des transactions futures. Ce moyen permet de réaliser une transaction de paiement sans que le véhicule ne s'arrête. Les conducteurs peuvent passer en gardant une certaine vitesse de circulation.

2.7. Choix de la solution

L'objectif poursuivi est l'automatisation des paiements de péage, un passage avec un temps d'arrêt réduit des véhicules et la collecte de données fiables. Notre choix portera alors sur un système de paiement automatique ou télépéage. Les réalités de notre sous-région nous obligeront à choisir un système paramétrable et ayant le plus de précision.

Afin de faire un choix objectif, nous allons nous référer au tableau suivant présentant un bilan au niveau de la précision, du prix et de la capacité de traitement de véhicules par heure (VPH) en correspondance avec les technologies existantes du système de paiement de péage routier 2006 par World Bank [7].

Tableau 2 Volume de péage et précision des technologies existantes de paiement de péage [7]

Moyen de paiement de péage	Nombre de véhicules par heure (VPH)	Prix par transaction (F CFA)	Précision (% de réussite d'une transaction)
Manuel ou « stop-and-go »	250-350	192.5 -247.5	98.0
Machine avec barrière	450-550	154-192.5	98.5
Reconnaissance d'identification de la plaque d'immatriculation	600-1000	1237.5	85.0
Carte intelligente	700-900	55 -104.5	99.5
Système de paiement automatique par RFID ou radio fréquence	900-1100	55 -72.05	99,96

Ce tableau montre l'avantage primordial du système de paiement automatique via la technologie de radio fréquence RFID (Radio Frequency Identification) en ce qui concerne la précision, l'économie et le flux de traitements.

Ce système permettra de mettre en relation des partenaires désireux d'être des fournisseurs de services en télépéage et l'institution officielle de gestion des péages routiers. Il pourra ainsi être arrimé à des plans stratégiques de développement dans les pays en voie de développement notamment le PND du Togo qui a pour objectif global de transformer structurellement l'économie, pour une croissance forte, durable, résiliente, inclusive, créatrice d'emplois décents et induisant l'amélioration du bien-être social.

Nous opterons alors pour un système de paiement automatique via un système d'identification de véhicules par la technologie RFID.

2.8. Spécification de la solution

La solution adoptée englobe plusieurs livrables qui seront chacun soumis à une évaluation technique et financière. Ces livrables sont :

- Une application d'administration permettant la gestion des partenaires (entités offrant des services de vente d'abonnements et de gestion des clients), la création des abonnements et la fourniture de statistiques détaillées sur les processus aux stations de péage ;
- Une application dédiée aux partenaires permettant la gestion des clients et leur interaction avec les partenaires (abonnements, paiements, soldes) ;
- Une application web disponible pour les clients (conducteurs et/ou propriétaires de véhicules), leur permettant de consulter les offres et l'état de leurs abonnements ;
- Un système d'identification de véhicules par RFID tournant sur une base de données centrale contenant les informations sur les tags et leur éligibilité ;
- Une API de communication entre l'application d'administration, l'application des clients, l'application des partenaires, le système d'identification de véhicules et la base de données.

2.9. Evaluation technique des livrables

Le choix des technologies à utiliser est essentiel pour la qualité et la performance des livrables à développer. Ainsi, il est important de faire une analyse approfondie des solutions techniques qui seront adoptées pour réaliser les différents modules de nos livrables.

2.9.1. Le système d'identification de véhicules via RFID

Il existe plusieurs familles de systèmes RFID dont le principal critère de différenciation est la fréquence de fonctionnement. Les badges RFID sont divers :

- Les cartes ;
- Les puces électroniques ;
- Les boîtiers.

Dans le cadre de notre projet, il est nécessaire de développer un moyen de vérification de l'éligibilité du badge à utiliser et capable de transmettre des informations et/ou signaux aux barrières et moteur des infrastructures du péage.

2.9.2. L'application d'administration

L'application peut être une application desktop ou une application web. Nous allons présenter les deux solutions possibles avec leurs diverses caractéristiques.

2.9.2.1. Une application desktop

Une application desktop est une application qui fonctionne de manière autonome sur un ordinateur de bureau ou portable. Contraste avec «application basée sur le Web », qui exige que le navigateur Web s'exécute. Les applications s'exécutent en utilisant les services du système d'exploitation pour utiliser les ressources matérielles.

Une application desktop présente de meilleures performances par rapport au temps de calcul, de réponse. Elle peut avoir aussi un niveau de sécurité élevé. Cependant pour être utilisée, l'application doit être déployée sur tous les postes des utilisateurs. La mise à jour s'avère compliquée.

2.9.2.2. Une application web

Une application web est une application manipulable grâce à un navigateur web. Une application web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique.

Dans la technologie la plus courante, l'application web s'oriente autour d'un serveur web sur lequel est branché le logiciel applicatif, le tout parfois accompagné d'un serveur de base de données. L'ensemble est appelé serveur d'applications.

Le code source du logiciel applicatif est placé directement dans des pages web. Ces pages sont stockées par le serveur. Lorsque le client demande une page, le serveur web va rechercher la page, puis exécute les instructions qu'elle contient. Ces instructions peuvent faire appel au serveur de base de données. Le serveur web transmet la page avec le résultat de l'exécution au client.

La transmission des informations entre le client et le serveur se fait selon le protocole HTTP, protocole également utilisé pour les sites web. Ce qui permet d'utiliser le même logiciel client : un navigateur web.

L'application web peut être réalisée en utilisant trois principales technologies à savoir :

- Un gestionnaire de contenu (CMS) ;
- Une application en un langage de programmation pur ;
- Une application réalisée grâce à un Framework.

2.9.2.3. Une application web réalisée avec un CMS

Un gestionnaire de contenu ou CMS (Content Management System) est un outil clés en main de base qui contient tout ce dont l'on a besoin pour gérer un site Web directement à l'aide de l'interface qu'il propose. On parle donc d'une section d'administration permettant, entre autres, à n'importe quel utilisateur, quel que soit son niveau de connaissance, de gérer rapidement et simplement le contenu d'un site ou certains de ses éléments visuels. Un autre avantage d'un tel outil est qu'il propose une multitude de plug-ins ou plugiciels en français (bouts de code déjà construits et aisément intégrables de telle sorte qu'on n'a pas à refaire ce que quelqu'un d'autre a déjà construit). Ces plugiciels permettent l'ajout de fonctionnalités/thèmes (apparences visuelles) facilement sans avoir à modifier le code.

Avantages

- Outil clés en main ;
- Facile d'accès ;
- Simpliste ;
- Plugiciels (plug-ins) ;
- Demande peu de connaissances en programmation.

Inconvénients

- Le contrôle peut être restreint ;
- Alourdissement du site (il est plus lent sur le plan de l'exécution).

2.9.2.4. Une application web depuis zéro

Développer une application web « from scratch », c'est assez simple, on part d'une feuille blanche. Il faut tout faire. Ce cas d'utilisation est assez restreint et on comprend assez vite pourquoi il faut recommencer du début à chaque fois. L'utilisation de cette méthode sera donc recommandée lorsque le site Web sera vraiment simple, comprendra peu de pages et très peu de fonctionnalités.

Avantages

- Entièrement personnalisable ;
- Contrôle total ;
- Aucune restriction ;
- Rapidité (aucun temps système).

Inconvénients :

- Beaucoup d'heures-personnes ;
- Requiert de bonnes connaissances en programmation.

2.9.2.5. Une application web avec un Framework

Un cadre d'application (Framework) est tout simplement un regroupement de bibliothèques de programmes pour différents champs d'application qui permettent d'effectuer certaines actions. Par exemple, une bibliothèque permettra l'accès à une base de données et une autre l'affichage à l'écran ou encore, la gestion des journaux d'activités (logs) de notre application en devenir.

En tant que tel, un cadre d'applications ne peut rien accomplir seul. Il faut consacrer des « heures personnes » pour créer une structure cohérente où tout fonctionne ensemble pour faire fonctionner votre site Web. En fin de compte, l'avantage ici est que vous avez le contrôle total sur ce qui est fait, car vous choisissez quoi faire et comment le faire. Il existe des plugins pour éviter de partir de zéro.

Avantages

- Entièrement personnalisable ;
- Contrôle accru ;
- Aucune restriction ;

- Plugiciels (plug-ins) ;
- Rapidité (peu de temps système).

Inconvénients

- Heures-personnes requises ;
- Requiert de bonnes connaissances en programmation.

2.9.3. L'application de gestion des processus des partenaires

L'application peut être une application desktop ou une application web. Nous nous baserons sur les caractéristiques décrites dans les titres précédents pour faire un choix.

2.9.4. L'application web : l'espace client

Nous proposons trois méthodes de développement pour réaliser l'application web qui sera l'espace des clients, permettant aux conducteurs de consulter l'état de leurs abonnements et de consulter les offres. Ces méthodes consistent à développer soit une application à l'aide d'un gestionnaire de contenu (CMS), soit une application web en un langage de programmation pur ou bien utiliser un Framework.

2.9.5. L'API de communication

L'API, pour Application Programming Interface, est la partie d'un programme qu'on expose officiellement au monde extérieur pour manipuler celui-ci. L'API est au développeur ce que l'UI est à l'utilisateur : de quoi entrer des données et récupérer la sortie d'un traitement. Initialement, une API regroupe un ensemble de fonctions ou méthodes, leurs signatures et ordre d'usage pour obtenir un résultat. [10]

La mise en place d'une API permet d'opérer une séparation des responsabilités entre le client et le serveur. Cette séparation permet donc une portabilité et évolutivité grandement améliorées.

Ainsi pour la conception de l'API de notre solution, nous proposons deux solutions :

- Le protocole SOAP (Simple Object Access Protocol);
- L'architecture REST (Representational State Transfer).

2.9.5.1. Une API SOAP [11]

SOAP (ancien acronyme de Simple Object Access Protocol) est un protocole d'échange d'information structurée dans l'implémentation de services web bâti sur XML. Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP. Le protocole SOAP est composé de deux parties :

- Une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement ;
- Un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

Avantages

Utiliser SOAP via HTTP facilite la communication et évite les problèmes de proxys et pare-feu par rapport à des technologies plus anciennes.

SOAP est :

- Assez ouvert pour s'adapter à différents protocoles de transport ;
- Indépendant de la plate-forme et du langage ;
- Extensible.

Inconvénients

- SOAP décrit la manière dont les applications doivent communiquer entre elles, certains considèrent que le couplage reste fort entre le serveur et ses clients. Une modification de l'API implique ainsi une évolution côté client, contrairement à une architecture orientée ressources telle que REST.
- En raison du nombre d'informations qu'impose le format XML, SOAP peut alourdir considérablement les échanges par rapport à certains middlewares.

2.9.5.2. Une API RESTful [12]

RESTful ou REST est l'acronyme de Representational State Transfer. Ceci est un style d'architecture constitué d'un ensemble de conventions et de bonnes pratiques à respecter permettant la construction d'applications (web, intranet ou encore web service). Le standard REST a été créé en 2000 par Roy Fielding dans sa dissertation "Architectural Styles and the Design of Network-based Software Architectures" et utilise les spécifications originelles du protocole HTTP, plutôt que de réinventer une surcouche comme ses concurrents (SOAP, XMLRPC). L'échange est donc basé sur des requêtes client et serveur c'est-à-dire qu'un client lance une requête HTTP, et le serveur renvoie une réponse. En plus de cela, des méthodes qui définissent les requêtes HTTP que le client peut effectuer telles que GET, PUT, POST, DELETE et bien plus encore.

Pour qu'une architecture soit considérée comme REST, elle doit respecter les six contraintes suivantes :

- **Orienté client-serveur**, c'est-à-dire que l'interface utilisateur est séparée de celle du stockage des données permettant aux deux d'évoluer indépendamment l'un de l'autre ;
- **Sans état**, chaque transaction, ou chaque action doit être traitée de manière indépendante. L'architecture REST ne permet pas de définir un flux d'action. C'est au client utilisant cette architecture de la définir comme il le souhaite ;
- **Mise en cache**, les clients peuvent mettre en cache les réponses. Le serveur doit donc donner des informations sur ses réponses pour permettre au client de savoir si celles-ci doivent être conservées dans le futur ;
- **Une interface uniforme**, identification des ressources (par exemple par un URI), manipulation des ressources à travers des représentations, un message autodéscriptif et l'utilisation d'hypermédia ;
- **Un code à la demande** (optionnel), offrir la possibilité pour les clients d'exécuter des scripts obtenus depuis le serveur.

Avantages

- **Séparation entre le client et le serveur** : L'architecture REST sépare totalement l'interface de l'utilisateur du serveur et de la base de données. Cela augmente la portabilité de l'interface à d'autres plateformes et augmente l'extensibilité des projets ;

- Indépendance vis-à-vis du type de plateforme ou des langages ;
- Simple à maintenir à cause de sa structure.

2.10. Choix des livrables

L'évaluation technique des livrables proposés a permis d'avoir les cartes en main pour choisir la solution la plus adaptée aux besoins et réalités de la sous-région. Dans cette partie, les choix de solutions pour chaque type de livrable seront faits.

2.10.1. Le système d'identification de véhicules via RFID

Nous allons préférer une carte RFID aux autres badges parce qu'elle est plus adaptée aux réalités de notre sous-région. Elle requiert moins de dépenses (humain et matériel). De plus, une carte RFID est facile à coller sur le pare-brise, à l'intérieur des véhicules.

Avec cette carte, les signaux électromagnétiques de la communication entre le lecteur et badge ne sont pas perturbés. Il existe des parebrises revêtus de métal qui rendent la communication moins précise. Le pourcentage de voitures avec de tels parebrises est très faible, seulement 0.5 pour cent de tous les véhicules [4].

Ce système fonctionnera avec un lecteur RFID capable de détecter une carte RFID à distance. Des signaux seront envoyés aux infrastructures électroniques du péage afin de lever les barrières de péage à l'éligibilité d'une carte.

2.10.2. L'application d'administration

Une application Desktop doit être déployée sur tous les postes des utilisateurs, pour être utilisée. Par conséquent elle offre un coût de maintenance beaucoup plus élevé par rapport aux applications web. Une application web serait plus adaptée dans notre cas et pour ce faire nous avons le choix entre partir de zéro ou utiliser un Framework pour sa réalisation. Nous faisons le choix d'utiliser un Framework pour la sécurité et le gain de temps qu'offre ce dernier.

2.10.3. L'application de gestion des processus des partenaires

Nous développerons une application web avec un Framework pour les mêmes raisons énoncées dans le titre précédent.

2.10.4. L'application web : l'espace client

La nécessité d'avoir un contrôle total sur l'application web exclut de facto le CMS comme moyen de réalisation de ce livrable. Il nous reste donc à choisir entre une application réalisée « from scratch » ou avec un Framework.

Il sera utilisé un Framework pour le gain de temps qu'il offre. L'utilisation d'un Framework renforce aussi la sécurité de l'application ; une des caractéristiques importantes des besoins de la future solution informatique.

2.10.5. L'API de communication

Nous avons choisi l'architecture REST pour la conception de l'API de communication entre la base de données, les applications de gestion de processus et l'application web de notre solution. Ce choix se justifie par les caractéristiques qu'offre cette architecture et sa modernité par rapport à SOAP. En effet, REST est facile à comprendre, utilise le protocole HTTP et les opérations basiques de CRUD (Create, Read, Update, Delete). Il supporte aussi plusieurs formats de données avec une prédominance du format JSON ce qui veut dire un meilleur support des navigateurs des clients.

2.11. Etude et choix de la base de données

Nous ferons un choix de la base de données à utiliser après une présentation des propositions de base de données.

2.11.1. Présentation des bases de données

Nous proposons d'utiliser un Système de Gestion de Base de Données Relationnelle (SGBDR). Il existe une multitude de SGBDR mais il sera présenté quatre (4) pour étude à savoir :

- Oracle Database ;
- MySQL ;
- Maria DB ;
- PostgreSQL.

Nous présenterons chacun d'entre eux en décrivant leurs caractéristiques techniques.

2.11.1.1. Oracle Database

Oracle Database dit Oracle est un système de gestion de base de données relationnelle (SGBDR) fourni par Oracle Corporation. Il utilise le langage PL/SQL. C'est le SGBDR qui occupe la première place sur le marché. Il est possible de télécharger gratuitement le serveur Oracle à des fins de développement mais le paiement intervient à la mise en production.

2.11.1.2. MySQL

MySQL est un Système de Gestion de Base de Données Relationnelle, qui utilise le langage SQL. L'une des spécificités de MySQL, c'est qu'il inclut plusieurs moteurs de bases de données et qu'il est par ailleurs possible au sein d'une même base de définir un moteur différent pour les tables qui composent la base. Cette technique est astucieuse et permet de mieux optimiser les performances d'une application.

La réplication est possible avec MySQL et permet ainsi de répartir la charge sur plusieurs machines, d'optimiser les performances ou d'effectuer facilement des sauvegardes de données.

2.11.1.3. Maria DB

Maria DB est un Système de Gestion de Base de Données (SGBD) disponible sous licence GPL. Ce système est un fork de MySQL, ce qui signifie que c'est un nouveau logiciel créé à partir du code source de MySQL.

2.11.1.4. PostgreSQL

PostgreSQL est un Système de Gestion de Base de Données Relationnel et Objet (SGBDRO) libre disponible sous licence BSD. PostgreSQL est un système de gestion de base de données relationnel-objet qui fonctionne sous plusieurs systèmes d'exploitation dont Linux, Mac OS X, Windows (depuis la version 8.0), FreeBSD, Solaris.

2.11.1.5. Tableau comparatif des divers SGBD

Le tableau suivant permet de faire une comparaison des divers SGBD proposés. Il permet de donner les avantages et inconvénients de chaque SGBD.

Tableau 3 Tableau comparatif d'Oracle Database, MySQL, Maria DB et PostgreSQL

SGBD	Avantages	Inconvénients
Oracle Database	<ul style="list-style-type: none"> • Dispose des dernières innovations et technologies répondant aux besoins des utilisateurs ; • Système de sécurité sophistiqué; • Supporte bien les montées en charge. 	<ul style="list-style-type: none"> • Coût très élevé surtout pour les petites entreprises ; • Le système requiert de grandes ressources serveur une fois installé ; • Requiert une infrastructure matérielle adaptée.
MySQL	<ul style="list-style-type: none"> • Disponible gratuitement; • Rapide et peu gourmand en ressources ; • Chaque table peut avoir un moteur de stockage différent ; 	<ul style="list-style-type: none"> • Pas de support natif de XML et OLAP.
Maria DB	<ul style="list-style-type: none"> • Rapide et stable ; • Cryptage des données et architecture extensible avec des plug-ins qui permettent de faire des customisations selon les besoins. 	<ul style="list-style-type: none"> • Les modifications sur les tables avec de grandes quantités de données causent des soucis
PostgreSQL	<ul style="list-style-type: none"> • Il est scalable et peut gérer des téraoctets de données ; • Supporte le type JSON ; • Dispose d'une variété de fonctions prédéfinies. 	<ul style="list-style-type: none"> • Documentation pas très explicite et la configuration peut être compliquée/

2.11.2. Choix de la base de données

Notre base de données ne correspond pas aux standards d'une grande base de données. Oracle Database est surtout destiné aux très grandes bases de données avec des milliards de lignes. Il n'est donc pas nécessaire de payer une licence d'exploitation et d'allouer des ressources considérables pour notre base de données.

Oracle étant hors-jeu, il reste à faire le choix entre les solutions open-sources plus précisément PostgreSQL et MariaDB (MySQL). PostgreSQL serait une bonne alternative à Oracle Database cependant elle est plus ou moins complexe sur le plan technique.

Finalement MySQL et MariaDB sont les seules options qui restent à notre disposition. Nous faisons le choix de MySQL car il est assez paramétrable et très compatible avec les outils à utiliser et surtout pour le monopole de connaissance que nous avons sur ce dernier.

2.12. Evaluation financière des modules de la solution

Après avoir fait l'évaluation technique et un choix des livrables, nous présentons une estimation financière des différents modules de notre solution. [13]

Tableau 4 Estimation financière des modules la solution

	L'application d'administration	L'application des partenaires	L'espace Client	Le système d'identification	L'API de communication
Nombre d'heures	60 x 8 (480)	60 x 8 (480)	20 x 8 (160)	50 x 8 (400)	60 x 8 (160)
Valeur humaine	3000	3000	3000	2000	3000
Coût humain	1 440 000	1 440 000	480 000	800 000	480 000
Coût matériel	386 450	386 450	386 450	18 000	386 450
TOTAL (FCFA)	1 826 450	1 826 450	866 450	818 000	866 450

2.13. Evaluation du coût total de la solution

Le tableau suivant résume l'évaluation financière du projet.

Tableau 5 Evaluation financière de la solution

Désignation	Coût (F CFA)
La base de données	446 450
Le système d'identification de véhicules	818 000
L'API de communication	866 450
L'application web des clients	866 450
L'application des partenaires	1 826 450
L'application d'administration	1 826 450
TOTAL (F CFA)	6 650 250

Après évaluation financière, le coût total de la solution est **Six Millions Six Cent Cinquante Mille Deux Cents Cinquante Francs (6 650 250) CFA**.

Les choix de solutions étant faits, nous sommes convaincus que ces choix techniques aideront à accomplir notre mission. L'exécution de ces solutions a suivi un programme précis que nous présenterons dans la partie suivante.

2.14. Planning de réalisation

Les différentes tâches exécutées ainsi que leurs durées respectives ont été consignées dans le tableau qui suit.

Tableau 6 Résumé du planning de réalisation

		Prévisionnel			Réalisation
Tâches		Début	Fin	Durée (jours)	Durée (jours)
1.	Phase préliminaire	01/07/2019	13/07/2019	12	12
2.	Modélisation UML	15/07/2019	20/07/2019	6	8
3.	Conception de la base de données	15/07/2019	22/07/2019	7	7
4.	Développement du système d'identification de véhicules	23/07/2019	29/07/2019	6	10
5.	Développement de l'API	30/07/2019	09/08/2019	10	14
6.	Développement des applications de gestion de processus	12/08/2019	24/08/2019	13	18
7.	Développement de l'espace client	25/08/2019	28/08/2019	4	5
8.	Phase de tests	28/08/2019	31/08/2019	4	7
9.	TOTAL	01/07/2019	31/08/2019	60	74

L'écart entre la prévision de l'exécution des tâches et leur réalisation est de quatorze (14) jours. Cet écart est dû à une prévision de la marge de manœuvre en deçà des réalités de l'exécution du projet.

CHAPITRE 3 ANALYSE ET CONCEPTION

Dans le processus de réalisation d'une application informatique, il est nécessaire, voire impératif de faire une analyse préalable de ce qu'on aura à réaliser.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse et de sécurité. La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Le choix d'un langage de modélisation moderne et efficace, répondant aux besoins actuels s'avère judicieux pour le développement de notre application. Ce chapitre sera consacré à la présentation des étapes de cette phase de modélisation.

3.1. Présentation de la méthode d'analyse

L'analyse et la conception des solutions informatiques ont connu deux (2) approches majeures au fil du temps :

- L'approche procédurale ;
- L'approche objet.

L'approche procédurale repose sur un paradigme de programmation considérant les différents acteurs d'un système comme des objets pratiquement passifs qu'une procédure centrale utilisera pour une fonction donnée.

Cependant, l'évolution et la complexité croissante des systèmes informatiques font fléchir cette approche devant certaines contraintes spécifiques qui s'imposent. Pour remédier à ces insuffisances, l'approche objet vit le jour.

L'approche objet s'appuie sur la manipulation des objets à partir de l'ensemble formé par les langages et les méthodes. Elle présente d'énormes avantages :

- Facilité d'organisation ;
- Réutilisation et méthode plus intuitive ;
- Possibilité d'héritage ;
- Facilité de correction ;
- Projets plus faciles à gérer.

L'intérêt principal réside dans le fait que l'on ne décrit plus par le code des actions à réaliser de façon linéaire mais par des ensembles cohérents appelés objets. L'approche objet offre d'une part une vision externe définissant les actions qu'il sera possible de mener sur le logiciel et d'autre part une vision interne dans laquelle seule la structure du logiciel est prise en compte.

L'approche objet sera utilisée pour l'étude détaillée de notre projet. En conséquence, parmi les multiples méthodes, nous utiliserons le modèle itératif et incrémental couplé au langage UML (Unified Modeling Language) dans sa version 2.0. L'approche itérative et incrémentale, car elle gère la segmentation du travail et la concentration sur les besoins et les risques, offrant ainsi des avantages au niveau du planning et des charges de travail.

3.1.1. Présentation du modèle itératif et incrémental [14]

L'approche itérative et incrémentale se résume comme suit :

- **Itératif** : le processus de développement est appliqué plusieurs fois. Le développement itératif aide à améliorer la qualité du produit ; une itération est un cycle de développement complet.
- **Incrémental** : chaque itération augmente la quantité d'information.

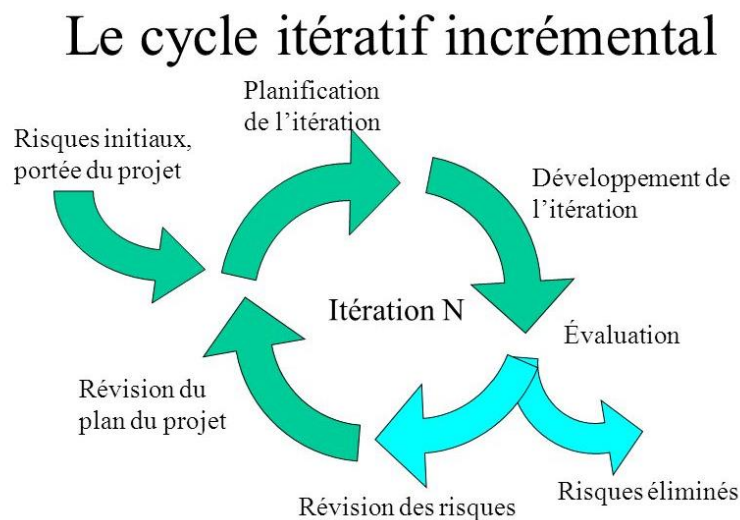


Figure 3 Présentation du cycle itératif et incrémental

Avec cette approche, le système croît avec le temps de façon incrémentale. Dans le développement incrémental, les tâches seront découpées en petits morceaux pour être développées au fil du temps et intégrées dès qu'elles sont terminées.

Dans ce cadre, les incréments les plus importants de notre projet sont :

- Le développement du système d'identification de véhicules ;
- Le développement de l'API de communication ;
- Le développement de l'application d'administration ;
- Le développement de l'espace Client ;
- La gestion des partenaires.

3.1.2. Présentation du langage de modélisation UML [15]

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.



Figure 4 Logo UML

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

Il permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation. Les créateurs d'UML

insistent tout particulièrement sur le fait qu'UML est un langage de modélisation et non une méthode.

L'UML utilise des éléments et les associe de différentes manières pour former des diagrammes qui représentent les aspects statiques ou structurels d'un système, ainsi que des diagrammes comportementaux qui capturent les aspects dynamiques d'un système.

3.1.2.1. Diagrammes UML structurels

UML comporte six (6) diagrammes statiques à savoir :

- **Diagramme des classes** : Diagramme UML le plus utilisé et fondement de toute solution orientée objet. Classes d'un système, attributs et opérations, et relations entre chaque classe. Les classes sont regroupées pour créer des diagrammes des classes lors de la modélisation de systèmes de grande taille.
- **Diagramme des composants** : Représente la relation structurelle entre les éléments d'un système logiciel, le plus souvent utilisé avec des systèmes complexes disposant de multiples composants. Les composants communiquent à l'aide d'interfaces.
- **Diagramme de structure composite** : Les diagrammes de structure composite sont utilisés pour présenter la structure interne d'une classe.
- **Diagramme de déploiement** : Illustre les infrastructures, physiques et logicielles, d'un système. Ils sont utiles lorsqu'une solution logicielle est déployée sur de nombreuses machines avec des configurations uniques.
- **Diagramme d'objets** : Montre les relations entre des objets à travers des exemples tirés du monde réel et permet de voir l'apparence d'un système à n'importe quel instant donné. Les données sont disponibles à l'intérieur des objets, elles peuvent donc être utilisées pour clarifier les relations entre des objets.
- **Diagramme des paquetages** : Il existe deux types de dépendances spéciales entre les paquetages : l'importation de paquetages et la fusion de paquetages. Les paquetages peuvent représenter les différents niveaux d'un système pour en révéler l'architecture. Les dépendances des paquetages peuvent être marquées pour mettre en avant le mécanisme de communication entre les différents niveaux.

3.1.2.2. Diagrammes UML comportementaux

UML comporte sept (7) diagrammes comportementaux ou dynamiques qui sont :

- **Diagrammes d'activités** : Flux de travail d'entreprise ou opérationnel représentés graphiquement pour montrer l'activité de chacune des composantes du système. Les diagrammes d'activités sont utilisés comme une alternative aux diagrammes états-transitions.
- **Diagramme de communication** : Semblable à un diagramme des séquences, mais l'accent est mis sur les messages transmis entre les objets. La même information peut être représentée par un diagramme des séquences et différents objets.
- **Diagramme global d'interaction** : Il existe sept types de diagrammes d'interaction, et ce diagramme montre l'ordre dans lequel ils agissent.
- **Diagramme des séquences** : Montre comment les objets interagissent les uns avec les autres et dans quel ordre. Il représente les interactions d'un scénario particulier.
- **Diagramme états-transitions** : Semblable à un diagramme d'activités, il décrit le comportement des objets qui se comportent de diverses manières dans leur état actuel.
- **Diagramme de temps** : À l'instar d'un diagramme des séquences, le comportement des objets est représenté sur un laps de temps donné. S'il y a un seul objet, le schéma est simple. S'il y a plus d'un objet, les interactions des objets sont présentées sur ce laps de temps précis.
- **Diagramme des cas d'utilisation** : Il représente une fonctionnalité spécifique dans un système et est créé pour illustrer comment différentes fonctionnalités sont interconnectées et montrer leurs contrôleurs (ou acteurs) internes et externes.

3.2. Présentation de l'outil de modélisation [16]

Il existe plusieurs outils de modélisation orientée objet qui supportent le langage UML mais nous avons choisi l'application desktop Draw.io pour sa gratuité et du fait qu'elle soit multi plateforme.

Draw.io est une application gratuite de création de diagrammes entièrement écrit en JavaScript à base du composant logiciel open source JGraph. Il existe en version en ligne, disponible depuis un navigateur web, et aussi en version hors ligne. Sa version en ligne permet de travailler directement avec G Suite/Google Drive et Dropbox pour partager ses diagrammes.

Avec la version hors ligne l'on peut créer et sauvegarder ses diagrammes en utilisant l'application desktop disponible pour MacOS, Windows et Linux.



Figure 5 Logo Draw.io

Draw.io fournit une interface intuitive avec la fonctionnalité de drag and drop. Avec Draw.io, l'utilisateur peut créer et éditer une variété de diagrammes comme :

- Des organigrammes ;
- Des diagrammes UML ;
- Des diagrammes ERD ;
- Des Business Process Models ;
- Des maquettings d'interfaces ;

Il constitue un outil puissant et flexible qui permettra de modéliser notre système.

3.3. Etude détaillée de la solution

Notre étude détaillée concernera la solution trouvée face à la problématique posée plus haut.

UML sera utilisé pour la modélisation du système. Les diagrammes UML présentés sont :

- Le diagramme de contexte statique ;
- Le diagramme de paquetages ;
- Les diagrammes de cas d'utilisation ;
- Le diagramme des classes ;
- Les diagrammes d'activités ;
- Les diagrammes des séquences ;
- Le diagramme de déploiement.

3.3.1. Le diagramme de contexte statique

Le diagramme de contexte statique délimite le domaine d'étude en précisant les acteurs en charge du système et l'environnement externe avec lequel ce dernier communique.

Le diagramme de contexte statique de notre projet est la figure qui suit.

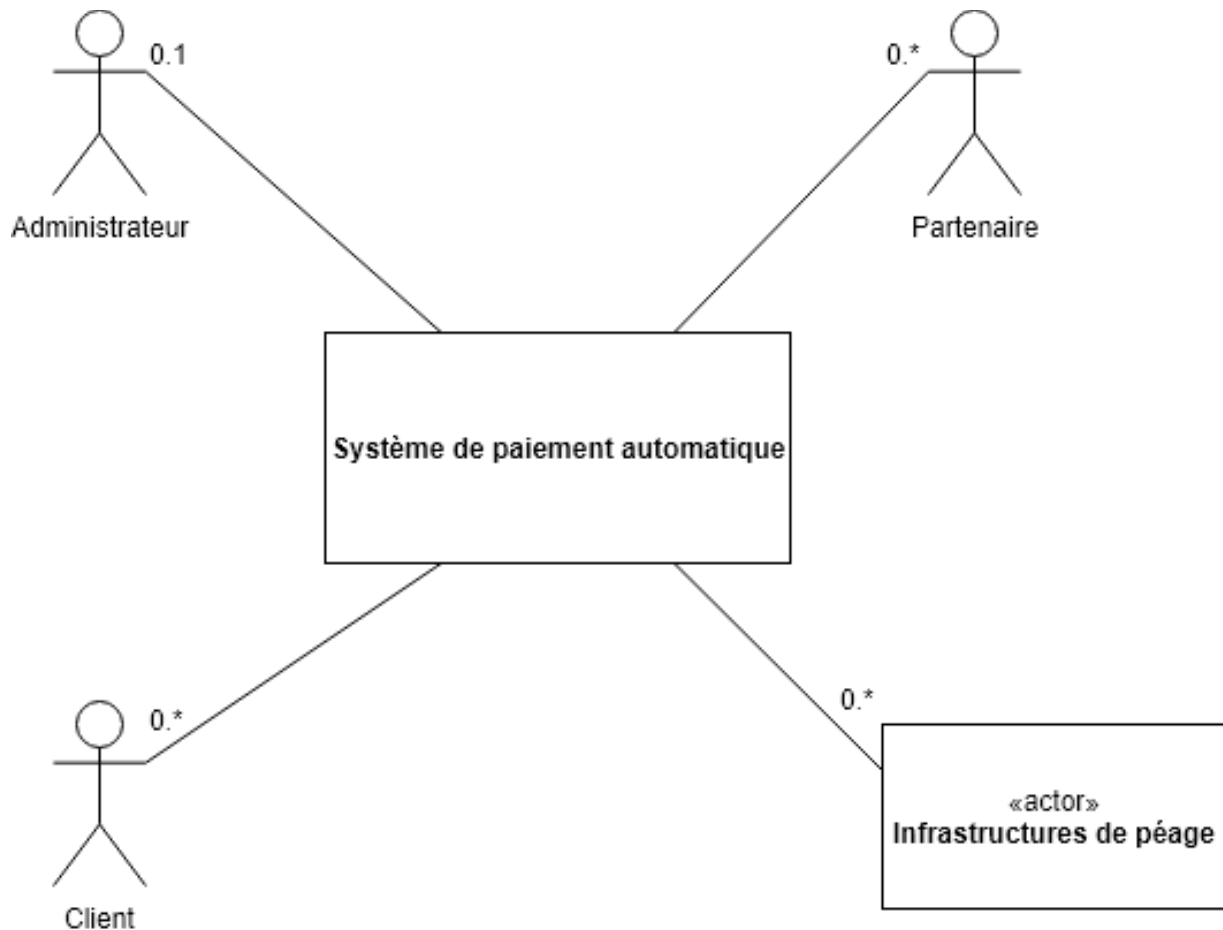


Figure 6 Diagramme de contexte statique

La description textuelle des acteurs qui interagissent avec le système est résumée dans le tableau suivant :

Tableau 7 Liste descriptive des acteurs du système

Acteurs	Descriptions
Principaux	
Le client	C'est l'utilisateur effectuant les opérations d'achat d'abonnements via le système. Ses informations sont stockées sur une carte RFID et lui permettent de passer au péage.
Le partenaire	Il est l'utilisateur qui offre des services aux clients. Il vend les abonnements via son espace.
L'administrateur	C'est l'utilisateur de l'application d'administration. Il a une vue sur le système par les statistiques sur les abonnements, les passages et les cartes. Il propose les abonnements aux partenaires.
Secondaires	
Les infrastructures de péage	Il s'agit d'une barrière et un moteur connectés à un lecteur RFID aux stations de péage. Ces infrastructures interagissent avec notre système à la détection d'une carte RFID par le lecteur.

3.3.2. Le diagramme de déploiement

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

Notre solution fonctionne selon le diagramme de déploiement suivant.

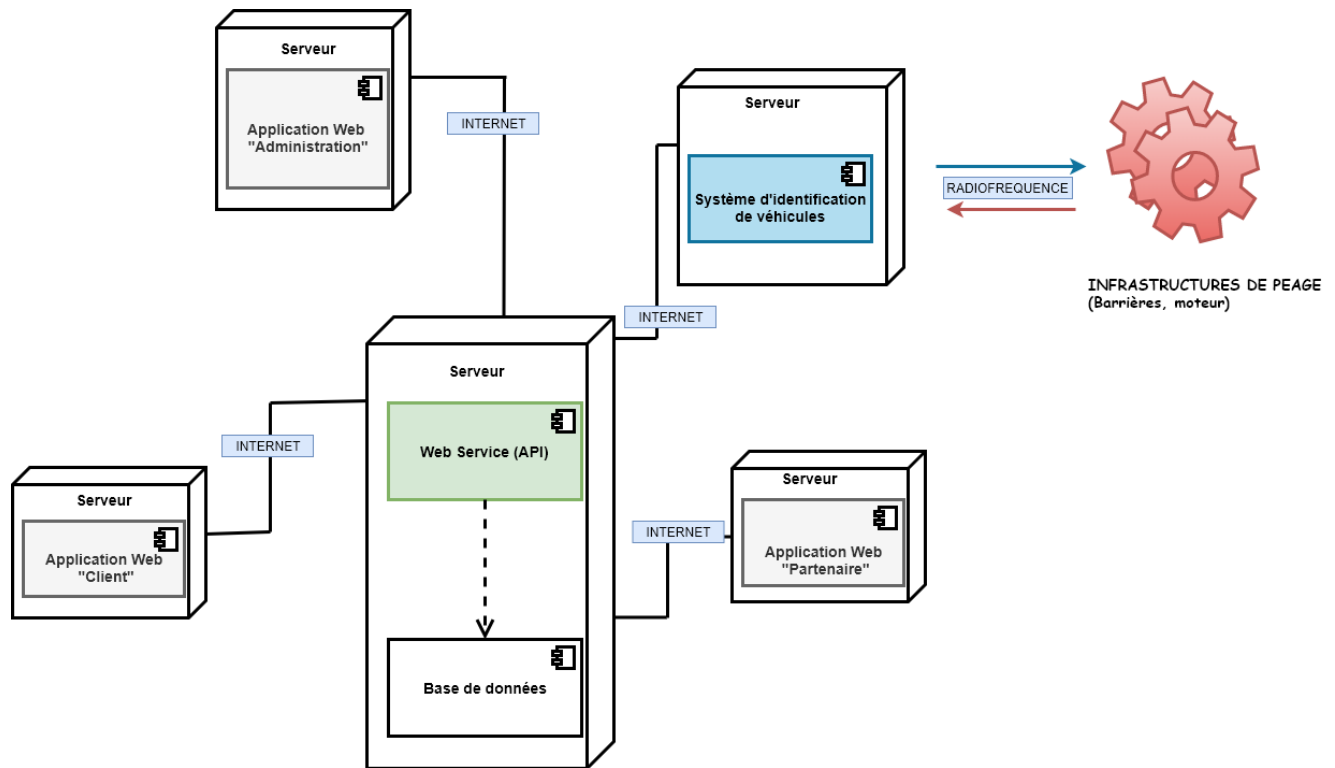


Figure 7 Diagramme de déploiement

Le diagramme de déploiement présenté contient trois (3) serveurs d'application et un serveur de base de données. Il présente également l'interaction entre ces serveurs. Le diagramme modélise l'architecture physique du système en affichant les relations entre les composants logiciels et matériels du système. La communication entre les serveurs se fera par internet.

3.3.3. Le diagramme des composants

Le diagramme des composants représente la relation structurelle entre les éléments d'un système logiciel, le plus souvent utilisé avec des systèmes complexes disposant de multiples composants. Les composants communiquent à l'aide d'interfaces.

La figure suivante illustre le diagramme des composants de notre projet.

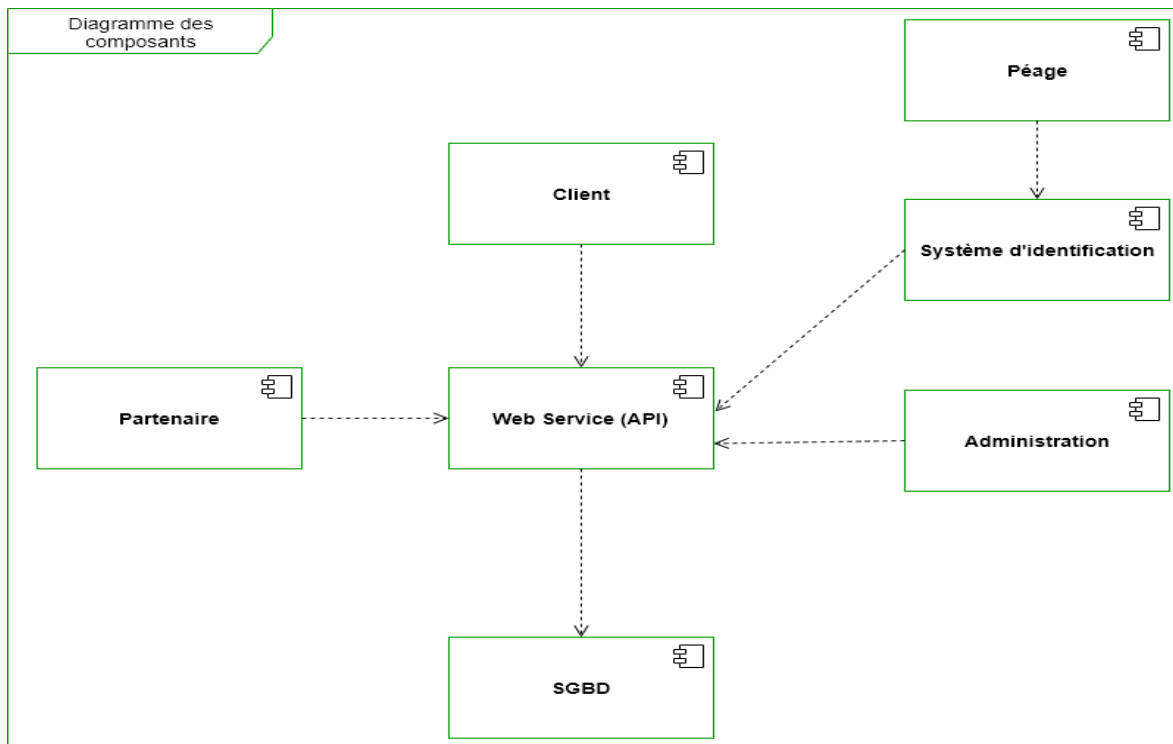


Figure 8 Diagramme des composants

3.3.4. Le diagramme des paquetages

L'ensemble du système à concevoir est découpé en différentes parties suivant des familles de fonctionnalités afin d'avoir une vision plus claire du travail à réaliser. Ces parties sont appelées paquets (package en anglais). Le découpage du système présente les paquets suivants :

- **Administration** : il prend en charge la gestion (création, modification, suppression et statistiques) des abonnements, des cartes et des partenaires ;
- **Partenaire** : il englobe les services offerts aux clients (ajout d'un client, vente d'abonnements) et la gestion processus des partenaires (consultation d'abonnements disponibles et statistiques) ;
- **Client** : ce package se rapporte au paiement en ligne des abonnements et à la consultation des abonnements du client ;
- **Péage** : il contient les services externes pour le fonctionnement de notre système tels que le lecteur de cartes, les barrières et le moteur aux stations de péage.

La figure suivante présente le diagramme de paquetages issu de l'analyse du projet.

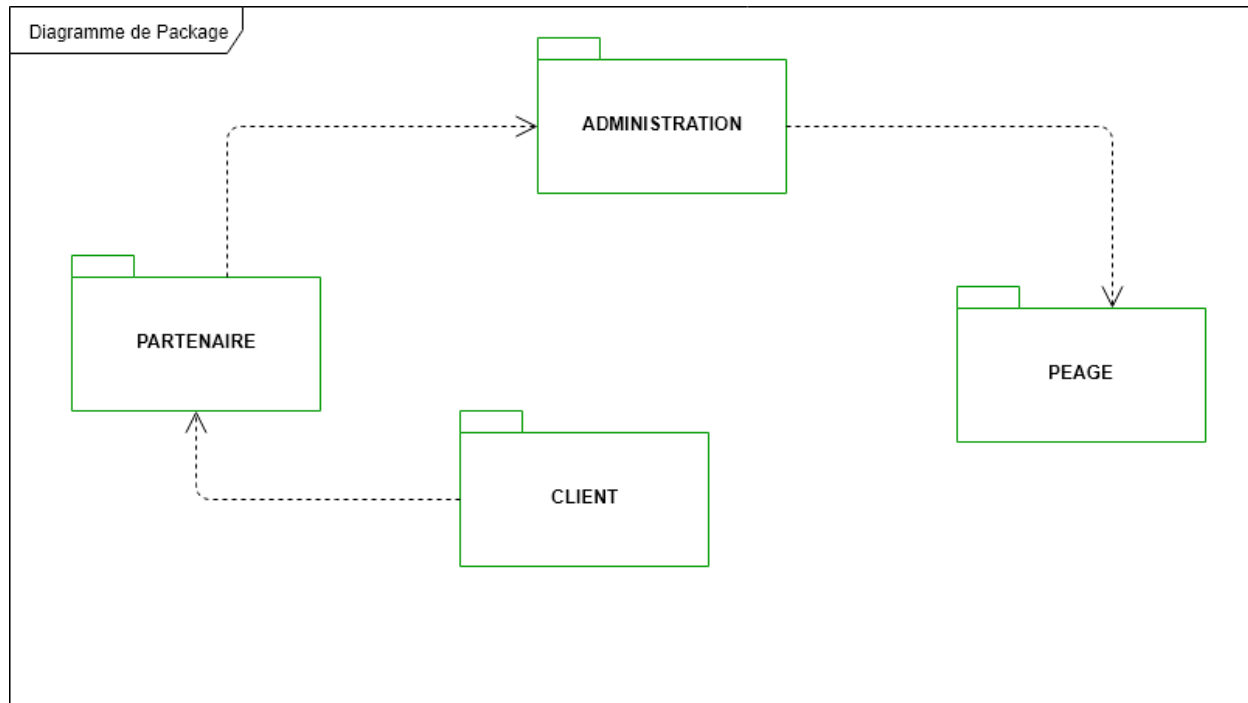


Figure 9 Diagramme de paquetages

3.3.5. Les diagrammes de cas d'utilisation

Différents diagrammes de cas d'utilisation ont été illustrés à savoir :

- Le diagramme des cas d'utilisation du paquet « Administration » ;
- Le diagramme des cas d'utilisation du paquet « Partenaire » ;
- Le diagramme des cas d'utilisation du paquet « Client ».

Ces diagrammes seront présentés avec un résumé de chaque cas d'utilisation.

3.3.5.1. Cas d'utilisation du paquet « Administration »

La figure suivante illustre le diagramme des cas d'utilisation du paquet « Administration ». Elle montre l'interaction de l'acteur « Administrateur » sur le système.

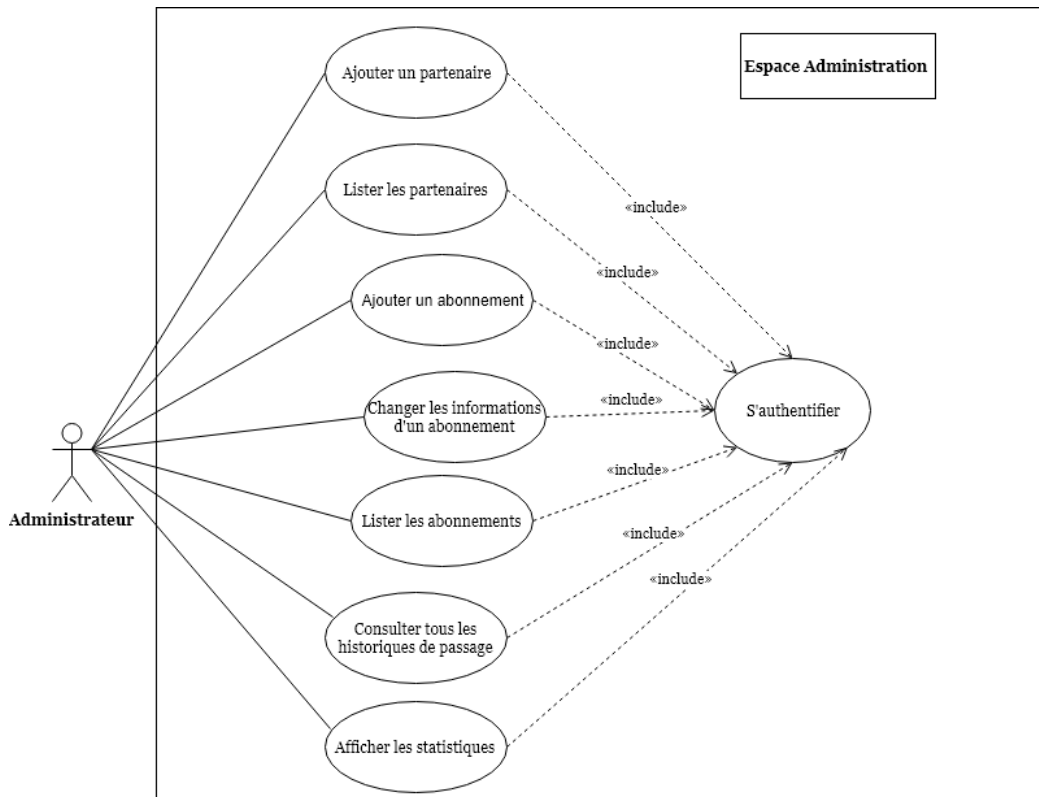


Figure 10 D0 d'utilisation du paquet « Administration »

Résumé des cas d'utilisation

- **Ajouter un partenaire :** Ce cas permet d'ajouter des partenaires désireux d'offrir des abonnements de péage ;
- **Lister les partenaires :** L'administrateur peut afficher la liste des partenaires ;
- **Ajouter un abonnement :** L'administrateur a la possibilité d'ajouter différents abonnements utilisables par les partenaires ;
- **Lister les abonnements :** Afficher la liste des abonnements proposés ;
- **Changer les informations d'un abonnement :** Ce cas permet de modifier les informations d'un abonnement ;
- **Consulter tous les historiques de passage :** Ce cas donne la possibilité à l'administrateur d'avoir une vue sur les historiques de tous les passages ;
- **Afficher les statistiques :** Dans ce cas, l'administrateur pourra avoir un aperçu des statistiques générales pour tous les partenaires.

3.3.5.2. Cas d'utilisation du paquet « Partenaire »

L'illustration du diagramme des cas d'utilisation du paquet « Partenaire » est la figure suivante.

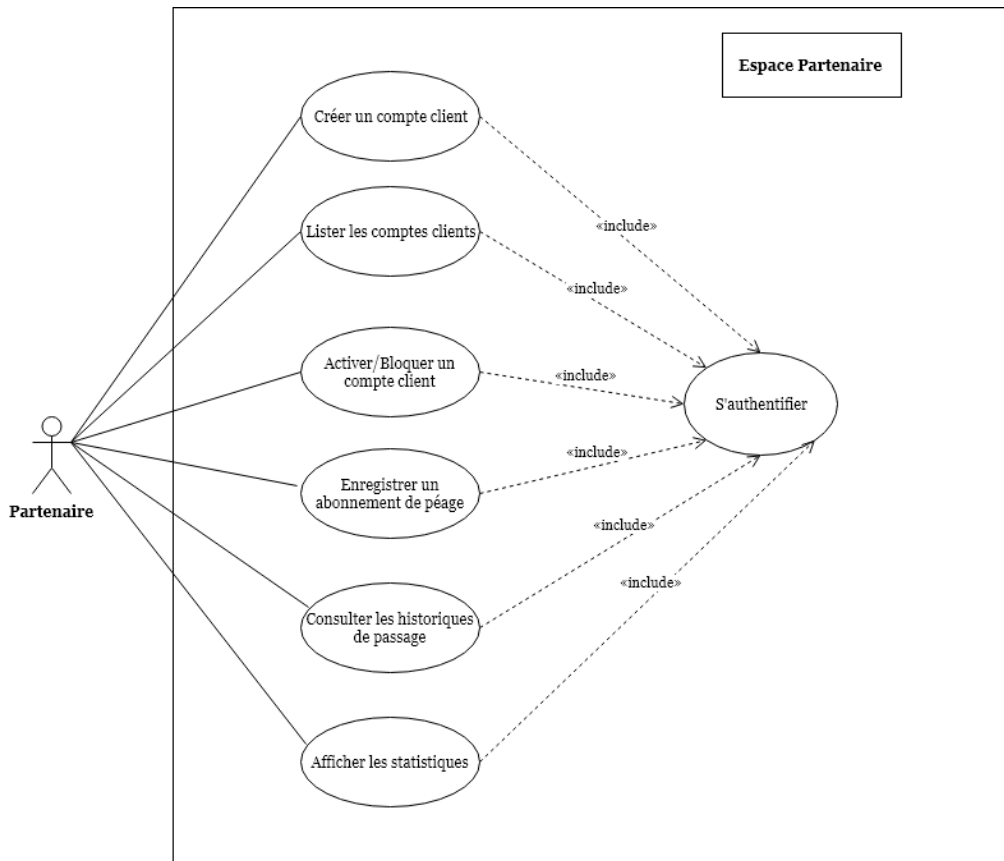


Figure 11 Diagramme des cas d'utilisation du paquet « Partenaire »

Résumé des cas d'utilisation

- **Créer un compte client** : Ce cas permet au partenaire d'ouvrir un compte à un client à l'achat d'un badge ;
- **Lister les comptes clients** : Le partenaire a la possibilité de voir tous les comptes clients ainsi que leurs informations ;
- **Activer/Bloquer compte client** : Le partenaire dans ce cas, a le monopole de la gestion du compte utilisateur et peut bloquer son compte en cas de non-respect des règles du péage ;
- **Enregistrer un abonnement de péage** : Ce cas permet de vendre un abonnement à un client ;
- **Consulter les historiques de passage** : Dans ce cas, le partenaire peut lister les historiques de passage d'un client donné ;

- **Afficher les statistiques :** Le partenaire peut voir les statistiques des clients, des abonnements et les sommes correspondantes.

3.3.5.3. Cas d'utilisation du paquet « Client »

La figure suivante présente le diagramme des cas d'utilisation du paquet « Client ».

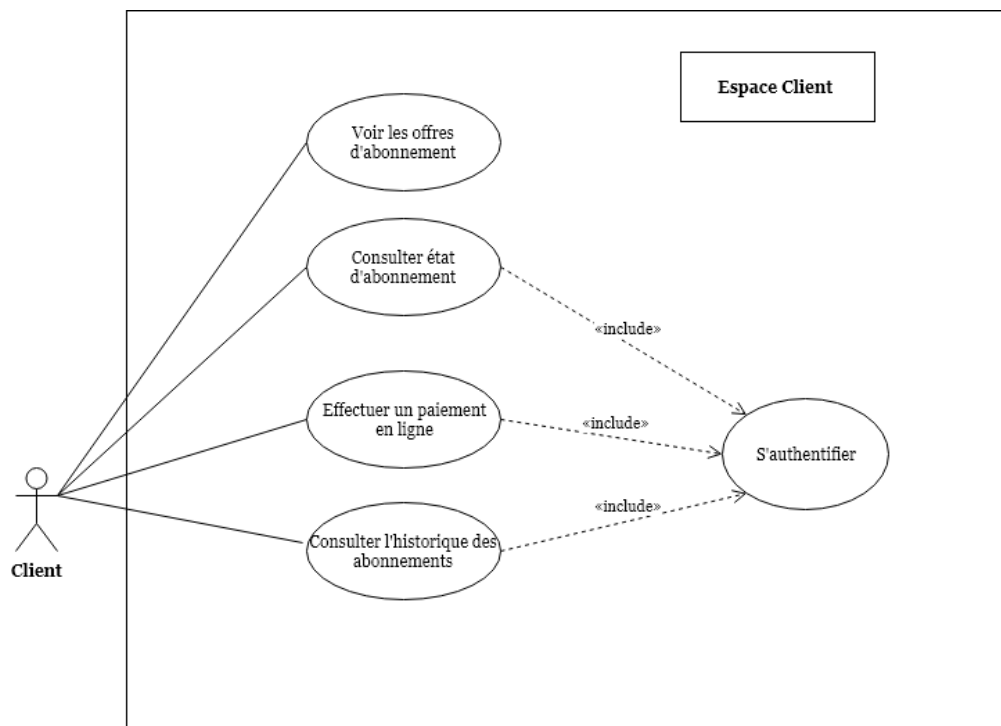


Figure 12 Diagramme des cas d'utilisation du paquet « Client »

Résumé des cas d'utilisation

- **Voir les offres d'abonnements :** Ce cas permet aux visiteurs d'avoir la liste des abonnements disponibles ;
- **Consulter état d'abonnement :** Le client peut avoir une vue sur l'évolution de son abonnement ;
- **Effectuer un paiement en ligne :** Ce cas donne la possibilité au client d'effectuer un paiement par un moyen de paiement en ligne ;
- **Consulter l'historique des abonnements :** Dans ce cas, le client pourra avoir un aperçu de l'histoire de ses abonnements.

3.3.5.4. Description textuelle de quelques cas d'utilisation

Ci-dessous quelques tableaux faisant une description textuelle de certains cas d'utilisation. Les cas considérés sont :

- S'authentifier ;
- Ajouter un abonnement ;
- Consulter l'état d'un abonnement.

Le cas d'utilisation « S'authentifier » est décrit dans le tableau suivant.

Tableau 8 Description textuelle du cas « S'authentifier »

Titre	S'authentifier
Résumé	Permet à l'utilisateur d'entrer ses informations de connexion dans le système pour avoir accès à ses ressources.
Acteurs concernés	Client (conducteur ou propriétaire d'un véhicule), Administrateur
Préconditions	Avoir été ajouté à la base de données des utilisateurs
Scénario nominal	<ol style="list-style-type: none"> 1. Le client ou l'administrateur charge l'application. 2. Le système lui affiche un formulaire de connexion. 3. L'opérateur renseigne ses informations de connexion. 4. Le système vérifie les informations de l'utilisateur 5. Le système lui donne accès à l'application
Extensions	<p>(a) : Informations de l'utilisateur incorrectes.</p> <p>L'enchaînement (a) démarre au point 4 du scénario nominal.</p> <ol style="list-style-type: none"> 7. Le système indique à l'utilisateur que ses informations sont invalides. 8. Le scénario nominal reprend au point 2.

Le tableau suivant présente la description du cas d'utilisation « Enregistrer un abonnement ». Il explique le processus de vente d'abonnement à un client par un partenaire.

Tableau 9 Description textuelle du cas « Enregistrer un abonnement »

Titre	Ajouter un abonnement
Résumé	Ce cas permet au partenaire de pouvoir attribuer un abonnement à un client
Acteurs concernés	Partenaire
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur charge l'application. 2. Le système lui affiche un formulaire de connexion. 3. Le partenaire renseigne ses informations de connexion. 4. Le système vérifie les informations de l'utilisateur. 5. L'utilisateur entre dans la partie de gestion des abonnements 6. L'utilisateur cherche le client à partir de son identifiant 7. L'utilisateur clique sur le bouton pour accéder au détail de l'étudiant 8. Il clique sur ajouter un abonnement 9. L'administrateur choisit l'abonnement en fonction des besoins du client
Extensions	<p>(a) : Informations de l'utilisateur incorrectes.</p> <p>L'enchaînement (a) démarre au point 4 du scénario nominal.</p> <p>8. Le système indique à l'administrateur que ses informations sont invalides.</p> <p>9. Le scénario nominal reprend au point 2.</p>

La description du cas d'utilisation « Consulter l'état d'un abonnement » est présentée dans le tableau suivant. Ce cas permet à un client d'avoir une vue sur l'évolution de son abonnement.

Tableau 10 Description textuelle du cas « Consulter l'état d'un abonnement »

Titre	Consulter l'état d'un abonnement
Résumé	Ce cas permet à un client d'avoir une vue sur l'évolution de son abonnement
Acteurs concernés	Client
Précondition	Se connecter et avoir fait un abonnement
Scénario nominal	<ol style="list-style-type: none"> 1. Le client accède à l'espace client. 2. Le système lui affiche un formulaire de connexion. 3. Le client renseigne ses informations de connexion. 4. Le système vérifie les informations du client. 5. Le client peut se diriger vers l'espace de consultation d'évolution de traitement. 6. Le système lui affiche l'évolution de son abonnement
Extensions	<p>(a) : Informations de l'utilisateur incorrectes.</p> <p>L'enchaînement (a) démarre au point 4 du scénario nominal.</p> <ol style="list-style-type: none"> 7. Le système indique à l'étudiant que ses informations sont invalides. 8. Le scénario nominal reprend au point 2.

3.3.6. Le diagramme des classes

Le diagramme des classes conçu montre la structure interne du système. Il présente les classes et les interfaces du système ainsi que les différentes relations entre celles-ci. Ce diagramme permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Le diagramme des classes utilisé pour la réalisation de la base de données est la figure suivante.

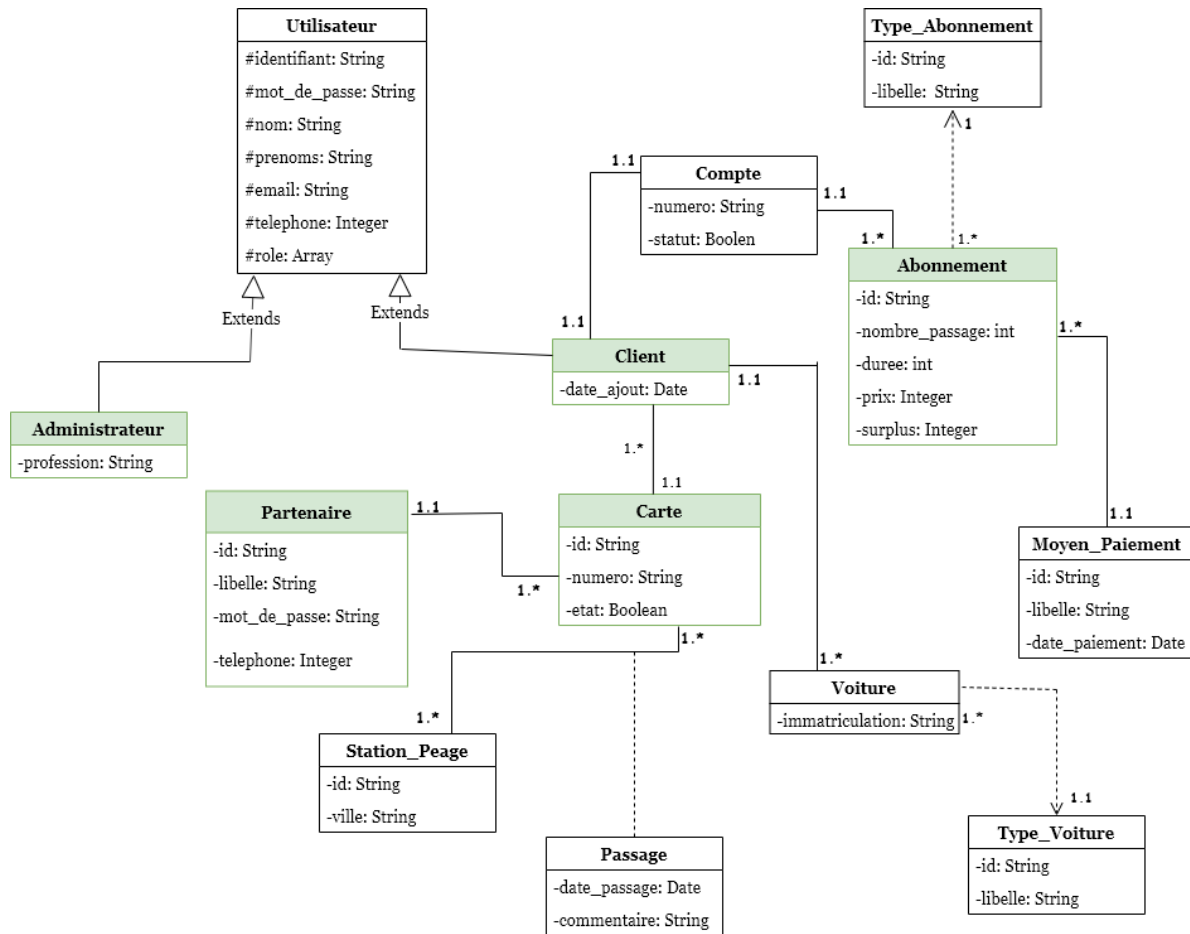


Figure 13 Diagramme des classes

3.3.7. Les diagrammes d'activités

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils représentent graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Les diagrammes d'activités conçus sont :

- Le diagramme d'activités « Gestion des abonnements » ;
- Le diagramme d'activités « Interaction Client – Partenaire » ;
- Le diagramme d'activités « S'authentifier ».

• Diagramme d'activités « Gestion des abonnements »

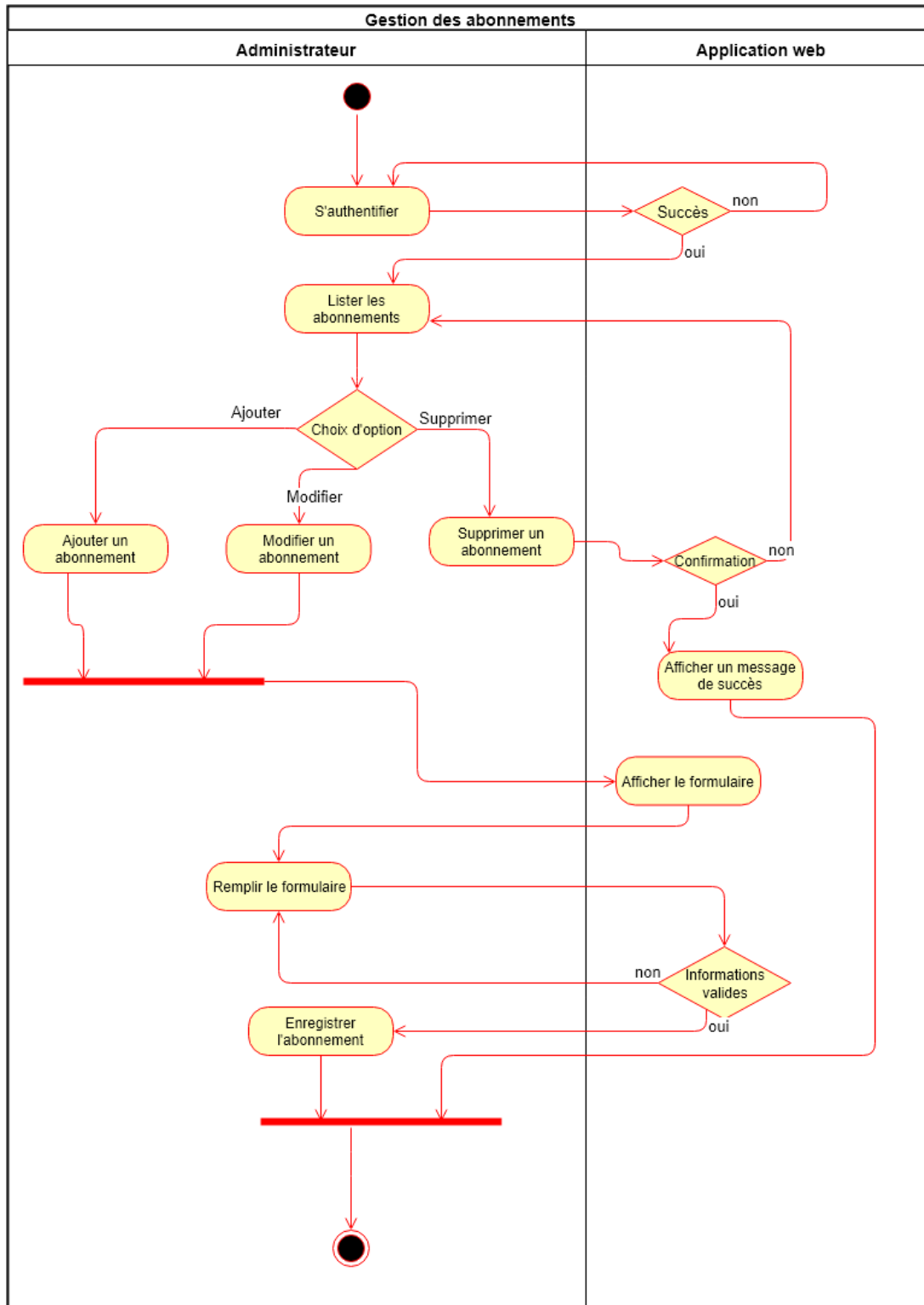


Figure 14 Diagramme d'activités « Gestion des abonnements »

• Diagramme d'activités « Interaction Client-Partenaire »

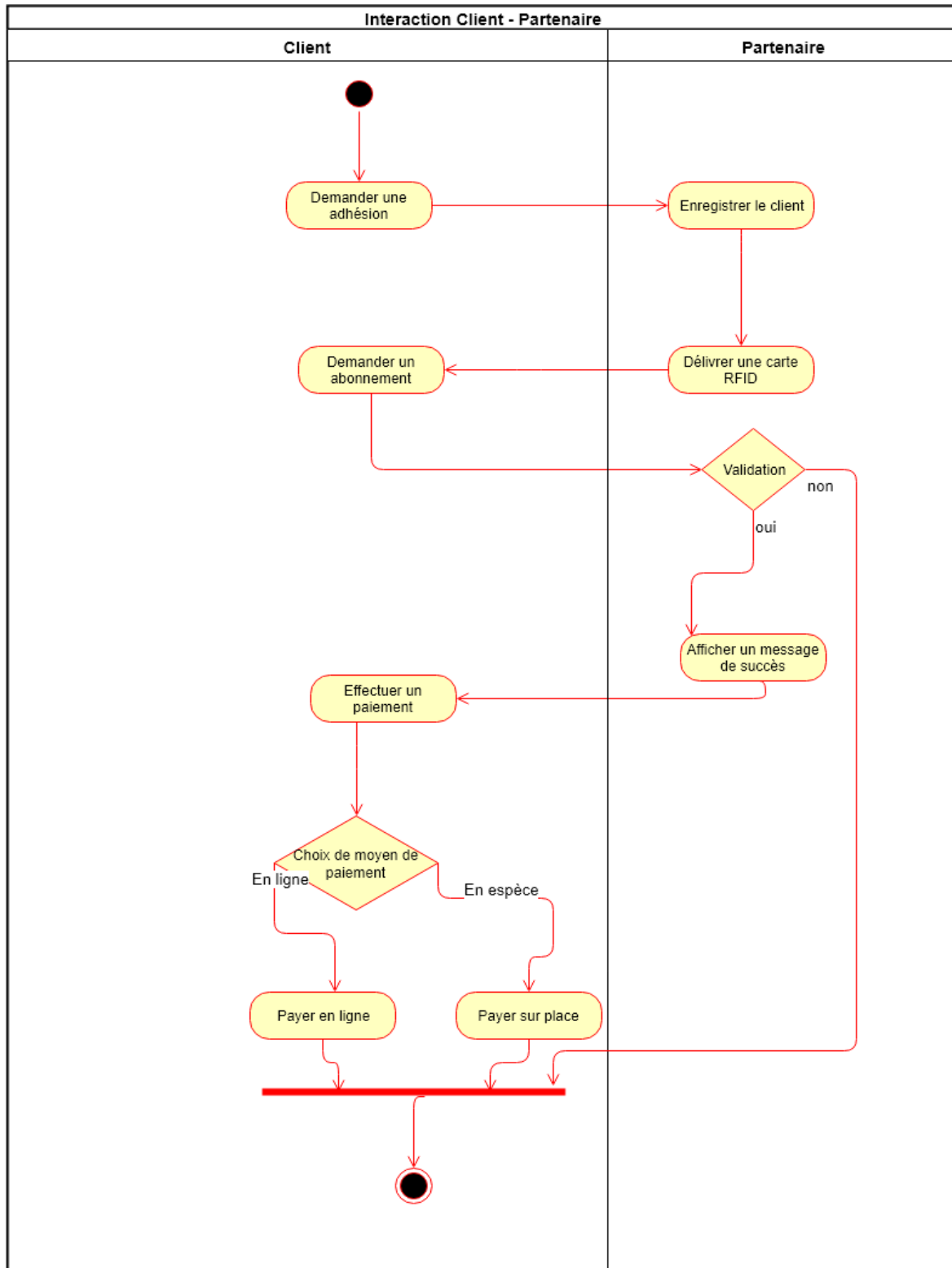


Figure 15 Diagramme d'activités « Interaction Client – Partenaire »

- Diagramme d'activités « S'authentifier »

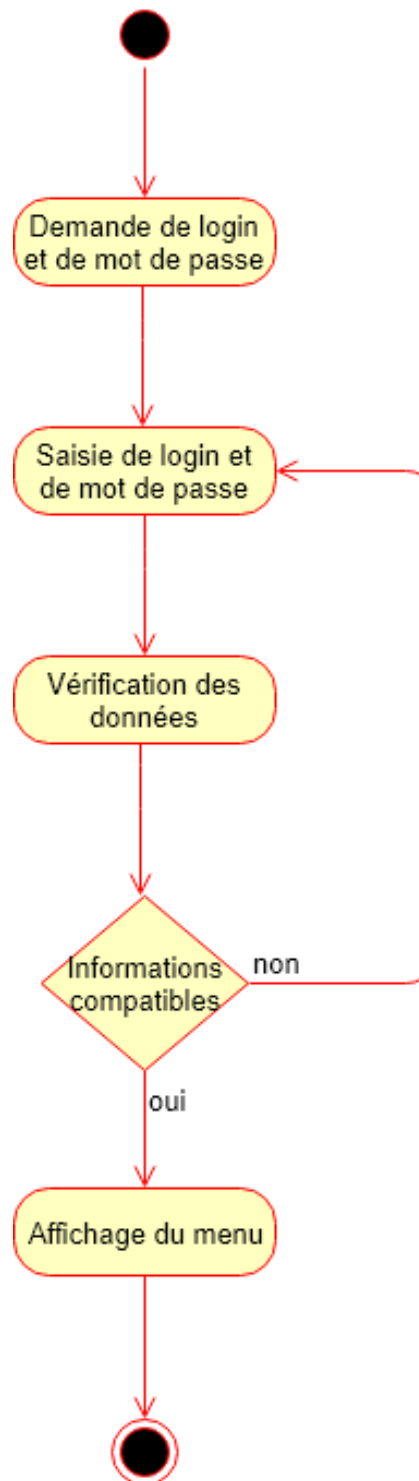


Figure 16 Diagramme d'activités « S'authentifier »

3.3.8. Les diagrammes des séquences

Les diagrammes des séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs.

Les diagrammes des séquences présentés sont :

- Le diagramme des séquences « Consulter l'état d'un abonnement » ;
- Le diagramme des séquences « Afficher les statistiques ».

Ces diagrammes sont illustrés sur les figures qui suivent.

- **Diagramme des séquences « Consulter l'état d'un abonnement »**

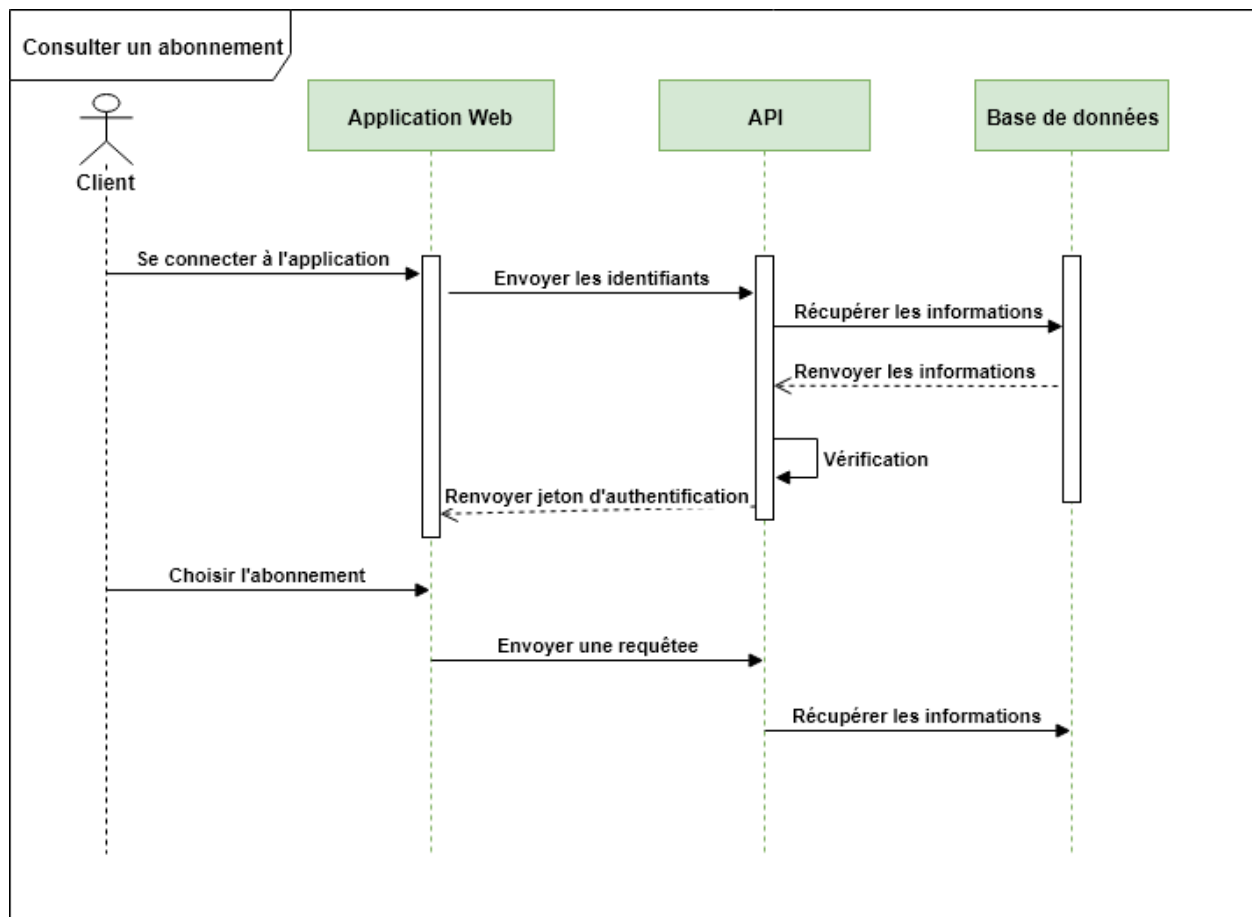


Figure 17 Diagramme des séquences représentant la consultation d'un abonnement

- Diagramme des séquences « Afficher les statistiques »

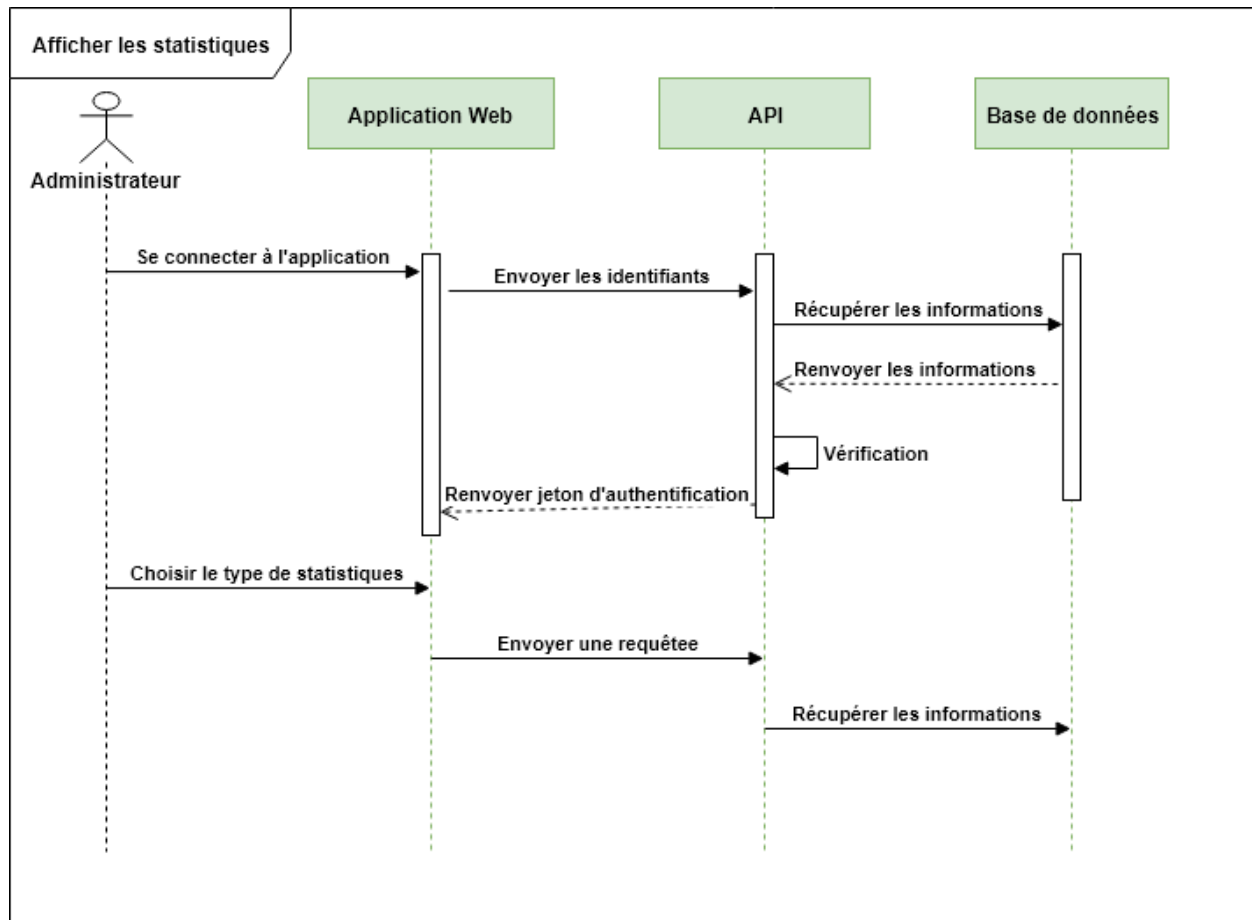


Figure 18 Diagramme des séquences de l'affichage des statistiques

L'analyse et la conception concernant notre projet a permis d'avoir une vision plus claire du système implémenté. Plusieurs moyens et outils ont été utilisés pour la mise en œuvre du projet. La présentation des résultats obtenus et les moyens de mise en œuvre feront l'objet du chapitre suivant.

CHAPITRE 4 RÉALISATION ET MISE EN ŒUVRE

Il sera présenté dans cette section, les outils et moyens de mise en œuvre de notre solution et ensuite la présentation des résultats obtenus.

4.1. Mise en œuvre

Dans cette partie, les grandes lignes de la mise en œuvre de notre solution seront abordées, en commençant par le matériel et les logiciels utilisés dans la réalisation de l'application. Il sera ensuite présenté la description de l'architecture logicielle de l'application ainsi que l'architecture matérielle sur laquelle elle a été déployée. Les détails du déploiement et le système de sécurité de l'application aussi seront exposés.

Le tableau suivant résume les systèmes développés et leurs modules.

Tableau 11 Récapitulatif des systèmes et leurs modules

Systèmes	Modules
Application d'administration	<ul style="list-style-type: none"> • Gestion des partenaires; • Gestion des types d'abonnements; • Gestion des cartes et des passages ; • Attribution de cartes et vente d'abonnements à un partenaire ; • Statistiques des processus du système.
Application des partenaires	<ul style="list-style-type: none"> • Gestion des clients, des cartes et des ventes ; • Statistiques des opérations effectuées.
Application web Client	<ul style="list-style-type: none"> • Consultation d'offres et d'état des abonnements ; • Achat d'abonnements en ligne.
Système d'identification de véhicules	<ul style="list-style-type: none"> • Identification de cartes; • Lecture des informations du client et envoi de signal aux infrastructures.

4.1.1. Matériel et logiciels utilisés

Plusieurs langages de programmation, technologies et Framework ont aidé dans l'exécution du projet. Nous présenterons :

- L'environnement de programmation ;
- Les langages de programmation ;
- Les technologies et Framework utilisés ;
- Les serveurs logiciels utilisés ;
- Le matériel de développement.

4.1.1.1. Environnement de programmation

L'éditeur de code utilisé pour la réalisation du projet est Visual Studio Code.

- **Visual Studio Code**

Visual Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et MacOS.



Il comprend la prise en charge du débogage, du contrôle de la coloration syntaxique et de la

Figure 19 Logo Visual Studio Code

complétion de code intelligent. Il est hautement personnalisable, permettant aux utilisateurs de changer le thème, les raccourcis clavier, les préférences et d'installer des extensions qui ajoutent des fonctionnalités supplémentaires. [17]

4.1.1.2. Langages de programmation

La programmation de l'API de communication a été faite en PHP dans sa version 7.1 couplée à certaines technologies. Ces langages ont été choisis car tout d'abord PHP est l'un des langages côté serveur le plus utilisé pour le développement web. En utilisant ce dernier l'intégration de modules

externes ou existants à notre solution sera plus facile. Enfin HTML5, CSS3 et JavaScript sont des langages fondamentaux à toute application web.

Le système d'identification de véhicules a été développé en Golang. Golang a été utilisé à cause de sa bonne gestion des threads et sa capacité à pouvoir utiliser des threads légers afin de gérer les messages provenant de plusieurs plateformes dans une architecture client-serveur. Golang offre une facilité de communication avec le matériel ; avantage qui s'intègre dans le cadre de notre projet qui contient des infrastructures de péage.

Ci-dessous, la présentation des langages énoncés (PHP, Golang, HTML5, CSS3, et JavaScript).

- **Le langage de programmation PHP**



Figure 20 Logo PHP

PHP (Hypertext Preprocessor), plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

- **Golang**

Golang, encore appelé Go est un langage de programmation compilée développé par Google.



Figure 21 Logo Golang

Go facilite et accélère la programmation à grande échelle : en raison de sa simplicité, il est donc concevable de l'utiliser aussi bien pour écrire des applications, des scripts ou de grands systèmes.

- **Le langage JavaScript**

Pour le dynamisme de certaines interfaces nous avons opté pour du JavaScript.



Figure 22 Logo JavaScript

JavaScript (qui est souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

- **Le langage de présentation HTML5**

HTML signifie « *HyperText Markup Language* » qu'on peut traduire par « langage de balises pour l'hypertexte ».



Figure 23 Logo HTML

Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure.

- **Le langage CSS3 [18]**

Le langage CSS a servi pour mettre en forme le design de notre application.



Figure 24 Logo CSS

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML (on inclut ici les langages basés sur XML comme SVG ou XHTML).

4.1.1.3. Technologies et Framework utilisés

La programmation des vues et des interfaces graphiques a été faite à l'aide d'un Framework.

Un Framework est, comme son nom l'indique en anglais, un "cadre de travail". L'objectif d'un Framework est généralement de simplifier le travail des développeurs, en offrant une architecture "prête à l'emploi" et qui permet de ne pas repartir de zéro à chaque nouveau projet.

Deux Framework ont été utilisés pour la garantie qu'ils offrent en termes de sécurité et leur capacité à accélérer le développement. Ces Framework sont :

- Vue JS de JavaScript ;
- Laravel de PHP.

- **Vue JS**

Vue.js (aussi appelé plus simplement Vue), est un Framework JavaScript Open source utilisé pour construire des interfaces d'utilisateur.

Vue simplifie et organise le développement d'applications web en implémentant un bon nombre de fonctionnalités telles que : Templates (modèles), composants, transitions et routage. [19]



Figure 25 Logo Vue JS

- **Laravel**

Laravel est un Framework web Open Source écrit en PHP, respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet.

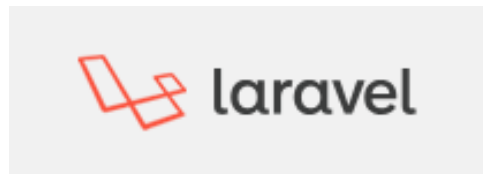


Figure 26 Logo Laravel

Laravel offre:

- Un système d'envoi de mail ;
- Un système de routage performant ;
- Un constructeur de formulaire très pratique ;
- Un système d'authentification pour les connexions ;
- Un créateur de requêtes SQL et un ORM performant (éloquent). [20]

En prime, Laravel intègre un système de migration pour les bases de données, un système de cache, une gestion des sessions.

Le cadre structurant imposé par Laravel est un style de développement selon le motif d'architecture MVC.

- **Architecture Modèle-Vue-Contrôleur (MVC) [21]**

Le pattern MVC permet de bien organiser son code source. Il permet de savoir quels fichiers créer, mais surtout à définir leur rôle. Lorsque l'utilisateur sollicite l'interface graphique, les actions suivantes sont exécutées :

- La requête est analysée par le contrôleur ;
- Le contrôleur demande au modèle d'effectuer les traitements ;
- Le contrôleur renvoie la vue adaptée si le modèle ne l'a pas déjà fait.

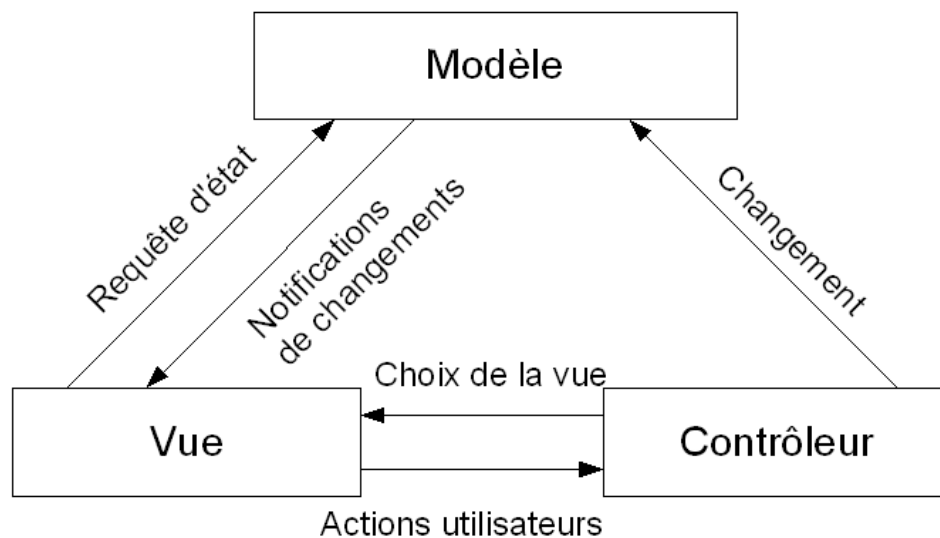


Figure 27 Architecture MVC

Pour faciliter l'élaboration du design de notre application nous nous sommes servis du Framework CSS3 Bootstrap dans sa version 4.

- **Le Framework Bootstrap 4 [23]**



Figure 28 Logo Bootstrap

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur) de sites et d'applications web développée par twitter. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

- **Ajax [25]**



Figure 29 Logo Ajax

AJAX (Asynchronous JavaScript + XML) n'est pas une technologie en soi, mais un terme désignant une « nouvelle » approche utilisant un ensemble de technologies existantes, dont : HTML ou XHTML, les feuilles de styles CSS, JavaScript, le modèle objet de document (DOM), XML, XSLT, et l'objet XMLHttpRequest. Lorsque ces technologies sont combinées dans le modèle AJAX, les applications Web sont capables de réaliser des mises à jour rapides et incrémentielles de l'interface utilisateur sans devoir recharger la page entière du navigateur. Les applications fonctionnent plus rapidement et sont plus réactives aux actions de l'utilisateur.

Afin d'identifier au mieux les véhicules au péage et de commander les infrastructures de péage, il a été utilisé en plus de Golang, une technologie de radio-identification.

- **La technologie de radio-identification RFID [26]**

La radio-identification, le plus souvent désignée par le sigle RFID (de l'anglais Radio Frequency Identification), est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes » (« RFID tag » ou « RFID transponder » en anglais).

Les radio-étiquettes sont de petits objets, tels que des étiquettes autoadhésives, qui peuvent être collés ou incorporés dans des objets ou produits. Les radio-étiquettes comprennent une antenne associée à une puce électronique qui leur permet de recevoir et de répondre aux requêtes radio

émises depuis l'émetteur-récepteur. Elles contiennent un identifiant et éventuellement des données complémentaires.

Dans le cadre de notre projet, il a été utilisé **une carte RFID** sans contact comme radio-étiquette.



Figure 30 Carte RFID sans contact

Cette carte, collée sur les véhicules, permettra d'enregistrer les informations liées au propriétaire d'un véhicule.

Le système d'identification de véhicules (par carte RFID) communiquera avec les infrastructures de péage. Pour mieux présenter notre système, un prototype d'infrastructures de péage a été conçu : une barrière et un moteur qui communiquent avec le système d'identification. La barrière se lèvera à l'identification d'une carte éligible (un client ayant un abonnement actif).

4.1.1.4. Les serveurs logiciels utilisés

Pour le serveur web d'interprétation des requêtes http nous avons utilisé le serveur web Apache dans sa version 2.

- **Serveur Apache**

Le logiciel libre Apache HTTP Server (Apache) est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.



Figure 31 Logo Apache

Pour le stockage de nos données et la gestion de la base de données nous avons utilisé le Système de Gestion de Base de Données Relationnelle (SGBDR) MySQL.

- **Le SGBDR MySQL**

MySQL est un SGBDR distribué sous une double licence GPL et propriétaire. Il a été choisi du fait de sa simplicité et de sa compatibilité avec les autres outils utilisés. [27]



Figure 32 Logo MySQL

4.1.1.5. Le matériel de développement

Un laptop Acer Aspire ES 17 et un Raspberry Pi 3 ont été utilisés pour le développement du système, avec les caractéristiques suivantes :

Tableau 12 Matériel de développement

Nom	Acer Aspire ES 17	Raspberry Pi 3
Processeur	Intel Celeron Processor N3350	Quad-core ARM Cortex-A53
Fréquence du processeur	1,1 GHz	1,2 GHz
Quantité de mémoire vive	4GB	1GB
Type de mémoire vive	DDR3	

La figure qui suit présente le Raspberry Pi 3 utilisé.

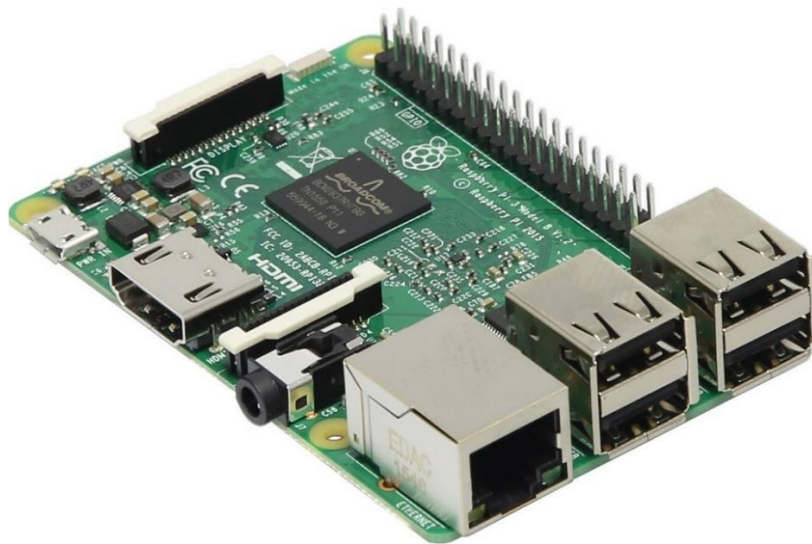


Figure 33 Raspberry Pi 3

Le Raspberry a servi de module matériel pour assurer la communication entre les applications, le système d'identification de véhicules, le moteur et les barrières des stations de péage.

4.1.2. Architectures et sécurité du système

Il est présenté dans cette partie :

- Les architectures logicielles et matérielles ;
- La sécurité du système.

4.1.2.1. Architectures logicielles et matérielles

Le système mis en place est basé sur une architecture trois (3) tiers. L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches, est l'application du modèle plus général qu'est le multi-tiers. Elle est basée sur l'environnement client-serveur. L'architecture logique du système est divisée en trois niveaux ou couches à savoir :

- Couche de présentation ;
- Couche de traitement ;
- Couche d'accès aux données.

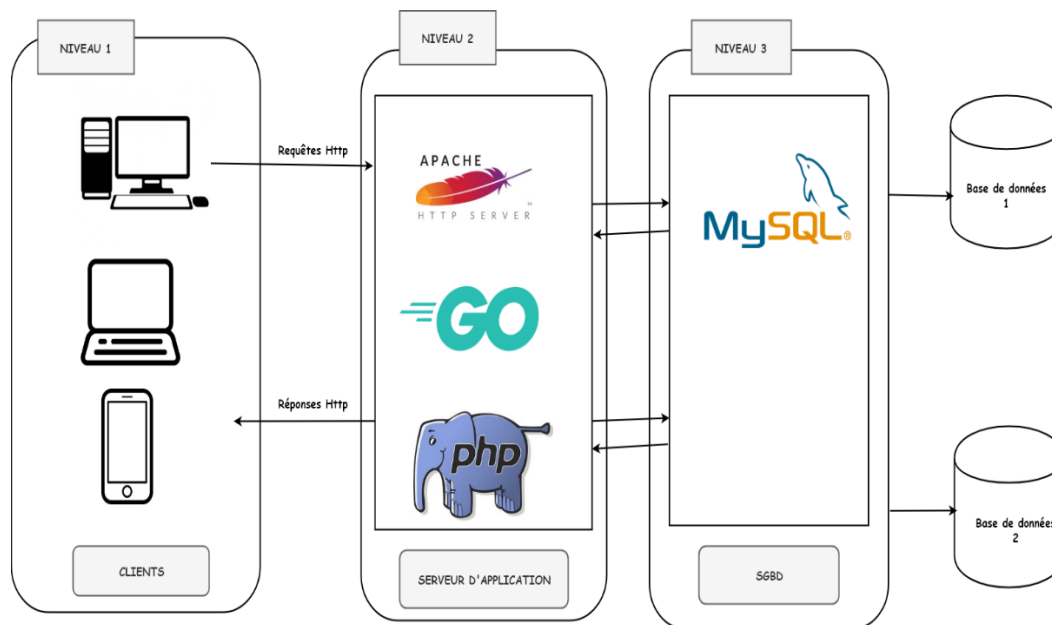


Figure 34 Architecture 3 tiers

• La couche de présentation

C'est la première couche qui compose l'infrastructure trois tiers : il s'agit de la partie rendu logiciel. Son rôle est d'afficher les données et de permettre à l'utilisateur final d'interagir avec ces dernières.

- **La couche métier**

Elle est en charge d'appliquer et de respecter les règles métiers (ou actes de gestion). C'est dans celle-ci que sont implémentées la logique applicative et la sécurité dans ce modèle d'architecture.

- **La couche d'accès aux données**

C'est la troisième couche qui compose l'infrastructure trois-tiers : elle correspond au serveur de base de données. Il s'agit de la couche d'accès aux données. Sur ce troisième tiers, un SGBD (Système de Gestion de Base de Données) est installé et ce serveur est requêté par le serveur applicatif afin d'utiliser un certain nombre de données. Le SGBD utilisé dans notre cas est MySQL.

4.1.2.2. Sécurité du système

Un certain nombre de mécanismes ont été mis en place afin d'assurer la cohérence, l'intégrité et la disponibilité des données ainsi que l'intégrité physique des fichiers sources de l'application sur le serveur.

Mesures de sécurité

- L'espace client et les applications d'administration ne sont accessibles que par un accès par identifiant et mot de passe avec un système de gestion de privilèges et de droits d'accès strict. L'application s'adapte automatiquement au privilège de l'utilisateur qui ne voit proposer sur l'interface que les fonctionnalités auxquelles il a accès ;
- Un système de journalisation automatisé de tous les accès à l'application est instauré ;
- Aucune suppression hard ou définitive n'est autorisée sur l'application. La seule suppression autorisée est une suppression logique par changement d'état d'attributs afin de faciliter les restaurations ;
- La base de données est automatiquement et systématiquement sauvegardée à l'aide d'un système de sauvegarde couplé avec une tâche planifiée ;
- L'accès direct à la base de données est restreint à au plus deux personnes. Toutes les actions sur les données et la base de données sont interfacées par l'application ;
- Les URL d'utilisation de l'API contiennent un mot de passe crypté rendant l'utilisation difficile à un opérateur humain autre que les scripts mis à cet effet.

Certaines mesures sécuritaires ont été adoptées pour garantir la confidentialité et la disponibilité de ces données à savoir :

- La sécurité de l'API ;
- L'architecture 3 tiers ;
- Le Pare-feu Laravel.

La sécurité des applications est primordiale du fait de la sensibilité des données traitées et de leurs enjeux.

- **Sécurité de l'API**

L'API de communication entre les applications, le système d'identification et la base de données est sécurisée grâce à la technologie du JWT (Json Web Token). Les routes de l'API ne sont accessibles qu'après une authentification.

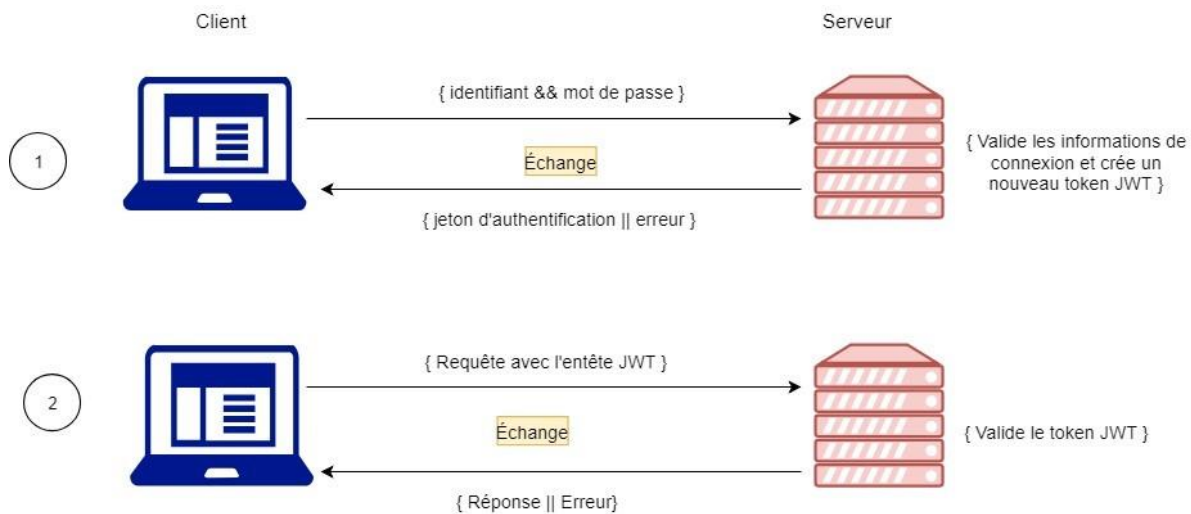


Figure 35 Processus d'authentification avec JWT Token

- **L'architecture 3 tiers**

Elle offre l'amélioration de la sécurité des données, en supprimant le lien entre le client et les données. Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer dans la couche d'accès aux données.

• Pare-feu Laravel

Le Framework Laravel utilisé, permet de mettre en place un pare-feu pour contrôler les actions des différents utilisateurs. Il contrôle les accès des utilisateurs à travers son outil de gestion de rôles. Chacun des acteurs du système n'a accès qu'à son interface appropriée après connexion.

Laravel offre aussi des mesures de protection contre certaines failles comme les failles XSS, l'injection SQL et la protection CSRF.

4.2. Présentation du système

Trois (3) espaces de travail ont été développées :

- L'espace d'administration ;
- L'espace des partenaires ;
- L'espace des clients.

Nous présenterons dans cette section la structure de ces espaces et les interfaces obtenues.

4.2.1. Structure de l'espace d'administration

La figure suivante illustre la structure de l'espace d'administration.

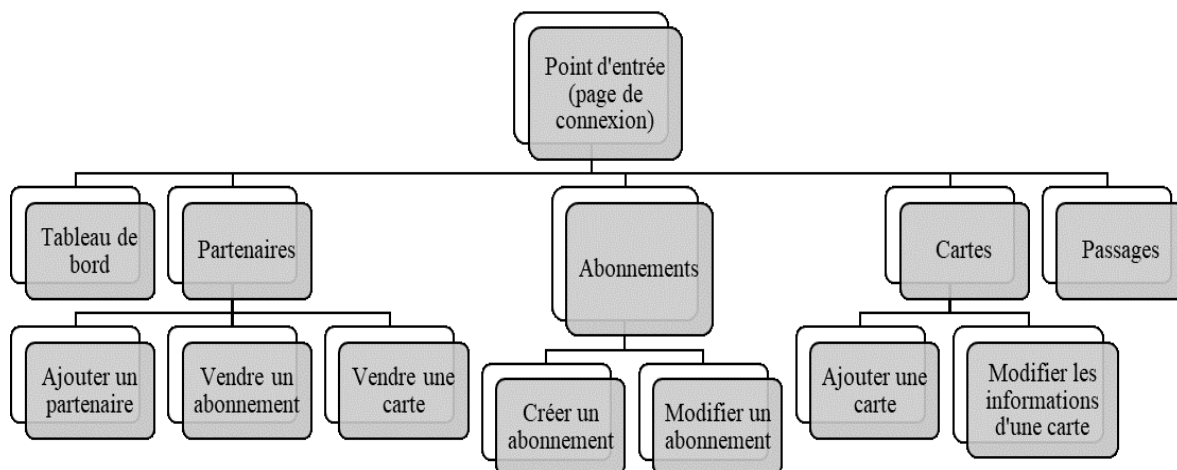


Figure 36 Structure de l'espace d'administration

4.2.2. Structure de l'espace des partenaires

Ci-dessous, une présentation de la structure de l'espace dédiée aux partenaires.

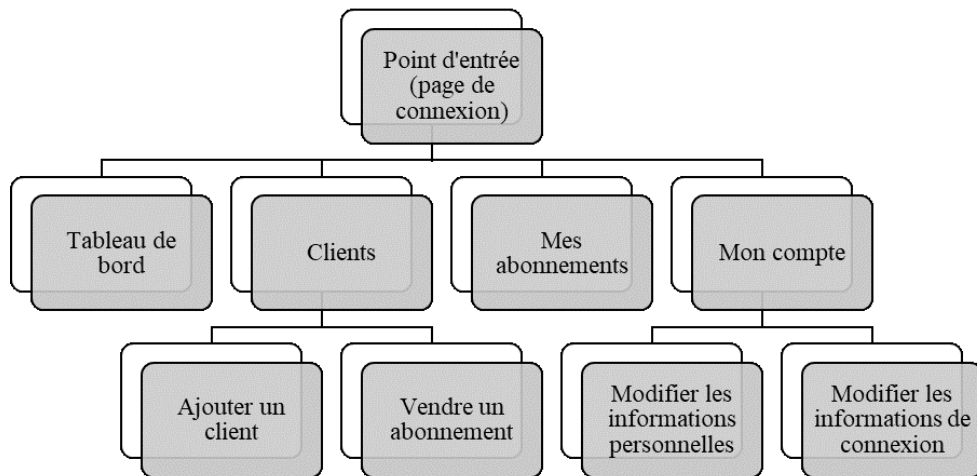


Figure 37 Structure de l'espace des partenaires

4.2.3. Structure de l'espace Client

La structure de l'application web des clients est présentée sur la figure suivante.

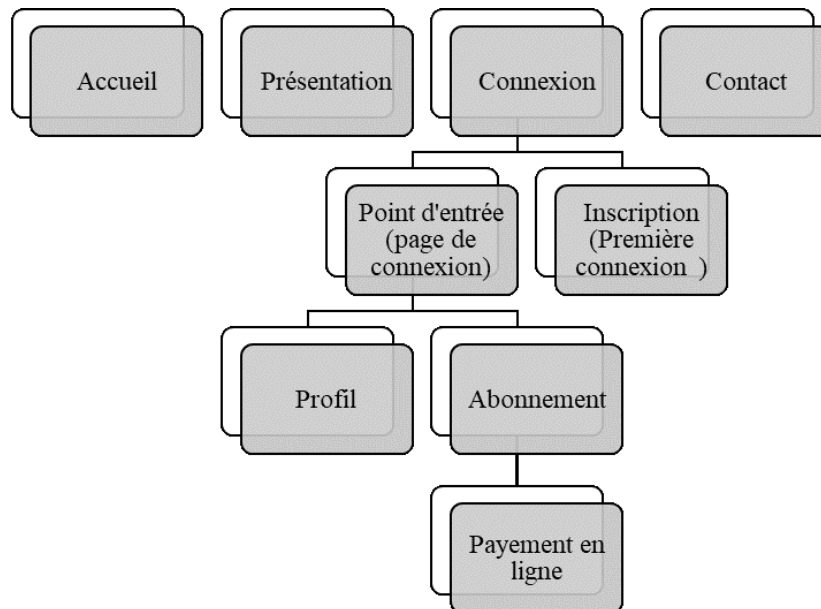


Figure 38 Structure de l'espace Client

4.2.4. Présentation de quelques interfaces de l'application

Nous présenterons quelques interfaces par espace de notre système. Les espaces requièrent toutes une option d'authentification et communiquent via la même API.

Ces espaces sont :

- L'espace d'administration ;
- L'espace des partenaires ;
- L'espace des clients.

4.2.4.1. L'espace d'administration

L'espace sera utilisée par l'institution chargée de la gestion des stations de péage. L'utilisation de l'application par l'administrateur requiert une option d'authentification.

Cet espace permet d'avoir une vue sur les opérations des partenaires et les passages aux stations de péage. L'administrateur gère également les cartes et les abonnements.

Les interfaces à présenter sont :

- La page d'authentification d'un administrateur ;
- Le tableau de bord ;
- La liste des partenaires ;
- L'interface de vente d'abonnement à un partenaire ;
- L'interface de création d'abonnement ;
- L'historique de passages.

- **Page d'authentification**

. L'interface suivante est la page d'authentification d'un administrateur.

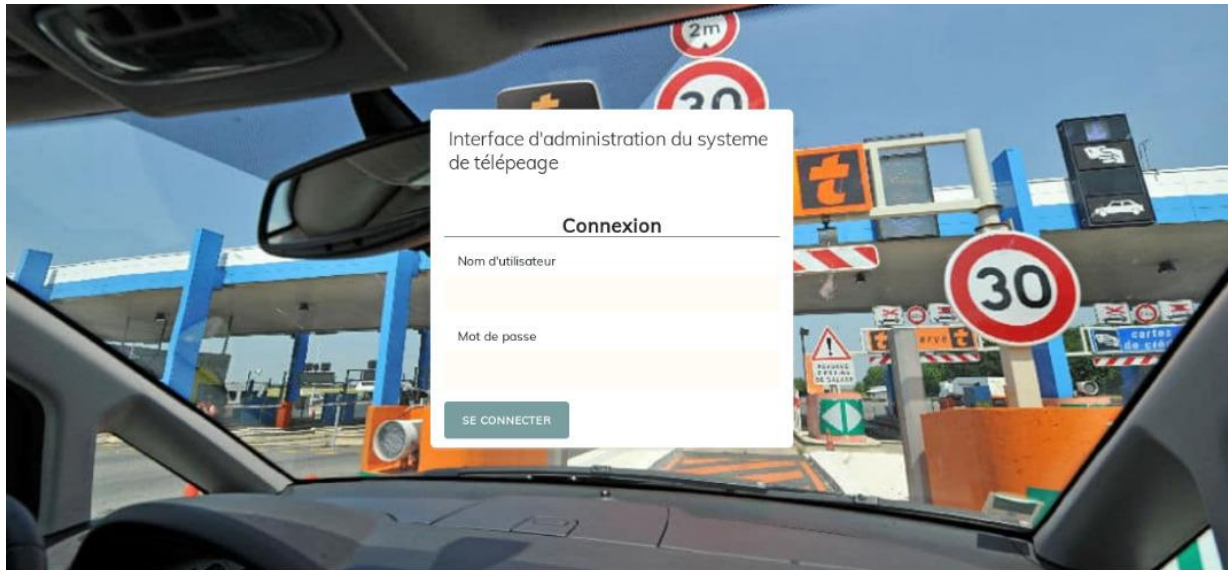


Figure 39 Page d'authentification de l'espace d'administration

- **Tableau de bord de l'espace d'administration**

L'administrateur a la possibilité d'avoir les statistiques sur les partenaires, les cartes, les ventes d'abonnements et les passages. Le tableau de bord de l'espace d'administration est le suivant.

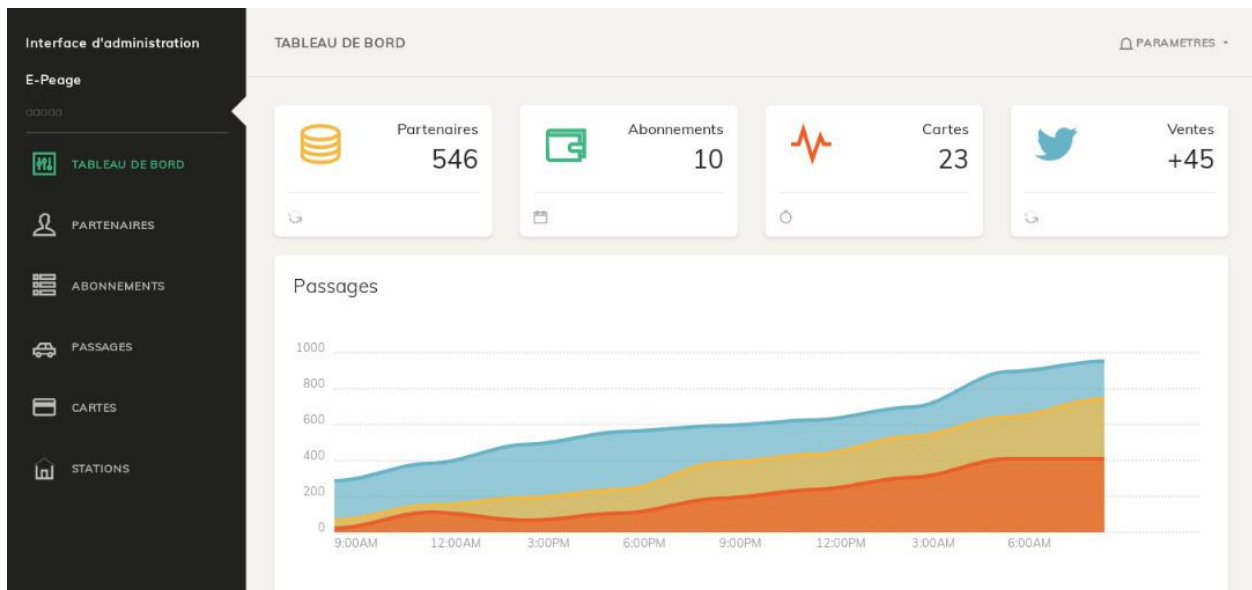


Figure 40 Tableau de bord de l'espace d'administration

- **Gestion des partenaires**

La liste des partenaires pouvant servir les clients peut être consultée par l'administrateur. L'administrateur pourra ainsi connaître les abonnements dont dispose un partenaire.

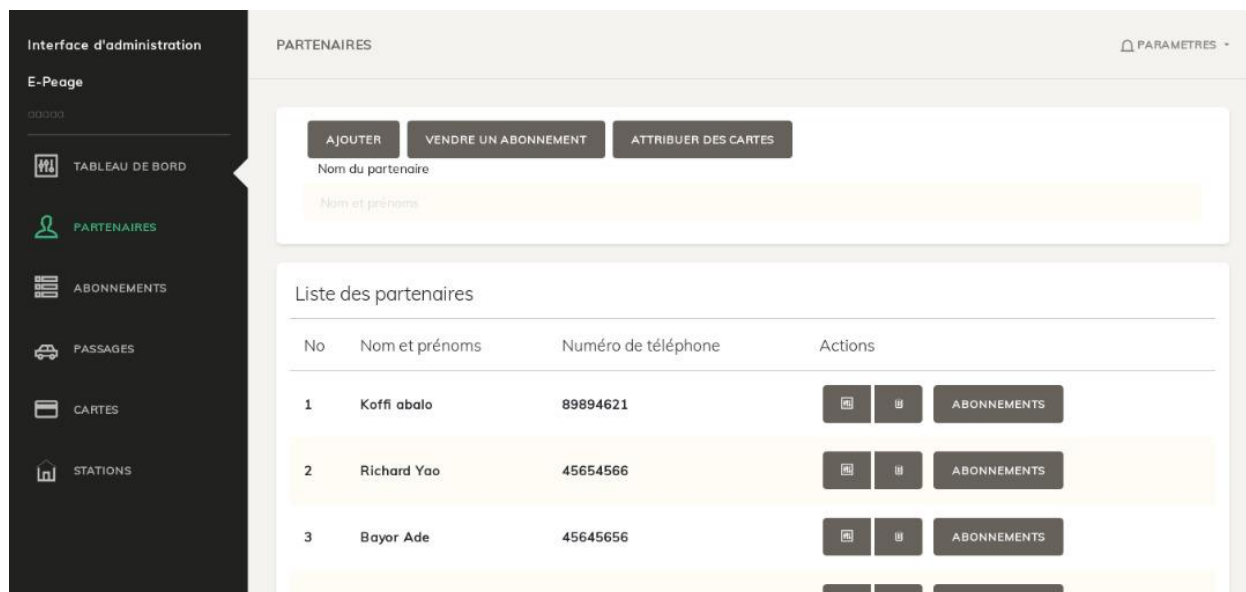


Figure 41 Affichage de la liste des partenaires

L'administrateur peut vendre des abonnements aux partenaires.

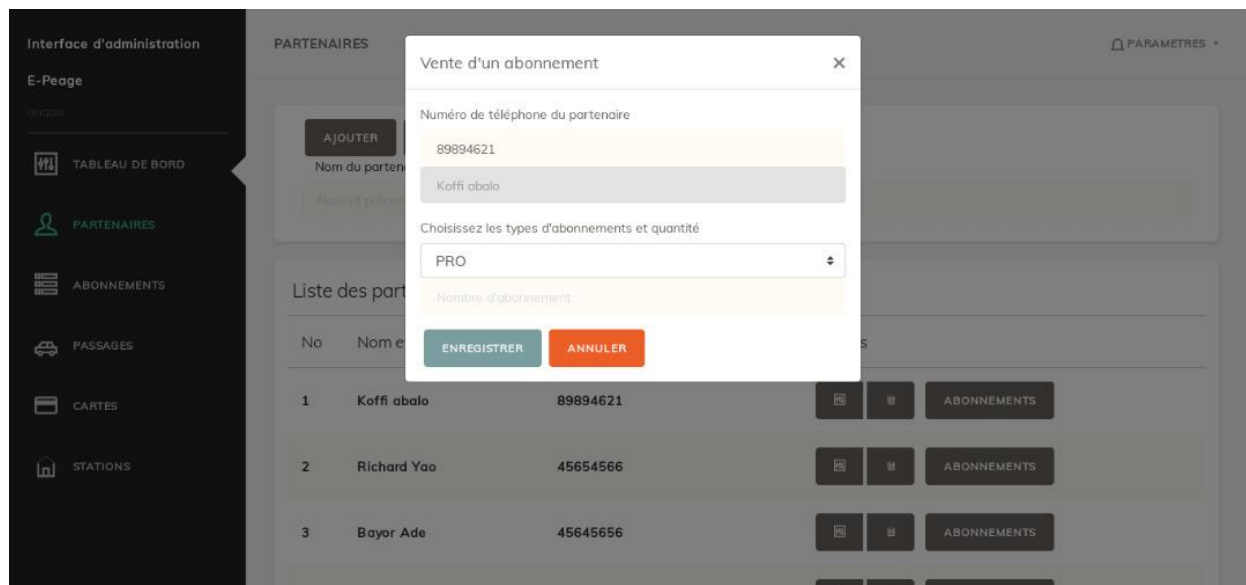


Figure 42 Interface de vente d'abonnements à un partenaire

- **Création d'abonnement**

L'interface permettant de créer un abonnement est la suivante.

The screenshot shows the 'Ajouter un type d'abonnement' modal form. The form has the following fields:

- Libellé de l'abonnement: Entrez le libellé
- Nombre de passage: 0
- Durée de validité de l'abonnement: 0
- Prix de l'abonnement: 0
- Surplus: 0

At the bottom of the modal are two buttons: 'VALIDER' (green) and 'ANNULER' (orange). In the background, a table titled 'Liste des abonnements' is visible, showing columns for 'No', 'Libellé', 'Prix (ECO)', 'Surplus', and 'Actions'.

No	Libellé	Prix (ECO)	Surplus	Actions
1	PRIVILEGE	10	6	[Edit] [Delete]
2	TOURISTE	20	0	[Edit] [Delete]
3	COMMERCIAL	500	20	[Edit] [Delete]

Figure 43 Interface de création d'abonnement

- **Ajout d'une carte RFID**

Afin d'identifier les véhicules, des cartes RFID seront enregistrées pour utilisation des clients.

L'interface suivante permet d'effectuer cette action.

The screenshot shows the 'CARTES' section of the E-Peage administration interface. It includes the following elements:

- A section titled 'Importer les cartes (Excel)' with a text input field containing 'codes.xlsx' and a 'Browse' button.
- A status message: 'Nombre de cartes actuellement enregistrées: 10000'.
- A section titled 'Rechercher une carte' with a text input field for 'Numero de carte'.
- A table with columns: 'No', 'Nom du client', and 'Numéro de la carte'.

Figure 44 Interface d'ajout d'une carte RFID

- **Historique de passages**

La page suivante donne la possibilité de s'informer sur les passages aux péages.

client	Numero de carte	Heure de passages
Koffi Abalo	6846543213	10:00
Bayor Ade	6545564546	10:10
Richard Yao	6545455564	10:30
Bayor Ade	6545564546	12:13
Bayor Ade	6545564546	15:42
Koffi Abalo	6846543213	20:03

Figure 45 Page des historiques de passages

4.2.4.2. L'espace des partenaires

L'espace sera utilisée par les partenaires de l'institution chargée de la gestion des stations de péage. Un partenaire effectuera alors des opérations après authentification.

Nous présenterons les interfaces suivantes :

- Le tableau de bord ;
- La page d'enregistrement d'un client ;
- La page présentant la liste des clients ;
- L'interface permettant de vendre un abonnement ;
- La liste des abonnements.

- **Tableau de bord de l'espace dédié aux partenaires**

Un partenaire a la possibilité d'avoir les statistiques sur ses clients, ses cartes, ses abonnements, ses ventes.

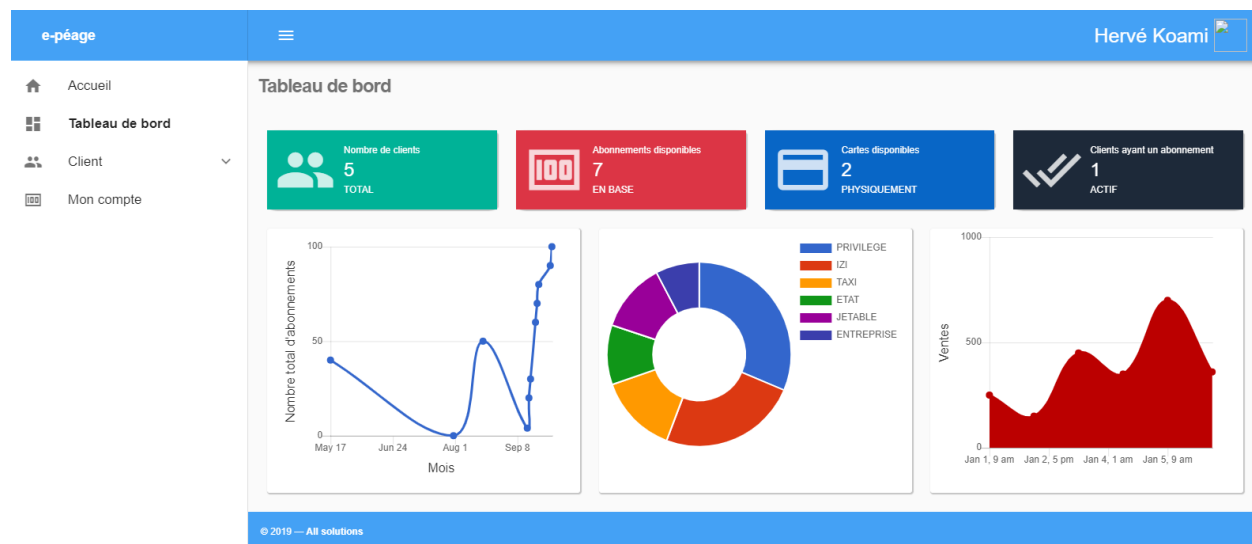


Figure 46 Tableau de bord de l'espace des partenaires

- **Gestion des clients**

Un client est enregistré par un partenaire à l'acquisition d'une carte. Ci-dessous la page permettant de remplir le formulaire d'enregistrement d'un client.

The form is titled 'Ajouter un client' and includes the following fields:

- Nom (Name)
- Prénoms (First names)
- Email
- Téléphone (Phone)
- Carte (Card)
- Statut (Activé/Désactivé) * (Status (Active/Inactive) *)

A green 'AJOUTER' (Add) button is located at the bottom right of the form.

Figure 47 Page d'enregistrement d'un client

La liste des clients peut être consultée sur la page suivante.

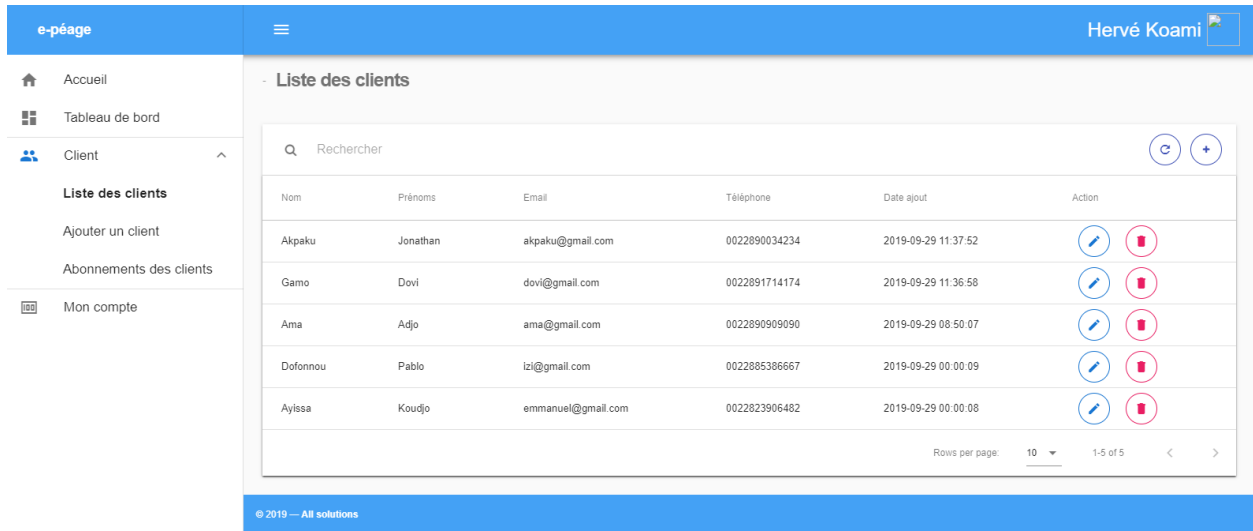


Figure 48 Page présentant la liste des clients

Le partenaire a la possibilité de vendre les abonnements à un client sur l'interface suivante.

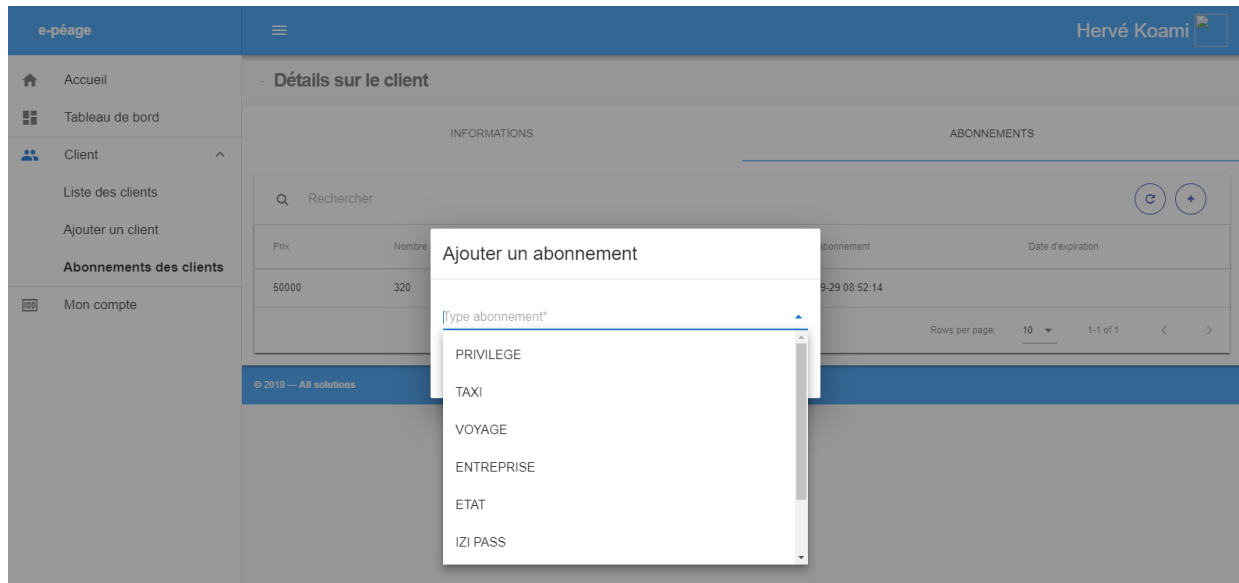


Figure 49 Interface de vente d'abonnement

- **Liste des abonnements**

La page suivante présente les abonnements dont dispose un partenaire.

Type	Nombre disponible	Date achat
PASS	8	2019-09-29 00:00:09
JETTABLE	14	2019-09-29 00:00:09
IZI PASS	13	2019-09-29 00:00:09
ETAT	8	2019-09-29 00:00:09
ENTREPRISE	6	2019-09-29 00:00:08
TAXI	13	2019-09-29 00:00:08
PRIVILEGE	15	2019-09-29 00:00:07

Figure 50 Interface des abonnements disponibles

4.2.4.3. L'espace client

L'espace client est dédiée aux conducteurs et/ou propriétaires de véhicules et de carte RFID. Ils pourront via cet espace, consulter les offres d'abonnements, l'état de leurs abonnements après connexion et effectuer des paiements d'abonnements en ligne.

Les interfaces à présenter sont :

- L'interface d'accueil du client ;
- La page de consultation de nouvelles offres ;
- L'interface de consultation des abonnements après connexion ;
- La page permettant à un client d'effectuer un paiement en ligne.

- **Interface d'accueil**

La page suivante sert d'accueil pour le client et l'informe sur les récents changements.

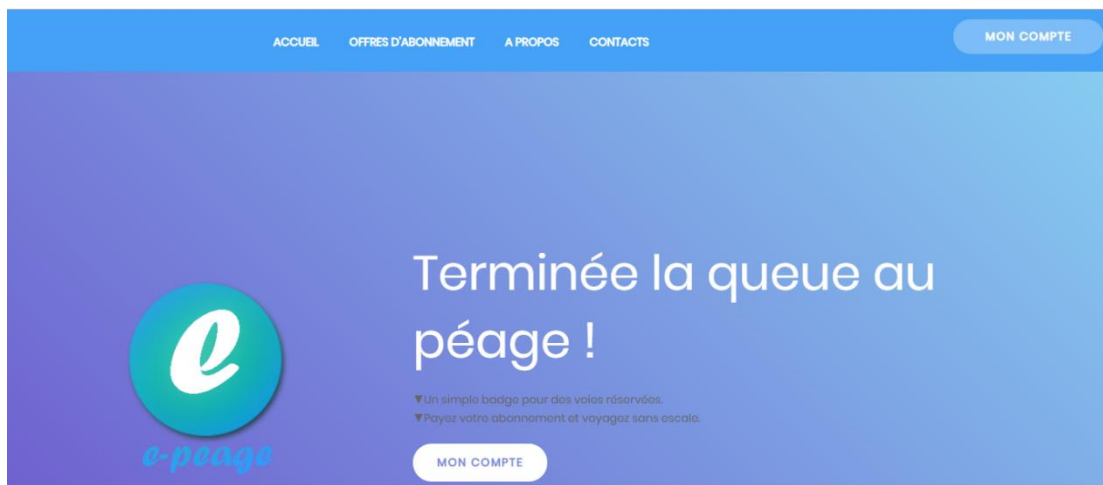


Figure 51 Interface d'accueil du client

- **Les offres d'abonnements**

L'interface qui suit présente les offres d'abonnements.

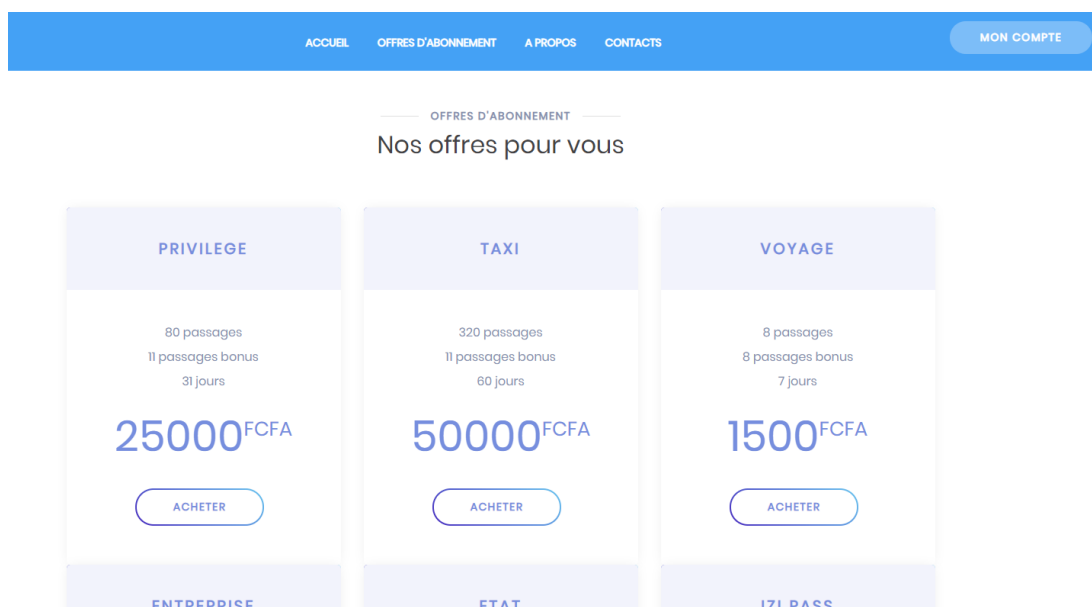


Figure 52 Interface présentant les offres d'abonnements

- **Consulter les abonnements**

La page suivante permet à un client d'avoir une vue sur ses abonnements.

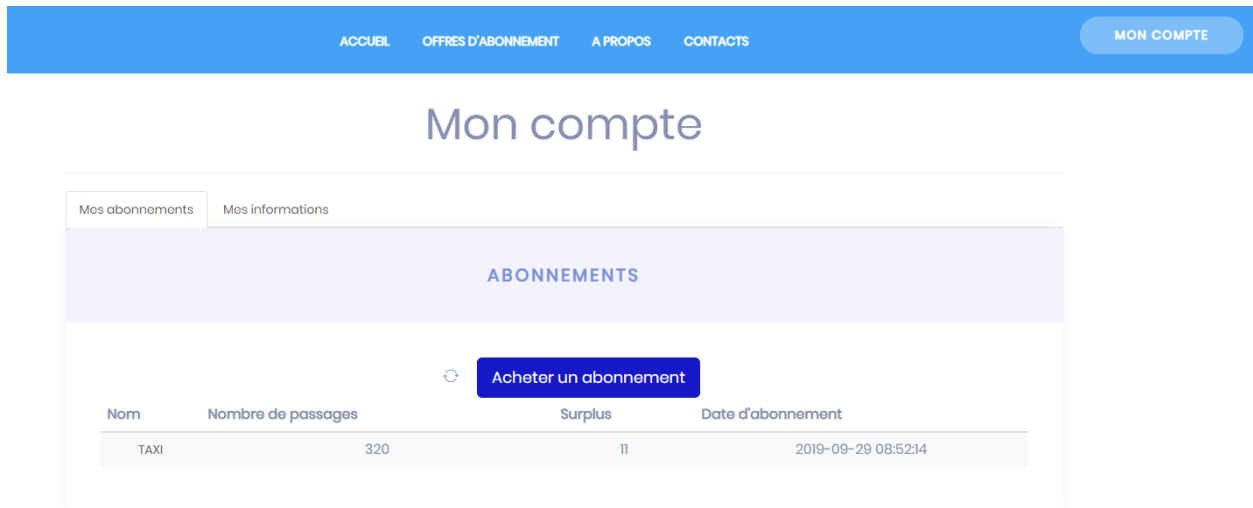


Figure 53 Interface des abonnements d'un client

- **Paiement en ligne**

Afin de permettre aux clients de payer leurs abonnements sans se déplacer, il a été développé l'interface suivante qui permet à un client d'effectuer un paiement d'abonnement à distance (en ligne).

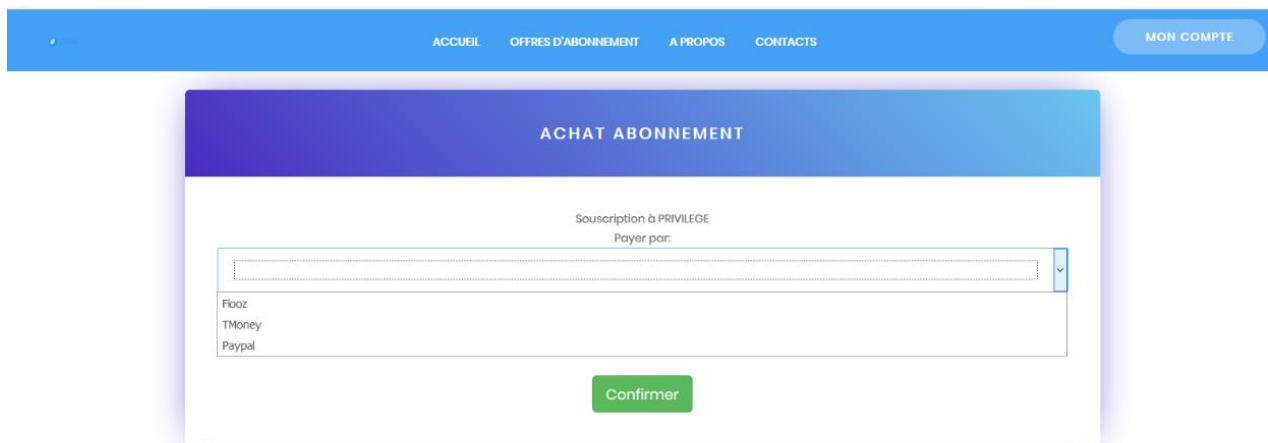


Figure 54 Interface de paiement en ligne

CONCLUSION

Ce document a présenté le travail réalisé au cours de nos dix (10) semaines de stage de fin de formation au sein de l'entreprise ALL Solutions. Le premier chapitre a permis de présenter les cadres de formation et de stage. Le chapitre suivant présente l'étude préalable du projet en passant par une étude de l'existant, sa critique et finalement une proposition de solutions face aux problèmes freinant la bonne marche des stations de péage. L'analyse et la conception de la solution proposée ont été exposées dans le troisième chapitre. Après avoir présenté la méthode d'analyse et l'outil de modélisation, une étude détaillée de la solution a été faite. Enfin le dernier chapitre du document est consacré à la réalisation et la mise en œuvre de la solution.

Les objectifs du projet ont été atteints grâce à une méthodologie claire et à une approche objet suivie d'une modélisation du système avec UML. L'ensemble des solutions a été développé dans un environnement PHP, Golang couplé à une technologie de radiofréquence, et grâce aux Framework Vue JS et Laravel, le tout relié à une base de données MySQL.

Nos travaux ont engendré un système de paiement automatique de péage routier ; système basé sur l'utilisation d'une carte d'identification par radiofréquence RFID.

Le système conçu permettra de rendre plus efficaces les processus liés à la gestion des péages car elle rend plus fluide le transport des personnes et des biens en réduisant le temps d'arrêt aux péages. Il permet également aux institutions de gestion des péages d'avoir une vision plus claire sur les passages de véhicules et de collecter des données plus fiables.

La réalisation du projet a été pour nous l'occasion de mettre en pratique les connaissances acquises au cours de notre formation. Elle nous a permis de nous immerger un peu plus dans le métier de génie logiciel, de perfectionner notre savoir-faire et d'engranger de nouvelles compétences et connaissances.

La solution mise en place pourrait bien évoluer étant donné que les technologies utilisées avancent à grand pas tous les jours. Par ailleurs, le système pourra intégrer une classification automatique des types de véhicules par ondes électromagnétiques et la communication entre les systèmes d'identification pourra se faire par un module GSM.

REFERENCES BIBLIOGRAPHIQUES

- [1] Amos ATSUVIA, Développement d'une application web d'exploitation des résultats de l'évaluation des enseignements à l'Université de Lomé, CIC-UL, décembre 2018.
- [2] « Centre Informatique et de Calcul (CIC), » [En ligne]. Available : <http://www.univ-lome.info/fr/universite/organisation/etablissements/cic>. [Accès le 01 Août 2019].
- [3] « Plan National de Développement 2018-2022, 03 Août 2019 » [En ligne]. Available : [https:// www.republictogolaise.com/pnd](https://www.republictogolaise.com/pnd) . [Accès le 01 Août 2019].
- [4] Minh Thuy LE, Thèse de doctorat de l'université de Grenoble : Contribution à la conception d'un système d'identification et de classification de véhicules par les ondes électromagnétiques, Grenoble (France), mars 2013.
- [5] Dr. Khali Persad, Dr. C. Michael Walton, et Shahriyar Hussain, « Electronic Vehicle Identification: Industry Standards, Performance, and Privacy Issues », Center for Transformation Research The University of Texas at Austin 3208 Red River, Suite 200 Austin, Texas 78705-2650, 0-5217-P2, Janvier 2007.
- [6] Enquête APPFIZ CORPORATION, « La vie d'un agent de péage, Tsévié (Togo), Février 2019 », 2019.
- [7] World Bank, « Toll Collections Systems: Technology Trend Impact on PPP's and Highway's Transport, Jack Opiola, April 2006. », 2006.
- [8] [En ligne]. Available : <https://fr.wikipedia.org/wiki/sgbd> [Accès le 09 Août 2019]
- [9] [En ligne]. Available : <https://fr.wikipedia.org/wiki/pib> [Accès le 13 Août 2019].

- [10] [En ligne]. Available : <http://sametmax.com/quest-ce-quune-api/>. [Accès le 09 Août 2019]
- [11] [En ligne]. Available : <https://www.supinfo.com/articles/single/2422-mise-place-une-api-restful-avec-symfony>. [Accès le 09 Août 2019].
- [12] [En ligne]. Available : <https://www.supinfo.com/articles/single/2422-mise-place-une-api-restful-avec-symfony>. [Accès le 09 Août 2019]
- [13] [En ligne]. Available : <https://www.votresalaire.org> [Accès le 10 Août 2019].
- [14] [En ligne]. Available : <https://www.irisa.fr/triskell/members/pierre-alain.muller/teaching/demarche>. [Accès le 11 Août 2019].
- [15] [En ligne]. Available : <https://www.lucidchart.com/pages/fr/quest-ce-que-le-langage-de-modelisation-unifie>. [Accès le 12 Août 2019].
- [16] [En ligne]. Available : <https://www.getapp.fr/software/91063/draw-dot-io>. [Accès le 12 Août 2019].
- [17] [En ligne]. Available : https://edutechwiki.unige.ch/fr/Visual_studio_code . [Accès le 13 Août 2019].
- [18] [En ligne]. Available : <https://developer.mozilla.org/fr/docs/Web/CSS>. [Accès le 24 Octobre 2018]
- [19] [En ligne]. Available : <https://fr.wikipedia.org/wiki/Vue.js> [Accès le 13 Août 2019]
- [20] [En ligne]. Available : <https://www.anthedesign.fr/developpement-web/php-avec-laravel/>. [Accès le 13 Août 2019].
- [21] [En ligne]. Available : <https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4678736-comment-fonctionne-une-architecture-mvc>. [Accès le 13 Août 2019].

- [22] [En ligne]. Available : <https://fr.wikipedia.org/wiki/jwt>. [Accès le 14 Août 2019].
- [23] [En ligne]. Available : [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework)). [Accès le 14 Août 2019]
- [24] [En ligne]. Available : <https://fr.wikipedia.org/wiki/orm>. [Accès le 14 Août 2019]
- [25] [En ligne]. Available : <https://developer.mozilla.org/fr/docs/Web/Guide/AJAX>. [Accès le 14 Août 2019]
- [26] [En ligne]. Available : <https://fr.wikipedia.org/wiki/rfid> [Accès le 13 Août 2019].
- [27] [En ligne]. Available : <https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1959710-decouvrez-mysql> [Accès le 14 Août 2019].

TABLE DES MATIERES

LISTE DES PARTICIPANTS DU PROJET	vii
LISTE DES TABLEAUX.....	viii
LISTE DES FIGURES	ix
GLOSSAIRE.....	xii
INTRODUCTION	1
CHAPITRE 1 PRÉSENTATION DES CADRES DE FORMATION ET DE STAGE.....	3
1.1. Présentation du cadre de formation : CIC-UL [1] [2].....	4
1.1.1. Historique	4
1.1.2. Les objectifs.....	4
1.1.3. Les activités	5
1.1.4. Les perspectives.....	6
1.1.5. Organisation administrative.....	6
1.2. Présentation du cadre de stage : Entreprise ALL Solutions.....	7
1.2.1. Description de l'entreprise.....	7
1.2.2. Mission de l'entreprise	8
1.2.3. Les activités	9
1.2.4. Organigramme structurel de l'entreprise	10
CHAPITRE 2 ÉTUDE PRÉALABLE DU PROJET	12
2.1. Présentation du sujet	13
2.2. Problématique du sujet.....	13
2.3. Intérêt du sujet.....	14
2.3.1. Objectifs.....	14
2.3.1.1. Objectifs généraux	14
2.3.1.2. Objectifs spécifiques	14

2.3.2.	Résultats attendus	15
2.4.	Etude de l'existant.....	15
2.4.1.	Le système de gestion des véhicules	15
2.4.2.	Le système de back office	15
2.5.	Critique de l'existant.....	16
2.5.1.	Le système de gestion des véhicules	16
2.5.2.	Le système de back office	16
2.6.	Proposition de solutions.....	17
2.6.1.	Le système de paiement de péage « stop-and-go » amélioré	17
2.6.2.	Le système de paiement automatique ou système de télépéage	17
2.7.	Choix de la solution	18
2.8.	Spécification de la solution.....	20
2.9.	Evaluation technique des livrables.....	20
2.9.1.	Le système d'identification de véhicules via RFID.....	20
2.9.2.	L'application d'administration	21
2.9.2.1.	Une application desktop.....	21
2.9.2.3.	Une application web réalisée avec un CMS	22
2.9.2.4.	Une application web depuis zéro	23
2.9.2.5.	Une application web avec un Framework.....	23
2.9.3.	L'application de gestion des processus des partenaires.....	24
2.9.4.	L'application web : l'espace client.....	24
2.9.5.	L'API de communication	24
2.10.	Choix des livrables	27
2.10.1.	Le système d'identification de véhicules via RFID	27
2.10.2.	L'application d'administration.....	27

2.10.3.	L'application de gestion des processus des partenaires	27
2.10.4.	L'application web : l'espace client	28
2.10.5.	L'API de communication.....	28
2.11.	Etude et choix de la base de données	28
2.11.1.	Présentation des bases de données	28
2.11.2.	Choix de la base de données	31
2.12.	Evaluation financière des modules de la solution	31
2.13.	Evaluation du coût total de la solution	32
2.14.	Planning de réalisation	32
CHAPITRE 3	ANALYSE ET CONCEPTION	34
3.1.	Présentation de la méthode d'analyse	35
3.1.1.	Présentation du modèle itératif et incrémental [14].....	36
3.1.2.	Présentation du langage de modélisation UML [15]	37
3.1.2.1.	Diagrammes UML structurels.....	38
3.1.2.2.	Diagrammes UML comportementaux	39
3.2.	Présentation de l'outil de modélisation [16]	39
3.3.	Etude détaillée de la solution	40
3.3.1.	Le diagramme de contexte statique	41
3.3.2.	Le diagramme de déploiement	42
3.3.3.	Le diagramme des composants.....	43
3.3.4.	Le diagramme des paquetages	44
3.3.5.	Les diagrammes de cas d'utilisation.....	45
3.3.5.1.	Cas d'utilisation du paquet « Administration »	45
3.3.5.2.	Cas d'utilisation du paquet « Partenaire »	47
3.3.5.3.	Cas d'utilisation du paquet « Client ».....	48

3.3.5.4. Description textuelle de quelques cas d'utilisation.....	49
3.3.6. Le diagramme des classes.....	51
3.3.7. Les diagrammes d'activités	52
3.3.8. Les diagrammes des séquences	56
CHAPITRE 4 RÉALISATION ET MISE EN ŒUVRE.....	58
4.1. Mise en œuvre.....	59
4.1.1. Matériel et logiciels utilisés	60
4.1.1.1. Environnement de programmation	60
4.1.1.2. Langages de programmation.....	60
4.1.1.3. Technologies et Framework utilisés	63
4.1.1.4. Les serveurs logiciels utilisés.....	67
4.1.1.5. Le matériel de développement	69
4.1.2. Architectures et sécurité du système	70
4.2. Présentation du système	73
4.2.1. Structure de l'espace d'administration	73
4.2.2. Structure de l'espace des partenaires.....	74
4.2.3. Structure de l'espace Client.....	74
4.2.4. Présentation de quelques interfaces de l'application.....	75
4.2.4.1. L'espace d'administration.....	75
4.2.4.2. L'espace des partenaires	79
4.2.4.3. L'espace client	82
CONCLUSION.....	85
REFERENCES BIBLIOGRAPHIQUES.....	86
TABLE DES MATIERES	89